Fraunhofer MEVIS

Universität Bremen

Dissertation
submitted to the
Faculty of Physics
University of Bremen, Germany
for the degree of
Doctor of Natural Sciences (Dr. rer. nat.)

Motion Tracking for Medical Applications using Hierarchical Filter Models

Sven Rothlübbers

Referees:
Prof. Dr. Matthias Günther
Prof. Dr. Gabriel Zachmann

Date of Defense:
2018-11-01

The 'Fraunhofer MEVIS' and 'Universität Bremen' logo a courtesy of the respective owners.

Abstract

A medical intervention often requires relating the treatment to the situation, which it was planned on. This is likely to differ from the current situation because the patient may have moved or changed pose. A typical cause for this is breathing motion. In order to execute the treatment unaffected by unwanted effects of motion, the positional difference must be accounted for and compensated. Real-time imaging allows for the detection of motion, but demands on-line interpretation of the incoming image data. Such task may be carried out by a tracking algorithm. With increasing demand for automation, there is a need for autonomous setup and execution. This leads to both interest in the knowledge of the interplay of algorithmic parameters on a low level and learning processes for transformations on high level.

In this thesis a hierarchical, Particle Filter based tracking algorithm is developed. Using a hierarchical tree representation, it allows for a coarse-to-fine optimization and to differentiate independent motion in different levels of detail on-line. The work consists of three major sections.

The initial part is concerned with a theoretical description of the individual nodes in such hierarchy. The properties of a limited Particle Filter model are investigated. Limitations include a purely diffusive state estimation and simplifying assumptions about the measurement process. Using different modeling approaches, properties of such algorithms are derived and approximated, such that a parametrization for applications can be derived.

The second section investigates the formulation for application to medical image data. Transformations and appearance of the data are described by a hierarchical tree. Each node in the tree represents a component of motion in the data. The state space dynamics of each node are governed by an algorithm presented in the first section. A sparse description for typical landmarks in medical image data is presented. A static tree model with two levels is developed and investigated on different image data. On artificial motion data, the basic function is evaluated. Real-time applications are then demonstrated on cases of 2D ultrasound, as well as 2D and 3D magnetic resonance imaging data.

The third section generalizes the description of a tree. It introduces the notion of 'association' between landmarks and nodes. The graph itself is derived from the data: processes for node generation and maintenance using clustering and sequential reinforcement are introduced. These allow for on-line adaptation to the underlying structure of the data. The individual processes are analyzed on artificial data. Finally, the function of the full algorithm is demonstrated on 2D MRI data of abdominal breathing motion.

The work presents novel means for unsupervised setup and real-time operation of a tracking algorithm within a clinical treatment setup.

Contents

Introduction **2**

1. Background **6**
1.1. Physiological Motion and Medical Treatment . . . 6
1.1.1. Anatomy . . . 7
1.1.1.1. Liver . . . 7
1.1.1.2. Prostate . . . 9
1.1.2. Characteristics of motion . . . 9
1.1.2.1. Respiratory Motion . . . 10
1.1.2.2. Cardiac Motion . . . 11
1.1.3. Non-Invasive, Focal Treatment Techniques . . . 12
1.1.3.1. Radiation Therapy . . . 12
1.1.3.2. High Intensity Focused Ultrasound . . . 13
1.1.4. Motion Management for Non-Invasive Treatment . . . 16
1.1.4.1. Controlling Motion . . . 17
1.1.4.2. Adapting to Motion . . . 17
1.1.4.3. Image Guided Motion Compensation for FUS . . . 19
1.1.5. Characteristics of Medical Imaging Techniques . . . 19
1.1.5.1. Magnetic Resonance Imaging . . . 20
1.1.5.2. Ultrasound Imaging . . . 22
1.2. Video Tracking and Image Registration . . . 25
1.2.1. State Encoding . . . 27
1.2.2. Appearance . . . 27
1.2.3. Transformation . . . 30
1.2.4. Statistical Modeling . . . 32
1.2.5. Prediction . . . 33
1.2.6. Optimization . . . 35
1.2.7. Data Association . . . 36
1.3. Recursive Bayesian Estimation . . . 40
1.3.1. Bayes' Theorem . . . 41
1.3.2. Sequential Update . . . 42
1.3.3. Kalman Filter . . . 44
1.3.4. Monte Carlo Sampling . . . 45
1.3.5. Particle Filter . . . 47
1.3.6. Particle Filter for Articulate Pose Estimation . . . 49

2. Gaussian Transition Filter Model **52**
2.1. Model . . . 53
2.1.1. Algorithm and Experiment . . . 53
2.1.2. Fully Gaussian Reference Model . . . 56
2.1.3. Sampling Stability . . . 57
2.1.3.1. Critical Step Approximation . . . 58
2.1.3.2. Approximative Analytic Solutions . . . 59
2.1.4. Residual Errors . . . 62
2.1.5. Noise . . . 64
2.1.5.1. Multi State Noise Model . . . 65
2.1.5.2. Two State Noise Model . . . 66
2.1.5.3. Transition point . . . 67
2.2. Computer Experiments . . . 68
2.2.1. Sampling Stability . . . 69
2.2.2. Residual Error . . . 70
2.2.3. Noise . . . 71
2.2.4. Implementation . . . 71
2.3. Results . . . 71
2.3.1. Sampling Stability . . . 71
2.3.2. Residual Error . . . 75
2.3.3. Noise . . . 77
2.4. Discussion . . . 79
2.4.1. Findings . . . 79
2.4.2. Parametrization Scheme . . . 81
2.4.3. Conclusion . . . 83
2.5. List of variables . . . 84

3. Fixed Hierarchy Model **86**
3.1. Modeling . . . 87
3.1.1. Hierarchical Transformation Model . . . 87
3.1.1.1. State and Transformation Space . . . 88
3.1.1.2. Estimates . . . 93
3.1.1.3. Temporal Evolution of States . . . 94
3.1.2. Contrast Based Appearance Model . . . 96
3.1.2.1. Reduced, Transformable Landmark Model . . . 96
3.1.2.2. Sparse, Gradient Based Description . . . 98
3.1.3. EPI Liver Feature Detector . . . 100
3.2. Experiments . . . 102
3.2.1. Simple Breathing Motion Model . . . 102
3.2.2. Sparse, Gradient Based Landmark Model . . . 105
3.2.3. 2D MRI Data . . . 106
3.2.4. 2D US Data . . . 108
3.2.5. 3D MRI Data . . . 111

3.3. Results . . . 114
3.3.1. Simple Breathing Motion Model . . . 114
3.3.2. Sparse, Gradient Based Landmark Model . . . 115
3.3.3. 2D MRI Data . . . 117
3.3.4. 2D Ultrasound Image Data . . . 119
3.3.5. 3D MRI Data . . . 121
3.4. Discussion . . . 122
3.4.1. Simple Breathing Motion Model . . . 122
3.4.2. Sparse, Gradient Based Landmark Model . . . 123
3.4.3. 2D MRI Data . . . 124
3.4.4. 2D US Data . . . 125
3.4.5. 3D MRI Data . . . 127
3.4.6. Common Findings . . . 127
3.4.7. Conclusion . . . 129
3.5. List of variables . . . 130

4. Dynamic Hierarchy Model 132
4.1. Model . . . 133
4.1.1. Node-to-Landmark Association . . . 134
4.1.2. Association Clustering, Introduction of new Child Nodes . . . 139
4.1.3. Association Rearrangement Between Siblings . . . 144
4.1.4. Full Algorithm . . . 148
4.2. Experiments . . . 150
4.2.1. Evaluation Conditions . . . 150
4.2.2. Artificial Data Setups . . . 152
4.2.3. 2D MRI Data Setups . . . 153
4.3. Results . . . 154
4.3.1. Artificial Data Setups . . . 154
4.3.1.1. Clustering . . . 154
4.3.1.2. Rearrangement . . . 158
4.3.1.3. Joint Algorithm . . . 161
4.3.2. 2D MRI Data Setups . . . 162
4.3.2.1. Rearrangement . . . 162
4.3.2.2. Joint Algorithm . . . 163
4.4. Discussion . . . 169
4.4.1. Association Clustering . . . 169
4.4.2. Association Rearrangement . . . 170
4.4.3. Impact of Noise . . . 171
4.4.4. Joint Algorithm . . . 173
4.4.5. Hierarchical Model . . . 174
4.4.6. Common Findings . . . 177
4.5. List of variables . . . 180

5. Discussion **182**
5.1. Contents and Relation to Literature 182
5.2. Application 188
5.3. Outlook 191
5.4. Conclusion 193

A. Further Material **196**

List of Figures **200**

List of Tables **203**

List of Algorithms **203**

Bibliography **204**

Acknowledgments **228**

Introduction

In medicine of the 20th and 21st century, various imaging techniques play an important role. Common modalities such as computed tomography (CT), magnetic resonance (MR) and ultrasound (US) imaging reveal spatially resolved data from the inside of the human body. The information they assist in gaining leads to immediate advantages in diagnosis and treatment. In the treatment situation in particular, they help the surgeon navigate inside a possibly complex three dimensional environment. At the same time, tools for intervention itself become less invasive and more automatized. With the former, the requirement for imaging techniques increases further. The latter leads to the inevitable need of machines interpreting the image data instead of a human operator. The gradually progressing automation of medical intervention demands algorithms, which are not only able to return a precise result, but also are easy to configure by a user and operate in real-time.

In a medical intervention, positional knowledge of the treatment region is a key to success. Particularly, in *non-invasive* intervention, which does not require mechanical penetration of the skin of the patient, support for positioning is required. Examples are the treatment of cancerous tumors by means of radiation therapy [Bas+12] or focused ultrasound [Ken05]. These methods destroy target structure by transferring energy through the surrounding tissue and disposing it within the target region. In order to adjust the targeting of the treatment system from the outside of the body, imaging is required. By imaging the target region, a relation between the imaging system and the target can be established: In a planning image, structures of interest are annotated. This annotation represents a physically correct location within the imaging system until either the system or the target are moved or changed. As long as the correspondence is valid, the operator can use the imaging system's coordinate system to navigate the treatment. It becomes invalid as the result of motion or deformation. The target can then be imaged again to regain the correspondence. The current and reference image's contents are used to establish a spatial correspondence and to transfer the annotation from the planning image to the current situation.

Establishing such a correspondence is one major purpose of medical image registration: Based on observations in medical images, translation and deformation of physical objects or the imaging system are inferred. Traditionally, medical image registration is concerned with singular, pairwise matching of arbitrary types of image data. A wide variety of approaches exists that aim to thoroughly analyze occurring deformations for different modalities, dimensionalities and target organs [Vie+16]. Under guidance of the operator, they link intervention planning to the current situation.

However, a detailed investigation of deformation is time consuming. Thus, particularly in radiation therapy, measures have been undertaken to assure that the established correspondence endures. Physiological motion, such as respiratory and cardiac motion, can easily spoil the established correspondence. Breathing in particular can cause displacements in the order of centimeters in the abdomen [Kea+06]. Early approaches therefore aimed for the control of motion by physically restricting it. This is potentially unpleasant to the patient. Moreover, since the update via imaging is infrequent if repeated at all, the procedure gives away control over the current situation.

Today, real-time imaging offers direct access to sequential updates of motion information. It can be used to adapt the ongoing treatment as it progresses. Not only the physical restriction to the patient can be loosened this way, but also the target region is under permanent control. However, such approach requires fast evaluation of sequentially incoming data. Aside from highly optimized registration algorithms, this leads to the field of video tracking [MC11]. Tracking algorithms use prediction and approximating assumptions in their components in order to achieve real-time applicability. In both registration and tracking, a trade-off exists not only between modeling detail, computational cost, result quality, but also the effort required for initialization.

A particular difficulty in organ motion is the presence of anisotropic tissue properties, which finds its most extreme expression in discontinuous displacement fields. The discontinuities are often encountered at the interfaces of different organs. The most notable example is the sliding motion of inner organs as a result of respiration [DST15]. While deformations within individual organs can be approached by many registration algorithms, matching multiple sliding and deforming organs can pose difficulties.

For a single organ, the transformation can be constrained by statistical population based model [McC+13; Pre+14]. The advantage of such approach is, that it reduces the transformation's dimensionality and therefore allows for fast computation and prediction. A drawback is that it requires a setup phase which defines correspondences between model and image data. Further, it can not describe motion unaccounted for — i.e. that which has not been included in the statistical model, but also that of other organs.

For multiple organs, registration methods employ generic deformation fields, which allow for discontinuities. These may have to be provided beforehand as the result of an organ segmentation [XCX11; Sch+12; Der+15]. Direct integration of motion segmentation into the optimization process is also described in the literature [KFC11; Kir+16; Pap+18]. Originating from image registration, these approaches recover a dense deformation field from image pairs. They take minutes for computation, as they produce more information than necessary for an intervention. Furthermore, they are not reliant on and therefore not designed to integrate sequential data.

The aim of this thesis is the establishment of an automated, real-time tracking algorithm for the analysis of physiological motion. The development of a generic, hierarchical tracking algorithm is presented in a bottom-up approach: It will describe the automation of the configuration of generic tracking components. These will then be joined into

a real-time application to medical image data. Finally, a process will be introduced that allows to learn detailed transformation properties from the stream of incoming data.

In this work a hierarchical, particle-filter based tracking algorithm is developed. The development is centered around, but ultimately not restricted to medical image data of respiratory motion. Using a hierarchical tree representation, it will allow for a coarse-to-fine optimization and to differentiate independent motion in different levels of detail. The initial part (*Gaussian Transition Filter Model* (2)) is concerned with a theoretical examination of a limited Particle Filter model. Limitations include a purely diffusive state estimation and simplifying assumptions about the measurement process. This model is intended for the later application as a node of the hierarchical graph. Using different approximative models, properties of such node are derived and approximated, such that a parametrization for applications can be derived.

Second part (*Fixed Hierarchy Model* (3)) adds the formulation of hierarchical tracking schemes and their application to medical image data. Initially, a static tree model with two levels is developed and investigated on different image data. Starting from artificial motion data, cases of 2D ultrasound, as well as 2D and 3D magnetic resonance imaging data are examined. In each case the algorithm is designed such that a real-time application is feasible.

The third part (*Dynamic Hierarchy Model* (4)) generalizes the description of a tree, introducing an association between features and nodes, such that the graph itself is learned from the data. Processes for creation and maintenance using clustering and sequential reinforcement are introduced. Consequences of the individual processes are analyzed on artificial data, before a qualitative assessment of the function is obtained from 2D MRI data.

These central parts in themselves are organized such that they host sections for experiments, results and an individual discussion. An overview of relevant parameters is available at the end of each chapter. Before beginning the first part, the background (*Background* (1)) for the problem at hand and approaches to its solution is laid out in detail. The work then closes with a finalizing discussion (*Discussion* (5)).

Publications

As a precursor to this work, the challenge contribution [Rot+14] describes an approach to Particle Filter based tracking of single vessel cross sections in ultrasound sequences. It has also been released as part of the summarizing work [Luc+15]. A particular implementation of the algorithm as described in chapter *Fixed Hierarchy Model* (3) is used as an element in [Mih+16] and [Sch+17].

1. Background

This chapter is intended to provide an overview into the anatomical and technical background of this work. It is subdivided into three sections. The initial section will cover general aspect of the topic, providing an introduction into the problem at hand. From medical perspective, this includes description of anatomy causing motion, and being subject to treatment. The characteristics of physiological motion are presented, as well as non-invasive treatment and necessary motion compensation techniques. These will be proceeded by a section focusing on the wider area of the applied tools; namely algorithms from video tracking and medical image registration. The final section dives into further detail, which is laid upon the application of sequential Monte Carlo methods to the problem of hierarchical pose recovery.

1.1. Physiological Motion and Medical Treatment

Medical conditions may concern any part of the human body. Some affect the whole system, such as infections. Others originate and reside in very distinct locations, such as cancerous tumor and bone fracture. Latter group is commonly treated in a localized therapy, which requires knowledge about the exact treatment location.

As long as the outer body surface is affected, localization is straightforward. Treatment and follow-up care do not require dedicated methods for localization. However, once the body insides are concerned, this changes. Treatment now requires direct access to the affected region — it becomes *invasive*, meaning that mechanical access is gained through surgery. The procedure is an additional burden to the patient and carries the risk of damaging healthy tissue. Some procedures carry the risk of spreading pathogens from the treatment region. In order to avoid this, an increasing number of medical imaging techniques have been developed. These have multiplied options for diagnosing and localizing the treatment target without establishing a mechanical access open intervention. Simultaneously, therapy tools have been reduced in size in order to lessen stress by an invasive treatment.

For select sets of medical conditions, *non-invasive* therapy devices exist which no longer require mechanical access in order to treat targets within the body through the intact skin. For instance, ionizing radiation and mechanical waves can be applied from outside the body and target locations inside. As no direct access to the target is established, these methods heavily rely on imaging techniques for localization. These serve the purpose of establishing a connection between diagnosis and treatment situation as well as maintaining it.

Latter task requires automatic analysis of image data for integrating new position information into the treatment. Image registration and tracking serve the purpose of extracting such motion or deformation data. In this work, an image based tracking system is established.

This first section will initially provide an overview over the related anatomy and associated motion. The related body parts are depicted in Figure 1.1. In this work, the lungs, the heart and the bladder are sources of motion or deformation. The motion and deformation of the liver, the kidneys and the prostate will be observed in image data.

After presenting options for particular types of treatment, strategies for handling the interference of motion with that treatment are reviewed. With the prospect of an image based motion compensation system, a last subsection present the characteristics of imaging techniques used as data sources in this work.

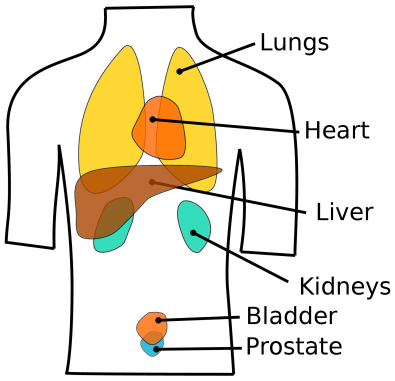


Figure 1.1.: Localization of organs and glands relevant in this work (male human).

1.1.1. Anatomy

The organs relevant in this work are both those that are subject to treatment and those which interfere with the treatment by causing temporal displacements. This initial subsection begins with a presentation of two organs of the former category. The example of the prostate will serve as a case for non-regular deformation occurring during intervention, whereas liver motion is heavily influenced by repetitive breathing motion.

1.1.1.1. Liver

Anatomy The liver is a central organ of the metabolic system. It hosts several vital functions, such as detoxification, synthesis of essential proteins and bile production.

The liver resides below the lungs and above the intestinal organs. It is separated from the lungs via the diaphragm and not fixed to the chest wall. The peritoneum, a sliding boundary tissue, assures that the liver may pass along the chest wall smoothly during breathing. Structurally, the liver is separated into *lobes*. There is a structural division

between left and right lobe. Further separation into lobes of smaller compartments is possible, even though no functional difference exists between lobes. The purpose of separating the liver into lobes is the orientation within an organ which is otherwise sparse of landmarks.

Two major vessels (Fig. 1.2) provide blood flow into the liver. Oxygen enriched blood from the heart is brought in by the *hepatic artery*. Blood enriched by nutrients, metabolic products and toxins is brought in from gastrointestinal tract, spleen and gall bladder via the *portal vein*. Both vessels branch out into vascular trees inside the liver volume and ultimately feed into vessels of the *central hepatic vein*, capturing the outflow of blood from the liver.

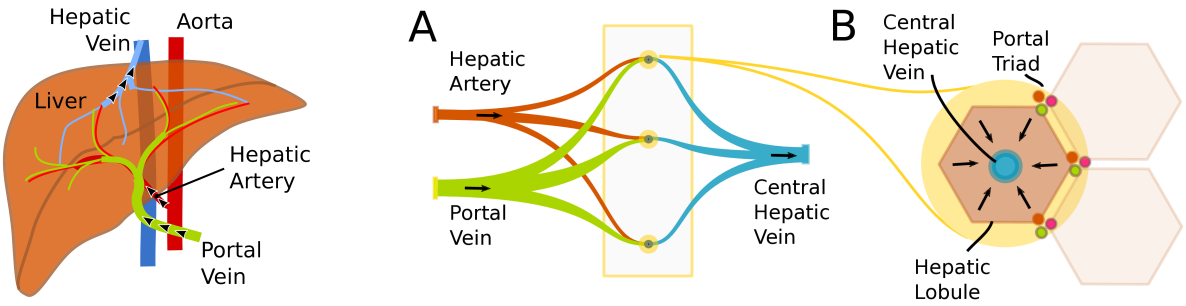


Figure 1.2.: Left: The liver and its major vessels. Right: Vascular flow (A) and micro structure (B) in the liver.

At fine scale, the basic structure of *liver lobuli* provides the functional hepatic cells. The lobuli encompass the venules of the central hepatic vein. In the corner of three adjacent lobuli, *portal triads* are found. A portal triad, consisting of arteriole and venule branches of hepatic artery and portal vein accompanied by a bile duct. Blood is provided within these triads and passes into the lobuli and into the branches of the hepatic vein. Within the lobules hepatic cells interact with the passing blood and provide in every portal triad.

Pathology Hepatocellular carcinoma (hcc) is an aggressive tumor developed by liver tissue. Being the sixth most prevalent cancer type, and third most frequent cause of cancer related death [FLB12], several methods of treatment have been investigated. In early stages, liver resection, transplantation and radio frequency ablation are favored choices of treatment. For instance, a resection can lead to a five year Survival Rate of 50% [Lim+12]. If surgery is contraindicated, local ablation techniques can be applied such as percutaneous ethanol injection, radio frequency or microwave ablation [VMR13].

Among these methods [PMI09] is high intensity focused ultrasound ablation [Li+07; Zha+09a], which still is an emerging therapy. Among its challenges is the both the suitable application of thermal energy as well as the controlled measurement thereof. FUS treatments for liver are subject to current research [Sch+17] (also see. *Image Guided Motion Compensation for FUS* (1.1.4.3)).

Landmarks Healthy liver tissues itself is evenly distributed and does not exhibit distinct features on a scale detectable by MRI and ultrasound imaging. Structural changes of the liver can however be detected. These are given by the outer boundary surface and the inner vascular structure. The shape of the liver is a globally observable landmark, provided the field of view is sufficiently large. Larger liver arteries, veins and bile ducts represent internal landmarks, which can be detected using MRI and ultrasound. These landmarks are crucial to the detection of motion.

Blood vessels, particularly those of the hepatic artery proper, change with the cardiac cycle. Pulsation affects both the diameter and the position of the vessel. This affects the appearance of vessel cross sections directly in the former case and indirectly — via through plane motion — in the latter. The effect is mostly visible on the major branches of the vascular trees and much less pronounced on smaller vessels. In different modalities the vessels of liver provide distinct contrasts.

1.1.1.2. Prostate

Anatomy The prostate is an exocrine gland of the male reproductive system, enclosed by bladder, rectum and pelvic floor. It has an approximate size of a chest nut and encloses the urethra. The prostate gland produces a secretion, which during male seminal emission is joined with the spermatozoon and ejected via the urethra. Due to sufficient distance to lungs and heart, the prostate is unaffected by the related short term motion. However, during an ongoing treatment the gland may change shape due to overall patient motion and in particular due to filling of the bladder [Moi+06] and peristaltic motion.

Pathology Among prostate pathologies, cancer is the most dangerous to develop. It has a high prevalence among Western countries and is reported to be the 'most frequently diagnosed non-skin cancer' and 'third leading cause of cancer death' [Haa+08].

Conventional treatments include radical prostatectomy as well as radiation, hormone and chemotherapy. *Salvage therapies* are applied, once conventional treatments did not produce positive responses. Studies using FUS as salvage therapy in the treatment of prostate cancer have been reported on [Cha+09; Cro+17]. Positive benefits of such treatments are reported [Bla+04; Uch+11; Asi+12]. For localized prostate carcinoma, a 5 year biochemical Survival Rate of 45% to 84% are reported [Zin+12]. Although benefits of MRI guidance (MRgFUS) are reported, experience in 2012 was limited. In general, the treatment of prostate cancer can lead to genitourinary side effects. These are reported rare in FUS treatments in [Ahm+12]. Within the application of FUS, this also favors the focal application to a whole-gland ablation [Sun+17]. Study [CT17] argues for applying FUS as a primary treatment for cases of mono focal and well-localized tumors.

1.1.2. Characteristics of motion

Patient motion during diagnosis or treatment can appear on different magnitudes and time scales and may be periodic or singular. The most prominent motion is related to breathing and heart beat which both are to some extent periodic. Non-periodic motion

may be caused by digestion processes which may happen on different time scales, such as gut motility or bladder filling. The physiological state of the patient may also play a role as when spontaneous movements are caused by illness or drifting motion as a result of relaxation or fatigue.

Constant contributions to motion which happens on scales which affect imaging and treatment commonly arise from the heart lung circulation. Both, heart and lungs are a major source of motion of internal organs.

1.1.2.1. Respiratory Motion

Breathing is a vital body function which promotes the exchange of gas between body and atmosphere. The lungs take up the gas inhaled into the lung and their thin membrane allows the diffusion of oxygen and carbon dioxide into the lung vessel and back respectively. They are located in the chest, underneath the ribs and above the abdomen and enclose the heart (Fig. 1.1). The diaphragm is the lower boundary which separates lungs and intestines. It is a thin muscle, which besides its separating effect is a main driver for breathing function. For inhalation the diaphragm contracts, compressing the abdomen and raising the ribs upwards and outwards. Intercostal muscles provide further assistance in expanding the rib cage. The resulting expansion of the lungs drags fresh air into them. Exhalation during shallow breathing requires little or no active assistance by muscles as the lungs are dragged back by elastic forces built up in the inhalation phase. Active exhalation may however be supported by the compression of abdominal muscles. Under relaxation, *shallow breathing* is predominant, whereas under exercise *deep breathing* occurs, in order to provide the body with additionally required oxygen.

Studies During the respiratory cycle, not only the lungs, but also the organs in their vicinity are affected by breathing motion. The predominant direction of diaphragm and liver motion is superior-inferior (SI). Davies et al. for instance report it in order of 12 mm±7 mm and 10 mm±8 mm [Dav+94] for the two organs. Keall et al. summarize several studies of breathing motion [Kea+06]. Maximum motion in the SI direction for pancreas, liver, kidney and diaphragm in deep breathing in the reported studies are 80 mm, 80 mm, 70 mm and 95 mm respectively, which is more than twice the amplitudes reported for shallow breathing.

Models Breathing motion has been characterized ([Wad54; Hoi+04; Mur04; Gir+07; Gie+05; Zha+09b; Jad+14] and models for describing the observable motion patterns have been presented [Low+05; Zha+09b]. These models describe the motion of interior organs by few parameters only — the breathing phase in the most simple case.

Reducing the motion into its main direction along the head-foot axis, the main component may be described by a regular breathing motion represented by even power cosine functions [Luj+99] (Fig. 1.3). In [Pla+05] the authors use semi-active MR markers on the outside of the chest and try to correlate the measured motion to the observed motion in internal organs. They find, that the correlation is highly dependent on the breathing maneuver — being stronger in cases of forced breathing than in quiet breathing.

The review article [Gir+13] describes main applications of respiratory gating in radiation therapy, separating the types of gating into four groups — the direct inclusion of additional movement margins into the planning scan, forced shallow breathing, breath hold, and respiration gating based.

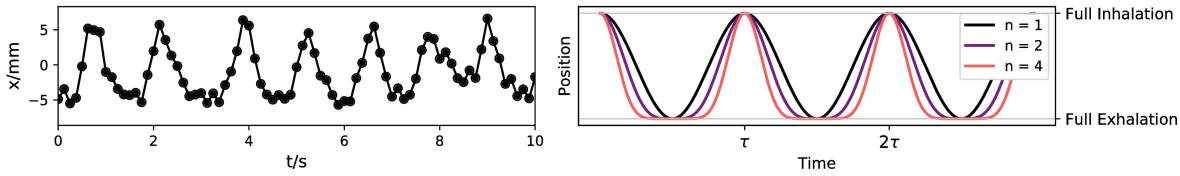


Figure 1.3.: Left: Liver displacement caused by breathing. Right: Cosine model for breathing motion along the major breathing direction (head-foot) after [Luj+99]. The power $(2n)$ of the cosine function determines the pronunciation of the exhale phase.

Liver The liver is strongly affected by breathing motion. Tracking algorithms require fast processing times to keep system latency down. Medical images of liver in particular are usually sparse, mostly exhibiting a contrast between liver vessels and surrounding liver tissues. Correlations have been reported [SSA04; Hoi+04; AY04; Gie+05] and can be made use of in a tracking application. Imaging capabilities do not reach down to the smallest scales of vascular structure, but can separate larger vessels from the surrounding tissue. Local correlation and sparsity offer ways to reduce computational cost in order to reach real-time performance.

Various tracking framework can be found in the literature. Many of recent approaches have been evaluated in two ultrasound liver motion tracking challenges in the years 2014 and 2015 [Luc+15; De +15]. These algorithms differ in the way they define and describe the target and in the way they estimate and regularize the observed motion. Some algorithms also introduce predictive components for future frames or error management and fall back solutions in case of uncertainty. For the sake of brevity, these will not be outlined here.

1.1.2.2. Cardiac Motion

A further vital source of regular motion within the body is the heart. The muscular hollow organ provides the blood flow in the circulatory system. The pumping function of the heart is supported by a contracting and expanding motion of the muscle. Motion of the heart is repeated at 72 beat per minute on average in rest [CWR10]. However, the heart rate changes depending on short term oxygen requirements or long term training [Hau+06] or disease [PJB09].

The heart is located between the lungs. Its motion therefore mostly affects its direct surroundings, which includes the upper abdomen. However, heart related motion may manifest in other regions of the body, as the arterial system transports on the pressure

differences. For instance, pulsation can be observed in liver blood vessels and even the head may undergo a slight motion, which can be attributed to the cardiac cycle [Zho+09].

If the cardiac phase is of interest, an electro cardiograph (ECG) may be used [DK04]. Similar phase data is obtained from an MRI based navigator [Mog+12]. Cardiac motion itself may be tracked using diagnostic ultrasound or fast MRI data. Tracking methods include model based approaches [Nil+07; Sol+07; Chu+08; Ma+10], snakes or active shape models [ACG96; HOR07] or Optical Flow [Suh+05; Lin+08; TBS12], which will be covered in section 1.2.

In this work cardiac motion only occurs as an indirect phenomenon, for instance as pulsate variations in other organs. The heart itself is not subject of investigation.

1.1.3. Non-Invasive, Focal Treatment Techniques

A wide spectrum of medical treatment techniques exist. Very roughly, their modes of actions may be based on chemical, biological and physical effects. Examples are for instance found in the administration of antibiotics, blood transfusion and radiation therapy, respectively. This section presents two non-invasive physical treatments, which require the localization of the treatment target from the outside of the body. Radiation and focused ultrasound therapy will be presented with respect to their mode of application and means of steering it.

1.1.3.1. Radiation Therapy

Radiation therapy (RT) is a medical treatment technique which uses the effect ionizing radiation has on human body cells. Goal of the treatment is the destruction of cells which have shown a harmful impact on the patient: The cell's deoxyribonucleic acid (DNA) molecule, carrying genetic instructions for metabolism, growth and proliferation is irreproducibility damaged when overexposed to ionizing radiation. Key mechanism in RT is cell death induced by the destruction of DNA. The technique is applied to malign tissue, which proliferates uncontrollably, such as cancer.

Due to its non-selective nature, radiation always damages healthy tissue as a side effect of the treatment. Therefore, the goal of a therapy is to apply most radiation to the malign tissue and keeping it low in the healthy. Some techniques apply the source of radiation close to the target (Brachytherapy), whereas the majority separate source and patient. The latter group is referred to as *external beam radiation therapy* (EBRT). In EBRT a beam of radiation is applied from the outside of the patient. It damages not only the tumor, but also healthy tissue in front of and behind it. In order to keep the damage to healthy tissue low, beams are applied from different angles, spreading out the acquired total *dose* in healthy regions, while maximizing it in the treatment area. Planning of the exact schedule of application is a substatial part of radiation therapy.

In general, different types of radiation can be used for EBRT. These differ in the type of radiation applied and therefore substantially in the depth-dose curves (Fig. 1.4). Common techniques apply photon beams. *X-ray and γ-ray* radiation are generated by

accelerating electrons from a heated cathode to an anode, typically consisting of a tungsten alloy. Hitting the anode, electrons transfer a part of their energy into X-ray or γ bremsstrahlung, which is used in the therapy. Penetration depth of the produced radiation depends on the photon energy and varies in different applications. Orthovoltage systems produce energies up to 600 keV and admit a usable penetration depth of 4 cm – 6 cm. Higher energies are produced using *linear particle accelerators* (LINACs) in the range of 4 MeV to 25 MeV. Some systems also use cobalt isotope ^{60}Co to produce photons with energies of 1.17 MeV and 1.33 MeV [CR15]. A typical treatment device is depicted in Figure 1.4. Less frequent sources of radiation are *charged particles*. Electron beams — obtained as a byproduct in the generation of X-ray and γ radiation — have a strong falloff after a low penetration depth. They are therefore used to treat superficial targets. Heavier particles, such as protons [Lev+05] or carbon ions [AK05] require dedicated acceleration units. Compared to photons they however exhibit a more interesting absorption profile. Photon absorption is largely exponential with increasing penetration depth. Particles in tissue however decelerate until the reach a critical minimum energy, right before coming to rest. At this *Bragg peak* [ML12], most of the particles energy is released into tissue. The absorption profile drops sharply after the peak, indicating that no energy is deposited beyond it. The Bragg peak leaves an enormous advantage to particle therapy, as it offers a maximum dose peak already in a one-dimensional treatment (i.e. without beam application from multiple directions). However, it also requires knowledge about the absorption properties of the tissue in path to the target and most of all, particle accelerators for the generation of particle rays.

Steering The treating ray of ionizing radiation can be altered in its transversal direction and shape in different techniques. Aside from moving the beam source itself, the transversal positioning can be influenced. A Multi-Leaf-Collimator (MLC) is used to reduce the treatment field of an unfocused beam of radiation [JR04]. By moving blocks of radiation absorbing material such as lead or tungsten into and out of the beam, the cross section profile of it can be changed. Primarily, this allows for sparing healthy tissue outside the transversal profile boundaries of the treatment region. On the other hand, keeping the dose within the healthy tissue, this benefit can be used to increase the applied dose in the treatment region.

Ionized rays on the other hand can be steered directly by deflecting the particles with magnetic quadrupole fields [Wee+96]. Other than wide beams of γ or X-ray radiation, particle beams may need to be widened to spread their transversal Gaussian intensity profile over a broader transversal area.

1.1.3.2. High Intensity Focused Ultrasound

Similar to radiotherapy, high intensity focused ultrasound (FUS, HIFU) is applied with the goal to destroy malign tissue [Mal+13]. Other than RT, FUS operates on mechanical waves and is therefore free of ionizing radiation. For the application of FUS, a transducer is placed on the patient's body. It is acoustically coupled via ultrasound gel. The transducer emits continuous ultrasound waves, which are focused to the target position inside

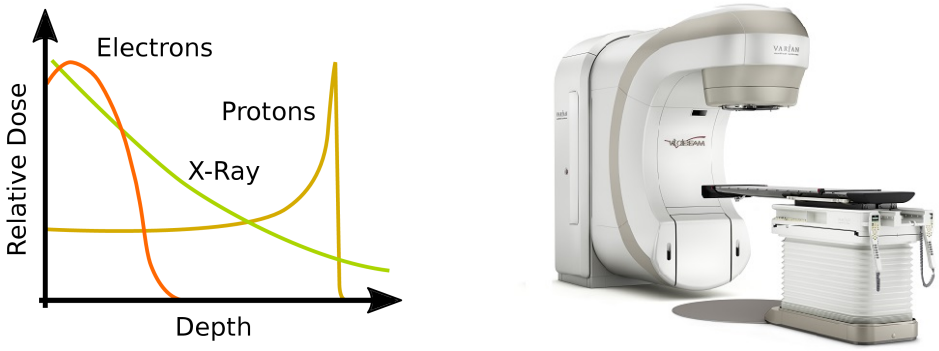


Figure 1.4.: Left: Typical depth dose curve in radiotherapy. For a detailed image see [AK05]. Right: A radio therapeutic system for the application of high-energy X-rays or electrons to the target region. Image courtesy of Varian Medical Systems (Palo Alto, United States of America, www.varian.com)

the body (Fig. 1.5). In the focus position, the mechanical waves interfere constructively. Depending on the applied energy, focused ultrasound causes different effects. A common target effect is an increase in temperature. The applied mechanical energy is absorbed by the tissue, such that its temperature rises, up to 90° Celsius. If tissue temperature increases beyond the level of denaturation of proteins and nucleic acids, *thermal ablation* is the consequence. This is the effect commonly sought in FUS therapy. Higher pressure can lead to cavitation, a process in which the tissue ruptures spontaneously. Since the effect is difficult to control, it is currently only used in *lithotripsy* [Yos+09; Ike+16], i.e. for kidney stone destruction. Using FUS, tissue is not ablated instantaneously. Similar to radiotherapy a thermal dose can be motivated, which reflects the amount of damage the tissue has received. Commonly, tissue in the focal spot needs to be treated several seconds in order to reach ablation. The focal spot is commonly shaped as an ellipsoid (e.g. $2\,\text{mm} \times 2\,\text{mm} \times 10\,\text{mm}$ [Ht95]) and usually smaller than the full target volume. The target volume is therefore separated into multiple *sonications* which are executed sequentially. Since treated tissue tends to block ultrasound, sonications farthest away from the transducer are executed first. Thus, similar to RT, FUS requires precise planning of the treatment, in which the target volume is covered as good as possible, while risk structures are excluded.

Limitations in the applicability arise from properties of ultrasound itself. Artifacts caused by acoustic shadowing, reverberation and refraction may arise. Ultrasound is reflected from gas within the body, which exists in the lungs, but also in the bowel. Even stronger reflection and absorption is caused by bone structure. Reflection can be sufficiently strong to create unpredictable secondary focal spots which damage tissue [Dub+08]. Furthermore, the tissue type influences properties of sound velocity and energy absorption, such that planning in homogeneous medium may lead to incorrect results in reality.

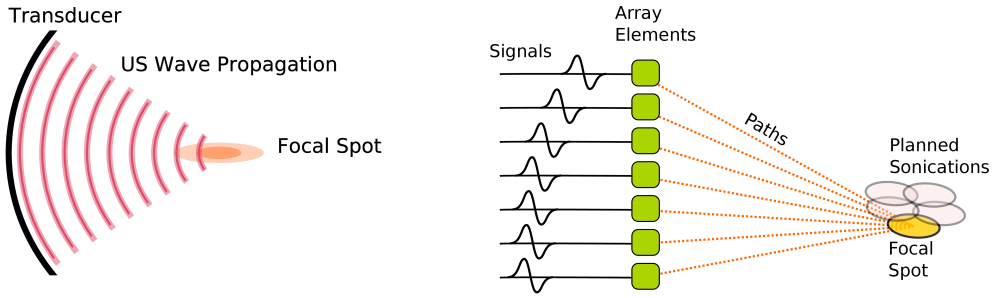


Figure 1.5.: Wave propagation and focal spot in FUS application. Left: A fixed focused transducer creates the focal spot in its *natural focus*. Right: A phased array system uses temporal delays in the excitation of individual element to steer the focal spot. In order to emphasis the phase offset, only a single pulse is depicted, whereas FUS typically applies continuous waves.

Steering There exist different types if FUS transducers. Single element transducers have a shape dependent fixed focus. Their focus can only be altered by moving the transducer. By contrast, phased array systems can control the focus without mechanically moving the transducer. They consist of hundreds of elements, which can be steered electronically and independently. Adapting the phase of the single elements, the position of maximal positive interference can be altered. This can be achieved within certain ranges. Steering to far away from the transducer's *natural focus* may cause a loss in efficiency and even generate undesirable secondary focal spots.

FUS Treatment for Prostate Salvage Therapy In prostate treatment the goal of a FUS procedure is the ablation of malign tissue through the application of focused ultrasound. An extensive review is found in [CT17].

For guiding the application MR imaging may be considered. During the diagnosis and planning, functional MR imaging provides valuable information about the location of malign tissue inside the prostate gland. Anatomical scans deliver information about common structures, such as capsule boundaries, bladder, transducer, which may be important for the localization. There are certain high risk structures such as the rectal wall and nerve bundles which can be integrated into the plan.

The treatment requires precise targeting in order to reach sufficient temperature, but also to spare important healthy structures. During the treatment, the anatomy may change due to slow drifting motions. These may arise from involuntary patient motion as well as bladder filling close to the prostate. The slight shifts and deformations have to be taken into account during the treatment.

Echo planar imaging (EPI, see sec. 1.1.5) allows for the quick acquisition of three dimensional images of the region of interest. The EPI's phase images may be used to derive information about temperature changes between a reference and the current scan, while the magnitude images may be used to derive motion. Within regular intervals, the 3D scans are acquired to observe both the general motion and the temperature changes

in the ROI. During heating, however, it may be desirable to use 2D imaging only in order to increase temporal resolution of the temperature information.

The detection of motion and deformation based on regular 3D EPI scans serves as a confirming check to assure that no drastic changes have appeared to the tissue and update the targeting if required.

1.1.4. Motion Management for Non-Invasive Treatment

Both RT and FUS rely strongly on knowledge of the treatment location during treatment itself. A correspondence between the planning and the current treatment situation must be established and maintained for a successful application. Particularly in radiotherapy *fractionated* applications are common, in which the a patient receives multiple *fractions* of therapeutic application for a series of days. For each fraction, the correct positioning must be assured. Several effects interfere with this goal. First, between planning scan and treatment the patient is likely to have changed positioning as both may take place in different environments. Furthermore, the patient may change its appearance due to a delay between both. In the time of hours, digestion may lead to changes in the arrangement of organs. In the time of days or weeks, the patient may have changed figure. Weight loss may for instance occur in-between sessions of radio therapeutic treatments. Due to these (long term) effect, it is necessary to establish a correspondence between the planning situation and the one right before treatment. During a treatment, the correspondence may suffer from involuntary patient motion. When extremities are treated, keeping still for several minutes may pose a problem for patients. Once the inner organs are affected, breathing motion, cardiac pulsation and digestive processes add to the list of movements, which have the potential to reduce the required correspondence quickly. It therefore crucial to cope with the possibility of pose changed during treatment. Techniques for handling correspondence uncertainty can roughly be classified into nonexclusive approaches *controlling* and *adapting to* motion (Fig. 1.6).

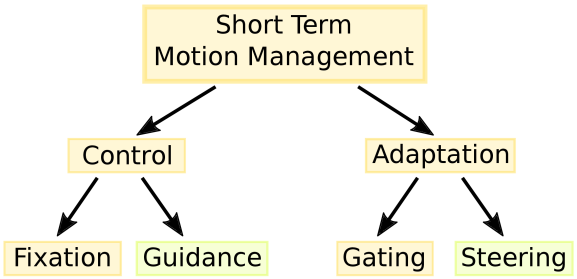


Figure 1.6.: Motion can be managed on side of the patient (control) and therapy system (adaptation).

1.1.4.1. Controlling Motion

In RT applications, the use of patient specific casts or plastic molds is found [KMP06]. These allow patients to return into their specific reference positioning in successive sessions. Further methods exist to reduced residual motion, such as external plastic foam supports or bite supports. Other body immobilization systems allow the fixation by stiffening the vacuum cushion, once the patient is positioned.

These techniques may also be applied to reduced motion in the abdominal region, but breathing for instance cannot be circumvented over for an arbitrary period. However, the patient can be *guided* into a controlled breath hold and this situation used as reference — treatment is only applied when the patient is in the reference position.

1.1.4.2. Adapting to Motion

Motion cannot always be controlled perfectly. Incorporating knowledge of real-time position information into the treatment allows to adapt to unavoidable, but also unexpected movements. Furthermore, it relaxes the constraints on patient based motion control. This offers the opportunity to a more comfortable treatment.

Depending on capabilities of the treating system, the application may be *gated* or even *steered*. The former term refers to an application of the treatment only when the patient is in a known reference position. Such application can be combined with breath hold, but it is also possible to use other events as gating trigger. For instance, shock waves for lithotripsy can be applied only in instances where the operator detects the target stone in the natural focal sport of the transducer.

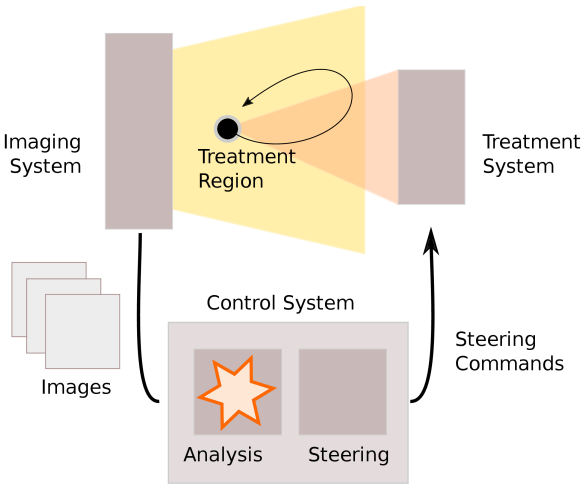


Figure 1.7.: Possible application for an algorithm automatically deriving motion from incoming medical image data: Steering an automatic treatment system.

If treatment locations can be controlled in real-time, steering may be considered (Fig. 1.7). In radiotherapy, steering can be accomplished in changing the configuration of a multi-leaf collimator to toggle photon exposures or by deflecting a beam of

charged particles. FUS applications control the focal spot via phased array transducers. While there exist robotic positioning systems for the transducer, these are commonly only used set the initial position, not to move the transducer along. In radiation therapy robotic systems exist for positioning linear accelerators [AMR08; Avk+16].

Along with the increasing degree of automation in medicine, more systems become capable of steering applicators in real-time. In order to provide these systems with the necessary location updates, tracking algorithms are required.

Handling Respiratory Motion Most approaches for handling breathing motion stem from RT applications. *Motion encompassing* methods integrate the uncertainty generated into the planning. Information about this uncertainty can be gained from CT. Acquiring a single slow CT [Lag+01], CTs of maximum inhalation and exhalation [AY04] or even a 4D CT [Low+03; For+03; Rie+05], the target region blurred by motion can be estimated. This estimate is then used for planning and treatment. *Gating* methods [Bal+98; Koc+04; Gir+13] can be linked to external and internal surrogates. External surrogates [AY04; Koc+04; Rua+08] commonly rely on optical tracking of markers placed onto the abdominal surface. The breathing phase can also be estimated by breathing belts, measuring the extent of chest or abdomen during breathing. Similarly, spirometers estimate the lung volume from airflow measurements. Internal surrogates [Che+07; SSA04; Zha+06; Pla+05; Kei05] can be found in gold fiducials, which produce dark spots in fluoroscopy images. In MRI, the 1D acquisition along the SI axis (*pencil beam navigator*, see 1.1.5) can be used to detect the diaphragm. Gating always relies on a correlation between gate and target. *Breath control* is largely related to gating. The patient holds breath in deep inspiration [Ros+00] or self-defined position [Kim+01]. This may be voluntary or initiated by active-breathing-control [Won+99; Gag+07]. It can be accompanied by respiratory monitoring. *Shallow breathing* may also be forced by fixating the patient with an abdominal compression [Lax+94]. This reduces the variability and magnitude of possible breathing patterns. *Real-time tracking* methods refer to methods which use an internal marker to repeatedly locate or infer the current target location. If the tumor is visible, it may be imaged itself. Otherwise an artificial fiducial in its vicinity or the host organ itself may serve as surrogates. Registration algorithms for the computation of dense deformation fields [SSA05; Zac+15] have been described, but unless highly parallelized, they come with a high computational cost which may render them inapplicable for real-time adaptation. In ultrasound liver tracking various tracking algorithms have been proposed [Mur04; Lin+12] and a wide range of algorithms has emerged alongside tracking challenges [Luc+15; De +15]. An MR tracking algorithm based on precomputed templates is proposed in [Bri+14]. In general, the latency is required to be sufficiently low (certainly $< 0.5\,\mathrm{s}$ [Kea+06]). Nevertheless, it may be necessary to include prediction algorithms [Ved+03; Hoi+06] in order to compensate for it.

1.1.4.3. Image Guided Motion Compensation for FUS

Motion compensation for focused ultrasound applications has been proposed and first studies have been executed. In [Aub+12], a system using simultaneous ultrasound guidance and MRI thermometry is explored in an *ex vivo* study. A balloon based driver was used to move a sealed piece of degassed *ex vivo* liver. This way respiratory motion with a cycle of 10s-12s was modeled. Motion was detected using Optical Flow and used for electronic steering of the FUS transducer. This way the authors achieved a duty cycle of 100%, i.e. continuous sonication. Results in a similar setting are obtained in [Str+15], using a breathing cycle of 3 s–7 s, which is closer to human motion. The study used the Particle Filter based, similarity transform tracker that will be presented in section 3.2.3. Respiratory gating is used to compensate breathing motion in a study on *in vivo* porcine liver [Zac+14]. The authors report a residual shift of 4 mm in liver position within the first ten minutes of the study. They ascribe it to muscle relaxation as a result of anesthesia. An MR based compensation system is proposed in [Cel+14] and examined on ex-vivo animal tissue in a motion phantom. A navigator MRI sequence is used to obtain displacement data of the phantom. The authors report accurate motion compensation for both targeting and thermometry. Speckle tracking in ultrasound data is used to obtain 1D motion information for *ex vivo* liver in a motion phantom in [Cha+15a]. In the study, a fixed focus transducer is moved by a robotic arm for motion compensation, which allows to apply energy during 80% of the cycle. A more general study of the impact of motion on the duty cycle can be found in [Loe+16]. The study also comes to the conclusion that, under the considered conditions, a history based motion prediction is more efficient than a linear extrapolation. The study [Cel+18] combines MRI and ultrasound motion compensation. Ultrasound is used to detect in-plane motion in the coronal plane using Optical Flow. An MRI navigator is used to detect through plane motion. *Ex vivo* experiments show successful ablation, whereas in volunteers successful acquisition of thermometry data is demonstrated. Motion compensation and treatment only pose an intermediary step of a FUS treatment. There are several options to organize a work flow. For a generalizing overview the reader is referred to [Loe+16].

1.1.5. Characteristics of Medical Imaging Techniques

The visible spectrum of electromagnetic radiation is not well suited to monitor processes inside the human body. With a penetration depth of few millimeters, it is suitable for the observation of surface properties, but not for structures beyond. For the investigation further inside the body, different *imaging modalities* have been established.

X-ray, computed tomography (CT) and positron emission tomography (PET) are based on the application and detection of ionizing radiation. Due to its high energy, this radiation is capable of passing through body tissues with lower attenuation than light, and can be detected by outside electronic detector units. These retrace the pathway of observed radiation in order to *reconstruct* an image by computation. As the application of ionizing radiation conveys certain health hazards, its application needs to be evaluated in each case. There exist techniques free of ionizing radiation, such as magnetic

resonance imaging and ultrasound. Since, in this work, MRI and ultrasound data will be used as input, their function and main properties will briefly be presented here.

Generally a common naming convention exists, which specifies the imaged region or orientation (Fig. 1.8). Planes and directions are oriented on the patient. The three main orthogonal axes of the patient are the head-foot (*superior/inferior* (SI)), front-back (*anterior/posterior* (AP)), and right-left (*dexter/sinister* (DS)) direction. Other common names for the directions are cranial/caudal for SI, dorsal/ventral for AP and left/right for DS. The main imaging planes are orthogonal to these directions. Interpreting the SI direction as longitudinal axis of the human body, the *transversal* plane is extended in AP and DS direction. The *coronal* plane combines SI and DS, whereas the *sagittal* plane joins AP and DS. The motion of structure within an imaging plane is referred to as *in-plane motion* as opposed to *through-plane motion*.

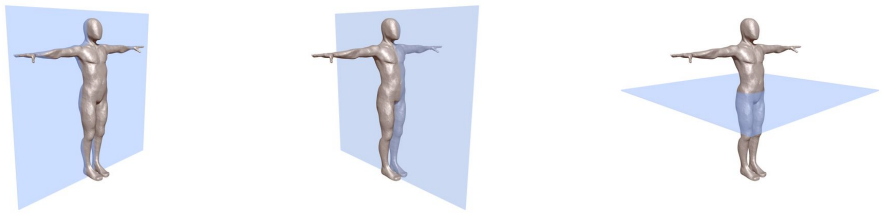


Figure 1.8.: Left to right: Coronal, transversal and sagittal imaging plane.

1.1.5.1. Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a technique which utilizes the presence of magnetic moments in atomic nuclei. The most prominent nucleus in this field is hydrogen. The magnetic moment of hydrogen is not exceptionally strong but present in abundance in the human body as part of water molecules.

In order to exploit magnetic moments for imaging, dedicated hardware is required. In the absence of an outer magnetic field, moments are randomly oriented. In a sufficiently strong outer magnetic field — usually the order of 1.5 T or 3 T — they can be partially aligned to form a net magnetization vector parallel to the field. Furthermore, the net magnetic moment $\vec{\mu}$ precesses about the axis of the external magnetic field $\vec{B}$ with the *Larmor Frequency* $\omega = \gamma B$, where γ denotes the gyro-magnetic ratio of the atom (hydrogen: $\gamma = 42.6\,\mathrm{MHz/T}$).

When an alternating electromagnetic field with frequency ω is applied in the plane perpendicular to $\vec{B}$, resonant effects tip the net magnetization. Depending on the total magnitude at Lamor frequency of such a pulse, the net magnetization can be brought from initial longitudinal to transversal (or a mixed) direction. As soon as a transversal component exists, it emits an electromagnetic signal, which can be detected by receiver coils. Variations of excitation pulse frequency and local magnetic field are used to encode spatial information. Local magnetic fields have to be changed several times by magnetic

gradient systems in order to obtain a image. The temporal order of these is referred to as MRI sequence. Contrast of an MRI experiments depends on tissue and sequence parameters. The choice of a sequence for a specific body region and application may be very specific.

Echo Planar Imaging [STM91] (EPI) represents a type of sequence, which allows for fast imaging, as it requires few excitations in order to capture an image. EPI is suitable for real-time imaging in 2D dimensions, with acquisition times in the order of 100 ms when acquiring 2D data. EPI images offers both a magnitude and phase information. While the former may be used for detecting motion, the latter may server as mean for inferring termperature changes [Sta+04; Wei+03; Rie+04].

Navigator In a navigator acquisition, a one dimensional image is acquired. Due to the shape of that line the acquisition is also referred to as *pencil beam* acquisition. Within the navigator 'image' intensity profiles indicate the motion along the beam direction. The beam is often placed orthogonal to the diaphragm in order to observe breathing motion. The juxtaposition of sequentially acquired navigators shows motion of intensity profiles in the direction of the beam and is called motion mode display. The acquisition of a navigator is faster than of complete images and can be integrated into the other sequences. However, since the navigator uses two crossing slice to select the beam region, these slices may appear as visible artifacts in the host sequence.

Target Organs For a focused ultrasound application, one is interested, not only in the current location of the target organ, but also in the temperature, which is currently being applied. Few MRI sequences suffice the requirements of a high image frame rate and temperature information. Echo planar imaging (EPI) delivers 2D slices with a sufficiently high update rate for real-time tracking (Fig. 1.9, left). Furthermore, besides magnitude images the sequence type delivers phase information which can be used for thermometry.

Liver: As an effect of the MR imaging process, a steady state magnetization builds up inside the imaging slice, which after an initial unsteady phase delivers a stable contrast of the liver tissue. However, blood which is not inside the imaging slice, does not experience the buildup of a steady state magnetization. Therefore, blood which streams into the imaging slice during the treatment is unsaturated. This causes the vessels to yield high signal intensity, which stand out before the liver vessels. Liver vessels serve as natural landmarks for EPI liver tracking. Inside the liver, no other landmarks are found in this modality. Due to the dependency on the inflow of unsaturated blood, the appearance of larger vessels depends on the cardiac cycle (Fig. 1.10).

Prostate: The prostate does not show small scale landmarks in EPI imaging (Fig. 1.9, right). It appears brighter than its immediate surrounding. If during an intervention a catheter resides inside the urethra, the urethra is visible as a dark region within the prostate. The three dimensional images of EPI prostate data shown in this work are acquired in intervals in the order of minutes and therefore are not affected by steady state effects.

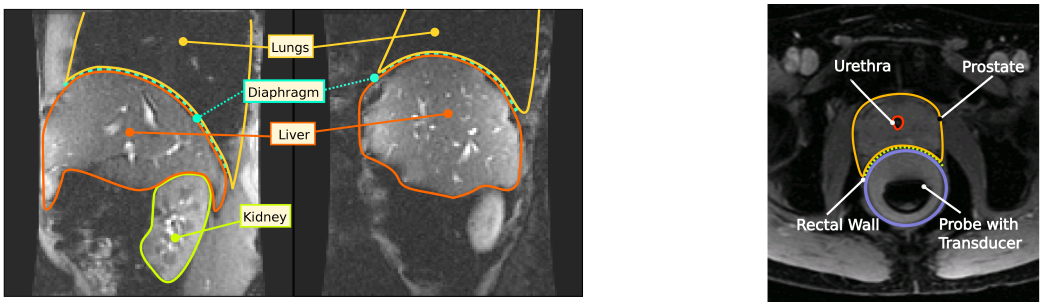


Figure 1.9.: Echo Planar Imaging; Left: Anatomy in two sagittal slices of liver in a volunteer. Right: Transversal image of the prostate during FUS intervention.

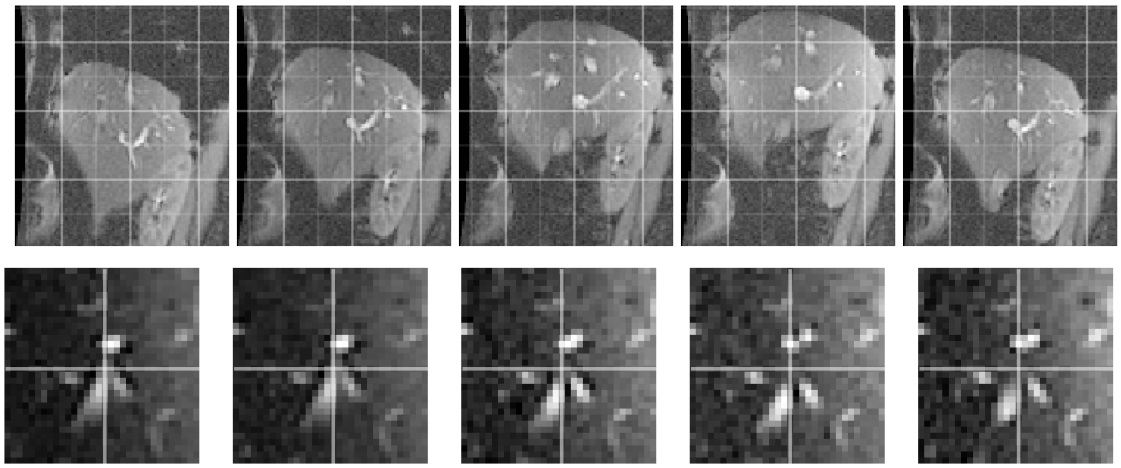


Figure 1.10.: Echo Planar Imaging, Top: Motion of liver and kidney (center) during the respiratory cycle, deep breathing. Bottom: Cardiac cycle in a liver vessel bifurcation during a held inhale phase.

1.1.5.2. Ultrasound Imaging

Ultrasound describes acoustic waves of frequencies higher than the human ear is able to perceive ($\approx$ 20kHz). For medical imaging applications, frequencies in the range of MHz are used — 1 MHz to 6 MHz for deep and 6 MHz to 15 MHz to superficial structures.

Observing the propagation and scattering behavior of waves in material, it is possible to obtain information about its internal structure. For mechanical properties such as stiffness and elasticity, sound waves can be applied. Doing so, attenuation inside the material and spatial resolution of the resulting data depend inversely on the frequency. For analysis of tissue in medical imaging, a viable frequency range is given by ultrasound.

In the propagation of sound waves, effects such as reflection, diffraction, dispersion and absorption can be observed. Geometric relations of transmission and reflection along material interfaces exist and the image reconstruction process relies on geometrical

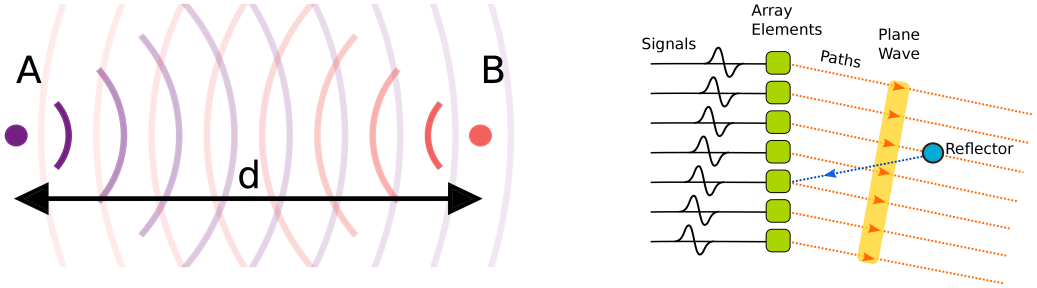


Figure 1.11.: Left: Pulses of ultrasound waves are sent from a source A into the material and reflected back to the source by a scatterer B. In homogeneous media the distance between A and B is linearly proportional to the speed of the sound wave in the medium and the time traveled. Right: In plane wave imaging, the transducer elements are excited by a phase delay, resulting in an angulated plane wave.

considerations. Wave properties such as constructive and destructive interference are also observed and lead to speckle.

Technical Realization For acquisition of ultrasound data, ultrasound is applied to the tissue externally. The ultrasound transducer consists of a piezoelectric crystal elements, which contract rapidly under applied voltage. Also, they emit a voltage when deformed by external forces. A piezo element can therefore be used to generate a mechanical wave as well as detect it. For imaging, elements emit a short pulse of an ultrasound wave with a few lobes of the imaging frequency. This wave passes through tissue and will be reflected on interface changes. The reflected pulse is detected by the same piezo elements used for transmission. Measuring the time delay Δt, assuming a constant speed of sound v_0 inside the tissue, distance d between source A and reflector B can be computed (Fig. 1.11, left) as $d = v_0 \Delta t/2$. The typical speed of sound is often assumed to be $v_0 = 1540\,\mathrm{m/s}$, as observed in water. Computing an echo profile along one dimension from the transducer is referred to as *A-mode* ultrasound. As the magnitude of the reflected signal is of interests, the raw signal is often transformed to its envelop (Scan Conversion). Furthermore, signal may be artificially emphasized (Time Gain Compensation) to balance out depth related losses. The acquisition of a 2D image in *B-mode* ultrasound can be realized by using a linear array of piezo elements (Fig. 1.11, right). The signal sent out from a particular piezo element can be reflected to all others. Considering the possible geometric pathways of the ultrasound signal in between, a 2D image can be reconstructed. Similar to MRI navigator acquisition, in motion mode (*M-Mode*) one line of ultrasound is monitored and the sequentially acquired lines are concatenated to show the temporal evolution of image intensity along this line.

Imaging Properties The resolution of ultrasound images is limited by the imaging frequency. Generally, structures smaller than the wave length can not properly be identified. Ultrasound image *speckle* is sourced in small scale structural inhomogenei-

ties [WH81; Dam+14], yielding a grainy, yet temporarily stable intensity patter in images (Fig. 1.12). Even though a tissue is macroscopically homogeneous, it will show a textured pattern in the image. Furthermore, in ultrasound imaging the ideal wave generation

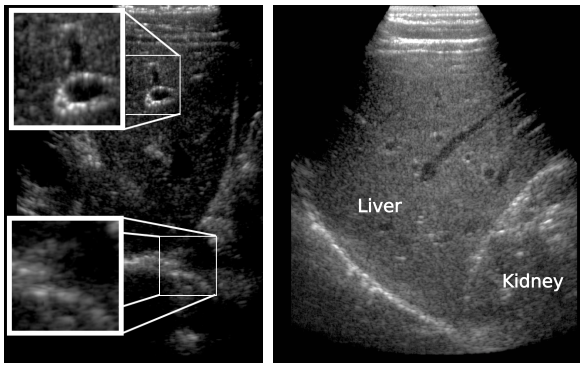


Figure 1.12.: Left: Ultrasound speckle in an abdominal image. Top inset: two vessels inside the liver. Bottom inset: Abdominal structure at the edge of / behind the liver. Right: Ultrasound image showing a section of liver and kidney.

and reception can only be approximated by the aperture of transducer, which results in image blur. Such can be modeled by approximating the imaging process as a convolution of the true object intensity I_{obj} and a point spread function I_{psf}

$$I(x, y) = I_{obj}(x, y) * I_{psf}(x, y). \tag{1.1}$$

The point spread function is commonly wide in the far field, since angular information is reduced. Deconvolution algorithms for the retrieval of the true object intensity are an ongoing topic of research [DPM15]. Once obtained from an image, a characteristic point spread function can be used to simulated ultrasound data by a certain transducer type [MG16].

Liver Similarly to MRI images, contrast providing structures in healthy liver tissue are given by the blood vessels. Ultrasound attenuation and reflection in blood are much lower than in tissue, causing the inside of blood vessels to appear dark. This also applies to bile ducts. Arteries and veins show different characteristics of their membrane. Since the former are to withstand higher blood pressures, their vessel wall is thicker. It is also more echogenic than liver tissue. In ultrasound images this shows as a bright surrounding of liver arteries. Vein vessel walls on the other hand do not show in ultrasound images. Landmarks of veins transition directly from liver tissue intensity to the dark of liquid. In liver both feeding vessels, the hepatic artery and portal vein provide artery-like bright contrast, whereas the hepatic veins show vein-like contrast. A hepatic veins vessel wall needs to withstand lower pressure and is therefore much less *echogenic*. Aside from the vascular, the liver tissue itself offers a particular speckle pattern. It is the result of the underlying structure of lobuli. This might be exploited for tissue classification [Cri15].

1.2. Video Tracking and Image Registration

Video tracking and image registration are wide fields in image analysis. Both aim at establishing a correspondence of objects observable in sets of images. This work is motivated from the point of a tracking algorithm. This section is intended to give an overview into the topic and highlight occasional relations to image registration.

Video tracking finds applications in vast areas of automated scene interpretation, such as automotive, surveillance and gaming industry. There exist several review papers and book on both tracking and image registration A comparative study of multiple single target trackers is found in [Sme+14]. Human motion capture is described in [MG01] and followup [MHK06] or with a particular accent on depth imagery in [CWF13].

The function of a tracking algorithm can be summarized in the following way (Fig. 1.13): Input to a tracking algorithm is provided by as a sequential stream of image data. From an initial input a description of the target is formed. For this, features of the initial data are identified and composed into a target representation. In later frames this representation is used to identify the target. After the initialization, the tracking phase begins. The algorithm uses each incoming frame to update the location of the target position. Besides the sheer localization, higher level management processes may occur. These may for instance include reasoning about occlusion and appearance of additional targets. At the end of each tracking step, the algorithm generates an output by extracting the relevant data from its internal state. This data generally consists of position estimates, but may include any other data which is relevant to proceeding algorithms. The proceeding stage then uses the data for higher level processes, such as the analysis for particular events in the time series and the control of further processes.

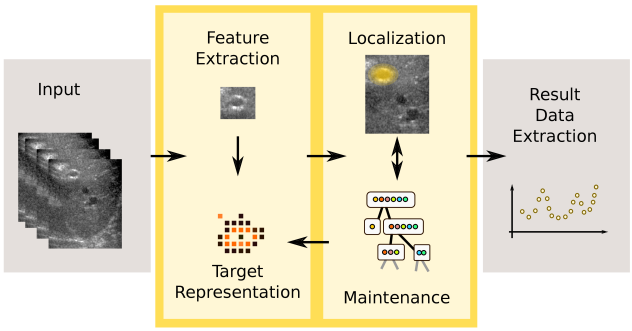


Figure 1.13.: Major elements and data flow of a tracking algorithm.

Internally, the target state is encoded in a low dimensional variable (Fig. 1.14, right). The algorithm is concerned with deriving the unknown state from the incoming image data. Appearance and transformation model are used to translate a state into an image representation, such that the state can be validated against the current data. Prediction and optimization are used to guide towards likely target states pro- and retrospectively.

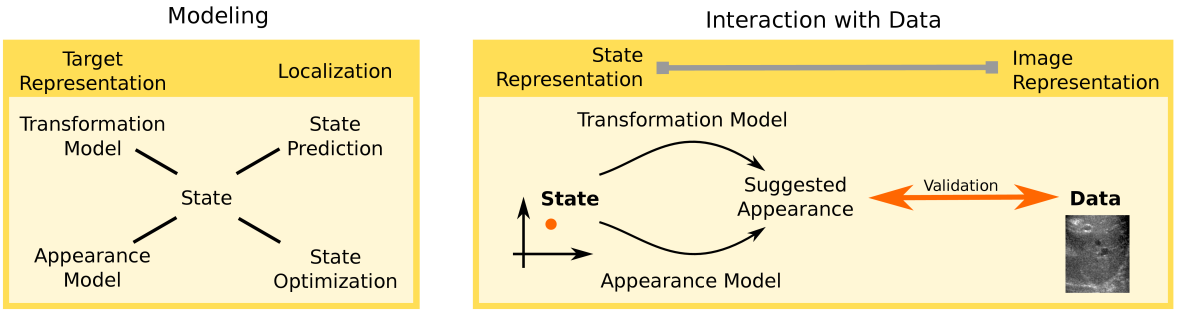


Figure 1.14.: Left: Modeling components for the tracking algorithm. Right: Translation from state space to image space. A low dimensional state is transformed into a high dimensional representation, which can be compared to actual image data.

The text book [MC11] presents five main components of a video tracker. These are methods to *extract* relevant information from a target area, a definition to *encode* the target state and *propagate* it over time. On higher levels, strategies for *managing* appearing and disappearing targets and finally, *interpreting* the observed tracks. In [EPV93; MV98; Vie+16] a common set of classifiers for medical image registration methods occurs. Among these are the *nature of the transformation*, *domain of the transformation*, the *modalities involved* and the *optimization procedure*. This work will use five criteria depicted in the left of Fig. 1.14 to classify approaches in literature:

- State Encoding
- Appearance Model
- Transformation Model
- State Prediction
- State Optimization

Generally, in medical image registration, varying terms exist to distinguish two images being matched. They differ by which of them is warped to match the other. The one not being warped is called *target* or *fixed* image, whereas the one being adapted to match is referred to as *source* or *moving* image [SDP13]. In this work the distinction will be based on the temporal ordering made use of in video tracking. The image of the reference situation will be referred to as *reference image* I_R and the image, to be matched to it currently, will be named *current image* I_C – the direction of the mapping is not specified by these terms. In fact, the mapping used here will be a vector field pointing from *reference* positions to *current* positions, which is the opposed notion to image registration terms.

1.2.1. State Encoding

Tracking a single target is centered around the inference of model parameters from a stream of images. Image data is high dimensional information that needs to be translated into a low dimensional target state in order to allow for an interpretation and also formulation of the target dynamics.

The target description is encoded in the state vector $\vec{x}$. The space such vector exists in is referred to as State Space in $\mathbb{R}^d$. Each dimension d of the state space describes one degree of freedom (DOF) of the model. In tracking applications, the number of degrees of freedom is typically low (< 10), whereas registration algorithms often approach high dimensional problems.

Often $\vec{x}$ contains information about the current location, shape or deformation of the target. It may also encode the appearance, which may help modeling variations over time. Furthermore, higher order properties can be encoded, such as positional derivatives or a class like motion-type. The abstraction level of the state space may even be raised, such that the state represents a current action and a cycle variable for that action. Often the reduction of dimensionality brought in by an abstract description allows to approach a tracking problem, e.g. instance choosing a suitable small set of degrees of freedom manually or reducing a larger set via Principle Component Analysis (PCA) of sample data.

The existing gap translating state and image information is covered by appearance and transformation model (Fig. 1.14, right). In this work the appearance model lays emphasis on modeling an actual, possible complex appearance. Such model describes local properties that are associated with the target and that will allow distinguishing it from the background. Once the target moves, the transformation between it and the capturing system is changes. These changes are described by the transformation model.

1.2.2. Appearance

The appearance model maintains information describing the known and often time independent properties of the target. While it is possible to motivate a model without taking into account a concrete appearance, often an initialization phase is used to at least adapt parts of the desired model to the current situation. The appearance model holds that information about the appearance of the target. This can for instance be the distribution of colors or other visual cues, or a more abstract representation. The appearance model is used to obtain a value for the matching quality between the current image and the proposed, transformed appearance. It requires a (transformed) target representation and a comparison function. Figure 1.15 depicts basic cases: In case of template matching, the appearance model holds an image representation and uses a voxel-wise matching function. In case of feature matching, the representation holds features which are compared in feature space. This section will give a brief overview of the wide range of approaches.

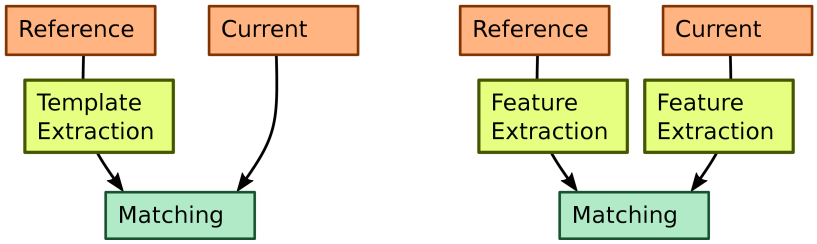


Figure 1.15.: Image and feature based matching. Left: In image based matching, the (warped) content of two images is compared. Right: Feature sets are extracted from both images and then compared.

Feature Descriptions The description of the appearance of the target model may contain several aspects. In particular, it is intended to describe *which* properties *where* to expect and possibly also *which not*. Localized models store individual information about particular features by assigning it to vectors relative to the targets center. Template based approaches define the intensity expected in a particular location relative to the target center. Non-localized models define location more roughly and use the collection of features without a precise spatial information. Such models summarize all features belonging to the target model region. These approaches typically employ histograms which count the occurrence of particular features within the region. These classes of approaches may be combined. Structural histograms for instance divide the target model into localized sub-regions and evaluate intensity histograms in each of them. On the other hand, histograms of spatially localized models may be detected within the target region. Generally, the concept of such combination leads to feature descriptors, such as Scale Invariant Feature Transform (SIFT) [Low99; Low04], Speeded Up Robust Image Features (SURF) [BTG06], Gradient Location-Orientation Histogram (GLOH) [MS03].

Another way of encoding the relationship between a voxel or patch to a fixed set of neighbors is given by the approach of Local Binary Patterns (LBP) [OPM02]. The voxel in question is compared to its neighbors by a decision rule which leads to a binary outcome. For instance the outcome is 1 if the outer location is brighter than the inner and 0 otherwise. The list of neighbors is usually sorted by their arrangement around the voxel of interest and the list of binary values is ordered accordingly, leading to a local binary pattern. Thus, for a local binary pattern with a neighborhood of 8 points, the feature of $2^8 = 256$ states. Neglecting the orientation, the feature is reduced to 59 possible states, since entries with only a single 1 are treated as identical for instance. The pattern which is assigned to a voxel typically hold information about the structure, labeling it as edges, corners, lines or blobs for instance. Generally, a histogram of LBP within a certain neighborhood is used, forming a feature vector.

Evaluation Measures for similarity or dissimilarity often are crucial element of the matching process itself. These measures find utility in different areas. When matching two image patches, often two vectors listing the voxels intensities are compared.

Comparison measures are however more general. Instead of intensity vectors, one might also use color histogram values or even arbitrary feature vectors. For each application, there exist dedicated comparison measures.

Among these are correlation measures such as Pearson's correlation [Pea96] and Spearman's rank correlation [Spe87]. Methods based on local gradient sign change assume that the number of signs in the reference and noisy current image is maximal [VL84; Ven+83], when images are most similar. This stochastic effect can deterministically be emphasized by adding correlated noise [Ven+88]. Measures of mutual information use the joint probability density (joint histogram), to determine similarity. If images match, the entropy of the joint histogram becomes minimal. For two not necessarily normalized feature vectors $\vec{A} = \{A_i\}_{i=1}^{N_V}, \vec{B} = \{B_i\}_{i=1}^{N_V}$, different measures can be described. Few measures related to this work will are presented in more detail:

Cosine similarity measures the direct agreement of the normalized vectors.

$$\mathrm{CS}\left(\vec{A},\vec{B}\right) = \frac{\vec{A}\vec{B}}{\|\vec{A}\|_2\|\vec{B}\|_2} = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}} \qquad \text{Cosine Similarity} \qquad (1.2)$$

It is equal to the cosine of the angle between two vectors $\vec{A}$,$\vec{B}$ and independent of their scale. Cosine similarity finds applications in text mining and data clustering. If the vectors themselves are reduced of their means $\mu_{\vec{A}}$, $\mu_{\vec{B}}$ before cosine similarity is applied, the resulting measure is referred to as Pearson's [Pea96] correlation coefficient $\mathrm{CORR}(\vec{A},\vec{B}) = \mathrm{cov}(\vec{A},\vec{B})/(\mathrm{var}(\vec{A})\,\mathrm{var}(\vec{B})$

$$\mathrm{CORR}\left(\vec{A},\vec{B}\right) = \frac{\sum_i (A_i - \mu_{\vec{A}})(B_i - \mu_{\vec{B}})}{\sqrt{\sum_i (A_i - \mu_{\vec{A}})^2}\sqrt{\sum_i (B_i - \mu_{\vec{B}})^2}}. \qquad \text{Pearson's correlation} \qquad (1.3)$$

A perfect linear correspondence yields $\mathrm{CORR} = 1$, a perfect inverse correspondence $\mathrm{CORR} = -1$ and the absence of correlation $\mathrm{CORR} = 0$. In signal and image processing, the correlation of two series or patches is often evaluated in terms of a temporal or spatial displacement, respectively. The spatial function is then also referred to as cross correlation in general and normalized cross correlation (NCC), if Pearson's correlation coefficient is applied.

A further measure closely related to the cosine similarity is the Bhattacharyya coefficient, which is used to compute the similarity of histograms for instance [CRM00]. It is motivated for computing the similarity of two discrete, normalized distributions and formulated as

$$BC(\vec{A},\vec{B}) = \frac{\sum_i \sqrt{A_i B_i}}{\sqrt{\sum_i A_i}\sqrt{\sum_i B_i}}. \qquad \text{Bhattacharyya Coefficient} \qquad (1.4)$$

Measures of dissimilarity are lowest, when image pairs match best. Often these are based on intensity differences. This is the case for the $L_1 = \sum |A_i - B_i|$ and $L_2 = \sqrt{\sum (A_i - B_i)^2}$ norms or related measures such as median absolute distances. The sum of squared distances

$$\mathrm{SSD}\left(\vec{A},\vec{B}\right) = \sum_{i=1}^{N_V} (A_i - B_i)^2 \qquad \text{Sum of Squared Distances} \qquad (1.5)$$

is used to define the target of least squares minimization. Generally, these measures do not account for global intensity changes. Similar to CORR a normalized input can be used (e.i. $A'_i = (A_i - \mu_{\vec{A}})/\sigma_{\vec{A}}$) in evaluation of the measure [EP08]. Each dissimilarity measure may be reformulated into a similarity measure, if required.

1.2.3. Transformation

While the appearance model may hold locally permanent spatial information, the transformation model defines additional transformations caused by motion and deformation.

In the most simple case, transformations are linear. More complex transformations can be separated into deformable and articulated models (Fig. 1.16). These consist of multiple parts, which undergo transformations with mutual constraints. Articulated models formulate interdependencies between different parts explicitly, and relative to each other. For instance in human pose estimation, the relative angles of limb joints are the degrees of freedom of the model. Since these are formulated relative, a hierarchical relationship of model components exists naturally. Deformable models use independent degrees of freedom to describe the motion of individual parts. They use an interaction term in the objective function to integrate constraints and dependencies implicitly.

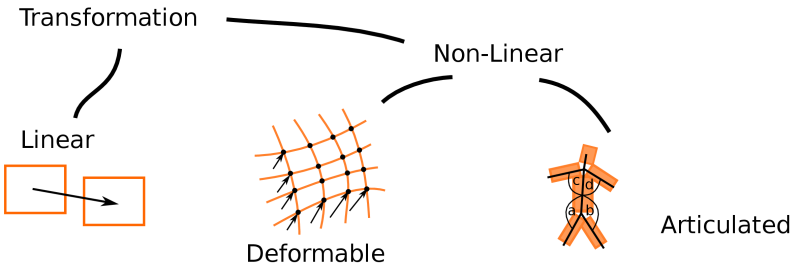


Figure 1.16.: Basic types of poses states and related transformations.

Linear Transformation Linear transformations have a wide range of applications in transformation models. They are used to describe displacement and linear deformation of a target region, and are further distinguished by the degrees of freedom they make use of. Rigid transformations allow translations and rotation. A similarity transformation additionally allows for scaling and an affine transformation further allows for shearing of the whole object.

Non-Linear, Deformable Transformation In deformable models, only an implicit dependency among individually transformed elements exists. In video tracking, deformable models are found rarely, since the image projection process barely provides the necessary constraints for 3D deformation modeling. A notable example for 2D dimensional deformable video tracking are Active Contour Models (ACM) [KWT88].

ACM, also referred to as 'Snakes' is an algorithm which tries to estimate the outline or an edge in an image by the introduction of an energy term to a point based spline model. Basic idea is that each configuration of a contour can be assigned an energy which may be influenced by various sources. The optimization process then starts out with a configuration close to the optimum and iteratively changes it in order to minimize the configurations total energy. Much more than video tracking, non-linear image registration is occupied with the computation of a dense image correspondence between current and reference image. Generalizing the formulation of snakes, the discipline poses the central question as an optimization problem for an energy of shape

$$E(I_R, I_C, W) = \mathcal{M}(I_R, W(I_C)) + \mathcal{R}(W) \tag{1.6}$$

The two terms describe data and model fitness. The data term encodes the cost $\mathcal{M}$ of the dissimilarity of matching reference image I_R and current image, transformed by W. Model or regularizer term $\mathcal{R}$ is used to add constraints into to the deformation W itself.

Transformations W may be motivated by physical models or interpolation theory. Physical models assume physical properties of the matched model [SDP13]. These include elastic models, viscous flow and others. The solution of the occuring partial differential euqation may require a high number of degrees of freedom. Approaches from interpolation theory reduced this number, as that they allow specifying the transformation field by a reduced set of support points. Between these points, the deformation field is interpolated. Methods differ in the choice of interpolation functions.

The regularizer $\mathcal{R}$ is used to integrate constraints into the model, which may for instance preserve topology, volume or impose a particular smoothness of the result. The matching function $\mathcal{M}$ is, in the light of video tracking, part of the appearance model.

The regularization in actual medical image data may exhibit a spatial and directional dependency. For instance, bones and inner organs show different rigidity. Furthermore, the displacement field may be discontinuous at the interfaces of different organs. Such discontinuities may have to be defined beforehand [XCX11; Sch+12; Der+15] or integrated into the optimization process [KFC11; Kir+16; Pap+18].

Articulate Transformation Articulate transformations introduce model constraints explicitly, modeling relationships between them by dedicated degrees of freedom.

Often a skeleton model is used to describe the basic ($\approx$ 20–30) degrees of freedom of the human body, or the hand. The hierarchical nature of these models allow for a sequential, hierarchical detection scheme in which the most prominent parts are detected first. For example, the authors of [GD96] used a graphical model to detect torso and head; then refine the result to limbs. When the pose state of such model is successfully matched, the result state has a direct interpretation as centroid position and angles of the skeletal joints. Generally, the body parts in such model are approximated by simple shapes. Cylinders [DBR00; SBF00; DR05; Hus07; Sai+15] and ellipsoids [YM18] are often applied to approximate the individual parts. Volumetric approaches compose individual limbs as mixtures of Gaussian density distributions [Sto+11].

In medical image analysis, similar approaches have first been documented for skeletal registration in micro CT data of mice [Bai+07]. They were later adopted to human skeletons [YPJ14; YJ14; Fu+17]. In these approaches, bones of bone sections are registered successively. The elements are constrained by explicitly modeled joints.

Composition Generally, transformations can be joined in order to reach a successful result more easily. Often, linear transformations provide the basis which compensate for global changes in position. For instance, in brain image registration, the brain is often coarsely aligned by a linear transformation before being matched using deforming transformations. Such combining approaches are often found for deformable matching; whereas articulate transformations can naturally integrate the linear positional changes.

1.2.4. Statistical Modeling

Both the transformation and appearance can benefit from statistical modeling. A statistical model requires samples of training data in order to derive statistical information from it. A generative model is able to sample from the joint distribution of state and observation $p(\text{State}, \text{Observation})$, whereas a discriminative may only sample from the conditional distribution $p(\text{State}|\text{Observation})$. While a discriminative model discriminates the observation data to determine a state, a generative model tries to also model the source.

Typically, statistical modeling is applied in tracking applications, as its is subject to more variation and more data is available. In medical image registration, statistical models appear in inter-subject registration via atlases [Lor+02; Che+10] or modeling organ motion [Mel+07].

Generative Models Generative models can be applied to increase the efficiency of the state description by learning a projection to a more meaningful state space than the one used to record the training data in. They for instance include mixture distribution models and models for subspace learning. For instance, the statistic variations of a scene background can be modeled by these approaches. The authors of [SG99] model the background pixel intensities as Gaussian mixtures, instead of fixed values. In [ORP00] a PCA analysis is used to explain major changes in the global illumination during the day. An example for subspace modeling is the Active Shape Model (ASM) [CT92; Coo+95]. A shape in this respect is the surface, or contour, which is defined by a set of control points. An ASM is constructed from a set of training shapes, each specified by control points which have correspondence among samples. They are matched by Procrustes alignment (*Geometric Model Based* (1.2.7)). The remaining variations in the model are expressed a via covariance matrix. With Principle Component Analysis (PCA) a fixed size set of the eigenvectors ranked for largest eigenvalues is extracted. These span the space of major motion components. The number of coefficients required to form a linear combination of these 'principle components' is commonly much smaller than that of the individual control points. In human pose estimation, the state can be reduced as far as

to describe it by Action Primitives [HWG07]. Active Appearance Models combine active shape models with an additional modeling of the texture features [CET98; CET01].

In regards to medical applications, population based motion models have been presented. A range of models exist for motion induced by respiration [McC+13; Pre+14], cardiac activity [Yu+13], or for prostate deformation [Hu+08].

Discriminative Models Discriminative models are primarily found in applications of the appearance model, since for predictions in the transformation model, a generative one would be required. These models are trained to discriminate a set of labeled input samples. Given a sufficiently large set of positive and negative examples, random forest classifiers can be trained to distinguish target and background. On-line [Saf+09] variants can be used for tracking. In [Zha+06] the considered patches for on-line classification are provided by a Particle Filter approach. An online random forest classifier is also used in [Wan+17]. In order to account for appearance changes, algorithms presented in [XSL13] and [Oro+12] repeatedly update the set of used patched. In an exploratory approach, the authors of [TSB17] use random forest to learn the correspondence between simulated state and measurement sequences. This can be seen as a sample of learning based approaches for tracking and registration [Vos+17; Yan+17].

1.2.5. Prediction

There exist different classes for estimating the current region of interest from the prior result. Algorithms may differ in the method used to predict the next location and in the method they sample it or use it for assignment (Fig. 1.17).

Prediction can roughly be separated into three categories. The most simple assumption is that no information about motion is available. In this uninformed case, conservative choices have to be made. It is possible to expect the target somewhere within the imaged region and conduct a global search. More optimistic guesses are carried out in the vicinity of the same location and result in a locally bounded search.

Second, implicit information may be obtained from the pair of reference and current image itself. For instance Mean Shift and Optical Flow — which will be presented in this section — may provide suggestions about motion directly based on derivative functions. This yields cues on what motion appeared in image space.

An explicit prediction can be used, if a state space model is available, which allows the prediction of a future state from the prior step only. Such predictions may range from a linear predictions of a Kalman Filter (KF) to arbitrary predictions of a Particle Filter (PF). These will be covered in detail in section *Recursive Bayesian Estimation* (1.3).

The sampling process for the predicted region depends on the tracking algorithm applied. The predicted region may be sampled uniformly, such as in template matching. Furthermore, a Gaussian distribution can be used to describe the prediction, which may be used as is or be sampled from. The Particle Filter is furthermore capable of using an arbitrary, discrete set of predictions as they are. Implicit techniques such as Mean Shift and Optical Flow generate a prediction of motion for the differences between the prior and current images themselves.

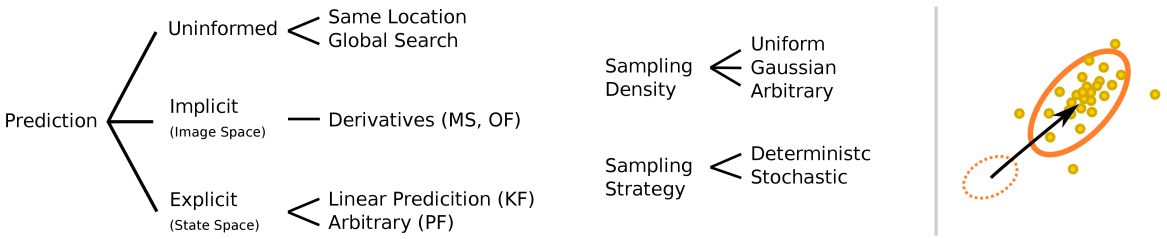


Figure 1.17.: Left: Options for predicting the new location of the object and sampling the predicted region. Right: Sample of linear prediction of Gaussian region with stochastic sampling.

If the predicted region is used to restrict the area of search for the next observation, it may be sampled from. Usually, a particular type of sampling is associated with one method. Template matching uses deterministic, uniform sampling within a rectangular bounding box. More elaborate non-uniform deterministic patterns for sampling have been developed, some of which can be observed in video block matching for video compression, which is closely related to template matching: Techniques such as diamond sampling [ZM00] are however applied iteratively. Similar options exist for regions defined by Gaussian distributions. While obviously stochastic approaches exist to sample from them, deterministic patterns have been introduced which approximate the Gaussian density in order to save computation time [PDD00]. Even for arbitrary distributions given by Particle Filters, fully random and deterministic (re-)sampling can be applied [HSG06]. While explicit predictions by Kalman and Particle Filter will be discussed in section *Recursive Bayesian Estimation* (1.3), three widely applied samples of uninformed or implicit methods are briefly presented here. All of these methods generate point estimates for each object or feature tracked as that they do not provide information about their uncertainty.

In Template Matching [PKB13] two regions of reference and current image are compared to each other. The technique matches one landmark, described by the template region from the reference into the current image. Commonly the rectangular template is compared to rectangular candidate patches in the current image and the center of the patch with highest similarity is chosen as updated location. In the next frame new candidate patches are selected in a rectangular neigborhood around the last result location. Although speeded up variants of template matching have been suggested [BH01], the comparison process in this technique is computationally expensive.

The term Optical Flow describes the vector field representation of motion between two related images of a scene. Assuming the total intensity in an image is conserved $dI/dt = 0$, the continuity in the spatiotemporal flow of intensity is expected. As the problem is ill-posed, further constraints have to be imposed to find a solution. Lukas and Kanade [LK81] assume that the intensity within a local region around the considered location is constant. This allows the computation of a sparse Optical Flow field. The vector field is not guaranteed to be smooth. Horn et al. [HS81] approach the solution

to Optical Flow by adding the constraint of smoothness to the vector field. Computing a dense vector field, a second weighted term of quadratic gradients is added into the evaluation of the target functional. The computation requires more computational time than the Lukas Kanade method. While in medical image registration, the Optical Flow field is the measure of interest, in video tracking Optical Flow is sometimes solely used as means to segment the target structure from the background.

Mean Shift is a kernel based mode-seeking algorithm [FH75]. It seeks the extrema of modes in a distribution or energy function. The algorithm is started with an initial poisition in state space. It converges towards a nearby extremum by the iteration of two steps. First, samples of the distribution are weighted by a kernel localized at the current position. A weighted mean position is computed, which is then used as the new center. This is repeated until a convergence criterion is met. The kernel size is referred to as bandwidth and influences the convergence behavior of the algorithm. Mean Shift can be applied to clustering and tracking applications. In video tracking, it can for instance be used on confidence maps computed by comparing the color histogram of the target to the image via the Bhattacharyya Coefficient. The work [CRM00] uses a Taylor expansion of the Bhattacharyya coefficient to deterministically set the mean-shift step.

1.2.6. Optimization

Other than prediction, the optimization is a component, which is used for detailed pose recovery in video tracking and extensively in medical image registration. For an overview on optimization in medical image registration the reader is referred to [SDP13]. Algorithms may operate on a continuous or discrete space and can be classified into deterministic and stochastic solvers.

A majority of approaches in image registration applies a continuous optimization, which refers to the obtained result state being continuous. In these, a defined functional needs to be optimized. Deterministic optimization procedures are given by common iterative solvers for non-linear equations. These use local approximations to the target functional to derive the direction and size for the applied steps in state space. A non-linear least squares problem may for instance be solved using the Gauss-Newton algorithm, its extension in the Levenberg–Marquardt [Lev44; Mar63] algorithm or quasi-Newton algorithms such as Broyden-Fletcher–Goldfarb-Shanno [Fle70; Fle70; Gol70; Sha70], conjugate gradient descent [Pow64] or downhill-simplex [NM65]. There further exist non-deterministic methods for optimization. Stochastic gradient descent [Rud16] takes gradient descent steps which are based on random subsets of the data. This approach introduces randomness into the descent, but can be much more efficient when applied to large input data. Evolutionary optimization methods have been used for matching, but are not widely applied [San+12]. Optimization processes based on sequential Monte Carlo Methods will in more detail be presented in *Particle Filter for Articulate Pose Estimation* (1.3.6).

Discrete methods use a discrete set of available deformations in each step of the optimization. This allows formulating the objective by means of Markov Random Fields (MRF) [FH06]. Such model is an undirected graph in which nodes represent individual

entities to be transformed. The random variable of such a node takes discrete labels and each node holds the probability distribution of these values. Interaction between nodes is described by the undirected edges of the graph. Three methods for optimizing an MRF exist. For two-label cases, the problem can be reformulated into a min-flow max-cut problem [FF56]. Other approaches [Kom06] are found in belief propagation methods [Loe04] and linear programming. The spatial layout of the graph used in these techniques does not necessarily need ot be regular. For instance [Hei+13] optimizes the layout of MRF nodes on a minimum spanning tree.

Multilevel Optimization While the state of the deformation is provided by a possible high dimensional pose vector, the cost function is a scalar. The cost landscape can exhibit many local optima, which distract the optimization process from reaching the global optimum. Furthermore, the evaluation of a high dimensional state on highly detailed data is more expensive than evaluating a low dimensional on low resolution data.

The idea of multilevel approaches [HM04; Paq+06; Ste+06; Moh12] therefore is the following: The overall deformation can be regarded as a composition of deformations occurring on multiple spatial scales. Large scale deformations require only a rough description by a coarse model. Thus, matching the coarse transformation first, can reduce computational cost. The cost of this coarse level can further be reduced, if the image data provides features on a low resolution scale that shows the desired coarse deformation. Thus, both the model and input data can be reduced in detail in order to provide for an inexpensive initial guess of the transformation. The multilevel optimization then uses this guess to proceed with the optimization on the next level of detail. The process continues until the target level of detail is reached. This way, much unnecessary optimization of high detail can be avoided on the low detail level. The iterative scheme of multilevel optimization agrees well with the iterative nature of most optimization strategies. In medical image registration, the use of image pyramids or wavelets is common. Between each level of the pyramid the image resolution is changed by a constant factor. For instance, the resolution can be doubled in each image dimension, using a Gaussian kernel filter for downsampling. While the reduction of model detail is up to the designer of the algorithm, a multilevel approach to the input data is not always possible. If model and input are reduced in detail simultaneously, the reduced input data is required to provide motion information on the reduced level. The reduction of model detail is a powerful tool for increasing the convergence rate and will be applied in this work. A general multi scale approach can be encodede in a hierarchy of operations, which in turn can be represented by a hierarchical graph.

1.2.7. Data Association

In the presence of multiple, individual observations, these need to be associated to individual parts of the model. There are two particular association problems to be discussed here. The first concerns detection of point wise associations, when an underlying geometrical model is known. This is a question which is posed in both image registration

and video tracking. The second concerns the association of measurements to individual objects, which is predominantly a question, occurring in video tracking. In the following, different approaches to solving association problems are presented.

Geometric Model Based In image registration, a typical association problem is the task of finding a matching transformation between two sets of points, which is referred to as geometric registration [SDP13]. Depending on the available data, these methods aim for finding the best transformation between two sets of (1) equal size and pairwise associations and (2) possibly unequal size and unknown associations. The implicit assumption to solve these problems is that there exists a geometric relationship between the two sets, which can be exploited for solving the association as well as the matching problem.

Procrustes alignment [Goo91] is used to match two point sets of known pairwise correspondences. After reducing both sets for their centroids, they are aligned by minimizing inter-point distances using a singular value decomposition. If the correspondence is unknown, Iterative Closest Points (ICP) matching [BM92] may be applied. The algorithm iteratively executes Procrustes alignment on the changing set of closest point pairs between both input point sets. There exist mixed problem, in which subsets of the data are pairwise associable — for instance, if points have a class label which indicates the type of feature they are derived from. Furthermore, one of the sets may be given by a parametrized geometric object. In this case, one needs to determine, which points are associated to the object and which are not — only the associated points are then used for fitting. A common basic algorithm in this context is random sample consensus (RANSAC) [FB81] matching, which iteratively fits the model, excluding outliers that contradict the fit too strongly.

These techniques are equally applicable for tracking tasks. Since they construct the model matching without a high level estimate, they are also referred to as bottom-up methods [Mar07]. Top-down approaches in this respect refer to methods, which use predictions to suggest and measure to confirm a model state and possible associations. These methods are related to the plain linear and deformable image registration, which acts on voxel properties, referred to as iconic methods [SDP13].

Observation-To-Object Video tracking poses further types of association problems. These are related to multiple, independently moving and possibly indistinguishable objects. In applications without unique models of targets tracked, the association between observations and a track are crucial. Two common assignment problems exist between measurements and targets. The first association problem, which is very much related to the tracking task itself, is the separation of the true target measurement from spurious measurements called clutter. An additional problem may be introduced when more than one target is present: Targets have to be distinguished from each other. With increasing complexity, cases of targets appearing and disappearing have to be managed. The problem of measurement assignment has been approached in various algorithms. A number of basic approaches are given in the following:

The Nearest Neighbor Standard Filter (NNSF) [CL90] assigns the nearest measurement which was observed relative to the predicted position to update the state. When multiple targets are to be assigned multiple observations, the nearest neighbor optimization can be modeled as the weighted connection of a bipartite graph (a graph with edges only between two sets of nodes). A solution to the related cost minimization is found for instance in the Hungarian algorithm [Kuh06]. The Probabilistic Data Association Filter (PDAF) [BDH09] considers cases in which zero, one or multiple measurement may be related to the target. Within an acceptance region around the estimated state (validation gate), all observed measurements are integrated into the result. PDAF reduces a complex density back to a Gaussian distribution in each step. The Joint PDAF (JPDAF) [FBS80; CB84] is an extension from single to multiple targets. The multiple hypotheses tracker (MHT) [Rei79], [Bla04] handles multiple hypotheses about temporal tracks. Other than NNSF and JPDAF, MHT is able to consider more than the very last step into the association problem. A 'track' in this respect refers to a temporal sequence of positions over a limited set of recent frames. Since the number of possible tracks increases exponentially with the track duration, pruning techniques are required. The Probability Hypothesis Density (PHD) Filter is based on random finite sets, for recursively estimating the number and state of multiple targets. The main goal of the description via random finite sets is to allow a flexible integration of a varying number of targets during tracking. The PHD filter does not maintain target association between steps. Thus added association techniques include track-to-estimate-association [PVS07], multiple hypotheses filter [PVS07] and clustering [PVS05]. The approach in [Liu+15] maintains association between consecutive steps by minimizing the overlap of successive particle clouds.

Hierarchical Graph Hierarchy is a concept, which is applied to organize and structure a complicated environment. This environment is represented as a set of elements and to be arranged in such a way that the arising structure allows insight into the interdependency of its items or groups. Examples for hierarchies can be found in many various disciplines from social structure, language and ancestry to infrastructure of cities and the world wide web. Elements in a hierarchy may represent concrete, distinguishable units (e.g.chicken, grapes and pasta in the food pyramid) or abstract groups and subgroup (e.g. different mechanical parts belonging to an engine, belonging to a combine harvester).

Hierarchies assign levels to elements. Each element exhibits a certain trait or label which defines its level. It can for instance be related to an epoch, a rank among element or a level of detail. Elements of a specific level are associated with those on levels directly above and below. In a Mono Hierarchy, an element can be associated with at most one parent above, but multiple children elements below (Fig. 1.18). A node without parent is referred to as a root node. Most hierarchies exhibit a single root. The complete connected graph, including the root, is referred to as hierarchical tree; multiple trees as forest. Nodes without further children are called leaf nodes and can be connected directly to their root or via intermediary internal nodes.

In tracking applications hierarchical graphs can be used to model a relationship be-

tween different parts of the appearance. For instance, in [Wan+16], the target is first sought globally by a single bounding box. The result is then optimized on two further hierarchical layers which match the location of super pixels and key points, respectively. Further approaches will be presented in *Particle Filter for Articulate Pose Estimation* (1.3.6).

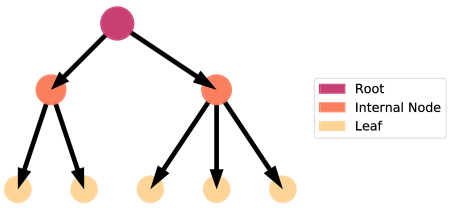


Figure 1.18.: A hierarchical tree with three levels.

Clustering Clustering is a way to detect groups of similar entities among a larger set. Detected similarities can be used to differentiate elements of the individual groups and to reduce the description of the input data. In tracking applications, clustering may for instance be utilized in order to reduce the description of a multi-modal mixture distribution, or with the aim to solve an assignment problem with multiple targets.

Clustering groups a set of input entities into sub-components, or clusters. The formation of clusters depends on the type and parameters of the clustering algorithm.

Partitional Clustering splits a set of input entities. Several approaches exist. For instance, if only pairwise relations between input entities are known, these can be described as a graph. Graph based clustering methods then determine a split of the graph that minimizes the dissimilarity among the result clusters. For these, only the distance between entities is required. If entities exist in a common space, which exhibits a distance measure, centroid based clustering methods can be applied. These try to optimize the distances between the available input entities and output cluster centroids. The most prominent example for this is the k-means algorithm [Llo82]. Parts of this work will rely on it.

In the k-means algorithm the number k of target clusters is used as a parameter. The k-means algorithm assigns the input elements to k clusters. The input elements have a property which they are clustered by, which serves as a position in a Euclidean space used for clustering. A cluster center exists in the same space and Euclidean distances can be computed between elements and cluster centers. There exist different variants of k-means algorithm. In Lloyd's formulation [Llo82], the algorithm consists of an iterative update scheme. In each step cluster centers are updated by (1) computing assignments and (2) using the assignments to update the center position. In the first step, each element is assigned to the nearest cluster center. Then each center is updated by computing the arithmetic mean position of the elements assigned to it. This process

is repeated until the algorithm converges. The k-means algorithm shares similarities to the expectation-maximization (EM) algorithm. However, the EM algorithm uses a soft assignment to clusters and computes cluster centers by a weighted mean. It approximates clusters as Gaussian distributions and therefore not only updates cluster centers, but also variances. Several other variants of k-means exist. In this work, spherical k-means will be of interest, which differs from k-means by the distance measure applied. Spherical k-means maximizes the cosine similarity (Eq. 1.2) between centroids and clustered entities.

Hierarchical Clustering does not explicitly separate the input set of elements into a single cluster configuration. Instead, a hierarchical cluster tree is built, in which the child clusters of a parent cluster share the same set of elements. The root of the tree holds all elements, which are then further split as levels of the tree progress. The splits along the tree may be ordered by a cluster cost, such that by choosing a desired value, a particular cluster configuration is obtained. In hierarchical clustering, two categories of algorithms can be distinguished. Agglomerative clustering iteratively joins single elements into larger components. Several methods exist, which primarily differ in the method for assigning new elements to existing clusters or fusing existing clusters. Divisive clustering, on the other hand, iteratively splits clusters into sub-clusters. This work will use a divisive hierarchical clustering scheme, in which nodes are created in a k-means like fashion oriented on Lloyd's algorithm.

Hebbian Learning Another form of establishing associations is learning them from examples. In associative learning as described by Hebb [Heb50], joint activity of neurons is promoted by strengthening the related neural connection between them. The notion paraphrased as 'What fires together, wires together' [Sha92] has spawned implementation into simulated artificial neural networks, which learn by success and failure. In its initial formulation, the strength $w_{i,j}$ of a neural connection between neurons i and j is updated via $\Delta w_{i,j} = \eta\, x_i y_j$. In this x_i and y_j are the (input) activity of neuron i and (result) activity of neuron j. The value η is referred to as learning rate. Several different formulations of this rule exist. Oja's Rule introduces an additional negative term to constrain the growth of weights [Oja82]. It is further generalized in the 'Generalized Hebbian Algorithm', where it is used for learning a singular value decompositions [Gor06]. Generally, by applying this rule to a directed, bipartite graph, it can be used as means to learn an association. This work will use a rule inspired by Hebb's in order to sequentially update association data.

1.3. Recursive Bayesian Estimation

In real-time applications incoming data has a sequential nature. Sensors update their state in regular intervals and provide the result to a workstation for further analysis. For tracking and navigation applications, such sensor might be an optical camera, a laser range finder, a radar system or any other type which allows for the inference of the targets state properties. These may include features such as position, orientation, velocity. If these updates are sufficiently frequent, it is very likely that subsequent

updates are related. The state information provided by the current update will not be found independent of the prior. For instance the new location of an object is constrained by its prior position, its prior velocity and the time which passed from the prior update. Furthermore, since sensors provide noisy measurements at no time can the target state be determined with complete certainty. There always remains a residual uncertainty in an estimate. The presence of uncertainty and the sequential dependency of states leads to the concept of conditional probability. The iterated application of conditional probability to repeatedly updated data leads to sequential tracking methods. The Kalman Filter yields an analytic solution by imposing strong conditions on the tracking process. The Particle Filter applies Monte Carlo approximation to remove many of the restrictions. This section presents the aforementioned concepts and methods in the presented order. Ultimately, this work will employ a Particle Filter for sequential motion estimation.

1.3.1. Bayes' Theorem

Inferring a system state from noisy observations is an important task for probability theory. For instance, it may be of interest to infer the state of a known model from the observations or even infer the model itself from gathered data. One may further be interested in deriving extending information from the data by making predictions about future states or states outside the observed bounds in general.

The probability which is assigned to an event depicts it's likeliness of happening when a certain experiment is undertaken. The probability space Ω includes all possible (countable) events ω_i which are assigned probabilities $P(X = \omega_i) \in [0, 1]$. For reasons of simplicity this probability for the outcome X being equal to ω_i, will be denoted as $P(\omega_i)$. These values for probability range from 0 representing an event which will occur *almost never* to 1 of an event which will happen *almost surely*. As it is assumed that the experiment will have exactly one outcome, the Law of Total Probability demands that $\sum_i P(\omega_i) = 1$ and also assumes that the discrete atomic events ω_i are mutually exclusive. Events may also defined by subset of events in Ω and have a probability of the sum of the included events $A \subset \Omega$ with $P(A) = \sum_{\omega_i \in A} P(\omega_i)$. The complement event $\bar{A}$ holds the complementary probability $P(\bar{A}) = 1 - P(A)$.

Multiple probability spaces may be defined in the same experiment and span a space of joint probability which depicts the probability of the joint occurrence of events from both spaces. In the simple case of two probability spaces Ω_A,Ω_B the joint probability space Ω_{AB} is two dimensional and the probability of an event $A \in \Omega_A$ occurring together with $B \in \Omega_B$ is written as $P(A \cap B)$. For events which are independent of each other, this probability is given by the product of the individual ones. If they are not independent of each other, this dependency is expressed in the conditional probability $P(X = x|Y = y)$ for and event x occurring, provided the other one y has. As before, $P(X = x|Y = y)$ will be abbreviated as $P(x|y)$. The joint probability can also be stated in the following way.

$$P(A)P(B|A) = P(A \cap B) = P(B)P(A|B) \tag{1.7}$$

This expression for Bayes' Theorem is usually solved for $P(B|A)$, $P(A)$ is expressed by conditional terms of $P(A|...)P(...)$ to underline it's interpretation as a normalizing constant to the right hand side.

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_i P(B_j)P(A|B_j)} \qquad \text{Bayes' Theorem} \tag{1.8}$$

In this formulation for a particular event B_i, the right hand sides denominator expresses the probability $P(A)$. The continuous variant reads as

$$P(\theta|\phi) = \frac{P(\theta)P(\phi|\theta)}{\int P(\theta')P(\phi|\theta')\,d\theta'}. \tag{1.9}$$

and is depicted in figure 1.19. The structure of Bayes' Theorem may not only be used for making statements about fixed known parameters and properties but also for inferring them. In Bayesian Inference the structure and concept of the Bayesian formulation of conditional probability is used making assumptions about unknown data and refining these with new data arriving. Bayes' Theorem is read as

$$\text{Posterior} \propto \text{Likelihood} \cdot \text{Prior} \tag{1.10}$$

and the denominators are identified accordingly to (Eq. 1.8) and (Eq. 1.9), whereas the nominator is a mere proportionality constant and not of interest.

Prior knowledge about the state of the variable may exist however, already before a measurement has taken place. In this case, it should be taken into account when gaining information from the measurement itself. Ideally the measurement acts as a refining step for the already existing knowledge. If no knowledge about the state of a variable exists, all states should be equally probable. This Principle of Indifference is based on the symmetry argument that without knowledge, all possible states should be interchangeable without affecting their probability. This knowledge does not necessarily need to be objective, but may in fact result of a subjective notion as well. The choice of a reasonable prior is one subtlety in Bayesian inference. Conditional probability may be given as a probability of an observation given a certain hypothesis. For inference it is necessary to invert this conditional probability: It may be more interesting to get insight into the probability for the hypotheses holding under the given observation. The final result is gained as the *posterior*, which similarly to the prior is represented as a probability distribution.

1.3.2. Sequential Update

In a real-time applications, new information is available with each new update by data. The state of the tracked object can update sequentially. Following the derivations of [DFG01], such sequential system update is defined stepwise

$$p(x_0) \tag{1.11}$$

$$p(x_k|x_{k-1}) \text{ for } k > 0 \tag{1.12}$$

$$p(z_k|x_k) \text{ for } k > 0 \tag{1.13}$$

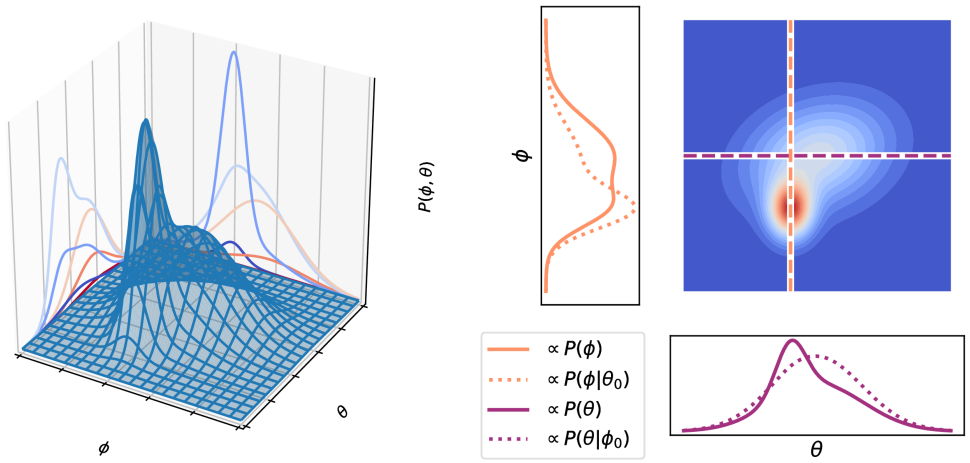


Figure 1.19.: Left: Example of a two dimensional probability density distribution $P(\theta, \phi)$. The sample projections of profiles in different regions of the distribution indicate different conditional probability distributions. Right: Marginal $P(\theta)$ and $P(\phi)$ and two concrete examples of conditional distributions $P(\theta|\phi = \phi_0)$ and $P(\phi|\theta = \theta_0)$ (dotted).

The vector valued state x_k describes the system state in step k. The knowledge about the state is encoded in the distribution $p(x_k)$. The step $k = 0$ indexes the prior state, before tracking commences and $k = 1$ is the first step in which new information is available. In the Markovian approximation (Fig. 1.20), each state x_k only depends on its direct predecessor x_{k-1} (Eq. 1.12). The new information is captured in measurements z_k. Each measurement z_k only depends on the current state z_k. The measurements is linked to the data which is provided to the algorithm.

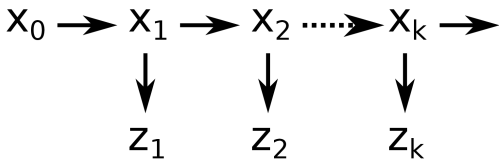


Figure 1.20.: Sequential update of states. Each state x_k depends on its prior x_{k-1}. The observation z_k depends on x_k.

Series of temporal states and observations are expressed as $x_{0:k} \stackrel{\text{def}}{=} \{x_0, ..., x_k\}$ and $z_{1:k} \stackrel{\text{def}}{=} \{z_1, ..., z_k\}$, respectively. At time k, the systems posterior distribution $p(x_{0:k}|z_{1:k})$ may be of interest, but even more the marginal, filtering distribution $p(x_k|z_{1:k})$, holding information about the current state k. From either of these, expectation values

$$I(f) = \mathbb{E}_{p(x_{0:k}|z_{1:k})}\left[f(x_{0:k})\right] \stackrel{\text{def}}{=} \int f(x_{0:k}) p(x_{0:k}|z_{1:k})\, dx_{0:k} \tag{1.14}$$

can be computed. For the filtering distribution, the conditional mean $f(x_{0:k}) = x_k$ and its covariance $f(x_{0:k}) = x_k x_k^T - \mathbb{E}_{p(x_k|z_{0:k})}[x_k]\mathbb{E}^T_{p(x_k|z_{0:k})}[x_k]$ are of interest.

The posterior distribution

$$p(x_{0:k}|z_{1:k}) = \frac{p(z_{1:k}|x_{1:k})p(x_{0:k})}{\int p(z_{1:k}|x_{1:k})p(x_{0:k})\,dx_{0:k}} \tag{1.15}$$

is obtained by the application of Bayes' Theorem and admits a recursive formulation

$$p(x_{0:k+1}|z_{1:k+1}) = p(x_{0:k}|z_{1:k})\frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|x_k)}{p(z_{k+1}|z_{1:k})}. \tag{1.16}$$

The filtering distribution $p(x_k|z_{1:k})$ obeys

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\,dx_{k-1} \qquad \text{Prediction} \tag{1.17}$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})\,dx_k}. \qquad \text{Update} \tag{1.18}$$

These equations require the solution of high dimensional integrals and are therefore not generally solvable. Among the most prominent approximations to them are the Kalman Filter and the Particle Filter. The Kalman Filter assumes that in every step the distribution and transition can be expressed as a Gaussian and therefore only requires propagation of mean and covariance. The Particle Filter on the other hand approximates the distribution by means of Monte Carlo methods and propagates the distribution as a set of sample points. The Kalman Filter will briefly be presented. After introducing the Monte Carlo approximation, the focus will be laid upon the Particle Filter.

1.3.3. Kalman Filter

The Kalman Filter [Kal60] describes an approach for filtering and predicting the state of dynamical system by minimizing the co-variance of system and measurement error. Published in 1960 it has been widely applied in research and applications, especially in navigation and tracking.

The filter separates the process into the unobservable process itself and the feedback about it, which can be gained by sensors. The discrete Kalman Filter assumes discrete temporal step. The state x_t of the system is considered a continuous distribution which has to obey certain assumptions.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{x-1} \qquad \text{State Update} \tag{1.19}$$

$$z_k = Hx_k + \nu_k \qquad \text{Measurement} \tag{1.20}$$

The state transition (Eq. 1.19) describes the linear transition from state x_{k-1} to x_k, where matrix A defines the state transition without external perturbations. This is altered by control u_{k-1} and Gaussian random noise w_{k-1}. The control u_{k-1} affects

the state via the dynamic matrix B. Noise terms v and w stem from distributions of zero mean and covariances R and Q, respectively. States are observable through the measurement Equation 1.20, where linear Hx_{k-1} is perturbed by Gaussian noise v_{k-1}.

The application of this filter describes the propagation of a Gaussian distribution of mean x_t and covariance P_t via two separate steps, *prediction* and *measurement*. The prediction step propagates the covariance and mean via Eq. 1.19:

$$\hat{x}_{k|k-1} = Ax_{k-1} + Bu_{k-1} \tag{1.21}$$

$$\hat{P}_{k|k-1} = A\hat{P}_{k|k-1}A^T + Q \tag{1.22}$$

In order to correct the predicted state by a measurement, the key values Kalman Gain $\hat{K} = P_{k|k-1}HS^{-1}$ and Innovation $\hat{y} = z_k - H\hat{x}_{x|k-1}$ and Residual Covariance $S = HP_{k|k-1}H^T + R$ are computed and used in a set of linear equations which are integrated into the state estimate.

$$\hat{x}_k = \hat{x}_{k|k-1} + \hat{K}\hat{y} \tag{1.23}$$

$$\hat{P}_k = \hat{P}_{k|k-1} - \hat{K}S\hat{K}^T \tag{1.24}$$

The Kalman Gain is motivated such that the filter is optimal, i.e. produces the least squares optimum state estimate.

Constraints of the Kalman Filter are given in its linearity and the restriction to Gaussian states and observations. The extended and unscented [JU04] Kalman Filter relaxes these constraints by locally linearizing non-linear functions: In the Extended Kalman Filter the dynamics are linearized around the current state. In the Unscented Kalman Filter, the propagation of the Gaussian is estimated via a set of σ *points*. These undergo the nonlinear transform individually and are then used to estimate a Gaussian posterior.

1.3.4. Monte Carlo Sampling

There are several integration and optimization problems which can't be solved analytically and to which exact computational solutions are not available. The principle of Monte Carlo simulation is based on replacing a possibly unfeasible continuous integration by sampling the state space at a randomly chosen, finite number of points. The N samples $\{x^{(i)}\}_{i=1}^N$ are drawn from the target distribution $p(x)$ and approximate it as a sum of Dirac-δ-distributions.

$$p_N(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^{(i)}}(x) \tag{1.25}$$

This allows for approximating an integral of the form $\int_x f(x)p(x)\,dx$ by a finite sum which for infinite N almost surely converges as

$$I_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \xrightarrow[N\to\infty]{} \int_x f(x)p(x)\,dx. \tag{1.26}$$

The target distribution p may not be easy to sample from and need further processing before it can be handled. There may however exist another function, which can be sampled from instead, but which require the generated samples to be corrected for differences of the two distribution. Another distribution q used to sample from is referred to as *proposal distribution*. This distribution is ideally similar to the target distribution and easier to sample from. Two Monte Carlo methods applying such proposal distribution are found in Rejection Sampling and Importance Sampling:

In Rejection Sampling, instead sampling of the target distribution $p(x)$ itself, samples are generated from a proposal distribution $q(x)$, which is easier to draw from. This proposal distribution needs to satisfy $p(x) \leq Mq(x)$ with a finite upper bound $M < \infty$. A newly drawn sample from $q(x)$ is only accepted if another sample drawn from uniform distribution $U_{(0,1)}$ is within the acceptance range ($u < p(x^{(i)})/(Mq(x^{(i)}))$), otherwise it is discarded. The less overlap between $p(x)$ and $Mp(q)$, the more samples have to be rejected ($P(accept) = 1/M$). With increasing number of dimensions the rejection rate increases rapidly and makes the algorithm inefficient.

Similar to rejection sampling a proposal distribution is introduced in Importance Sampling. However, instead of binarizing the introduction of the sample into the result, it is weighted. Importance sampling algorithms introduce an importance weight $\omega(x) := p(x)/q(x)$ for bringing target distribution $p(x)$ into correspondence with proposal distribution $q(x)$. This way the estimate of the integral becomes

$$I_N(f) = \sum_{i=1}^{N} f(x^{(i)})\omega(x^{(i)}) \xrightarrow[N\to\infty]{} \int_x f(x)\omega(x)q(x)\,dx. \tag{1.27}$$

This may also be understood as sampling the target density $p(x)$ as

$$\hat{p_N} = \sum_{i=1}^{N} \omega_i(x)\delta_{x^{(i)}}(x). \tag{1.28}$$

Sampling again from this estimate is known as Sampling Importance Resampling [Rub88]. For drawing M i.i.d. samples from the distribution given by N samples an M-fold resampling step may be used. This allows samples from $\hat{p_N}$ to be drawn multiple times and leads to a target density estimate

$$\tilde{p_N} = \frac{1}{M} \sum_{i=1}^{M} \delta_{\tilde{x}^{(i)}}(x). \tag{1.29}$$

Figure 1.21 depicts a sample case of importance sampling. Left images show proposal distributions and samples drawn from them, whereas right plot feature weighted histograms. In top and center plots equally weighted samples are drawn from a uniform or Beta distribution, respectively. Their weighted histograms recover the shapes of the related distributions. In the bottom case importance sampling is used to weigh samples from a Beta distribution to match a uniform distribution.

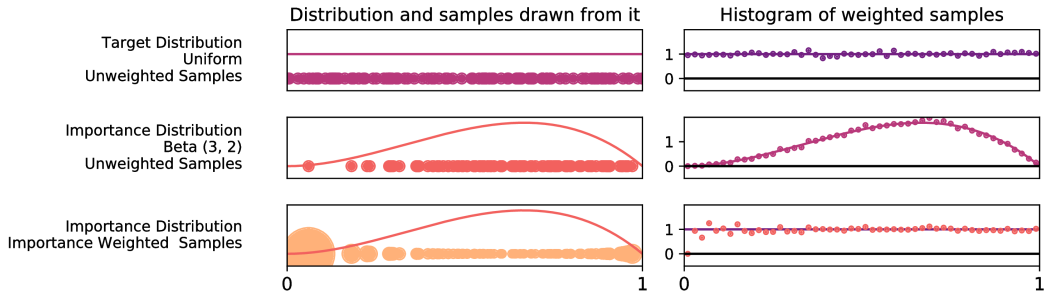


Figure 1.21.: Example: Using importance sampling to sample from a uniform distribution using samples from a Beta distribution. Left: Distribution drawn from (100 samples), and samples drawn from it. Right: Histogram of weighted samples (10000 samples).

1.3.5. Particle Filter

A Particle Filter algorithm [DGA00], also referred to as CONDENSATION [IB98] or SIR algorithm, is used for real-time tracking when non-Gaussian distributions are involved. The prediction and update equations 1.17 and 1.18 are approximatively solved by approximating the filtering distribution by a weighted particle set $\{(s_k^i, \pi_k^i)\}_{i=1}^{N_S}$ such that

$$p(s_k|z_{1:k}) \approx \sum_{i=1}^{N_S} \pi_k^i \delta\left(s_k - s_k^i\right). \tag{1.30}$$

Note that throughout this work states of the particle filter will be denoted by the letter s and weights by the letter π. In this expression a particle at step k has a state s_k^i and a weight π_k^i. As presented in [MC11], the individual weight is then computed as from importance sampling:

$$\pi_k^i \propto \frac{p_k(s_k^i|z_{1:k})}{q_k(s_k^i|z_{1:k})} \tag{1.31}$$

Then the filtering prior distribution is approximated as $\{(s_{k-1}^i, \pi_{k-1}^i)\}_{i=1}^{N_S}$ such that from prediction step (Eq. 1.17) follows the predicted prior

$$p(s_k|z_{1:k-1}) \approx \sum_{i=1}^{N_S} \pi_{k-1}^i p\left(s_k|s_{k-1}^i\right). \tag{1.32}$$

Then, the update equation (Eq. 1.18) is used to determine weights

$$\pi_k^i \propto \frac{p(z_k|s_k^i) \sum_{j=1}^{N_S} \pi_{k-1}^j p\left(s_k^i|s_{k-1}^j\right)}{q_k(s_k^i|z_{1:k})}. \tag{1.33}$$

In the first implementation to visual tracking [IB98], the algorithm draws samples from the predicted prior (Eq. 1.32), such that the weighting process (Eq. 1.31) simplifies to

$$\pi_k^i \propto p(z_k|s_k^i). \tag{1.34}$$

Within this formulation, expectation values can be computed.

$$\hat{s_k} = I(f_t = s_k) = \frac{\sum_{i=1}^{N_S} s_k^i \pi_k^i}{\sum_{i=1}^{N_S} \pi_k^i}. \tag{1.35}$$

The covariance is computed

$$\sigma^2 = \frac{\sum_{i=1}^{N_S} \pi_k^i (\hat{s_k} - s_i)^T (\hat{s_k} - s_i)}{\sum \pi_k^i}. \tag{1.36}$$

Other than these direct properties of the filtering distribution, measures for the quality of the filtering distribution's sample set itself may be motivated. In [MI00] the authors introduce the Survival Rate α and Survival Diagnostic D. The Survival Diagnostic is interpreted as the number of particles which are chosen by the resampling process and get propagated in to the next distribution.

$$D_k = \frac{1}{\sum_{i=1}^{N_S} (\pi_k^i)^2} \geq 1 \tag{1.37}$$

In this formulation, the weights are required to be normalized $\sum_{i=1}^{N_S} \pi_k^i = 1$. The Survival Rate is then motivated as the portion of samples which survive the resampling process in the sense that they remain in the same portion of state space. In the case of large sets the Survival Rate can be estimated as $\alpha = \frac{D}{N_S}$.

Particle Filter Recursion In application, the filter is applied once for every new update by data. A set of weighted samples is used to approximate the probability density. This set is updated in three cyclic steps (Fig. 1.22): resampling, propagation and measurement. In order to sample from the predicted prior, first the prior is sampled from using resampling and then the sampled particles are propagated with individual dynamics.

In the *resampling* step, new unweighted samples are drawn from the prior set of weighted samples. Resampling reduces the probability of continuing to follow unlikely hypotheses and backs up the number of samples in likely regions of the state space. Multiple particles may be drawn from the same parent. The initially motivated procedure of Multinomial Resampling treats each sample draw independently, such that there exists no bias in the choice. After using resampling to sample from the prior, dynamics are applied. The *propagation* step is executed separately for each individual sample. The allows for any dynamic transformation, which is one of the advantages over the restricting assumptions of the Kalman Filter. However, in this work a Gaussian noise process will be used, since exact dynamics are mostly unknown.

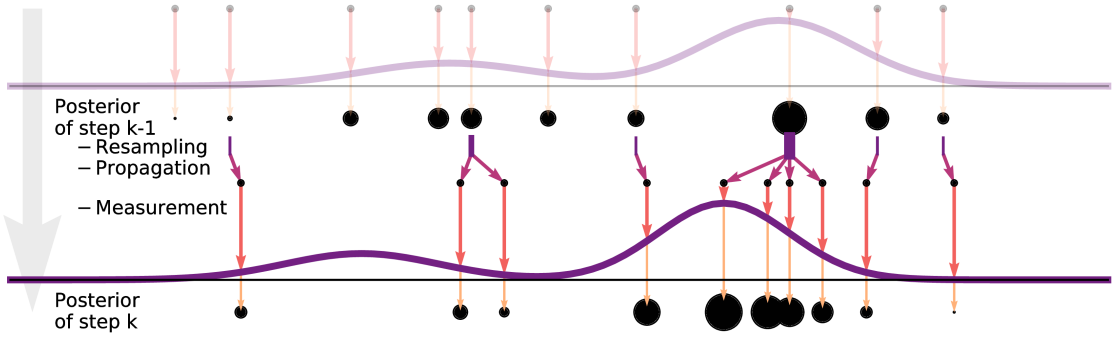


Figure 1.22.: Particle Filter step with ten particles. In the resampling ten new particles are drawn from 7 parent particles. Their states are dispersed by Gaussian diffusion in the propagation step. The measurement step the assigns new weights to them.

Next, the still unweighted particles are assigned weights by the likelihood function $\mathcal{L}$ in update or *measurement* step. This yields the posterior particle set. Finally the posterior particle set is used to compute the posterior position estimate, which for instance may be a weighted mean. The posterior is then used as prior in the next step.

Result: Position estimates $\tilde{x_k}$ for all k_{max} steps.
Initialize particles $\{s_i^0, \pi_i^0\}_{i=1..N_S}$
for $k = 1..k_{max}$ **do**
  Resample s_i^{k*} from $\{s_i^{k-1}, \pi_i^{k-1}\}$
  Estimate s_i^k from s_i^{k*}
  Assign weights $\pi_i = \mathcal{L}(s_i^k)$
  Compute position estimate $\tilde{x_k}$

Algorithm 1.1: Particle Filter Step

1.3.6. Particle Filter for Articulate Pose Estimation

Particle Filter tracking has been applied to articulate motion recovery. Multiple approaches can be found, solving different aspects in the problem of recovering a complex pose. This section will review examples found in the literature.

The term *Layered Sampling* [Sul+99] describes the decomposition of the single resampling procedure into a series using a decomposed prior and likelihood functions iteratively approaching the true likelihood and can be seen a precursor to the annealed Particle Filter. The cost of searching a high dimensional state space can be reduced if the independence of sub spaces is exploited. Then search and dynamics can be applied

independently. Separation of state space into sequentially tracked, orthogonal subspaces has been presented as Partitioned Sampling [MB99; MI00]. Partitioned sampling can not only reduce the cost when the sub spaces are fully independent, but also if one subspace depends on the other, as in a hierarchical model. In this case, the order to evaluation is given by the dependency, but the computational advantage is obtained nonetheless. An application, which not only captures a separable relation between two objects, but an actual hierarchy with multiple siblings is demonstrated for human pose estimation in [MH03].

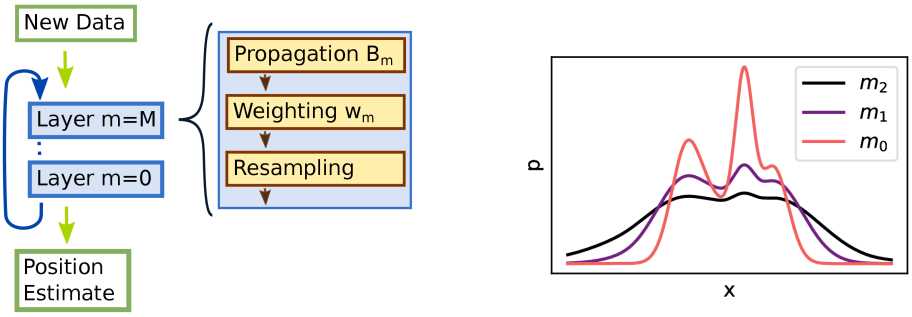


Figure 1.23.: Annealed Particle Filtering. Left: Block diagram. Right: Detail of the distribution increases with each iteration in the schedule progressing as $m_2 \rightarrow m_1 \rightarrow m_0$.

The Annealed Particle Filter uses an annealing schedule to iteratively approach the desired optimum (Fig. 1.23). Simulated Annealing is a stochastic optimization process, which is motivated by the physical description of the annealing process. The technique is base on the optimization process of Simulated Annealing (SA) proposed by [KGV83] in 1983. SA is a stochastic optimization technique, motivated by the approach of a state of low energy in the slow annealing process of metal. The algorithm seeks the optimum of a possibly high dimensional energy function by simulating stochastic jumps. Using the Metropolis-Hasting algorithm [Met+53], these are only accepted if the energy is optimized, or the temperature is sufficiently high. During simulation the temperature is slowly lowered. If this process is executed sufficiently slow, the global optimum is found. Starting with high temperatures allows the algorithm to escape from local minima.

The concept of Simulated Annealing can be applied to high dimensional problems; for instance in pose estimation. The use of an Annealed Particle Filter is proposed by [DBR00], where human poses are tracked using a model of 30 degrees of freedom.

The annealing process of the filter separates a single filter step into M successive ones. With each temporal step one annealing schedule with decreasing estimation variance B_M and β_m increases. Along the lines of SA literature [Nea01], layers are traversed from $m = M$ to $m = 0$, such that β_M increases, representing the cooling process: A dedicated weighting function $w_m(Z, X)$ is chosen as $w_m(Z, X) = w_0(Z, X)^{\beta_m}$ with monotonically increasing $\beta_M < ... < \beta_1 < \beta_0$. Only if one is interested in the statistics of w_0 it is necessary that the final $\beta_0 = 1$. It is not required for finding the maximum. In [DBR00] the choice of β_m is motivated by the measure of Survival Diagnostic $\mathcal{D}$, and the related

particle Survival Rate $\alpha = \mathcal{D}/N_S$. As $\mathcal{D}$ depends monotonically on β_m, β_m can be chosen to fulfill a desired α_m. While there exists no general receipe for the choice of α_m, the authors report good results for $\alpha_m = 0.5 \,\forall\, m$, and covariance $P_m = P_0 \prod_M^{m-1} \alpha_m$.

In an application to hand tracking [WLH01], the authors use constraints of natural hand motion to reduce the configuration space to a more efficient subspace (*Generative Models* (1.2.4)). A soft hierarchical model is introduced in [DR05]. The model hierarchy is implicitly integrated into the model constraints. An Adaptive Diffusion process is introduced which in an iterated optimization determines the diffusion strength in the transition kernel from the variance of the previous result. Applying this process to the single degrees of freedom individually, the model is able to quickly fix the degrees of freedom, which are already determined with certainty. The majority of remaining variance is then applied to the degrees of freedom which are yet unclear. This way, the state space volume search is reduced more quickly, which leads to better performance. The method is proposed similarly to Covariance Scaled Sampling in [ST01; ST03], where the current system covariance is set to a multiple of the prior result covariance.

The paper [Hus07] presents a hierarchical scheme for tracking the human body. In this, single levels of the hierarchy are iteratively optimized before the next level is approached. In [CCP11] the authors use a set of different body models in the different annealing stages. In Structural Annealing they use models with different degrees of freedom, where the low detail layer states are subspaces of the states of high detail layers. Instead of applying the filtering layers in forward direction, they use a backward propagation in order to refine the states of the coarse levels by the results of the fine ones.

2. Gaussian Transition Filter Model

Task of a motion tracking algorithm is to determine the state of an object as it moves. Under ideal conditions, such algorithm is capable of following reliably, while maintaining a low residual error in the track estimate. However, in realistic conditions, perturbations and flaws in the input of the system may hamper the proper function. A malfunction can be formulated as a failure to design and parameterize the algorithm properly. It is then not able to adapt to the nature of the presented environment. For the application of a given algorithm it is therefore of major interest to estimate its capabilities, given a defined scenario.

Aim of this chapter is to provide a simple tracking model for motion of unknown direction and finite velocity, which is to be tracked using a Particle Filter. In order to study the interplay of parameters in a controlled environment, the chapter is entirely based on simulation and will not integrate medical image data yet. It will model and investigate the performance of filtering with a Gaussian state transition kernel, which is a common naive approach to state transition in pose estimation tasks. The properties of this particular model will be derived and validated with respect to three aspects:

- Interplay of parameters in an unperturbed situation
- Residual errors to expect in the state estimate
- Effects of measurement perturbed by a constant level of noise

In further chapters this model will — with slight extension — be applied as tracking model for nodes on a hierarchical tree. Such tree will allow for the description of articulate deformation while requiring low degrees of freedom for the individual nodes.

Structure of this chapter Section *Model* (2.1) begins with a description of the filter, introducing the tracking process and the available parameters. The probability for maintaining a track in a defined environment is analyzed in section *Sampling Stability* (2.1.3). Interplay of parameters is investigated in order to separate regimes of likely tracking success and loss of track using approximations to the initially proposed model. In the following section *Residual Errors* (2.1.4), source and scaling behavior of residual error in the position estimate are motivated for different regimes of operation. Section *Noise* (2.2.3) will propose ways to model the occurrence of noise in the measurement function. Models, estimating thresholds for instabilities, emerging in the presence of noise, are proposed.

The model sections provide grounds for verifying the proposed behavior in an experimental section. *Computer Experiments* (2.2) provides a series of experiments and describes the way data is collected and evaluated. Obtained results and evaluations are presented in proceeding *Results* (2.3). Section *Discussion* (2.4) recapitulates most

important findings. The chapter will conclude with concrete suggestions for the choice of parameters in real-world applications.

2.1. Model

A Particle Filter uses a sampling process to investigate more likely regions for new object positions more thoroughly than others. When detection depends on the sampling process, a viable proposal distribution is required to successfully track an object. The prediction of this viability may be difficult, when non-trivial motion and complex motion models are involved. One option to resolve this predicament, is to sacrifice the detailed description of motion for an approximative one.

In an extreme case, motion is only known up to its magnitude — for instance, when motion is known to be spatially cyclic but its phase unstable, undergoing abrupt changes. Such situation requires a proposal distribution which is unbiased with respect to the direction of motion. The change in state can be modeled by a random walk transition kernel[1]. In this chapter, a *Gaussian Random Walk transition kernel* will be investigated for application in a Particle Filter. It may be interpreted as expecting no motion at all, but a significant amount of uncertainty therein. As the model is of one of the most simple forms, insight gained into it is generic and may therefore be utilized in various applications.

Here, a measurement function is of interest, which unlike a Gaussian, only has a limited non-zero range; such as may be obtained by the application of an image filter. The investigation of the effects of this limitation is a key element of this chapter. This section briefly presents the basic model and introduces two likelihood function: First, a Gaussian likelihood function is motivated, which is primarily used to derive an analytical reference solution. Second, a truncated Gaussian of limited support is motivated, as means to describe the desired filter result likelihood. Further section will be referring to both when deriving success probabilities and expected errors for position estimates. This section will model the success probability for an artificial experiment setting executing a particular, simplified algorithm.

2.1.1. Algorithm and Experiment

The applied tracking algorithm is a simple Particle Filter. It is depicted in Algorithm 2.1. It uses a drift free Gaussian transition kernel $\mathcal{N}(s; 0, \sigma_E^2)$, which means the estimation is a Gaussian diffusion process of fixed variance σ_E^2.

$$p(s_k|s_{k-1}) = \mathcal{N}(s_k; s_{k-1}, \sigma_E^2) \tag{2.1}$$

This chapter assumes that the state space and the space of observations are identical, i.e. that there is no transformation between them. In order to emphasize, that the Gaussian transition (Eq. 2.1) affects the dynamics in state space, the particle states are

[1] i.e. there exist no higher order state space dimensions, such as velocity.

denoted by letter s. A set of N_S samples is used to approximate the tracking density, and is re-sampled in each tracking step. Samples $i \in \{1..N_S\}$ have states s_i and weights π_i and the position estimate is given by the biased estimator

$$\tilde{s} = \frac{\sum_{i=1}^{N_S} s_i \pi_i}{\sum_{i=1}^{N_S} \pi_i}. \tag{2.2}$$

Test environment is a one dimensional linear motion in a potentially d-dimensional state space. The fixed motion has velocity v in k_{max} steps indexed by $k \in 0..k_{max}$, such that ground truth position $s_k = kv$. In the initial step at $k = 0$ the state distribution of the tracker is set to a Dirac δ-peak at the position of the tracked object $s_0 = 0$. The observation process is given by a likelihood function $\mathcal{L}(s, s_k, \sigma_A)$, where σ_A^2 is related to the variance of a peak around ground truth position s_k. An experiment is defined

Result: Position estimates $\tilde{s_k}$ for all k_{max} steps.
Set $s_0 = 0$
Set all N_S particles to $(s_i^0, w_i^0)_{i=1..N_S} = (s_0, 1/N_S)$
for $k = 1..k_{max}$ **do**
  Set ground truth $s_k = kv$
  Resample s_i^{k*} from $\{(s_i^{k-1}, \pi_i^{k-1})\}$
  Estimate $s_i^k = \eta$ with $\eta \propto \mathcal{N}(s_i^{k*}; 0, \sigma_E^2)$
  Assign weights via likelihood $\pi_i = \mathcal{L}(s_i^k, s_k, \sigma_A)$
  Compute position estimate $\tilde{s_k}$ from (Eq. 2.2)

Algorithm 2.1: Simulation of Artificial Tracking Experiment

by setting five parameters. The tracking algorithms parameters are given by estimation variance σ_E^2 and sample count N_S. Experiment's duration is given by the number of steps k_{max}. The object to be followed is expressed by its linear velocity v and its appearance width σ_A in a likelihood function. Two distinct likelihood functions are used for different purposes.

In order to allow for analytic expressions of the non-sampled tracking process (*Fully Gaussian Reference Model* (2.1.2)), a Gaussian likelihood function $\mathcal{L}(s_k, \sigma_A) \propto \mathcal{N}(x; s_k, \sigma_A^2)$ is applied. It yields non-zero values in the whole state space, which allow the tracker to converge to the true position from any current state. This function is used for the reference situation in which the tracking process is analytically tractable (sec. *Steady State* (2.1.2)).

If the likelihood is the direct result of a measurement process, it is commonly neither Gaussian shaped nor normalized. Additionally, the filter result may not even allow convergence from any point in state space, when multiple maxima exist. Such a function has a limited range in which the tracking distribution converges towards the optimum. In order to simulate such a behavior, besides the default Gaussian likelihood, a truncated Gaussian likelihood is used[2].

[2] A similar model can for instance be found in [KA01].

The simulation of a likelihood function with a limited non-zero range is established by approximating the Gaussian as its second order Taylor expansion, which is set to zero for negative values. This will allow the computation of a clear threshold for loss of track in the Particle Filter simulations. The truncated Gaussian observation likelihood function is

$$\mathcal{L}_{trunc}(s; s_k, \sigma_A^2) = \begin{cases} 1 - \left(\frac{(s-s_k)^2}{2\sigma_A^2}\right), & \text{if } |s| < \sqrt{2}\sigma_A \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

The function 2.3 reaches zero at $(s - s_k)^2 = 2\sigma_A^2$. The related experiment is depicted in Figure 2.1.

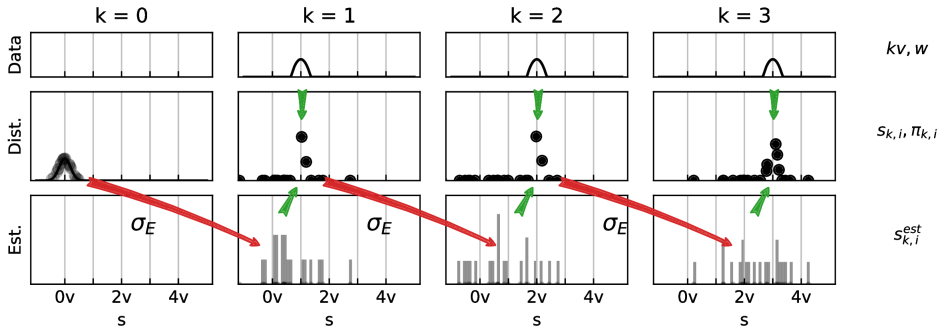


Figure 2.1.: Example: First steps of the experiment with $N_S = 20$ samples and target velocity $v = 4$. Prediction (red arrow) by Gaussian of $\sigma_E = 4$ transforms the weighted sample set (center row) into the unweighted set (bottom row). The update step (green arrows) joins predicted set and a truncated Gaussian likelihood function of $\sigma_A = 1$ (top row) into the posterior (center row). The initial distribution (at $k = 0$) is sampled from a Gaussian of $\sigma = 0.8$ (center, left). The right hand side shows the variables involved.

Being the result of for instance a measurement process, the likelihood is subjected to noise and artifacts. Noise within the likelihood function however, may have an influence

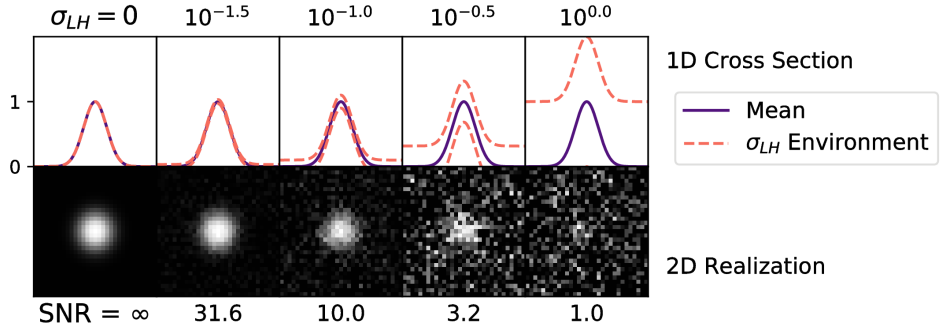


Figure 2.2.: Two-dimensional Gaussian likelihood function with noise.

on the tracking result. One type of noise is investigated (Fig. 2.2): Gaussian noise is added to the measurement function on evaluation $\mathcal{L}_1 = \mathcal{L}_0 + \eta$ with $\eta \propto \mathcal{N}(0; 0, \sigma_{LH}^2)$ and negative results restricted to zero. The signal to noise ratio of the result is then estimated as SNR $= I_0/\sigma_{LH}$ where I_0 denotes peak likelihood. In order to keep the number of parameters low, the noise model is implemented free of temporal and spatial correlation. Each time the function is evaluated, a new random variable is drawn (Fig. 2.2).

2.1.2. Fully Gaussian Reference Model

Without investigating the Particle Filter model yet, it is possible to gain insight into the proposed tracking process by applying the single step with analytic descriptions. The random walk tracking model with three model parameters estimation variance σ_E^2, measurement variance σ_A^2, velocity v allows the computation of a steady state for tracking with ideal Gaussian distributions. The reference model will be used to compute stead states results. Related descriptions may be found in the literature of time-invariant Kalman Filters. A graphical illustration of the process is found in the supplementary material (page 196).

In this model, the Gaussian state density is updated by estimation and measurement step from prior target density described by mean μ_{k-1} and variance σ_{k-1}. The estimation process, given as a convolution with a diffusive Gaussian of mean 0 and variance σ_E^2, yields a new distribution with

$$\mu_{est} = \mu_{k-1} \qquad \sigma_{est}^2 = \sigma_{k-1}^2 + \sigma_E^2. \tag{2.4}$$

Application of the observation, by multiplication with a Gaussian of mean s_k, the true position and variance σ_A^2 yields the measurement density and thus the final density estimate for k:

$$\mu_k = \frac{\frac{\mu_{est}}{\sigma_{est}^2} + \frac{s_t}{\sigma_A^2}}{\frac{1}{\sigma_{est}^2} + \frac{1}{\sigma_A^2}} \qquad \sigma_k^2 = \frac{\sigma_{est}^2 \sigma_A^2}{\sigma_{est}^2 + \sigma_A^2} \tag{2.5}$$

Introducing equations 2.4, this yields

$$\mu_k = \frac{\mu_{k-1}\sigma_A^2 + s_k(\sigma_{k-1}^2 + \sigma_E^2)}{\sigma_{k-1}^2 + \sigma_E^2 + \sigma_A^2} \qquad \sigma_k^2 = \frac{(\sigma_{k-1}^2 + \sigma_E^2)\sigma_A^2}{\sigma_{k-1}^2 + \sigma_E^2 + \sigma_A^2}. \tag{2.6}$$

Steady State Iterated application of (Eq. 2.6) leads to a steady state in which the result follows the target with velocity v and a constant spatial lag l such that $s_{k-1} - \mu_{k-1} = l = s_k - \mu_k$, $\mu_k - \mu_{k-1} = v = s_{k-1} - s_k$ and $\sigma_k = \sigma_{steady} = \sigma_{k-1}$. Using these conditions in (Eq. 2.6) steady state variance σ_{steady}^2 and lag l_{steady} are found.

$$l_{steady} = v\left(\frac{\sigma_A^2}{\sigma_{steady}^2 + \sigma_E^2}\right) \qquad \sigma_{steady}^2 = \frac{\sigma_E^2}{2}\left(-1 + \sqrt{1 + 4\frac{\sigma_A^2}{\sigma_E^2}}\right) \tag{2.7}$$

Time evolution (Eq. 2.6) and steady state variance σ^2_{steady} are independent of velocity v. Variance σ^2_{steady} shows two different scaling regimes:

$$\lim_{\sigma_A/\sigma_E \to 0} \sigma^2_{steady} = \sigma^2_A \qquad \lim_{\sigma_A/\sigma_E \to \infty} \sigma^2_{steady} = \sigma_A \sigma_E \tag{2.8}$$

For small values ($\sigma_A \ll \sigma_E$) of observation variance σ^2_A, using the square root's (in Eq. 2.7) first order Taylor expansion leads to $\sigma^2_{steady} = \sigma^2_A$. For large $\sigma_A \gg \sigma_E$, on the other hand, the steady state is $\sigma_A \sigma_E$. The steady state value of the estimating distribution's variance is

$$\sigma^2_{steady,Est} = \sigma^2_{steady} + \sigma^2_E = \frac{\sigma^2_E}{2}\left(1 + \sqrt{1 + 4\frac{\sigma^2_A}{\sigma^2_E}}\right). \tag{2.9}$$

Analogous to (Eq. 2.8) the scaling for limit cases is

$$\lim_{\sigma_A/\sigma_E \to 0} \sigma^2_{steady,Est} = \sigma^2_E \qquad \lim_{\sigma_A/\sigma_E \to \infty} \sigma^2_{steady,Est} = \sigma_A \sigma_E. \tag{2.10}$$

Similarly to the variances, steady state lag scales differently in the two regimes. It exhibits a linear dependency to both the velocity v and the ratio $\sigma^2_A/(\sigma^2_{steady} + \sigma^2_E)$. The latter equals the ratio of observation variance to steady state estimation variance (Eq. 2.9). Therefore, steady state lag scales as

$$\lim_{\sigma_A/\sigma_E \to 0} l_{steady} = v\frac{\sigma^2_A}{\sigma^2_E} \qquad \lim_{\sigma_A/\sigma_E \to \infty} l_{steady} = v\frac{\sigma_A}{\sigma_E}. \tag{2.11}$$

For the proposed tracking application, the regime $\sigma_A < \sigma_E$ is of interest. Within that region, the estimating distribution is wider than the observation region and lag is comparably low. The presented equations yield analytic results for an idealized situation, to serve as a reference for the sampled model.

2.1.3. Sampling Stability

Depending on its parametrization, a Particle Filter may be able to produce proposal densities suitable for tracking the target object[3]. This requires particles sufficiently close to — within the range of — the likelihood maximum to be drawn. In this section, a model for simulating loss of track caused by a likelihood function of limited peak extent is proposed. Model to be approximated is the sampled, finite step tracking model in 2.1.1. A worst-case approximation to its basic step is used. This allows studying the effect of finite sampling. This section does not take noise into account.

[3] An example of a successful case is displayed in Figure 2.1, an unsuccessful case in Figure A.1 on page 196.

2.1.3.1. Critical Step Approximation

In each tracking step, particles of the filter need to provide a viable support, such that the important region of the likelihood function is sampled. For being able to follow an object, it may be sufficient[4] to have at least one particle in such an area of importance.

The proposed model assumes that there is a valid region of the likelihood function and an invalid one. The valid region — centered around the true object's position — is required to be sampled from. Samples drawn from the invalid region are considered lost. As a consequence of the existence of a valid region, prior samples may only exists therein. Task of the tracking algorithm in this setting is to spread at least one sample from the prior valid region to the current. This leads to the definition of the worst case, the critical step.

In the worst case tracking step, the prior is a δ distribution within the prior valid region, but farthest away from the current valid region.

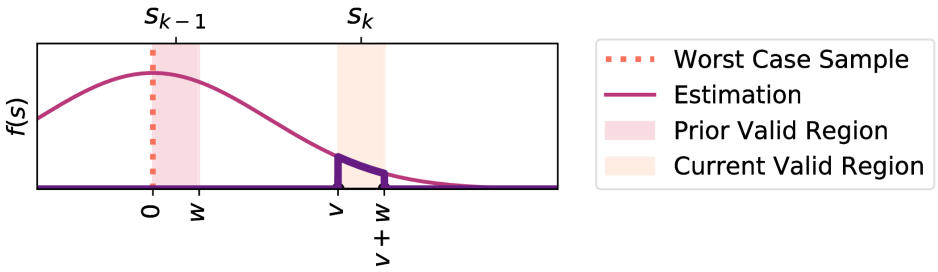


Figure 2.3.: Model of the critical step in the linear tracking problem with velocity v and valid observation region size w.

Prior and current regions of width w are centered around prior and current ground truth positions s_{k-1} and s_k, which are v units apart from each other (Fig. 2.3). The distribution in the prior valid region consists of samples which reached only the lagging side to the left and are therefore farthest away from the current valid region. The estimation process uses a Gaussian kernel to estimate new positions. Therefore the probability of reaching the current valid region is given as

$$p_S = \int_v^{v+w} \mathcal{N}(s; 0, \sigma_E^2)\, ds = \int_{v/\sigma_E}^{(v+w)/\sigma_E} \mathcal{N}(s; 0, 1^2)\, ds. \tag{2.12}$$

For a set of N_S independent samples, the probability of any sample reaching the valid region is

$$p_M = 1 - \overline{p_M} = 1 - (1 - p_S)^{N_S}. \tag{2.13}$$

[4]Given, the likelihood function is clearly peaked and this peak can be detected.

Due to resampling, in Particle Filter tracking, a single successful sample[5] is sufficient to branch off a new estimating distribution. Thus, if any sample reaches the valid region, the step was successful with respect to not losing track of the object. Result of k_{max} critical consecutive tracking steps is only successful, if all of them are.

$$p_{success} = (p_M)^{k_{max}} = \left(1 - (1 - p_S)^{N_S}\right)^{k_{max}} \tag{2.14}$$

The sequential occurrence of the critical tracking step leads to failure eventually. The expected number of steps until failure is the expectation value of a geometric distribution. The success probability decreases exponentially with the number of steps and with all steps being independent of each other the process is memoryless. This is the case in random resampling.

In an actual application, the state distribution may recover to a less critical one, which is not δ-shaped, but spread over the valid observation range. Additionally, it may be possible to reacquire the target after a single failed step. These effects are not modeled for reasons of simplicity.

In order to determine a useful parametrization a criterion for the choice of the success probability of the full experiment needs to be set. If one is interested in a distinct probability q for a series of k_{max} successful steps, the multi-sample success probability must be chosen such that

$$p_M|_q = q^{1/k_{max}}. \qquad \text{Quantile Condition} \tag{2.15}$$

2.1.3.2. Approximative Analytic Solutions

In order to parametrize the tracking process, trade-offs between different variables are possible. Generally, the model is invariant to changes where both w and v increase linearly with σ_E. The relation of v and w to σ_E suggest to replace both v and w by their normalized forms v/σ_E and w/σ_E. In this section, variables denoted with a prime[6] are normalized by σ_E, e.g. $w' = w/\sigma_E$.

More specific details are obtained from the expressions for single and multi-sample success probabilities. For $w \ll \sigma_E$, single-sample success (Eq. 2.12) is approximated as

$$p_S|_{w \ll \sigma_E} \approx w\mathcal{N}(v; 0, \sigma_E^2) = \frac{w}{\sigma_E}\mathcal{N}\left(\frac{v}{\sigma_E}; 0, 1\right) = w'\mathcal{N}\left(v'; 0, 1\right). \tag{2.16}$$

Furthermore, demanding $p_M \overset{!}{=}$ const. and taking the logarithm of Eq. 2.13 leads to $N_S \ln(1 - p_S) = \ln(1 - p_M) \overset{!}{=}$ const. In the interesting case, when p_S is small, the logarithm can be approximated by its first order Taylor expansion $p_S N_S = c_0$. Without further approximation, the direct expression for c_0 is

$$c_0 = -\ln(1 - p_M) = -\ln(1 - \sqrt[k_{max}]{q}). \tag{2.17}$$

[5] Provided that unsuccessful samples obtain negligible weights, as assummed here.

[6] For the sake of readability, not every expression of type $\cdot/\sigma_E$ is replaced by the primed one.

Entering Eq. 2.16 yields $\frac{w}{\sigma_E}\mathcal{N}(v/\sigma_E; 0, 1^2)N_S = c_0$. Applying the logarithm yields the first order approximation to the Critical Step Model.

$$\left(\frac{v}{\sigma_E}\right)^2 = 2\ln\left(\frac{w}{\sqrt{2\pi}\sigma_E}\frac{N_s}{c_0}\right) \qquad 1^{\text{st}} \text{ Order, 1-dimensional} \tag{2.18}$$

This equation describes the relationship of central components for $w \ll \sigma_E$.

Success Probabilities For the purpose of validating the models with respect to success probability, full expressions as in Eq. 2.14 are denoted. For the default model this is

$$p_{success} = \left(1 - \left(1 - \int_{v/\sigma_E}^{(v+w)/\sigma_E} \mathcal{N}(s; 0, 1^2)\, ds\right)^{N_S}\right)^{k_{max}}. \tag{2.19}$$

For the first order approximation (Eq. 2.18), the multi-sample probability in the inner brackets is approximated and therefore

$$p_{success}^{1st} = \left(1 - e^{-wN_S\mathcal{N}(v;0,\sigma_E^2)}\right)^{k_{max}}. \tag{2.20}$$

These expressions yield the success probability for a given parametrization and will be used for verification.

Velocity Correction Due to the pessimistic assumption in each step, the Critical Step Model yields lower bounds for the performance. These may be corrected replacing the minimum velocity by an expectation. Instead of the left boundary of the valid region, the unweighted[7], expected sample location within the valid region is used. The uncorrected v is corrected to v^*.

$$v^* = \frac{\int_v^{v+w} s\mathcal{N}(v; 0, \sigma_E^2)\, ds}{\int_v^{v+w} \mathcal{N}(v; 0, \sigma_E^2)\, ds}. \tag{2.21}$$

Multiple Dimensions In a multi-dimensional state space the following assumption is used to derive a success probability: The target is to be followed using an isotropic multivariate Gaussian distribution and localized by an isotropic likelihood function. As the number of dimensions of the state space is increased, motion remains one dimensional. The object needs to be found along the dimension of movement (section 2.1.3.1) while still being detected (in an unchanged location) in the remaining $d-1$ dimensions. The additional dimensions only affect the single-sample probability. Therefore, Eq. 2.12 becomes

$$p_S = \int_v^{v+w} \mathcal{N}(s; 0, \sigma_E^2)\, ds \left(\int_{-w/2}^{w/2} \mathcal{N}(s; 0, \sigma_E^2)\, ds\right)^{d-1}. \tag{2.22}$$

[7] A particular shape of a likelihood peak is ignored.

A successful single sample travels v units in the direction of motion and remains close to the object in all others.

In this formulation, dimensions are evaluated independently from each other. Therefore it is only an approximation to the situation of an isotropic likelihood[8]. Similar to (Eq. 2.18), the terms of the $d-1$ non-motion dimensions can be approximated as

$$B = \lim_{w\to 0} \int_{-w/2}^{w/2} \mathcal{N}(s; 0, \sigma_E^2)\, ds = \frac{w}{\sqrt{2\pi}\sigma_E}. \tag{2.23}$$

The additional dimensions act as a bias B on the single sample success probability and only justified for small w/σ_E. Expression B also exists in the integral of 1st dimension, the dimension of motion. The additional $d-1$ dimensions generalize (Eq. 2.18) to

$$\left(\frac{v}{\sigma_E}\right)^2 = 2\ln\left[\left(\frac{w}{\sqrt{2\pi}\sigma_E}\right)^d \frac{N_S}{c_0}\right] \qquad 1^{\text{st}} \text{ Order, } d\text{-dimensional.} \tag{2.24}$$

Spatial measures such as velocity v and appearance width w emerge as their relative versions v/σ_E and w/σ_E, respectively. Equation 2.24 only offers a real-valued solution if

$$\left(\frac{w}{\sqrt{2\pi}\sigma_E}\right)^d \frac{N_S}{c_0} \geq 1. \tag{2.25}$$

Optimum Parametrization In order to study the relationship of parameters, it is convenient to center the equations around the estimation kernel's width σ_E and consider v and w as multiples thereof. In a tracking experiment, σ_E may not be a given measure, and rather act as a tuning parameter. Using w as the fundamental element of the parametrization process, σ_E is then used to optimize the search window and gain maximum velocity v/w. Note that this is an optimization with respect to the computational effort required for being able to follow the object. It is not an optimization of result error. The ratio v/w is investigated for the 1st order approximation (Eq. 2.24). Rescaling the equality, such that the left hand side's v/σ_E is transformed into v/w leads to $\nu^2 = 2\ln\left(Aw'^d\right)/w'^2$, with $w' = w/\sigma_E$ and $\nu = v/w$. Term $A = N_s/((2\pi)^{d/2} c_O)$ is always greater than zero for valid parameters. An extremum is found by setting $\partial v'^2/\partial w' = 2(d - 2\ln Aw'^d)/w'^3 \stackrel{!}{=} 0$, located at $w' = \sqrt{e}/A^{-1/d}$. Since $\partial^2\nu/\partial^2 w'|_{w'=\sqrt{e}/A^{-1/d}} \leq 0$ this only extremum is a maximum. Therefore, in order to maximize $\nu = v/w$, parameter should be chosen such that

$$\frac{v}{w}\Big|_{opt} = \frac{\sqrt{d}}{\sqrt{2\pi e}} \left(\frac{N_S}{c_0}\right)^{1/d} \tag{2.26}$$

$$\frac{\sigma_E}{w}\Big|_{opt} = \frac{1}{\sqrt{2\pi e}} \left(\frac{N_S}{c_0}\right)^{1/d}. \tag{2.27}$$

[8] An isotropic likelihood may be integrated into the term for the $d-1$ dimensions, but yields a more complicated multi-dimensional integral. Instead, (Eq. 2.22) is used as approximation.

Dividing equations 2.26 and 2.27 by another, the maximal velocity in units of σ_E in d dimensions is $\frac{v}{\sigma_E}|_{opt} = \sqrt{d}$. A brief overview of these relations is shown in Figure 2.4 in the view with respect to σ_E (left) and to w (right).

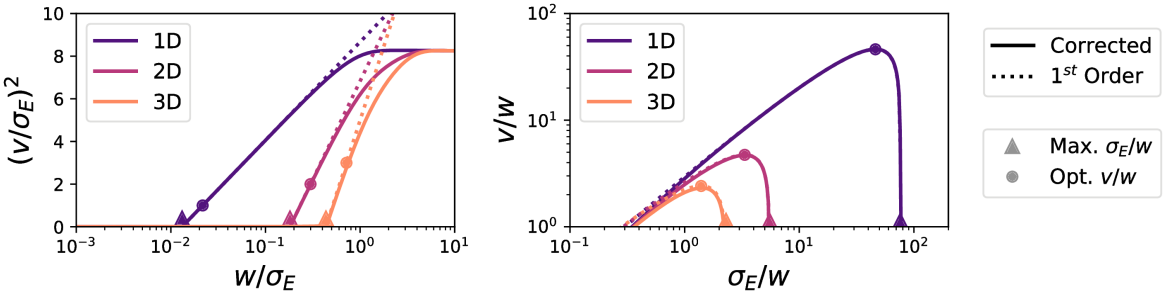


Figure 2.4.: Overview of extremal and optimal parametrization as suggested by Eq. 2.25, Eq. 2.26, Eq. 2.27. View centered around σ_E (left) or w (right). The right hand side clearly shows the optimum parametrization for maximum v/w.

2.1.4. Residual Errors

The position output of the tracking algorithm is obtained using the weighted mean biased estimator (Eq. 2.2). When sampling from a Gaussian distribution in order to estimate the mean value, the result holds a standard error. Its level depends on the Gaussian's variance as well as the number of samples. The standard error of sampling from Gaussian of variance σ^2 is $\sigma/\sqrt{N}$ with N being the number of samples. In the proposed tracking algorithm, samples to estimate the observation Gaussian are not drawn from the observation function, but the proposal distribution instead. Furthermore, they are weighted. Thus, the plain computation of the standard error to the observation Gaussian does not lead to the desired result. This section motivates error measure for different regimes of σ_A/σ_E, assuming a Gaussian likelihood.

Motion-free case While N_S samples are present, due to their weights, not all of them contribute to the position estimate in equal parts. Behavior of the residual error may be described using an effectively available sample ratio r. At $v = 0$, lag may be neglected and proposal distribution and likelihood peak are considered aligned. The ratio of effectively contributing samples depends on the ratio of σ_A/σ_E. For $\sigma_A > \sigma_E$ it is $r(\sigma_A/\sigma_E)|_{\sigma_A>\sigma_E} = 1$. In the opposite case $\sigma_A < \sigma_E$, samples from the wide estimating distribution may be drawn on the tails of the observation and, as a result of their low weight, not contribute much to the estimate. In this case, the effectively available number of samples is estimated as $r(\sigma_A/\sigma_E)|_{\sigma_A<\sigma_E} = \sigma_A/\sigma_E$. Hence, the related standard

error is

$$\epsilon_{res} \propto \frac{\sigma_A}{\sqrt{N_S r(\sigma_A/\sigma_E)}}. \tag{2.28}$$

Multiplying equation 2.28 by $1/\sigma_E$, the influence of σ_E can be eliminated. With normalized error measure $\epsilon'_{res} = \epsilon_{res}/\sigma_E$ and the normalized observation width $\sigma'_A = \sigma_A/\sigma_E$ this yields

$$\epsilon'_{res} \propto \frac{\sigma'_A}{\sqrt{N_S r(\sigma'_A)}}. \tag{2.29}$$

This formula separates the two regimes of inverse relationships of σ_E and σ_A. For the use in a tracking application, the regime $\sigma_A < \sigma_E$ is to be considered, which exhibits another transition: Once the effectively used number of samples becomes too low, the residual error cannot be decreased further. Considering $N_S \frac{\sigma_A}{\sigma_E} = N_{min}$ fulfilled, error (Eq. 2.28) becomes

$$\epsilon_{res} \propto \sigma_E \frac{N_{min}}{N_S}. \tag{2.30}$$

The level of residual error for an effective sample size equal to N_{min}, is independent of σ_A and the residual error actually dominated by σ_E instead. The three regions of scaling are depicted in figure 2.5, along with the cases of motion introduced.

Not only may the samples be limited, the biased estimator also uses their weights to compute the result. This means that the biased estimate is actually closer to the correct position than estimated here. While the missing factor will be explored via fitting, it may be estimated in the following way: Using a biased estimator, which samples from a Gaussian and weighs the samples by the same Gaussian is identical to sampling from the product of the Gaussian with itself. Thus, in this particular case, the variance estimate would be reduced by factor 0.5.

In multiple dimensions, state space is considered to be sampled by an isotropic, multivariate Gaussian with covariance $\sigma_E^2 I$. The sample depletion term has to be used in powers of d and the Euclidean distance error in d dimensions obtains an additional factor of $\sqrt{d}$. The d-dimensional version of (Eq. 2.29) reads

$$\epsilon'_{res} \propto \frac{\sigma'_A \sqrt{d}}{\sqrt{N_S r^d(\sigma'_A)}}. \tag{2.31}$$

In analogy to the derivation of (Eq. 2.30), level of sample depletion in d dimensions is obtained by setting $N_S r^d(\sigma'_A) = N_{min}$:

$$\epsilon_{res} \propto \sigma_E \sqrt[d]{\frac{N_{min}}{N_S}} \tag{2.32}$$

Motion In the case of motion, the tracking distribution reaches a steady state for a fixed velocity v. Residual error is composed of the established lag (Eq. 2.7) as well as the standard error of the mean of the tracking distribution.

$$\epsilon = \sqrt{l^2 + \sigma^2/N} \tag{2.33}$$

While for large N_S the term of the observation uncertainty vanishes, lag does not. A depiction of this error measures is found in the right figure 2.5. The gray line displays the cases for $v = 0$, whereas red lines are generated for $v = 0.01...1$ with logarithmic spacing. Lag mostly dominates regions where σ_A becomes close to or larger than σ_E.

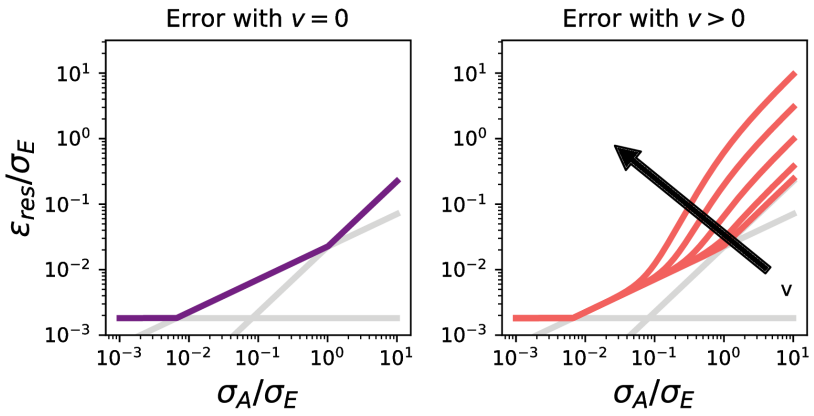


Figure 2.5.: Overview over residual errors in 1D. Left: Motion free case. Right: Samples of cases with motion $v > 0$. ($N_S = 1000$, Grey lines indicate proportionalities of the different regions.)

2.1.5. Noise

In previous section, noise has been neglected ($\sigma_{LH} = 0$). A noisy measurement introduces a random variable η into the measurement process. This variable is uncorrelated with the unperturbed likelihood function. It will decrease the quality thereof, since samples in the noise region receive a non-zero average weight. Assignment of higher weights in formerly unlikely regions affects both the computation of position estimates as well as the quality of proposal samples after resampling. This section suggest explanations for some results which will be obtained when noise is involved. Two models, for estimating the residual error in cases $v = 0$ and $v > 0$, are proposed. A noise induced regime transition, observed in both, is motivated. Common notion is that samples can be associated with discrete states: a sample may belong to the *correctly working* or *a or the noise state*, respectively. All models describe time evolution of discrete state probabilities and associate variance estimates to final probability distributions.

2.1.5.1. Multi State Noise Model

Aim of this model is the explanation of residual errors observed at $v = 0$. The initial set of samples is considered to be centered in the correct position of the object. This implies that all accumulated distance directly contribute to the observed error measure.

In each step, a sample may be dispersed either onto the true object's likelihood peak or the surrounding noise region. Samples in the object area are considered in the correct, *the ground state*, whereas those in the noise region are in *a noise state*. Samples in a noise state have a finite survival probability which is proportionate to the mean level α of noise. They may remain noise bound for several steps. The state of a sample indicates how many steps it already spent in noise, without returning to the ground state. States are indexed $j = 0...\infty$. Index $j = 0$ describes ground state, in which a sample is centered at the likelihood function's peak area. With each step that a sample spends in the noise region, it is dispersed by estimation with variance σ_E^2. Thus, state j corresponds to a sample drawn from a distribution of variance $\sigma_j^2 = \sigma_A^2 + j\sigma_E^2$. Instead of a concrete ensemble of N_S samples, the state density $\{\rho_j^k\}$ is simulated. It is initialized as $\rho_j^0 = \delta_{j,1}$ and updated $k \Rightarrow k+1$ for k_{max} steps via the successive application of

$$\rho_1' = \sum_{j=1}^{\infty} \rho_j^k \gamma_j \qquad \rho_j' = \rho_{j-1}^k (1-\gamma_{j-1}) \quad \forall j > 1 \qquad \text{Transition} \tag{2.34}$$

$$\rho_j^{k+1} = \frac{\rho_j' w_j}{\sum_{l=1}^{\infty} \rho_l' w_l}. \qquad \text{Resampling} \tag{2.35}$$

A sample may undergo either of the two transitions in (Eq. 2.34). Reflection to the ground state s_0 happens, when a sample from state j is drawn into the correct object's location with probability γ_j. Otherwise, a sample advances to the next state. In one dimension, the reflection probability can be estimated as $\gamma_j = \sigma_A/\sigma_j$. After transition, resampling (Eq. 2.35) occurs with weights chosen $w_0 = 1$ as the signal and for $j > 0$ with $w_j = \alpha$ proportionate to the noise reference level.

This model does not guarantee a steady state. If mean noise level α is sufficiently low, a steady-state may be reached, but otherwise the distribution will diverge. After k_{max} steps, simulation is stopped and the present distribution $\{\rho_j\} = \{\rho_j^{k_{max}}\}$ (Fig. 2.6) is used to compute the expected error level: $N_S\rho_j$ samples are present in each state, of which in the estimation process only fractions γ_j and $1-\gamma_j$ reach either correct peak or noise area, respectively. Correct samples correspond to weight 1 and variance σ_A^2, whereas noisy ones have weight α and variance σ_j^2. Final resulting variance is estimated as

$$\hat{\sigma}^2 = \frac{\sum_j N_S\rho_j[\gamma_j\sigma_A^2 + (1-\gamma_j)\alpha^2(\sigma_A^2 + j\sigma_E^2)]}{(\sum_j N_S\rho_j[\gamma_j + (1-\gamma_j)\alpha])^2} \tag{2.36}$$

$$= \frac{1}{N_S}\frac{\sum_i s_i[\gamma_j\sigma_A^2 + (1-\gamma_j)\alpha^2(\sigma_A^2 + j\sigma_E^2)]}{(\sum_j s_j[\gamma_j + (1-\gamma_j)\alpha])^2} \tag{2.37}$$

A depiction of the simulation of this model is seen in Figure 2.7.

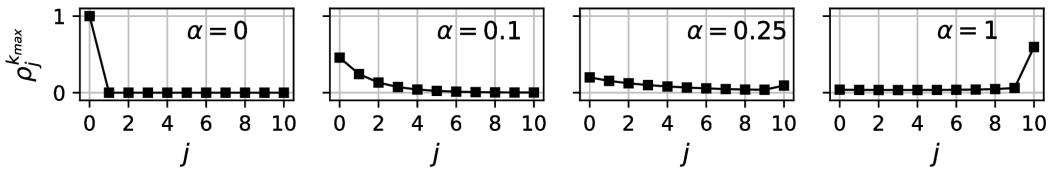


Figure 2.6.: Multi State Noise Model:Samples of simulated distributions for $\sigma_A = 0.1$, $\sigma_E = 1$ and $k_{max} = 10$ steps.

2.1.5.2. Two State Noise Model

The previous model does not reveal an analytic steady state. The two-state model simplifies the situation, in order to admit a steady state. This is then used to formulate a noise related sample depletion for the Critical Step Model. Other than the previous model, this one will consider the case $v > 0$.

Dropping the differentiation of noise states, only **c**orrect ground and **n**oise states with densities $\rho_C^k + \rho_N^k = 1$ remain. With motion, the only transition from noise to ground state considered is the resampling process. Accidental, correct estimation from noise to the target region is neglected.

$$\rho_{C,E}^k = \gamma \rho_C^k \qquad \text{Estimation + Observation} \tag{2.38}$$

$$\rho_C^{k+1} = \frac{\rho_{C,E}^k}{\rho_{C,E}^k + \alpha(1 - \rho_{C,E}^k)} \qquad \text{Resampling} \tag{2.39}$$

In estimation and observation step (Eq. 2.38), only a fraction $\rho_{C,E}$ with $0 \leq \gamma \leq 1$ of samples reaches the target position. Factor γ is identified as the single sample success probability p_S. Samples in noise are lost, unless regained in resampling (Eq. 2.39) with weights $w_C = 1$ and $w_N = \alpha$. Evolution of ρ_C^k is given by $\rho_C^{k+1} = \rho_C^k / [\rho_C^k (1 - \alpha) + \alpha/\gamma]$. Steady states after transition and resampling are

$$\rho_{C,E}^{steady} = \frac{\gamma - \alpha}{1 - \alpha} \qquad \rho_C^{steady} = \frac{1 - \alpha/\gamma}{1 - \alpha}, \tag{2.40}$$

respectively. Obviously, it is required that $\gamma \overset{!}{\geq} \alpha$, in order to not violate the normalization $0 \leq \rho_C^{steady} \leq 1$. With level $\alpha > 0$ of noise, the fraction of samples contributing to the tracking process in the expected manner is reduced to $\tilde{N_S} = N_S \rho_C^{steady}$. In the computation of the full success probability, this measure needs to be adapted accordingly. Since $\gamma = p_S$, the term $\tilde{N_S}$ introduces p_S into the exponent of the multi-sample success probability (Eq. 2.13). Deriving the 1st order approximation as before, this leads to c_0 being replaced by

$$c_0' = \alpha N_S + (1 - \alpha) c_0. \tag{2.41}$$

The approximation describes noise-induced reduction of available samples for the detection process. It does not take into account that noise also deteriorates the position estimate.

2.1.5.3. Transition point

Both models show an interesting transition between a stable and an unstable regime. The influence of α on this transition will be motivated by computing a variance estimate for the Two State Noise Model. Here, after estimation, $n_{C,E} = N_S \rho_{C,E}^{steady}$ correct and $n_{N,E} = N_S(1 - \rho_{C,E}^{steady})$ noise samples exist. Inside the noise area, samples are weighted by α and considered to have an arbitrary variance σ_E^2. Within range of the object, samples show variance σ_A^2 and are weighted by 1. Then, total variance of the two states is estimated as

$$\hat{\sigma}^2 = \frac{n_C^E \sigma_A^2 + n_N^E \alpha^2 \sigma_E^2}{(n_{C,E} + \alpha n_{N,E})^2} = \frac{\sigma_E^2}{N_S} \left[\frac{\alpha^2}{\gamma^2} + \frac{(\sigma_A^2/\sigma_E^2 - \alpha^2)(\gamma - \alpha)}{\gamma^2(1 - \alpha)} \right]. \tag{2.42}$$

In the 1D case of $v = 0$, $\gamma = \sigma_A/\sigma_E$ is identified[9]. Limit terms $\lim_{\gamma \to 0} \sigma^2 = \frac{\sigma_E^2}{N_S} \frac{\alpha^2}{\gamma^2}$ and $\lim_{\gamma \to \infty} \sigma^2 = \frac{\sigma_E^2}{N_S} \gamma$ intersect at

$$\gamma = \alpha^{2/3}. \qquad \text{Approximate Transition Point} \tag{2.43}$$

Together with error graphs of the Multi and Two State Noise Model, this is depicted in figure 2.7.

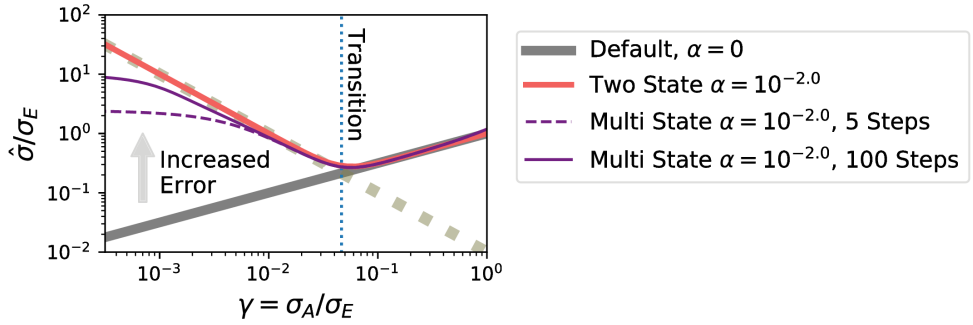


Figure 2.7.: Error levels estimated by Multi State Noise Model and Two State Noise Model: In the noise case, the error level of the Multi State Noise Model depends on the number of simulated steps, whereas the Two State Noise Model is independent thereof. The different scaling regimes of the error estimate have been motivated for the Two State Noise Model (Eq. 2.42) and lead to a noise dependent transition point (Eq. 2.43) which visually matches both models.

This relation approximates the transition between error estimates being dominated by incorrect samples in noise and those drawn in the correct object's location. While equation 2.40 suggest that full sample depletion $\rho_C^{steady} = 0$ occurs at $\gamma = \alpha$, sample depletion with respect to result stability appears at $\gamma = \alpha^{2/3}$ already (Fig. 2.8). This is interpreted as an increased effective level of noise ($\alpha \to \alpha^{2/3}$, $\alpha < 1$), which is present

[9]For $\alpha = 0$, residual error (Eq. 2.28) $\sigma^2 = \sigma_A^2/(N_S \sigma_A/\sigma_E)$ is re-obtained.

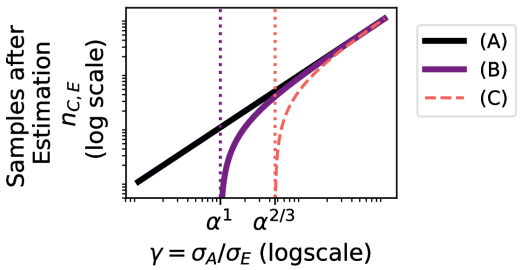
in Equation 2.40. Assuming that this assessment not only holds for the point of full depletion, then replacing α by $\alpha^{2/3}$ in Equation 2.40 yields a curve progression that not only models the final depletion, but also suggest a progression towards it (Fig. 2.8).

$$\rho_{C,E}^{steady} = \frac{\gamma - \alpha^{2/3}}{1 - \alpha^{2/3}} \tag{2.44}$$

Likewise, the sample depletion in the Critical Step Model can be estimated more pessimistically by replacing 2.41 by

$$c_0' = \alpha^{2/3} N_S + (1 - \alpha^{2/3}) c_0. \tag{2.45}$$

Using this correction term, the influence of noise can be estimated for cases where the velocity is greater than zero.



(A) $\alpha = 0$ in (Eq. 2.40)
(B) $\alpha > 0$ in (Eq. 2.40)
(C) $\alpha > 0$ in (Eq. 2.44)

Figure 2.8.: Sample depletion dependent on relative observation width σ_A/σ_E: (A) without noise in measurement. With with noise in measurement (B) full depletion is reached when $\gamma = \alpha$. Depletion with respect to result stability is reached at $\gamma = \alpha^{2/3}$. In this case the depletion curve (orange, dashed) is motivated by interpreting Equation 2.40 as an increased effective noise level (Eq. 2.44).

2.2. Computer Experiments

Experiments are conducted to validate the behavior of the proposed model or assess it, where yet unknown. This is achieved by simulating tracking scenarios that the theoretical model is intended to describe. Each consists of multiple independent simulations. In a single simulation, the target to be tracked is moved linearly without spatial and observation noise. In four major experiment groups track stability and residual errors are investigated.

Every experiment therein relies on the computation of error measures. For each temporal step of the experiment, the distance between the tracking result $\tilde{\vec{s}_k}$ and the ground truth $\vec{s}_k$ is computed as absolute Euclidean distance $d_k = ||\tilde{\vec{s}_k} - \vec{x}_k||_D$ in D dimensions. From these k_{max} measures such as maximum $D_{max} = \max_k d_k$ and root mean squared distance $D_{rms} = \sqrt{\sum_k d_k^2 / k_{max}}$ are derived. Measure D_{max} is strongly influenced by any

track failure and hence used to detect it. It describes the distance which was not tracked along after failure. Measure D_{rms} is used to validate the residual error in the successful regime.

2.2.1. Sampling Stability

For a fixed success probabiltiy, the Critical Step Model offers a boundary line in parameter space which separates the mostly successful from the mostly unsuccessful regime. This is to be fitted by data (Fig. 2.9). A single experiment is conducted in which an independent variable is altered (A). The results are then classified individually and the transition point along this variable is determined via logistic regression (B). This process is repeated altering a second independent variable (e.g. σ_A). This yields transition points for the first, dependent on the second independent variable. A model is then fit to the set of transition points (C).

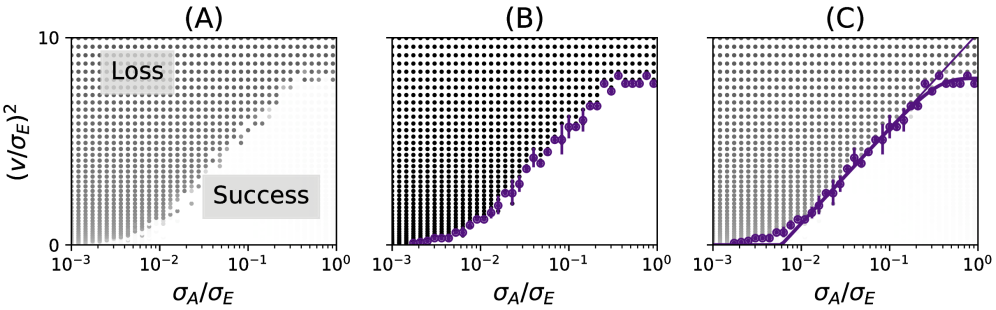


Figure 2.9.: Three steps of model fitting (see text).

Data Sampling Simulations are independent of each other and have a fixed success probability. In experiments multiple simulations are run with parameters within a specific range. Sampling single parameter sets only once in order to maximize the number of different sets will be referred to as *dense sampling*. For statistics, multiple simulations with same settings are required, which will be referred to as *repeated sampling*.

Classification Error in an simulation may have arisen from steady state lag, uncertainty in the observation or from track failure. Assuming a simulation, which does not allow for lag to build up, then only the error of the observation is allowed. Then, the upper limit to error measures is given by the observation's σ_A. A limit is chosen such that

$$S_{lagfree} = \begin{cases} 1 & \text{if } D_{max} < 3\sigma_A \\ 0 & \text{otherwise} \end{cases} \qquad \text{Lag Rejecting} \tag{2.46}$$

This thresholding procedure is applicable if a mostly lag free situation is expected. If, on the other hand, lag can not be ruled out, it is accepted to a fraction α_T of the total

distance in the following way.

$$S_{lag} = \begin{cases} 1 & \text{if } D_{max} < \max\left(3\sigma_A, \alpha_T k_{max} v\right) \\ 0 & \text{otherwise} \end{cases} \qquad \text{Lag Accepting} \qquad (2.47)$$

In practice, an algorithm which does not accumulate lag is favored. However, since there still exists a difference between a complete loss of track and the buildup of a limited lag, both thresholding rules are of interest.

Transition Fitting In order to determine a regime transition, the set is fixed except for a single parameter (commonly v) which is scanned. The results are then classified using (Eq. 2.46) or (Eq. 2.47). Binarized this way, they are used as dependent variable in a logistic regression in order to estimate a transition (commonly ν) from stable to unstable regime. In further steps, the transition needs to be associated with a success probability $p_{success}$. This will be determined beforehand for the known parametrization ($w = \sqrt{8}\sigma_A$) in case of the truncated Gaussian model.

Numerical Approximation An analytic expression only exists for the 1st order case (Eq. 2.24). Threshold velocity for the non-approximated model is computed numerically by $v^* = \arg\min_v |p_{success}(v, w, \sigma_E, k_{max}) - q|$, where q is the target success probability. For the computation of the threshold value v^* in the proceeding step it is sufficient to sample the space of v densely and select the parameter yielding the minimum result. This estimate may further be refined using an optimization algorithm such as a Golden Section search [Kie53].

Model Fitting Fitting (see step (C) in figure 2.9) is based on results obtained for the one dimensional case. Fitting factors f and g scale the input $w/\sigma_E = f\sigma_A/\sigma_E$ and $\tilde{N_S} = gN_S$ to their effective values. Given a set of experimental threshold velocities ν_i for and parameter sets s_i with parameters indexed by i , the fitness of the parameter set is computed as $\xi(f, g) = \sum_i \kappa(s_i) \left[\nu_i - v^*(\hat{s_i})\right]^2 / \sum_i \kappa(s_i)$. with $\hat{s_i}$ being the re-scaled parameter sets. Not all sections of the data can be explained by all models. Depending on the parameters s_i, function $\kappa(s_i)$ either returns 1 or 0 if the set is included in the fit or not. Fitness function is optimized using a Nelder-Mead [NM65] optimization scheme.

2.2.2. Residual Error

The second section of experiments is concerned with residual errors. Simulations are carried out using dense sampling and the resulting mean error level D_{rms} is compared to theory or theoretical curves are fit against it. Experiments scan residual errors over σ_A on a logarithmic scale, for few fixed σ_E. Sample count N_S is altered multiple times in order to confirm scaling of standard errors. Step count k_{max} is chosen large enough to allow for the buildup of a steady state when $v > 0$.

2.2.3. Noise

Noise is introduced into the settings and its influence is compared to predictions given by models proposed in section *Noise* (2.1.5). The influence of noise on achievable velocity and error measures is examined. Furthermore the effect of using multiple iterations of tracking on the same image is analyzed. In the model derivation the average noise level alpha is motivated with respect to the average weight of the foreground peak. The foreground peaks intensity is estimated by integrating $\alpha_{ref} = \int_{x=-\sigma/2}^{\sigma/2} \mathcal{N}(s; 0, \sigma^2)\, ds \approx 0.3829$. In an experiment, α is therefore obtained by computing the pure noise level of the Gaussian noise model $\alpha_0(\sigma_{LH}) = 0.5 \int_{x=0}^{\infty} \mathcal{N}(x; 0, \sigma_{LH}^2)\, dx$ and normalizing it to $\alpha(\sigma_{LH}) = \alpha_0(\sigma_{LH})/\alpha_{ref}$.

2.2.4. Implementation

The algorithm is implemented in *C++* and controlled via a *MeVisLab (MeVis Medical Solutions, Germany)* module using *Python* scripts. Code was executed on a standard PC with an *Intel Core i7* processor 3.4GHz and 32GB of RAM on *Windows 7 64bit* in a single thread.

2.3. Results

Along the lines of the described experiments, the following results are obtained.

2.3.1. Sampling Stability

Classification Decision Criterion Before investigating the effects in actual thresholded data, the difference in the two classification rules is investigated. The two rules — either rejecting or accepting lag — yield different results in classifying success and failure. In this section their different behavior is briefly investigated to distinguish similarities and differences in their functioning. Figure 2.10 shows a typical parameter scan, in which the

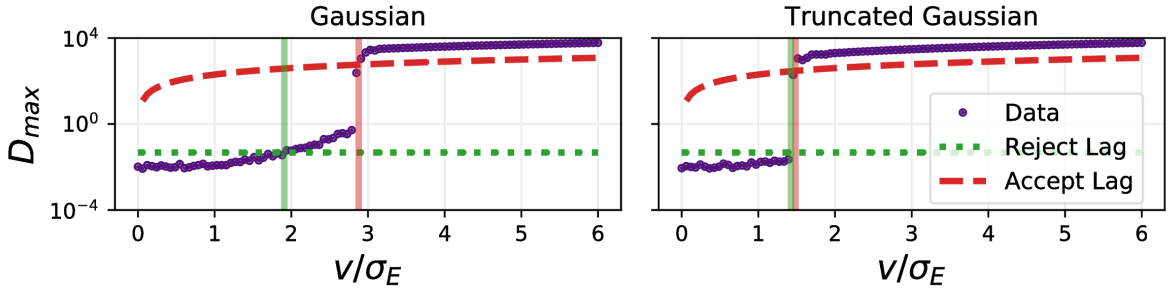


Figure 2.10.: Lag accepting (Eq. 2.46) and lag rejecting (Eq. 2.47) thresholds applied to data (dashed, dotted resp.).

maximum error D_{max} is computed for experiments scanned over v, for a fixed σ_E. Result curves show the characteristic transition from low-error phase to the failure phase, in which the error is close to the maximum distance. The threshold parametrization computed using (Eq. 2.47) with $\alpha_T = 0.2$ is depicted by a vertical dashed line.

The first thresholding at $D_{max} = 3\sigma_A$ is useful in cases where no lag can be established, or lag is negligible. This applies to the truncated Gaussian model (right) with σ_A not being very small. For the Gaussian model, condition (Eq. 2.46) underestimates the performance, as it would interpret errors arising by lag as failure. In the Gaussian case which leads to lag, classification rule (Eq. 2.47) leads to much better results in terms of detecting a clear threshold. The dashed line $\propto v$ for $\alpha_T = 0.2$ intersects the transition point from lag-phase to failure-phase well. Both threshold conditions yield interesting cases for the classification of experiments. While only the lag-free condition directly corresponds to the Critical Step Model, the lag accepting rule shows useful properties for finding thresholds including lag.

Threshold Velocity Fitting Fitting different models to the experimental data is undertaken in this section. In a first step, the success-probability which is observed at the classification threshold was determined by fitting the known model[10], yielding $p_{success} = 0.58$. This means that experiment in the *working* regime succeed to 58%. With the known value for $p_{success}$, other models could be fit for w and N_S (table 2.1). The table summarizes fitted $\tilde{N_S}/N_S$ and w/σ_A and cost f_{fit}. The cost f_{full} denotes the cost of the fitted function on the full dataset ($\nu_i = 1 \forall i$ in Eq. 2.2.1). A subset of the related data and corresponding fits are visualized in figure 2.11. Results for the model without velocity correction are not depicted, as they were always outperformed by the velocity corrected model: The velocity corrected model is able to describe the plateau region (2.11), whereas the non-corrected model is not.

The first order approximation does not explain the plateau region for large σ_A/σ_E: Any evaluation of the f_{full} for this model yields significantly higher results than the fitting cost obtained in the 1st order region ($f_{full} \gg f_{fit}$). The numerically approximated, velocity corrected model however is capable of fitting this non-linear section well (Fig. 2.11).

Without fitting N_S, the corrected model is capable of describing the plateau region when lag does not occur. When lag is present (Gaussian, Lag accepting rule), the threshold level can only be fit when N_S is used as fitting parameter. Fitting N_S also leads to slight increase for the selected $\tilde{N_S}/N_S \approx 1.2..1.3$, the plateau could already be matched without it. The cost for the full data set is slightly decreased in these cases.

For select values of σ_A/σ_E, data was sampled repeatedly (100 times) along the v/σ_E axis. This allowed for the computation of an empirical success probability, which when compared to the model's predictions, leads to the same result as presented. Furthermore it confirms the theory of the success probabilty computation, but for brevity only a figure in the supplementary materials is shown (Figure A.3 on page 197).

[10] 1st order, truncated Gaussian, lag rejecting rule ($w = \sqrt{8}\sigma_A$, $\tilde{N_S} = N_S$)

Model	Likelihood	Lag	$\tilde{N_S}/N_S$	w/σ_A	f_{fit}	(Comp.)	f_{full}
1st Order	Gaussian	Accept	(1)	76.92	0.26	$\ll$	20.35
		Reject	(1)	6.71	0.21	$\ll$	6.73
	Truncated	Accept	(1)	3.06	0.51	$\ll$	2.40
		Reject	(1)	$(\sqrt{8})$	0.27	$\ll$	1.36
Corr. - N_S	Gaussian	Accept	(1)	79.48	0.60	$\ll$	20.23
		Reject	(1)	6.86	0.24	$<$	0.54
	Truncated	Accept	(1)	3.09	0.49	$<$	0.71
		Reject	(1)	2.86	0.46	$\approx$	0.50
Corr. + N_S	Gaussian	Accept	4.39	17.04	0.55	$\approx$	0.57
		Reject	1.28	5.33	0.27	$\approx$	0.30
	Truncated	Accept	1.23	2.51	0.52	$\approx$	0.52
		Reject	1.27	2.23	0.36	$\approx$	0.37

Table 2.1.: Fitting results. 'Corr. $\pm N_S$' denotes fiting the velocity corrected model with and without allowing N_S to be fit. A large difference in f_{fit} and f_{full} indicates that the ramp section could be fitted, but the plateau was missed.

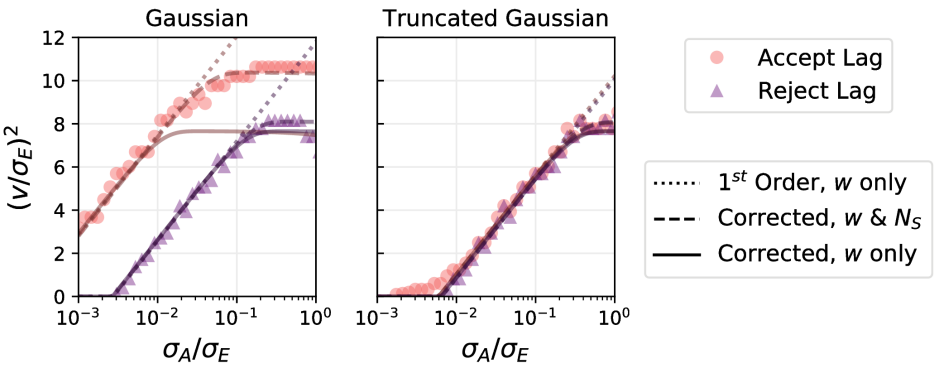


Figure 2.11.: Data thresholding and fitting results (table 2.1).

Multiple Dimension The 1D translation model is extended into a 2D and 3D model by adding more degrees of freedom to the state space. An experiment is conducted with isotropic likelihood functions for varying observation widths σ_A in order to assess sample depletion due to increasing number of independent dimensions as proposed by the bias terms.

Figure 2.12 depicts results obtained from simulations of the experiment in the top row. For various values of σ_A, the threshold velocity has been determined. The columns of the figure display results for the already known 1D and the extended 2D and 3D cases. The bottom row shows the same data, but in the view centered around σ_A. Simulations have been executed for two values of sample counts $N_S \in [128, 1024]$. Theoretical curves depicted are generated using the scaling factors presented in table 2.1, using the first order and the velocity corrected model.

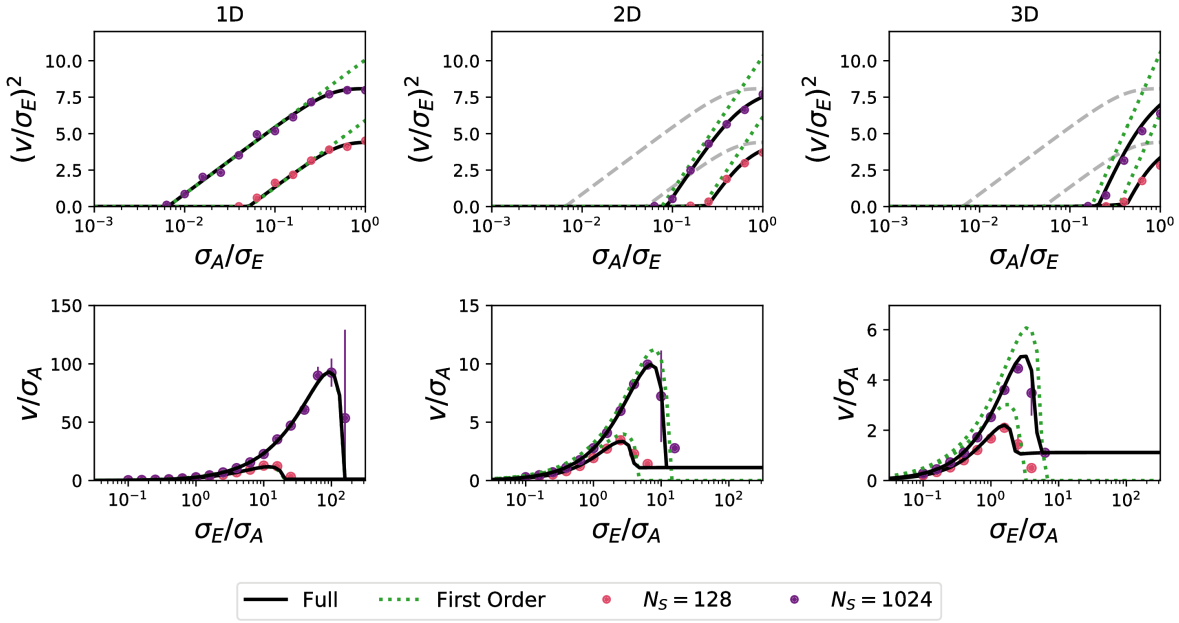


Figure 2.12.: Tracking experiment in a space of 1, 2, and 3 degrees of freedom (columns) in views focused around σ_E(top) and σ_A(bottom). (Settings: $k_{max} = 1000$, $\sigma_E = 1$)

The quality of the 1st order approximation decreases with the number of dimensions, as the interesting region is driven towards higher σ_A/σ_E. This has been discussed in the derivation, but is now also confirmed by the experimental data. The view centered on σ_A (bottom row), reveals, while the first order model determines the optimum (peak) position sufficiently well, it overestimates the performance v/σ_A. A drawback of the velocity correction is visible for large σ_E/σ_A: In this part an incorrect base level remains, which is not predicted by the 1st order model. The base level at $v = \sigma_A$ is an artefact of the velocity prediction.

Formula (Eq. 2.22) for d-dimensional case, is an approximation to the situation simulated in the sense that the simulation used a truly isotropic likelihood, while the

approximative equation treats dimensions fully independently. Despite this difference, the curves are in agreement with the experimental results. An overestimation of the results, which would be caused by the approximation might only be seen in the $2D$ and $3D$ Gaussian case for large σ_A/σ_E.

2.3.2. Residual Error

Residual Error without Motion The first set of experiments deals with the situation of $v = 0$. Figure 2.13 displays results for one (left) and three (right) dimensional state space. It is compared to error predicted by equations (Eq. 2.28) and (Eq. 2.30). Measures in this figure are normalized by σ_E.

The three motivated regimes of error scaling can be identified (A) $\sigma_A > \sigma_E$ or (B) and (C) $\sigma_A < \sigma_E$ with and without sufficient samples respectively. Scaling in regions (A) and (B), motivated to be $\approx \sqrt{2}$ was determined in the range 0.7 to 0.85 via fitting in four cases in 1D. The factor of region (C) was found to be 1.8 for the two likelihood functions.

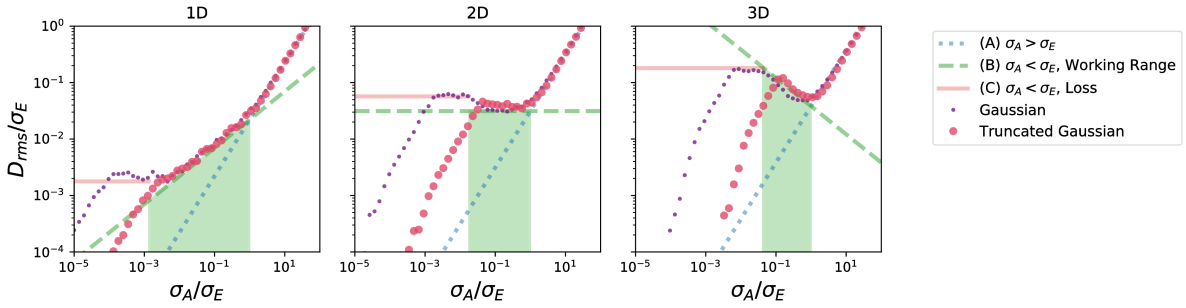


Figure 2.13.: Residual error D_{rms}/σ_E for $v = 0$ (Settings: $k_{max} = 1000$, $\sigma_E = \{0.1, 1, 10\}$, $N_S = 1024$)

Major difference between the Gaussian and the truncated Gaussian observation are observed in the area where the cut-off range of the truncated model takes effect. Similar to the truncated Gaussian model, the Gaussian model shows an error level falloff for very small σ_A/σ_E, which was not modeled. It is likely the result of computational imprecision.

Within the result data and model, the optimal parametrization for motion may be observed. For the sake of brevity, this is depicted in the supplementary material in Figure A.4. With the lag-rejecting rule, 'optimal' parametrization is found at the level of sample depletion, before regime (B) ramp meets (C). Accepting lag when it appears [11], regime (C) is integrated into the area of acceptable parametrization.

Center and right hand side shows the comparable $2D$ and $3D$ case. Due to the additional dimensions the scaling of errors (Eq. 2.31) (Eq. 2.32) with σ_A/σ_E differ and are matched by the experimental data. Regimes have not specifically been fit to the regions — instead the default values have been used. They do however fit the curve

[11] e.g. in the Gaussian case

characteristics well. This confirms both, treating the single dimensions independently as well as the suggested sample depletion process.

Residual Error with Motion In the second set, experiments are concerned with the additional effects introduced by motion with $v > 0$. Figure 2.14 depicts the scaling of residual errors with changes of v (left) and σ_A/σ_E (right).

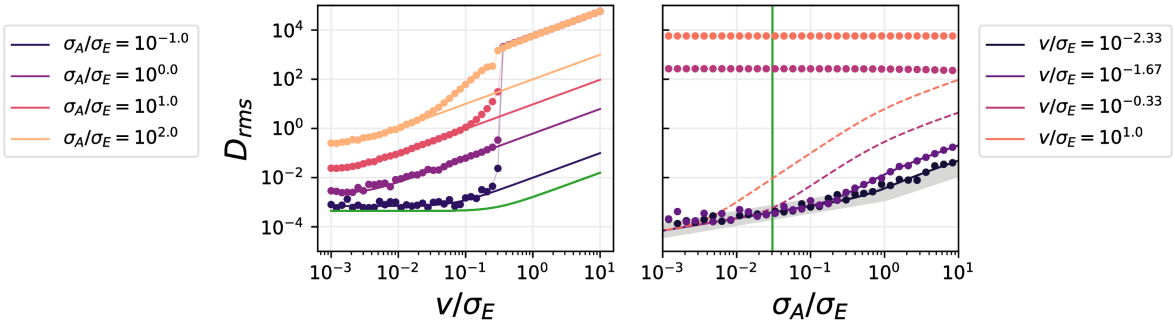


Figure 2.14.: Residual standard errors at $v > 0$. Dots: measured error; Lines: Residual standard error in steady state as proposed by equation 2.33 (right side: dashed, if expected in regime of track-loss); Residual standard errors at $v > 0$. (Settings: Gaussian model, $N_S = 1024$, $k_{max} = 1000$)

Left of 2.14 displays the obtained error plotted against velocity v, with v ranging from 0 up to values ($v/\sigma_e > 6$) where all depicted tracks fail. There are two regimes which are explained by equations in this figure. At v being close or equal to zero, the model $v = 0$ explains the error level. Continuing to higher values of v, the lag described in section 2.1.2 starts to dominate and the steady state 2.33 yields a useful approximation to the error. The region is best met for $\sigma_A = \sigma_E$. Higher and lower ratios produces higher error levels a the transition into the regime of error loss at $v/\sigma_E \approx 0.2$. In the last regime ($v/\sigma_E > 0.3$) the track cannot be maintained. Results depicted show the Gaussian likelihood case.

On the right hand side, the figure shows ϵ_{res} plotted against σ_A/σ_E for different v. Theoretically expected values are drawn by thick and dashed lines depending on whether track is expected to be successful or not. In the two cases when it is expected to fail, the residual error level is the track length, showing as horizontal line in the top. Again, major deviations arise for those parametrization which are still in the regime of successful outcome, but close to the threshold to track-loss.

Aside from the results, theoretical graphs and position of the optimum parametrization are drawn into the figure, showing that the optimum is found in a region where the least lag is to be expected.

2.3.3. Noise

Motion Free Case With additional noise, the residual error and its transition between stable and unstable regime are analyzed and compared to prediction of the *Multi State Noise Model* (2.1.5.1).

The introduction of noise leads to instabilities for narrow likelihood functions (Fig. 2.15). Left plot displays both experimental data (single dots) and predictions by the proposed Multi State Noise Model (filled lines) and Two State Noise Model (dotted lines). From

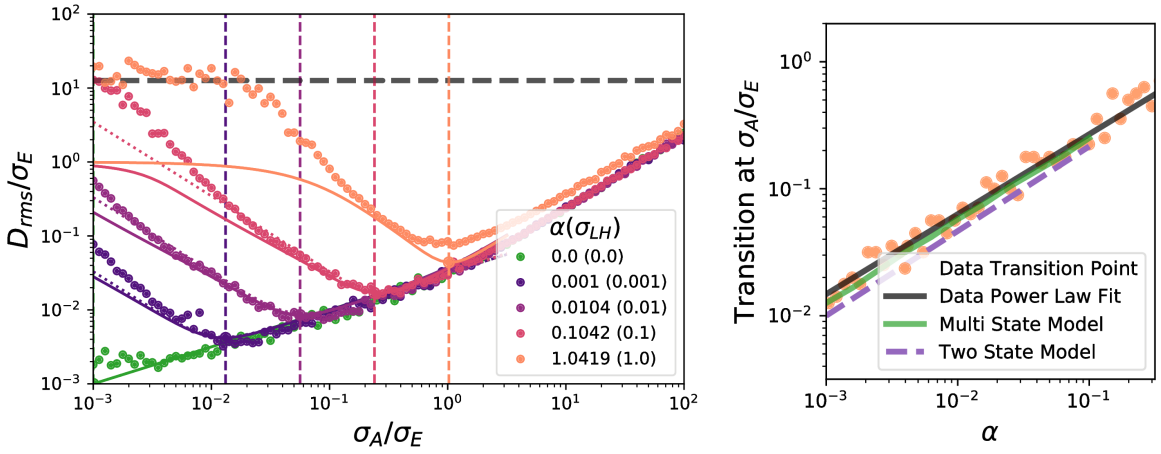


Figure 2.15.: Left: Residual errors obtained when noise is of different levels is introduced ($v = 0$). Dots: data. Filled lines: Multi State Noise Model, Dotted lines: Two State Noise Model. Dashed horizontal: Error level due to fully random walk of a single sample, dashed verticals: model minima (Settings: $N_S = 1000$, $k_{max} = 1000$, $\sigma_E = 1$, Gaussian, 10 averages). Right: Relationship of error minima to background noise level. (50 averages)

the reference case (green), an increase in noise level increases the residual error in the region of low σ_A/σ_E. For $\sigma_{LH} > 0$ transition points exists, where the result error detaches from the $\sigma_{LH} = 0$ reference. The level of the maximum obtained error (dashed horizontal line) is matched by the expected error of the random walk of a single particle. The level for this is computed as the mean absolute deviation of samples drawn from a zero mean Gaussian distribution of variance $k_{max}\,\sigma_E^2$, which following [Gea35] is $\sigma_E\sqrt{2k_{max}/\pi}$.

The comparing Multi-State model for $v = 0$ was iterated for $k_{max} = 1000$ steps and expected error (Eq. 2.36) computed. It shows a similar effect of creating transition points. Their locations (vertical dashed) visually match the transition points of the experimental data. The error level in the working regime is mostly correctly reproduced (except the case $\sigma_{LH} = 1$). However, the error in the regime of track loss is not matched. In particular, the maximum error observed in experiments is not matched by the model. The Two State Noise Model shows a very similar transition, but approximates the region

of track loss more strongly. It does not produce the observed plateau region for $\sigma_A/\sigma_E \to 0$.

The right plot of 2.15 shows the experimentally obtained transition points, more densely sampled for different noise levels σ_{LH} (dots). Transitions predicted by the model are overlaid as green dashed line. Fitting the experimental data by a power law yields $\sigma_A/\sigma_E \approx 1.14\alpha^{0.63}$ (black line). Fitting the model prediction in the same way yields $\sigma_A/\sigma_E \approx 1.12\alpha^{0.65}$ (green line). The transition of the Two State Noise Model is explicitly given by $\sigma_A/\sigma_E = \alpha^{2/3}$. It matches the experimentally observed transitions worse than the Multi State Noise Model.

Motion Case With motion, reachable maximum velocity is analyzed via thresholding. The lag-accepting case could not be fit well using (Eq. 2.40) as sample depletion term. The Two State Noise Model for $v > 0$ underestimates errors. Instead, thresholding was executed rejecting lag (Fig. 2.16). The expected threshold was computed using (Eq. 2.40), but replacing α by $\alpha^{2/3}$, such that sample depletion occurs at the approximate error transition (Eq. 2.43).

The curve shape in two and three dimensions is matched. With noise the performance

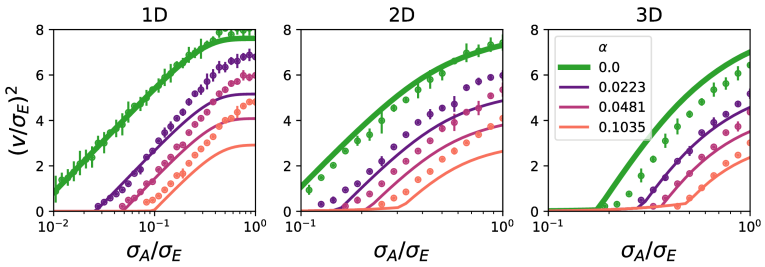


Figure 2.16.: 1D, 2D, 3D results for different levels of noise, classified without allowing lag. Results and velocity corrected model. ($N_S = 1000$, $k_{max} = 1000$, velocity corrected, N_S fitted, lag rejected)

is occasionally underestimated.

Multiple Iterations The effect of multiple iterations is examined by iterating single steps N_I times. Figure 2.17 (left) displays results for two experiments in which the product $N_S N_I$ is constant. Without noise ($\sigma_{LH} = 0$), and in the regime $\sigma_A < \sigma_E$, cases with the same $N_I N_S$ lead to identical results (not shown). Thus, model curves can be computed using $N'_S = N_I N_S$ as sample count. With noise $\sigma_{LH} > 0$, spreading samples into multiple steps yields a higher maximum velocity. For small $\sigma_{LH} = 0.001$ results can be approximated by using $\sigma'_E = \sqrt{N_I}\sigma_E$ and $N'_S = N_I N_S$. For large $\sigma_{LH} = 0.1$, this approximation underestimates the obtained results.

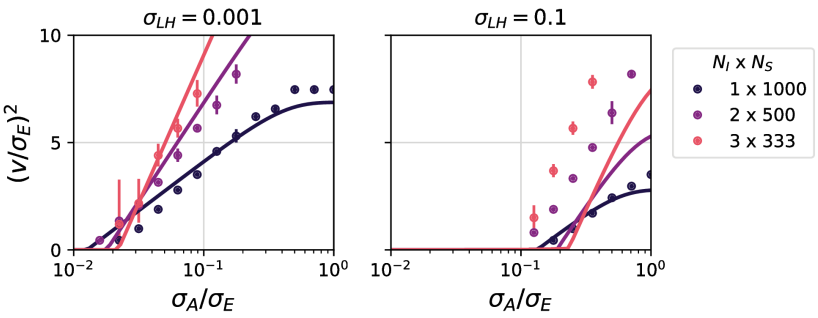


Figure 2.17.: Effect of spreading samples over N_I iterations, such that $N_S N_I = \text{const}$. Left: Low noise; Right: High noise.

2.4. Discussion

In this chapter, interdependence of parameters and expectable error measures in simple Particle Filter with a Gaussian random walk transition kernel were analyzed. In the noise free case, stability and residual errors could be modeled. With noise involved, the threshold between stable and unstable regime could be derived.

The discussion will cover the findings of this chapter and draw a conclusion, advocating a scheme for an application.

2.4.1. Findings

Three models, namely for the investigation of sampling stability, residual errors and track stability in random noise were proposed.

Sampling Stability Concerned with the availability of sufficient samples for object detection, the Critical Step Model (*Sampling Stability* (2.1.3)) uses worst case assumptions to formulate pessimistic success probabilities. Approximations allow for analytic expressions in extreme cases. The 1D model explains the interdependence of arising tracking parameters (*Threshold Velocity Fitting* (2.3.1)). Extensions to more than one-dimensional state spaces (*Multiple Dimension* (2.3.1)) are added in integral formulation and as first order analytic approximation. With increasing numbers of degrees of freedom, the first order approximation model increasingly overestimates the achievable maximum velocity. The model could be fit to experimental data yielding threshold functions between a mostly successful and mostly failing regime of parametrizations. Two likelihood functions were used to simulate tracking experiments — a Gaussian and a truncated Gaussian, which approximates the Gaussian up to second order only. The former was used to confirm predictions of a non-sampled fully Gaussian reference model, whereas the latter was required to confirm approximative predictions of the proposed model.

Residual Errors The chosen estimator is biased by the proposal distribution's spread, which it is not corrected for. While it is biased by the sample spread, it is computationally efficient. Using this estimator leads to specific residual errors, which were investigated. Three distinct scaling regimes could be observed for residual error, which differ in the amount of samples available in the ground truth target region. Predictions for these regions were confirmed in 1, 2 and 3 dimensions. Their values are related to the computation of the position by a biased estimator. Validating results could be obtained for cases without (*Residual Error without Motion* (2.3.2)) and with motion (*Residual Error with Motion* (2.3.2)).

Noise In the motion-free variant, a model for the explanation of the obtained error level was suggested by the Multi State Noise Model and Two State Noise Model. They reproduces the emergence of an instable regime, as observed in experimental data (*Motion Free Case* (2.3.3)). The scaling of transition points is very similar to the one observed in experiment and the error level in the stable regime is confirmed. As the Two State Noise Model only approximates the effects in noise, the Multi State Noise Model is better suited to fit the transition. The Two State Noise Model is not suited to explain error in the regime of track loss. The Multi State Noise Model is able to predict the behavior correctly. However, the maximum error predicted for the unstable regime was lower than the experimental value, due to neglecting sample degeneracy in the model. Assuming independence of samples, the model computes the variance as reduced by factor $1/N_S$, when all samples occupy the highest error mode. Nevertheless, the resampling process causes a degeneracy, which is further strengthened by uneven weightings due to noise. Therefore, the effective sample count is much lower in higher noise modes. This explains the experimental error at the level of a single-sample random walk.

Even though the model does not capture effects introduced by sample degeneracy, it is able to predict transition points between stable and unstable regime. It predicts the transition better than the Two State Noise Model, but the Two State Noise Model is allows for an analytic derivation of the transition. In both proposed models, single steps are no longer completely independent from each other, as in the noise free Critical Step Model. However, in the Two State Noise Model model it was possible to obtain a steady state result and treat single steps as if they were independent again.

The Two State Noise Model for $v > 0$ can be used as a motivation to explain the general drop in velocity reachable of the thresholded result data (*Motion Case* (2.3.3)). The replacement of α by $\alpha^{2/3}$ in the steady state solution yields roughly the correct threshold behavior by dragging the sample count to zero at the threshold to instability. The underestimation in for large σ_A/σ_E is likely an artifact of the data thresholding, rejecting the present lag at a particular cutoff. In two and three dimension the quality of the predicted values also does not perfectly match the data. Generally this approach preserves the general behavior. However, due to the derivation, it might lead to a loss of correct interpretation of success probability in c_O. With the presented experiment this could not be verified. The empirical success probability would have to be measured to fully confirm the suggested depletion term (Eq. 2.45).

In the literature of multi target tracking, often a Poisson distributed noise of false alarms is used [FBS80]. Such type of noise describes the faulty detection of incorrect target candidates in a preprocessing step — concrete realizations of localized measurements. The even noise level α, which is presented in this chapter, is a value which the measurement process senses even outside the actual target range. It describes an imperfect image filter, which after is not able to perfectly separate sought target from background. Thus, other than concrete, rare realizations of a false alarm, a general level of uncertainty is assumed. This might for instance be caused by strong noise in the input data (low SNR).

Multiple Iterations The choice of a few iterations improves performance, due to the estimating distribution spreading out, when the object is not found. For low noise the results can be explained by setting introducing altered values for σ_E and N_S into the single-step model (*Multiple Iterations* (2.3.3)). At high noise levels multiple iterations perform better than the altered model is able to predict. A reason for this must be sought in multi-step effect, which cannot be described by the proposed model. In any case, a lower bound can be estimated.

Finally, obtaining multiple iterations at the expense of a lowered number of samples comes with the cost of shorter memory of the particle set. In particular, a multi-modal sample set may deteriorate, when a single mode dominates in multiple iterations.

Limitations For the sake of simplicity, this section did not cover several possible extensions. (1) The model does not adapt to — even linear — velocity, as there is no velocity term. Here, this is an unwanted source of lag, but reflects the notion that nothing about the motion is known, other than its maximum magnitude. (2) Furthermore, neither non-isotropic likelihood function, nor non-isotropic estimation kernel are introduced. It is however possible to extend the model in this direction, but approximations are more difficult. (3) A fixed size diffusion kernel is assumed. The diffusion kernel's width might be adapted to the measured result standard deviation, as suggested in literature [DR05]. (4) The resampling scheme is assumed to be multinomial resampling, thus completely unbiased and random. There exist other algorithms in literature [DC05; HSG06; CKC17], which implement resampling differently. While random resampling may be less optimal to the application, its indepdence assumption is helpful for modeling. Therefore, a lower limit for the performance is obtained. (5) The theory for multiple iterations only poses a lower limit, but in case of strong noise is outperformed by the algorithm. Measurement exponentiation by an annealing schedule was considered, but not integrated into the scheme.

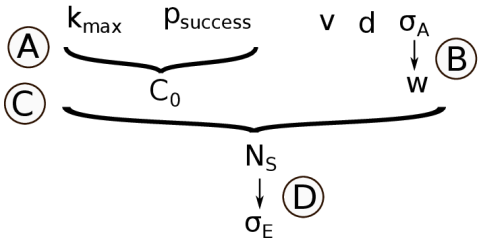
2.4.2. Parametrization Scheme

This chapter offers a 'recipe' (Fig. 2.18) for parameterizing a specific Particle Filter algorithm for following a target, which admits a likelihood function of limited support. It essentially sets the parametrization for being able to successfully detect the target along

a path. In order to further refine the obtained position estimate, additional iterations, such as an annealing schedule may be employed. Due to the complexity of the process, this has not been covered in this chapter. However, it will be utilized in the following.

The parametrization scheme for the detection stage starts out by determining the obtainable σ_A from the data: The measurement function for features is applied to the image input and the characteristic scale of the response peaks is obtained. Of these, the smallest σ_A is to be used as reference. Furthermore, a baseline noise level can be measured outside these peaks. If it exists, either the measurement function may be changed to suppress it, or the average level α may be carried into a noise based computation.

With $\alpha = 0$, it is possible to define typical parameters and demand a certain success probability. The velocity v is to be maintained for k_{max} of frames without loss of track with the probability $p_{success}$ in d dimensions. From k_{max} and $p_{success}$, the value c_0 is computed in (Eq. 2.17). If the measurement uncertainty is obtained as standard deviation σ_A it may need ot be scaled in order to convert it into a valid range w. Rough conversion values can be found in table 2.1. Then, equation (Eq. 2.26) allows the computation of the number of required samples. Finally, the required estimation range σ_E is obtained from (Eq. 2.27).



Step	Reference
A	(Eq. 2.17): $c_0 = -\ln(1 - \sqrt[k_{max}]{p_{success}})$
B	tab. 2.1
C	(Eq. 2.26): $\frac{v}{w}\big\|_{opt} = \frac{\sqrt{d}}{\sqrt{2\pi e}} \left(\frac{N_S}{c_0}\right)^{1/d}$
D	(Eq. 2.27): $\frac{\sigma_E}{w}\big\|_{opt} = \frac{1}{\sqrt{2\pi e}} \left(\frac{N_S}{c_0}\right)^{1/d}$

Figure 2.18.: Proposed parametrization scheme for the noise free ($\alpha = 0$) case.

If typical σ_A differ for the dimensions of state space, σ_E is to be set for each of them individually.

With noise involved, value c_0 in equations (Eq. 2.26), equations (Eq. 2.27) in steps C,D is to be replaced as in (Eq. 2.17), which leads to a different number of samples required to follow the target. However, the position estimate may still suffer from noise. At this point in (Eq. 2.17), term α may be replaced by $\alpha^{2/3}$, in order to emphasize the deprivation of samples for the estimate as motivated in *Transition point* (2.1.5.3).

It is not necessarily required to improve estimate quality in this manner. The application of multiple iterations per step may be used to improve the quality. Unless the number of samples is extremely low, it has been shown to be beneficial to spread samples over multiple iterations, keeping $N_S N_I$ constant. The additional iterations may be used to improve the estimate quality and reduce lag. For this, an annealing schedule may be applied, which has not been discussed in this chapter.

Example An exemplary computation might be structured like this: In 2D EPI image data of liver motion, smallest landmarks take approximately on voxel in extent. The valid range of a likelihood function is $w = 1\,\text{vx}$. Directed, rapid motion in the sequence shows as $v = 2\,\text{vx/frame}$, over an duration of 15 frames from full inhalation to exhalation. With a goal of being able to follow the motion in $p_{success} = 90\%$ of the cases, computation (C) yields the required $N_S \approx 169$ particles. Step (D) reveals the required $\sigma_E = \sqrt{2}\,\text{vx} \approx 1.4\,\text{vx}$. Doubling the expected velocity to $v = 4\,\text{vx/frame}$ yields $N_S \approx 677 \approx 4 \times 169$ particles at a $\sigma_E = 2\sqrt{2}\,\text{vx} \approx 2.8\,\text{vx}$. Increasing the desired success probability to 0.99 would raise the number of particles to $N_S = 250$ and $N_S = 999$, respectively.

Approximating the object region by a truncated Gaussian of $\sigma_A = w/\sqrt{8}$, the expected residual error in the track estimate is given by (Eq. 2.33). It is $\epsilon \approx 0.12\,\text{vx}$ for case $\sigma_E = \sqrt{2}$ and $\epsilon \approx 0.06\,\text{vx}$ for case $\sigma_E = 2\sqrt{2}$. This inverse effect is a result of lag being reduced by a wider σ_E (Eq. 2.11) and is affordable only by the $4\times$ increased number of samples.

Noise in the measurement process may deteriorate the result. The updated equation 2.41 leads to the same estimating σ_E. However, an increased number of samples is required. For $v = 2\,\text{vx/frame}$, the sample count at $\alpha = 0.01$ increases from 169 to 226.

2.4.3. Conclusion

This chapter has investigated and described properties of a very simple Particle Filter model. Particular properties of the model include a limitation of the support of the likelihood function and the reduction of the estimation process to a plain Gaussian diffusion. Three sections investigated and explained different properties of the model:

- Sampling stability with respect to a potential loss of track was described, such that a parametrization for a desired success probability is possible.
- In case of a successful track, estimates for the expectable residual error have been motivated and confirmed.
- For the case of noise being added into the measurement process, models have been derived to explain the impact on the track stability and decrease of estimate quality.

Though these properties only cover a particular type of algorithm, a basic understanding for the interdependency of parameters can be gained and employed in the configuration.

2.5. List of variables

Group	Var.	Brief	Detail
Common	$\mu_{...}$	Mean	Mean (of a Gaussian distribution)
	$\sigma_{...}$	Standard Deviation	Standard deviation (of a Gaussian distribution)
	$\mathcal{L}$	Likelihood Function	
Indices	i, k	Indices	Indices for samples $i = 1..N_S$
			Index of a temporal step $k = 1..k_{max}$
Particles	s_i^k	Sample State	State of sample i in step k
	π_i^k	Sample Weight	Weight of sample i in step k
	σ_E	Estimation Width	Standard deviation of Gaussian transition kernel in estimation process
	σ_A	Appearance Width	Standard deviation of Gaussian likelihood of the observation
Motion	v	Velocity	Dimensionless linear velocity in x-direction
	s_k	Position	True position in step k
Noise	σ_{LH}	Measurement Noise	Standard deviation of additive measurement noise
	α	Measurement Noise	Effective, mean level of noise, relative to maximum likelihood value.
Sampling Stability	w	Region Width	Width of a valid region to be sampled from for a success step
	p_S	Single Sample Probability	Probability of a single sample reaching the valid region
	p_M	Multi Sample Probability	Probability for any of multiple samples reaching the valid region
	$p_{success}$	Success Probability	Probability for successfully sampling from the valid regions of k steps, using N_S samples
Multi-State Model	ρ_j^k		Probability mass of all particles having missed the valid region j times. j is the discrete index of a "noise mode".
	γ_j		Probability of detecting the valid region from noise mode j.
Two-State Model	ρ_N^k		Probability of a sample being in the 'noise' region.
	ρ_C^k		Probability of a sample being in the 'correct' region.
	γ		Probability of a sample remaining in the valid region during after estimation and observation.
	$\rho_{C,E}^k$		Probability of a sample being in the 'correct' region, after estimation and observation.

3. Fixed Hierarchy Model

In order to apply the theoretical model of the previous chapter to real-word scenarios, it needs to be coupled to the actual sensor data. For the given medical image data this means that for the present landmarks, both their appearance and possible transformations need to be modeled. This chapter will establish a hierarchical transformation and appearance scheme for medical image data.

When observing an object to be tracked, different features and scales of features allow for its localization. At multiple image resolutions, these may appear as differently sized blobs. Search range for and uncertainty in the position of a feature are often related to its scale. This can be made use of for an iterative optimization approach to tracking. Starting localization with large scale features, a fast, but still coarse position estimate is obtained. Proceeding iterations then refine the estimate using smaller scale features up to the desired level of detail.

At the same time, multiple small scale features may only become distinguishable by a pattern in their spatial arrangement. If multiple, similar features exist on a small scale level, they can only be distinguished using additional information from a larger scale. This applies for instance to many liver blood vessel cross sections in 2D EPI image data.

In both cases, hypotheses are passed on through multiple levels of detail to form a final state estimate. This approach can yield efficient optimization algorithms, and be expressed as a hierarchical evaluation scheme. This chapter proposes a hierarchical scheme, which allows different scale transformations and observation using a hierarchical tree. Nodes of the tree represent the actual tracking entities, responsible for the optimization process. Edges specify the relationships between them.

The single instance of *Gaussian Transition Filter Model* (2) yields a suitable model for very simple situations with few degrees of freedom. In order to apply it to more articulate situations, multiple of these instances will be joined as nodes in a hierarchical tree. This chapter will present the extension to the required hierarchical transformation. The scheme is described in a general fashion and this chapter will present concrete results for a fixed two level hierarchy. Further steps to approach medical image data are shown. The chapter will present a sparse representation and matching function for landmarks. With the intent of real-time applicability, a sparse gradient based description is suggested. Moreover, a feature detector for the automated, initial detection of liver vessels is motivated.

In the experimental section, the basic components are examined individually. For a controlled environment, initial experiments are conducted based on artificially generated data. The second part of the experiment then moves towards annotated medical image

series. Investigations of the proposed two-level model are undertaken on three example cases.

3.1. Modeling

Advancing from the plain theoretical model in chapter *Gaussian Transition Filter Model* (2) to a realistic application, two particular properties of the algorithm need to be specified in more detail. The theoretical model used a simplifying Gaussian likelihood function in state space. In an application, the relation between the state space and the data needs to be defined (Fig. 3.1). The transformation model will translate a point in state space — a hypothesis — into a linear transformation. The appearance model will describe a feature or landmark in the data in a local landmark coordinate system. The transformation will be used to transform a point in landmark space into image space. By joining transformation and landmark model a function to evaluate the hypothesis is obtained. Applying it to the image then yields the weight for the hypothesis.

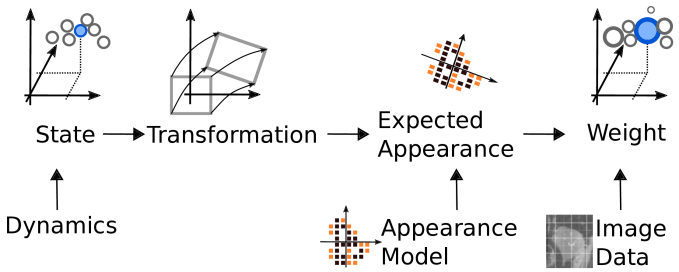


Figure 3.1.: When working with real work data, transformation and appearance have to be specified.

This section present the transformation model and an appearance model for medical image data. The appearance model will be designed to describe landmarks observable in the image data. The transformation model will be used to transform them into image space. A hierarchical model will be presented in this chapter. Child nodes provide residual transformations to the transformations of their parent node and describe the motion of smaller regions than their parents. Each node's configuration is driven by an own state space. The dynamics of each node in the tree will use a model closely related to chapter *Gaussian Transition Filter Model* (2).

3.1.1. Hierarchical Transformation Model

A tracking algorithm typically assumes that the tracked object does not change its arrangement beyond an acceptable magnitude — it requires a minimum amount of correlation among appearance and motion. Correlation is the main motivation for the introduction of a hierarchical evaluation scheme. It manifests itself in the joint presence

of landmarks and their related joint motion. Hierarchy is intended to dissect both appearance and motion into correlated and uncorrelated segments: In a hierarchical scheme, nodes of a common parent share a common property. A parent node encodes the correlated appearance and motion of its child nodes. Among each other, children show smaller scale correlation than their parent node. Thus, proceeding from root to leaves of such a tree, the correlation among entities as well as their scale of appearance and distinguishable motion decrease.

Here, nodes of the hierarchy are chosen as entities of the optimization process of tracking. Results of parent nodes are passed on along the edges to child nodes, which incorporate their parent's result into the optimization process. While the node in its functions itself will be specified later, it holds descriptive elements along the hierarchical tree — transformations and observations.

3.1.1.1. State and Transformation Space

The hierarchy of transformations aimed for, here, is a *scene graph* [Cla76] like tree (Fig. 3.2): It uses child levels to describe residual transforms with respect to transformations on parent level. The state space of a child level encodes the difference of transformation to the parent level. This implies that the interpretation of state space on child level depends on states on the parent level. States of different levels are not expected to be compatible to each other: The full transformation of a child level cannot necessarily be expressed as an absolute state in either space. Therefore, states are converted to transformations first (Fig. 3.1), which are then joined into a single transformation (Fig. 3.2). This auxiliary space will be referred to as *transformation space*. The motif of *state spaces* is however kept in order to maintain low numbers of easily interpretable degrees of freedom.

As in a scene graph, transformations are computed as paths, starting from the root node and finishing at the current node of interest. These kinds of computations are required on multiple occasions for tracking. The full path transformation of a node is used to update the related weight of the particle set, or to return the transformations center as output. The paths from root and the path to a leaf, excluding the node itself, will be referred to as *root path* and *leaf path*, respectively. For each node, there exists at most one root path, whereas the number of leaf paths is not limited. The root path defines the parent coordinate system of a node. It is therefore a necessary element for the definition of transformation hypotheses. Leaf paths only need evaluation if one is interested in the location of the leaves of the node. This is for instance the case if the appearance model of a node is defined as a conjunction of models only available at the leaves. This will be presented in the further text.

In one tracking step, the hierarchical graph is updated level by level, traversing from root to leaves. For a single node, the evaluation of a transformation chain appears along a path, such that it is not affected by any neighboring paths. Thus, in the following, transformation chains are derived with a selected path in mind: Indices for particular nodes on the same level of a tree level are neglected for convenience. The same applied to the temporal step k. On each level l of the path, a transformation matrix M_l can be

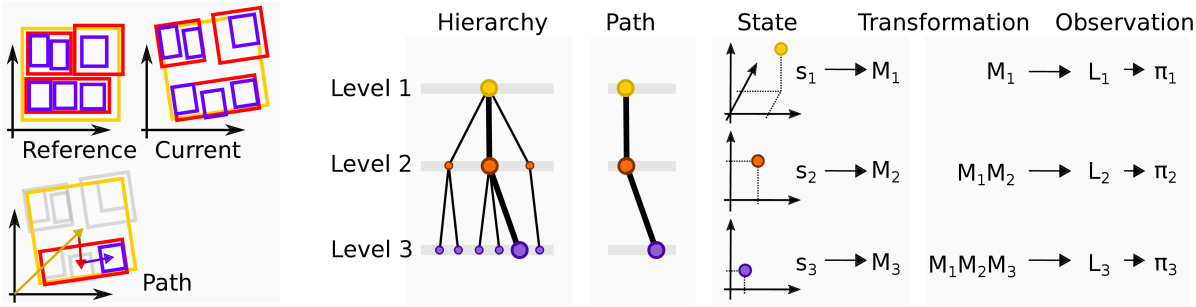


Figure 3.2.: Hierarchy, states, transformations and observation for nodes along a path. Example in which level 1 allows for 3 DOF (translation x,y, rotation around z) and levels 2 and 3 for 2 DOF (translation x,y). Left: Sample transformation from reference to current (top) and one highlighted path of the current transformation (bottom). Right: Realization in the algorithm.

formed, which depends on the associated ensemble of particles $\{S^l\}$. The joint, or *full* transformation for a path from root to level l is the product

$$M_l^{full} = M_1 M_2 ... M_{l-1} M_l = \prod_{i=1}^{l} M_i \tag{3.1}$$

Within each node, the level related transformation M^l consists of a fixed and a dynamic part.

$$M_l = M_l^{pc} M_l^{dyn}(f(\{S_l\})) \tag{3.2}$$

Static part M_l^{pc} describes the time independent transformation from parent's to current child's coordinate system's origin. In general, this transformation differs for all children of the same parent. It is typically set at the time of initialization and may also be regarded as a fixed property of the edge between parent and child node. The dynamic part $M_l^{dyn}(f(\{S_l\}))$ is governed by a model as presented in chapter *Gaussian Transition Filter Model* (2) and the feature of a (child) node. It captures transformations with respect to the child coordinate system's origin. These transformations depend on the state which function $f()$ extracts from the associated sample set $\{S_l\}$. Dependent on context, function f may yield individual states from the particle set or statistical measures such as MAP and mean state. The former is required in order to be able to assign weights to the individual states. The latter is used to obtain estimates for the level, but also to reduce complexity along the hierarchical tree. In order to evaluate a path, a function f must be chosen for every level involved. Here, we use few dynamic and a static transformation of interest. Dynamic estimates are given in prior and current ensemble transformation estimates $\hat{M}_{l,k-1}$ and $\hat{M}_{l,k}$ such as mean or MAP state transformation of the related level. A further particular, static transformation is given in the reference where $M_l^{dyn} = M_l^{dyn}(s_{ref}) = I$ and therefore $M_l = M_l^{pc}$. This transformation

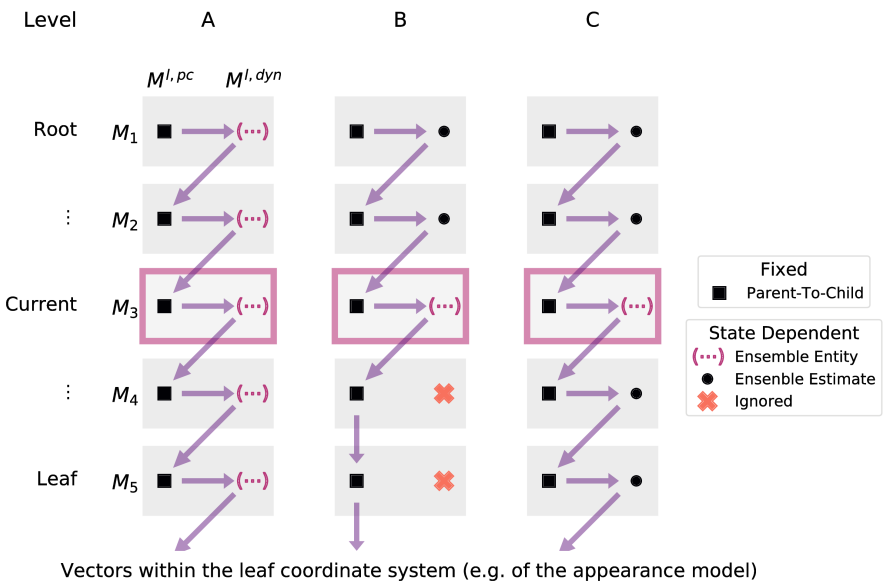


Figure 3.3.: Transformation along a path of five levels as a product of level dependent transformations M_l (eqns. 3.1, 3.2). Each level's transformation M_l is composed of a static parent-to-child M_l^{pc} and a dynamic transformation M_l^{dyn} derived from the related level's sample set. In general (A) any sample of each level might be combined among each other, but can not be represented by level-bound weights. In order to compute weights for the samples of the current level, all other levels have to be set to a unique transformation.

relates stage l to its reference transformation, as that it ignores the state of the sample set.

Evaluation of different transformations from sample sets along a path allows for many combinatorial options. Three of these are depicted in Figure 3.3. Generally, it is possible to combine single state-related transformations with $f(S^l) = s_i$ of any level among each other (case A). This leads to a vast increase in complexity. It is particularly inconsistent to assigning unique weights to the states of each level. Therefore, when processing or obtaining results from a certain level, only states on one level are treated independently. For all other — predecessor and successor — levels, a single transformation is required (cases B,C). In this sense, the interpretation of individual states depends on the current root and leaf paths of the node.

In both configurations (B,C) root path transformations are set by ensemble estimates. While (B) seeks a known reference configuration, (C) tries to detect one which has been updated by altered leaf path transformation. In this sense, the transformation path of type (B) is used when adapting the transformation only in top-down-direction. Paths of type (C) feed back updated transformations and make them available to parent levels.

Transformation (Eq. 3.1) of the current node l therefore is

$$M_l^{full} = \left(\prod_{i=1}^{l-1} \hat{M}_i \right) M_l \tag{3.3}$$

This transformation may suffice for the current level. However, if one is interested in the leaf transformation, given this current level's transformation, one needs to set the leaf path as well. Figure 3.3 shows two configurations in this regard. Configuration (B) sets the leaf path transformation by the reference configuration

$$M_L^{full,(B)} = M_l^{full} \prod_{i=l+1}^{L} M_i^{pc} \tag{3.4}$$

In configuration (C), the leaf path up to node L is set by transformation estimates, such that

$$M_L^{full,(C)} = M_l^{full} \prod_{i=l+1}^{L} \hat{M}_i \tag{3.5}$$

This implies that when updating level l, levels $> l$ are integrated by an updated version, instead of their reference as in (B).

Weights stored on each level represent the quality of a match of the local transformation, given already computed predecessors and possibly, successors. Two ways of associating appearance models to the hierarchical tree can immediately be considered: First, an appearance model can be defined for each node individually. Second, along a hierarchical tree it may also exist as a conjunction of multiple smaller models. A *node based* approach associates individual models to all nodes. These landmark models are not updated. Therefore, this approach seeks a reference representation. A *leaf based* approach assigns them to leaf nodes only (Fig. 3.4). The appearance model for a node is then computed by transforming the leaf models into the node's coordinate system. For this, the updated leaf paths (C) or reference leaf paths (B) are available. Therefore, depending on what is used, the leaf based modeling can either seek the prior or the reference configuration, respectively.

This chapter will employ a node based approach. The algorithm will therefore seek the reference appearance of landmarks. The motivation for this is the prevention of model drift. It is the common approach of medical image registration. While the leaf based approach will be tried out here, it is further generalized in the proceeding chapter *Dynamic Hierarchy Model* (4) before being applied thoroughly.

Two Level Model Moving towards an actual application, the 'Two Level Model' will introduce a fixed hierarchy of two levels — the root and directly attached leaf level (Fig. 3.5). In this chapter, root and leaf level will be referred to as *global* and *local* levels, respectively. On root and leaf level, global and local states S^G, S^L specify transformations M^G and M^L, respectively. A global level transformation is computed

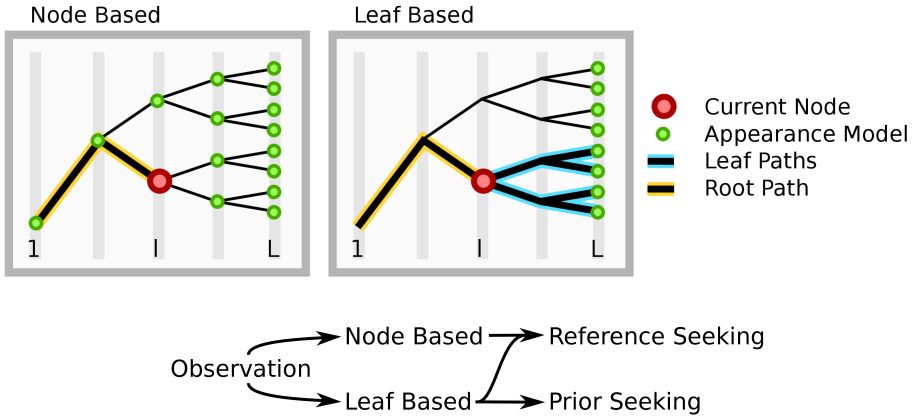


Figure 3.4.: Node and leaf based appearance modeling. The node based approach provides individual appearance models for each node. The leaf based approach provides them for leaves only. Node appearance models are then constructed from prior or reference leaf paths and leaf models (Prior/Reference Seeking, resp.).

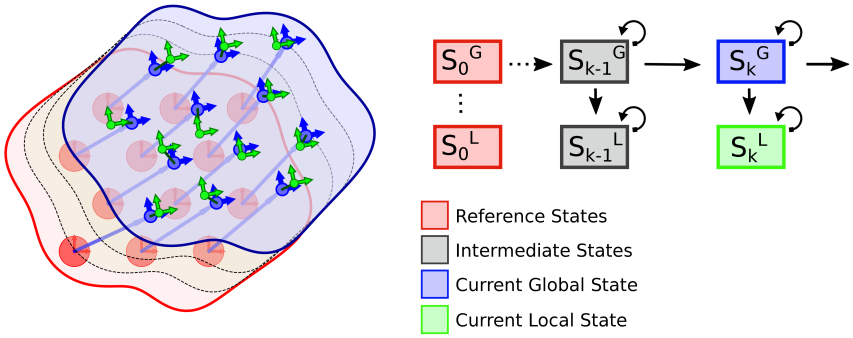


Figure 3.5.: Left: Global transformation from reference state (red) to current global state (blue) and local transformation within the current global coordinate system (green). Right: Filtering scheme applied in the tracking process: The global state S^G are estimated based on their temporal priors, while local states' S^G estimation uses the current global state as prior. In each estimation step an annealing step can be introduced (circular arrow).

as the product of temporarily constant root and current global state's transformation $M^G_{k,i} = M^{0G} M^G(s^G_{k,i})$, where k and i index step and particle, respectively. Transformations $M^G_{k,i}$ are used to obtain respective weights $\pi^G_{k,i}$ from the global appearance model $\mathcal{L}^G$. Result of the global stage is a weighted mean state $\hat{s}^G_k$, which is associated with the transformation

$$\hat{M}^G_k = M^{0G} M^G(\hat{s}^G_k) \tag{3.6}$$

with $\hat{s^G_k} = \sum_i \pi^G_{k,i} s^G_{k,i}$, and normalized weights $\sum_i \pi^G_{k,i} = 1$. It is provided to the local stage as reference transformation prior.

Analogously, local transformations are computed for the leaf stage. For a particular leaf the static parent-to-child transformation is joined with current local state transformations $M^L_{k,i} = M^{GL} M^L(s^L_{k,i})$, where i indexes local states. These exist in the prior coordinate system, estimated as $\hat{M}^G_k$, such that full estimates of local transformations are conditioned on the global estimate:

$$\hat{M}^{L|G}_{k,i} = \hat{M}^G_k M^L_{k,i} = M^{0G} M^G(\hat{s}^G_k) M^{GL} M^L(s^L_{k,i}) \tag{3.7}$$

These are used to transform the appearance models positions $\vec{v}$, obtain the local stages weights $\pi^{L|G}_{k,i}$. Again, $\sum_i \pi^{L|G}_{k,i} = 1$ and finally the mean local transformation

$$\hat{M}^{L|G} = M^{GL} M^L \left(\sum_i \pi^{L|G}_{i,k} s^L_{i,k} \right) \tag{3.8}$$

Again, an annealing schedule may be applied before finally, the mean transformation estimate

$$\hat{M}^{full}_k = \hat{M}^G_k \hat{M}^{L|G}_k \tag{3.9}$$

is obtained. As each appearance model's coordinate system is chosen to be centered locally (at vector $\vec{0}$), the two level position estimate for a landmark is computed as $\hat{M}^{full}_k \vec{0}$.

3.1.1.2. Estimates

Position estimates and uncertainty are computed per level. Here, they are obtained for the node they are computed for. For the chosen current level, the parent transformation estimate is regarded as static. In the selected reference seeking approach child transformations are not considered. For the two level model this means for the computation of root level position estimates (Fig. 3.6, case A): The transformation chain's current set of transformations is given by the root's set. Child transformations are not set, besides the child individual transformations M_{gl}. In order to estimate the mean position for leaf nodes, the mean is computed as

$$\hat{\vec{r}}_k|\text{root} = \sum_i \pi^G_{i,k} M^{0G} T^G_{i,k} M^{GL} \vec{0} \tag{3.10}$$

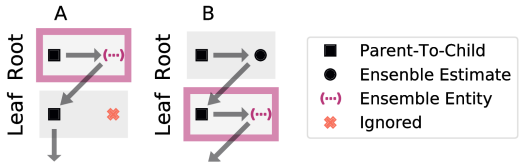


Figure 3.6.: Transformation stages of the two level filter: Obtaining transformations for a correction by the global stage only (left) or for both stages (right).

with associated variance of spatial distance

$$\sigma^2_{\vec{r}_k}|\text{root} = \sum_i \pi^G_{i,k} \left| \hat{\vec{r}}_k|\text{root} - M^{0G} T^G_{i,k} M^{GL} \vec{0} \right|^2 . \tag{3.11}$$

For the local level (Fig. 3.6, case B), the global transformation is regarded as given, such that conditional position and variance estimate are

$$\hat{\vec{r}}_k|\text{leaf} = \sum_i \pi^l_{i,k} \hat{M}^G_k M^{GL} M^G_{i,k} \vec{0} \text{ and} \tag{3.12}$$

$$\sigma^2_{\vec{r}_k}|\text{leaf} = \sum_i \pi^G_{i,k} \left| \hat{\vec{r}}_k|\text{leaf} - M^{0G} T^G_{i,k} M^{GL} \vec{0} \right|^2 \tag{3.13}$$

The notation $\cdot|\text{root}$ describes estimated location after correcting for the correlated component of motion, whereas $\cdot|\text{leaf}$ denotes those after an additional correction of local differences.

3.1.1.3. Temporal Evolution of States

The proposed hierarchical scheme treats parent level results as inputs to their child nodes: Nodes may receive prior input by their parent node's posterior transformation estimate and their own temporal distribution. These will be termed *hierarchical prior* and *temporal prior*. A hierarchical prior is available to all, except the root node. Temporal priors are available to all nodes, but are conditioned on respective parent coordinate systems, except for the root node. The root is the only node to function as an independent Particle Filter. All others must integrate a mixture of hierarchical and temporal priors in order to generate and interpret new states (Fig. 3.7, left). Details of the particular integration depend on the tracking algorithm. In general, the hierarchical tree itself is intended to reflect a known, fixed representation of the target to be modeled. Throughout the course of tracking it remains unchanged, and the system's dynamics are managed by the node entities of the hierarchy. If an underlying hierarchy is not known, it may be necessary to adapt the current tree into another, which will be discussed in chapter *Dynamic Hierarchy Model* (4). The tracking process is executed from root to leaves, processing the levels sequentially. Each node is updated given information gained on prior temporal steps and hierarchical levels. In a common scheme, the

tracking process uses the input information to run a single or multiple update passes for its distribution of local transformation states (Alg. 3.1). With each iterative step, different σ_E and λ for estimation and annealing may be applied. The adaptation of σ_E may be done, setting the diffusion values in the respective dimensions of state space equal to the empirical standard deviation therein. This will be referred to as *adaptive diffusion*, whereas leaving diffusive values constant will be referred to as *static diffusion*. The default model is chosen to only propagate the temporal state prior of the root node. Leaf nodes' states are reset to reference in each pass, but disperse due to their dynamics within a pass. This is done in order to prevent local nodes from drifting, if their local model is uncertain.

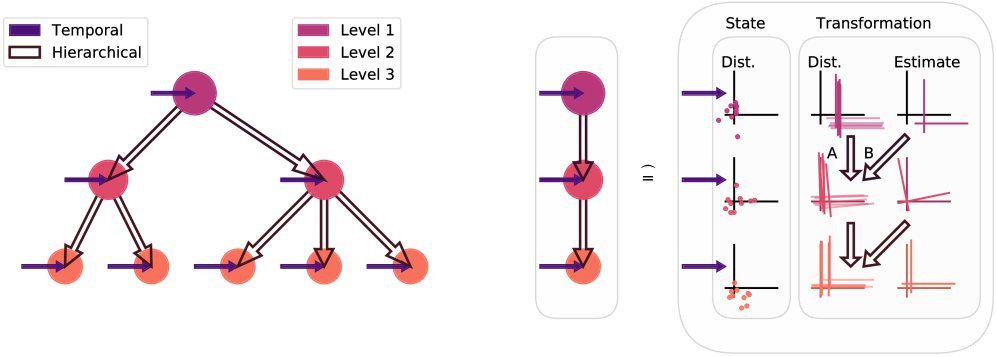


Figure 3.7.: Temporal and hierarchical priors in hierarchical tree (left) and their propagation along a branch (right). Cases (A),(B) correspond to using the ensemble of transformations or an estimate thereof as prior for the child level (see Fig. 3.3)

Input : Parent transformation T_P and temporal prior
Parameters: N_S, $\mathcal{L}$, $\{(\sigma_E, \lambda)\}$
Result: Updated distribution and transformation estimate
Use parent $\hat{T}_P$ as reference transformation.
for (λ, σ_E) in $\{(\lambda, \sigma_E)\}$ **do**
  Re-sample s_i^* from own prior $\{(s_i, \pi_i)\}$
  Estimate $s_i = \eta$ with $\eta \propto \mathcal{N}(s_i^*; 0, \sigma_E^2)$
  Assign weights $\pi_i = \mathcal{L}^\lambda(\hat{T}_P T(s_i))$
Compute mean state $\hat{x}$ and transformation estimate $T(\hat{x})$

Algorithm 3.1: Generic Filter Block

3.1.2. Contrast Based Appearance Model

Data acquired from physical sensors may not be interpretable directly. Often, it is obtained from a stream of data which is interpretable by a human. Further processing may be required, in order to make it interpretable by an algorithm. This task is undertaken by the appearance model. It defines a likelihood function, which specifies, how strongly the data is in agreement with proposed state hypotheses.

This section will present a sparse appearance model — an entity which allows for assigning weights to pose hypotheses. The model will describe landmarks by the contrast on the edges of structures. It will be reduced to the main describing entities in order to allow for real-time applications. Its phenomenological likelihood function will be designed to yield high values on matches and zero in non-matching regions.

3.1.2.1. Reduced, Transformable Landmark Model

A model is chosen, which changes the evaluation function by a spatial transformation and applies it to the current image. Aim of such model is the fast identification of a reference landmark in a current image.

The model will hold weighted positions and an associated measurement function. Two distinct properties determine the general functioning: (1) The *reduction process* for selecting the spatial layout. (2) The selection of a *measurement function* and related *assignment of weights* to the spatial layout. After a common introduction to the parameters, these properties will be discussed.

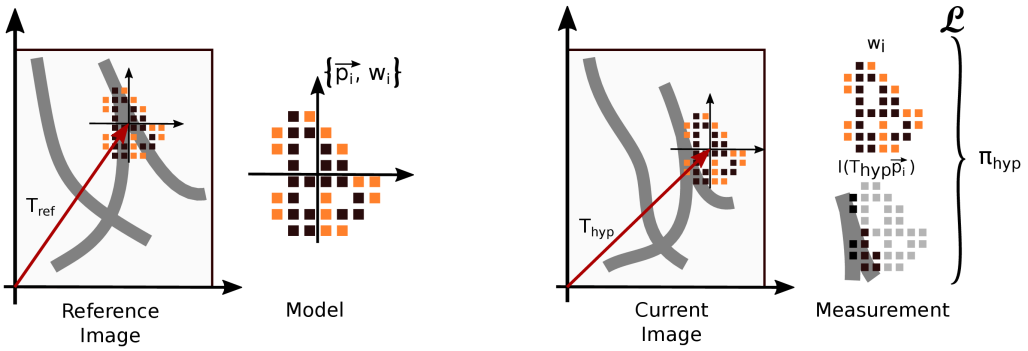


Figure 3.8.: Left: Fixed transformation T_{ref} positions the model in its reference location. Right: Hypotheses transformations T_{hyp} are used to evaluate hypotheses in the current image. Measurement function $\mathcal{L}$ joins weights $\{w_i\}$ and intensities $\{I(T_{hyp}p_i)\}$ into sample hypotheses weight π_{hyp}. Contrast based weights $w_i \in \{+1, -1\}$ indicated by color.

Parameters The model itself is specified by a set of weighted positions and an associated evaluation function: A set of N_P relevant voxel positions $\vec{p}_i$ is defined with respect

to the transformation center of the reference landmark. Initial voxel positions are therefore given by applying the reference transformation $T_{ref}\vec{p_i}$. A descriptive value w_i is associated with each position $\vec{p_i}$ (Fig. 3.8). Here, this value may for instance represent the initial patch's voxel intensity within the reference image. Alternatively, a derived weight, such as the response of a filter applied to the image may be of interest.

State hypotheses are provided to the model as transformation matrices T_{hyp}. Once parametrized by $\{\vec{p_i}, w_i\}$ with $i \in 1..N_P$, the evaluation function

$$\mathcal{L}(T_{hyp}|I_{cur}, \{\vec{p_i}, w_i\}) \tag{3.14}$$

assigns a weight value to transformation hypotheses T_{hyp}, given the current image I_{cur}. There are several options to determine a weight via $\mathcal{L}$. Three variants applied in this work are depicted in table 3.1.

For instance, in template matching, the set $\{\vec{p_i}\}$ includes *all* voxels of the landmark, carrying their initial intensity set $w_i = I_{ref}(T_{ref}\vec{p_i})$ as descriptive values. Transformation hypotheses T_{hyp} consist of integer translations along the Euclidean axes and the evaluation function is given as a correlation between the set of descriptive values w_i and the set of intensities at $I_{cur}(T_{hyp}\vec{p_i})$.

	Weights w_i	Measurement function $\mathcal{L}(T_{hyp}\|I_{cur}, \{\vec{p_i}, w_i\})$
SSD	$I_{ref}(T_{ref}\vec{p_i})$	$\exp(-\beta_{SSD}\sum_i(I(T_{hyp}\vec{p_i}) - w_i)^2)$
Correlation	$I_{ref}(T_{ref}\vec{p_i})$	$\mathrm{cov}(I(T_{hyp}\vec{p_i}), w_i)/[\mathrm{var}(I(T_{hyp}\vec{p_i}))\,\mathrm{var}(w_i)]$
Contrast	$+1, -1$	see section 3.1.2.2

Table 3.1.: Chosen measurement functions $\mathcal{L}$ and related choice of weights w_i.

Plain template matching is however computationally expensive and restricted in the degrees of freedom. Here, arbitrary transformations are allowed, but the set of descriptive points is reduced and the weighting function adapted.

Reduction Function The computational cost of evaluating a transformation hypothesis scales at least linearly with the number of points. In order to lower this cost, the set of points is reduced, when the model is generated. In this work, a set of initial candidate voxels is translated into a set of candidate entities. An energy function for subsets of these entities is defined and the subset minimizing the energy is chosen. Entities are then translated back into voxel positions and assigned a weight for integration into the appearance model. The detailed process is described as follows and depicted in figure 3.9:

Within a given range of an initial position, all voxel positions are considered for integration into the appearance model. A selection process extracts entities from the initial region and selects a subset from it, which holds N_P voxels. These entities may for instance be voxel positions (single site) or position pairs describing a gradient (paired). With each entity i, an energy ϵ_i and a spatial center position $\vec{c_i}$ are associated.

From the set $\mathcal{A}$ of available entities, only a fixed size subset $\mathcal{A}^+$ is to be selected to form the model. This subset is chosen to minimize an 'energy' function which consists of single

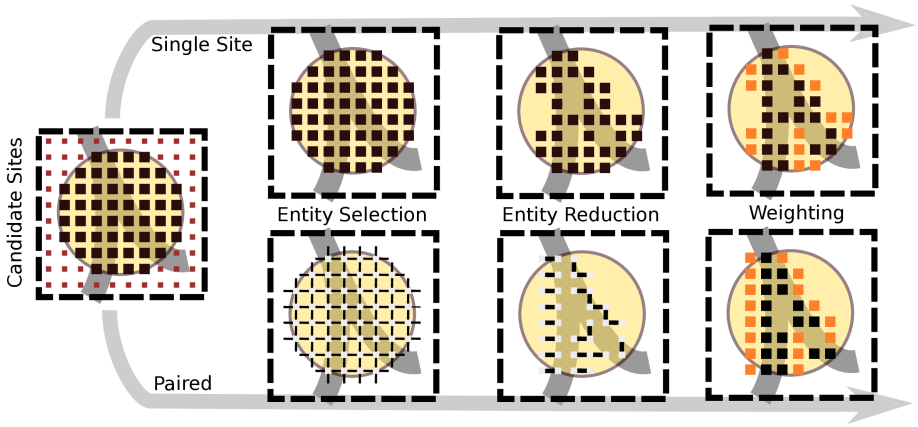


Figure 3.9.: Generation process for appearance model. The reduction process is either single site based (top) or site pair based (bottom) and determines which voxels to integrate into the description. These voxels are then represented by weights, chosen depending on the desired measurement function.

term energies ϵ_i and symmetric pair energies $\kappa_{i,j} = \kappa_{j,i}$. The energy of configuration a is defined as

$$E(a) = \sum_{i \in a} \epsilon_i + \sum_{i,j \in a, j>i} \kappa_{i,j} \tag{3.15}$$

The terms ϵ_i assign single entity energies, which rank entities among each other. Interaction term $\kappa_{i,j}$ is used to disperse chosen pairs and modeled as Heaviside-function $\kappa_{i,j} = \kappa_0 \Theta(d_{min} - |\vec{c}_i - \vec{c}_j|)$ which contributes $\kappa_0 \gg \epsilon_i$ if pairs are closer to each other than a minimum distance d_{min}. The optimization process is to determine the minimum energy configuration

$$\mathcal{A}^+ = \arg\min_a E(a) \tag{3.16}$$

The functional (Eq. 3.16) of configuration a can not be minimized deterministically. Simulated Annealing and alternatively a greedy algorithm are applied to approximate a solution. After the selection of entities, these are integrated into the appearance model as weighted voxel positions. The selected weights depend on the chosen measurement function (e.g. table 3.1). **Note:** Here, a selection process for point pairs was shown. In order to select a contrast type set of voxels using single site selection, in this work the selection process is run twice, with half the target number of points — once with positive and once with negative intensities as single site energies.

3.1.2.2. Sparse, Gradient Based Description

Common medical image data such as MRI magnitude images are represented by scalar, grey scale values. The level of image intensities is influenced by local as well as global

factors. Local factors are given by the imaged structures themselves and represent the way the imaged tissue responds to the imaging modality. Additional global changes in image intensity can arise from varying sensitivities of the modality in different areas of an image. These may for instance be coil sensitivity maps in MR or depth specific attenuation in ultrasound imaging. In order to generate a model which is independent of global intensity changes, local contrast descriptions can be suitable. This section proposes a concrete configuration for the reduction process and a contrast based weighting function.

Contrast Based Point Reduction In the generation process, pairs of points which form a gradient are considered for integration into the model. These pair entities are chosen as the pairs within a circular neighborhood around the desired landmark position. In order to keep growth of the number of candidates linear, only pairs with a point distance below a limit D_{max} are chosen. Each pair i with voxel positions $\vec{r}_{i,0}$ and $\vec{r}_{i,1}$ is characterized by its relative intensity difference $\Delta b_i = |I_{ref}(\vec{r}_{i,0}) - I_{ref}(\vec{r}_{i,1})|/(\max I_{ref} - \min I_{ref})$, the points' absolute spatial distance $D_i = |\vec{r}_{i,0} - \vec{r}_{i,1}|$ and the center position $\vec{c}_i = 0.5(\vec{r_{i,0}} + \vec{r}_{i,1})$.

The single entity energy term is chosen proportional to the intensity difference and inversely scaled by the distance $D_i = |\vec{r}_{i,0} - \vec{r}_{i,1}|$ with a power $\gamma \geq 0$. Scaling parameter γ is introduced to favor short distance pairs in cases of pairs with similar intensities (see Fig. 3.10).

$$\epsilon_i = -\left(\frac{\Delta b_i}{D_i^{\gamma}}\right)^2 \tag{3.17}$$

After optimization of (Eq. 3.16), the select set $\mathcal{A}^+$ of entities is split into equal size sets of *dark* and *bright* voxels, defining the lower and higher intensity voxels of all pairs each.

Contrast Based Observation Function In the contrast based weighting function the set of points associated with the model $\mathcal{P} = \{\vec{p}_l, w_l\}_{l=1..N_P}$ is split[1] into *bright* and *dark* subsets $\mathcal{P}_b$ and $\mathcal{P}_d$ which differ in their weights:

$$w_l = \begin{cases} +1 & \text{if } (\vec{p}_l, w_l) \in \mathcal{P}_b \\ -1 & \text{otherwise} \end{cases} \tag{3.18}$$

Weights w_l are used to compute a weight given image intensity at transformed positions of the descriptive positions $I(T_{hyp}\vec{p}_l)$. Image I and transformation T_{hyp} under consideration are merged with the weights in dot product kernel

$$\psi'(T_{hyp}, I, \mathcal{P}) = \sum_{l=1}^{N_P} w_l I(T_{hyp}\vec{p}_l). \tag{3.19}$$

[1]Here, the classification leading to this split is part of the pair based reduction algorithm already.

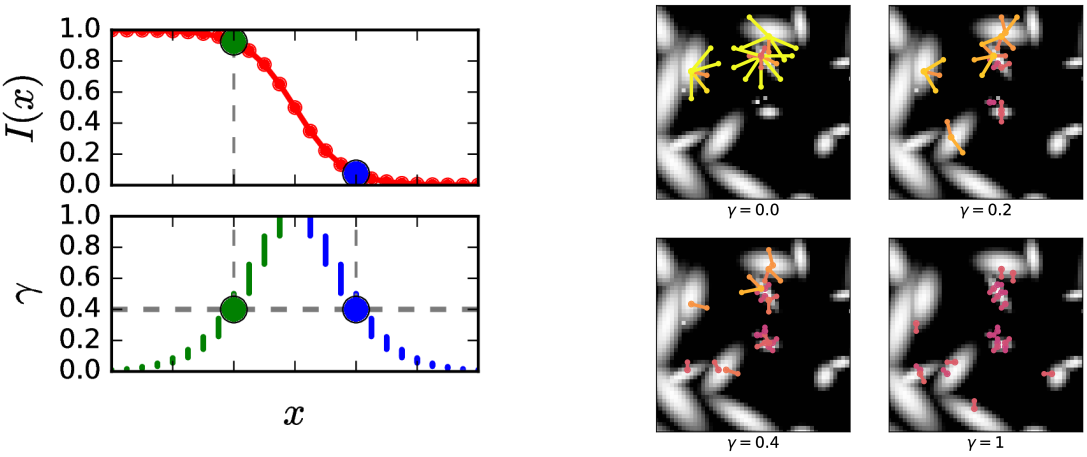


Figure 3.10.: Left: Example case effect of scaling parameter γ (eq. 3.17) on the point selection process for a 1D gradient (top). Voxel position x takes only discrete values. Bottom: Configurations for different values of γ (green: bright point, blue: dark point) and sample selection for $\gamma = 0.4$. Right: Process applied to 2D test image data using different γ.

such that

$$\mathcal{L}_\lambda(\psi') \propto \begin{cases} [\psi'(T_{hyp}, I, \mathcal{P})]^\lambda & \text{if } \psi'(T_{hyp}, I, \mathcal{P}) \geq 0 \\ \varepsilon \to 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

Equation 3.18 suppresses negative values of the kernel result ψ' which would imply inverted contrast. The parameter λ is used to emphasize high weighting in an annealing schedule. In practice, the value ε is used to circumvent an all-zero particle set. Value ε is strictly zero, if any hypotheses reaches $\psi' > 0$ and a positive constant if all $\psi' = 0$.

3.1.3. EPI Liver Feature Detector

In order to automatize the application for EPI liver motion tracking, it is necessary to detect landmarks in an initial image. Here, a Local Binary Pattern (LBP) based detector is motivated. While the motivation stems from LBP, instead of a target histogram a particular target pattern will be sought.

Seeking ideal vessel cross-sections in a single initial frame, one can demand, that the vessel should be bright and isotropic. A suitable vessel cross section must admit strong radial gradients. The proposed detector will therefore demand that radial gradients exceed tangential ones, in several directions. If the radius of the pattern is chosen in the order of the cross section, tangential gradients will indicate whether the vessel is isotropic. Since for each radial direction, such a binary decision can be made, the idea of local binary patterns is used. The decisions in the LBP is, whether the radial gradient

exceeds the tangential(Fig. 3.11). With center intensity b_R and external intensities b_i with cyclic index i, the radial gradient for direction i is $|b_R - b_i|$. Tangential gradients are found in $|b_i - b_{i+1}|$,$|b_i - b_{i-1}|$ and $|b_{i-1} - b_{i+1}|$. Requiring all tangential gradients to be lower than a fraction α of the radial yields decision rule

$$\mathrm{BIT}_i = \max\{|b_i - b_{i+1}|, |b_i - b_{i-1}|, |b_{i+1} - b_{i-1}|\} < \alpha |b_R - b_i|. \tag{3.21}$$

for a single bit of the binary pattern.

Applying the filter to an input reference image, leads to a binary pattern map. Despite the common practice of deriving local LBP histograms, here the map is used directly: For circular spots, only pattern 0 with $\mathrm{BIT}_i = 0 \forall i$ is of particular interest. Voxels with pattern 0 are masked and considered as landmark candidates. In order to select a subset, the voxels are ranked by the initial image intensities and voxels with highest intensity chosen first, which reflects the observation, that vessels appear brighter than the background. A minimum distance constraint prevents neighbor voxels from both being selected.

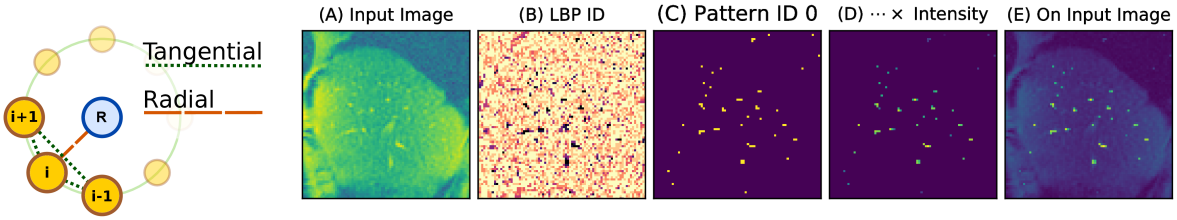


Figure 3.11.: Left: Instead of comparing reference (R) intensity to current intensity (i), the gradient based LBP compares the radial gradient ($R \rightarrow i$) to the tangential gradients ($i \leftrightarrow i+1$),($i \leftrightarrow i-1$),($i-1 \leftrightarrow i+1$). Right: Application of gradient based LBP to a liver vessel EPI image. From the input image (A), a LBP map is extracted (B) and filtered for pattern 0 (all radial gradients stronger than tangential, C). These locations are then weighted by the initial intensity (D and in E as overlay) and the N_{LM} best as initial position.

3.2. Experiments

This section describes experiments applying the two level algorithm to artificial and medical image data. Artificial data allows for the examination of the tracking performance in a controlled environment. In order to asses this performance, a simplified scenario is defined, which captures the main properties of the task. Once the algorithm is set up to solve it, controlled perturbations are introduced. Behavior, which was not regarded during parametrization, is evaluated and limitations of the algorithm explored. Artificial data of a simple breathing model will be used to understand the capabilities and limitations of the two-level transformation model. This model will be free of dedicated appearance descriptions. Inversely, a model for artificial landmark representation will be introduced to study the proposed appearance model. The experiments then proceed to medical image data. Initially the case of 2D EPI liver motion is analyzed. In this, the EPI liver feature detector is investigated as well as the two-level filter for tracking. The tracking algorithm is then adapted to other scenarios of medical image data: 2D ultrasound image data of liver motion and 3D EPI images of prostate deformation.

The two-level filter is implemented using node based appearance models. For the two-level model this mostly implies that the root appearance model may be specified more independently from the leaf models. Initial input to the algorithm is given by a set of positions specified in a reference frame. These determine locations of landmarks to be used for tracking and position estimates to be obtained for. For each of these landmarks, a local appearance model is generated by the sparse, local gradient based approach presented in *Sparse, Gradient Based Description* (3.1.2.2). Due to their restricted, local range these are not strongly affected by global intensity changes. For the leaf level, the obtained descriptions are used as they are. For the global model, the procedure is repeated — possibly using different parameters, such as wider regions — and the obtained models are joined into a single. This is done by joining their sets of positions and their sets of weights. The transformation center of the root model is chosen as the mean of the input positions.

The core algorithms itself were implemented in *C++* and controlled via a *MeVisLab (MeVis Medical Solutions, Germany)* module using *Python* scripts. Code was executed on a standard PC with an *Intel Core i7* processor 3.4GHz and 32GB of RAM on *Windows 7 64bit* in a single thread.

3.2.1. Simple Breathing Motion Model

A two dimensional motion pattern is defined, in which a global component is driven by a single phase parameter. The model's scaling, translation and rotation parameter are driven by the $\cos^4$ model as described for the translation component of breathing motion by Lujan [Luj+99]. Without perturbations, all landmarks undergo the global transformation. Driving term $\alpha_k = \cos^4(\pi k/T_F)$ with $k = [0..k_{max}]$ alters the elements of the global reference transformation simultaneously:

$$M_k^G = M_{TY}^G(A_T\alpha_k)M_S^G(1 + A_S\alpha_k)M_R^G(A_R\alpha_k) \tag{3.22}$$

The global translation M_{TY}^G, scaling M_S^G and rotation M_R^G transformations are defined with respect to center of the global model in the initial situation $k = 0$. Translation is chosen in the y-direction, scaling given as an isotropic scaling in the local xy-plane and rotation is about the through-plane z-axis. Aside from a perfect global motion pattern, there will exist components of motion which do not abide by the global scheme. Most of these may be handled on a local scale as they only pose small, localized deviations from the pattern. However, it is also possible that certain parts of the model belong to another, contradicting 'global' motion pattern, for instance a permanently static component. Influence of these kinds of perturbations to the idealized global model is investigated in simulation. For the simulation of artificial motion data, artificial Gaussian likelihood functions are used (*Algorithm and Experiment* (2.1.1)). The measurement function of single landmarks is given as $\mathcal{L}(x, \sigma_A)$, sampled by a point sample at the transformed location x in image space. Measurement of the global model is a sum of measurements at landmark positions transformed by the global model, in order to soften constraints in the presence of contradicting landmarks. Perturbations in the observation process lead to identical landmarks not being equally well observable, up to the point of single landmarks not being observable at all. The typical appearance width σ_A may change or the likelihood function may artificially be discretized by voxels. A reference parametrization is defined. The parameters are chosen similar to those which will be observed in an MRI guided breathing motion experiment, with the goal of a realistic setup. This implies, that the reference parametrization is not the one to achieve best performance in any setting. For instance, the chosen, realistic frame rate will be limited and therefore cause measurable error. Parameters are chosen as follows:

A set of landmarks is to be transformed, for instance representing a set of observable vessel cross sections visible in sagittal liver images. Their layout is depicted in (Fig. 3.12). It is defined for the exhale state and consists of twelve landmarks. These are arranged on a triangular grid, such that neighboring landmarks are 40 mm apart from each other. The grid is placed within a rectangular region of 120 mm by 120 mm in inhale position. During motion this region is transformed about its center position.

A series of two breathing cycles is to be tracked with update intervals of T_F seconds per frame. Given frame rate $1/T_F$, the number of frames $k_{max} = N_C \lceil T_{cycle}/T_F \rceil$ is chosen as dependent variable in each experiment. Here, T_{cycle} is the period of each of a breathing cycle. For the sake of simplicity $T_{cycle} = 1$, which leaves the interpretation of $1/T_F$ to be the frame rate per breathing cycle. With respect to physiology, amplitudes for the reference are chosen large $A_T = 40\,\text{mm}$, $A_S = 20\%$ and $A_R = 5°$. Joining both unperturbed transformation and observation, total likelihood landscape or artificial image is computed as

$$I_k(\vec{r}) = \sum_{i=1}^{N_{LM}} B\left(\vec{r} - T_k^G \vec{r}_i^{GL} - \vec{p}_{k,i}\right). \tag{3.23}$$

Here, B indicates the weight which can directly be read from image from the transformed image position. The indexes k and i indicate the temporal step and the landmark, respectively. Perturbations are introduced with respect to this definition. In experiments,

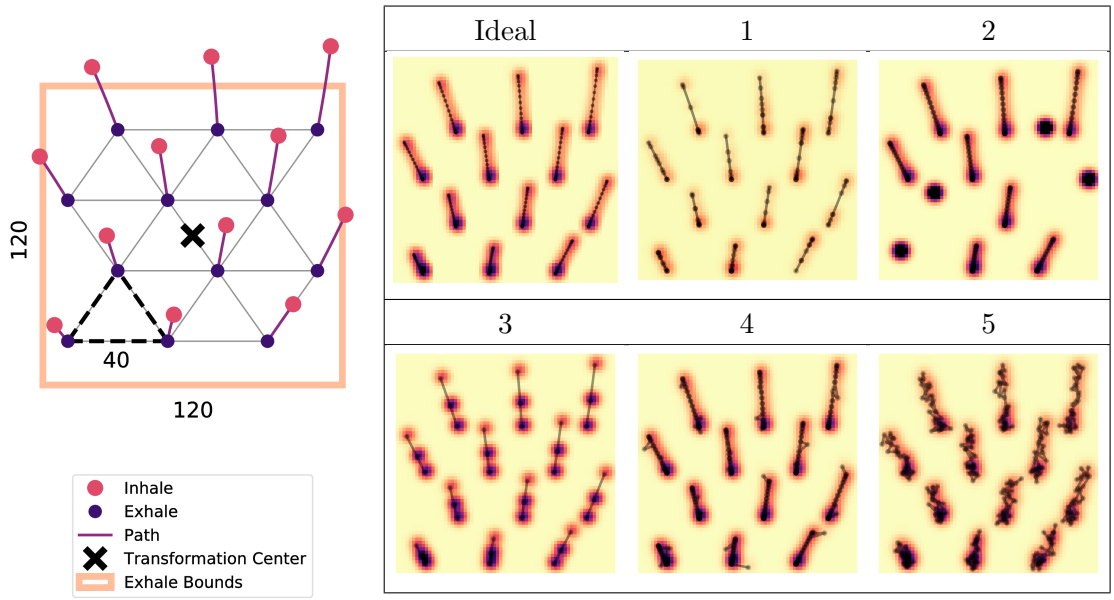


Figure 3.12.: Left: Reference layout for the landmarks for the artificial motion model, defined by a regular triangular grid in the exhale state. The region and transformation center is marked by X. Motion paths and inhale state are drawn for scale reference (with $A_T = 20\,\text{mm}, A_S = 0.2, A_R = 5°$). Right: Test scenarios as perturbations of the ideal motion: Dotted black lines indicate actual motion of the single landmarks throughout the cycle. Image color, showing the superposition of all generated image of the sequence, encodes state density of the single landmarks. For detailed description, see text.

the reference frame ($k = 0$), which is used for initialization, is free of perturbations and identical in all scenarios.

Case 1. Static Motion Component A fixed size subset of landmarks permanently remains in the initial position instead of moving with the global model. This simulates a secondary and conflicting mode of motion. The test parameter is the size of the subset of static landmarks $|L_{static}| = 0..12$. In each simulation this subset L_{static} is randomly drawn from all landmarks L. The global transformation is replaced as $T_{i,k} = T_0^G$ if $l_i \in L_{static}$, otherwise $T_{i,k} = T_k^G$.

Case 2. Reduced Observability is simulated by restricting the set of visible landmarks to a fixed size, random subset for each frame. The global B_0 is replaced by an individual $B_{k,i} \in \{0, B_0\}$ for invisible and visible landmarks, respectively.

Case 3. Frame Rate Dependency For the varying frame rate scenario the cycle is sampled at different frame durations $T_F/T_{cycle} = 0.01..0.3$.

Case 4. Single Deviation From Global Model In the case of a single landmark deviating from the globally observed motion pattern, in each frame a single landmark $l_{k,j}$ is chosen randomly and a deviation of *fixed distance* $R_{dev} = 0..10$ and random direction $\alpha_k =$

$\mathcal{U}(0, 2\pi)$ in the 2D plane is added to the transformation.

$$\vec{p_{k,i}} = \delta_{i,j} R_{dev} \left(\sin(\alpha_k), \cos(\alpha_k)\right)^T \tag{3.24}$$

Case 5. Joint Deviation From Global Model The case of local deviations from the global motion pattern is simulated by adding random spatial Gaussian noise for each landmark and frame with values $\sigma_{loc} = 0..10$.

$$\vec{p_{k,i}} = \mathcal{N}(\vec{0}, \mathrm{diag}(\sigma_{loc}^2)) \tag{3.25}$$

A single breathing cycle is used in a test scenario and each scenario is tested independently of the others. Tracked positions are compared to the known reference. Mean, maximum and median error are computed and averaged over 100 repetitions. All measures are obtained for the desired and available positions only — meaning that static and temporarily hidden sites are excluded.

3.2.2. Sparse, Gradient Based Landmark Model

Besides the transformation model, the appearance model is to be tested. Test structures are chosen as elliptic spots of intensity with count, angle and eccentricity drawn from uniform distributions between provided ranges for each subset of the test data. The density is chosen such that sufficient spots fall into the initialization region. Angles are drawn from $[0, 2\pi]$. The intensity profile is negative quadratic ($I_0(1 - x^2)$, where x is linearly interpolated from $x = 0$ in the ellipsis' center and $x = 1$ on its outline).

Within the image, an appearance model is initialized. Centered at an initial position $\vec{r_0}$, the spatial representation is initialized from all voxels within an initial radius of typically r =20 vx. These voxels with position $(\vec{p} - \vec{r_0})^2 < r^2$ define the set $\{\vec{p_i}\}$ of initialization positions. Once the appearance model is initialized from this set, it provides the response function $\mathcal{L}()$. A spatial response of the model to an image is computed by evaluating the appearance model on the set of initial positions $R = \mathcal{L}(\{\vec{p_i}\})$ — comparable to a 2D correlation function, limited to the initialization area. Responses are evaluated (Fig. 3.13) on the initial image (R_0) and the initial image, perturbed by Gaussian noise (R_1). The perturbed image is computed as $I_{noise}(x, y) = \max\left(0, I_0(x, y) + \mathcal{N}(0, \sigma_{noise}^2)\right)$. Value σ_{noise} is use to drive the noise level and the resulting empirical signal-to-noise is estimated as SNR= $E(I_0)/\sqrt{\mathrm{Var}(I_{noise} - I_0)}$. These responses R_0,R_1 are treated as posterior distributions generated by the likelihood function on a flat prior within the initialization region. Hence, R_0 and R_1 are normalized before being evaluated further. A measure of performance of a single response is given in the Survival Diagnostic (Eq. 1.37), which is $M_{SD}(R) = \left(\sum_{x,y} R^2(x, y)\right)^{-1}$. The similarity of the spatial responses to one another (perturbed and unperturbed) is computed using Pearson's correlation (Eq. 1.3), as $M_{PC}(R_0, R_1) = \mathrm{cov}(R_0, R_1)/(\mathrm{var}(R_0)\,\mathrm{var}(R_1))$.

Aside from paired reduction, two alternative single site reduction functions are chosen for comparison. Their single term energy $|\epsilon_i|$ is either the intensity itself, or the response to a narrow Marr-Hildreth kernel filter [MH80]. The appearance model's weighting

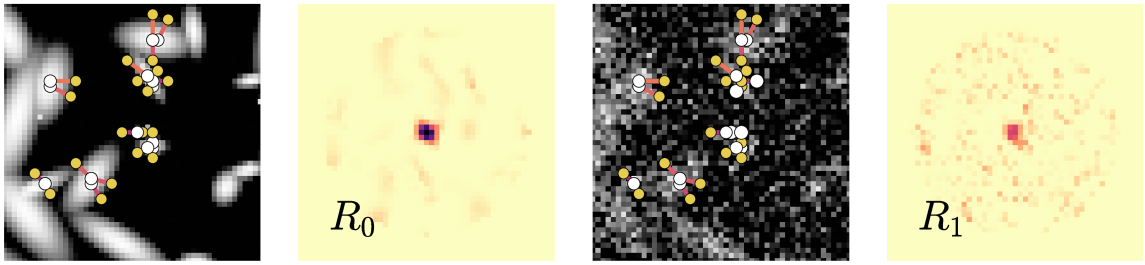


Figure 3.13.: (A) Initial image and model, (B) Response R_0 of the model (B) to the initial image, (C) Noisy copy of the image (SNR = 0.85), (D) Response R_1 of the model to the noisy image.

function is altered for comparison. The sum-of-squared-distances measure (Tab. 3.1) is used as alternative, dissimilarity based function. The exponent $\beta_{SSD} \approx 5 \cdot 10^{-4}$ is chosen sufficiently high to suppress a non-negligible response in positions of obvious disagreement[2] of the transformed model and image content, but no further. This is done such that on a linear scale, a response offset to unlikely regions is visibly brought close to zero. Influence of the two major parameters $\gamma = -0.3..1.0$ and the number of points $N_P = 10..143$ integrated into the model on the performance on the static test data set is evaluated with respect to the presented measures. Optimization (Eq. 3.16) of the entity selection is executed as a Simulated Annealing process and as a greedy computation. In Simulated Annealing with 10000 steps and linearly increasing[3] β, each alternation of the configuration a is computed by exchanging two elements in the set by random alternatives. A new configuration b is accepted certainly if $E(b) < E(a)$ or with probability $\exp(-\beta(E(b) - E(a)))$ in the opposed case $E(b) > E(a)$. The greedy algorithm on the other hand, selects the best single energy pairs in each step. It does so by neglecting those within the limit range of already selected ones.

3.2.3. 2D MRI Data

Data A volunteer data set of 10 2D EPI liver motion data [Tan+16] is used to evaluate the performance of the proposed tracking algorithm (Fig. 3.14). The data consist of sagittal 2D EPI images series showing motion of liver and kidney of 10 volunteers. These total a number of 120 series with 300 frames each. The voxel size is 1.875 mm in plane with a slice thickness of 6 mm which are typical values for real-time thermometry imaging with EPI. The image repetition time is 144ms. Due to saturation effects in the EPI imaging process, rather than the first, the tenth image is used as reference. At the tenth image, the global intensity level has decayed to a stable steady state and intensity levels in proceeding frames are comparable. Expert annotations of liver blood vessels were known in four sets and these were used for training. Random frames are annotated

[2] The region further away from the desired likelihood peak, in a noise free image.
[3] $\beta = \beta_0 t_{current}/t_{max}$

with around 15 landmarks. The remaining annotation for six test sets was not available to the author.

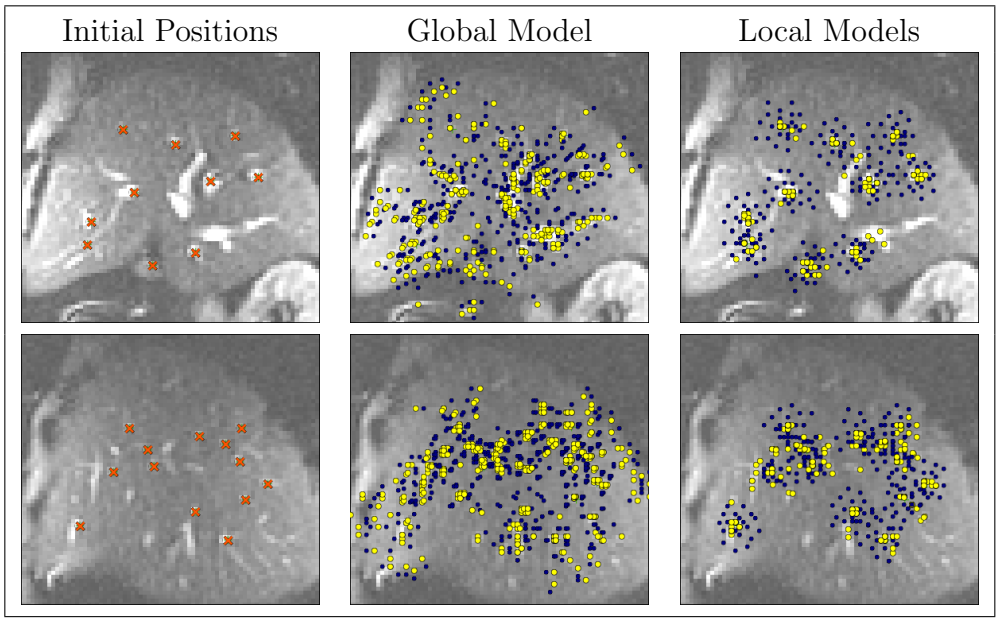


Figure 3.14.: Initialization samples of the appearance model on EPI slices of liver data with small (top) and large (bottom) vessels. Global and local models are depicted by their bright and dark components (yellow and blue respectively).

Feature Detector In order to asses the performance of the feature detector presented in *EPI Liver Feature Detector* (3.1.3), the reference annotation of the MRI dataset is used. A local maximum detector is compared to the proposed differential local binary pattern with an 8-neighborhood. The detector results are compared to the manual annotation of 1200 reference landmarks: Result of running a feature detector is a ranked list of positions[4]. The list is compared to the list of reference positions. For each reference position, the closest detection is selected. The distance between ground truth and detection is measured, and the rank of the detection is stored. A good detection result is characterized by low residual distances between detection and annotation and high ranks of the related detection. For different detector kernel sizes, rank and distance distribution are investigated in order to decide for an appropriate scale.

Tracking The two level model is initialized by defining contrast based appearance models for local and global stages (Fig. 3.14). Local appearance models are generated individually. For the global model, another set of local models is generated with different parameters. Their appearance is then joined to form the final global model. Evaluation of results for the test sets was carried out by a third party. Input to the

[4]As described above, candidate voxels are ranked by the initial intensity for both detectors.

evaluation were tracking results as position estimates with related standard deviations. A scalar deviation was computed as isotropic standard deviation, based on the squared 2D distances to the mean position. The evaluation was carried out once with the full set of landmarks. In a second evaluation, only landmarks with standard deviation below a threshold were included.

For the non publicly available expert annotation, sizes of the local appearance model are chosen 15 mm which is approximately the size of vessels. The larger patches which are joined to the global model are chosen 35 mm. With $\sigma_T^L = 2\,\text{mm}$ and $\sigma_T^G = 8\,\text{mm}$ local and global translatory motion scales are chosen in a comparable ratio. Scaling ($\sigma_S^G = 0.01$) and rotation ($\sigma_R^G = 0.001$) parameters of the global model are chose to model slow changes. Global and local sample counts are chosen 500 and 100. Annealing schedules with linearly increasing exponent $\Lambda = \{0.6, 1.2, 1.8\}$ and $\Lambda = \{0.6, 1.2\}$ were chosen for the observation iterations. Global and local appearance models are initialized with radii 35 mm, 15 mm, minimal point distances 4.5 mm, 2.5 mm and point counts 35 and 20, respectively.

For a further, general evaluation, annotations for the training sets were added manually by a non-trained expert. For this, the test set data was annotated in random frames. This includes frames where not all landmarks' positions are clearly determined. This annotation was used to investigate the difference between the two stages of the algorithm. For the second experiment, the parametrization of the algorithm differs slightly. This is related to both being conducted with temporal difference and the range of algorithmic options increased. For instance, setting (1) uses three global iterations with fixed diffusion of 8 mm standard deviation using 500 samples. Setting (2) replaces these by five iterations of adaptive diffusion of 3 mm using only 100 samples. Furthermore, parametrization offers more degrees of freedom for leaf level transformations and adds explicit limits for root level scaling and rotation. The dynamic settings are summarized in table 3.2.

3.2.4. 2D US Data

Data The two stage hierarchical tracking scheme was also applied to two dimensional ultrasound data. The algorithm was evaluated on the 2D data of the challenge on liver ultrasound tracking 2014 [Luc+15], against an annotation generated by non-medical-expert colleagues at Fraunhofer MEVIS.

A dataset consists of a series of 2D images. For the first image — being the reference — a set of (1 to 5) landmarks is given in voxel coordinates. An annotation denotes their voxel position in select proceeding frames. The tracking algorithm produces position estimates for all given landmarks and all frames. In frames with available annotation, the lateral and axial voxel distance (d_x,d_y) between position estimates and annotation is computed and converted to a distance error $\epsilon = \sqrt{(d_x V_y)^2 + (d_y V_y)^2}$ in mm, where V_x and V_y denote lateral and axial voxel extent in mm, respectively. Mean errors are computed over the related annotation as averages per landmark, or average of all landmarks.

	(1) Experiment with official annotation		(2) Detail experiment with alternative annotation	
	Root	Leaf	Root	Leaf
T_X	8.0 mm	2.0 mm	3.0 mm	2.0 mm
T_Y	8.0 mm	2.0 mm	3.0 mm	2.0 mm
S_X	0.01	0	0.002 (0.9, 1.1)	0.01
S_Y	0.01	0	0.002 (0.8, 1.2)	0.1
R_Z	0.001	0	0.01 (-0.05, 0.05)	0.0
Prior Diff.	Temporal Static	Hierarchical Static	Temporal Adaptive	Hierarchical Static
N_S	500	100	100	100
Λ	$\{0.6, 1.2, 1.8\}$	$\{0.6, 1.2\}$	$\{2, 4, 6, 8, 10\}$	$\{1, 2, 3\}$
Obs R	35 mm	15 mm	28 mm	15 mm
Reduction	paired (8 pairs)	paired (30 pairs)	paired (8 pairs)	paired (30 pairs)
Weighting	contrast	contrast	contrast	contrast

Table 3.2.: Settings of the two level algorithm, applied to liver MRI data.

Input : Reference positions, tracked position estimates indexed by landmark ID, time and level; voxelsize

Parameters: $\{\vec{p^{ref}_{i,t}}\}, \{\vec{p^{track}_{i,t,l}}\}$, $i = 1..L, t = 1..t_{max}, l = 1..L_{max}$, Voxel size as vector $\vec{V}$

Result: Landmark, time and level specific absolute distances $d_{i,t,l}$ in mm

for each element $(t_0, i_0, \vec{p^{ref}})$ in $\{\vec{p^{ref}_{i,t}}\}$ **do**

  for each element $(t_1, i_1, l, \vec{p_{track}})$ in $\{\vec{p^{track}_{i,t,l}}\}$ **do**

    if $t_0 = t_1$ & $i_0 = 1$ **then**

      Compute distance vector $\vec{D} \leftarrow \vec{p^{track}} - \vec{p^{ref}}$

      ... scalar distance $d^{vx}_{i_0,t_0,l} \leftarrow |\vec{D}|_2$ in voxel

      ... scalar distance $d^{mm}_{i_0,t_0,i_l} \leftarrow (D_x^2 V_x^2 + D_y^2 V_y^2 + D_z^2 V_z^2)^{1/2}$ in mm

Algorithm 3.2: Computation of Result Measures for Evaluation

Preprocessing The data requires preprocessing to make it approachable by the tracking algorithm. These steps include the generation of mask images and filtering of the reference image.

In order to guide the tracking algorithm towards the relevant sections of ultrasound images, masks are generated by applying thresholding and morphological filters. The reference image is thresholded by its lowest value, which then is followed by dilation and erosion filters. This closes gaps in the mask. Masks are used both for filtering landmark seed points as well as positions used to initialize the landmarks' appearance models.

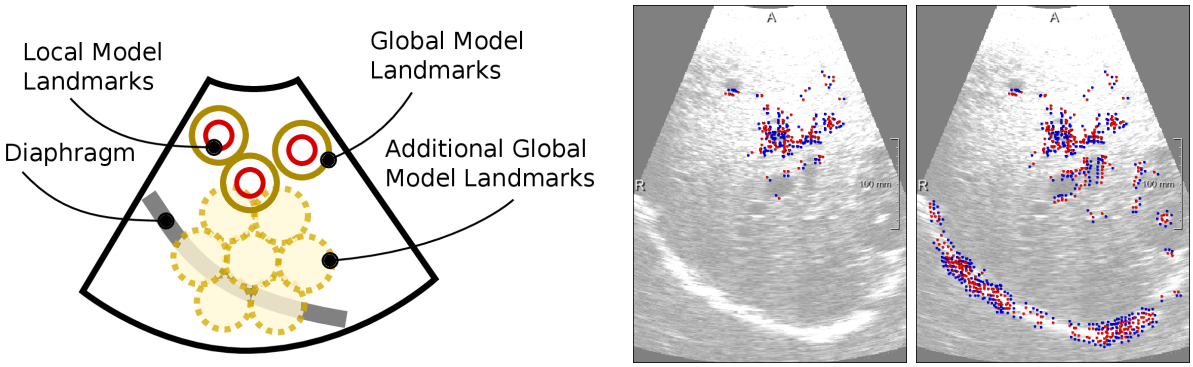


Figure 3.15.: Left: Landmark input for local and global model with and without additional diaphragm landmarks as schematics. Right: Global model generated in actual data without (left) and with (right) additional landmarks. Local models are identical in both cases and not depicted. It is to be noted that the shown ultrasound imagery is acquired with the transducer below the rib cage, facing upwards. Therefore the diaphragm, which anatomically lies above the liver, shows below it.

For a contrast based appearance model, one is interested in observing large scale gradients rather than single ultrasound speckles. Therefore, the reference image is preprocessed using a median filter, which suppresses minima between speckles. Also, ultrasound images show a non-isotropic point spread function for lateral and radial direction. Furthermore, as a result of direct reflection, gradients in axial direction appear stronger. Lateral point spread is often stronger than axial. In order to account for an imbalance in these gradients, the kernel of the median filter is chosen non-isotropic (15 axial×5 lateral voxels). This preprocessing step shapes the input for the building the appearance model and is only required for the initial image.

Tracking For tracking, the dynamic settings are set as depicted in table 3.3. Two modes of application are explored. In *plain application* of the two level model, only given landmarks are joined into the global appearance model. After matching the global model, the local models are optimized given the global result.

Compared to EPI data, fewer vessel landmarks are visible and annotated (3-5) in the data. Therefore, some landmarks are added to justify a global model and make use of the well visible diaphragm region. *Addition of landmarks into the lower image*

region can force the identification of the diaphragm (Fig. 3.15). Center for a triangular grid of seven additional landmarks is chosen in relative units in the lower image region $(x, y) = (0.5, 0.6)$, with relative distances of 0.25. The additional landmarks are only integrated into the global appearance model and not represented as additional leaf nodes.

	Root	Leaf
T_X	3.0 mm	2.0 mm
T_Y	3.0 mm	2.0 mm
S_X	0.002 (0.9,1.1)	0.01 (0.8, 1.2)
S_Y	0.002 (0.8,1.2)	0.01 (0.8, 1.2)
R_Z	0.01 (-0.05,0.05)	0.0
Prior	Temporal	Hierarchical
Diff.	Adaptive	Static
N_S	600	500
N_I	$3(\Lambda \in \{4, 8, 12\})$	$3(\Lambda \in \{4, 8, 12\})$
Obs R	0.1 img. size	25 vx
Reduction	paired (60 pairs)	paired (30 pairs)
Weighting	contrast	contrast

Table 3.3.: Settings of the two level algorithm, applied to 2D liver ultrasound data.

Due to the variability in the appearance in ultrasound data, different contrast based appearance models are explored (Fig. 3.16). The dot product kernel (Eq. 3.19) of the contrast based model multiplies weights w_i with intensity measurements $I(T\vec{p_i})$. For both factors an alternative is explored. In the reference case, the set of reduced voxel positions is assigned integer weights ± 1, which is an approximation for dividing regions into dark and bright areas. As an alternative, these values are chosen proportional to the voxel intensity within the range of all select intensities as in [Rot+14], thus assigning a soft weight $\in [-1, 1]$. An alternative to the measured intensity is gained by reducing the measured intensity values by their mean.

3.2.5. 3D MRI Data

Data Three dimensional EPI of prostate motion or deformation during FUS treatment is investigated. The data features multi slice 3D MRI data, with higher in-plane than through plane voxel resolution (e.g. 1.6 mm ×1.6 mm ×6 mm). Slice alignment is axial. The algorithm is provided with meshes of the current location of the prostate capsule and possibly urethra and rectal wall, as these pose locations have to be avoided during treatment. Three dimensional images are acquired regularly during the procedure in order to confirm the correct positioning and to verify a baseline for thermometry. Updates are much less frequent than in 2D liver motion examples. Updates appear irregular, in

	'Dark'/'Bright' weights w_i from Reference Image	Measurement intensity in Current Image	
Option 1 (Reference)	$+1$ -1	$I(T\vec{p}_i)$	with $I^{min} = \min_i(I(T\vec{p}_i))$ $I^{max} = \max_i(I(T\vec{p}_i))$ $I^{mean} = N_P^{-1} \sum_{i=1}^{N_P} (I(T\vec{p}_i))$
Option 2 (Alternative)	$+(I(T\vec{p}_i) - I^{min})/(I^{max} - I^{min})$ $-(I^{max} - I(T\vec{p}_i))/(I^{max} - I^{min})$	$I(T\vec{p}_i) - I^{mean}$	

Figure 3.16.: Alternative initialization of appearance model weights and alternative measurement of the intensity. In the contrast based appearance model, initialized weight and measured intensity are joined in a product kernel (Eq. 3.19).

the order of few minutes, as confirmation scans in between steps of the application of focused ultrasound. Task of the tracking component is to detect whether and how the prostate changed location.

In twenty three data sets, an annotation of three structure types is used to determine the validity of the algorithm (Fig. 3.17). The structures types are prostate capsule, urethra and rectal wall. The annotation is generated manually by a non-trained expert and provides a deformation vector for select positions, indicating the motion from reference to the related current image. The tracking result is then used to estimate a deformation vector for the annotated reference position. Then, estimated and annotated deformation are compared. Their tip distance is chosen as error measure and the length of the annotated vector is counted as reference motion, which is to be compensated.

Model Input to the algorithm are three dimensional EPI MRI images. In the reference, the regions of interest are provided as meshes. Motion information is extracted as a three dimensional sparse vector field. The global transformation allows translation and scaling, shearing and rotation in axial (x,y) plane. Transformations leading to motion through the axial plane are neglected, since the voxel resolution in z direction is low. Furthermore, it is much more challenging to account for a full three dimensional transformation when annotating a dataset. Local transformations are given by x,y translations within the global prior coordinate system. In this setup all deformations appear in the axial planes of the 3D volume. The configuration of the parameters is summarized in table 3.4.

Reference meshes are used to derive the target region by mapping the interior of the capsule mesh to the image region. Furthermore, meshes are down sampled. The resulting coarser meshes' edge points are used as seeds for landmarks. These seeds reside on the outside of the related image region. Seeds within the region are generated by adding a regular grid inside the mask area. The grid is chosen as a regular triangular grid within each axial slice. The initial positions provided to the tracking algorithm are therefore mesh edges and volume grid positions.

Appearance models are generated differently on global and local level. The global stage uses a sparse edge description, whereas fully initialized image patches with SSD based weighting define local patches. This speeds up global detection and leaves refinement to the local stage. Since transformations allow motion in plane, local appearance models within each plane are two dimensional. The global transformation model, joining all local models is three dimensional.

During tracking, the global stage uses multiple iterations with adaptive diffusion to settle into an optimum position as a mean result. Local models use this optimum's transformation to base their relative motion on. On the local level, a limited number of iterations is used and fixed translation diffusion values of 1 mm in (x,y) direction are applied.

Figure 3.17 depicts a sample case. In the left image, prostate with urethra in the center are annotated in reference position (yellow). In current images, the prostate has moved towards the bottom left corner of the image (green boundary). The center image shows a result based on the global deformation result, the right one based on the local results (yellow arrows). The local deformation moves the urethra position further to the true location. Furthermore, the top right boundary moves closer to the true location with the local correction.

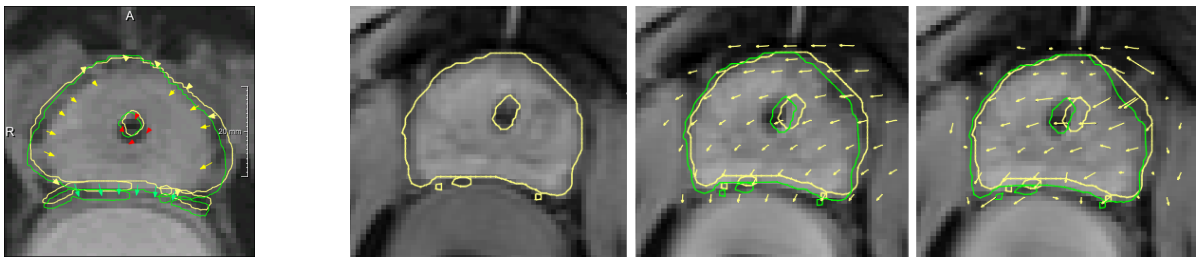


Figure 3.17.: Left: Deformation (contours) and annotation (vectors) in three regions (capsule, urethra, rectal wall) Right: Deformation detection: Reference image, global and local deformation detected on current image. Yellow and green outline represent the reference mesh and the current, deformed mesh respectively.

	Root	Leaf
T_X, T_Y	2.0 mm	1.0 mm
S_X, S_Y	0.1, 0.1	-
R_Z	0.01	-
Sh_{XY}	0.05	-
Prior	Temporal	Hierarchical
Diff.	Adaptive	Static
N_S	100	500
N_I	$4(\Lambda \in \{2, 4, 6, 8\})$	1
Obs R	10 vx	3 vx
Reduction	paired (10 pairs)	none
Weighting	contrast	SSD

Table 3.4.: Settings of the two level algorithm, applied to liver MRI data. (Sh_{XY} indicates shearing in XY plane.)

3.3. Results

3.3.1. Simple Breathing Motion Model

The effect of different perturbations to a large scale default motion are investigated in proposed scenarios (Fig. 3.18). For all values of the perturbations, spatial errors are computed which indicate deterioration of the results. Perturbation are introduced with respect to the 12 landmark reference case. Simulation results are averaged over ten executions and computed for both the case of static diffusion and adaptive diffusion.

Figure 3.18 depicts the influence of changes of landmarks present in the moving component. (A) depicts the dependence of mean error measures on the number of landmarks present in the model. Error values are stable for landmark numbers of five and more. With four landmarks the error measure begins to deteriorate. The error measure worsens drastically when using adaptive diffusion, but in case of static diffusion is limited. A subset of landmarks of the 12 landmark-model is held static in (B). If this subset is larger or equal to half the number of landmarks (6 in this case), the track is lost. The track can be kept with stable error measure for four of twelve landmarks and begins to deteriorate for five. In case of success, error measures are lower for the adaptive diffusion model. In (C) only a random subset of the twelve landmarks is visible in each frame after the first. Showing less than four landmarks in each frame leads to loss of the track. With four and more landmarks the motion can be followed. Then, errors for adaptive diffusion are lower than for static. They also remain more stable as the number of visible landmarks is decreased.

Figure 3.18 shows errors introduced by scaling and noise. For (D), the experiment is scaled spatially, which leads to a linear increase in error measures, if the observation is continuous. In order to simulate a discretization, the likelihood function is only sampled

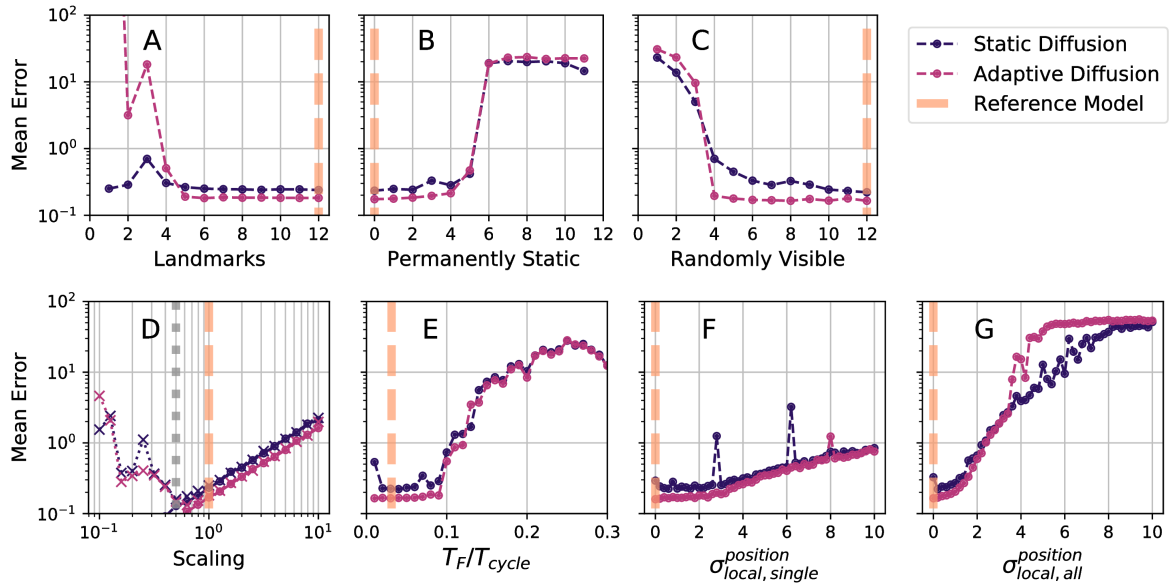


Figure 3.18.: Top: Error measures depending on the number of present / static and visible landmarks. Bottom: Error measures depending on spatial and temporal changes and spatial noise. Plots B,C,E,F and G correspond to the described perturbation cases 1-5, respectively.

at integer positions in space, by rounding the continuous position to the next integer. Then, once the experiment's scaling reaches $\approx 0.5 = 1/\sigma_A$ (dotted gray vertical), tracks are lost.

In (E) the frame rate T_F is scanned as ratio of the breathing cycle T_{cycle} duration. Error measures are stable up to $T_F/T_{cycle} \approx 0.09$, which for a breathing cycle of 4 s corresponds to 360 ms. For larger values, error measures start to deteriorate strongly.

Plots (F) and (G) show the increase of error with increased spatial uncertainty. Single local deviations (F) start to influence the mean error at standard deviations of ≈ 3, whereas local deviations of all landmarks (G) already affect the performance at standard deviation 1.

3.3.2. Sparse, Gradient Based Landmark Model

Different reduction and weighting schemes are examined at varying SNR levels. Figure 3.19 shows the drop in Pearson's correlation M_{PC} as the noise increases (landmark patch radius = 20 vx, point limit $N_P = 50$). For each algorithm the correlation M_{PC} at high SNR levels is very close to one. With decreasing SNR, these values drop. At SNR = 1 the drop in correlation has visibly started in all cases. In this regime, performances of the algorithms is separated by the point reduction routine. Point based routines reach mean correlation values between 0.85 and 0.90, whereas the point pair based initialization still reaches values above 0.95. With signal-to-noise ratio decreasing beyond 0.5 the main

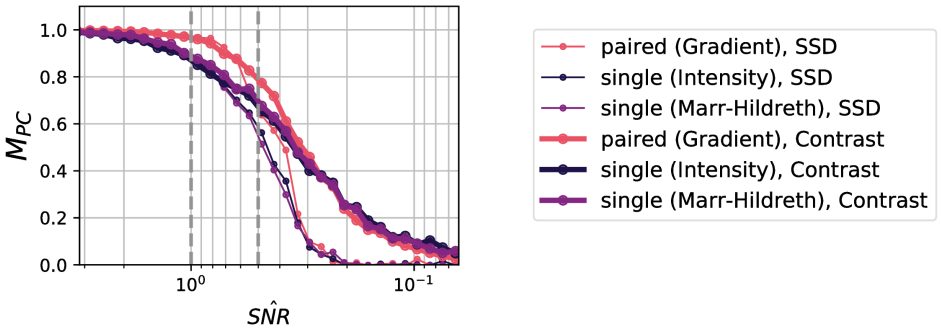


Figure 3.19.: Pearson's correlation of the response R_0 to the initial, noise free image to the response R_1 to the same image, perturbed by Gaussian noise.

falloff is determined by the weighting function. SSD-based weighting shows a shorter tail as compared to contrast based weighting.

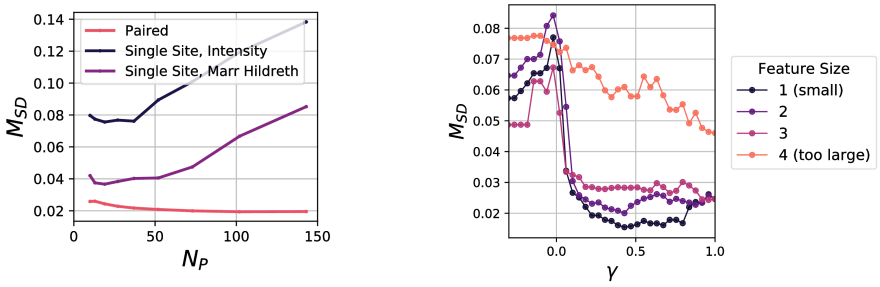


Figure 3.20.: Left:Survival Diagnostic dependent on point count, compared for different reduction schemes. Low M_{SD} indicates a specific model with low description error. Right: Drop of the Survival Diagnostic, as γ increases on symmetric artificial blob data.

Independent of the reduction strategy, with increasing number of points, the target description error will initially drop, as seen by the Survival Diagnostic M_{SD}(Fig. 3.20). With further increasing N_P this may however change. The error is reduced further for paired point selection. Beyond a certain number of points, the error of models reduced by single site selection increases again. In general, the Survival Diagnostic of single site reduction based on intensity is higher than based on the Marr-Hildreth filter and both are higher than that of paired reduction.

Effect of single pair weighting is studied on artificial test data set with random fixed size structures of eccentricity 0 ($a_0 = a_1$). The effect of parameter γ on the response to these images was evaluated by the Survival Diagnostic M_{SD} (Fig. 3.19). Here, for parameter values $\gamma \geq 0$ a drop in M_{SD} can be observed for increasing γ up until $\gamma \approx 0.3$. Smaller structures lead to smaller Survival Diagnostics. Structures larger than the maximum pair length D do not show a clear drop in M_{SD}. In this case, both sites of a gradient point pair are already located on the same gradient slope.

Equation 3.16 was approximated via a greedy algorithm or Simulated Annealing. Comparing Pearson's correlation graphs as in figure 3.19, only a slight advantage is visible (not depicted). With Simulated Annealing, mean correlation values are approximately 1% higher, along the function's slope.

3.3.3. 2D MRI Data

EPI Liver Feature Detector A local maximum and an LBP based detector are applied to the reference MRI images. Figure 3.21 depicts the distribution of ranks and distances for different kernel sizes of the LBP based detector. Low ranks and distances are favorable. A majority of samples achieves distances below $\approx 3\,\text{mm}$ and ranks below 40. A more sparse distribution of "outliers" exists with much larger distances and ranks. The 50^{th} and 90^{th} percentile of the marginal distributions along distance and rank axes are highlighted.

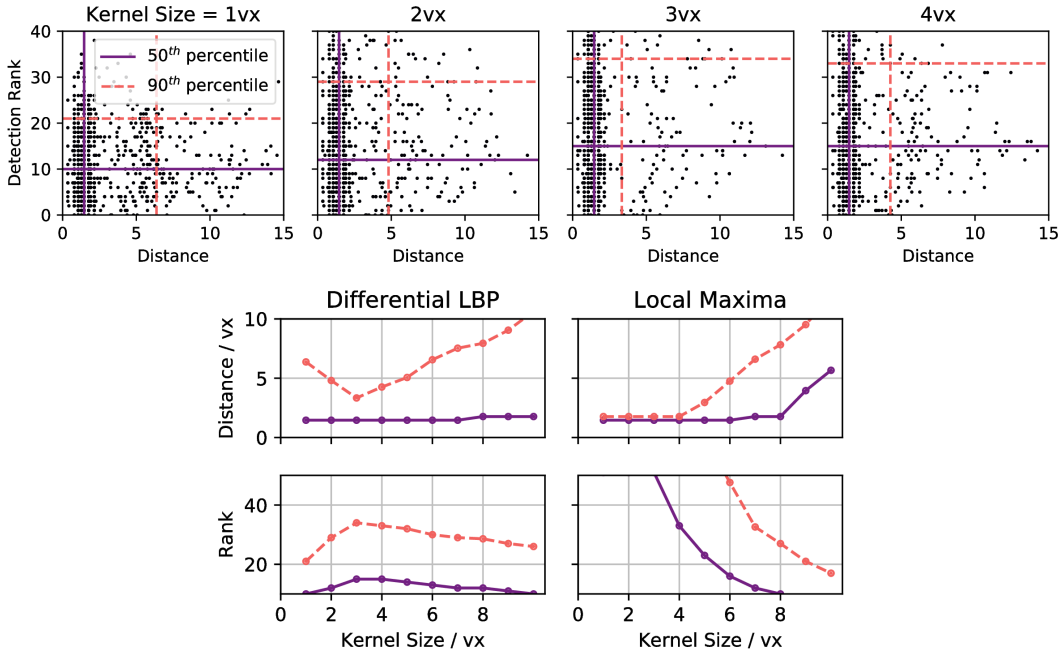


Figure 3.21.: Top: Exemplary scatter plots of the results of detections of the 1200 ground truth landmarks by the proposed LBP based kernel filter with different sizes. Bottom: Distances between ground truth position and feature detection for different kernel sizes, 50^{th} and 90^{th} percentiles.

Values of 50^{th} and 90^{th} percentile for both detectors and measures are shown in figure 3.21 in more detail.

For the LBP based measure, the 50^{th} percentile of distance error is mostly stable at 1.8 vx, the 90^{th} shows a clear minimum for a 3 vx kernel size. The depiction of ranks shows maxima in both percentile curves for the kernel size of 3 vx.

The maximum filter shows a different qualitative result. For small kernel sizes, detections are closer than one voxel to annotations. They are, however, so dense that detections closest to the annotation reach very low ranks (> 50). Only for larger kernels ($>\approx$5 vx), the number of detections becomes sparse and ranks drop. In this region, the distance error however starts to increase.

The local maxima filter yields distance error comparable to the optimum value of the LBP based filter for 5 vx kernel size. However, the ranks of these detections are much lower than for the LBP based filter. Similarly, when the local maxima filter reaches a comparable rank (kernel size 7 vx), the error measure is more than twice the error of the LBP optimum.

Tracking In the test set the average motion of the landmarks and the ground truth with respect to the initial image is 5 mm–5.5 mm (table 3.5), with 95% of the motion remaining below 15 mm. The algorithm reaches a mean tracking error of 0.94 mm to the expert annotations on the test data set. Excluding uncertain landmarks with standard deviation above 5 mm reduced the number of valid landmarks from 3705 to 2657, which does not increase the performance significantly, but reduces the 95th percentile error from 2.07 mm to 2.03 mm. The mean error of the reduced set is 0.93 mm. With the given voxel size of 1.875 mm the algorithm reaches sub-voxel accuracy with respect to the mean error. With the presented parametrization the algorithm requires approximately 2.25 ms of processing time per landmark and frame without parallelization. Thus, with the test set models consisting of 4 to 17 landmarks it reaches real-time performance. The average error of 0.94 mm is an improvement of 0.16 mm with respect to previously [Tan+16] reported results on the same data.

Results	**All** (3705) / mm			**Only valid** (2657) / mm		
	Mean	Std.Dev.	95%	Mean	Std.Dev.	95%
LM Mo.	5.24	4.29	14.07	5.23	4.29	14.04
GT Mo.	5.39	4.30	13.99	5.38	4.29	13.96
Error	0.94	0.72	2.07	0.93	0.72	2.03

Table 3.5.: Results for motion tracking in EPI liver images against expert annotation to the test data set. Top: Absolute motion of the landmarks in annotated frames with respect to the reference frame. Center: Absolute motion of the annotations in the annotated frames with respect to the reference frame. Bottom: Residual absolute distance between tracking result and annotation in the annotated frames.

Data is further evaluated against a reference in which the non-available test set is replaced by a manual annotation (Fig. 3.22), in order to study the effect of global and local models. In all cases, the algorithm yields median error levels below typical voxel sizes in the images. Local error levels are found to be lower than global error levels. Outliers are attributable to few landmarks among each set, which the algorithm

performs on worse than the rest. These are in particular small liver vessels in the outer region of the liver. Error levels of the training sets (indices 0,1,5,6) are lower than those of the test sets, even though in this case the algorithm is not particularly optimized for the training set — the training sets are annotated by the alternative, non-expert annotation.

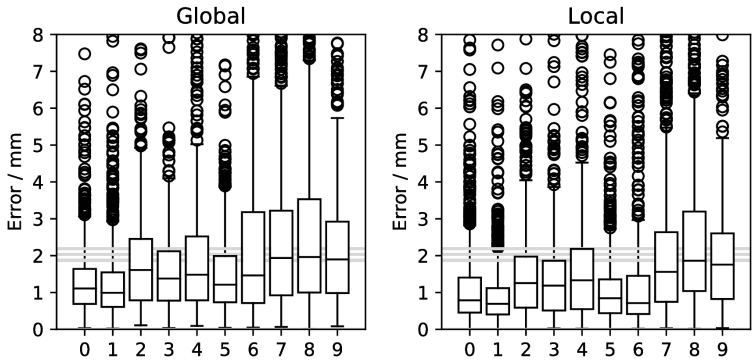


Figure 3.22.: 2D MRI: Sample results, comparing global and local errors of the ten series. Light gray lines indicate voxel sizes present in the datasets.

Due to relatively low resolution and the number of points or gradients used to describe local image patches can be reduced. With 8 gradient point pairs local and global models of vessels are sufficiently well defined to support a stable detection of the global motion. With such a parametrization, tracking result is obtained within 5.0ms on average and 9.6ms maximum.

3.3.4. 2D Ultrasound Image Data

The algorithm is parametrized as in (Fig. 3.3) and applied to the given dataset. Statistics of the obtained errors are depicted in (Fig. 3.23) for each landmark individually. The mean error of a landmark, averaged over all obtained error measures, is 2.27 mm on global and 1.86 mm on local level. Averages of average errors per landmark are 2.72 mm and 2.23 mm respectively.

Adding global landmarks into the diaphragm region changes these error levels to 2.32 mm and 1.87 mm or 3.01 mm and 2.48 mm respectively. In all cases the local optimization decreases error levels. The addition of landmarks slightly increases them.

For each landmark a global and local position estimate and resulting error measure exist in the obtained result data. While the local level is able to decrease error measure on average, it does not do so for every landmark and time step. Histogram (Fig. 3.23, left) shows a correlation between local and global error measures, for the whole dataset. Applying linear regression (Fig. 3.24) to the pairs of local stage ϵ^L and global stage ϵ^G error measures, the relation $\epsilon^L = 0.57\,\text{mm} + 0.58\epsilon^G$ is found (depicted as line), with Pearson's correlation coefficient of 0.67. This implies that equality is reached for $\epsilon^G = \epsilon^L \approx 1.38\,\text{mm}$. For $\epsilon^G < 1.38$, the local stage error ϵ^L is smaller then ϵ^G and

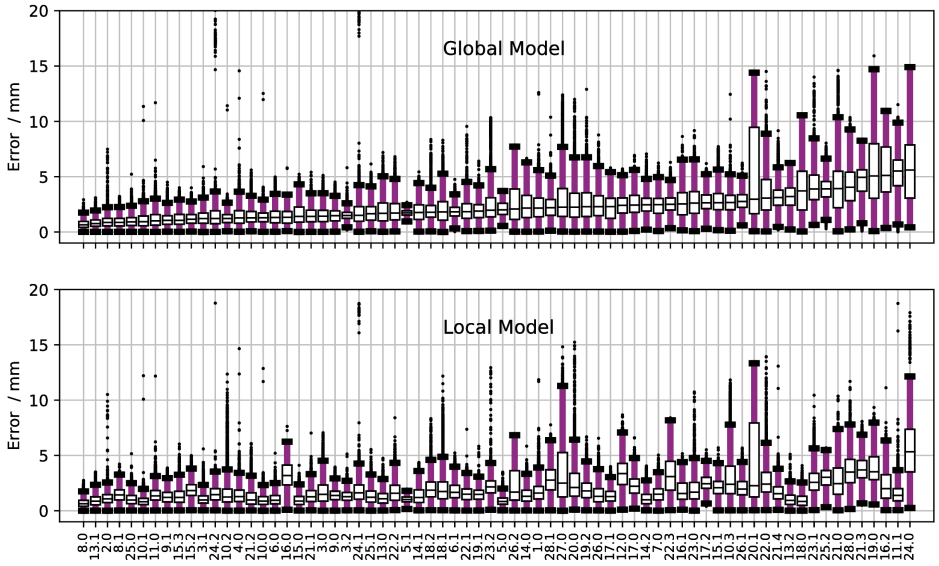


Figure 3.23.: 2D US: Application of to ultrasound image data. Results for individual landmarks obtained on global (top) and local stage (bottom). Values are sorted by the global model's median error.

vice versa. This effect is seen in more detail by computing the average local stage error for global stage error bins (overlaid graph). The mean local stage error does not fall below 0.9 mm, even for the $\epsilon^G \approx 0$ bin. On the other hand, for large ϵ^G, the mean local error lies below the linear regression. With seven additional root model landmarks in the diaphragm region, a qualitatively similar result is obtained. However, the slope of the linear regression is lowered in that ϵ^L =0.68 mm + $0.47\epsilon^G$ with correlation coefficient of 0.67. Equality is reached for $\epsilon^G = \epsilon^L \approx$1.28 mm. In the compared settings, the default case yields an average ϵ^G =2.23 mm and ϵ^L =1.86 mm. With additional landmarks, these are ϵ^G =2.48 mm and ϵ^L =1.87 mm. The reasons for the lowered slope when using additional landmarks is not found in a lowered local stage error. Instead, additional landmarks increase the global stage's error level, while local stage errors remain comparable.

Results described above are obtained for the default sequences, cycled frame by frame. Introducing large jumps of 200 frames per step, the temporal correlation between subsequent images is mostly lost. Error levels are raised in general (Fig. 3.24). They however show strong outliers in sequences without additional landmarks. These extreme outliers are widely suppressed when additional landmarks are installed. The model with additional landmarks shows error levels up to 18 mm, whereas the model without exhibits six outliers in the range of 18 mm – 45 mm.

The contrast based appearance model was altered by two changes: (1) Weights of the model are chosen proportional to initial intensity (2) The mean of the current intensity was removed before the application of the product kernel. With the chosen setup of using a median filter image as reference input, change (1) could not increase the performance

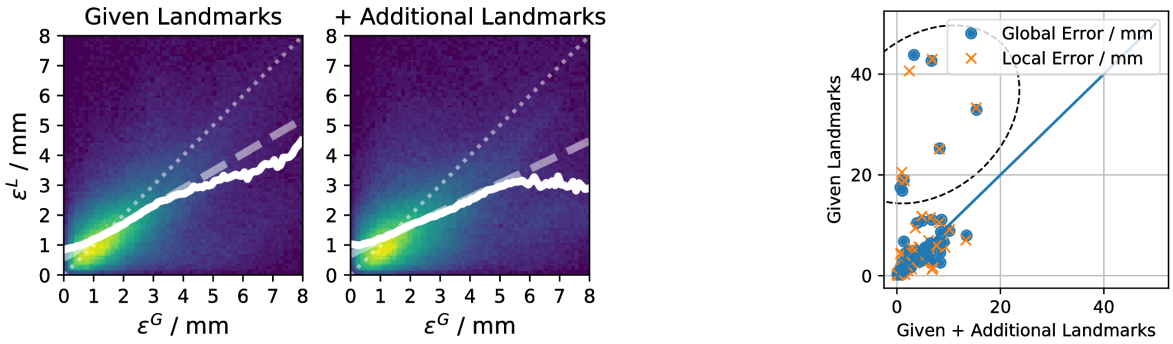


Figure 3.24.: 2D US: Left: Histogram of joint occurrences of global and local error measures ϵ^G, ϵ^L, linear regression thereof (straight line) and mean local error level (graph). Right: Error levels in an "uncorrelated jump" sequence of 200 frames step width, compared for tracking without (default) and with additional diaphragm landmarks. The default model produces outliers (encircled), from which the track cannot recover. With additional landmarks, these outliers do not appear.

and added strong outliers. Change (2) is found to improve performance of the global model, changing a mean error of 2.60 mm to 2.51 mm.

In plain application to the annotated data, a mean computation time of 29.5 ms is found for each dataset, whereas the mean computation time per landmark is 10.1 ms. The additional seven global landmarks in the diaphragm region introduce an offset in computation time, leading to an average 68.2 ms among datasets[5].

3.3.5. 3D MRI Data

For prostate motion detection, the motion component of the algorithm was parametrized as shown in (Tab. 3.4). Landmark errors in the 3D MRI data are compared separately (Fig. 3.25), by structure type. For each type, the figure relates reference motion to error measures obtained for the global and local level. For each time point the error mean for all structures and as per structure type is computed. The statistics of these errors over all annotated time points is shown. For all structures errors, the median of mean can be reduced from 2 mm (no correction) to 1.4 mm (global correction) and 1 mm (local correction). Capsule landmarks reduce median distance of 2.3 mm to a residual error of 1.1 mm and 1.0 mm, respectively. A similar reduction can be observed for urethra landmarks. The base error reduction by the global model is also present in the rectal wall landmarks. However, the local correction does not decrease the error further. It even slightly increases (median $\approx$ 1.4 mm) compared to the global reduction (median $\approx$ 1.2 mm).

[5]Since the additional landmarks are only integrated into the root model, a time-per-landmark of the two model cases would not be comparable and is not shown.

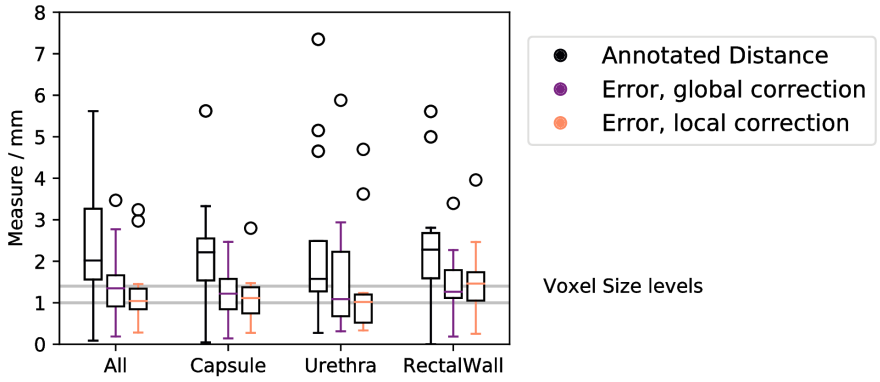


Figure 3.25.: 3D MRI: Statistics of mean error measures in single time points in all prostate datasets.

Most of the computation time is spent for the initialization phase. Mostly the process of ranking gradient point pairs in a 3D dataset leads to a processing time of a few (less than 5) seconds. The update of the tracking result, when provided with new current data can be executed in an average 920 ms with a standard deviation of 200 ms. These timings are obtained without parallelization.

3.4. Discussion

This chapter provided introducing experiment to tracking applications in medical image data. A simple hierarchical tree model and a sparse landmark representation are both investigated on artificial before being applied to medical image data.

3.4.1. Simple Breathing Motion Model

A hierarchical two-level motion model was used to provide ground truth data, which obeys the transformation the two-level filter was to track. Artificial perturbations were introduced to study the limitations.

The given reference model showed a stable parametrization plateau for wide ranges of imaging frame rates. Extending the frame time T_F by 200% (to ≈ 350 ms in a 4 s cycle) was still in the stable region. With the set global motion translation amplitude of 4 cm and even $\cos^2$ motion, this corresponds to peak velocity of $\approx 0.7 \cdot 4\,\text{cm} = 2.8\,\text{cm}$ per frame.

Introduction of spatial deviations of single landmarks lead to loss of the landmark, as soon as it exits the convergence range given by local diffusion values. This was caused by the local landmarks solely relying on the global hierarchical motion prior instead of the their temporal. If only few landmarks showed random spatial movement within the global component, the global motion detection was not affected. Once all landmarks

exhibited uncorrelated spatial deviations from the global motion, the track could be lost completely.

With increasing strength of the local deviations, the likelihood peak of the global motion spreads out. It decreases in strength and becomes multi modal. Tracking is sensitive to these perturbations and error estimates decrease strongly from spatial deviations from 1 mm and more. Hence, for a well functioning model it is required that there exists a set of landmarks which represents the common moving structure well. This set should outweigh the set of static landmarks and the set of landmarks with strong local displacements.

In order to successfully run a tracking algorithm in the given artificial scenario, the time per frame needs to be lower than a tenth of the breathing cycle. If the global model allows scaling and rotation, these degrees of freedom need to be bound by a minimum set of simultaneously visible landmarks. If this is not given, the state space extents for these degrees of freedom need to be restricted or fully removed. In general, using adaptive diffusion allows to reduce error with respect to using static diffusion. This is however only true for the case of success. The case of failure, adaptive diffusion leads to higher error measures. This is caused by the adaptive process spreading the support samples further in order to find a mode.

The influence of the number of landmarks present in the moving model, revealed important properties of the global model. As long as the number of local landmarks was sufficiently high to specify clear constraints, the global model could follow the related motion. An observation fixing fewer degrees of freedom that the state space offered, left states space dimensions under determined. Especially in the case of adaptive diffusion, the standard deviation increased drastically in these excess dimensions, causing instability. This may be avoided by restricting maximum diffusive values and extent of the state space dimensions themselves: A sufficient number of landmarks must permanently be present. Further, for the given parametrization, a ratio of 1/3 of the landmarks could be chosen to be static without hampering the proper detection of the global motion component. Since the algorithm lacks a propagation term besides Gaussian diffusion, it accepted static components as the sought target more easily than the moving. Hence, a fraction of more than 50% of landmarks being static leads to certain failure. The presence of competing modes of motion will be approached in the proceeding chapter.

Generally, the artificial motion model could yield robust results over much wider ranges of parameters than proposed reference parametrization assumed.

3.4.2. Sparse, Gradient Based Landmark Model

For modeling landmarks, a sparse contrast based feature description is proposed. Its basic properties were evaluated on artificial image data of features resembling liver vessels in EPI image data.

The appearance model requires both a function for reducing descriptive points to a fixed size set and a weighting function applied to the resulting set.

Point pair based point reduction shows to be more robust than single point based, since a context of two points belonging to a local gradient is preserved. This leads to

higher resistance against Gaussian noise. The gain is equal for SSD based and contrast based weighting. For higher levels of noise, correlation of the SSD based observation decreases more quickly than the contrast based. In this range, the additive nature of the contrast weighting's kernel can compensate the external additive noise better than the SSD. This, however, mostly affects the range of very low SNR < 0.5.

A point-pair based reduction function is presented and found to outperform two single-point based reduction functions, in that it yields models more resilient to Gaussian noise. In single site reduction, an increase in the number of points may deteriorate the model. Pair based reduction is found to circumvent this problem. A tuning parameter γ can be used to weigh emphasis between intensity and spatial extent of the point pairs. It controls an agglomeration process of gradient point pairs around high contrast regions.

After point reduction, the appearance model requires a weighting function for application. A contrast based weighting function is compared to an SSD based one. For low SNR values they are found to perform similarly well for reasonably low SNR. The SSD based function is however more restrictive to high noise values as to model appearance changes (data not shown). While both functions can successfully be applied to the given sparse sample data, the contrast based function is recommended for quick large scale adaptation. For detailed, small scale adaptation, SSD based weighting is favorable, due to the more precise target encoding.

Only slight benefit can be gained by optimizing the reduction functional by Simulated Annealing as opposed to a greedy algorithm. Since SA requires much more computation, greedy optimization is suitable for fast initialization in real-time applications. Therefore, in order to operate a fast, noise-resistant algorithm, the combination of contrast pair based initialization with a contrast based weighting function is chosen.

The idea of using a simplified weighting kernel may be found in the literature. In [KMI02], the authors use *increment sign correlation*, where the signs of the differences of direct neighbor voxels are stored. This technique is refined in [KSI03] to motivate a selective correlation coefficient.

The method proposed here offers a range-to-gradient optimization parameter, which can be used to adjust to different gradient slopes or scales in the image. Although this parameter γ will be used single valued in this work, it might be used for statistical modeling, since it may create multiple representations from the same region.

3.4.3. 2D MRI Data

Two dimensional EPI data of liver motion caused by breathing were used to investigate a feature detector and the two-level tracking model.

An LBP based kernel filter is found to outperform a local maximum filter as it reaches either lower distance errors or higher ranks of the detections. Furthermore, the LBP based filter has the advantage of a fixed (here 8) neighborhood[6], whereas the cost of the evaluation of a maximum filter scales quadratic with the kernel size. Using the kernel size of 3 vx for the LBP based detector, the rank's 90^{th} percentile is located at $\approx$ 30–40

[6]If the number of neighbors was increased for large kernel sizes, the scaling would be linear.

features. This means, that for the considered data, features up to rank 30–40 need to be considered for integration, when using only this optimum kernel size. While an optimum kernel size can be found for the data considered, in the actual application a range of kernel sizes is used. If the first one does not return a sufficient amount of landmarks, the process is repeated with another size. Given that the annotation shows 20 features at most, this means that 10–20 additional landmarks are detected. While these often reside in the liver as well, they do not always. False positives arise for instance in the spine region or the kidney, if visible. While the former is likely to interfere with the construction of a viable tracking model, the latter may even stabilize it. The kidney does not follow the exact motion as the liver, but is affected by breathing motion in a very similar way. Therefore, it is not necessary to remove kidney landmarks from the model before tracking.

Applied to real-time EPI images of liver motion, the two level tracking algorithm is capable of following the presented global motion pattern well. The local stage can be used to decrease global error levels further. In the given dataset, some landmarks are barely visible. These can be difficult to annotate and track, as they may only be visible in particular phases of the breathing cycle. Few cases of deep inhalation posed difficulties. This could however be resolved by introducing adaptive diffusion.

3.4.4. 2D US Data

In 2D ultrasound, the two-level model is generally found to show the desired effect: A global model introduces stability, whereas the local one refines current position estimates.

Available ultrasound reference data differed from MRI data in that few landmarks only are annotated. In most of the data, 2D liver ultrasound shows fewer vessels than 2D liver EPI images do. In multiple cases of US data only one landmark was given and the global model was less efficient. Parameters such as scaling — intended to be evaluated on larger, global ranges — act on a single landmark and can introduce degrees of freedom which are not well defined. As seen in artificial data experiments, this may lead to instability, if adaptive diffusion is used on the global level.

Speckle noise leaves the plain initialization of a contour based appearance model with the need to overcome abundant dark inter-speckle regions. A median filter could be used to ameliorate the process. It was found sufficient to apply the filter to the initial image only, which saves computation time during application. Naturally, some ultrasound images show a strong bias for (vertical) gradients, in the direction of sound. In the selection of gradients for contour based description, horizontal and vertical gradients are both important and a compensation for the bias is required. Therefore, the median filter is chosen with a non-isotropic kernel, which is found to balance the mixture of gradient directions. Its settings are naturally data dependent, but the single chosen setting did yield reasonable results for all observed types of ultrasound data. With this type of preprocessing the algorithm is fundamentally different from Speckle Tracking. While Speckle Tracking tracks the motion of individual maxima in the back scattered signal, the algorithm in this work aims to detect the motion of larger scale level structure.

Long handheld ultrasound sequences can show drifts in vessel and diaphragm positions. These occur as absolute as well as changes relative to other landmarks. Often, absolute difference were related to the actual motion of interest. Relative differences, on the other hand, can be attributed to changes in the slice alignment, as a result of probe or volunteer position changes. Due to the irregular structure of liver vasculature, there are cases in which the inter-landmark drifts can not be matched by a global similarity transformation. Also, changes in the vasculature may appear as motion, but start to contradict the positioning of the diaphragm.

In an application, the relation of diaphragm and liver vessels should not drift strongly throughout the course of an image sequence. If such a drift is present, it is an indicator for the imaging plane having changed with respect to the target. In this case, the tracking result loses its value, as the detected landmarks — even though they may look similar — do not correspond to the reference markers.

Additional landmarks located on the diaphragm are found to stabilize the track when the imaging plane is stable as well. Otherwise, they can even destabilize it, by misguiding the global model. As a reflecting surface, the diaphragm provides a strong contrast, which can be much more robust to slight angulation of the imaging plane, than vessel landmarks. If the image plane is held constant with respect to the liver, the diaphragm can provide a stabilizing tool, if vessels are not clearly identifiable. Due to the strong contrast, the diaphragm however dominates the optimization of the global model. Thus, if the imaging plane is angulated, vessels may appear to have moved with respect to the diaphragm. A global model optimized to the diaphragm location then provides an unsuitable prior for the detection of vessels. Computationally, the addition of diaphragm landmarks is relatively expensive, as a grid of seven such landmarks already outweighs the often few present vessel landmarks. In this regard, a more delicate selection process for support structures might be useful. They could for instance be based on a segmentation of the diaphragm, and explicitly integrate the diaphragm edges into the model.

Both, 2D US and 2D MRI data, can suffer from through-plane motion (TPM). TPM causes tracked structure to disappear, new structure to appear. It can change the appearance of vessel landmarks as the cross section is altered. Generally, in both types of data TMP can be found. In MRI it is related to tissue moving through the fixed imaging plane. In ultrasound data it may also be caused by an unstable location of the handheld ultrasound probe. The effect in both modalities differs. In MRI data TPM affects the neighbor relations of landmarks. The appearance often is not altered drastically. In fact, cardiac pulsation shows more pronounced changes in the appearance of large vessels than TPM. The low amounts of TPM only affect a low fraction of landmarks, such that the global modal remains stable. In ultrasound, TPM visibly changes the appearance of landmarks. Just as in MRI, it changes the neighborhood relations. However, since fewer vessel landmarks are visible in the ultrasound data, these changes affect the global model much more strongly. Means to stabilize the global model have been proposed in integrating the diaphragm in to it. This could amend the effect of changing landmark appearance, but for high magnitude TPM is also found to introduce conflicts into the global model.

3.4.5. 3D MRI Data

In three dimensional EPI image data the motion of prostate, rectal wall and urethra could be followed. While for prostate capsule and urethra the additional local optimization was found to decrease error, this distinction could not be made for the less clearly defined rectal wall region.

The presented global model is able to capture the overall motion in the available prostate motion sets. It does however not describe motion of the prostate in detail. Particularly the urethra shows motion which in some datasets is decoupled from the motion of the prostate. In those, it does not always follow the global motion or exaggerates it strongly. It is not clear whether this is conventional deformation or the effect of ablation in certain regions of the prostate. Here, a local correction is able to capture the motion of the urethra to certain extents, yet not fully. This may be seen in figure 3.17, where the urethra motion is better captured by the local correction. The optimization of local landmarks is limited in range, such that it is not possible to capture fully arbitrary urethra motion. This is done to prevent more uncertain landmarks losing track.

Rather than motion, the prostate MRI data shows deformation. In given sets, the observable global motion was limited to ranges of 5 mm to 8 mm. Global adaptation of the algorithm is possible in larger areas of few centimeters. However, the algorithm is capable of capturing larger global motions than present in the provided data. This was tested by artificially shifting image content in either reference or current image.

As a common problem, landmarks at high contrast boundaries produce the most certain position estimates. The estimates' standard deviations on edges such as the prostate border and urethra are found to be lower than those within comparably even intensity areas within the prostate capsule. In these areas, the local correction tends to follow the global prior. Here, the algorithm might benefit from a regularization between neighboring sites.

In the generated annotation, rectal wall structure is the region of largest tracking error. This is related to the structure only yielding a constraint perpendicular to the wall. In this case, the local correction performs even or worse compared to the global. Another argument for this effect is the annotator also not being able to distinguish the motion well — in this region, an annotator also must estimate the motion by what seems plausible on a global level. Hence, benefit of a local correction cannot be shown with certainty by the annotation.

3.4.6. Common Findings

This section proposed a particular implementation of a two level filter. The choice of two levels facilitates the generation of a simple hierarchy. Both levels of a hierarchy require parametrization. Fast, lightweight operations can be left to the initial root position estimation, whereas more complex details can be computed on leaf stages. In particular, the algorithm is designed for real-time matching of a sparse field. While the leaf stage covers additional, local deviations from the global pattern, it does so in a limited way. Aside from further refining the leaf stage, its result might be used as prior input for other

algorithms. It could be forwarded to an interpolation or to a registration algorithms, which converge to their optimum more quickly, given a useful prior. The simplified global motion model is shown to yield robust results for a series of test scenarios with simplified perturbations observed in medical image series. In this artificial environment, the limits of the algorithm have been explored with respect to tracking breathing motion. The knowledge about these limitations may serve as a tool for the discrimination of valid from invalid input. There is no dedicated interaction or regularization term in the optimization on the local level. Simple mass spring FEM model might be introduced as interaction terms on the leaf stage. This might increase computational cost drastically and is not directly compatible with the underlying stochastic optimization scheme. Here, the direct interaction of local landmarks is replaced by the assumption of highly correlated motion described by the global model. Local correction accounts for errors in this assumption.

In each of the three proposed applications to tracking, a real-time performance could be reached during execution. In the 2D cases this means that results were computed in less than the frame duration of approximately $100ms$. The case of 3D data yielded tracking results within one second while images are updated every few minutes. In all cases, the initialization phase — particularly the reduction of the landmark description — required more time. Depending on the sparsity, a factor of five could be found. In the implementation, this however has to be executed once. Since the algorithm is implemented for a single CPU thread, there are straightforward options to increase performance. In particular, there are serveral options to parallelize the execution. This will be considered in detail in the final discussion.

The presented implementation of the two level algorithm operates on node-based appearance models and discards temporal priors for the leaf stage — the algorithm seeks landmark representations as observed in the reference situation. Using a leaf based appearance model and maintaining distribution priors on the leaf stage, the global representation is updated with each step. In an assessment in the 2D EPI case, this was found to be unstable over time, if the state space of the leaf level is unbounded. The reason is that single leaf node, if unstable within a single frame may drift among the global model and be unable to recover. These deteriorate both quality of the global model as well as their own temporal prior. Hence, in order to operate this implementation, more high level logic is required, which sorts unstable and stable landmarks from the global model. In this chapter this problem has been approached by only using hierarchical priors for the local model, i.e. disregarding their temporal priors. Even if a landmark is temporarily undetectable, the filter will return a position estimate, which then is mostly determined by the global estimate. This resolves the necessity for additional maintenance, but possibly increases the distance between global current and reference representation in some phases of the track.

Outlook The proposed two-level algorithm is very general and can be extended to adopt alternative appearance models easily. Additional levels may be integrated and tailored to different situations. The two-level approach demonstrates the most simple

hierarchical application. Its output, being a sparse deformation field, may need to be processed in further steps. Multiple ways to proceed from this data include spatial interpolation to dense deformation field via piecewise affine warp [MB04] or the use in a dedicated, statistical shape motion model for spatiotemporal prediction [Pre+14]. As an extension, methods for the interpolation of the result vector field might be considered. For thin plate splines [Duc77] there exist extensions which allow the integration of anisotropic position uncertainty for elastic registration [Roh+01].

In general, joining landmarks into clusters for evaluation shows the expected stabilizing effect, if the correct landmarks are chosen. Joining vessels into clusters shows a stabilizing effect, as well as joining clusters with further structure such as the diaphragm. This however requires three levels, which cannot be handled by the two level model. The motivation of a fixed three level model is not trivial, as the new inner layer of nodes requires knowledge of the actual components of motion. These are not known beforehand and cannot be derived from the initial image. The following chapter will suggest a solution to these limitation.

3.4.7. Conclusion

In this chapter the necessary components for applying the 'building blocks' of chapter *Gaussian Transition Filter Model* (2) to actual medical image data have been presented. They have been extended by

- A generic description of a hierarchical tree for tracking articulated motion
  - using a transformation hierarchy
  - offering different ways to encode the appearance model along the tree
- A sparse description of landmarks, which allows for real-time processing. It features a tuning parameter that defines what scales of gradients are integrated into the model.
- An inexpensive feature detector for liver vessels in EPI image data

Application to three sample cases of medical image data has demonstrated the real-time applicability. The hierarchical model has been shown beneficial for the robustness of the results, but in this chapter was limited to a fixed, two level model. The proceeding chapter will extend this model to allow a generic tree to adapt and also to formulate a generic description of appearance from the tree.

3.5. List of variables

Group	Var.	Brief	Detail
Reduction Function	ϵ_i		Single entity energy
	$\kappa_{i,j}$		(Repulsive) entity interaction
Contrast Based Reduction	Δb_i	Intensity Diff.	Absolute intensity difference of a gradient point pair.
	D_i	Spatial Dist.	Absolute spatial distance of a gradient point pair.
	γ	Tuning	Tuning parameter for the selection process; shifting emphasis either towards high contrast or short distance gradients entities.
Experiment	SNR	Signal-to-noise ratio	Estimate of the signal-to-noise ratio of an artificial image.
Result Measures	R_0, R_1		Local response of a filter to the reference image patch, and to a current image patch, resp.
	M_{SD}		'Survival Diagnostic' obtained from a response.
	M_{PC}		Pearson's Correlation between two responses (reference and current).
Artificial Breathing Model	k	Frame Index	Index of a temporal step. With $k = 0..k_{max}$, where $k = 0$ denotes the reference.
	T_F		Fixed duration of a single frame.
	T_{cycle}		Fixed duration of a breathing cycle.
	A_T, A_S, A_R	Amplitudes	Amplitudes of global translation, scaling and rotation, resp.
	σ_A	Appearance Width	Width of a landmarks likelihood function.

4. Dynamic Hierarchy Model

When a model is initialized from a single reference frame, detailed information of expectable motion may be unknown. A set of landmarks may be detected, but their mutual relationships and motion patterns are initially unknown. Assuming the relationship of landmarks may be captured in a hierarchical graph, a single initial frame does not suffice to derive this graph. It must be learned by observing the temporal evolution. In this chapter, a hierarchical model will be dynamically generated and updated using on-line using hierarchical clustering and agglomerative adaptation. Nodes will be added to or and removed from the tree. Goal is to obtain a motion hierarchy which is able to encode the observable image content.

Key idea is to start tracking from a situation of least information about the expectable motion. In this, the describing hierarchical tree consists of a sole root node. Only in the reference state, the root node model certainly is able to describe the observed state. In further progress of the motion, the model may need to be adapted. As with distance to the reference frame, the matching error increases, the model indicates its descriptive limitations (Fig. 4.1). Once this happens however, additional information of the motion is available, since all frames up to the current are known. This may be used to update the motion model in such a way that it better describes the observable motion pattern. The

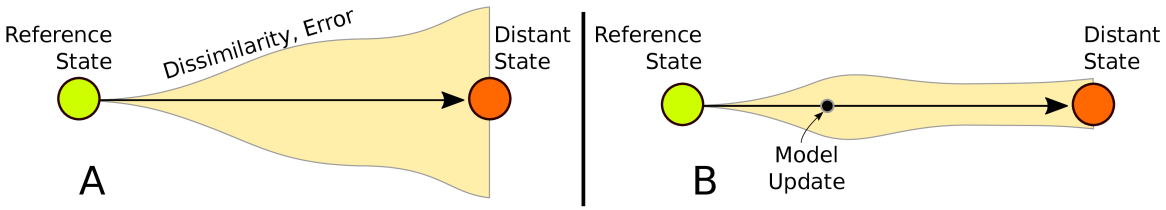


Figure 4.1.: Moving from reference to a distant state, the error in the description may increase, due to limitations of the model itself (A). Updating the model may account for appearances not initially anticipated (B) and thus reduce the error in the distant state.

approach in this section relies on updating the underlying motion hierarchy. New nodes will be introduced to capture yet undescribed residual motion. Further, mechanisms are introduced which limit the growth of the tree. Existing nodes will be updated in their function and possibly removed if they do not maintain a benefit.

The chapter is subdivided into the following sections: The first part describes the model by introducing the notion of associations between landmarks and nodes of the hierarchical tree description of motion. The concept will be used as a basis for two

distinct adaptation processes described in the following two sections: A top-down clustering approach, and a bottom-up rearrangement and merging process. The subsequent experiments section defines artificial scenarios which investigate basic behavior of the proposed model. Furthermore, the adaptation process is explored on EPI image data of liver motion. The chapter is summarized with an outlook on possible use cases, extensions and simplifications of the algorithm.

4.1. Model

In the previous chapter *Fixed Hierarchy Model* (3), a model hierarchy was provided in advance. The algorithm used the same structure for any kind of data: A global common motion and individual local deviations thereof. For a general case, where the tree is not predefined, it must be derived from the data itself. For this, in a first step, the hierarchical model must be defined such that it allows for sequential steps of adaptation.

Within the vast space of possible tree configurations, it is not possible to evaluate all configurations or even monitor them comparably over multiple steps. The choice of a hierarchical tree therefore requires that only trees with an already measured benefit over others are considered. These may then be refined further. This suggests an approach of hierarchical clustering in which the tree is built up from the root node by recursively applying split operations to the existing nodes.

Here, such approach is chosen, which splits existing clusters into sub-clusters. The tracking process starts out with a root node, holding a representation and describing the motion of all landmarks. New nodes are created by splitting existing leaf nodes into further clusters of landmarks (Fig. 4.2, A), which are then appended. If sibling nodes are found to express the same motion, they can be joined again (Fig. 4.2, B).

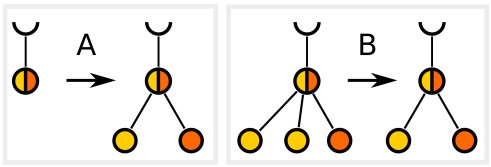


Figure 4.2.: Hierarchical split and join operation. (A) Adding child nodes to explain independent motion of sub components. (B) Removing excess components by merging nodes.

In the hierarchy, nodes describe clusters of common motion. When new sub-components are formed, new appearance models and transformations for these must be defined. Splitting an existing node must create new appearance models for the child nodes. For this, the connection between a node and its landmark clusters is relaxed. This *Model* section will introduce the notion of *associations* between features and nodes, which will lead to a more generic formulation of the tracking process. It will allow for on-line buildup and reconfiguration of a hierarchical tracking tree. A particular set of rules for

associations will be used as boundary condition for rearranging associations themselves and even the hierarchical tree.

4.1.1. Node-to-Landmark Association

In the previous chapter *Fixed Hierarchy Model* (3), two variants of associating the appearance to the hierarchical tree have been described. The *node based* appearance model assumed that each node of the fixed hierarchical tree was provided an own model. This does not allow for of the hierarchical tree without generating entirely new models. The *leaf based* appearance model on the other hand assumed that the leafs of a fixed tree hold the building blocks to construct the appearance models of the parents. In this model, each node is associated to all appearance models of its leaf nodes. The leaf based model already conveys the idea of constructing the appearance model from a set of smaller models. These are however not idenpendent of the tree as they are explicitly associated with the leaf nodes.

In order to generalize the idea of a hierarchical tree for tracking, one must acknowledge that landmarks exist independently of the tree itself. Nodes in the hierarchy represent observable clusters of common motion. One may interpret the hierarchical tree as the result of a hierarchical clustering process. In this, the root node is *associated* with all landmarks. This set of associations is subdivided on next level by splitting it among the child nodes of the root. Each landmark now is associated to the root and associated to one of the root's children. These children may further be subdivided until there are only leaf nodes with an association to a single feature each left (Fig. 4.3). In order to

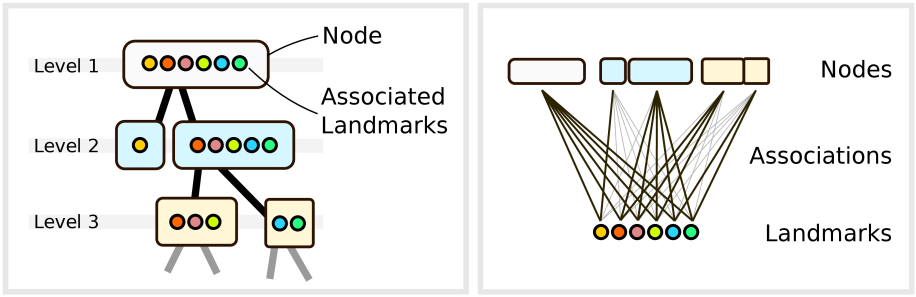


Figure 4.3.: Left: Propagation of landmark associations along a hierarchical tree. Right: The notion of associations between nodes and landmarks will be used to dynamically alter the hierarchical tree.

introduce a general scheme of hierarchical clustering and tracking for an arbitrary set of landmarks, the notion of an *association* between nodes and landmarks is introduced. Associations create the binding element between the hierarchy and the set of landmarks. Considering them as individual entities will allow rearranging the function of the hierarchy dynamically. As association has different properties, the most important one being

a label-like state. The motivation for associations will start out with a binary state association, but then be extended to a ternary description.

Each node i in the tree is associated to at least one landmark. Landmarks may receive associations from multiple nodes. There may not be more than one association between node and landmark. The *association* entity describes properties of the relationship between node i and a landmark j. Its primary property is a label-like state, which in the most simple case would take a binary value

$$S^A_{i,j} \in \{0,1\} \stackrel{\text{def}}{=} \{\text{not present}, \text{present}\} \tag{4.1}$$

It indicates whether the association exists ($S^A_{i,j} = 1$) or not ($S^A_{i,j} = 0$). In this chapter, index i will be used for nodes, j for landmark. An association or one of its properties is indexed by i, j.

The binary state of an association (Eq. 4.1) describes associations in a fixed tracking hierarchy. States *present* and *not present* are permanent. They are however not independent along the hierarchy. Hierarchical clustering motivates a normalizing condition: The association of landmark j to node i, depends on sibling and parent node's association (Fig. 4.4).

1. **Availability** A landmark j is available to node i, if node i does not have a parent node or the parent node p is associated to the landmark ($S^A_{p,j} = 1$). If the landmark is available, it may or may not be associated to node i ($S^A_{i,j} \in \{0,1\}$); otherwise it may not ($S^A_{i,j} = 0$).

2. **Normalization** Among a set of sibling nodes $\mathcal{S}$, to which landmark j is *available*, exactly one is associated to it.
   $\sum_{i \in \mathcal{S}} S^A_{i,j} = 1$.

Figure 4.4.: Rule set for forwarding associations along a hierarchical tree.

Rule (1) determines, which kinds of associations are allowed, whereas (2) defines that exactly one of them is realized. Adhering to both, there are now three states an association may obtain (Fig. 4.5). It may be *forbidden* by rule (1). Unless it is forbidden, it is *allowed*, i.e. in a state which is further differentiated by rule (2). Among allowed sibling associations to the same landmark, most are not established (*weak* association) except for a single association which is established (*strong* association).

This means: Strongly associated landmarks of a parent node are split up among the child nodes. Each of them is strongly associated to exactly one child and weakly to the other children. A node cannot be associated (strongly, weakly) to a landmark if the parent node is not strongly associated to it. The dynamics of the tree will be introduced with the exchange of weak and strong associations.

The type of the association determines its contribution to the tracking process. Forbidden associations do not partake in it. Differentiation between weak and strong associ-

Set	State	**M**easure **U**pdate
Binary	*not present*	
	present	*M U*
Ternary	*forbidden*	
	weak	(*M*)
	strong	*M U*

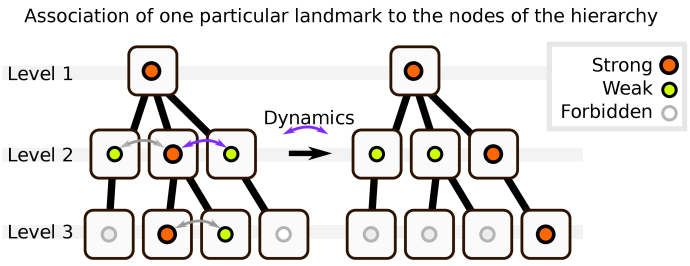


Figure 4.5.: Left: In order to allow for updates of associations, a third *weak* association state is introduced. To a limited extent it takes part in the measurement process for a node, but does not affect the node's result in the update. Weak and strong associations can be exchanged, giving room for a dynamical tree configuration. Right: The association to one particular landmark is passed along a hierarchical tree. Obeying the set of rules (Fig. 4.4), individual, single paths of strong associations are formed for each landmark. The rule set allows changing weak and strong associations to sibling nodes (arrows), which is the basis for dynamic tree adaptation.

ation affects the dynamics of the model. Only *strong* associations are treated as *present* associations in the binary state: They update their set of sample weights and provide them to the update step of the associated node. Weak associations will partake in a reduced kind of measurement, which is not integrated into the tracking result, but employed to occasionally update the associations themselves. Dynamics will be introduced to alter states between *weak* and *strong* associations, and altering these will affect the differentiation of *allowed* and *forbidden* state on descendant levels along the hierarchy.

This particular set of rules leads to an arrangement where a landmark strongly associated with the root node passes on this strong association along a single path until a leaf is reached. Direct sibling nodes to strongly associated nodes receive weak associations with the landmark, whereas associations to all other nodes are forbidden (Fig. 4.5). The rules ensure that the number of strong associations to a landmark does not increase more than $\mathcal{O}(L)$ with L levels of the hierarchy. They reflect the idea of clustering the target model into complementary sets of landmarks.

The following terminology is used: If a node and landmark share a strong association, this work may reference them as *associated*. Otherwise the association type will be stated explicitly.

Properties and Usage of an Association For tracking individual nodes, further properties of an association are of interest, which are required for the tracking process (Fig. 4.6). These will be differentiated from properties of a node by a superscript A: First, the transformation matrix $M^A_{i,j} \in \mathbb{R}^{4x4}$ from the node's coordinate system to the landmark center. Second, the set of weights $\pi^A_{i,j,k}$ with $k = 1..N^i_S$ indexing samples of the sample set of node i. The weights of an association are related to the sample set of the node. The node proposes a common set of sample states indexed by k to all of

its associated landmarks. Each of them weighs these hypotheses individually and stores the result in the set of $\pi^A_{i,j,k}$. Furthermore, being related to node i, the association is implicitly appendant to a hierarchical level l_i. On the other end, it is related to a landmark, which is characterized by an appearance model — given by a set of points p_l and a measurement function $\mathcal{L}$.

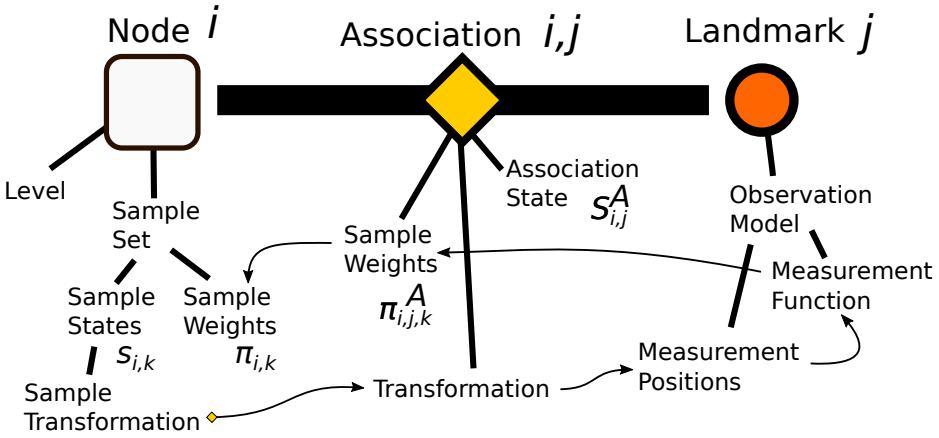


Figure 4.6.: The association of a node to a landmark joins the properties of a node with those of the landmark. It specifies details of the relationship via its own properties. The thin arrow path below indicates the ordering of the tracking process from the creation of new sample transformations to their weighting (see text).

Associations are a central element for tracking. Task of an existing strong association $S^A_{i,j} = 1$ is to obtained a weight for landmark j for sample transformations given by node i, in order to allow for a state update of node i. The tracking process for node i depends on the set of landmarks j with an existing strong association $S^A_{i,j} = 1$. The associations' transformation matrices $M^A_{i,j}$ localize the landmarks j in the coordinate system of node i.

The sample set of node i holds N_S samples indexed by $k = 1..N_S$; with states $s_{i,k}$ and weight $\pi_{i,k}$. Node i begins a tracking iteration by propagating the particle states to form new proposals $s_{i,k}$. These then require weighting. For this, weights of the associated landmarks are computed first and then joined into weights for the node. With $M_{i,k}$ summarizing the node center transformation of sample k, the *associated* sample weight $\pi^A_{i,j,k}$ of association j is computed as

$$\pi^A_{i,j,k} = \mathcal{L}(M_{i,k} M^A_{i,j} | I_{current}), \tag{4.2}$$

It represents a likelihood for transformation k of node i dependent on observation data of landmark j. Weights $\pi^A_{i,j,k}$ are then used to update the node's weights $\pi_{i,k}$. In order to join weights for the nodes sample set, a product or sum may be considered. A product would require the simultaneous identification of all landmarks in order to yield a weight greater than zero. Anticipating the model not being able to describe the correct motion

in the first place[1] or random disappearance of landmarks, a sum is used.

$$\pi_{i,k} \propto \sum_j S^A_{i,j} \pi^A_{i,j,k}. \tag{4.3}$$

Potentiating by an annealing schedule is applied to the $\pi_{i,k}$ as before (not to the associated $\pi^A_{i,j,k}$). The resulting, updated node weights are used to obtain a position estimate for the node and associated landmarks. The center of a landmark in its own coordinate system is $\vec{0}$. With $\hat{M}$ being the current transformation estimate of a node, its center estimate is $\hat{M}\vec{0}$ and associated landmark estimates are $\{\hat{M}T^A_j\vec{0}\}_j$. Note that the association sample sets are not taken into account for the association position estimates, and the estimates are based on the joint set of landmarks.

Aside from being used to update the node's sample weights $\pi_{i,k}$, the association weights $\pi^A_{i,j,k}$ are stored and will be used as input to hierarchical clustering in section 4.1.2. Furthermore, dynamics for the *weak* associations will be introduced in section 4.1.3.

Transformation Path Detail As in the previous chapter *Fixed Hierarchy Model* (3), the hierarchy of motion clusters poses the transformation hierarchy. As the tree is derived from the data itself, new degrees of freedom occur dynamically.

When new child nodes are appended to a parent via a hierarchical split, the parent-to-child transformation needs to be defined. A possible choice for M^{pc} might be from parent to the center of the associated landmarks' geometrical centers. However, if the set of landmarks associated to the new node changes, it is difficult to update the transformation center, since the node's states are defined relative to it. An option to avoid the additional degrees of freedom introduced by M^{pc} lies in setting them to unity (Fig. 4.7). Only exception is the fixed root's M^{pc}_1. As a consequence, a node's transformation center is not necessarily related to the positions of the associated landmarks. In fact, in the reference situation, all node transformation centers coincide in the position which is chosen as root reference. This choice leads to further simplifications in that the association transformations $M^A_{i,j}$ are identical for any node i; which is convenient when a child node inherits a landmark association from its parent.

This means further, that degrees of freedom, such as scaling and translation cannot be offered locally, and that except for the root level, all further levels should resort to translations only. For the purpose of demonstrating hierarchy building as means to outmaneuver a lack of degrees of freedom, this work uses only translations on every node — in particular, the root node is not provided extra degrees of freedom.

[1] The correct motion is to be tracked by building a correcting hierarchy.

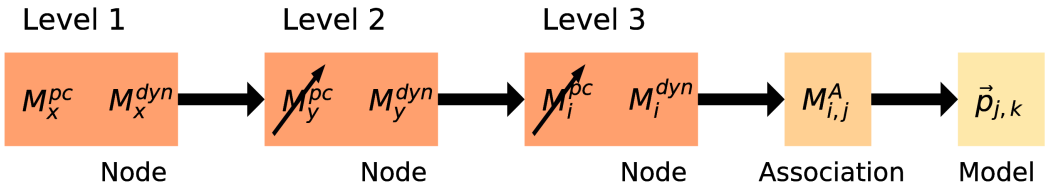


Figure 4.7.: Transformation path for a landmark j associated to a node i on level 3 of the hierarchy, used to transform the appearance model j's measurement points $p_{j,k}^{\vec{}}$. Setting parent-to-child transformations M_i^{pc} to unity for nodes after the root level is used to simplify the transformation chain at the cost of reducing the possible degrees of freedom (see text).

4.1.2. Association Clustering, Introduction of new Child Nodes

In order to organize the build-up of a hierarchical tree of associations, common motion components among them need to be detected — means for clustering them are required. Hierarchical clustering divides existing clusters into further clusters based on similarity or distance measures of the parent cluster members. Here, the members of a node are given as (strong) associations from a single node i to its landmarks. Node i represents the parent cluster and the process of clustering its strongly associated landmarks is to determine new clustered subsets of strong associations for the child nodes.

The main motivation of clustering is to determine whether and how an existing leaf node needs to be split into further child nodes in order to describe the observed motion properly. The assumption in the configuration of a node (here: set of strongly associated landmarks) is that the related appearance model can describe the observations in the data, i.e. that the set of associated landmarks undergoes a common transformation. Figure 4.8 depicts a simplified scenario of a single node with two associated landmarks. First, the sought appearance is defined by the landmark descriptions and the related node-to-landmark transformations $M_{i,j}^{A}$ (left). Thus, when the node's coordinate system is transformed into image coordinates, the two landmarks are sought in different positions. In the depicted example, the node may propose translations. The distance between the proposed landmark positions does not change. Thus, when the object consisting of landmarks 1 and 2 undergoes a translation, both appearance models will yield a maximum response for the same translation. In this case, data and model agree. The response is unimodal and it is not necessary to subdivide the node into further components.

However, the data may contradict the transformation model. In the example landmarks 1 and 2 move away from each other. Then, the appearance models of the nodes yield maximum responses in different parts of the state space. The joint response for the node becomes multi modal. Each mode indicates a sub component of motion. Splitting the set of landmark to new sibling nodes can then help to adapt. This is the motivation for the clustering process.

In this work, the response of a landmark j of node i is captured in the weight set $\pi_{i,j,k}^{A}$ of

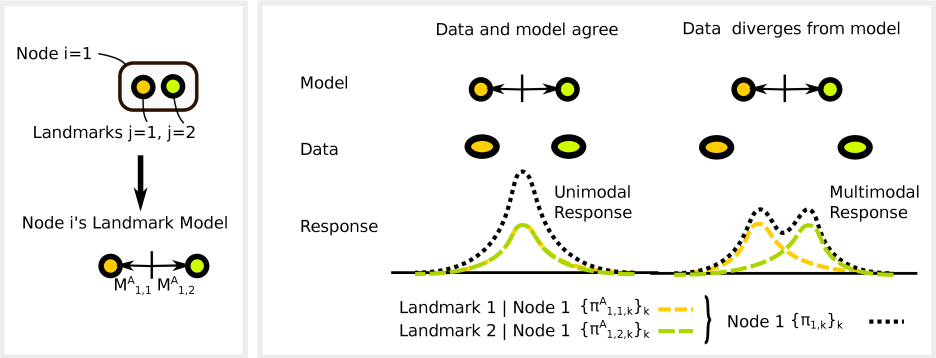


Figure 4.8.: Left: The appearance model of a node is a joint model of the appearance models of the strongly associated landmarks. Right: If the data can be explained by a single transformation of the joint model, the response in state space is mono modal. If the data starts to diverge from the model, the response becomes multi modal. This event indicates that further subdivision of the associated landmarks is necessary.

the association. The nodes measurement $\pi_{i,k}$ will become multi modal, when the weight sets of the associated landmarks yield different responses to the common hypotheses indexed by k. In order to detect sub components among the strong associations, it is therefore necessary to cluster their responses $\pi^A_{i,j,k}$.

In order to split a cluster i with a set of associations into new sub clusters, associations must be compared to each other. Each association to landmark $j \in \{1..N^i_F\}$ produces a set of weights $\pi^A_{i,j,k}$ for samples $k \in \{1..N_S\}$ (eq. 4.2). These sets are updated in every tracking step and reflect the individual response landmark j generates, given the common set of states $\{s_{i,k}\}_k$ of node i. Interpreting these sets as feature vectors, a similarity measure for two landmarks j_1, j_2 with vectors $\vec{\pi}^A_{i,j_1}$,$\vec{\pi}^A_{i,j_2}$ can be found in the cosine similarity (Eq. 1.2). For two weight sets $\vec{\pi}^A_{i,j_1}$,$\vec{\pi}^A_{i,j_2}$ this means that their normalized versions are compared for their similarity in expression of the components. Since weights $\pi^A_{i,j,k} > 0 \forall i,j,k$, their similarity is always positive $\mathrm{CS}\left(\vec{\pi}^A_{i,j_1}, \vec{\pi}^A_{i,j_2}\right) \in [0,1]$. Cosine similarity will be a key measure in the clustering process.

The clustering algorithm (Alg. 4.1) itself is implemented in iterative assign-and-update fashion oriented on Lloyd's [Llo82] implementation of the *k-means* algorithm. In this, K randomly initialized centroids are updated iteratively, with two distinct steps within each iteration: First, cluster assignments are determined by assigning input vectors to their closest cluster centers. Then, new cluster centroids are obtained by computing the mean vector of each cluster. The original k-means algorithm uses a Euclidean distance measure and returns exactly K clusters[2]. The proposed clustering algorithm differs in both the computation of distances as well as cluster centers and output — instead of K output clusters, the algorithm will produce $K^* \leq K$, as some clusters can be empty.

[2]Upper case K is used for the number of clusters; lower case k however continues to index dimensions of the input vectors. Individual clusters will instead be indexed by y.

Input to the algorithm is a set of entities to be clustered. In order to cluster the strongly associated landmarks of node i, the weight sets $\pi^A_{i,j,k}$ of the strong associations to the strongly associated landmarks j are used. A weight set $\pi^A_{i,j,k}$ will be denoted as vectors $\vec{\pi}^A_x = \vec{\pi}^A_{i,j}$, where k is absorbed in the vector notation. The index $x = 1..N_{input}$ indexes the set of strong associations $S^A_{i,j} = 1$ of node i; and by that the input vectors for clustering. Among the N_{input} input vectors, K clusters, indexed by $y = 1..K$ are sought. Each cluster is characterized by its center $\vec{c}_y$. Input vectors $\vec{\pi}^A_x$ and cluster centers $\vec{c}_y$ exist in $\mathbb{R}_0^{+N_S}$. The implicit normalization of the cosine similarity constrains them to the N_S dimensional unit hypersphere during the computation of similarity.

One option for using the cosine similarity would be to *assign* input vector to the cluster centers which are most similar. However, instead of using the similarity measure directly, the formulation is inverted here: Rather than assigning by a similarity to what a cluster *is*, the assignment is obtained by a dissimilarity to what a cluster *is not*. This approach allows for empty result clusters and therefore relaxes the considerations of a correct number of target clusters. For this, auxiliary complement cluster centers $\vec{c^c_y}$ are defined to be the element wise complement $c^c_{y,k} = 1 - c_{y,k}$. Instead of maximizing the similarity $\mathrm{CS}(\vec{\pi_x}, \vec{c_y})$, the assignment step minimizes the similarity to the complement center:

$$l_x = \arg\min_y \mathrm{CS}(\vec{\pi_x}, \vec{c^c_y}) \tag{4.4}$$

Figure 4.9 (left) depicts the measure for a two-dimensional space (k_0, k_1). Motivation for this approach is that if multiple very similar cluster centers in the local similarity neighborhood of multiple input vectors, only one will receive all assignments. This feature also is the reason that the number of obtained clusters can be lower than the input K (right).

The *update* operation uses the assigned labels l_x in order to produce new cluster centers. The new center of cluster y is computed as mean

$$c_y\prime = \frac{\sum_x \delta(l_x, y)\vec{\pi_x}}{\sum_x \delta(l_x, y)} \qquad \text{if} \sum_x \delta(l_x, y) > 0. \tag{4.5}$$

The algorithm 4.1 is initialized by assigning the cluster centers to random input vectors. Then an iterative procedure cycles the assign-and-update step N^C_I times. During the iteration phase, if the cluster does not hold labels ($\sum_x \delta(l_x, y) = 0$), it is assigned an element wise random linear combination of two random input vectors. As a result of the algorithm, each input vector is assigned a label $l_x \in 1..y$, indicating which cluster it belongs to. A cluster receives $N^{label}_y = \sum_x \delta(l_x, y)$ such assignments. Empty clusters are ignored. Only clusters with $N^{label}_y > 0$ are regarded as actual clusters, such that the resulting number of clusters $K^* = \sum_y (1 - \delta(N^{label}_y, 0))$ is smaller than or equal to K. In order to judge the obtained clusters with respect to their mutual similarity, a cosine similarity $M^{sim,cl}_{y_1,y_2} = \mathrm{CS}(\vec{c_{y_1}}, \vec{c_{y_2}})$ is computed.

Acceptance Condition Clustering algorithm 4.1 always returns potential clusters. The decision about actually using the result to create new nodes is made as follows. The

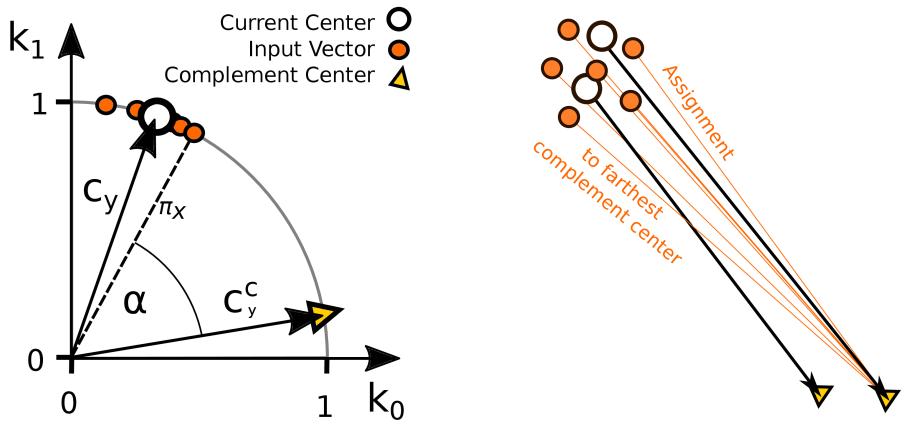


Figure 4.9.: Left: Clustering based on the maximum distance to the complement center: From center $\vec{c_y}$, the complement center $\vec{c_y^c}$ is derived. Input vectors $\vec{\pi_x}$ are assigned to clusters with a minimum cosine similarity CS to the complement center (visualized in angle α) to the complement center. (Vector tips depicted on unit sphere, in order to hint to the normalization of CS.) Right: Qualitative depiction of only one of two close cluster centers being assigned all entities in the local neighborhood.

criterion for acceptance or rejection of a clustering result is based on the mutual similarity of the result clusters. The matrix $M^{sim,cl}_{y_1,y_2}$ of the similarity of clusters to each other requires interpretation. Since the result cluster similarities are influenced by the contributing landmarks, their similarities are used as reference. A full landmark similarity matrix

$$M^{sim,f}_{j_1,j_2} = \mathrm{CS}\left(\vec{\pi^A_{1,j_1}}, \vec{\pi^A_{1,j_2}}\right). \tag{4.6}$$

is generated after the very first step of tracking. The clustering result is used to create new child nodes, if the minimum similarity among clusters is smaller than a proportion θ_{bind} of the similarity of landmarks among the available $\mathcal{L}_0$ landmarks involved in the clustering process.

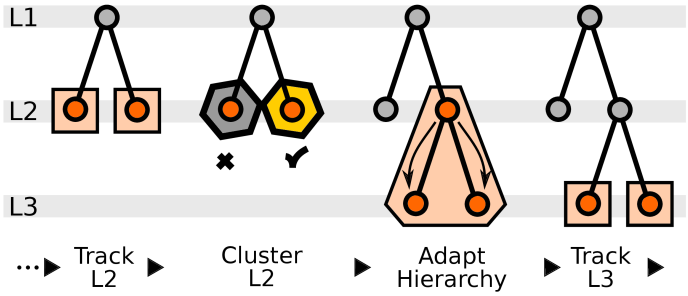
$$\min_{y_1,y_2} M^{sim,cl}_{y_1,y_2} < \theta_{bind} \min_{j_1,j_2 \in \mathcal{L}_0} M^{sim,f}_{j_1,j_2} \qquad \text{Cluster Condition} \tag{4.7}$$

If all conditions are fulfilled, the new cluster nodes are appended to the parent node. Each node receives strong associations to the related cluster's landmarks and weak associations to the landmarks related to sibling clusters.

The creation of clusters is considered for leaf nodes of the hierarchy. After a leaf node has been updated, its internal state is used to cluster its strong associations (Fig. 4.2). If this result suffices the set clustering conditions, the new nodes are appended to the parent. The algorithm proceeds, using the newly created clusters for tracking immediately.

Input : N input vectors $\{\vec{\pi_x}\}_{x=1..N_{input}} \in [0,1]^{N_S}$
Parameters: Number of iterations N_I^C
Maximum number of clusters K
Initialization For each cluster center $y \in 1..K$ choose initial center $\vec{c_y}$ by actual input vector. Add component-wise random noise in order to avoid degeneracy
for N_I^C iterations **do**
Compute complement centers $\vec{c_y^C} = \vec{1} - \vec{c_y}$
Assign labels $l_x = \arg\min_y \mathrm{CS}\,(\vec{\pi_x}, \vec{c_y^c})$
Update centers $c_y\prime = \frac{\sum_x \delta(l_x,y)\vec{\pi_x}}{\sum_x \delta(l_x,y)}$ if $\sum_x \delta(l_x,y) > 0$
Result: Input vector labels $\{l_x\}_{x=1..N_{input}} \in \{1..K\}$
Cluster centers $\{\vec{c_y}\}_{y=1..K}$
Assigned labels/cluster $N_y^{label} = \sum_{x=1}^{N_{input}} \delta(l_x, y)$
Assigned clusters $K^* = \sum_{j=1}^{K}(N_y^{label} > 0)$
Cluster Similarity Matrix $M_{y_1,y_2}^{sim,cl} = \mathrm{CS}\,(\vec{c_{y_1}}, \vec{c_{y_2}})$

Algorithm 4.1: Association Clustering Algorithm



Algorithm 4.2: Implementation of clustering: After a leaf node is tracked, it is considered for clustering. If the clustering step suffices the conditions, the result is accepted and the hierarchy updated. Newly created nodes are immediately integrated in the further tracking process.

4.1.3. Association Rearrangement Between Siblings

The approach of hierarchical clustering is applicable for iteratively introducing subdivisions into existing clusters. These have been created in a single step's clustering result and may have to be adapted as new information is provided to the tracking algorithm. Here, a long term maintenance process is proposed, which evaluates the performance of existing clusters in order to optimize the hierarchy. The optimization is done by exchanging *weak* and *strong* associations.

The parent-children relationship is key element of rearrangement. The set of children holds the same set of strong associations as the parent set does; yet subdivided among them. Introducing a mobility among these strong associations means allowing them to move from one child to another in exchange for the weak counterpart (Fig. 4.10). Via normalization, only one child cluster is assigned the strong association to an available landmark.

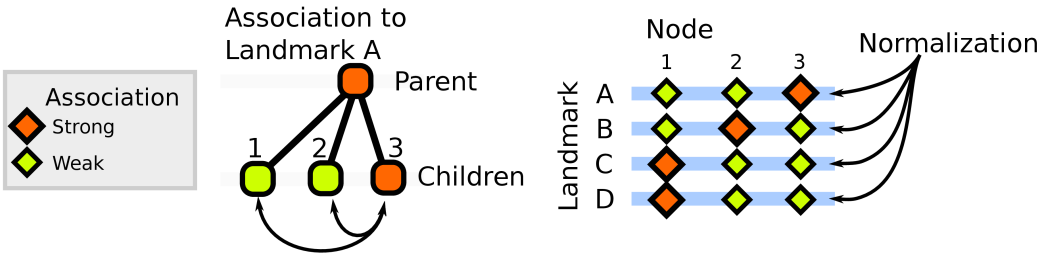


Figure 4.10.: Rearrangement of associations allows swapping strong and weak associations among the same parent (left), altering the node-to-landmark association matrix (right): The normalization rule (Fig. 4.4) assures each landmark to be strongly associated to a single cluster, whereas a cluster may hold strong associations to multiple landmarks.

In order to detect and justify a relocation of a strong association, an actual competition between child nodes is motivated. For this, in addition to its hard label like state $S^A_{i,j} \in \{forbidden, weak, strong\}$, each association is provided a soft state $s^A_{i,j} \in [0,1]$. The soft state does not indicate a label directly, but reflects how well a landmark has been fitting and currently fits to the proposals given by the related node. It is updated in each step. It also is subject to normalization

$$\sum_{i \in \mathcal{S}} s^A_{i,j} = 1 \tag{4.8}$$

among landmark j and a set $\mathcal{S}$ of sibling nodes, where the parent is strongly associated to j. The soft state translates to a hard state in such way that the strong association is established between the sibling with the highest soft state and all other associations to siblings are weak. Thus, by adapting the soft states, the presence of class states is updated. Since only strong associations have a major contribution to the computational cost, it remains limited.

In order to introduce dynamics, the soft states themselves will be strengthened by a reinforcing term. Similar to Hebbian learning [Oja82] subsequent normalization (Eq. 4.8) will be used as an implicit penalty term for those associations, which were reinforced the least. With reinforcing term $r_{i,j}$, the update — executed after the tracking step of all children completed — then reads

$$s_{i,j}^{A\prime} = \frac{s_{i,j}^{A} + r_{i,j}}{\sum_{i'} \left(s_{i',j}^{A} + r_{i',j}\right)} = \frac{s_{i,j}^{A} + r_{i,j}}{1 + \sum_{i'} r_{i',j}}. \tag{4.9}$$

The reinforcing term $r_{i,j}$ is chosen proportionate to a product of a direct association reinforcement $r_{i,j}^{Direct}$ for successful proposals and an attraction term $r_{i,j}^{Att}$ favoring larger clusters over smaller ones.

$$r_{i,j} = f^{AR} r_{i,j}^{Direct} r_{i,j}^{Att} \tag{4.10}$$

Global factor f^{AR} describes the reinforcement strength of the two individual terms and therefore the adaptation rate. Both will briefly be described before an example of their simplified functionality is presented.

Direct reinforcement strengthens associations between a node and any *available* landmark the node could provide a viable position proposal for: The reinforcing term $r_{i,j}^{Direct}$ is chosen proportionate to the quality $q_{i,j}$ of a proposal by the related cluster i for the location of landmark j. Quality $q_{i,j}$ is defined as

$$q_{i,j} = \frac{\mathcal{L}(\hat{M}_i M_{i,j}^{A} | I_{current})}{\pi_j^{ref}}. \tag{4.11}$$

where π_j^{ref} is the reference weight, landmark j obtained in the initial frame. Node i's position estimate $\hat{M}_i$ is chosen its maximum-a-posteriori (MAP) estimate, e.i fully defined by the highest weighted sample. The quality $q_{i,j}$ therefore represents the relative weight, the landmark j would obtain if it was transformed by node i's MAP transformation. In order to leave the total, direct reinforcement $\sum_{i \in \mathcal{S}} r_{i,j}$ among a set $\mathcal{S}$ of children, unaffected by their number, the quality is normalized such that:

$$r_{i,j}^{Direct} = \frac{q_{i,j}}{\sum_{i' \in \mathcal{S}} q_{i',j}} \tag{4.12}$$

An attractive reinforcement term additionally favors nodes, which already have many strong associations. Here this term is chosen, such that it increases monotonously with the number of associations, converging reciprocally to one. Hence, reinforcement of small clusters is reduced.

$$r_{i,j}^{Att} = \frac{Z_{i,j}}{1 + Z_{i,j}} \tag{4.13}$$

$$\text{with } Z_{i,j} = \left(1 - \delta(S_{i,j}^{A}, 1)\right) + \sum_{j\prime} \delta(S_{i,j\prime}^{A}, 1) \tag{4.14}$$

In this, $Z_{i,j}$ is the number of strong associations to node i, if node i was to gain (if $S^A_{i,j} = 0$) or keep (if $S^A_{i,j} = 1$) the strong association. The attractive term will be treated as optional factor — if it is disabled, it is set to 1 for any i, j. Other than the direct reinforcement, the attractive reinforcement is not normalized among siblings in order to assure the maximum $r^{att}_{i,j}$ remains bounded < 1. The following example will present the effect of reinforcement and give a motivation for the introduction of the attractive term.

Example The update of soft associations (Eq. 4.9) follows an exponential relaxation. With only the direct reinforcement term active ($r^{Att}_{i,j} = 1$), typical behavior can be derived in the following way (Fig. 4.11): Consider a set of L landmarks which are available to

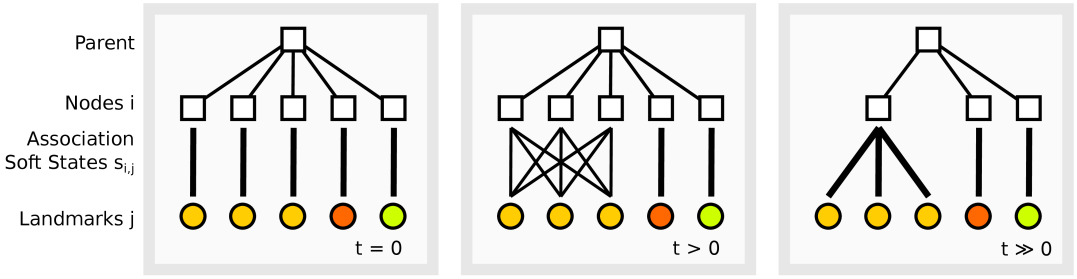


Figure 4.11.: Sample scenario (see text): Among a set of 5 sibling nodes describing motion of individual landmarks ($t = 0$). Within these, three landmarks move alike, such that the three related nodes start to compete for each of them ($t > 0$). By competition, the soft states between the L nodes and L landmarks are updated. Ultimately, the three nodes are joined into a cluster associated to the three landmarks ($t \gg 0$).

a set of L child nodes $\mathcal{S}$ and further, a reference situation $t = 0$ in which each of the child nodes is strongly associated to exactly one of them, bijectively. Initial soft states are in accordance to the strong/weak associations either 1 or 0, respectively. Further, let there exist a subset of L landmarks which belong to the same motion. Then, each of the nodes associated to the L landmarks and able to track correctly, can provide viable position estimates for any landmark among them. Assuming that their proposals are of equivalent quality $q_{i,j} = q \approx 1$, then the temporal update (Eq. 4.9) of the soft states s_t of the related associations, reads

$$s^A_{i,j,t+1} = \frac{s^A_{i,j,t} + f^{AR} q/L}{1 + f^{AR} q}. \tag{4.15}$$

Steady state is reached for $s_{i,j,\infty} = 1/L$. Substitution $y_t = s^A_{i,j,t} - 1/L$ leads to $y_{i,j,t+1}/y_{i,j,t} = (1 + f^{AR} q)^{-1}$ and reveals temporal evolution (Fig. 4.12, left).

$$s^A_{i,j,t} = \frac{1}{L} + \left(s^A_{i,j,0} - \frac{1}{L} \right) e^{-\ln(1+f^{AR} q)t}. \tag{4.16}$$

Requiring the difference between $s_{i,j,0}$ and $s^A_{i,j,t}$ to drop to a fraction $\epsilon(s^A_{i,j,t} - s^A_{i,j,0})$ with $\epsilon \to 0$, the time required for this is captured in

$$1/t_{adapt} = \ln(\epsilon)\ln(1 + f^{AR}q) \underset{f^{AR}q \to 0}{=} mf_{AR}. \tag{4.17}$$

In experiments, both ϵ and q are unknown. They may either be fitted individually or in the approximation $f_{AR}q \to 0$ as their product $m = q \ln z$, which represents the effective portion of adaptation rate f^{AR}.

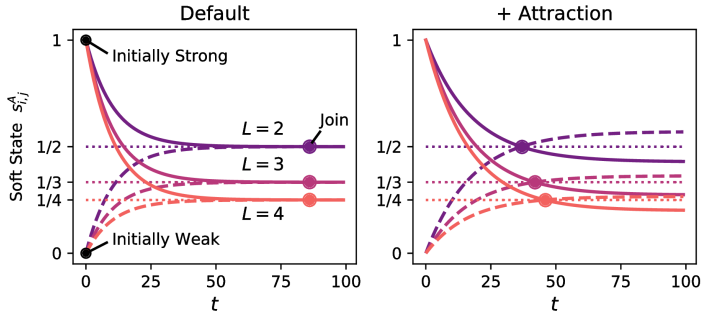


Figure 4.12.: Left: Soft state levels of initially strong associations (filled line) approache those of the weak associations (dashed line) until nodes are joined (dot, $\epsilon = 10^{-5}$) close to the steady state level (dotted line). Right: Enabling the attraction term alters the steady states of strong and weak association and leads to a more distinct intersection point (dots) at the same soft state level $s^A_{i,j} = 1/L$.

Enabling an *attractive term* in this scenario leads to different adaptation rates for strong and weak associations (Fig. 4.12, right) and to faster merging altogether. The factor for established associations[3] ($z = 1$) is $f_1 = z/(z+1) = 1/2$, whereas the rate for non-established associations[4] ($z = 2$) is $f_2 = 2/3$. Other than before, steady states differ for association types. Using an ansatz

$$s^A_{i,j,t+1} = \frac{s^A_{i,j,t} + f^{AR}qr^{Att}_{i,j}/L}{1 + f^{AR}q\sum_{i'} r^{Att}_{i,j}/L} \tag{4.18}$$

and progressing as before, it can be shown that the steady state for association i, j is $s^A_{i,j,\infty} = r^{Att}_{i,j}/S$ with $S = \sum_{i'} r^{Att}_{i,j}$. In the current scenario, where the attraction $r^{Att}_{i,j}$ of a strong association is $r_1 = 1/2$ and others are $r_1 = 2/3$, the sum is $S = r_1 + (L-1)r_2$, which can be used to derive further properties. In contrast to the limit convergence observed before, the introduction of attraction leads to a true intersection. It can be shown that its level remains at the previous convergence level $s^A_{i,j} = 1/L$. Time of this intersection increases with L.

[3]Leaving the strong association at the current cluster would leave the cluster with one association.
[4]Switching the association to any other node would leave it with two associations.

Implementation Associations are updated as the hierarchy is traversed. The tracking algorithm progresses from root to leaf level, evaluating node of the same level jointly. Once a level has been updated, the rearrangement algorithm updates the associations. It refers to the available landmarks, determined by the parent level, to update the associations of the current level. This may cause a change in the layout of strong associations on the current level. Therefore, in a second step, the algorithm then forwards the updated associations to the next level, rearranging the available and forbidden associations. Figure 4.4 shows the related algorithm 4.3 of such computation on the second level of the hierarchy. Additionally, removal cases are handled: If a node on the current level is left without strong associations, it is removed. This applies to any of its descendants, which likewise lack strong associations. After the available associations for the next level have been updated, the rearrangement process is finished and tracking may continue on the next level. Further removal of nodes appears at the end of the tracking step: If a set of children is reduced to a single child, if is identical to its parent with respect to its associations. Hence it is removed.

4.1.4. Full Algorithm

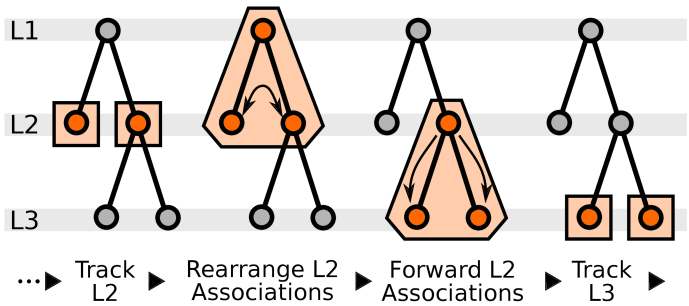
In the full algorithm, both clustering and rearrangement process are available (algo. 4.5). Before updating the hierarchy structure, a level is tracked. For each node tracked, position estimates are extracted for the related landmarks. Analogous to chapter *Fixed Hierarchy Model* (3), the position estimate of a landmark is determined by the transformation of the node which it is related to on the target level. This means that that as long as landmark is part of a cluster of multiple nodes, the cluster's sample weights $\pi_{i,k}$ are responsible for the computation of the estimate. This way, the position estimates of the same cluster share the same transformation. The obtained position estimates are stored as a property of the respective association.

During runtime, leaf nodes are checked for the generation of new clusters. If the cluster creation condition is fulfilled, new nodes are created. Right after creation they are tracked once, in order to bridge the adaptation gap, which made the clustering necessary in the first place. Ideally the generated clusters can better describe the new situation. After this, all nodes of the current level are involved in an update of their association rearrangements. In case, an update of the tree structure is required, it is performed immediately.

Once all levels have been updated, a vector field is to be obtained. Here, for each landmark the position estimates of a common hierarchical level are returned, i.e. position estimates of associations related to the requested level. If a landmark does not provide a position estimate for the requested level, it returns its highest level estimate instead.

Update of nodes on level l
for each (level $l-1$) parent node p **do**
  let $\mathcal{S}$ be the set of strongly associated landmarks of p
  track child nodes of level l, using current $\mathcal{S}` // Track
  for each landmark $j \in \mathcal{S}$ **do** `// Rearrange
    `// update soft states`
    for each (level l) child i of parent p **do**
      update $r_{i,j}^{Direct}$ (w. i's MAP transformation as $\hat{M}_i$) (Eq. 4.12)
      update $r_{i,j}^{Att}$ (Eq. 4.13)
      update reinforcement $r_{i,j}^{A}$ (Eq. 4.10)
    update and normalize soft state $s_{i,j}^{A}$ (Eq. 4.9)
    `// update states`
    with $i' = \arg\max_i s_{i,j}^{A}$
    set $S_{i',j}^{A} \leftarrow$ *strong*
    set $S_{i,j}^{A} \leftarrow$ *weak* for $i \neq i'$
  Remove nodes i without strong associations (incl. descendants)
  for each landmark j **do** `// Forward`
    for each (level l) child i of parent p **do**
      for each (level $l+1$) child g of parent i **do**
        if $S_{i,j}^{A} \neq 1$ **then**
          set $S_{g,j}^{A} \leftarrow$ *forbidden*
        else
          if $S_{g,j}^{A} = 0$ **then**
            set $S_{g,j}^{A} \leftarrow$ *weak*

Algorithm 4.3: Association Rearrangement Algorithm



Algorithm 4.4: Tracking algorithm progressing from level 2 to level 3, rearranging associations on the hierarchy.

```
for level l from 1 to L do
    for each node on level l do
        track node
        assign position estimates to the related associations
        if node is leaf then
            check formation of new nodes (clustering)
            if new nodes gained then
                track nodes
    for each node on level l do
        update associations (rearrangement)
for each landmark, extract position estimates from its associations for a
common level l (if missing, fall back to highest level)
```

Algorithm 4.5: Final Hierarchical Tracking Scheme

4.2. Experiments

Automatically derived hierarchies are studied on artificial data as well as EPI image data. Intent of using artificial data is the assessment of the algorithm being able to adapt to an unknown clustered motion. This part will be open to the use of multiple stages of hierarchical splits. A second part, using EPI image data, will investigate the effects of such algorithm on specific medical image data.

4.2.1. Evaluation Conditions

Artificial Data As described in previous experiments, artificial motion data is generated by directly defining a Gaussian mixture measurement function. Individual Gaussian distributions represent indistinguishable responses of landmarks. Other than in chapter *Fixed Hierarchy Model* (3), the artificial ground truth data is not composed of one global component of motion, but of multiple clusters. Each landmark i with $i \in 1...N_L$ is assigned a cluster $c_i \in 1..N_C$ with $N_C \leq N_L$. Different types of motion are given as samples (Fig. 4.13).

The set of landmarks belonging to a node is moved by the centroid motion of it. The position of a landmark is computed as a combination of the position $\vec{p_i^c}$ given by the unperturbed node transformation and a random positional term $\vec{\eta}$, which is individual to each landmark. In every step, the term $\vec{\eta}$ is chosen a random variable drawn from an isotropic, multivariate zero mean Gaussian with standard deviation $\sigma_{pos} \in [0, 1]$. This term represents an unpredictable jitter component which affects the true position of the landmarks.

The general appearance of landmarks manifests itself as an isotropic Gaussian likelihood of base width σ_A. In order to study the effect of unequal appearances for differing landmarks, variation is introduced to σ_A, such that for each landmark and frame the

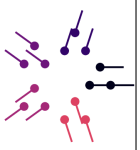
Independent			Optimizing		Combined
Star	Bend	Stretch	Warp	Liver	Heart
Radial split into N_C independent clusters of N_N nodes each	Bend motion of a 1D row	Stretch and translation motion of a 1D row	Bend and stretch	Three Clusters: ((Liver, Kidney), Transducer)	Two clusters, one scaling, the other translating
					

Figure 4.13.: Six basic artificial motion patterns are studied. Landmarks which follow the same motion, are grouped into ground truth clusters (indicated by color).

effective width is chosen $\sigma_A + f_r(1 - 2\xi)$, with scaling $f_r \in [0, 1]$ and $\xi \in [0, 1]$ is drawn from a uniform distribution $\mathcal{U}$.

2D MRI Data Medical image data is given by the EPI image data of liver motion, presented in section *2D MRI Data* (3.2.3).

Initial Root Layout Clustering and rearrangement algorithm have an opposite effect on the hierarchical structure. While clustering divides nodes, rearrangement joins them. In order to evaluate their effect in bare form, initial settings for the root node differ for both cases. For clustering, a *unified* root node, referencing all landmarks, is used. This

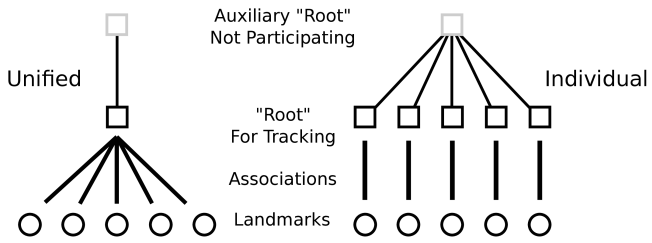


Figure 4.14.: Initial conditions for assessing clustering (left) and rearrangement (right). For clustering, all landmarks are associated to a single node (*unified*), whereas for rearrangement, they initially exist associated to *individual* nodes. In order to allow for these starting conditions a formal *auxiliary root* node is introduced, with the only purpose of making the individual nodes siblings.

is then to be split by clustering (Fig. 4.14, left). For rearrangement, landmark clusters

are yet to be joined. In the initial setting, landmarks are therefore maximally different, such that each receives an own *individual* 'root' cluster. This is possible by using an inactive auxiliary node. From this setting, the rearrangement process is then to restore the ground truth clusters (Fig. 4.14, right).

Result Measures The evaluation of the result of an arbitrary hierarchical tree may focus on both its result output and the tree itself. Result output will be gained level-wise. Dependent on the question, different result measures will be of interest. **Error**: A direct quality measure to this output is the distance between annotation or ground truth and the result vector field. If no reference exists, or the reference is incomplete — which may occur in medical data — an indirect measure is applied. The result vector field is used to compare local image similarity between reference position patches and current patches. **Jaccard Similarity:** In order to estimate how well an obtained hierarchy reproduced a known ground truth cluster layout, the mean of the maximum Jaccard similarities between ground truth and result cluster pairs is computed: In the set of ground truth clusters $\mathcal{C}^{GT}$, each cluster $\mathcal{C}_g^{GT}$ holds the set of landmark labels associated to it. The set of result clusters $\mathcal{C}^R$ consists of the result label sets.

$$\mathcal{J} = \underset{\mathcal{C}_g^{GT} \in \mathcal{C}^{GT}}{\text{mean}} \left(\max_{\mathcal{C}_r^{r} \in \mathcal{C}^r} \frac{\left|\mathcal{C}_g^{GT} \cap \mathcal{C}_r^R\right|}{\left|\mathcal{C}_g^{GT} \cup \mathcal{C}_r^R\right|} \right) \qquad \text{'Jaccard Similarity'} \tag{4.19}$$

For each hierarchy node, the set of strongly associated landmarks poses such a set. For each ground truth cluster $\mathcal{C}_g^{GT}$, the best match with respect to Jaccard similarity is determined. These are then averaged over all ground truth clusters for the final result score. This score for the similarity between ground truth clusters and result clusters will in short also be referred to as *Jaccard Similarity*. **Patch Wise NCC**: Normalized cross correlation is used to compare local reference patches to their corresponding patches in the current image. The result is averaged over landmarks. Note that the tracking algorithm itself uses a different (contrast based) measure on a sparsified set of evaluation positions. **Adaptation Rate**: The adaptation time is measured as duration between initial frame and the first frame of a significant structural change in hierarchy. It will be measured for association rearrangement. The adaptation rate is the reciprocate thereof. The *first occurrence of a significant structural change* can be defined in different ways, but always relates to the layout of strong associations and the hierarchy itself. In rearrangement scenarios, the algorithm is initiated with a high number of individual components and then reduces them. For artificial data, typical times for this process are given in the times t_{onset} and t_{finish}, where the first nodes are joined and the final layout is reached, respectively. In medical data, the finishing layout is not defined. Here, the relevant structural change is related to the number of present nodes having dropped below a certain fraction.

4.2.2. Artificial Data Setups

Clustering Artificial data is used to investigate the clustering algorithm. Without noise, the qualitative behavior of the algorithm when applied to artificial data is inves-

tigated. The influence of the binding threshold ($\theta_{bind} = 0..1$) and clustering iterations ($N_I^{cl} = 1..100$) are observed. Then, the special case of the configurable cluster split in the *Star* dataset more detailed analysis of the clustering behavior are undertaken ($f_r = 0..1$,$\sigma_{pos} = 0..1$). In particular the influence of noise to the clustering result is investigated. Since in these experiments the ground truth is known, both result error and Jaccard similarity serve as measures for evaluation. In clustering experiments, the initial layout of the root node is *unified*.

Rearrangement The rearrangement algorithm has the purpose of optimizing associations between nodes and landmarks. Implicitly — by moving all strong associations away from and therefore removing a node — it is able to alter the hierarchy itself. As before, the rearrangement algorithm is initially investigated on unperturbed artificial data. The relation of adaptation rate and varied reinforcement ($f^{AR} = 0..1$) is analyzed by fitting. Noise cases are then qualitatively investigated by introducing three levels of each noise type ($f_r \in \{0, 0.5, 1.0\}$,$\sigma_{pos} \in \{0, 0.03, 0.5\}$) into the artificial *Liver* data set and observing an adaptation process. From these, difficult noise settings are selected and more thoroughly analyzed with respect to their adaptation time under different noise conditions. In particular, the difference of default and attractive reinforcement is investigated. In a rearrangement experiment, the initial layout of the root node is *individual*.

Joint Algorithm The full algorithm, including clustering and rearrangement is applied to the artificial dataset *Liver*. Three distinct cases with attractive rearrangement are considered (tab. 4.1). They use a common attractive reinforcement $f^{AR} = 0.05$, but differ in the initial root layout and the use of clustering.

Setting ID	Initial Root Layout	Clustering θ_{bind}	Rearrangement f^{AR}	Attraction
1	Unified	0.9	0.05	Yes
2	Individual	0.9	0.05	Yes
3	Individual	0.0	0.05	Yes

Table 4.1.: Settings for joint clustering and rearrangement experiments on artificial data.

4.2.3. 2D MRI Data Setups

The qualitative behavior of the full algorithm, applied to EPI image data is investigated. In the experiments, the central tracking components are parametrized identically for every node. Transformation parameters are set to translation $T_x = T_y = 2\,\text{vx} \approx 3.5\,\text{mm}$–$4.0\,\text{mm}$. The evaluation of a node consists of 3 iterations with annealing schedule $\Lambda = 3, 6, 9$ using 100 samples. Adaptive diffusion is chosen. Each node uses the temporal prior

of its distribution without reset. Patches for appearance models are chosen larger than for the two-level model. Their radius is 20 vx $\approx$ 35 mm–40 mm. Pair based gradient selection of 48 gradients per patch is used. Measurement function is chosen contrast based. Different variations of clustering and rearrangement are investigated.

Rearrangement In order to measure the adaptation time depending on the reinforcement strength f^{AR}, few exemplary datasets are simulated, with different f^{AR}. The hierarchy is initialized as *individual* root layout. The temporal evolution of the number of nodes on the first level is recorded.

Joint Algorithm The algorithm is initialized with a *unified* root node and the hierarchy constructed by appending new nodes. For the given landmarks — and therefore only in the liver region — errors are computed. In experiment, where the error is of interest, initial landmarks are chosen to be the set of annotated landmark, extended by a set of 20 automatically detected landmarks. The former only reside in the liver area, whereas latter may also be found in other regions. No ground truth cluster annotation can be produced for the data. In experiments where the adaptation to available motion clusters is of interest, the local patch (radius 8 vx) based NCC is used as a result measure. In such experiment, only 20 automatically detected landmarks are required. Further, the qualitative behavior is shown by observing the formed clusters on the second level of the hierarchy after running the algorithm with different settings.

4.3. Results

4.3.1. Artificial Data Setups

4.3.1.1. Clustering

Noise Free Situation Performance of the clustering algorithm, applied to a noise free situation is shown in figures 4.15 and 4.16. Figure 4.15 displays two sample cases, where binary cluster splits are accepted. In motion pattern *Liver*, three components *liver*, *kidney* and *transducer and spine* behave individually. Two splits occur sequentially between steps (5,6) and (9,10). The first split separates the static *transducer and spine* cluster from the moving liver and kidney component, whereas the second separates liver and kidney motion. The second sample shows the *Star* dataset, with five clusters of three nodes moving evenly away from each other. In this case, multiple splits occur in the same step between frames (3,4). In detail, the algorithm traverses nodes by level. A first split occurs after the root node/level has been tracked. This creates two new nodes on the next level. The tracking algorithm progresses to the next level and tracks the newly created nodes, which then too are identified to require a split. This process continues until the displayed hierarchy is fully established.

A sample of an obtained cluster hierarchy is depicted in detail in figure 4.16, showing how landmarks are associated to nodes of the result hierarchy on different levels. In the

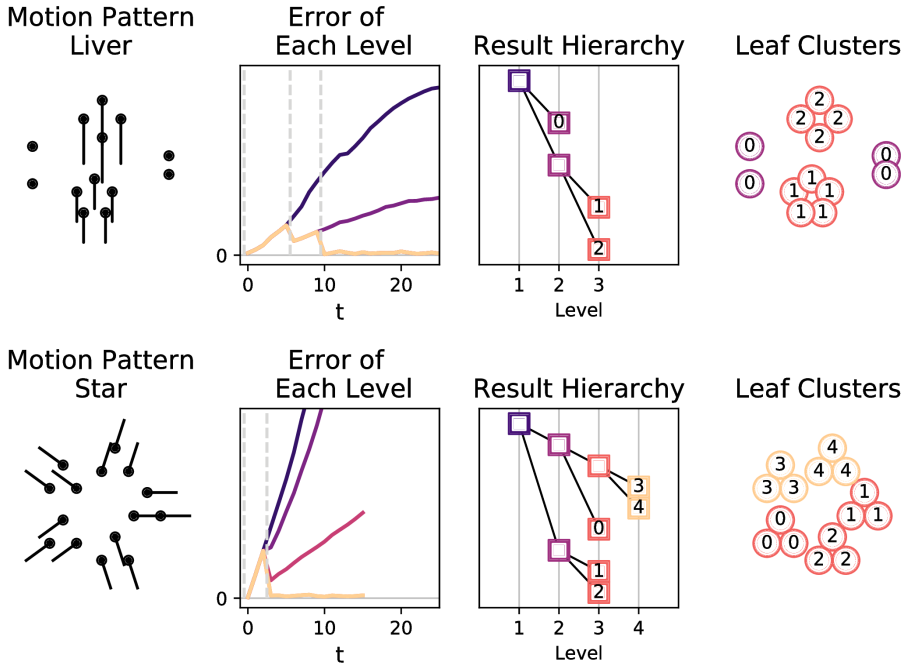


Figure 4.15.: Simulation of non-perturbed artificial motion of type *Liver* (top) and *Star* (bottom). In the resulting, correctly generated hierarchies, each leaf node corresponds to one cluster in the motion data. 'Result Hierarchy' and 'Leaf Clusters' depict the situation at the end of the simulation.

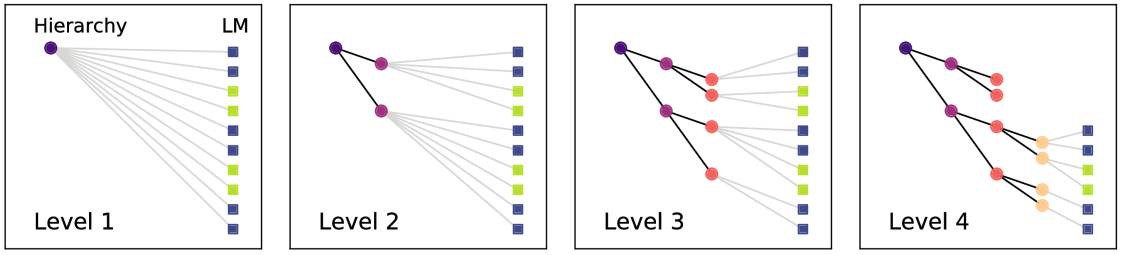


Figure 4.16.: Assignment of landmarks (LM) to the nodes of the clustered hierarchy with binary splits in the *Bend* data set. Associations to landmarks drawn, if present on level. Setting is a linear scaling of a row of landmarks in a state space of only translations.

displayed example, also only binary splits are allowed, leading to a hierarchy with nodes up to level four. The full set of ten landmarks is fully associated to levels 1-3, whereas only a subset of six is additionally associated with the fourth level of detail.

Binding Level θ_{bind} The threshold level influences at what time new clusters are generated from existing ones. Figure 4.17 displays an overview of errors obtained after generation of the hierarchy, comparing results in the individual levels. In general error decreases with increasing levels, for all motion data sets. Distinguishing a case with $K = 2$ or $K = 15$ possible sub clusters per step, shows that the adaptation in case of the former situation emerges more slowly — the full adaptation to the motion requires more levels, when the hierarchical tree is restricted to a composition of binary clusters only.

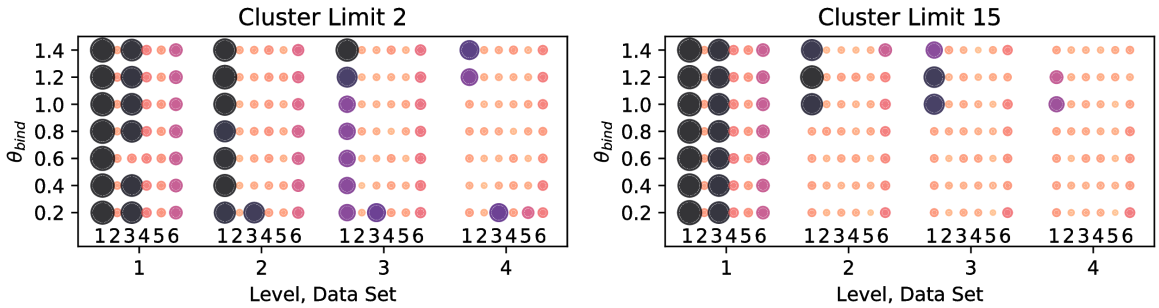


Figure 4.17.: Level (1-4) wise errors in different test sequences (1-6) and for different binding thresholds θ_{bind}. Cluster seed counts $K = 2$ (left) and $K = 15$ (right).

Influence of the binding threshold $\theta_{bind} > 1$ is shown on the ordinate. For low thresholds $\theta_{bind} < 0.4$ and binary splits (left), clusters are created late and more outliers are obtained. This effect is not observed, if up to $k = 15$ are sought. Similarly, for large $\theta_{bind} \geq 1$, clusters are created immediately and produce outliers as well. The range of $0.5 < \theta_{bind} < 1$ yields results which represent correct clustering.

With respect to the number of nodes created (Fig. 4.18, left), two observations are made. First, $\theta_{bind} = 1$ poses the expected transition threshold beyond which clustering appears immediately and in this case leads to a full fragmentation into single landmarks. Second, the smoothness of transition of the node count from regime $\theta_{bind} < 1$ to $\theta_{bind} > 1$ depends on the recoverability of the ground truth clusters. In sequences such as *Liver*, *Warp*, *Bend* and *Start* few (< 5) clusters undergo the same motion. In these cases a constantly low number of clusters is obtained in the regime $\theta_{bind} < 1$. Sequences *Heart* and *Stretch* show a scaling component which cannot well be described by few clusters. In these cases, the transition at $\theta_{bind} = 1$ is smooth.

The exact clusters generated depend on the number of iterations spent in the clustering routine (Fig. 4.18, right). For the given noise free data, the result stabilizes above $N_I^{cl} \approx 10$. With increasing number of iterations the results shows a better adaptation, such that the number of nodes required for the tree is reduced.

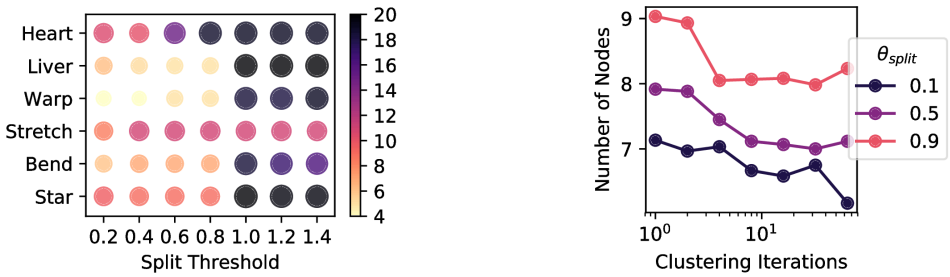


Figure 4.18.: Left: Number of nodes in the final hierarchy for particular split inducing similarity thresholds, for 6 sample motion types. Right: Average number of nodes in the constructed hierarchy, dependent on number of iterations in the clustering process.

Noise Additional effects are introduced when the data is noisy. The sample of the *Star* data set shows a split into N_C clusters of N_L landmarks each. A functioning split algorithm therefore is required to obtain the N_C clusters. Figure 4.19 shows basic properties of a structural transition induced by noise. In this, the binding threshold is set to an exemplary $\theta_{bind} = 0.9$. It shows Jaccard similarity (left) and error levels (right) as heat maps, with the two noise levels as axes. The depicted values are obtained by averaging results of 10 iterations of simulations of the *Star* dataset with 5 landmarks on each of the 5 clusters.

A structural transition is visible in the Jaccard similarity measure and it is mostly influenced by positional noise. It is most prominent for low observation noise $f_R \approx 0$: For low $\sigma_{pos} \lesssim 0.2$, similarities close to $\mathcal{J} = 1$ can be reached (A). For high positional noise $\sigma_{pos} \gtrsim 0.5$, $\mathcal{J}$ reaches its lowest values with $\mathcal{J} \approx 0.3$ (C). Between these two regimes of minimum and maximum $\mathcal{J}$, a transition (B) occurs during which from $\mathcal{J}$ decreases from (A) $\rightarrow$ (B).

The observed transition is relatively independent of f_R. For large $f_R \gtrsim 0.6$ it begins to start earlier, for smaller values of σ_{pos}. The regime of high and transitioning $\mathcal{J}$ are reduced and the minimum $\mathcal{J}$ is reached for lower values of σ_{pos}.

A very similar transition can be observed in the error measure. With low positional error σ_{pos} in region (A), the result error remains low. It increases in the proceeding regime (B) and is reduced to initial values again in regime (C). For low f_R, extreme outliers are found only in high noise regime (C). For very high observation noise $f_R \gtrsim 0.9$ these outliers exist in all regimes and dominate the results.

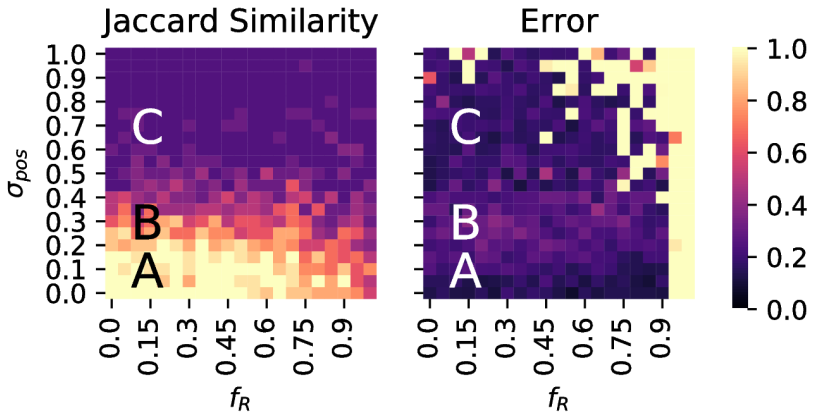


Figure 4.19.: Error level and Jaccard similarity of clustering the motion in the *Star* data set, with $N_L = N_C = 4$. Results of a single run, depending on spatial jitter and observation noise σ_{pos}, f_R, respectively. Data obtained for binding threshold $\theta_{bind} = 0.9$.

4.3.1.2. Rearrangement

Different aspects of association rearrangement to an initial case of landmark-individual nodes are investigated.

Rearrangement in the Noise Free Case Initially, a noise free case and the sole effect of direct association reinforcement f^{AR} is regarded (Fig. 4.20).

The figure shows an example of a single run of the *Liver* data set. The total number of nodes as well as the Jaccard similarity (Eq. 4.19) are depicted over time. With $f^{AR} = 0$, both measures remain the same. For values $f^{AR} > 0$, there exists a transition where the node count starts to decrease monotonously and another when the node count reaches a lower boundary. For $0 < f^{AR} < 1$, the samples show the correct number of nodes (3) being reached. In these cases, the node count transition points coincide with transition points of the Jaccard similarity. The similarity starts to increase as nodes are reduced and reaches $\mathcal{J} = 1$, once the correct three clusters are left. The higher f^{AR} the earlier the transitions occur. A reciprocal relationship between the f^{AR} and the time of both onset and finish of the transition is observed (Fig. 4.21). For a larger $f^{AR} = 10$, the transitions occur as rapid changes, implying fast adaptation. However, once a cycle of the liver motion is completed $t \approx 50$, the three existing clusters are joined into two, accompanied by a drop $\mathcal{J} : 1 \searrow 0.6$. After this transition, the number of nodes remains the same, but $\mathcal{J}$ shows a cyclic behavior, alternating between 0.5 and 0.6 along with the motion cycle.

Measuring adaptation times dependent on the reinforcement strength, leads to results as depicted in figure 4.21. Here, the time of *onset* is defined by the point when the hierarchy is adapted for the first time, i.e. when the first pair of single landmark clusters is joined into one of two landmarks. The *finish* time represents the time, the number of

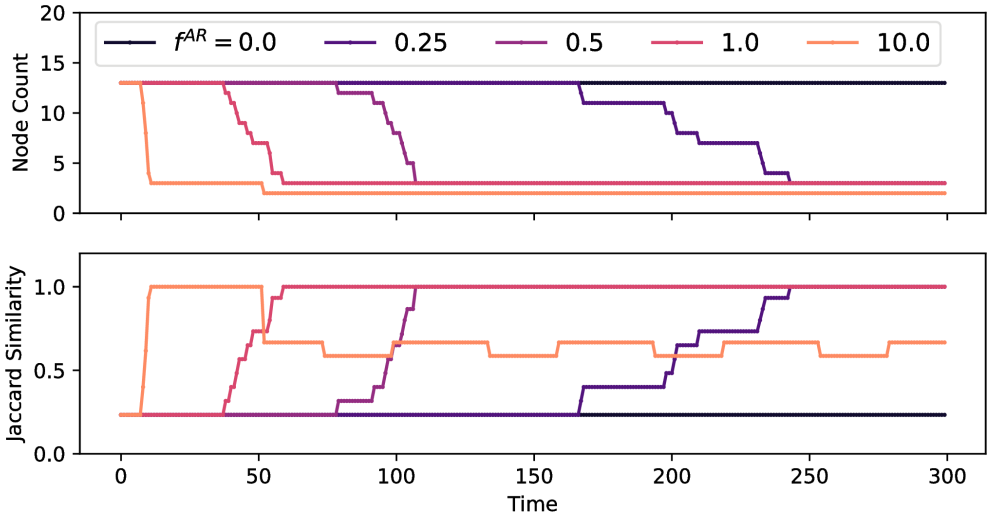


Figure 4.20.: From a start of completely individual landmarks, rearrangement of associations leading to the formation of clusters. The adaptation process is influenced by the rearrangement rate f^{AR}. Settings: *Liver* set, no noise ($\sigma_{pos} = 0$, $f^r = 0$), no attractive reinforcement.

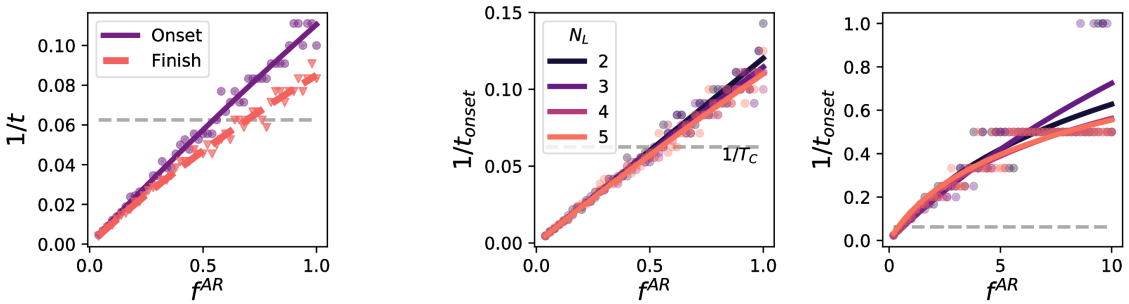


Figure 4.21.: Left plot: Fitted onset and finish time for the cyclic *Star* dataset, depending on reinforcement f^{AR}. Right plots: Inverse onset time observed for different number of landmarks in each cluster of the *Star* dataset. The figures show different ranges of f_{AR}; for small f_{AR} a linear relationship $1/t_{onset} \propto f_{AR}$ may be used as an approximation to the data.

nodes is identical to the number of ground truth clusters. The left figure 4.21 shows how the adaptation rates $1/t$ for onset and finish depend on f^{AR} in the range $[0, 1]$. They can both well be fit by the non-approximated adaptation rate (Eq. 4.17), in which the quality q and drop ϵ are used as fitting parameters. Focusing on the onset time, the right plots show the regimes $f_{AR} \in [0, 1]$ and $f_{AR} \in [0, 10]$. For small f^{AR} the linear approximation in (Eq. 4.17) is justified. Furthermore, the adaptation rate is mostly independent of the number of landmarks present in each cluster. For small f^{AR} the overlap between the individual curves is visible. A slight divergence is visible for rate higher than $1/T_C$, in which the adaptation appears faster than one motion cycle.

Rearrangement in the Noise Case Still without attractive reinforcement, two types of noise are introduced to the scenario. Jitter noise (σ_{pos}) and observational noise f_r show different effects on the adaptation process. The adaptation is depicted by the Jaccard similarity $\mathcal{J}$ (Fig. 4.22). Among three plots for $f_r \in \{0, 0.5, 1\}$, three graphs distinguish $\sigma_{pos} \in \{0, 0.03, 0.5\}$. With $\sigma_{pos} \in \{0, 0.03\}$, the adaptation is possible in any case. Noise value $\sigma_{pos} = 0.03$ is however close to an upper boundary $\sigma_{pos} = 0.05$, where no adaptation[5] occurs for $f_r = 0$. For $\sigma_{pos} = 0.5$, no adaptation occurs at $f_r = 0$. The introduction of very strong observation noise $f_r = 1$ (e.i. $\sigma_{A'} = \sigma_A \mathcal{U}(0, 2)$) leads to a slower adaptation. As a result of strong observation noise, some landmarks are temporarily invisible to the Particle Filter, which means that some clusters cannot provide useful proposals for both others and themselves. In this case, the onset of the adaptation also begins earlier.

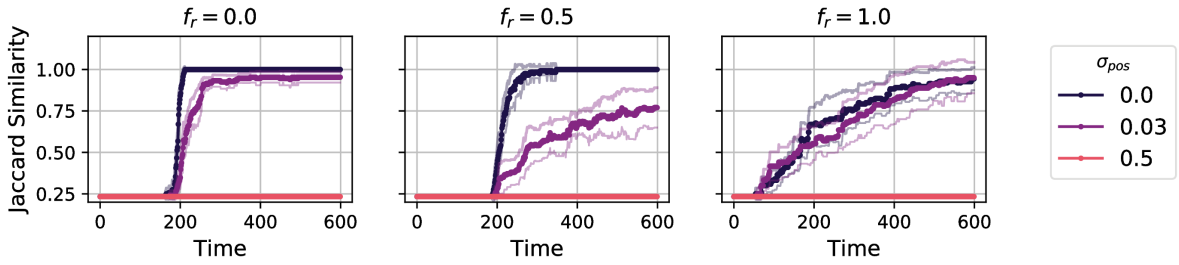


Figure 4.22.: Sample cases of introduction of noise into the case $f^{AR} = 0.05$. Observation noise cases juxtaposed, positional jitter noise $\sigma_{pos} \in \{0, 0.03, 0.5\}$ as colored graphs. Settings: *Liver* set, no attraction, 10 repetitions, shown as mean (thick) and standard deviation (thin line).

Rearrangement and Attraction Enabling the attractive term during motion, associations to clusters with more association receive stronger reinforcement than competing associations to clusters with few.

[5]This is not depicted. Instead a value $\sigma_{pos} = 0.5$ (even more clearly in the non-working regime) is shown.

Figure 4.23 shows the observable behavior in noise free conditions (left) and with different degrees of noise added (right). In the noise free situation, the linear relationship for low reinforcement f^{AR} and adaptation rate $1/t_{onset}$ is observed. For the given artificial data, the relation of rate to reinforcement is $m = 0.114$ without and $m = 0.226$ with the attractive term enabled; the attractive adaptation process is twice as fast as the reference.

The introduction of position noise interferes with the adaptation process (right, heat maps). Without the attractive term (top row) low values of noise prolong the adaptation process. In the sample data, no adaptation is observed for $\sigma_{pos} > 0.2$ within the 900 steps of the experiment. This shows as no onset time being determined (white space) and stable, low Jaccard similarity.

The attractive term allows adaptation at higher noise values (bottom). It is successful for all depicted noise levels $0 \leq \sigma_A \leq 1$. Similar to noise free data, the onset appears earlier than in the non-attractive version. However, at higher levels of reinforcement (here $f^{AR} > 0.3$) over fitting appears and clusters are reduced to fewer clusters than actually are present. This shows as a reduction of $\mathcal{J}$ after for $f^{AR} > 0.3$.

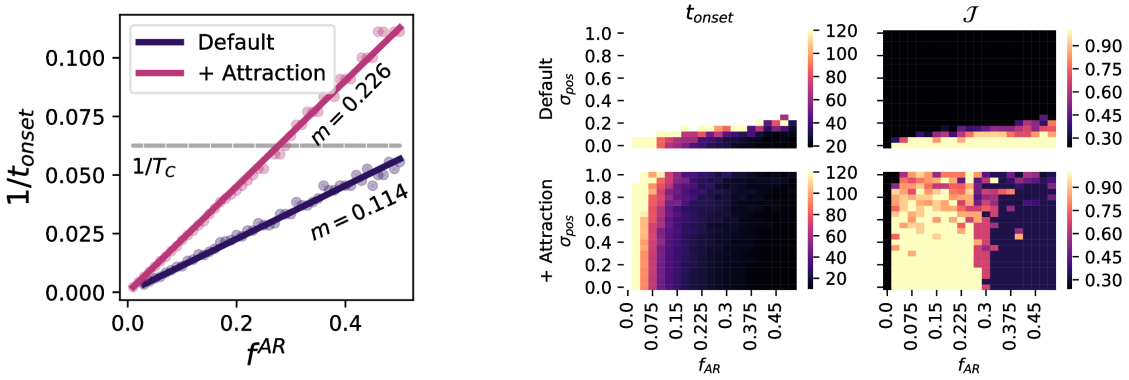


Figure 4.23.: Left: The initial adaptation rate is increased by adding preferential attachment; depicted in a noise free situation. Right: Effect of rearrangement f^{AR} under spatial jitter noise conditions σ_{pos} shown as starting time of adaptation ('onset time') and final Jaccard similarity, in the *Liver* data set. Top: Default adaptation case. Bottom: Attractive reinforcement enabled. White space in 'onset time' corresponds to no onset during the simulation of 900 steps.

4.3.1.3. Joint Algorithm

Enabling both the clustering process and association rearrangement yields additional options to configure the algorithm. Three exemplary cases are compared here (Fig. 4.24). The left of the figure shows a summary of the average J over the time course, whereas on the right three exemplary single runs are depicted by their level wise error over time.

All cases share a common setting for association reinforcement with $f^{AR} = 0.05$. Two cases show the *individual root* starting condition, where each landmark exists in an own

cluster. The two cases differ in the application of clustering. In both of them, an initial 'warm up' phase $t < 70$ is observed during which $\mathcal{J}$ remains low as well as the errors on any level. To be precise, there is only one level as the hierarchy has not been altered. Errors above level one are 'fall back' results of level 1, and therefore identical. After this phase, $\mathcal{J}$ and the error of level 1 increases. In case clustering is employed, nodes on new levels occur and show a lower error than the now increased level 1 error, but reduced by the newly added levels.

The third case uses the 'unified' starting condition, where landmarks exist as a joint cluster. In this setting, the Jaccard similarity reaches 1 within the first 30 steps. Level 1 error is drastically higher.

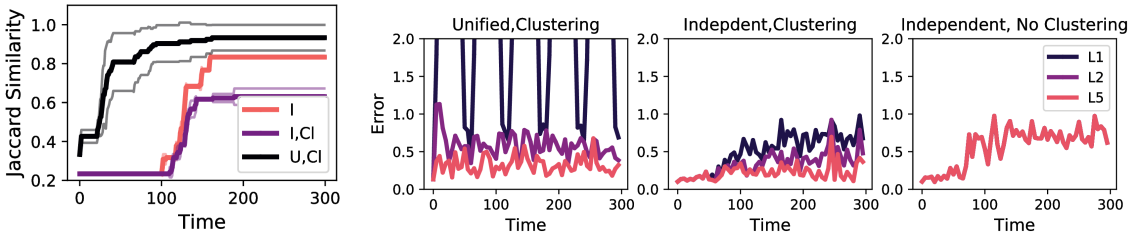


Figure 4.24.: Three particular combinations of settings. I/U: Independent/Unified initial root layout. CL: Clustering ($\theta_{bind} = 0.9$), Attractive reinforcement with $f^{AR} = 0.05$. Left: Jaccard similarity 10 average experiment with $\theta_{bind} = 0.9$. Right: Error/Level in a single experiment.

4.3.2. 2D MRI Data Setups

4.3.2.1. Rearrangement

The rearrangement algorithm is applied to EPI image data and the adaptation process monitored as number of nodes remaining from an initial situation with *individual* landmarks. Left hand figure 4.25 compares the time required for rearrangement within a single sample dataset. The two cases of plain rearrangement and attractive rearrangement are compared. Time $N_C < 100\%$ describes the time of the first clusters merging with another. Linear regression results are overlayed to the data. Both cases with and without attraction show similar onset times. Slopes without and with attraction are of comparable order (0.369 ± 10^{-3},$0.336 \pm \cdot 10^{-3}$). The time of less than 70% of initial nodes remaining, differ strongly. Without attraction, the adaptation rate's slope is reduced to 0.047 ± 10^{-3}, whereas with it remains at 0.298 ± 10^{-3}. Furthermore, without attraction, for $f^{AR} < 1.3$ the number of nodes does not fall below 70% during the 300 steps of the data set.

Instead of the adaptation rate, the right hand figure 4.25 shows the number of nodes as a function of time, dependent on reinforcement strength f^{AR}. More clearly than the left it shows that attractive reinforcement has a much more pronounced adaptation

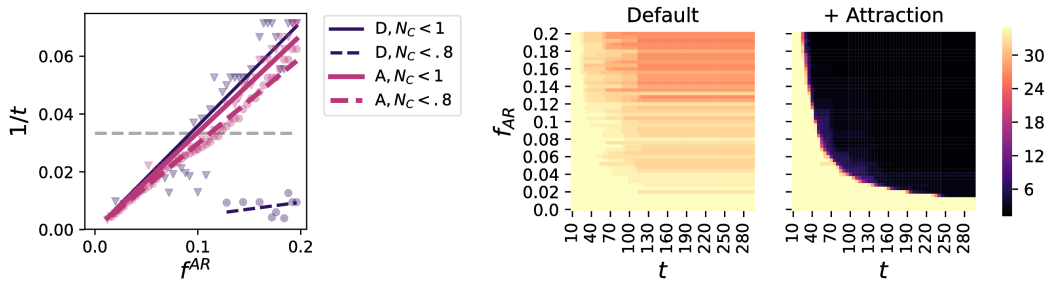


Figure 4.25.: Left: Comparison of adaptation rates in a single EPI dataset, when starting with individual landmarks. Times $N_C < 100\%$, $N_C < 80\%$ refer to the step where the number of individual nodes first drops below the related percentage. Right: Adaptation as number of nodes at times t, dependent on reinforcement f^{AR}.

process. While the onset times are comparable in both cases, the default case does not transition into the regime of very few clusters. Default rearrangement reduces clusters from approximately 35 to 25 over many time steps, whereas attractive rearrangement lowers the number to less than 5 in a few steps.

Executed for multiple data sets, the ratio m of adaptation rate and reinforcement strength was found mostly independent of the sets. Particularly for attractive reinforcement m is found in the same range $m \approx 0.30$.

4.3.2.2. Joint Algorithm

Error Level With the automatically generated second level of the hierarchy, alternative error levels are obtained with respect to the first. Figure 4.26 shows errors on second and first level subtracted from each other. In the majority of cases, the change of error between the levels is negligible ($-0.02\,\text{mm}$ mean), with levels between $\pm\ 0.5\,\text{mm}$.

There exist outliers, which increase the error levels strongly. In data set 23 there exist two landmarks, which are on the outer rim of liver and field of view. The related

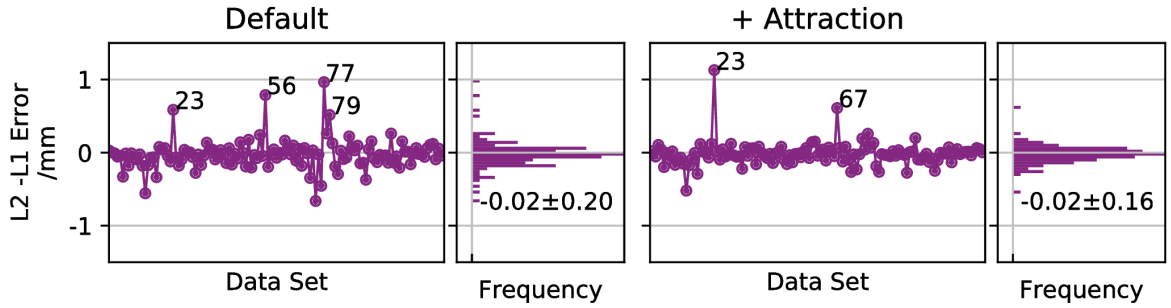


Figure 4.26.: Comparison (difference) of level 1 and 2 error, with and without attractive term. Clustering at $\theta_{bind} = 0.9$, reinforcement $f^{AR} = 0.05$.

blood vessels are small, such that the region is misidentified to belong to the static outer component. Similar issues occur in sets 77 and 79, where small landmarks on the outer liver sections are only temporarily visible. In this case, they are joined into a separate cluster, but the cluster is less stable than the root component.

Patch based NCC The left of figure 4.27 shows the changes of the patch based NCC results for all datasets: The change of NCC is computed as the mean NCC for a zero-deformation vector field subtracted off the respective deformation vector field of the algorithm. Other than the error measure, which is only available for the annotated liver section, an NCC value is averaged over every tracked landmark and frame.

The major increase in NCC is already obtained on the first level (black dots). While the average reference NCC without correcting vector field is found to be 0.52 (not shown), the first level is able to increase the value by 0.2 on average. It can clearly be seen that in every case, the second level of the hierarchy is able to increase the NCC measure between single patches. The default rearrangement (triangles) process generates level 2 clusters, which increase the similarity more strongly than the attractive one (circles). The right side of the figure shows the increase for further levels, summarized by mean and standard deviation. With increasing number of levels, both cases of rearrangement approach a maximum value around a mean at ≈ 0.23. Attractive rearrangement however approaches it more slowly. Major improvements are found on levels 2 and 3, whereas a fourth level does not increase results much further.

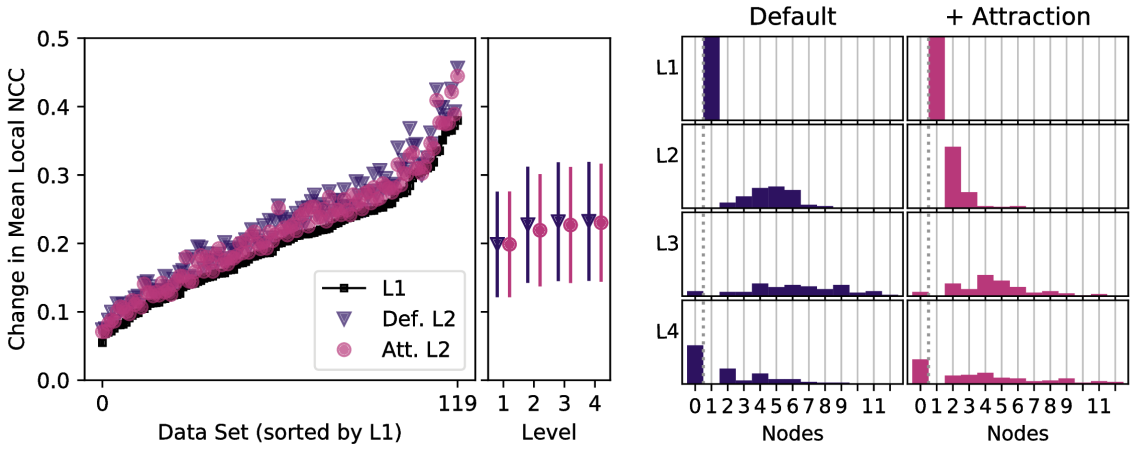


Figure 4.27.: Left: Change in normalized cross correlation between reference and current patches (radius 8 vx). Values of 120 datasets are sorted by the L1, the root level baseline for both optimizations. For comparison, level 2 values with default reinforcement (Def. L2) and attractive reinforcement (Att. L2) are shown. Right inset: Mean and standard deviation of level 1 and 2 summarized. Right figure: Histograms of the number of nodes on each level.

The right hand side of the figure shows properties of the related result hierarchies as histograms of the number of obtained nodes per level. Samples in the histogram are given by the final hierarchies of all data sets. After the trivial result of a single level 1 root node in both cases, the major difference is seen on level 2. In the default case, a wide distribution around $4-5$ nodes on average is built. Attractive rearrangement on the other hand leads to a pronounced peak, where almost all datasets show either 2 or 3 nodes, with the majority ($\approx 80\%$) showing two. Both distributions spread further on the third level. This level is also the first level to exhibit cases of 0 node being present. This effect continues on level 4 approximately 40% (default) and 30% (attractive) of the data sets do not produce nodes of this level of detail.

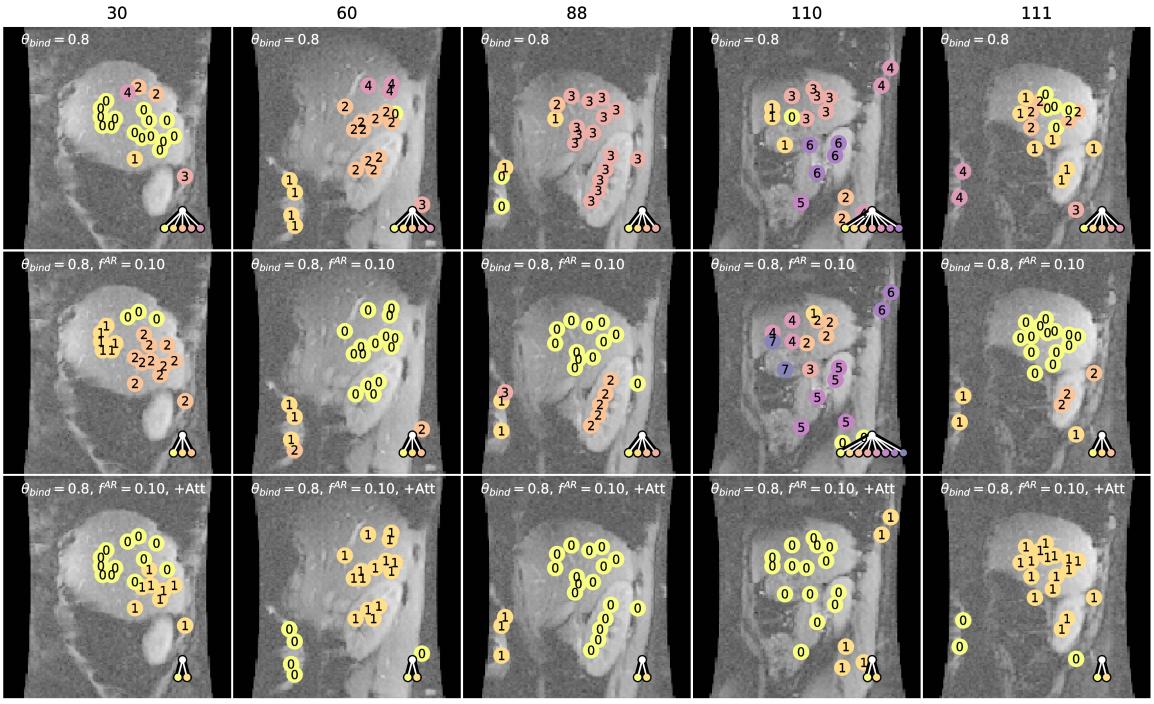


Figure 4.28.: Effect of three different modes (Clustering only, clustering and reinforcement, clustering and attractive reinforcement) on the establishment of second level clusters on 5 sample data sets (columns). Showing labels after running the full dataset (289 steps), drawn into initial landmark position, overlayed to reference frame. The bottom corners shows the first two levels of the hierarchical tree.

Qualitative Effect of Rearrangement Process The effect of association rearrangement is demonstrated by samples of second level association results in figures 4.28 and 4.29. These show the obtained cluster associations as labels on the landmark positions in the reference frame. The EPI shows moving liver and kidney in the center. Anterior and posterior abdominal wall are shown left and right of them. In some cases, a therapeutic transducer mock up is attached to the anterior abdominal wall. The automatic

feature detector mostly identifies posterior wall, liver, kidney and transducer as points of interest. The anterior abdominal wall is rarely assigned landmarks.

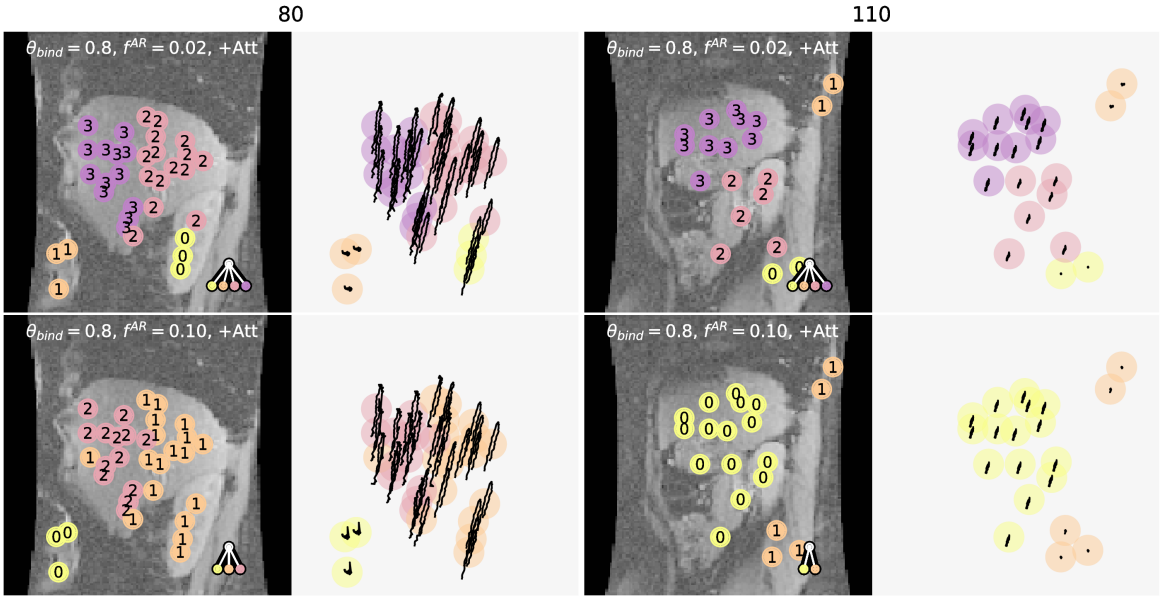


Figure 4.29.: Two sample cases 80, 110 with different strengths of attractive reinforcement ($f^{AR} \in \{0.02, 0.1\}$) showing the motion of the obtained clusters within the second half of the time series.

Figure 4.28 depicts results of three algorithmic combinations, in which clustering is executed at $\theta_{bind} = 0.8$. It is run as such (case 1, top), combined with default (case 2, center) and attractive (case 3, bottom) reinforcement at $f^{AR} = 0.1$. The three variants differ in the number of clusters which exist on level 2 at the end of the sequence, which is depicted in the figure. While case (1) keeps the clusters as they are produced, (2) shows a reduced and rearranged result. Not only are fewer clusters obtained, but they are also locally more coherent. This is particularly visible in set 30, where the liver is clearly separated into three distinct regions of movement.

The introduction of the attractive reinforcement term underlines this behavior. In all sample cases, the second level is separated into exactly two components. In cases 60,88,110,111 where a static component (abdominal wall, transducer) has been selected by the feature detector, this component is split off the moving liver/kidney cluster. Case 30, in which only liver landmarks are selected, shows a split of the liver into two sub components.

Figure 4.29 depicts the actual motion of detected clusters for two sample data sets at two levels of attractive rearrangement. In both cases, the static and moving component are correctly identified. For the weaker $f^{AR} = 0.02$, four clusters remain at the end of the sequence — two within the moving and two within the static component. With the stronger $f^{AR} = 0.1$, the static portion in both cases is joined into a single cluster.

The moving component in case 110 also is joined. However, the moving component in case 80 continues to be modeled by 2 clusters. The reason for this is found in a higher amplitude of motion and a scaling component of the liver which cannot be described by a single cluster.

Hierarchical Tree Figure 4.30 depicts two levels of an evolved hierarchy on data set 110, which demonstrates the increasing detail. The root level is not depicted[6]. On the second level three components have been separated (1) Liver, (2) abdominal structure underneath and (3) static spine region. The third level separates each of them into two components. The liver is separated into a major liver section an one larger, pulsating vessel in the upper region. Both the lower abdominal and the spine region are separated into their upper and lower parts.

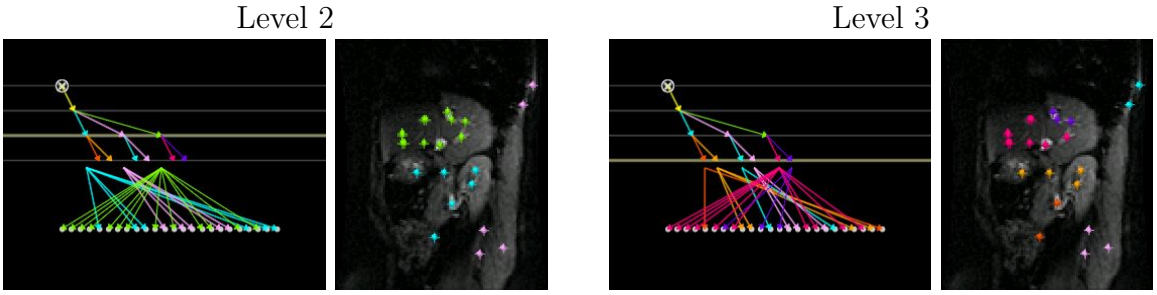


Figure 4.30.: Hierarchical tree and deformation vector fields for levels 2 and 3 (left to right) in a three level model, as visualized in a MeVisLab implementation. Each left image shows the hierarchical tree, the select level (thick horizontal line) and landmarks associated to the level underneath. Motion clusters are color coded.

Figure 4.31 compares two result vector fields obtained in chapters *Fixed Hierarchy Model* (3) and *Dynamic Hierarchy Model* (4). Results already differ on the first level, since the two-level model allows for scaling, whereas the dynamic model for consistency is reduced to translations only. On this stage, motion of the liver is roughly captured by both, but the position estimates for the transducer in the left corner are incorrect. The second level of the two-level model is able to capture the position of the transducer correctly. However, due to the conflicting motion over liver and transducer, the quality of the global prior deteriorates. The individual landmarks on the second stage show more jitter than the still clustered regions of the dynamical model. In this case, the motion of transducer and liver is separated into individual components. Furthermore, the missing scaling transformation is compensated by the presence of left and right liver clusters. Here, the highest detail of the dynamical model is reached on level 3, which still shows clusters of up to three landmarks. In the presented sample, both algorithms fail to detect one single landmark in the spine region as static. The reason is that the patch integrates structure of the kidney, which dominates the detection.

[6]For a depiction of a root level please see figure 4.31

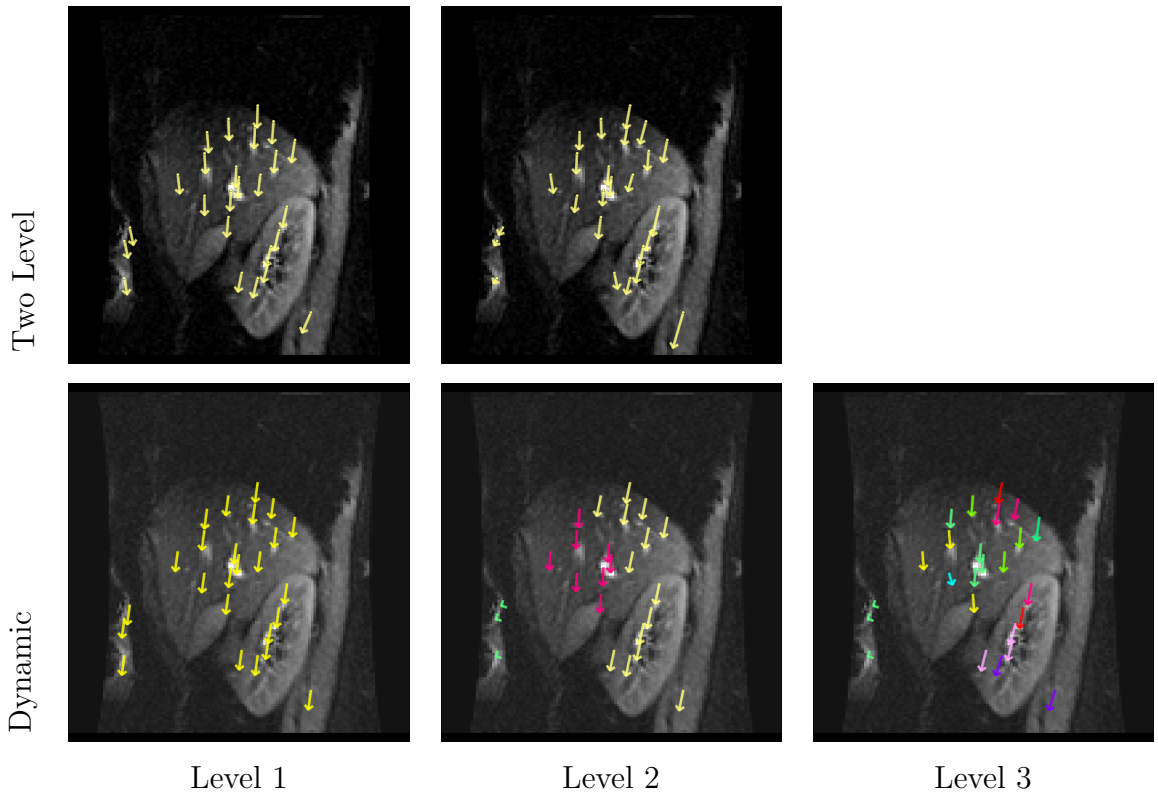


Figure 4.31.: Sample comparison of two-level model (top) and dynamic model (bottom). The two level model shows global level 1 (left) and local level 2 (right). The dynamic model shows levels 1,2 and 3 (left to right). In the dynamic model, colors indicated cluster labels. While the two-level model uses a global scaling transformation, the dynamic model composes the scaling solely from multiple translated clusters.

Computation Time The computation time is linear to the number of tracking units involved in the process. In this section the computation time on EPI image was fit as 5.3ms + $0.82N_A$, where N_A is the number of strong associations established. Since in the setup, 20 landmarks are involved on root level the total number of associations on second level is 40. This number can be found higher, if nodes on the third level are created. As in chapter *Fixed Hierarchy Model* (3), the computation time depends on the parametrization of each a tracking unit. The major computation time of the algorithm is taken by tracking the single nodes, and therefore strongly depends on the number of strong associations. The offset of 5.3ms is related to maintenance processes, which do not scale linearly with the number of strong associations. These include data preprocessing, and maintenance of the hierarchy by clustering and rearrangement.

4.4. Discussion

This chapter presented an algorithm for generating and maintaining a hierarchical tree layout for tracking articulate motion in medical images. The different aspects of the results are grouped into findings on the single components as well as common findings.

4.4.1. Association Clustering

Initially a clustering algorithm is presented which determines the necessity of splitting a motion cluster into further sub-cluster. On unperturbed artificial, the algorithm is able to recover the ground truth clusters of motion. It can be used to sequentially adapt to an increasing level of differentiation in the motion patterns of the data; e.g. it is applied as a tool to construct a hierarchical representation by repeating partitional clustering. The performance of the algorithm is deteriorated in the presence of strong noise, which will be discussed separately.

In the literature clustering has been applied to particles in a single set. Often the aim is to determine modes in a singular particle set. The goal may be the prevention of sample impoverishment [SMW02; YA05] by explicitly keeping modes or to infer the presence of multiple individual targets from them [RVG04; Cla+07] — particles of the same set are clustered by their state. In particular to applications of the Probability Hypothesis Density filter, clustering of modes may be required [PVS05]. In this work however, *particle sets* are clustered among each other by their *weight vector*. These sets share a common set of state vectors, but as they are associated to individual landmark, differ in their weights. To our knowledge this is the first work to cluster trackers by their individual responses to the same scene. In the investigated scenario, few landmarks are required to be clustered quickly. It was found to converge in approximately 10 iterations, but applied with 100 iterations without remarkable loss in performance. The major part of the computation time is spent for the tracking nodes of the hierarchy. The algorithm shares similarities with spherical k-means [Buc+12], such that measures are based on cosine similarity. Other than clustering entities which are already separable, the algorithm aims to identify the formation of clusters as the transition from a unimodal distribution to K^* individual ones. This also is the reason for not minimizing cosine dissimilarity to cluster centers, but to maximize distance to complement cluster centers. Further, a relation to correlation clustering can be seen. Correlation clustering partitions complete graphs by specifying the relationships between entities. These relationships, specified in the edges may be of discrete [BBC04] or general weighted nature [Dem+06]. Other than the method in this work, correlation clustering does not rely on cluster centroids. More general than this work, it can freely determine the optimal number of clusters and therefore poses an alternative implementation to this work. However, its complexity scales stronger than linear with the number of cluster entities [Cha+15b].

In the proposed implementation, the binding threshold of a cluster is chosen a constant which relates to the reference similarities within the set of landmarks to be clustered. Although it is a relative value already, cases have been observed in which for instance either levels of $\theta_{bind} = 0.5$ or $\theta_{bind} = 0.8$ produced better outcome. As lower θ_{bind} causes

later splits, a larger portion of motion may already be visible in the data, leading to better results. However, an overly conservative threshold may also prevent the creation of new clusters entirely. Thus, the choice of the binding threshold depends on the aim of clustering: If major differences in motion are to be detected, a lower threshold should be chosen. A higher binding threshold ensures that most certainly clusters are created; even if they do not represent entirely different motion.

A particular feature of the proposed clustering process is that K does not define the result number of cluster, but an upper limit. In the presented examples, the number of clusters is never overestimated in the initial clustering step[7], even though more centers are offered. This is an effect of assigning entities to the farthest complement centers. If multiple very similar complement centers are present, only one of them will receive all of the assignments. The reason is that the complements are by definition distant from the clustered entities and in direct competition it is likely that only on is the farthest.

The complement centers define the seed points of Voronoi-like cells for label assignment. Since the hard decision is required to reduce the number of result clusters, a straight forward EM implementation would not produce a comparable result. Both, computation of the mean center and definition of the complement center $\vec{c^a} = \vec{1} - \vec{c}$ are arguable. The feature space is not Euclidean, due to the normalization and positivity of the weights. Furthermore multiple dimensions of the input vectors (measurement results) are likely to be correlated. Further research may be required to refine both an appropriate definition of cluster and complement centers.

Some extensions to the clustering algorithm are straight forward to implement. The data of multiple tracking steps can be used as clustering input by simple concatenation. Such multi step joins might also add a time dependent weighting into the distance measure. The introduction of a spatial component into the clustering input may also be thought of. However, a mixed similarity measure for distances and state space measurements would have to be defined. Aside from spatial distances, properties of the landmarks could be matched. For instance, in ultrasound imaging landmarks which fall on similar tissue structure could be promoted to join the same clusters. An alternative approach to the clustering problem is found in using the full similarity matrix of all landmarks to derive the full tree from. This, however, poses the question, how to maintain the similarity matrix, since the hierarchical tracking process only provides conditional similarity measures between sibling node landmarks and not the entire set. If such approach was available, it would have the advantage of allowing a permanent, long-term storage of cluster information in the global matrix, whereas the proposed algorithm loses such information if a set of child clusters is removed.

4.4.2. Association Rearrangement

The motion free case of plain, attraction free, association rearrangement ($f^{AR} > 0$) shows the desired behavior of clusters forming around the actually present components of motion for the three-component example. The duration for the formation is strongly

[7] splitting of the hierarachy's root node

related to f^{AR} itself. For an excessively large values ($f^{AR} = 10$), a rapid adaptation can be observed, which however is followed by an over-adaptation: Once the motion returns to the reference state after the first cycle, the two motion components are joined into a single. As they are joined they are no longer able to recreate the third component. A rearrangement process still takes place between the two remaining clusters. This shifts single, uncertain associations between the remaining components, with the periodicity of the motion cycle, which explains the periodicity in the Jaccard similarity measure. For reasonably small reinforcement rates ($f^{AR} < 1$), the expected adaptation occurs as result of the expected relaxation process. Clusters of the sample dataset are recovered and free from over adaptation.

For the exponential relaxation the approximating reciprocal relation between adaptation time and reinforcement strength f^{AR} was found justifiable. This implies that the adaptation process is mostly independent of the concrete data. While it does differ for environments (e.g. artificial and medical data), within these, the relation of adaptation rate to reinforcement was found to be relatively stable. Hence, it may be measured from few sample data points and then be interpolated to fit the needs of the application. If there exist landmarks in a state explainable by identical transformation, they will be joined after a typical time, prescribed by the adaptation rate. This means that, if the system returns to the reference state and remains there sufficiently long, the differentiation between clusters will be lost. On the other hand, if the system remains in a differentiated state sufficiently long, clusters can be optimized to fit these. As the former is an unwanted effect, it yields an upper boundary to the adaptation rate. A higher rate on the other hand is favorable for faster adaptation to the data — a trade off to be considered during parametrization. For the intended application, the interpretation of rearrangement as a 'repair' process is intended, and small values are favored.

4.4.3. Impact of Noise

The integration of noise in artificial data was implemented as jitter noise σ_{pos} and observation noise f_R. Noise affects both the clustering and the rearrangement procedure. With increasing levels of noise, faulty results are obtained. In this, for both clustering and rearrangement, jitter noise was found to have a more severe impact on the result than observation noise: To the algorithm it is more disrupting if a landmark's likelihood peak is in an unexpected place than if it is of unexpected width.

The response to noise was evaluated at different binding levels. For small binding thresholds, a noisy environment leads to equally sub-optimal results for high and low levels of jitter noise. This is an effect of the threshold choice itself rather than the noise level. Differentiated behavior is observed for sufficiently large θ_{bind}, where independent of σ_{pos}, low errors are obtained: For low σ_{pos} the error remains low since the cluster creation performs as intended. For high σ_{pos}, the sensitive θ_{bind} leads to a result in which all landmarks exist in individual clusters. On artificial data, this leads to a lower error, since each landmark is tracked individually. However, it will not help stabilizing the tracking process, as no hierarchy can be built from it. The transition between these extreme regimes can clearly be observed for positional noise. There also exists a similar

transition for observation noise, but it occurs as very high relative changes $f_R > .6$. These implies random changes of within $\pm 60\%$ of the measurement peak's standard deviation. Therefore, major sensitivity is observed for position noise. Similar to noise free data, there exists a trade off on the choice of θ_{bind} in noisy data. It must be chosen sufficiently high for an early and meaningful result. Choosing it too high (sensitive) may also deteriorate the result, as in noise more clusters than necessary will be detected. In practice values between $\theta_{bind} = 0.5$ and $\theta_{bind} = 0.8$ were found useful. After all, the clustering process relies on short term correlations. If these are hidden in spatial jitter noise, the long term analysis of association reinforcement is required.

For association reinforcement, adding spatial and observation noise shows differing effects on the adaptation process. The addition of random noise to the with σ_A of the individual likelihood peaks, does not prevent an adaptation, even for $f^r = 1$, where σ_A may alter up to $\pm 100\%$. Spatial noise shows to have a much more severe influence on the adaptation. Values of $\sigma_{pos} \approx 0.03$, being small with respect to the applied $\sigma_A = 2$ can decrease the adaptation rate. Large values $\sigma_{pos} = 0.5$ may completely impair it.

This effect is a result of a type of reinforcement which strengthens associations when a proposal of another node matches just as well as the proposal of the node the landmark is currently associated to. Without jitter noise, weights gained by proposals of nodes of the same cluster are of identical magnitude[8]. Once noise is involved, the proposal of a non-associated node in the same ground truth cluster will inevitably be smaller than that of the associated (Fig. 4.32). The algorithm does not anticipate any amount of position noise and consequently interprets it as actual different motion. Hence, the sole application of the rearrangement process is not able to cope with overly large amounts of positional noise. A solution to this problem has been found in an attractive rearrangement term which favors larger clusters to smaller ones. The particular implementation of attractive reinforcement was found to not only increase the adaptation rate, but also reduce the number of resulting nodes on the second level of the hierarchy.

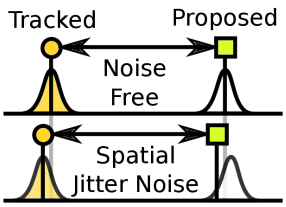


Figure 4.32.: When the result of a tracked landmark or cluster is used to propose the position of another landmark, the proposal may receive a low weight if there is spatial noise present.

Interpretation of Jitter Noise Spatial jitter noise is a major inhibitor of a proper performance to the algorithm. It hampers clustering and delays the rearrangement

[8]If the individual nodes are well localized, the proposals are essentially identical.

process. The effect it models is an entirely independent random motion of landmarks. In this respect it can be regarded as a drastic case of unknown physiological motion.

For a proper function of the algorithm it is important that competing local motions remains sufficiently low with respect to the larger component. Only then can the correct major components be detected on first levels of the hierarchy. If it is unfeasible to achieve this situation for a short time, a more conservative clustering threshold must be chosen, such that the global component may increase in strength before a split occurs.

The problem may be reformulated, in order to be solved. If there exists such a local random component, then the measurement functions of the individual landmarks may be chosen wider; for instance by spatially smoothing the filter function for each landmark. This transfers a portion of spatial noise into observation noise and may therefore stabilize the clustering result.

4.4.4. Joint Algorithm

The joint application of association clustering and rearrangement was examined on artificial and EPI image data. The former case showed qualitatively that the initial setting to a unified root leads to the fastest adaptation. The reason for this is the clustering algorithm being designed to react to unforeseen short term changes in the data. Rearrangement, operated with a low reinforcement rate, acts as a longer term corrective process.

Applied to EPI image data, a stable relation of adaptation rate to reinforcement strength was found for association rearrangement. The joint algorithm could not be shown to significantly reduce spatial error from first to second hierarchical level. This is however largely related to the fact that an annotation only existed for one component — the liver — which happens to be the dominating cluster. Thus, in the majority of cases, the root will already undergo the motion of the liver, such that the split on the second level is more likely to improve error measures everywhere else instead. Such improvement could be shown using the patch based NCC measure of the full set of points. Due to changes in patches is only possible to interpret relative changes of the measure[9]. These however were in every case positive, which confirms a better alignment of motion component outside the liver region.

A major portion of the computation time depends on the evaluation of single nodes. This portion scales linearly with the number of samples, iterations per step, strong landmark associations and descriptive points per landmark. All of these pose possible variables for optimization. Herein, the number of strong association can indirectly be limited by setting a maximum level for the hierarchical tree. The presented configuration can be applied in real-time with computation times between 50 and 100 ms without parallelization.

[9] After the initial frame, is practically impossible to obtain NCC values of 1 due to these local changes and noise.

Relation to Unsupervised Learning Both the clustering and rearrangement process of associations are means of unsupervised learning. The clustering algorithm is used to determine independent motion compartments from a current cluster and to evaluate the necessity of a split. The association rearrangement process manages already split components. It is strongly related to Hebbian learning. As stated before, freely reinterpreting Hebb [Heb50; Sha92] this is: *What can be explained together, joins a common cluster together.* If a cluster can provide a viable proposal position for a landmark, their mutual association is strengthened. Instead of updating synaptic weights, this algorithm updates soft associations. The reinforcement strength f^{AR} is directly related to the Hebbian learning rate. The additional normalization after update is also found as the motivation to Oja's rule [Oja82]. However, in Hebbian learning the reinforcement is proportionate to the product of input and output activity. Here, this term originates from a measurment, which has a potentially non-linear dependency on the input data. Moreover, there exists no neural update as in a Hebbian feed forward network. The resulting soft associations are used differently: They are not used other than to determine the highest value among them. This then decides over the dynamics. This winner-takes-all rule translating soft to hard associations may be interpreted as a tool of competitive learning.

4.4.5. Hierarchical Model

The hierarchical scheme has been found a suitable description of the occurring motion. By design it is required to identify components of large motion first and refine it with increasing levels. Only then, a benefit from the hierarchical structure can be gained.

The proposed rules for hierarchy maintenance suggest every landmark to have a unique path of strong associations from root to a leaf. In particular, it is not possible that multiple siblings own a strong association to the same landmark each. This approach shows benefits and drawbacks alike.

First, if a parent level clustered correctly, the rule prevents a landmark from being assigned to any other child of a parent than its own. In particular, there may be situations in which another parent coordinate system produces a seemingly useful position estimate just because there exist landmarks which look alike. Then, the higher level rule allowing only available landmarks to be propagated, prevents a random switch to another parent — the landmark is simply not available to them.

If on the other hand, the parent level is clustered incorrectly and unable to recover, an unfortunate situation may arise. In this, the child level would suggest moving the landmark to the correct parent, but is prevented from it by the availability rule. This is depicted in figure 4.33. The left hand side shows the separation of a sample case of four landmarks into three ground truth components using two stages. Such may be observed if one component (1) splits off the others (23) before a second split of (2) and (3) into own components appears. In the depiction, component (2), consisting of two landmarks (B) and (C) is correctly assigned to a single leaf. The right hand side shows a tree in which this component is separated on the first split, which creates pairs (AB) and (CD) of clusters (12) and (23), respectively. In order to fully adapt, landmark (12) and

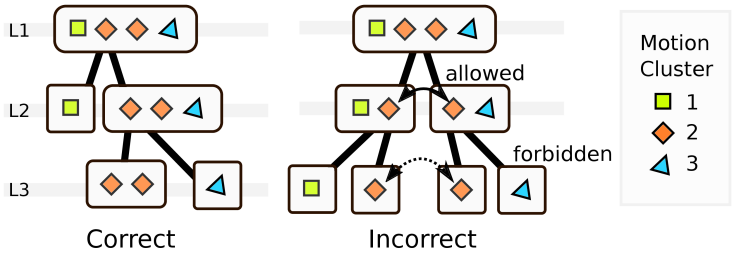


Figure 4.33.: Sample hierarchies where four landmarks (ABCD) describing three components (123) of motion are involved and clusters (23) share a common motion. Left: Correctly clustered tree, where a single landmark cluster is split off in the first stage and another on the second. Right: Incorrect (less stable) representation, where the motion cluster of two landmarks is split on the first level and propagated to both second level branches. In order to fix the tree, a rearrangement on the second level is required, since an exchange on the third is forbidden by the tree rules.

(23) are split on a second level, creating two individual clusters for motion cluster two. These two single clusters are now able to follow the motion of cluster two. However, as by rules of the hierarchy, since they are no direct siblings, they are not allowed to generate position proposals for each other.

They are not subjected to a common rearrangement process and cannot be joined on their level directly. Instead, they require the rearrangement to occur on the upper level, where the incorrect split appeared in the first place. This behavior is an intended consequence of the rules the association tree is built from. However, in this particular situation, there is a chance that the correction on the upper level does not occur: If on the first level the incorrect clusters of components (12) and (23) are dominated by the contributions of (1) and (3), they are not able to mutually produce valid proposals for component 2. Then, a rearrangement of component (2) is not possible and the correct cluster cannot be recovered.

'Appending' and 'Replacing' Cluster Creation The two ways of creating clusters are given in appending new nodes to the parent, or by replacing the parent by the child nodes. The appending split is useful, if the underlying level is a mere optimization to the parent level. In this case, the parent level can provide a rough estimate of the location. The child level can optimize individual portions thereof. In contrast, when the data shows a full split into truly independent motion, the remaining parent level cannot describe that motion. Often, the parent then follows the motion of one dominating child cluster, which can produce a stable outcome. If there is no single dominant cluster, the parent level may then introduce instability, alternating between transformations of the children. In this case, a replacing split would be favorable over an appending one.

For the EPI data algorithm, appending splits were used. There might, however, be ways to introduce both types. For instance two different cluster binding thresholds could

be used, inducing a replacing split for large cluster distances and an appending one for small.

Alternatives to Association Normalization As proposed and implemented, strong associations are only forwarded to one child among siblings. This is a process designed to keep the obtained position proposals of each level unique, keep the computational load limited and a direct consequence of the motivation by hierarchical clustering. However, different ways of normalizing the class forwarding might be thought of. It might be reasonable to forward an associations to multiple children, at least temporarily. This could be used to evaluate competing cluster hypotheses for instance. Competing clusters would both keep an association to the same landmark for a certain time of settlement. Similarly, some should not be forwarded at all. By normalization, a landmark is forced to be associated strongly to one cluster. If it does not support the cluster's proposals, then the landmark can be considered faulty and its strong association should not be forwarded.

These considerations affect the hard association state, which was introduced to convey the idea of hierarchical clustering. The continuous soft states on the other hand were normalized similar to Oja's rule [Oja82]. Since this could be extended to include orthonormalization [Gor06], the introduction of an orthonormalization process into this works update might be worth considering.

Soft Association Weighted Estimate In this work, a position estimate for a landmark is obtained from the cluster it is strongly associated to — the hard association is used. An alternative, form of output computation is found in weighting the position estimates of all clusters by the soft associations to the current landmark. This could also be source to a higher level variance measure, which can incorporate the level of contradiction between clusters. A major benefit of such an approach, however, would be the prevention of abrupt jumps, when the hard association is switched.

Landmark Subdivision In this work, the appearance model of a landmark holds a spatial layout. The proposed algorithm treats landmarks as the smallest units which are associable among nodes of the hierarchy. However, landmark descriptions themselves are generated by joining smaller units; e.g. gradients voxel pairss. This means, that generally the hierarchy could be continued to a finer scale, if desired. Also one might also correlate the response of these individual units to the motion the landmark is associated to. This way the layout could be adapted to better represent the actual motion. This is particularly interesting for landmarks that cover two contradicting clusters. A landmark optimized to fit the cluster which it is currently associated to would reduce contradictions.

4.4.6. Common Findings

Locality in Medical Image Data Even though no direct information about local relations of landmarks is provided to the algorithm, a locality in the clustering or rearrangement result may often be observed. Primarily, this relates to the core assumption, that spatially close landmarks undergo correlated motion. An additional factor may, however, be identified in a correlation of neighboring appearance models. As their spatial layouts exhibit a common overlap, their response functions are partially correlated. Such correlation expresses itself as similarity in the clustering process, leading to the assignment of a joint cluster to close landmarks.

Transformation Chain As discussed in the implementation, the transformation offered for the single nodes are restricted to translations, in order to prevent additional degrees of freedom of undefined transformation centers. It is generally possible to introduce more degrees of freedom to the root level, since on this stage the transformation center needs to be stated explicitly anyway. Here, the lack of direct additional degrees of freedom in the motion of individual nodes is compensated by the hierarchical tree of nodes.

As presented here, state and transformation space were separated. States needed to be converted to transformations first, before being joined on multiple levels. This is a result of the additional degree of freedom introduced by the undefined transformation center — it leaves states of multiple levels incompatible to each other. However, if the transformation centers are identical, as it was argued for in this chapter, then joining states of different levels would be possible. This would lead to a wider applicability of the Particle Filter functionality. Since the hierarchical prior would be available as an actual sample set, it could be sampled from and joined with the temporal prior directly, whereas in the current implementation it is given by a transformation estimate.

Stability of the Evolving Hierarchy Association rearrangement is implemented as a permanent optimization and repair process. If not explicitly stopped, it will continue to move associations along the tree. The more motion components differ, the more stable the resulting hierarchy is. There are however cases, which are particularly unstable, in the sense that a landmark repeatedly switches its associated cluster. Commonly this is found in sibling clusters that describe similar motion. For instance, if landmarks of the same global cluster — e.g. liver — are split into two spatially separated clusters, those on the centerline between the clusters are likely to repeatedly change their association. They may receive a better estimate from cluster one in the inhale and better estimate for from cluster two in the exhale phase. Thus, the effect is often related to a landmark being well described by multiple competing clusters. The effect depends not only on the data itself, but also on the prior hierarchy. With each switch of a landmark association, the new clusters provide different kinds of proposals and compete for their landmarks differently. In order to stop the adaptation, one would have to stop rearrangement or gradually decrease it in strength. Otherwise there is no final result hierarchy.

Association Initialization and Adaptation Rate In the presented algorithm, the initial soft state of a newly created strong and weak association is 1 and 0 respectively. The choice is motivated by the idea, that the process which created them is highly certain about the new separation. In particular, it implies that the result of the clustering process is to be trusted. This trust leads to relatively long rearrangement times, in cases the choice was incorrect. A possibly better choice which might be considered could be found in using details of the cluster result to initialize the associations' soft states. For instance, independent of whether a strong or weak association is created, it's soft state might be set to the similarity between the related landmark and the cluster centroid that causes the creation of its new cluster. This would reflect the uncertainty in the clustering result and lead to potentially quicker convergence of the rearrangement process.

Conclusion An approach for the on-line generation and maintenance of a hierarchical tracking scheme has been proposed and analyzed in select configurations and environments. Aim of the algorithm is to iteratively determine a common motion among sets of landmarks and then derive a hierarchical tree motion model. This chapter demonstrated the general feasibility of such an approach.

- A generic framework for adapting a hierarchical model which encodes transformations and appearance of the target model has been proposed.
  - It introduces the idea of managing *associations* between the tree and the set of known landmarks in the data.
  - Motivated by hierarchical clustering, a set of rules for association imposes intuitive limitations in the construction and dynamic adaption of a tree.
- An *association clustering* process is used to detect new sub components of motion and integrate them into the hierarchical description. Instead of clustering motion paths it directly acts on responses of landmarks to a common set of proposed transformations.
  - A k-means like implementation with a novel similarity criterion was shown to yield the desired components of motion from this type of input.
  - Unlike k-means it can produces a lower (better fitting) number of result clusters than suggested by the parametrization.
- With *association rearrangement* a new technique of maintaining the hierarchical tree's structure was proposed. Its long term maintenance effect was found to be a valuable addition to the short term node creation process of clustering.
  - Based on gathered evidence from the image stream, association to landmarks can be exchanged between sibling nodes. This leads to a restructuring of the hierarchy.
  - An attractive term favoring larger clusters of landmarks can be applied to speed up the rearrangement process.

The model was able to identify motion components on line. The maintenance process of the hierarchy is computationally inexpensive compared to the tracking process. This allows maintaining real-time applicability. Uncorrelated spatial random motion of landmarks was found to be a cause of failure for both processes. It contradicts the

assumption of a joint motion. The introduction of an attractive term for determined motion clusters was found to ameliorate the situation. The chapter presented one particular implementation of a hierarchy building algorithm. Ways for simplification of the setup, not implemented in this work, have been proposed. In general, there are extensive options for similar, alternative approaches. The algorithm has relationships to several fields of research. These will be laid out in the proceeding, final discussion of this thesis.

4.5. List of variables

Group	Var.	Brief	Detail
Association Model (page 137)	i, j, k	Indices	Node $i \in 1..N_N$ Landmark $j \in 1..N_F$ Sample $k \in 1..N_S$
	$S^A_{i,j,t}$	State	Label state of the association between node i and landmark j at time t.
	$s^A_{i,j,t}$	Soft State	Normalized scalar, representing how well landmark j fits to motion cluster i of a set of sibling nodes $i \in \mathcal{S}$.
	$s_{i,k}$	Sample State	Sample state k of node i's sample set.
	$\pi_{i,k}$	Sample Weight	Sample weight k of node i's sample set. Weight $\pi_{i,k}$ corresponds to state $s_{i,k}$.
	$\pi^A_{i,j,k}$	Associated Sample Weight	Associated sample weight k of node i's sample set, given an observation by landmark j. Weight $\pi^A_{i,j,k}$ corresponds to state $s_{i,k}$.
	M^{pc}_i	Parent $\rightarrow$ Child	Transformation from node i's parent center to node i's center.
	$M^A_{i,j}$	Node $\rightarrow$ Landmark	(Association) Transformation from node i's center to landmark j's center.
Clustering (page 139)	x, y	Indices	Input vector $x \in 1..N_{input}$ Cluster $y \in 1..K$
	K	Seeds Count	Maximum number of clusters to be obtained.
	$\vec{\pi^A_x}$	Input vector	Input vector entity.
	$\vec{c_y}$	Cluster centers	Obtained cluster centers
	CS	Cosine Similarity	Measure for similarity between input vectors, cluster centers and both types.
	N^{cl}_I	Iterations	Number of iterative steps of the clustering algorithm.
Cluster Result	K^*	Cluster Count	Number of clusters actually obtained
	l_x	Label	Label assigned to input vector
	N^{label}_y	Label Count	Labels / Cluster
	$M^{sim,cl}_{y_1,y_2}$	Cluster Similarity Matrix	Similarity between obtained clusters y_1 and y_2.
Interpretation	$M^{sim,f}_{j_1,j_2}$	Landmark Similarity Matrix	Reference similarity between features j_1 and j_2.
	θ_{bind}	Relative Binding Level	Relative similarity value, which when undercut induces a split of the related cluster.
Association Rearr. (page 144)	f_{AR}	Association Reinforcement	Rate at which functioning associations are strengthened.
	L		Number of nodes competing for an association to the same landmark.
	$r^{Att}_{i,j}$	Attractive reinforcement	Factor of attractive reinforcement. If no attraction: $r^{Att}_{i,j} \forall i, j$. Otherwise, the term grows with the number of existing strong associations.
	q	Quality	Additional quality factor, dominated by the nature of the input / observation data.

Group	Var.	Brief	Detail
Evaluation (page 152)	$\mathcal{C}^{GT}, \mathcal{C}^{R}$	Set of clusters	Ground truth and result sets of clusters, where each cluster is a set of labels.
	$\mathcal{J}$	'Jaccard Similarity'	Jaccard similarity between $\mathcal{C}^{GT}, \mathcal{C}^{R}$ as defined in (Eq. 4.19)
	ϵ	Error	Error with respect to either a known ground truth or an annotation. The error level always refers to a particular hierarchical level l of detail.
Artificial Data (page 152)	N_C,N_L	*Star* Settings	Number of clusters and landmarks in artificial motion set *star*.
	σ_{pos}	Position Noise	Standard deviation of Gaussian random noise, individual for each landmark and step
	f_r	Observation Noise	Relative radial deviation of the width of the measurement peak.
Timing	t	Time	Index of a discrete temporal step.
	t_{onset}	Onset Time	First step in which the hierarchy differs from its initial state.
	t_{finish}	Finish Time	First step in which a target hierarchy state is reached (for instance $\mathcal{J} = 1$).

5. Discussion

Detection and tracking of motion in medical image data streams are a key ingredient for automating image guidance in therapeutic applications. Task of an algorithm in this field is to determine a mapping between one reference image and another of the current situation in real-time. This work presents an approach to a mostly automated solution for articulate motion tracking based on medical image data. A focus is laid upon the identification of general motion patterns, which are considered mostly unknown beforehand. Particularly, no *a priori* statistical models are used. The idea of this work is to capture the motion of a sparse set of features in a hierarchical tree. In this tree, the root node describes global motion. Clusters of common motion are differentiated on the further levels.

Three sections are presented. The first lays modeling grounds, describing a single node in the hierarchical tree to be used for tracking. The second part uses such a node to construct a two level tree of a global and a local stage. Compared to the global, the dependent local stage is found to improve tracking results in artificial and medical image data. The global stage itself uses more spatial information, resulting in a more robust prior to the local stage. However, it assumes a globally linear movement. This is only given if the target structure is segmented properly from others and moves linearly itself. In order to design a more generic algorithm, the third part presents a multilevel model, which adaptively identifies clusters of motion. The adaptation integrates both the separation of independent motion and a successive further refinement in the optimization.

5.1. Contents and Relation to Literature

This section will summarize the content of this work and its relation to literature approaches.

Gaussian Transition Filter Model Chapter *Gaussian Transition Filter Model* (2) presents a simple Particle Filter model which later poses the basis for computations of each node of the hierarchy. Notable characteristics of this model are the assumption of a solely diffusive estimation process and a likelihood of limited non-zero range. For fundamental properties of this simplified tracker model, theoretical explanations and models are derived and validated by computer experiments.

In this theoretical section, a target moving with fixed linear velocity is to be followed steadily. While the model could fully be described analytically, when the likelihood

function was chosen as a Gaussian distribution, the chapter primarily discusses measurement of limited non-zero range: A Gaussian distribution, truncated to its non-zero, second order Taylor expansion is assumed. It is non-zero close to the true target position and zero further outside. This is to mimic two effects of sampled measurement that can be obtained from image data. Primarily, it models an observation function which can distinctively reject samples that are too different from the target; a discriminative decision. Secondly, it denies an infinite attractive range of a full Gaussian distribution; something which by sampled detection can not be achieved. This anticipates properties of a measurement process observed in later chapters.

For this model of limited range detectability, the interplay of central parameters is described. These are sample count, diffusive search range, maximum velocity, dimensionality, number of steps and desired success probability. For particular cases the dependency can be approximated into analytical expression, which leads to simple parametrization schemes.

Next, predictions about the expectable level of residual errors in the track estimate, as well as the scaling behavior thereof can be motivated. Over wide ranges of parameters the Gaussian and truncated Gaussian model show very related behavior.

Finally, noise is added to the measurement process. This leads to the zero-value of the measurement becoming non-zero occasionally. However, since noise is spatially independent it does not introduce an attraction towards the true target position. Such noise can be interpreted as uncertain or incorrect false positive detection or just a plain image noise which could not be suppressed by the evaluating image filter. As samples may become lost in noise, the track can fail more easily and the position estimate deteriorates. Models for the explanation of track loss under these circumstances are motivated and found to explain observed parameter thresholds. To the author's knowledge, this is the first analysis in this particular direction.

This chapter does not explain all properties of tracking nodes as they are made use of in further chapters. For instance, as presented, the effect of multiple iterations in a very noisy likelihood environment cannot be explained by the presented modelings. The same applies to the prediction of the expectable result of an annealing schedule. However, both of these effects improve performance, and the chapter models a lower boundary.

Adding a Hierarchical Filter Scheme and Medical Image Data A step into the direction of applications to medical data is undertaken in chapter *Fixed Hierarchy Model* (3). Moving towards an application to medical images, a sparse feature representation is promoted. A process for reducing an initial set of descriptive entities in a local image patch to a smaller subset of the most distinct ones is presented. This allows deriving the filter function for individual image features from the image itself. A tuning parameter may be used to adapt the selection process to different image appearances. The chapter leads to a fast, greedy reduction process, which enables the selection of larger scale landmarks while remaining computationally efficient.

This way, each landmark obtains an individual appearance model, which is selected as a sparse gradient based representation. As in the field of *iconic image* registration, localized properties are derived from initial regions. These are then used for matching. No statistical modeling of motion or appearance is applied. In particular, models are explicitly not updated by generating new representation from later frames. This is done in order to prevent drift. Generally, the algorithm would support the integration of multiple differing descriptions derived from the same frame of reference. The sequence of frames might then be used to learn the best fitting description or combinations thereof. This however has not been explored. Rather than updating the appearances of individual landmarks, their clusters are updated.

Furthermore, a hierarchical model is introduced. An iterative, regular increase in the level of detail is referred to as multilevel approach [HM04] in medical image registration. The two-level approach in chapter *Fixed Hierarchy Model* (3) represents a particular case. It uses the global level for stabilization, whereas the local stage tweaks the local position estimate. The properties of nodes on the two levels can be set individually. This allows optimizing the performance of the algorithm to the target scenario. For instance, the deformation model of the global stage shows more degrees of freedom than the local, whereas the global landmark model is decreased in detail. The latter is particularly beneficial in three dimensional data. Furthermore, appearance models on both stages can integrate different aspects of the data. An example is found in ultrasound tracking, where the global model incoporates the diaphragm region, but local models solely track liver vessels.

The model assumes that the global deformation can be described by a large, global layout and transformation, which is still low in degrees of freedom. This poses requirements to the data. Only if it obeys this assumption, the algorithm can succeed. Otherwise, while the global stage may find an optimum, it may misrepresent some local regions. In the two-level model this means that the transformation priors forwarded to local stages would be faulty. Thus, the restriction to the correct data remains and only is loosened by the extension in chapter *Dynamic Hierarchy Model* (4).

The local level nodes are not regularized among each other. Downside of this approach is that uncertain landmarks are not subjected to correction from their neighbors. In case of increased local uncertainty, single landmarks introduce more error. On the other hand, a regularization is only justified, when the constraints between landmarks are known. For instance, imposing an elastic constraint between sliding boundaries, such as ribcage and liver, may lead to erroneous results, likewise.

Compared to registration approaches, the presented algorithm can naturally integrate temporal prior distributions into state estimation. A major benefit of this is observed for the global stage. Due to the lack of an additional hierarchical prior, it acts similar to a conventional Particle Filter. The local level was found to be more stable when solely using the hierarchical prior, which prevents the accumulation of drift after frames of uncertain detections. With further reasoning introduced, the local level might also benefit from temporal priors.

Dynamically Adapted Hierarchy In chapter *Dynamic Hierarchy Model* (4) the algorithm is generalized with respect to the underlying hierarchical model. The proposed multilevel model updates the hierarchical tree during the execution. The clusters formed therein may be regarded as a dynamically adapted, implicit interaction between the existing landmarks. Therein, the formation of multiple levels of motion cluster is an approximation to the actual interaction. Landmarks of the same cluster move identically. Those of different clusters are independent, except for a common parent proposal transformation. As clusters and their boundaries are unevenly distributed, the interaction between landmarks is as well. The cluster detection process in itself is not localized. Therefore the range of the obtained interaction is unlimited. The detection process of common clusters follows a path of associative learning. Freely reinterpreting Hebb's rule [Heb50; Sha92] this would be: What can be explained together, joins a common cluster together. In the presented data the non local approach seems justified: For instance, when observing breathing motion, the anterior and posterior chest wall are not visibly connected in sagittal 2D images. However, they are correctly joined into a common cluster of motion[1]. Thus, even though no knowledge about a rigid connection of the two is available to the algorithm, it can be identified by observing the evidence in the image stream.

Relation to Video Tracking and Image Registration In medical image registration, a majority of approaches relies on the deterministic optimization of a global functional. The application of a Particle Filter is uncommon. An advantage of the Particle Filter is the availability of measures of uncertainty. It implements a weighted mean estimate of the distribution and a variance can be estimated for each node. The approaches presented in the literature mostly focus on using a Particle Filter for a global or preprocessing stage. An implementation for breathing motion tracking is found in [Smi+12]. In this, the authors use a second order autoregressive transition to model the state transition. Transformations are given by a low dimensional PCA subspace model, which describes the motion of markers attached to the patient's chest. In [ACA12] a Particle Filter based algorithm is proposed for global brain image registration. Similar to this work, the algorithm iteratively approaches the MAP estimate by reducing the diffusion term of a Gaussian random walk transition in each step. In order to approach elastic registrations, the authors in follow-up work however suggest a secondary stage based on Optical Flow [Mej+11] or expectation-maximization [Arc+13]. A precursor to this work [Rot+14] uses a simple model to track individual landmarks in ultrasound liver data. While the algorithm in the related challenge was found to be very precise, it required a well-designed landmark description for each class of data provided. In this work the algorithm was extended and generalized in many regards.

Two stage approaches are found in the literature of both video tracking and medical image registration. Approaches [Mej+11; Arc+13] join a global Particle Filter stage by a different, more detailed post processing algorithm. A two-stage approach is also

[1]Or rather 'no' motion; as the patient lies still within the imaging system.

promoted in [Lot+16] for the registration of large histological slices. The approaches commonly promote the separation of a coarse global from a local high-frequency stage, since the optimization of a complete global functional is computationally very expensive. In tracking applications more abstract stages can be found, such as the *adaptive coupled layer* approach in [CKL13; XSL13]. It uses a global/local feedback loop for globally managing the set of patches which lead to local identification of the target region: The global stage acts as a feedback to update the patches in the representation.

The final algorithm of *Dynamic Hierarchy Model* (4) represents a generalized variant of pose estimation, in the sense that the motion model can be expressed as hierarchical graph. However, unlike common skeletal models, the parent-child relationship does not provide strict conditions for the child level. It rather lays out the prior for the proceeding search of the child state. In this work, the objective is not formulated in a joint state space. As in *partitioned sampling* [MI00], parent and child state are separated. They are each assigned an own objective function. Furthermore, an iterative annealing scheme similar to [DR05] is applied for optimization. The two level model described in chapter *Fixed Hierarchy Model* (3) defines a pose model beforehand as common in pose estimation. However, the dynamic model in *Dynamic Hierarchy Model* (4) derives the required joint splits from the data and constantly updates them. The individual joints of motion are not known beforehand, derived from the data during tracking and are updated with new incoming evidence.

In the light of image registration, the developed algorithm for medical data could be classified in the following way: It is a mono modal, stochastic matching algorithm which uses linear transformations of non-local clusters in a hierarchical evaluation scheme to obtain a sparse deformation field.

In the literature on medical image registration, approaches of local interaction with a globally set strength are often found. Deformable image registration builds upon a regularizing term which for instance encourages smooth deformation fields. Spline deformation may impose side constraints [Sch+01; RAD03]. These may also be motivated by biomechanical models [Fer+01; Glo+10] of Finite Element Model (FEM) simulation. While these can be equipped with spatial distribution of differing local interactions, such requires further a-priori knowledge of the underlying structure.

Particular challenges are found in modeling the discontinuities of deformation between individual organs These may for instance stem from segmentation [Sch+12; Der+15], which requires a preprocessing algorithm. Approaches modeling discontinuous deformation fields also have been proposed [KFC11; Kir+16; Pap+18]. Another approach for estimating the directivity of local interactions is found in a *Markov Random Fields* (MRF) based algorithm in [Hei+13]. It optimizes the interaction of MRF nodes on a minimum spanning tree, computed by incorporating neighbor patch similarities as distance measures into Prim's algorithm [Pri57]. This leads to non-isotropic interaction, which in MRF acts locally. These are able to identify discontinuities, but as many dense registration algorithms, require more computational time than acceptable for real-time applications. A reason for this can be seen in the differing approach to the matching process. In image registration often the entire dense field is to be estimated up to the

highest level of detail. The method proposed in this work not only produces a sparse field, but it also only integrates higher levels, if appearing residual motion demands it. Finally, registration based approaches use local regularization, whereas in this work landmarks of a joint motion cluster are locally unrestricted in their choice of elements.

The similarity to population based motion models [McC+13; Pre+14] exists in the property of a strongly reduced number of dimensions in the transformation model. While generative statistical models provide constraints implicitly, the presented algorithm is required to learn them. The advantage compared to statistical models is its genericity, or lack of initial specification. The algorithm is able to track different organs at once and therefore does not depend on a detailed initialization phase that defines correspondences between model and image.

Association The work shares similarities with the problem of Multiple Target Tracking (MTT) algorithms, in such way that multiple landmarks are to be followed. Similar to MTT, landmarks may be indistinguishable. Particularly small liver vessels in EPI images show very similar appearance. However, other than in MTT, these landmarks are widely mutually dependent in their motion. Therefore, the primary approach, particularly in *Dynamic Hierarchy Model* (4), is to treat landmarks as part of a common motion as long as justifiable. Once the algorithm detects that this assumption no longer applies, new child clusters are introduced. The boundaries between individual sibling clusters are then constantly renegotiated. This circumvents the most *'thorny'* [RWS95] problem of direct association matching: The nature of the data allows shifting an association problem between observations and tracked entities to an association problem of known landmarks to changing clusters of motion. The Particle Filter in this sense can be understood as nearest neighbor association, or gated evaluation. Each node relies on being provided a useful prior of a larger scale cluster. Since the larger cluster is expected to already have resolved larger scale motion, the search space of the child node can be reduced. This leads to an implicit gating. Since the width of the searching distribution is lessened, detections that lead to incorrect assignments are less probable. Thus, the incorrect assignments are further away from the proposed hierarchical prior. Nevertheless, the algorithm might benefit from assignment reasoning among sibling nodes. Currently, no mechanism is implemented to prevent assigning multiple landmark models to the same location. Methods such as *probabilistic exclusion* [MB99] could be installed to amend this. Furthermore, the reinforcement algorithm shares a resemblance to multiple hypotheses tracking [Rei79] such that over multiple steps, the association to variable candidate motion clusters is updated. However, the available hypotheses are only based on already existing sibling clusters and therefore limited. No exponential branching takes place. Otherwise the combinatorial options of cluster-reconfigurations in single steps alone would make a real-time application difficult.

Result Interpretation The result of each tracking step is rich in position estimates. In the two level model, each landmark is associated to an estimate of the global and the local stage. In the dynamically changing hierarchy model, each strong association

produces an estimate for a landmark position. The presented implementation uses the estimates of a desired target level as an output. If a landmark lacks an estimate on that level, the algorithm uses the highest available level of that landmark instead. This way sparse fields for any target level are available. Aside from the position estimate, each vector in the field holds a label of the cluster it originated from.

Depending on the application, these fields may be interpreted in different ways. An operator might manually choose the cluster, which is related to the desired motion. The target vector field extracted from it could reside in the same cluster or any other child level. Other than a spatial selection, a temporal selection could be applied. One might select a cluster which is both large and which shows a wide range motion. The selection of the target vector field could also be automatized by selecting only those vectors which find their origin in an annotated region of interest. They would not have to be part of a common parent cluster. As consequence of selecting the components of interest, the computational efforts for the others could be reduced.

Computation Time and Parallelizeability In this work, the algorithm could be brought to real-time performance. In order to achieve this, it was necessary to reduce the description of individual landmarks in *Fixed Hierarchy Model* (3), since their evaluation takes the majority of computation time. Limits to the number of landmarks and levels of the hierarchical tree have been imposed. The algorithm was implemented in a sequential fashion for CPU evaluation and would greatly benefit from future parallelization. On each occasion of independent computations, they can be parallelized. These include simultaneous evaluation of nodes on the same level (factor $\approx 1..20$ in this work). On each node, weighting of the sample sets can be parallelized (factor $N_S \approx 100..500$). Finally, dot product kernel in the landmark description may evaluate all kernel results simultaneously (factor $N_P \approx 20..50$). Several processes are not parallelizeable. Among these are the evaluation of nodes on different levels, the computation of variances and maintenance processes, such as clustering and reinforcement. In general, a high performance gain can be expected from parallelization. Porting the algorithm to a GPU implementation would also yield options to inexpensively integrate computations of image detail pyramids. These would well match the hierarchical scheme.

5.2. Application

User Interaction For a medical application, automation and support of user interaction are of interest. By tracking and segmenting the scene into groups of common motion, the algorithm produces more information than is required for an actual motion compensation. An interpretation of the result is thus called for, before it is passed on. For such interpretation the user must specify the target region. This could prospectively be achieved by masking the area of interest, such as the liver. Then, landmarks which relate to that region would be prioritized or highlighted in the result. It could also be achieved on line by allowing the user to choose a motion component which during runtime appears most representative for the ROI. This could in particular be of interest,

if the target region itself is sparse of features. The selection of such a seed region would allow to specify the target motion component on the second hierarchical level. Since the algorithm joins landmarks which move in similar fashion such selection would include landmarks outside the ROI which are found to move alike and thus stabilize the result.

Training and Application Phase Initially, the algorithm does not have information about expectable motion other than its maximum amplitude. The detailed motion model has to be derived from the data stream. Since the algorithm's outcome depends on an individual series of input images, the resulting hierarchical model is not predetermined. Generation and application of the model should be separated and interrupted for a sanity check by the operator.

During the generation, or training phase, the algorithm is presented a common motion which is expected to occur during the procedure. Motion may be exaggerated, the patient may take a deeper breath than in later application. The model is then built from the data of this phase. This may already be done on line, but also variants requiring multiple passes or interaction may be thought of. On line, the training is executed multiple times on the same dataset, leading to further stabilization of the associations. An interactive component might be introduced by allowing the operator to manually choose a splitting point for the clustering algorithm, by defining which frame to use for the first creation of clusters. The training phase could be used to check for unstable landmarks i.e. those which produce outliers of the variance of their position estimate.

After the training and inspection phase is finished, the hierarchy is fixed. The set of representative landmarks is set to those within the target region and associated to the target motion component. During the proceeding application, the algorithm continues tracking without altering the hierarchy and provides the results for the target landmarks to further units. These could include a higher level motion prediction and a steering component of a treatment device.

General Parametrization The general parametrization of the full algorithm should be separated into two distinct stages — for developer and user. Both of them require sample data.

In developer stage, the general properties of the algorithm need to be set (Alg. 5.1). This stage separates two phases — setting the parameters for the tracking components (nodes) and the creation of hierarchy (tree). For nodes, general choices are made: What appearance model suits the landmarks in the data? The response to the model determines the basic parameters that are chosen for a node. Depending on the width of the response peak in an image, and a potential residual noise level outside, search range and the number of samples are chosen. If the require velocity can not be reached with a reasonable amount of samples, filters which yield a broader response need to be selected. This is described in detail in chapter *Gaussian Transition Filter Model* (2). The further configuration of nodes involves the choice of an iterated tracking scheme. Experience in this work suggests to favor multiple repetitions ($N_I = 3..5$) with low sample counts ($N_S = 50..100$) and medium annealing factors ($\lambda = 2..4$) to yield reasonable results. In

fact, in EPI data, a certain variability of these is allowed without producing qualitatively different results. A higher number of iterations also allows for more conservative choices of spatial uncertainty per step. The propagation of estimating variance should be chosen *adaptive*, such that it decreases, once the target has been detected. The determined parametrization must then be applicable to the individual motion components in the sample data.

Individual Node Configuration
- Choose appearance model, degrees of freedom
- Measure response to sample landmarks
  - Determine require number of samples N_S and search range σ_E.
- Define a number of iterations and an annealing schedule

Hierarchy Buildup
- Set binding threshold
- Measure relation between reinforcement and adaptation rate
  - Set association reinforcement level

Algorithm 5.1: Developer Stage Parametrization Scheme

Once this has been accomplished, hierarchy buildup may be approached. In this, settings for the clustering algorithm are chosen first. The binding threshold should be selected in a way that clusters are produced once the algorithm is presented with a clear distinction of what clusters to expect. If no clusters are produced, the (sensitivity of) the binding threshold must be increased. It is acceptable if the clustering result is overly sensitive and produces more clusters than the operator would initially expect.

The rearrangement process then is parametrized to clean up incorrect or excess clusters with the progression of the sample data. If a clear distinction of major components is desired, an attractive term should be used. For configuration it is necessary to measure the approximately linear dependency of the adaptation rate to the reinforcement strength. If latter is too low, adaptation will be slow. If it is too high, clusters will be lost in times when they are indistinguishable; i.e. when they can explain observations of each other. The rate should therefore be oriented on the typical durations in which clusters are indistinguishable. This particularly affects the reference state. For cyclic data, the reinforcement should be chosen, such that the resulting adaptation rate is not higher than the cycle frequency. Generally, it should never be higher than the inverse of a possible resting phase in the reference state.

The developer stage sets the core algorithm parameters, whereas the proceeding user stage reduces them to a subset that may be required for adaptation to particular scenarios. In individual cases, values which change the most with patient or device are given in motion magnitude, cycle duration and resolution. Hence, for specific adaptation, a scaling value for motion parameters and appearance model should be provided to the user. If algorithm lags and loses structure, expected motion magnitude must be increased. If a customizable appearance model is used, as in this work, the user may be given the option to change the magnitude of gradients included into the model. Choo-

sing wider gradients leads to a more robust large scale matching, but can decrease the quality of individual position estimates. For hierarchy generation, a relative value for clustering binding threshold and reinforcement strength should be provided. If clusters are expected, but not formed, the sensitivity can be increased. If they vanish too early, the reinforcement rate needs to be decreased. This leaves the user with four values to set, which in ideal cases do not need to be adapted.

5.3. Outlook

Fields of Application Applications of the presented work are found in generic real-time tracking scenarios; particularly in the identification of breathing motion. As initially presented, necessities for generic tracking solutions are found in automated, non-invasive treatments. Currently the most prominent are radiation therapy and focused ultrasound guidance. A fast adaptation to the current situation is a major requirement for these applications. Not only does the presented algorithm work in real-time, it may also help further automate the work flow by identifying relevant modes of motion automatically.

Aside from therapy, a use in diagnostic applications is possible. Generally, the stabilization of a particular component of motion could be regarded as the desired goal itself. A plain 'freezing' function for particular components of motion might be desired in applications, where one is interested in a particular residual motion. For instance, by freezing motion of breathing, one may observe residual components related to cardiac pulsation more clearly.

Similar to therapy, a secondary imaging system (e.g. ultrasound) could assist in motion correction of a primary imaging system. For instance in *prospective motion correction* [Mac+13] the real-time adaptation of the MRI imaging parameters prevents motion artifacts. Exporting the detection of motion to a system other than MRI, valuable time can be saved for the actual MRI acquisition. In *retrospective motion correction* [God+16], the advantage of real-time processing falls short of the advantages of a detailed image registration process. Nevertheless, the additional step of automatically dissecting motion vector fields into clusters might be an interesting addition; simplifying the result before the actual correction.

Possible Tree Specialization The two-level model of chapter 3 uses specific configurations for global and local levels. This was shown beneficial for performance. Such approach of a regular, predefined hierarchy could be extended to further levels. Dedicated appearance models for different levels of the hierarchy could be applied together with a multilevel image representation. Each level of tracking would receive a different level of detail input.

For reasons of simplicity and compatibility, in the generic model of chapter *Dynamic Hierarchy Model* (4), different levels of the hierarchy were treated equally. They were assigned identical degrees of freedom, motion parameters, and — by association — landmarks. There are multiple options to specialize the hierarchy for different applications. One observes that there are different types of hierarchical splits: On the initial levels,

each new level indicates, that there actually exist independent components of motion. The motion components identified in such *hard* splits may undergo entirely different patterns. Therefore, in these levels of the hierarchy it would be reasonable to offer more degrees of freedom, which better suit the expected components. For instance, scaling and rotation degrees of freedom aid the adaptation on the first level. In further proceeding levels, the differences between levels are mostly composed of optimizing *soft* splits. Due to their reduced numbers of landmarks, these are sensitive to excess degrees of freedom. Thus, for these levels, both the degrees of freedom and the extent of expected motion should be reduced.

The first splitting levels could use a temporal prior for the estimation, whereas the optimizing stages only rely on the hierarchical priors. Finally — similar to the two-level-model — an additional dedicated single-landmark leaf level could be used to explicitly optimize every landmark individually.

Other Areas of Application Generally, the algorithm as such may be applied to other areas of articulated physiological motion. A study of ultrasound data used for the hierarchy adapting approach is in order. Also, other sources of motion — particularly cardiac — are yet to be explored.

Further applications might be based on the detection of multi modality. The presented clustering algorithm is designed to detect the change from a mono modal to a multi modal distribution. In this work such an event indicates the birth of new modes of motion. It could be applied to other processes which spawn new entities in the same place, e.g. cell division in in-vitro cell tracking applications. Furthermore, it could be used to detect the split of one transformation into multiple. Such appear for instance if two histological slices are to be registered, but one of them is ruptured into multiple parts.

Generally, the clustering algorithm can be used to detect differences in two weighted particle distributions. A further application along the lines of this work is the identification of an on line detection of a suitable reduction level for landmarks: In this work the set of gradient-like entities is reduced to their N_P top ranked ones. Thus, models of differing N_P share a common description, which responses can be computed for. Then, N_P could be chosen such that the difference in the response does not indicate a split.

The concept of adaptation could be altered for image registration, where only two images are available. In such case, associations cannot be rearranged and only the clustering process is available for iterated adaptation. A single clustering result is likely to be unreliable. Therefore, one might generate different clustering results from the same sample set. Approaches might be using multiple random subsets thereof (bootstrap aggregation) or varying binding thresholds. Multiple results obtained by such faulty approach could be joined, for instance by a majority vote, as seen in Random Forest classification.

A further advantage of the hierarchical approach is the presence of cluster labels, and the relation to pose estimation. Such agree well with skeletal bone segmentation tasks. Hierarchical approaches for whole body skeletal registration [YPJ14; YJ14; Fu+17] have been applied, but report that computation time for registration is 'not enough to support

on line applications' [Fu+17]. These approaches use a predefined hierarchy and common deterministic optimization methods to find a solution. The process as promoted here could accelerate such matching. It could even detect unknown constraints between connected bones. For instance, in cases such as foot bone registration [Hu+11], many bones reside in a close neighborhood and are loosely connected. Interpreting each bone as a landmark, the algorithm presented in this work may be used to detect structural clusters by itself and derived the required hierarchical scheme for each case individually.

The elements of unsupervised learning, namely clustering and association reinforcement, could be applied to other situations. It is useful for data in which different amounts of correlation among the observed entities are suspected and these are to be dissected into a hierarchy. The algorithm presented is a method to reduce the dimensions of the data and adaptively increase them only if the data indicates a necessity.

5.4. Conclusion

This work formulates a novel approach to a generic framework for hierarchical matching. It is presented with a special application to medical image data. Designed to automatically capture articulate motion of respiration, different aspects are emphasized. These are a real-time performance and an uncomplicated setup that requires a minimum of prior information or user interaction.

Initially, description and simulation of a theoretical model are used to build an understanding of the central Particle Filter based units of optimization. A general connection between parametrization and resulting success probability is established. Error levels of position estimates and the point of track loss in the presence of noise are modeled. The obtained relations can be used to estimate a required parametrization for particular target situation. They help understand the trade off between the invested computation and the capabilities of the designed algorithm. Approaching actual medical data, a sparse landmark description is suggested and a fixed two level hierarchy is presented for tracking. In three applications to medical image data the real-time applicability is demonstrated. In the case of MR liver image data, an automated feature detector for liver vessels is presented, which allows for the design of a fully autonomous algorithm. However, the fixed two-level model is only suitable for tracking the joint and detailed motion of a single organ. A further chapter therefore generalizes it to an arbitrary hierarchy. Aside from allowing for a generic structure, means of unsupervised learning are introduced to derive said structure from the incoming data. The presented final algorithm is designed to fulfill three goals towards an automatic application. It admits real-time applicability, does not require detailed user interaction in the setup phase and can learn the underlying patterns in an image series on line. For clinical applications this implies reduced efforts in the setup of an image based real-time tracking system.

In summary, a novel generic real-time landmark matching algorithm has been developed for use in medical intervention. It derives the patient specific motion pattern on line. This way the algorithm avoids the need for detailed a-priori knowledge of a motion. Furthermore, the correspondences between such a model and the data need not

be known on setup. Therefore the requirement for annotations of the reference data is strongly relaxed. These properties allow for further automation of clinical interventions that require real-time steering. The underlying mechanisms are however more general and allow for applications well beyond medical ones.

A. Further Material

In *Gaussian Transition Filter Model* (2) the successful sampled tracking is depicted in Figure 2.1. Compared to this, Figure A.1 shows a unsuccessful sampled tracking, where the prediction's with has been reduced from $\sigma_E = 4$ to $\sigma_E = 1$. On the other hand, Figure A.2 depicts the fully Gaussian reference model.

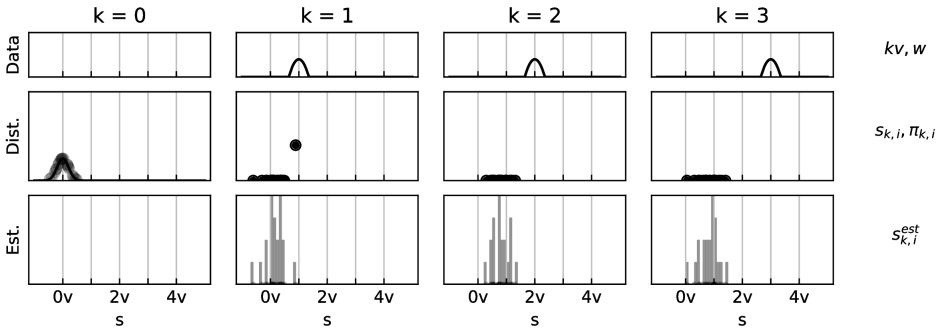


Figure A.1.: First steps of an unsuccessful experiment with $N_S = 20$ samples and target velocity $v = 4$. Prediction by Gaussian of $\sigma_E = 1$ transforms the weighted sample set (center row) into the unweighted set (bottom row). The update step joins predicted set and a truncated Gaussian Likelihood function of $\sigma_A = 1$ (top row) into the posterior (center row). The initial distribution is sampled from a Gaussian of $\sigma = 0.8$ (center, left). In the first step a proposal sample is places into the non-zero likelihood region. From $k = 2$ this is not the case. The target is lost.

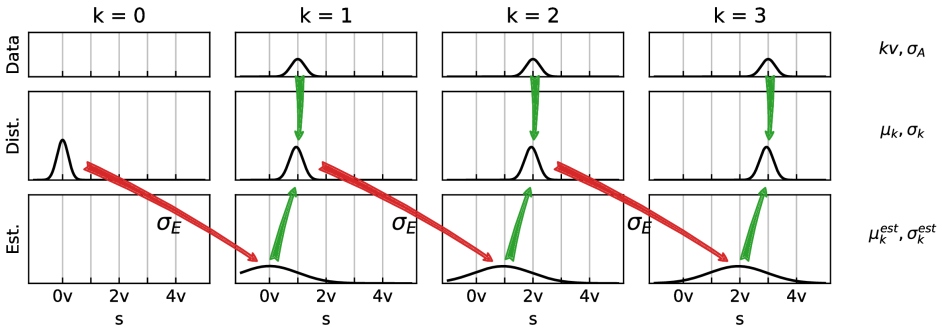


Figure A.2.: First steps of tracking in the Gaussian reference model. Parametrized equivalent to the successful sampled case in Figure 2.1.

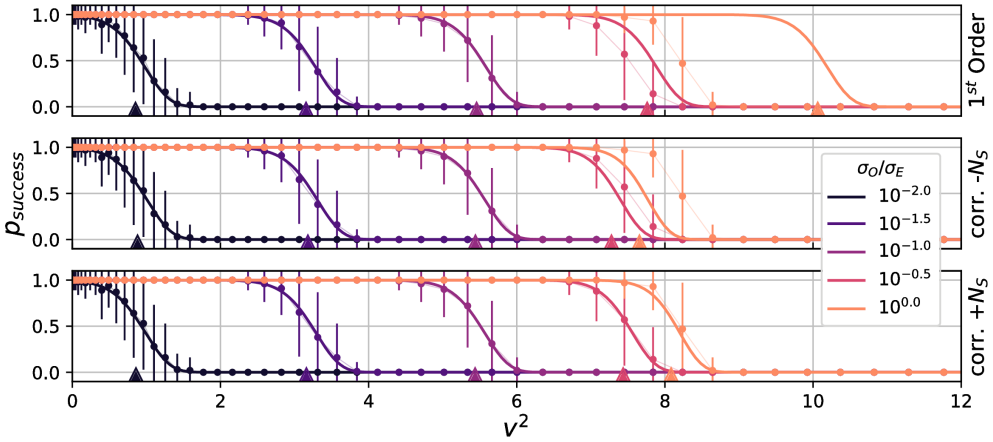


Figure A.3.: Predicted and empirical success probabilities for five distinct values σ_A/σ_E - the first three values correspond to the linearizable range, whereas the last two reside in the plateau area. The figure shows three plots: The 1st order model and the default model with velocity correction where N_S is either fit or not. For each value of σ_A/σ_E, statistics of the empirical $p_{success}(v)$ are shown in the two lower plots by mean (dots) and standard deviation (whiskers). This is overlaid with predictions for $p_{success}$ of the fitted models (thick curve) and the prediction of the expected maximum velocity (bottom triangles).

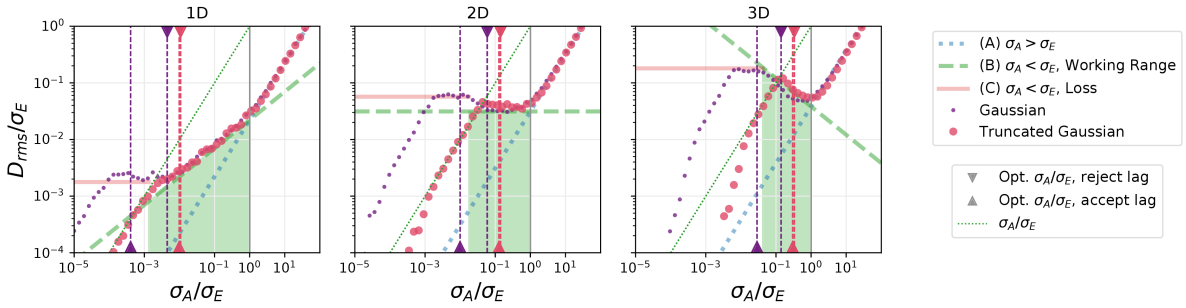


Figure A.4.: Residual error D_{rms}/σ_E for $v = 0$, augmented by estimates of optimal parametrization. (Settings: $k_{max} = 1000$, $\sigma_E = \{0.1, 1, 10\}$, $N_S = 1024$). Equivalent depiction as in Figure 2.13, but with additional hints where the optimal parameterizations are found.

List of Abbreviations

Medicine	RT	Radio Therapy
	EBRT	External Beam Radiation Therapy
	GTV	Gross Tumor Volume: Gross, palpable, visible tumor volume
	CTV	Clinical Target Volume: Clinical target volume, including the GTV and small malign cell clusters
	PTV	Planning Target Volume: Geometric volume including inter-fractional and intra-fractional changes
	MLC	Multileaf Collimator: Device for shaping a treatment beam in radiotherapy, dynamically obstructing the beam in areas where it is not desired
	LINAC	Linear Particle Accelerator
	FUS	Focused Ultrasound
	HIFU	High Intensity Focused Ultrasound
Imaging	MR(I)	Magnetic Resonance (Imaging)
	US	Ultrasound
	CT	Computed Tomography
	TMP	Through-Plane Motion
Distance	NCC	Normalized Cross Correlation
	SSD	Sum of Sqaured Distances
	MI	Mutual Information
Features	DOF	Degree(s) of freedom
	LoG	Laplacian of Gaussians (image filter)
	DoG	Difference of Gaussians (image filter)
	SIFT	Scale invariant feature transform
	SURF	Speeded up robust features
	GLOH	Gradient location-orientation histogram
	LBP	Local Binary Patterns

Statistics	*i.i.d*	'independent and identically distributed'
	MCMC	Markov Chain Monte Carlo
	SMC	Sequential Monte Carlo
	SIR	Sampling Importance Resampling
	SA	Simulated Ananealing
	SMC	Sequential Monte Carlo
	MR	Mixture Reduction
	MAP	Maximum a Posteriori
	MLE	Maximum Likelihood Estimate
	MMSE	Minimum Mean Squared Error
	MRF	Markov Random Fields
	PCA	Principal Component Analysis
	PF	Particle Filter
	KF	Kalman Filter

List of Figures

1.1. Localization of organs and glands within the (male) human body 7
1.2. Liver and its major vessels, vascular flow 8
1.3. Sample of 1D breathing motion and of the breathing model after [Luj+99] 11
1.4. Sample images of depth dose curve and radiotherapy device 14
1.5. Wave propagation for FUS application 15
1.6. Options for motion managment 16
1.7. Possible application for tracking landmarks in medical image data 17
1.8. Imaging planes oriented on the human body 20
1.9. Liver and prostate anatomy in Echo Planar Imaging 22
1.10. Motion of liver and kidney during the respiratory and cardiac cycle . . . 22
1.11. Wave propagation in diagnostic ultrasound 23
1.12. Sample images of abdominal ultrasound 24
1.13. Major elements and data flow of a tracking algorithm 25
1.14. Modeling components for the tracking algorithm 26
1.15. Image and feature based matching 28
1.16. Basic types of poses states and related transformations 30
1.17. Options for predicting the new object's location 34
1.18. A hierarchical tree with three levels 39
1.19. Sample of a two dimensional probability distribution 43
1.20. Sequential update of states 43
1.21. Importance sampling 47
1.22. Particle Filter step with ten particles 49
1.23. Annealed Particle Filtering 50

2.1. Example of the first steps of the simulation experiment 55
2.2. Sample of two-dimensional Gaussian likelihood function with noise 55
2.3. Model of the critical step in the linear tracking problem 58
2.4. Overview of extremal and optimal parametrization 62
2.5. Overview over residual errors in 1D 64
2.6. Samples of simulated distributions of the Multi State Noise Model 66
2.7. Error levels estimated by Multi State Noise Model and Two State Noise Model 67
2.8. Artificial sample depletion introduced by noise 68
2.9. Three steps of experimental model fitting 69
2.10. Sample for lag accepting and lag rejecting thresholds 71
2.11. Data thresholding and fitting results 73
2.12. Tracking experiment in a space of 1, 2, and 3 degrees of freedom 74

2.13. Residual error D_{rms}/σ_E for $v = 0$ 75
2.14. Residual standard errors at $v > 0$ 76
2.15. Residual errors obtained when noise is of different levels is introduced . . 77
2.16. 1D, 2D, 3D results for different levels of noise 78
2.17. Effect of spreading samples over N_I iterations 79
2.18. Proposed parametrization scheme for the noise free ($\alpha = 0$) case. 82

3.1. Introduction of transformation and appearance model 87
3.2. Hierarchy, states, transformations and observation for nodes along a branch 89
3.3. Transformation along a path 90
3.4. Node and leaf based observation 92
3.5. Global and local transformation 92
3.6. Transformation stages of the two level filter 94
3.7. Temporal and hierarchical priors in hierarchical tree 95
3.8. Sample appearance model 96
3.9. Generation process for appearance model 98
3.10. Example case effect of scaling parameter 100
3.11. Sample images for the LBP based feature detector 101
3.12. Samples of artificial breathing motion data 104
3.13. Samples of artificial appearance data 106
3.14. Initialization samples of the appearance model on EPI slices of liver data 107
3.15. Initialization samples of the appearance model on liver ultrasound data . 110
3.16. Alternative initialization of appearance model weights 112
3.17. Sample deformation result in prostate MRI data 113
3.18. Result error measures on artificial breathing motion data 115
3.19. Responses to artificial appearance data via Pearson's Correlation 116
3.20. Responses to artificial appearance dependent on parameters N_P and γ . . 116
3.21. Examplary scatter plots for results of the feature detector 117
3.22. Results for 2D MRI data 119
3.23. Results for 2D US data 120
3.24. Results for 2D US data (Detail) 121
3.25. Results for 3D MRI data 122

4.1. Motivation: Adaptation via introduction of additional degrees of freedom 132
4.2. Hierarchical split and join operation 133
4.3. Propagation of landmark associations along a hierarchical tree 134
4.4. Rule set for forwarding associations along a hierarchical tree. 135
4.5. Forwarding rule of three association states 136
4.6. Properties of an association between a node and a landmark 137
4.7. Transformation path for a landmark 139
4.8. Clustering as means to cope with multimodal response 140
4.9. Clustering based on the maximum distance to the complement center . . 142
4.10. Rearrangement of associations 144
4.11. Example: Nodes competing for an association to a landmark 146

4.12. Example: Soft state evolution of initially strong association 147
4.13. Six basic artificial motion patterns studied 151
4.14. Initial conditions for assessing clustering and rearrangement 151
4.15. Two samples: Simulation of non-perturbed artificial motion 155
4.16. Sample: Assignment of landmarks to the nodes of the clustered hierarchy 155
4.17. Level wise errors in different test sequences 156
4.18. Detail result of clustering artificial data 157
4.19. Result of clustering the motion in the noisy *Star* data set 158
4.20. Sample timeline of Jaccard similarity when rearranging associations . . . 159
4.21. Relationship of adaptation time to reinforcement 159
4.22. Sample cases of introduction of noise into the case 160
4.23. Experimental results for the adaptation in the noise free situation 161
4.24. Three particular settings for clustering and rearrangement 162
4.25. Experimental results for the adaptation on medical image data 163
4.26. Comparison of level 1 and 2 error, with and without attractive term . . . 163
4.27. Result sample: Change in normalized cross correlation 164
4.28. Effect of three different modes of operation on establishment of clusters . 165
4.29. Two sample cases of attractive reinforcement 166
4.30. Hierarchical tree and deformation vector fields for levels 2 and 3 167
4.31. Sample comparison of two-level model and dynamic model 168
4.32. Discussion of the impact of spatial jitter noise 172
4.33. Sample image to discuss a potential deadlock situation 175

A.1. Example: First steps of an unsuccessful experiment 196
A.2. Example: First steps of tracking in the Gaussian reference model 196
A.3. Predicted and empirical success probabilities 197
A.4. Extended graphics: Residual error augmented by estimates of optimal parametrization 197

List of Tables

2.1. Fitting results of the Critical Step Model 73

3.1. Chosen measurement functions $\mathcal{L}$ and related choice of weights w_i. 97
3.2. Settings of the two level algorithm, applied to 2D liver MRI data 109
3.3. Settings of the two level algorithm, applied to 2D liver MRI data 111
3.4. Settings of the two level algorithm, applied to liver MRI data 114
3.5. Results for motion tracking in EPI liver images against expert annotation to the test data set 118

4.1. Settings for joint clustering and rearrangement experiments on artificial data. 153

List of Algorithms

1.1. Particle Filter step 49

2.1. Simulation of artificial tracking experiment 54

3.1. Generic filter block 95
3.2. Computation of result measures for evaluation 109

4.1. Association clustering algorithm 143
4.2. Implementation of clustering 143
4.3. Association rearrangement algorithm 149
4.4. Graphical depiction of association rearrangement process 149
4.5. Final hierarchical tracking scheme 150

5.1. Developer stage parametrization 190

Bibliography

[ACA12] E. Arce-Santana, D. U. Campos-Delgado, and A. Alba. "Affine Image Registration Guided by Particle Filter". In: *IET Image Processing* 6.5 (July 2012), pp. 455–462.

[ACG96] A. A. Amini, R. W. Curwen, and J. C. Gore. "Snakes and Splines for Tracking Non-Rigid Heart Motion". In: *Computer Vision — ECCV '96*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Apr. 1996, pp. 249–261.

[Ahm+12] H. U. Ahmed, R. G. Hindley, L. Dickinson, A. Freeman, A. P. Kirkham, M. Sahu, R. Scott, C. Allen, J. Van der Meulen, and M. Emberton. "Focal Therapy for Localised Unifocal and Multifocal Prostate Cancer: A Prospective Development Study". In: *The Lancet. Oncology* 13.6 (June 2012), pp. 622–632.

[AK05] U. Amaldi and G. Kraft. "Radiotherapy with Beams of Carbon Ions". In: *Reports on Progress in Physics* 68.8 (2005), p. 1861.

[AMR08] J. R. Adler, A. Muacevic, and P. Romanelli. "CyberKnife Radiosurgery". In: *Principles and Practice of Stereotactic Radiosurgery*. Springer, New York, NY, 2008, pp. 171–178.

[Arc+13] E. Arce-Santana, D. U. Campos-Delgado, F. Vigueras-Gómez, I. Reducindo, and A. R. Mejía-Rodríguez. "Non-Rigid Multimodal Image Registration Based on the Expectation-Maximization Algorithm". In: *Image and Video Technology*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Oct. 2013, pp. 36–47.

[Asi+12] A. D. Asimakopoulos, R. Miano, G. Virgili, G. Vespasiani, and E. Finazzi Agrò. "HIFU as Salvage First-Line Treatment for Palpable, TRUS-Evidenced, Biopsy-Proven Locally Recurrent Prostate Cancer after Radical Prostatectomy: A Pilot Study". In: *Urologic Oncology* 30.5 (Sept. 2012), pp. 577–583.

[Aub+12] V. Auboiroux, L. Petrusca, M. Viallon, T. Goget, C. D. Becker, and R. Salomir. "Ultrasonography-Based 2D Motion-Compensated HIFU Sonication Integrated with Reference-Free MR Temperature Monitoring: A Feasibility Study Ex Vivo". In: *Physics in Medicine and Biology* 57.10 (May 2012), N159–171.

[Avk+16] V. Avkshtol, Y. Dong, S. B. Hayes, M. A. Hallman, R. A. Price, M. L. Sobczak, E. M. Horwitz, and N. G. Zaorsky. "A Comparison of Robotic Arm versus Gantry Linear Accelerator Stereotactic Body Radiation Therapy for Prostate Cancer". In: *Research and Reports in Urology* 8 (Aug. 2016), pp. 145–158.

[AY04] S. Ahn and B. Yi. "A Feasibility Study on the Prediction of Tumour Location in the Lung from Skin Motion". In: *The British journal of radiology* 77.919 (2004), pp. 588–96.

[Bai+07] M. Baiker, J. Milles, A. M. Vossepoel, I. Que, E. L. Kaijzel, C. W. G. M. Lowik, J. H. C. Reiber, J. Dijkstra, and B. P. F. Lelieveldt. "Fully Automated Whole-Body Registration in Mice Using an Articulated Skeleton Atlas". In: *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. Apr. 2007, pp. 728–731.

[Bal+98] J. M. Balter, K. L. Lam, C. J. McGinn, T. S. Lawrence, and R. K. Ten Haken. "Improvement of CT-Based Treatment-Planning Models of Abdominal Targets Using Static Exhale Imaging". In: *International Journal of Radiation Oncology, Biology, Physics* 41.4 (July 1998), pp. 939–943.

[Bas+12] R. Baskar, K. A. Lee, R. Yeo, and K.-W. Yeoh. "Cancer and Radiation Therapy: Current Advances and Future Directions". In: *International Journal of Medical Sciences* 9.3 (Feb. 2012), pp. 193–199.

[BBC04] N. Bansal, A. Blum, and S. Chawla. "Correlation Clustering". In: *Machine Learning* 56.1-3 (July 2004), pp. 89–113.

[BDH09] Y. Bar-Shalom, F. Daum, and J. Huang. "The Probabilistic Data Association Filter". In: *IEEE Control Systems* 29.6 (Dec. 2009), pp. 82–100.

[BH01] K. Briechle and U. D. Hanebeck. "Template Matching Using Fast Normalized Cross Correlation". In: *Optical Pattern Recognition XII*. Vol. 4387. International Society for Optics and Photonics, Mar. 2001, pp. 95–103.

[Bla+04] A. Blana, B. Walter, S. Rogenhofer, and W. F. Wieland. "High-Intensity Focused Ultrasound for the Treatment of Localized Prostate Cancer: 5-Year Experience". In: *Urology* 63.2 (Feb. 2004), pp. 297–300.

[Bla04] S. S. Blackman. "Multiple Hypothesis Tracking for Multiple Target Tracking". In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (Jan. 2004), pp. 5–18.

[BM92] P. J. Besl and N. D. McKay. "A Method for Registration of 3-D Shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992), pp. 239–256.

[Bri+14] L. Brix, S. Ringgaard, T. S. Sørensen, and P. R. Poulsen. "Three-Dimensional Liver Motion Tracking Using Real-Time Two-Dimensional MRI". In: *Medical Physics* 41.4 (Apr. 2014).

[BTG06] H. Bay, T. Tuytelaars, and L. V. Gool. "SURF: Speeded Up Robust Features". In: *Computer Vision – ECCV 2006*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, May 2006, pp. 404–417.

[Buc+12] C. Buchta, M. Kober, I. Feinerer, and K. Hornik. "Spherical K-Means Clustering". In: *Journal of Statistical Software* 50.10 (Sept. 2012), pp. 1–22.

[CB84] K.-C. Chang and Y. Bar-Shalom. "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers". In: *IEEE Transactions on Automatic Control* 29.7 (July 1984), pp. 585–594.

[CCP11] C. Canton-Ferrer, J. R. Casas, and M. Pardàs. "Human Motion Capture Using Scalable Body Models". In: *Computer Vision and Image Understanding* 115.10 (Oct. 2011), pp. 1363–1374.

[Cel+14] Z. Celicanin, V. Auboiroux, O. Bieri, L. Petrusca, F. Santini, M. Viallon, K. Scheffler, and R. Salomir. "Real-Time Method for Motion-Compensated MR Thermometry and MRgHIFU Treatment in Abdominal Organs: MRgHIFU Treatment Method in Abdominal Organs". In: *Magnetic resonance in medicine* 72 (Oct. 2014).

[Cel+18] Z. Celicanin, G. Manasseh, L. Petrusca, K. Scheffler, V. Auboiroux, L. A. Crowe, J.-N. Hyacinthe, Y. Natsuaki, F. Santini, C. D. Becker, S. Terraz, O. Bieri, and R. Salomir. "Hybrid Ultrasound-MR Guided HIFU Treatment Method with 3D Motion Compensation". In: *Magnetic Resonance in Medicine* 79.5 (May 2018), pp. 2511–2523.

[CET01] T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active Appearance Models". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 23.6 (June 2001), pp. 681–685.

[CET98] T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active Appearance Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Springer, 1998, pp. 484–498.

[Cha+09] V. Chalasani, C. H. Martinez, D. Lim, and J. Chin. "Salvage HIFU for Recurrent Prostate Cancer after Radiotherapy". In: *Prostate Cancer and Prostatic Diseases* 12.2 (2009), pp. 124–129.

[Cha+15a] L. A. Chanel, F. Nageotte, J. Vappou, J. Luo, L. Cuvillon, and M. de Mathelin. "Robotized High Intensity Focused Ultrasound (HIFU) System for Treatment of Mobile Organs Using Motion Tracking by Ultrasound Imaging: An in Vitro Study". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Aug. 2015, pp. 2571–2575.

[Cha+15b] S. Chawla, K. Makarychev, T. Schramm, and G. Yaroslavtsev. "Near Optimal LP Rounding Algorithm for CorrelationClustering on Complete and Complete K-Partite Graphs". In: *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*. STOC '15. New York, NY, USA: ACM, 2015, pp. 219–228.

[Che+07] J. Chen, R. J. Lee, D. Handrahan, and W. T. Sause. "Intensity-Modulated Radiotherapy Using Implanted Fiducial Markers with Daily Portal Imaging: Assessment of Prostate Organ Motion". In: *International Journal of Radiation Oncology, Biology, Physics* 68.3 (July 2007), pp. 912–919.

[Che+10] A. Chen, M. A. Deeley, K. J. Niermann, L. Moretti, and B. M. Dawant. "Combining Registration and Active Shape Models for the Automatic Segmentation of the Lymph Node Regions in Head and Neck CT Images". In: *Medical Physics* 37.12 (Dec. 2010), pp. 6338–6346.

[Chu+08] E. O. Chukwu, E. Barasch, D. G. Mihalatos, A. Katz, J. Lachmann, J. Han, N. Reichek, and A. S. Gopal. "Relative Importance of Errors in Left Ventricular Quantitation by Two-Dimensional Echocardiography: Insights from Three-Dimensional Echocardiography and Cardiac Magnetic Resonance Imaging". In: *Journal of the American Society of Echocardiography: Official Publication of the American Society of Echocardiography* 21.9 (Sept. 2008), pp. 990–997.

[CKC17] P. Carmier, O. Kyrgyzov, and P.-H. Cournède. "A Critical Analysis of Resampling Strategies for the Regularized Particle Filter". In: *arXiv:1705.04219* (May 2017). arXiv: `1705.04219`.

[CKL13] L. Cehovin, M. Kristan, and A. Leonardis. "Robust Visual Tracking Using an Adaptive Coupled-Layer Visual Model". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.4 (Apr. 2013), pp. 941–953.

[CL90] J. C. McMillan and S. S. Lim. "Data Association Algorithms for Multiple Target Tracking". In: (July 1990), p. 37.

[Cla+07] D. Clark, I. T. Ruiz, Y. Petillot, and J. Bell. "Particle PHD Filter Multiple Target Tracking in Sonar Image". In: *IEEE Transactions on Aerospace and Electronic Systems* 43.1 (Jan. 2007), pp. 409–416.

[Cla76] J. H. Clark. "Hierarchical Geometric Models for Visible Surface Algorithms". In: *Communications of the ACM* 19.10 (Oct. 1976), pp. 547–554.

[Coo+95] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. "Active Shape Models-Their Training and Application". In: *Computer Vision and Image Understanding* 61.1 (Jan. 1995), pp. 38–59.

[CR15] L. S. Chin and W. F. Regine. *Principles and Practice of Stereotactic Radiosurgery*. Springer, Jan. 2015.

[Cri15] A. Cristea. "Ultrasound Tissue Characterization Using Speckle Statistics". Theses. Universite Claude Bernard Lyon 1, Dec. 2015.

[CRM00] D. Comaniciu, V. Ramesh, and P. Meer. "Real-Time Tracking of Non-Rigid Objects Using Mean Shift". In: *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings.* Vol. 2. 2000, 142–149 vol.2.

[Cro+17] S. Crouzet, A. Blana, F. J. Murat, G. Pasticier, S. C. W. Brown, G. N. Conti, R. Ganzer, O. Chapet, A. Gelet, C. G. Chaussy, C. N. Robertson, S. Thuroff, and J. F. Ward. "Salvage High-Intensity Focused Ultrasound (HIFU) for Locally Recurrent Prostate Cancer after Failed Radiation Therapy: Multi-Institutional Analysis of 418 Patients". In: *BJU international* 119.6 (June 2017), pp. 896–904.

[CT17] C. G. Chaussy and S. Thüroff. "High-Intensity Focused Ultrasound for the Treatment of Prostate Cancer: A Review". In: *Journal of Endourology* 31.S1 (Apr. 2017), S30–S37.

[CT92] T. F. Cootes and C. J. Taylor. "Active Shape Models — 'Smart Snakes'". In: *BMVC92*. Springer, London, 1992, pp. 266–275.

[CWF13] L. Chen, H. Wei, and J. Ferryman. "A Survey of Human Motion Analysis Using Depth Imagery". In: *Pattern Recognition Letters*. Smart Approaches for Human Action Recognition 34.15 (Nov. 2013), pp. 1995–2006.

[CWR10] N. R. Colledge, B. R. Walker, and S. H. Ralston. *Davidson's Principles and Practice of Medicine: With STUDENT CONSULT Online Access*. 21st revised edition. Edinburgh New York: Churchill Livingstone, Mar. 2010.

[Dam+14] V. Damerjian, O. Tankyevych, N. Souag, and E. Petit. "Speckle Characterization Methods in Ultrasound Images – A Review". In: *IRBM* 35.4 (Sept. 2014), pp. 202–213.

[Dav+94] S. C. Davies, A. L. Hill, R. B. Holmes, M. Halliwell, and P. C. Jackson. "Ultrasound Quantitation of Respiratory Organ Motion in the Upper Abdomen". In: *The British Journal of Radiology* 67.803 (Nov. 1994), pp. 1096–1102.

[DBR00] J. Deutscher, A. Blake, and I. Reid. "Articulated Body Motion Capture by Annealed Particle Filtering". In: *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings.* Vol. 2. 2000, 126–133 vol.2.

[DC05] R. Douc and O. Cappe. "Comparison of Resampling Schemes for Particle Filtering". In: *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.* Sept. 2005, pp. 64–69.

[De +15] V. De Luca, E. J. Harris, M. A. L. Bell, and C. Tanner. "Challenge on Liver Ultrasound Tracking CLUST 2015". In: *Proceedings of the MICCAI 2015 Workshop CLUST 2015* (2015).

[Dem+06] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. "Correlation Clustering in General Weighted Graphs". In: *Theoretical Computer Science*. Approximation and Online Algorithms 361.2 (Sept. 2006), pp. 172–187.

[Der+15] A. Derksen, S. Heldmann, T. Polzin, and B. Berkels. "Image Registration with Sliding Motion Constraints for 4D CT Motion Correction". In: Mar. 2015.

[DFG01] A. Doucet, N. de Freitas, and N. Gordon. "An Introduction to Sequential Monte Carlo Methods". In: *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, New York, NY, 2001, pp. 3–14.

[DGA00] A. Doucet, S. Godsill, and C. Andrieu. "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering". In: *Statistics and Computing* 10.3 (July 2000), pp. 197–208.

[DK04] B. Desjardins and E. A. Kazerooni. "ECG-Gated Cardiac CT". In: *American Journal of Roentgenology* 182.4 (Apr. 2004), pp. 993–1010.

[DPM15] C. Dalitz, R. Pohle-Frohlich, and T. Michalk. "Point Spread Functions and Deconvolution of Ultrasonic Images". In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 62.3 (Mar. 2015), pp. 531–544.

[DR05] J. Deutscher and I. Reid. "Articulated Body Motion Capture by Stochastic Search". In: *International Journal of Computer Vision* 61.2 (Feb. 2005), pp. 185–205.

[DST15] V. De Luca, G. Székely, and C. Tanner. "Estimation of Large-Scale Organ Motion in B-Mode Ultrasound Image Sequences: A Survey". In: *Ultrasound in Medicine & Biology* 41.12 (Dec. 2015), pp. 3044–3062.

[Dub+08] T. J. Dubinsky, C. Cuevas, M. K. Dighe, O. Kolokythas, and J. H. Hwang. "High-Intensity Focused Ultrasound: Current Potential and Oncologic Applications". In: *AJR. American journal of roentgenology* 190.1 (Jan. 2008), pp. 191–199.

[Duc77] J. Duchon. "Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces". In: *Constructive Theory of Functions of Several Variables*. Lecture Notes in Mathematics. Springer, Berlin, Heidelberg, 1977, pp. 85–100.

[EP08] G. D. Evangelidis and E. Z. Psarakis. "Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization". In: *IEEE transactions on pattern analysis and machine intelligence* 30.10 (Oct. 2008), pp. 1858–1865.

[EPV93] P. A. van den Elsen, E. J. D. Pol, and M. A. Viergever. "Medical Image Matching-a Review with Classification". In: *IEEE Engineering in Medicine and Biology Magazine* 12.1 (Mar. 1993), pp. 26–39.

[FB81] M. A. Fischler and R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* 24.6 (June 1981), pp. 381–395.

[FBS80] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. “Multi-Target Tracking Using Joint Probabilistic Data Association”. In: *1980 19th IEEE Conference on Decision and Control Including the Symposium on Adaptive Processes*. Dec. 1980, pp. 807–812.

[Fer+01] M. Ferrant, A. Nabavi, B. Macq, F. A. Jolesz, R. Kikinis, and S. K. Warfield. “Registration of 3-D Intraoperative MR Images of the Brain Using a Finite-Element Biomechanical Model”. In: *IEEE transactions on medical imaging* 20.12 (Dec. 2001), pp. 1384–1397.

[FF56] J. L. Ford and D. Fulkerson. “Maximal Flow through a Network”. In: *Canadian Journal of Mathematics* (1956).

[FH06] P. F. Felzenszwalb and D. P. Huttenlocher. “Efficient Belief Propagation for Early Vision”. In: *International Journal of Computer Vision* 70.1 (May 2006), pp. 41–54.

[FH75] K. Fukunaga and L. Hostetler. “The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition”. In: *IEEE Transactions on Information Theory* 21.1 (Jan. 1975), pp. 32–40.

[FLB12] A. Forner, J. M. Llovet, and J. Bruix. “Hepatocellular Carcinoma”. In: *Lancet (London, England)* 379.9822 (Mar. 2012), pp. 1245–1255.

[Fle70] R. Fletcher. “A New Approach to Variable Metric Algorithms”. In: *The Computer Journal* 13.3 (Jan. 1970), pp. 317–322.

[For+03] E. C. Ford, G. S. Mageras, E. Yorke, and C. C. Ling. “Respiration-Correlated Spiral CT: A Method of Measuring Respiratory-Induced Anatomic Motion for Radiation Treatment Planning”. In: *Medical Physics* 30.1 (Jan. 2003), pp. 88–97.

[Fu+17] Y. Fu, S. Liu, H. Li, and D. Yang. “Automatic and Hierarchical Segmentation of the Human Skeleton in CT Images”. In: *Physics in Medicine and Biology* 62.7 (2017), pp. 2812–2833.

[Gag+07] B. Gagel, C. Demirel, A. Kientopf, M. Pinkawa, M. Piroth, S. Stanzel, C. Breuer, B. Asadpour, T. Jansen, R. Holy, J. E. Wildberger, and M. J. Eble. “Active Breathing Control (ABC): Determination and Reduction of Breathing-Induced Organ Motion in the Chest”. In: *International Journal of Radiation Oncology, Biology, Physics* 67.3 (Mar. 2007), pp. 742–749.

[GD96] D. M. Gavrila and L. S. Davis. “3-D Model-Based Tracking of Humans in Action: A Multi-View Approach”. In: *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. June 1996, pp. 73–80.

[Gea35] R. C. Geary. “The Ratio of the Mean Deviation to the Standard Deviation as a Test of Normality”. In: *Biometrika* 27.3-4 (1935), pp. 310–332.

[Gie+05] D. P. Gierga, J. Brewer, G. C. Sharp, M. Betke, C. G. Willett, and G. T. Y. Chen. "The Correlation between Internal and External Markers for Abdominal Tumors: Implications for Respiratory Gating". In: *International Journal of Radiation Oncology*Biology*Physics* 61.5 (Apr. 2005), pp. 1551–1558.

[Gir+07] P. Giraud, Y. De Rycke, J.-C. Rosenwald, and J.-M. Cosset. "Conformal Radiotherapy Planning for Lung Cancer: Analysis of Set-up Uncertainties". In: *Cancer Investigation* 25.1 (Feb. 2007), pp. 38–46.

[Gir+13] P. Giraud, A. Houle, P. Giraud, and A. Houle. "Respiratory Gating for Radiotherapy: Main Technical Aspects and Clinical Benefits, Respiratory Gating for Radiotherapy: Main Technical Aspects and Clinical Benefits". In: *International Scholarly Research Notices, International Scholarly Research Notices* 2013 (Mar. 2013).

[Glo+10] B. Glocker, T. Hauke Heibel, N. Navab, P. Kohli, and C. Rother. "TriangleFlow: Optical Flow with Triangulation-Based Higher-Order Likelihoods". In: (2010).

[God+16] F. Godenschweger, U. Kägebein, D. Stucht, U. Yarach, A. Sciarra, R. Yakupov, F. Lüsebrink, P. Schulze, and O. Speck. "Motion Correction in MRI of the Brain". In: *Physics in medicine and biology* 61 (Feb. 2016), R32–R56.

[Gol70] D. Goldfarb. "A Family of Variable-Metric Methods Derived by Variational Means". In: *Mathematics of Computation* 24.109 (1970), pp. 23–26.

[Goo91] C. Goodall. "Procrustes Methods in the Statistical Analysis of Shape". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 53.2 (1991), pp. 285–339.

[Gor06] G. Gorrell. "Generalized Hebbian Algorithm for Incremental Singular Value Decomposition in Natural Language Processing." In: Jan. 2006.

[Haa+08] G. P. Haas, N. Delongchamps, O. W. Brawley, C. Y. Wang, and G. de la Roza. "The Worldwide Epidemiology of Prostate Cancer: Perspectives from Autopsy Studies". In: *The Canadian Journal of Urology* 15.1 (Feb. 2008), pp. 3866–3871.

[Hau+06] A. J. Hautala, A. M. Kiviniemi, T. H. Mäkikallio, H. Kinnunen, S. Nissilä, H. V. Huikuri, and M. P. Tulppo. "Individual Differences in the Responses to Endurance and Resistance Training". In: *European Journal of Applied Physiology* 96.5 (Mar. 2006), pp. 535–542.

[Heb50] D. Hebb. "Organization of Behavior". In: *Journal of Clinical Psychology* 6.3 (July 1950), pp. 307–307.

[Hei+13] M. P. Heinrich, M. Jenkinson, M. Brady, and J. A. Schnabel. "MRF-Based Deformable Registration and Ventilation Estimation of Lung CT". In: *IEEE Transactions on Medical Imaging* 32.7 (July 2013), pp. 1239–1248.

[HM04] E. Haber and J. Modersitzki. “Numerical Methods for Image Registration”. In: (2004).

[Hoi+04] J. D. P. Hoisak, K. E. Sixel, R. Tirona, P. C. F. Cheung, and J.-P. Pignol. “Correlation of Lung Tumor Motion with External Surrogate Indicators of Respiration”. In: *International Journal of Radiation Oncology*Biology*Physics* 60.4 (Nov. 2004), pp. 1298–1306.

[Hoi+06] J. D. P. Hoisak, K. E. Sixel, R. Tirona, P. C. F. Cheung, and J.-P. Pignol. “Prediction of Lung Tumour Position Based on Spirometry and on Abdominal Displacement: Accuracy and Reproducibility”. In: *Radiotherapy and Oncology* 78.3 (Mar. 2006), pp. 339–346.

[HOR07] J. Hansegård, F. Orderud, and S. I. Rabben. “Real-Time Active Shape Models for Segmentation of 3D Cardiac Ultrasound”. In: *Computer Analysis of Images and Patterns*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Aug. 2007, pp. 157–164.

[HS81] B. K. P. Horn and B. G. Schunck. “Determining Optical Flow”. In: *Artificial Intelligence* 17 (1981), pp. 185–203.

[HSG06] J. D. Hol, T. B. Schon, and F. Gustafsson. “On Resampling Algorithms for Particle Filters”. In: *2006 IEEE Nonlinear Statistical Signal Processing Workshop*. Sept. 2006, pp. 79–82.

[Ht95] C. R. Hill and G. R. ter Haar. “Review Article: High Intensity Focused Ultrasound–Potential for Cancer Treatment”. In: *The British Journal of Radiology* 68.816 (Dec. 1995), pp. 1296–1303.

[Hu+08] Y. Hu, D. Morgan, H. U. Ahmed, D. Pendsé, M. Sahu, C. Allen, M. Emberton, D. Hawkes, and D. Barratt. “A Statistical Motion Model Based on Biomechanical Simulations for Data Fusion during Image-Guided Prostate Interventions”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* 11.Pt 1 (2008), pp. 737–744.

[Hu+11] Y. Hu, W. R. Ledoux, M. Fassbind, E. S. Rohr, B. J. Sangeorzan, and D. Haynor. “Multi-Rigid Image Segmentation and Registration for the Analysis of Joint Motion from Three-Dimensional Magnetic Resonance Imaging”. In: *Journal of Biomechanical Engineering* 133.10 (Oct. 2011), p. 101005.

[Hus07] Z. L. Husz. *Evaluation of a Hierarchical Partitioned Particle Filter with Action Primitives*. 2007.

[HWG07] Z. L. Husz, A. M. Wallace, and P. R. Green. “Human Activity Recognition with Action Primitives”. In: *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. Sept. 2007, pp. 330–335.

[IB98] M. Isard and A. Blake. “CONDENSATION—Conditional Density Propagation for Visual Tracking”. In: *International Journal of Computer Vision* 29.1 (Aug. 1998), pp. 5–28.

[Ike+16] T. Ikeda, S. Yoshizawa, N. Koizumi, M. Mitsuishi, and Y. Matsumoto. "Focused Ultrasound and Lithotripsy". In: *Advances in Experimental Medicine and Biology* 880 (2016), pp. 113–129.

[Jad+14] R. Jadon, C. A. Pembroke, C. L. Hanna, N. Palaniappan, M. Evans, A. E. Cleves, and J. Staffurth. "A Systematic Review of Organ Motion and Image-Guided Strategies in External Beam Radiotherapy for Cervical Cancer". In: *Clinical Oncology (Royal College of Radiologists (Great Britain))* 26.4 (Apr. 2014), pp. 185–196.

[JR04] M. Jeraj and V. Robar. "Multileaf Collimator in Radiotherapy". In: *Radiology and Oncology* 38.3 (2004), pp. 235–240.

[JU04] S. J. Julier and J. K. Uhlmann. "Unscented Filtering and Nonlinear Estimation". In: *Proceedings of the IEEE* 92.3 (Mar. 2004), pp. 401–422.

[KA01] E. B. Koller-Meier and F. Ade. "Tracking Multiple Objects Using the Condensation Algorithm". In: *Robotics and Autonomous Systems*. European Workshop on Advanced Mobile Robots 34.2 (Feb. 2001), pp. 93–105.

[Kal60] R. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME – Journal of Basic Engineering* 82 (Series D) (1960), pp. 35–45.

[Kea+06] P. J. Keall, G. S. Mageras, J. M. Balter, R. S. Emery, K. M. Forster, S. B. Jiang, J. M. Kapatoes, D. A. Low, M. J. Murphy, B. R. Murray, C. R. Ramsey, M. B. Van Herk, S. S. Vedam, J. W. Wong, and E. Yorke. "The Management of Respiratory Motion in Radiation Oncology Report of AAPM Task Group 76". In: *Medical Physics* 33.10 (Oct. 2006), pp. 3874–3900.

[Kei05] Y. T. Keith R Britton. "Evaluation of Inter- and Intrafraction Organ Motion during Intensity Modulated Radiation Therapy (IMRT) for Localized Prostate Cancer Measured by a Newly Developed On-Board Image-Guided System". In: *Radiation medicine* 23.1 (2005), pp. 14–24.

[Ken05] J. E. Kennedy. "High-Intensity Focused Ultrasound in the Treatment of Solid Tumours". In: *Nature Reviews Cancer* 5.4 (Apr. 2005), pp. 321–327.

[KFC11] S. Kiriyanthan, K. Fundana, and P. Cattin. "Discontinuity Preserving Registration of Abdominal MR Images with Apparent Sliding Organ Motion". In: Sept. 2011, pp. 231–239.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing". In: *SCIENCE* 220.4598 (1983), pp. 671–680.

[Kie53] J. Kiefer. "Sequential Minimax Search for a Maximum". In: *Proceedings of the American Mathematical Society* 4.3 (1953), pp. 502–506.

[Kim+01] D. J. Kim, B. R. Murray, R. Halperin, and W. H. Roa. "Held-Breath Self-Gating Technique for Radiotherapy of Non-Small-Cell Lung Cancer: A Feasibility Study". In: *International Journal of Radiation Oncology, Biology, Physics* 49.1 (Jan. 2001), pp. 43–49.

[Kir+16] S. Kiriyanthan, K. Fundana, T. Majeed, and P. C. Cattin. "Discontinuity Preserving Image Registration through Motion Segmentation: A Primal-Dual Approach". In: *Computational and Mathematical Methods in Medicine* 2016 (2016).

[KMI02] S. Kaneko, I. Murase, and S. Igarashi. "Robust Image Registration by Increment Sign Correlation". In: *Pattern Recognition* 35.10 (Oct. 2002), pp. 2223–2234.

[KMP06] E. Klein, S. Mutic, and J. A. Purdy. "Treatment Aids for External Beam Radiotherapy". In: *Technical Basis of Radiation Therapy*. Medical Radiology. Springer, Berlin, Heidelberg, 2006, pp. 167–177.

[Koc+04] N. Koch, H. H. Liu, G. Starkschall, M. Jacobson, K. Forster, Z. Liao, R. Komaki, and C. W. Stevens. "Evaluation of Internal Lung Motion for Respiratory-Gated Radiotherapy Using MRI: Part I—Correlating Internal Lung Motion with Skin Fiducial Motion". In: *International Journal of Radiation Oncology * Biology * Physics* 60.5 (Dec. 2004), pp. 1459–1472.

[Kom06] N. Komodakis. "Optimization Algorithms for Discrete Markov Random Fields, with Applications to Computer Vision". PhD thesis. University of Crete, Computer Science Department, 2006.

[KSI03] S. Kaneko, Y. Satoh, and S. Igarashi. "Using Selective Correlation Coefficient for Robust Image Registration". In: *Pattern Recognition* 36.5 (May 2003), pp. 1165–1173.

[Kuh06] Kuhn H. W. "The Hungarian Method for the Assignment Problem". In: *Naval Research Logistics Quarterly* 2.1-2 (Oct. 2006), pp. 83–97.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes Active Contour Models". In: *International Journal of Computer Vision* 1.4 (Jan. 1988), pp. 321–331.

[Lag+01] F. J. Lagerwaard, J. R. Van Sornsen de Koste, M. R. Nijssen-Visser, R. H. Schuchhard-Schipper, S. S. Oei, A. Munne, and S. Senan. "Multiple "Slow" CT Scans for Incorporating Lung Tumor Mobility in Radiotherapy Planning". In: *International Journal of Radiation Oncology, Biology, Physics* 51.4 (Nov. 2001), pp. 932–937.

[Lax+94] I. Lax, H. Blomgren, I. Näslund, and R. Svanström. "Stereotactic Radiotherapy of Malignancies in the Abdomen. Methodological Aspects". In: *Acta Oncologica (Stockholm, Sweden)* 33.6 (1994), pp. 677–683.

[Lev+05] W. P. Levin, H. Kooy, J. S. Loeffler, and T. F. DeLaney. "Proton Beam Therapy". In: *British Journal of Cancer* 93.8 (Oct. 2005), pp. 849–854.

[Lev44] K. Levenberg. “A Method for the Solution of Certain Non-Linear Problems in Least Squares”. In: *Quarterly Journal of Applied Mathmatics* II.2 (1944), pp. 164–168.

[Li+07] Y.-Y. Li, W.-H. Sha, Y.-J. Zhou, and Y.-Q. Nie. “Short and Long Term Efficacy of High Intensity Focused Ultrasound Therapy for Advanced Hepatocellular Carcinoma”. In: *Journal of Gastroenterology and Hepatology* 22.12 (Dec. 2007), pp. 2148–2154.

[Lim+12] K.-C. Lim, P. K.-H. Chow, J. C. Allen, F. J. Siddiqui, E. S.-Y. Chan, and S.-B. Tan. “Systematic Review of Outcomes of Liver Resection for Early Hepatocellular Carcinoma within the Milan Criteria”. In: *The British Journal of Surgery* 99.12 (Dec. 2012), pp. 1622–1629.

[Lin+08] M. G. Linguraru, N. V. Vasilyev, G. R. Marx, W. Tworetzky, P. J. Del Nido, and R. D. Howe. “Fast Block Flow Tracking of Atrial Septal Defects in 4D Echocardiography”. In: *Medical image analysis* 12.4 (Aug. 2008), pp. 397–412.

[Lin+12] C. K. Lin, F. C. Lin, F. L. Lian, K. H. Chang, M. C. Ho, J. Y. Yen, and Y. Y. Chen. “Ultrasound Image-Guided Algorithms for Tracking Liver Motion”. In: *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. July 2012, pp. 51–56.

[Liu+15] Y. Liu, P. Wang, Y. Gao, J. Wang, and R. Fu. “Data Association Combined with the Probability Hypothesis Density Filter for Multi-Target Tracking”. In: *2015 International Conference on Wireless Communications Signal Processing (WCSP)*. Oct. 2015, pp. 1–6.

[LK81] B. D. Lucas and T. Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision”. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI’81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[Llo82] S. Lloyd. “Least Squares Quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137.

[Loe+16] A. J. Loeve, J. Al-Issawi, F. Fernandez-Gutiérrez, T. Langø, J. Strehlow, S. Haase, M. Matzko, A. Napoli, A. Melzer, and J. Dankelman. “Workflow and Intervention Times of MR-Guided Focused Ultrasound – Predicting the Impact of New Techniques”. In: *Journal of Biomedical Informatics* 60 (Apr. 2016), pp. 38–48.

[Loe04] H. A. Loeliger. “An Introduction to Factor Graphs”. In: *IEEE Signal Processing Magazine* 21.1 (Jan. 2004), pp. 28–41.

[Lor+02] M. Lorenzo-Valdes, G. I. Sanchez-Ortiz, R. Mohiaddin, and D. Rueckert. “Atlas-Based Segmentation and Tracking of 3D Cardiac MR Images Using Non-Rigid Registration”. In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sept. 2002, pp. 642–650.

[Lot+16] J. Lotz, J. Olesch, B. Müller, T. Polzin, P. Galuschka, J. M. Lotz, S. Heldmann, H. Laue, M. González-Vallinas, A. Warth, B. Lahrmann, N. Grabe, O. Sedlaczek, K. Breuhahn, and J. Modersitzki. "Patch-Based Nonlinear Image Registration for Gigapixel Whole Slide Images". In: *IEEE Transactions on Biomedical Engineering* 63.9 (Sept. 2016), pp. 1812–1819.

[Low+03] D. A. Low, M. Nystrom, E. Kalinin, P. Parikh, J. F. Dempsey, J. D. Bradley, S. Mutic, S. H. Wahab, T. Islam, G. Christensen, D. G. Politte, and B. R. Whiting. "A Method for the Reconstruction of Four-Dimensional Synchronized CT Scans Acquired during Free Breathing". In: *Medical Physics* 30.6 (June 2003), pp. 1254–1263.

[Low+05] D. A. Low, P. J. Parikh, W. Lu, J. F. Dempsey, S. H. Wahab, J. P. Hubenschmidt, M. M. Nystrom, M. Handoko, and J. D. Bradley. "Novel Breathing Motion Model for Radiotherapy". In: *International Journal of Radiation Oncology, Biology, Physics* 63.3 (Nov. 2005), pp. 921–929.

[Low04] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110.

[Low99] D. G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–.

[Luc+15] V. D. Luca, T. Benz, S. Kondo, L. König, D. Lübke, S. Rothlübbers, O. Somphone, S. Allaire, M. A. L. Bell, D. Y. F. Chung, A. Cifor, C. Grozea, M. Günther, J. Jenne, T. Kipshagen, M. Kowarschik, N. Navab, J. Rühaak, J. Schwaab, and C Tanner. "The 2014 Liver Ultrasound Tracking Benchmark". In: *Physics in Medicine and Biology* 60.14 (2015), p. 5571.

[Luj+99] A. E. Lujan, E. W. Larsen, J. M. Balter, and R. K. T. Haken. "A Method for Incorporating Organ Motion Due to Breathing into 3D Dose Calculations". In: *Medical Physics* 26.5 (May 1999), pp. 715–720.

[Ma+10] M. Ma, M. van Stralen, J. H. C. Reiber, J. G. Bosch, and B. P. F. Lelieveldt. "Model Driven Quantification of Left Ventricular Function from Sparse Single-Beat 3D Echocardiography". In: *Medical Image Analysis* 14.4 (Aug. 2010), pp. 582–593.

[Mac+13] J. Maclaren, M. Herbst, O. Speck, and M. Zaitsev. "Prospective Motion Correction in Brain Imaging: A Review". In: *Magnetic Resonance in Medicine* 69.3 (Mar. 2013), pp. 621–636.

[Mal+13] G. Malietzis, L. Monzon, J. Hand, H. Wasan, E. Leen, M. Abel, A. Muhammad, P. Price, and P. Abel. "High-Intensity Focused Ultrasound: Advances in Technology and Experimental Trials Support Enhanced Utility of Focused Ultrasound Surgery in Oncology". In: *The British Journal of Radiology* 86.1024 (Apr. 2013).

[Mar07] D. Marimón Sanjuán. "Advances in Top-Down and Bottom-Up Approaches to Video-Based Camera Tracking". PhD thesis. École Polytechnique Fédérale de Lausanne, 2007.

[Mar63] D. Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". In: *SIAM Journal on Applied Mathematics* 11.2 (1963), pp. 431–441.

[MB04] I. Matthews and S. Baker. "Active Appearance Models Revisited". In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 135–164.

[MB99] J. MacCormick and A. Blake. "A Probabilistic Exclusion Principle for Tracking Multiple Objects". In: *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.* Vol. 1. 1999, 572–578 vol.1.

[MC11] D. E. Maggio and D. A. Cavallaro. *Video Tracking: Theory and Practice.* 1st. Wiley Publishing, 2011.

[McC+13] J. R. McClelland, D. J. Hawkes, T. Schaeffter, and A. P. King. "Respiratory Motion Models: A Review". In: *Medical Image Analysis* 17.1 (Jan. 2013), pp. 19–42.

[Mej+11] A. R. Mejia-Rodriguez, E. R. Arce-Santana, E. Scalco, D. Tresoldi, M. O. Mendez, A. M. Bianchi, G. M. Cattaneo, and G. Rizzo. "Elastic Registration Based on Particle Filter in Radiotherapy Images with Brain Deformations". In: *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Aug. 2011, pp. 8049–8052.

[Mel+07] A. Melbourne, D. Atkinson, M. J. White, D. Collins, M. Leach, and D. Hawkes. "Registration of Dynamic Contrast-Enhanced MRI Using a Progressive Principal Component Registration (PPCR)". In: *Physics in Medicine & Biology* 52.17 (2007), p. 5147.

[Met+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092.

[MG01] T. Moeslund and E. Granum. "A Survey of Computer Vision-Based Human Motion Capture". In: *Computer Vision and Image Understanding* 81 (Mar. 2001), pp. 231–268.

[MG16] O. Mattausch and O. Goksel. "Image-Based PSF Estimation for Ultrasound Training Simulation". In: *Simulation and Synthesis in Medical Imaging.* Lecture Notes in Computer Science. Springer, Cham, Oct. 2016, pp. 23–33.

[MH03] J. Mitchelson and A. Hilton. "Simultaneous Pose Estimation of Multiple People Using Multiple-View Cues with Hierarchical Sampling". In: *In British Machine Vision Conference.* 2003.

[MH80] D. Marr and E. Hildreth. "Theory of Edge Detection". In: *Proceedings of the Royal Society of London. Series B, Biological Sciences* 207.1167 (Feb. 1980), pp. 187–217.

[MHK06] T. B. Moeslund, A. Hilton, and V. Krüger. "A Survey of Advances in Vision-Based Human Motion Capture and Analysis". In: *Computer Vision and Image Understanding*. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour 104.2–3 (Nov. 2006), pp. 90–126.

[MI00] J. MacCormick and M. Isard. "Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking". In: *Proceedings of the 6th European Conference on Computer Vision-Part II*. ECCV '00. London, UK, UK: Springer-Verlag, 2000, pp. 3–19.

[Mih+16] S. Mihcin, S. Rothluebbers, J. Schwaab, J. Strehlow, C. Tanner, S. Tretbar, T. Preusser, M. Günther, and J. Jenne. "Real Time MR and US Motion Tracking for Abdominal US/MRgFUS." In: Mar. 2016.

[ML12] C.-M. C. Ma and T. Lomax. *Proton and Carbon Ion Therapy*. Vol. Imaging in Medical Diagnosis and Therapy. Hoboken: CRC Press, 2012.

[Mog+12] M. H. Moghari, R. H. Chan, S. N. Hong, J. L. Shaw, L. A. Goepfert, K. V. Kissinger, B. Goddu, M. E. Josephson, W. J. Manning, and R. Nezafat. "Free-Breathing Cardiac MR with a Fixed Navigator Efficiency Using Adaptive Gating Window Size". In: *Magnetic Resonance in Medicine* 68.6 (Dec. 2012), pp. 1866–1875.

[Moh12] D. Mohr. "Model-Based High-Dimensional Pose Estimation with Application to Hand Tracking". PhD thesis. Bremen, 2012.

[Moi+06] V. Moiseenko, M. Liu, S. Kristensen, G. Gelowitz, and E. Berthelet. "Effect of Bladder Filling on Doses to Prostate and Organs at Risk: A Treatment Planning Study". In: *Journal of Applied Clinical Medical Physics* 8.1 (Apr. 2006), pp. 55–68.

[MS03] K. Mikolajczyk and C. Schmid. "A Performance Evaluation of Local Descriptors". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. June 2003, pp. 257–263.

[Mur04] M. J. Murphy. "Tracking Moving Organs in Real Time". In: *Seminars in Radiation Oncology*. High-Precision Radiation Therapy of Moving Targets 14.1 (Jan. 2004), pp. 91–100.

[MV98] J. B. A. Maintz and M. A. Viergever. "A Survey of Medical Image Registration". In: *Medical Image Analysis* 2.1 (Mar. 1998), pp. 1–36.

[Nea01] R. M. Neal. "Annealed Importance Sampling". In: *Statistics and Computing* 11.2 (Apr. 2001), pp. 125–139.

[Nil+07] M. M. Nillesen, R. G. P. Lopata, I. H. Gerrits, L. Kapusta, H. J. Huisman, J. M. Thijssen, and C. L. de Korte. "Segmentation of the Heart Muscle in 3-D Pediatric Echocardiographic Images". In: *Ultrasound in Medicine & Biology* 33.9 (Sept. 2007), pp. 1453–1462.

[NM65] J. Nelder and R. Mead. "A Simplex Method for Function Minimization Comput". In: *The Computer Journal* 7 (Jan. 1965).

[Oja82] E. Oja. "Simplified Neuron Model as a Principal Component Analyzer". In: *Journal of Mathematical Biology* 15.3 (Nov. 1982), pp. 267–273.

[OPM02] T. Ojala, M. Pietikainen, and T. Maenpaa. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (July 2002), pp. 971–987.

[Oro+12] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. "Locally Orderless Tracking". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. June 2012, pp. 1940–1947.

[ORP00] N. M. Oliver, B. Rosario, and A. P. Pentland. "A Bayesian Computer Vision System for Modeling Human Interactions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (Aug. 2000), pp. 831–843.

[Pap+18] B. W. Papież, J. M. Franklin, M. P. Heinrich, F. V. Gleeson, M. Brady, and J. A. Schnabel. "GIFTed Demons: Deformable Image Registration with Local Structure-Preserving Regularization Using Supervoxels for Liver Applications". In: *Journal of Medical Imaging* 5.2 (Apr. 2018).

[Paq+06] D. Paquin, D. Levy, E. Schreibmann, and L. Xing. "Multiscale Image Registration". In: *Mathematical biosciences and engineering: MBE* 3.2 (Apr. 2006), pp. 389–418.

[PDD00] V. Philomin, R. Duraiswami, and L. S. Davis. "Quasi-Random Sampling for Condensation". In: *Proceedings of the 6th European Conference on Computer Vision-Part II*. ECCV '00. London, UK, UK: Springer-Verlag, 2000, pp. 134–149.

[Pea96] K. Pearson. "Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 187 (Jan. 1896), pp. 253–318.

[PJB09] C. Perret-Guillaume, L. Joly, and A. Benetos. "Heart Rate as a Risk Factor for Cardiovascular Disease". In: *Progress in Cardiovascular Diseases* 52.1 (2009), pp. 6–10.

[PKB13] N. Perveen, D. Kumar, and I. Bhardwaj. "An Overview on Template Matching Methodologies and Its Applications". In: *IJRCCT* 2.10 (Oct. 2013), pp. 988–995.

[Pla+05] C. Plathow, H. Zimmermann, C. Fink, R. Umathum, M. Schöbinger, P. Huber, I. Zuna, J. Debus, W. Schlegel, H.-P. Meinzer, W. Semmler, H.-U. Kauczor, and M. Bock. "Influence of Different Breathing Maneuvers on Internal and External Organ Motion: Use of Fiducial Markers in Dynamic MRI". In: *International Journal of Radiation Oncology, Biology, Physics* 62.1 (May 2005), pp. 238–245.

[PMI09] S. Padma, J. B. Martinie, and D. A. Iannitti. "Liver Tumor Ablation: Percutaneous and Open Approaches". In: *Journal of Surgical Oncology* 100.8 (Dec. 2009), pp. 619–634.

[Pow64] M. J. D. Powell. "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives". In: *The Computer Journal* 7.2 (Jan. 1964), pp. 155–162.

[Pre+14] F. Preiswerk, V. De Luca, P. Arnold, Z. Celicanin, L. Petrusca, C. Tanner, O. Bieri, R. Salomir, and P. C. Cattin. "Model-Guided Respiratory Organ Motion Prediction of the Liver from 2D Ultrasound". In: *Medical Image Analysis* 18.5 (July 2014), pp. 740–751.

[Pri57] R. C. Prim. "Shortest Connection Networks and Some Generalizations". In: *The Bell System Technical Journal* 36.6 (Nov. 1957), pp. 1389–1401.

[PVS05] K. Panta, B. Vo, and S. Singh. "Improved Probability Hypothesis Density (PHD) Filter for Multitarget Tracking". In: *2005 3rd International Conference on Intelligent Sensing and Information Processing*. Dec. 2005, pp. 213–218.

[PVS07] K. Panta, B. n Vo, and S. Singh. "Novel Data Association Schemes for the Probability Hypothesis Density Filter". In: *IEEE Transactions on Aerospace and Electronic Systems* 43.2 (Apr. 2007), pp. 556–570.

[RAD03] G. K. Rohde, A. Aldroubi, and B. M. Dawant. "The Adaptive Bases Algorithm for Intensity-Based Nonrigid Image Registration". In: *IEEE transactions on medical imaging* 22.11 (Nov. 2003), pp. 1470–1479.

[Rei79] D. Reid. "An Algorithm for Tracking Multiple Targets". In: *IEEE Transactions on Automatic Control* 24.6 (Dec. 1979), pp. 843–854.

[Rie+04] V. Rieke, K. K. Vigen, G. Sommer, B. L. Daniel, J. M. Pauly, and K. Butts. "Referenceless PRF Shift Thermometry". In: *Magnetic Resonance in Medicine* 51.6 (June 2004), pp. 1223–1231.

[Rie+05] E. Rietzel, G. T. Y. Chen, N. C. Choi, and C. G. Willet. "Four-Dimensional Image-Based Treatment Planning: Target Volume Segmentation and Dose Calculation in the Presence of Respiratory Motion". In: *International Journal of Radiation Oncology, Biology, Physics* 61.5 (Apr. 2005), pp. 1535–1550.

[Roh+01] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. H. Kuhn. "Landmark-Based Elastic Registration Using Approximating Thin-Plate Splines". In: *IEEE Transactions on Medical Imaging* 20.6 (June 2001), pp. 526–534.

[Ros+00] K. E. Rosenzweig, J. Hanley, D. Mah, G. Mageras, M. Hunt, S. Toner, C. Burman, C. C. Ling, B. Mychalczak, Z. Fuks, and S. A. Leibel. "The Deep Inspiration Breath-Hold Technique in the Treatment of Inoperable Non-Small-Cell Lung Cancer". In: *International Journal of Radiation Oncology, Biology, Physics* 48.1 (Aug. 2000), pp. 81–87.

[Rot+14] S. Rothlübbers, J. Schwaab, J. Jenne, and M. Günther. "MICCAI CLUST 2014 - Bayesian Real-Time Liver Feature Ultrasound Tracking". In: *Challenge on Liver Ultrasound Tracking CLUST 2014* (2014), pp. 45–52.

[Rua+08] D. Ruan, J. A. Fessler, J. M. Balter, R. I. Berbeco, S. Nishioka, and H. Shirato. "Inference of Hysteretic Respiratory Tumor Motion from External Surrogates: A State Augmentation Approach". In: *Physics in Medicine and Biology* 53.11 (2008), p. 2923.

[Rub88] D. Rubin. "Using the SIR Algorithm to Simulate Posterior Distributions". In: *Bayesian Statistics 3*. Ed. by M. Bernardo, K. DeGroot, D. Lindley, and A. Smith. Oxford University Press, 1988.

[Rud16] S. Ruder. "An Overview of Gradient Descent Optimization Algorithms". In: *arXiv:1609.04747* (Sept. 2016). arXiv: `1609.04747`.

[RVG04] M. M. Romera, M. A. S. Vázquez, and J. C. G. Garcia. "Tracking Multiple and Dynamic Objects with an Extended Particle Filter and an Adapted K-Means Clustering Algorithm". In: *IFAC Proceedings Volumes*. IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004 37.8 (July 2004), pp. 310–315.

[RWS95] C. Rago, P. Willett, and R. Streit. "A Comparison of the JPDAF and PMHT Tracking Algorithms". In: *, 1995 International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95*. Vol. 5. May 1995, 3571–3574 vol.5.

[Saf+09] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. "On-Line Random Forests". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. Sept. 2009, pp. 1393–1400.

[Sai+15] S. Saini, N. Zakaria, D. R. A. Rambli, and S. Sulaiman. "Markerless Human Motion Tracking Using Hierarchical Multi-Swarm Cooperative Particle Swarm Optimization". In: *PLoS ONE* 10.5 (May 2015).

[San+12] J. Santamaria, S. Damas, J. Garcia-Torres, and O. Cordon. "Self-Adaptive Evolutionary Image Registration Using Differential Evolution and Artificial Immune Systems". In: *Pattern Recognition Letters* 33 (Dec. 2012), pp. 2065–2070.

[SBF00] H. Sidenbladh, M. J. Black, and D. J. Fleet. "Stochastic Tracking of 3D Human Figures Using 2D Image Motion". In: *Computer Vision — ECCV 2000*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, June 2000, pp. 702–718.

[Sch+01] J. A. Schnabel, D. Rueckert, M. Quist, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit, F. A. Gerritsen, D. L. G. Hill, and D. J. Hawkes. "A Generic Framework for Non-Rigid Registration Based on Non-Uniform Multi-Level Free-Form Deformations". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Oct. 2001, pp. 573–581.

[Sch+12] A. Schmidt-Richberg, R. Werner, H. Handels, and J. Ehrhardt. "Estimation of Slipping Organ Motion by Registration with Direction-Dependent Regularization". In: *Medical Image Analysis* 16.1 (Jan. 2012), pp. 150–159.

[Sch+17] M. Schwenke, J. Strehlow, D. Demedts, S. Haase, D. Barrios Romero, S. Rothluebbers, C. v. Dresky, S. Zidowitz, J. Georgii, S. Mihcin, M. Bezzi, C. Tanner, G. Sat, Y. Levy, J. Jenne, M. Günther, A. Melzer, and T. Preusser. "A Focused Ultrasound Treatment System for Moving Targets (Part I): Generic System Design and in-Silico First-Stage Evaluation". In: *Journal of Therapeutic Ultrasound* 5 (Dec. 2017).

[SDP13] A. Sotiras, C. Davatzikos, and N. Paragios. "Deformable Medical Image Registration: A Survey". In: *IEEE Transactions on Medical Imaging* 32.7 (July 2013), pp. 1153–1190.

[SG99] C. Stauffer and W. E. L. Grimson. "Adaptive Background Mixture Models for Real-Time Tracking". In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Vol. 2. 1999, 252 Vol. 2.

[Sha70] D. F. Shanno. "Conditioning of Quasi-Newton Methods for Function Minimization". In: *Mathematics of Computation* 24.111 (1970), pp. 647–656.

[Sha92] C. J. Shatz. "The Developing Brain". In: *Scientific American* 267 (Oct. 1992), pp. 60–7.

[Sme+14] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. "Visual Tracking: An Experimental Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (July 2014), pp. 1442–1468.

[Smi+12] R. Smith, A. Rahni, J. Jones, and K. Wells. "Recursive Bayesian Estimation for Respiratory Motion Correction in Nuclear Medicine Imaging". In: *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. Oct. 2012, pp. 2942–2945.

[SMW02] J. N. Sanchez, A. Milstein, and E. Williamson. "Robust Global Localization Using Clustered Particle Filtering". In: *arXiv:cs/0204044* (Apr. 2002). arXiv: cs/0204044.

[Sol+07] O. I. I. Soliman, B. J. Krenning, M. L. Geleijnse, A. Nemes, J. G. Bosch, R.-J. van Geuns, S. W. Kirschbaum, A. M. Anwar, T. W. Galema, W. B. Vletter, and F. J. ten Cate. "Quantification of Left Ventricular Volumes and Function in Patients with Cardiomyopathies by Real-Time Three-Dimensional Echocardiography: A Head-to-Head Comparison between Two Different Semiautomated Endocardial Border Detection Algorithms". In: *Journal of the American Society of Echocardiography: Official Publication of the American Society of Echocardiography* 20.9 (Sept. 2007), pp. 1042–1049.

[Spe87] C. Spearman. "The Proof and Measurement of Association between Two Things. By C. Spearman, 1904". In: *The American Journal of Psychology* 100.3-4 (1987), pp. 441–471.

[SSA04] A. Schweikard, H. Shiomi, and J. Adler. "Respiration Tracking in Radiosurgery". In: *Medical Physics* 31.10 (Oct. 2004), pp. 2738–2741.

[SSA05] A. Schweikard, H. Shiomi, and J. Adler. "Respiration Tracking in Radiosurgery without Fiducials". In: *The international journal of medical robotics + computer assisted surgery: MRCAS* 1.2 (Jan. 2005), pp. 19–27.

[ST01] C. Sminchisescu and B. Triggs. "Covariance Scaled Sampling for Monocular 3D Body Tracking". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* Vol. 1. 2001, pp. 447–454.

[ST03] C. Sminchisescu and B. Triggs. "Estimating Articulated Human Motion With Covariance Scaled Sampling". In: *International Journal of Robotics Research* 22 (Jan. 2003).

[Sta+04] R. J. Stafford, R. E. Price, C. J. Diederich, M. Kangasniemi, L. E. Olsson, and J. D. Hazle. "Interleaved Echo-Planar Imaging for Fast Multiplanar Magnetic Resonance Temperature Imaging of Ultrasound Thermal Ablation Therapy". In: *Journal of magnetic resonance imaging: JMRI* 20.4 (Oct. 2004), pp. 706–714.

[Ste+06] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. "Model-Based Hand Tracking Using a Hierarchical Bayesian Filter". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.9 (Sept. 2006), pp. 1372–1384.

[STM91] M. K. Stehling, R. Turner, and P. Mansfield. "Echo-Planar Imaging: Magnetic Resonance Imaging in a Fraction of a Second". In: *Science (New York, N.Y.)* 254.5028 (Oct. 1991), pp. 43–50.

[Sto+11] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt. "Fast Articulated Motion Tracking Using a Sums of Gaussians Body Model". In: *Proceedings of the IEEE International Conference on Computer Vision*. Nov. 2011, pp. 951–958.

[Str+15] J. Strehlow, X. Xiao, M. Domschke, M. Schwenke, I. Karakitsios, S. Mihcin, J. Schwaab, Y. Levy, T. Preusser, and A. Melzer. "US-Tracked Steered FUS in a Respiratory Ex Vivo Ovine Liver Phantom". In: *Current Directions in Biomedical Engineering* 1.1 (2015), pp. 294–297.

[Suh+05] M. Suhling, M. Arigovindan, C. Jansen, P. Hunziker, and M. Unser. "Myocardial Motion Analysis from B-Mode Echocardiograms". In: *IEEE Transactions on Image Processing* 14.4 (Apr. 2005), pp. 525–536.

[Sul+99] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. "Object Localization by Bayesian Correlation". In: *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1068–.

[Sun+17] K. M. Sundaram, S. S. Chang, D. F. Penson, and S. Arora. "Therapeutic Ultrasound and Prostate Cancer". In: *Seminars in Interventional Radiology* 34.2 (June 2017), pp. 187–200.

[Tan+16] C. Tanner, Y. Zur, K. French, G. Samei, J. Strehlow, G. Sat, H. McLeod, G. Houston, S. Kozerke, G. Székely, A. Melzer, and T. Preusser. "In Vivo Validation of Spatio-Temporal Liver Motion Prediction from Motion Tracked on MR Thermometry Images". In: *International Journal of Computer Assisted Radiology and Surgery* 11.6 (June 2016), pp. 1143–1152.

[TBS12] P. Torkashvand, H. Behnam, and Z. A. Sani. "Modified Optical Flow Technique for Cardiac Motions Analysis in Echocardiography Images". In: *Journal of Medical Signals and Sensors* 2.3 (2012), pp. 121–127.

[TSB17] K. Thormann, F. Sigges, and M. Baum. "Learning an Object Tracker with a Random Forest and Simulated Measurements". In: *2017 20th International Conference on Information Fusion (Fusion)*. July 2017, pp. 1–4.

[Uch+11] T. Uchida, S. Shoji, M. Nakano, S. Hongo, M. Nitta, Y. Usui, and Y. Nagata. "High-Intensity Focused Ultrasound as Salvage Therapy for Patients with Recurrent Prostate Cancer after External Beam Radiation, Brachytherapy or Proton Therapy". In: *BJU international* 107.3 (Feb. 2011), pp. 378–382.

[Ved+03] S. S. Vedam, V. R. Kini, P. J. Keall, V. Ramakrishnan, H. Mostafavi, and R. Mohan. "Quantifying the Predictability of Diaphragm Motion during Respiration with a Noninvasive External Marker". In: *Medical Physics* 30.4 (Apr. 2003), pp. 505–513.

[Ven+83] A. Venot, J. F. Lebruchec, J. L. Golmard, and J. C. Roucayrol. "An Automated Method for the Normalization of Scintigraphic Images". In: *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine* 24.6 (June 1983), pp. 529–531.

[Ven+88] A. Venot, J. Y. Devaux, M. Herbin, J. F. Lebruchec, L. Dubertret, Y. Raulo, and J. C. Roucayrol. "An Automated System for the Registration and Comparison of Photographic Images in Medicine". In: *IEEE Transactions on Medical Imaging* 7.4 (Dec. 1988), pp. 298–303.

[Vie+16] M. A. Viergever, J. B. A. Maintz, S. Klein, K. Murphy, M. Staring, and J. P. W. Pluim. "A Survey of Medical Image Registration - under Review". In: *Medical Image Analysis* 33 (Oct. 2016), pp. 140–144.

[VL84] A. Venot and V. Leclerc. "Automated Correction of Patient Motion and Gray Values Prior to Subtraction in Digitized Angiography". In: *IEEE transactions on medical imaging* 3.4 (1984), pp. 179–186.

[VMR13] M. Vivarelli, R. Montalti, and A. Risaliti. "Multimodal Treatment of Hepatocellular Carcinoma on Cirrhosis: An Update". In: *World Journal of Gastroenterology* 19.42 (Nov. 2013), pp. 7316–7326.

[Vos+17] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum. "End-to-End Unsupervised Deformable Image Registration with a Convolutional Neural Network". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Lecture Notes in Computer Science. Springer, Cham, Sept. 2017, pp. 204–212.

[Wad54] O. L. Wade. "Movements of the Thoracic Cage and Diaphragm in Respiration". In: *The Journal of Physiology* 124.2 (May 1954), pp. 193–212.

[Wan+16] J. Wang, N. Yu, F. Zhu, and L. Zhuang. "Multi-Level Visual Tracking with Hierarchical Tree Structural Constraint". In: *Neurocomputing* 202 (Aug. 2016), pp. 1–11.

[Wan+17] W. Wang, C. Wang, S. Liu, T. Zhang, and X. Cao. "Robust Target Tracking by Online Random Forests and Superpixels". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2017), pp. 1–1.

[Wee+96] H. F. Weehuizen, A. H. Botha, J. L. Conradie, P. M. Cronje, R. K. Fisch, D. T. Fourie, H. N. Jungwirth, H. Kettner, L. M. Roels, and P. J. Theron. "Beam Control for Proton Therapy". In: (Aug. 1996), E41.

[Wei+03] C. Weidensteiner, B. Quesson, B. Caire-Gana, N. Kerioui, A. Rullier, H. Trillaud, and C. T. W. Moonen. "Real-Time MR Temperature Mapping of Rabbit Liver in Vivo during Thermal Ablation". In: *Magnetic Resonance in Medicine* 50.2 (Aug. 2003), pp. 322–330.

[WH81] P. N. T. Wells and M. Halliwell. "Speckle in Ultrasonic Imaging". In: *Ultrasonics* 19.5 (Sept. 1981), pp. 225–229.

[WLH01] Y. Wu, J. Y. Lin, and T. S. Huang. "Capturing Natural Hand Articulation". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* Vol. 2. 2001, 426–432 vol.2.

[Won+99] J. W. Wong, M. B. Sharpe, D. A. Jaffray, V. R. Kini, J. M. Robertson, J. S. Stromberg, and A. A. Martinez. "The Use of Active Breathing Control (ABC) to Reduce Margin for Breathing Motion". In: *International Journal of Radiation Oncology, Biology, Physics* 44.4 (July 1999), pp. 911–919.

[XCX11] Y. Xie, M. Chao, and G. Xiong. "Deformable Image Registration of Liver with Consideration of Lung Sliding Motion". In: *Medical Physics* 38.10 (Oct. 2011), pp. 5351–5361.

[XSL13] J. Xiao, R. Stolkin, and A. Leonardis. "An Enhanced Adaptive Coupled-Layer LGTracker++". In: *2013 IEEE International Conference on Computer Vision Workshops*. Dec. 2013, pp. 137–144.

[YA05] T. Yang and V. Aitken. "Uniform Clustered Particle Filtering for Robot Localization". In: *Proceedings of the 2005, American Control Conference, 2005.* June 2005, 4607–4612 vol. 7.

[Yan+17] X. Yang, R. Kwitt, M. Styner, and M. Niethammer. "Quicksilver: Fast Predictive Image Registration – A Deep Learning Approach". In: *NeuroImage* 158 (Sept. 2017), pp. 378–396.

[YJ14] S. Yip and R. Jeraj. "Use of Articulated Registration for Response Assessment of Individual Metastatic Bone Lesions". In: *Physics in Medicine & Biology* 59.6 (2014), p. 1501.

[YM18] S. Y Cheng and M. M Trivedi. "Articulated Human Body Pose Inference from Voxel Data Using a Kinematically Constrained Gaussian Mixture Model". In: (Apr. 2018).

[Yos+09] S. Yoshizawa, T. Ikeda, A. Ito, R. Ota, S. Takagi, and Y. Matsumoto. "High Intensity Focused Ultrasound Lithotripsy with Cavitating Microbubbles". In: *Medical & Biological Engineering & Computing* 47.8 (Aug. 2009), pp. 851–860.

[YPJ14] S. Yip, T. Perk, and R. Jeraj. "Development and Evaluation of an Articulated Registration Algorithm for Human Skeleton Registration". In: *Physics in Medicine and Biology* 59.6 (Mar. 2014), pp. 1485–1499.

[Yu+13] Y. Yu, S. Zhang, J. Huang, D. Metaxas, and L. Axel. "Sparse Deformable Models with Application to Cardiac Motion Analysis". In: *Information Processing in Medical Imaging: Proceedings of the ... Conference* 23 (2013), pp. 208–219.

[Zac+14] C. Zachiu, D. de Senneville, S. Crijns, B. Raaymakers, C. Moonen, M. Ries, M. Ries, and M. Ries. *A Framework for Slow Physiological Motion Compensation During HIFU Interventions in the Liver: Proof of Concept.* 2014.

[Zac+15] C. Zachiu, N. Papadakis, M. Ries, C. Moonen, and B. D. de Senneville. "An Improved Optical Flow Tracking Technique for Real-Time MR-Guided Beam Therapies in Moving Organs". In: *Physics in Medicine and Biology* 60.23 (2015), p. 9003.

[Zha+06] M. Zhang, V. Moiseenko, M. Liu, and T. Craig. "Internal Fiducial Markers Can Assist Dose Escalation in Treatment of Prostate Cancer: Result of Organ Motion Simulations". In: *Physics in Medicine and Biology* 51.2 (Jan. 2006), pp. 269–285.

[Zha+09a] L. Zhang, H. Zhu, C. Jin, K. Zhou, K. Li, H. Su, W. Chen, J. Bai, and Z. Wang. "High-Intensity Focused Ultrasound (HIFU): Effective and Safe Therapy for Hepatocellular Carcinoma Adjacent to Major Hepatic Veins". In: *European Radiology* 19.2 (Feb. 2009), pp. 437–445.

[Zha+09b] T. Zhao, W. Lu, D. Yang, S. Mutic, C. E. Noel, P. J. Parikh, J. D. Bradley, and D. A. Low. "Characterization of Free Breathing Patterns with 5D Lung Motion Model". In: *Medical Physics* 36.11 (Nov. 2009), pp. 5183–5189.

[Zho+09] X. Zhong, C. H. Meyer, D. J. Schlesinger, J. P. Sheehan, F. H. Epstein, J. M. Larner, S. H. Benedict, P. W. Read, K. Sheng, and J. Cai. "Tracking Brain Motion during the Cardiac Cycle Using Spiral Cine-DENSE MRI". In: *Medical Physics* 36.8 (Aug. 2009), pp. 3413–3419.

[Zin+12] C. Zini, E. Hipp, S. Thomas, A. Napoli, C. Catalano, and A. Oto. "Ultrasound- and MR-Guided Focused Ultrasound Surgery for Prostate Cancer". In: *World Journal of Radiology* 4.6 (June 2012), pp. 247–252.

[ZM00] S. Zhu and K. K. Ma. "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation". In: *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society* 9.2 (2000), pp. 287–290.

Acknowledgments

This work has emerged around the question of how to automatically track breathing motion in real-time medical image data. The full topic of this work has evolved in the context of ongoing research questions. As such, it has grown with each furthering matter and found support in different projects and institutions. A number of people are in some way related to it, which I would gratefully like to acknowledge here.

First of all I thank my supervisor Prof. Dr. Matthias Günther for his support in the ongoing investigation. Our regular exchange of ideas has been very inspiring and I am very grateful for the freedom I was given working on the topic.

I thank Prof. Dr. Gabriel Zachmann for taking the time and reading the thesis as secondary assessor.

Moreover, I thank the further examiners Prof. Dr. Vasily Ploshikhin and Prof. Dr. Stefan Bornholdt.

For making this work possible in the first place, I'd like to express my gratitude to both the mediri GmbH (Heidelberg, Germany) and Fraunhofer MEVIS (Bremen, Germany).

I'd like to thank Jürgen Jenne for sharing detailed insight into any ultrasound related question and Dr. Markus Wenzel for his support as secondary consultant.

I'm much obliged to Alina, David, Jürgen, Lena, Markus, Richard and Simon for proof reading sections and providing helpful hints for writing.

Much of the motivation for this work stems from my recent projects. First, I'd like to thank the entire Trans-Fusimo team. The colleagues at Fraunhofer MEVIS, Diego Barrios-Romero, Dr. Caroline von Dresky, Sabrina Haase, Prof. Dr. Tobias Preusser, Dr. Michael Schwenke and Jan Strehlow are to be thanked in particular. It has been a very pleasant and inspiring working atmosphere in this ambitious project. From ETH Zürich, I thank Prof. Dr. Christine Tanner for very helpful discussion and feedback on volunteer data of breathing motion. I'd like to thank Senay Mihcin for joint experiments in test data acquisition.

In the same spirit, I thank the teams of PTPS at all contributing sites alike. I thank Gilad Halevy, Shlomi Rudich, Dr. Yeruham Shapira and Itay Yelin from Insightec (Haifa, Israel) for providing sample data of prostate FUS treatment and valuable elucidation on the interpretation of it. I thank Dr. Johannes Gregorii, Alba Malavé, Dr. Jan Moltz, Dr. Christian Rieder and Lennart Tautz for the great cooperation in developing the

demonstrator application.

Moreover, for casual discussion from the early days of this work, I'd like to thank Dr. Joachim Georgii and Johannes Lotz.

Being my site of work, Fraunhofer MEVIS is to be thanked for the pleasant and open working environment. Among the, in any way very agreeable, staff I'd like to give special credit to the MRI enthusiasts Ben, Cristoffer, Daniel, David, Ina, Jochen, Johanna, Klaus, Marco, Nora, Saulius and Simon. I'd also like to thank students Alina, Lena, Jörn and Vincent for interesting discussions and support in ongoing projects.

Furthermore, I'd like to acknowledge the valuable discussions with my changing pools of direct bureau colleagues, lunch partakers and coffee appreciators. In particular I'd like to thank Andre, Felix, Hendrik, Henning, Matthias and Richard for absolutely crucial on and off topic discussions.

Last, but not least I would like to thank my family and specifically thank my parents and sister and Alina for their enduring encouragement throughout this work. Thank you for your support in the arguably most intense final phase. This work would not have been possible without you.

Funding

- The research leading to results of this work has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreements no. 270186 and no. 611889 http://www.trans-fusimo.eu.
- The work was supported by the German Ministry for Economic Affairs and Energy. Gefördert durch: Bundesministerium für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestages