Object localisation, dimensions estimation and tracking

by

Ricardo Sánchez Matilla

Bachelor in Telecommunication Engineering

Master in Telecommunication Engineering

A dissertation submitted to
The School of Electronic Engineering and Computer Science
in partial fulfilment of the requirements of the
Degree of Doctor of Philosophy
in the subject of
Electronic Engineering

Queen Mary University of London
Mile End Road
E1 4NS, London, UK

November, 2019

Declaration

I, Ricardo Sánchez Matilla, confirm that the research included in this thesis is my own work, that is duly acknowledged, and my contributions are indicated. I have also acknowledged previously published materials.

I attest that reasonable care has been exercised to ensure the originality of this work, and, to the best of my knowledge, does not break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the college has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree to any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Ricardo Sánchez Matilla

Date: 29^{th} November 2019

Supervisor

Prof. Andrea Cavallaro

Author

Ricardo Sánchez Matilla

Object localisation, dimensions estimation and tracking

Abstract

Localising, estimating the physical properties of, and tracking objects from audio and video signals are the base for a large variety of applications such as surveillance, search and rescue, extraction of objects' patterns and robotic applications. These tasks are challenging due to low signal-to-noise ratio, multiple moving objects, occlusions and changes in objects' appearance. Moreover, these tasks become more challenging when real-time performance is required and when the sensor is mounted in a moving platform such as a robot, which introduces further problems due to potentially quick sensor motions and noisy observations. In this thesis, we consider algorithms for single and multiple object tracking from static microphones and cameras, and moving cameras without relying on additional sensors or making strong assumptions about the objects or the scene; and localisation and estimation of the 3D physical properties of unseen objects. We propose an online multi-object tracker that addresses noisy observations by exploiting the confidence on object observations and also addresses the challenges of object and camera motion by introducing a real-time object motion predictor that forecasts the future location of objects with uncalibrated cameras. The proposed method enables real-time tracking by avoiding computationally expensive labelling procedures such as clustering for data association. Moreover, we propose a novel multi-view algorithm for jointly localising and estimating the 3D physical properties of objects via semantic segmentation and projective geometry without the need to use additional sensors or markers. We validate the proposed methods in three standard benchmarks, two self-collected datasets, and two real robotic applications that involve an unmanned aerial vehicle and a robotic arm. Experimental results show that the proposed methods improve existing alternatives in terms of accuracy and speed.

Contents

Published work related to the thesis

Journal papers

[J1] R. Sanchez-Matilla, K. Chatzilygeroudis, A. Modas, N. Ferreira Duarte, A. Xompero, P. Frossard, A. Billard, A. Cavallaro. Benchmark for Human-to-Robot Handovers of Unseen Containers with Unknown Filling. *IEEE Robot. and Automa. Lett.*, Apr., 2020.

Conference papers

[C1] R. Sanchez-Matilla, A. Cavallaro. Motion Prediction for First-person Vision Multi-object Tracking. *IEEE Procc. of Euro. Conf. on Comp. Vis. Work.*, Glasgow, UK, Aug., 2020.

[C2] A. Xompero, R. Sanchez-Matilla, A. Modas, P. Frossard, A. Cavallaro. Multi-view shape estimation of transparent containers. *Procc. of IEEE Int. Conf. on Acoust., Spee. and Sig. Proc.*, Barcelona, Spain, May, 2020.

[C3] L. Wang, R. Sanchez-Matilla, A. Cavallaro. Audio-visual sensing from a quadcopter: dataset and baselines for source localization and sound enhancement. *Procc. of IEEE/RSJ Int. Conf. on Intell. Rob. and Sys.*, Macau, China, Nov., 2019.

[C4] R. Sanchez-Matilla, A. Cavallaro. A predictor of moving objects for first-person vision. *Procc. of IEEE Int. Conf. on Ima. Proc.*, Taipei, Taiwan, Sept., 2019.

[C5] R. Sanchez-Matilla, A. Cavallaro. Confidence intervals for tracking performance scores. *Procc. of IEEE Int. Conf. on Ima. Proc.*, Athens, Greece, Oct., 2018.

[C6] L. Wang, R. Sanchez-Matilla, A. Cavallaro. Tracking a moving sound source from a multi-rotor drone. *Procc. of IEEE/RSJ Int. Conf. on Int. Rob. and Sys.*, Madrid, Spain, Oct., 2018.

[C7] R. Sanchez-Matilla, F. Poiesi, A. Cavallaro. Online multi-target tracking with strong and weak detections. *IEEE Procc. of Euro. Conf. on Comp. Vis. Work.*, Amsterdam, The Netherlands, Oct., 2016.

Electronic version of the papers are available at https://risama.github.io/.

To everyone who always believe in me

List of abbreviations

2D	Two-dimensional
3D	Three-dimensional
ACF	Aggregate Channel Features
AVQ	Audio-Visual Quadcopter
BM	Brownian Model
CORSMAL	Collaborative Object Recognition Shared Manipulation and Learning
CPD	Combined Pedestrian Detection
DF	Detection free
DoA	Direction of arrival
DoF	Degrees of freedom
DTDPM	Discriminatively Trained Deformable Part Models
EA	Early association
EA-PHD-PF	Early Association Probability Hypothesis Density Particle Filter
EKF	Extended Kalman Filter
EM	Exponentially-weighted motion predictor
FAF	False alarm per frame
FN	False negative
FP	False positive
Frag	Fragmented trajectories
FRCNN	Fast Recurrent Convolutional Neural Network
GM	Global motion
GMG	Global motion ground
GT	Ground truth
IDS	Identity switch

IOU	Intersection over union
KF	Kalman Filter
LMP	Linear motion prediction
LoDE	Localisation and object Dimensions Estimator
LoDE-IR	Localisation and object Dimensions Estimator with infrared
LR	Linear regressor
LSTM	Long Short-Term Memory
MDF	Multiple Detector Fusion
MDP	Markov Decision Process
MF	Median Filter
MGT	Manually annotated ground truth
ML	Mostly lost trajectories
MOT	Multiple object tracking
MOTA	Multiple Object Tracking Accuracy
MOTB	Multiple Object Tracking Benchmark
MOTP	Multiple Object Tracking Precision
MT	Mostly tracked trajectories
NOCS	Normalized Object Coordinate Space
PF	Particle Filter
PHD	Probability Hypothesis Density
PHD-PF	Probability Hypothesis Density Particle Filter
PVNet	Pixel-wise Voting Network
RANSAC	Random Sample Consensus
RCNN	Recurrent Convolutional Neural Network
RGB	Red green blue
RGBD	Red green blue depth
SDP	Scale Dependant Pooling
SH	Simple homography-based object motion predictor

SNR	Signal-to-noise ratio
SOT	Single object tracking
SP	Static object prior knowledge
SRP	Steered Response Power
SRP-PHAT	Steered-Response Power Phase Transform
StoCS	Stochastic Congruent Sets
TBD	Tracking-by-detection
TF	Time-frequency
TLD	Tracking Learning Detection
TP	True positive
UAV	Unmanned aerial vehicle

List of symbols

$\alpha_{k,\lambda}$	Homography normalisation factor
β	Interpolation factor
δ	Dirac delta
Δ_b	Time between temporal blocks
$\dot{\theta}_b^i$	Audio particle i at time b
$\dot{s}_k^j$	Combined detection confidence
γ	Skipping frames parameter
$\mathbf{H}_{k\|k-1}$	Homography between frames at time $k-1$ and k
Λ_k	Set of object identities at time k
λ	Object identity
κ	Video frame rate
λ^w, λ^h	Absolute error of width/height estimation
$\mathbb{C}$	Set of circumferences
$\mathbb{N}$	Set of natural numbers between 0 and 255
$\mathbb{P}'_k, \mathbb{P}''_k, \mathbb{P}'''_k$	Set of matched/filtered/inlier keypoints at time k
$\mathbb{P}_k$	Set of keypoints at time k

$\mathbb{P}_k^*$	Set of detected keypoints at time k
$\mathbb{Q}$	Sampled 3D circumference
$\mathbb{U}$	Set of detections to be combined
$\mathbb{X}_k$	Set of estimated states at time k
$\mathbb{Z}_k$	Set of detections at time k
$\mathbb{Z}_k^+, \mathbb{Z}_k^-$	Set of strong/weak detections at time k
$\mathbf{a}$	Audio-visual geometrical alignment parameter
$\mathbf{c}, \mathbf{C}$	2D/3D object centroid estimate
$\mathbf{I}^c$	Image from camera c
$\mathbf{I}_k$	Image at time k
$\mathbf{K}, \mathbf{E}$	Intrinsic/extrinsic camera parameters
$\mathbf{m}$	Semantic segmentation map
$\mathbf{N}$	Noise matrix
$\mathbf{p}_k^n$	Keypoint n at time k
$\mathbf{Q}$	3D circumference
$\mathbf{S}_k$	Signal at time k
$\mathbf{T}$	Object motion matrix
$\mathbf{W}$	Object widths at different heights
$\mathbf{x}_{k,\lambda}$	Estimated state of the λ-th object at time k
$\mathbf{z}_k^i$	Detection i at time k
$\mathcal{B}(\cdot)$	Camera undistortion function
$\mathcal{D}(\cdot)$	Detector
$\mathcal{N}(\cdot)$	Gaussian distribution
$\mathcal{T}(\cdot)$	Tracker
$\tilde{\mathbb{X}}$	Set of ground-truth annotations for a sequence
$\tilde{\mathbf{x}}_{k,\lambda}$	Ground-truth annotation of λ-th object at time k
$\tilde{u}, \tilde{v}$	Horizontal/vertical location component of a ground-truth annotation
$\tilde{w}, \tilde{h}$	Width/height of a ground-truth annotation

$\mu_{k,\lambda}$	Squared error of the prediction for λ-th object at time k
L_k	Number of particles at time k
$\omega^i_{k,\lambda}$	Association cost between i-th detection and λ-th object at time k
$\dot{\mathbf{x}}^i_{k,\lambda}$	Particle i of λ-th object at time k
$\dot{\mathbb{X}}_{k,\lambda}$	Set of particles with identity λ
$\psi(\cdot)$	Projection function
π^i_k	Weight of the i-th particle
ρ	Number of particles per object
$\rho_b(\cdot)$	Spatial confidence function of time block b
σ	Standard deviation
$\Sigma_{k,\lambda}$	Diagonal of standard deviations
τ_s	Strong/weak confidence threshold
$\tau(\cdot)$	Triangulation function
θ_b	Direction of arrival detection at time block b
T_P, T_F	Number of past/future time steps to observe/predict
Δ_k	Time between time steps
$\theta, \bar{\theta}$	Direction of arrival and angular velocity of a detection/estimation
θ_a, θ_v	Direction of arrival estimated via audio/video
ξ	Camera distortion parameters
u, v	Horizontal and vertical location component of a detection/state
$\bar{u}, \bar{v}$	Horizontal and vertical velocity component of a detection/state
w, h	Width and height of a detection/state
c	Camera index
F	Camera focal length
$f_p(\cdot), f_o(\cdot)$	Generic prediction/observation model function
G	Prediction matrix
J_k	Number of newborn particles for a new trajectory
K	Number of time steps in a signal

k	Time
K_λ	Number of time steps where λ-th object exist
L	Number of sampled circumferences
N	Number of samples per circumference
O_C, O_M	Camera/microphone array origin
P	Precision
$p_k(\cdot)$	Newborn importance function
R	Recall
r	Circumference radius
s_k^i	Confidence of detection i at time k
T	Transpose operator
W, H, C	Image width, height and number of channels
z_l	Height of l-th circumference

Chapter 1

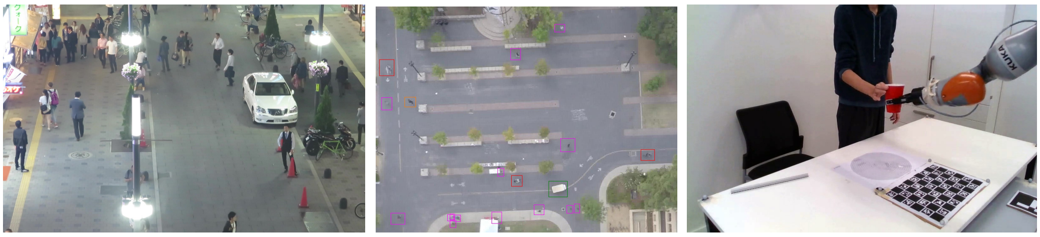
Introduction

In this chapter, we first show the motivation for the work done in this thesis (Sec. 1.1); then, we introduce the problem formulation (Sec. 1.2); then, we show the contributions of this thesis (Sec. 1.3); and, finally, we present the organisation of the rest of the thesis (Sec. 1.4).

1.1 Motivation

The availability of high quality and inexpensive sensors such as microphones and video cameras, and the increasing need for analysis and understanding the behaviour of objects (e.g. people) have generated a great interest in the artificial intelligence field in the last decades. Localisation and tracking of objects form the basis of a large variety of applications including surveillance (Fig. 1.1a), extraction of patterns from the objects' behaviour (Fig. 1.1b) and robotic applications (Fig. 1.1c).

Vision-based multiple object tracking (MOT) has been studied in-depth during the last decades [15, 25, 149, 217] and has achieved impressive results since object detectors drastically improved their performance in the deep-learning era [66, 154, 211]. However, when the camera is installed on a moving platform, such as a ground robot or a flying unmanned aerial vehicle, the tracking performance deteriorates. This is due to additional challenges such as noisy observations, specially produced by motion blur due to camera motion, multiple moving objects with unknown motion models, pronounced object occlusions when a ground robot navigates in a crowded scene and similar object appearance among objects when, for instance, the illumination is poor or objects look similar (e.g. a group of bees). To handle these challenges, existing works often



(a) (b) (c)

Figure 1.1: Examples of video-based applications that benefit from object tracking: (a) surveillance, (b) automatic pedestrian flow monitoring and (c) robotic applications. Figure (a) extracted from [125]. Figure (b) extracted from [158].

over-simplify the problem by relying on strong assumptions such as the full knowledge of the objects (e.g. known object appearance [141]) or of the scenes (e.g. planarity of the scene [120]), or requiring additional sensors (e.g. depth cameras [39] or electromagnetic measurements [9]).

Audio-based single object tracking (SOT) can be of great help when vision features are deteriorated (e.g. in a poorly illuminated scene). When the sensor is composed of multiple microphones (i.e. microphone array), existing algorithms, based on strong mathematical formulations, can produce accurate results [21, 178]. However, when these algorithms are required to work in noisy environments (e.g. in a flying unmanned aerial vehicle - UAV), the performance is significantly degraded. When the target sound source is a person talking out loud, the signal-to-noise ratio (SNR) received by a microphone array installed in a UAV is extremely low ($<$ -15 dB) [C3]. To handle these challenges, existing works often rely on complex calibration procedures (e.g. microphone array calibration [77]) or use prior information (e.g. noise correlation matrix [76]).

Over the past decade, important progress has been made towards introducing robots in human-inhabited environments, such as factories and households. In this context, the capability of robots to exchange a wide range of objects with humans is particularly important. For supporting such robotic manipulation tasks, localising and estimating the physical properties (e.g. dimensions) of unseen objects (e.g. a drinking cup) in 3D is required. In this direction, several vision-based algorithms achieved very accurate results. However, most solutions require either a full knowledge of the 3D objects physics [41, 46, 141, 185, 205] or markers on the object and a motion capture system [122] to work. Therefore, when objects are unseen (i.e. no prior knowledge of the object is available) and markers are undesirable, these methods are inaccurate or fail to work.

In this thesis, we tackle the above challenges and introduce

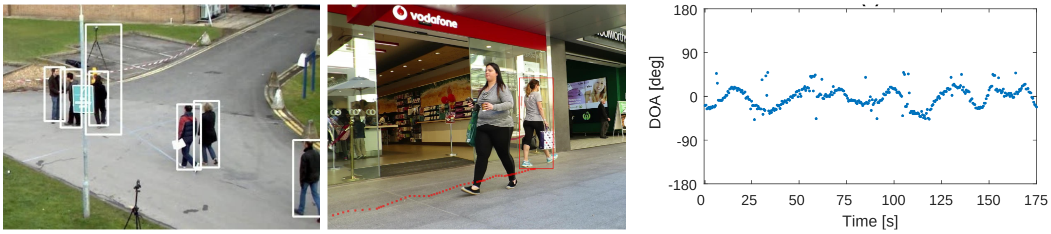
- an online multiple object tracker that relies on an accurate object motion prediction that is aware of the camera motion and does not use object appearance information for tracking multiple objects with a moving uncalibrated camera, without assumptions on the objects or the scene [C1, C4, C7];

- an automatic method for quantifying the uncertainty in the ground-truth annotations of any tracking dataset, and for generating confidence interval for a given evaluation score by only using existing annotations [C5];
- an audio-visual framework that combines audio processing techniques and online tracking to enable, for the first time, to track a moving sound source from a UAV in an outdoor scenario [C3, C6];
- a vision-based multi-view algorithm that is able to localise and estimate the dimensions in 3D of an unseen object which is then combined with a robotic arm for performing real-world dynamic human-to-robot handovers without the need for either a full 3D object model or markers on the object [C2, J1].

1.2 Problem formulation

Let's have a sensor (i.e. an RGB camera or a microphone array) that perceives the environment and encodes the information into a signal $\mathbb{S} = \{\mathbf{S}_k\}_{k=1}^{K}$, at each time step k, where K is the total number of time steps. For each k, let a detector, $\mathcal{D}$, estimate the location of the objects of interest in the environment (e.g. people) as $\mathbb{Z}_k = \mathcal{D}(\mathbf{S}_k)$, where $\mathbb{Z}_k = \{\mathbf{z}_k^j\}_{j=1}^{|\mathbb{Z}_k|}$ are the observations (or detections) and j is the observation index. The detector can miss to localise existing objects (false negatives), generate inaccurate observations or observations that refer to objects that do not exist in the environment (false positives). For each detection $\mathbf{z}_k^j$, the detector provides a confidence score s_k^j. For each k, let a tracker, $\mathcal{T}$, estimate the identity of each of the objects and refine their location (tracking state) as $\mathbb{X}_k = \mathcal{T}(\mathbb{X}_{1:k-1}, \mathbb{Z}_k, \mathbf{S}_k)$, where $\mathbb{X}_{1:k-1}$ are the previous tracking states, and $\mathbb{X}_k = \{\mathbf{x}_{k,\lambda} : \forall \lambda \in \Lambda_k\}$ are the current tracking states with $\Lambda_k = \{\lambda_k^i\}_{i=1}^{|\Lambda_k|}$ the set of current tracking identities. In this thesis, detections and tracking states are encoded as: *bounding boxes* or *points*, when the sensor is a camera; and as *direction of arrival* of a target sound source, when the sensor is a microphone array (Fig. 1.2).

When the sensor is an *RGB camera*, the signal at each time k is an image (frame), $\mathbf{S}_k = \mathbf{I}_k \in \mathbb{N}^{W \times H \times C}$ of width W, height H and $C = 3$ channels, with $\mathbb{N}$ being the set of natural numbers between 0 and 255. For each time k and each j-th detected object, we define a bounding box



(a) (b) (c)

Figure 1.2: Sample of the three types of observations considered in this thesis. When the sensor is a camera: (a) bounding box and (b) point; and, when the sensor is a microphone array, (c) direction of arrival (DOA). Note that the bounding box in (b) is just shown for visualising that the observed points indicate the location in between a person's feet. Images (a) and (b) from [125].

observation as

$$\mathbf{z}_k^j = \left(u_k^j, v_k^j, w_k^j, h_k^j\right)^T, \tag{1.1}$$

where (u_k^j, v_k^j) are the image coordinates (e.g. object centre), and (w_k^j, h_k^j) are the width and height on the image plane and T is the transpose operator. For each time k, we define the tracking state with identity λ as

$$\mathbf{x}_{k,\lambda} = \left(u_{k,\lambda}, \bar{u}_{k,\lambda}, v_{k,\lambda}, \bar{v}_{k,\lambda}, w_{k,\lambda}, h_{k,\lambda}\right)^T, \tag{1.2}$$

where $(u_{k,\lambda}, v_{k,\lambda})$ and $(\bar{u}_{k,\lambda}, \bar{v}_{k,\lambda})$ are the horizontal and vertical components of the image location and velocity, respectively; and $(w_{k,\lambda}, h_{k,\lambda})$ are width and height of the state. When detections and tracking states are points, the definitions are analogous but we discard the dimension components.

When the sensor is a *microphone array*, the signal at each time is a voltage defined as $\mathbf{S}_k = \mathbf{A}_k \in \mathbb{R}^M$, with M the number of microphones within the array and $\mathbb{R}$ the set of real numbers. For each time k and each j-th detected object, we define an observation as

$$\mathbf{z}_k^j = (\theta_k^j), \tag{1.3}$$

where θ_k^j is the direction of arrival of the target sound source. For each time k, we define the tracking state with identity λ as

$$\mathbf{x}_{k,\lambda} = (\theta_{k,\lambda}, \bar{\theta}_{k,\lambda})^T, \tag{1.4}$$

where $\theta_{k,\lambda}$ and $\bar{\theta}_{k,\lambda}$ are the estimated direction of arrival and the angular velocity, respectively.

1.3 Contributions

The main contributions of the thesis are the following:

1. An online multi-object tracker, based on the Probability Hypothesis Density Particle Filter framework, that introduces the concept of performing data association just after the prediction stage, thus avoiding the need for computationally expensive labelling procedures such as clustering which is common in this type of frameworks. We propose a simple yet powerful multi-detector fusion for generating an over-populated detection output that, then, we use to introduce the concept of strong and weak detections. High-confidence (strong) detections are used for label propagation and object initialisation. Low-confidence (weak) detections only support the propagation of labels, i.e. tracking existing objects. Furthermore, we exploit perspective information in prediction, update and newborn particle generation. The proposed tracker, that does not use any image information, runs in real time and results show that our method outperforms alternative online trackers on several tracking benchmarks in terms of accuracy, false negatives and speed [C7].
2. An accurate model to forecast the position of moving objects by disentangling global and object motion without the need of camera calibration or planarity assumptions. The proposed predictor uses past observations to model the motion of objects in an online fashion by selectively tracking a spatially balanced set of keypoints and estimating scene transformations between pairs of frames. The predictor can forecast up to 60% more accurately than state-of-the-art predictors while being resilient to noisy observations. In addition, the proposed predictor is robust to frame-rate reduction and outperforms alternative approaches while processing only 33% of the frames with moving cameras. This allows one to intentionally reduce the video frame rate and hence the energy consumption, an important aspect for moving platforms. We integrated the object motion predictor in a real-time multi-target tracker. Experimental results show that the use of the proposed model improves tracking accuracy with respect to using traditional linear predictors in moving-camera scenarios. Moreover, the proposed tracker obtains results that are comparable to using a linear prediction model by processing only half the number of frames, thus making it suitable for resource-constrained platforms where less number of frames might be desirable [C4, C1].
3. A method for jointly localising objects and estimating their dimensions in 3D using two calibrated RGB cameras and without relying on depth information, markers or 3D models.

We first localise the 3D centroid of the object by using semantic segmentation and projective geometry. Then, we estimate the dimensions of an object by sampling at different heights a set of sparse circumferences with iterative shape fitting and image re-projection to verify the sampling hypotheses in each camera using semantic segmentation masks. We evaluate the proposed method on a novel dataset, that we made publicly available, of objects with different degrees of transparency and captured under different backgrounds and illumination conditions. Experimental results show that the proposed method outperforms existing alternatives that use depth maps in terms of localisation success and dimension estimation accuracy. The proposed method obtains an object localisation success of 86.96% with an average object width estimation error of 0.5 cm. Furthermore, we present a multimodal method that uses the proposed method together with a robotic arm for performing dynamic human-to-robot handovers of unseen objects without relying on motion capture systems, markers, or prior knowledge of the specific objects. We distribute, as open source, the implementation of the overall pipeline to enable comparisons and facilitate research progress [J1, C2].

4. A method to track a moving sound source such as a human speaker from an unmanned aerial vehicle only using audio signals. We combine a time-frequency filtering, peak detection, and particle filtering. The effectiveness of the proposed method is exemplified with experiments that use real-recorded data with a drone platform and a moving sound source. We collected and made publicly available an audio-visual dataset recorded outdoors with an 8-microphone circular array and a camera mounted on a unmanned aerial vehicle. For the evaluation, we propose to use the on-board camera for the sound source localisation annotation for which we design an audio-visual calibration procedure composed of camera calibration, audio-visual temporal alignment and geometrical alignment for allowing the concurrent use of both audio and video streams, which are independently generated [C3, C6].

5. An automatic method for quantifying the uncertainty in the ground-truth annotations, which are often generated using semi-automatic procedures such as linear interpolation, in tracking datasets. To account for this uncertainty when comparing trackers, the proposed method calculates a confidence interval for a given evaluation score and dataset to complement existing tracking scores only using information from the existing ground-truth annotations.

The confidence intervals quantify the uncertainty in the annotation and allow us to appropriately interpret the ranking of trackers with respect to the chosen tracking performance score. Results conducted in widely used tracking benchmarks indicate that existing tracking accuracy metrics cannot be used to rank trackers confidently; moreover, the tracking precision often contains no information that can be used reliably for ranking trackers [C5].

1.4 Organisation of the thesis

The thesis is organised as follows.

Chapter 1: we exposed the motivation and problem formulation for the work of this thesis comprising object localisation, dimensions estimation and tracking from generic input signals. Also, we listed the main five contributions of the thesis.

Chapter 2: we review the existing related work for object localisation and dimensions estimation, object tracking and performance measures for these tasks. We focus the analysis of the related work in object motion prediction and multi-object tracking where we specifically focus in the Probability Hypothesis Density Particle Filter (PHD-PF) framework as it is the base of some of the contributions presented in this work.

Chapter 3: we present an audio-visual calibration for allowing audio and video signals to be used concurrently; then, we introduce a novel algorithm for multi-view object localisation and dimensions estimation of unseen objects; we present the self-collected CORSMAL containers dataset that is used for validation of the method presented in this section; lastly, we propose a multi-modal algorithm, based on the object localisation and dimensions estimation algorithm, to perform human-to-robot handovers of unseen objects with a real robotic arm.

Chapter 4: we display three tracking algorithms, all based on the probability hypothesis density particle filter framework, that tackle different challenges. The first challenge is extremely-low only audio signals (Sec. 4.1). The second challenge is multiple and time-varying number of objects from a static camera (Sec. 4.2). The third challenge is an extension of the previous one where the camera is also moving (Sec. 4.3). A second self-collected dataset is introduced, the Audio-Visual Quadcopter (AVQ) dataset, where the single-object tracker is evaluated. In addition, we introduce a procedure that creates confidence interval for a given evaluation score and dataset to account for the inaccuracies that exist in the dataset annotations.

Chapter 5: we summarise the achievements presented in this thesis and discuss the known limitations and future work.

Chapter 2

Related work

In this chapter we present an overview of existing vision-based object localisation and dimensions estimation algorithms (Sec. 2.1), tracking methods (Sec. 2.2), object motion prediction (Sec. 2.3), multi-object tracking (Sec. 2.4), as focus on the probability hypothesis density particle filter (Sec. 2.5), performance measures for the previous tasks (Sec. 2.6), and finally draw some discussion (Sec. 2.7).

2.1 Object localisation and dimensions estimation

Detection is to localise objects of interest (e.g. people or cars) given an input signal. Detectors can be divided on the basis of they work in 2D (e.g. images) or 3D, as introduced in Sec. 1.2. Object detectors that work in the image plane (2D) can be classified as traditional [47, 54, 197] or deep-learning-based [66, 154, 155, 211]. In this thesis, we employ existing 2D object detectors which generate observations for the forthcoming tracking methodologies introduced in Sec. 4. For further information about 2D object detection, we refer the reader to existing surveys [75, 107, 224]. Similarly, we use state-of-the-art sound source localisation algorithms and their literature review lies outside the scope of this thesis.

In this section, we discuss methods that are able to localise and to estimate 3D physical properties of the objects of interest. Detecting objects in 3D and estimating their properties (*e.g.* dimensions, shape), as well as their 6 Degrees of Freedom (DoF) pose (location, orientation), is important for several robotic tasks, such as grasping [164, 194], manipulation [36] and human-to-robot handovers [122]. However, objects can widely vary in shape, size, material, and transparency, thus making the estimation of their properties through vision a challenging problem. Furthermore, the 3D detection of an object and the estimation of its properties are challenging under different viewpoints, illuminations, intra-class variations, object rotation and scale changes, and occlusions. Moreover, there often exist the requirement for performing with high accuracy and high speed for being suitable for real-time applications. Vision approaches to estimate the physical properties of an object can be marker-based, model-based, or inference based. Marker-based approaches [64, 88, 122] rely on motion capture systems to accurately track the pose of a human hand or object in 3D [64]. These approaches are economically expensive and require markers to be installed on objects which might be undesirable as the properties of the objects might change due to the markers. To avoid using markers, feature points [48, 109] can be localised in an image and matched against a 3D object model to estimate the object pose by solving a Perspective-n-Point problem [98]. However, this strategy may fail when objects exhibit limited texture or are captured under unfavourable lighting conditions [128]. Model-based approaches exploit prior knowledge on the object and its physical properties through dense 3D models [70, 103, 141, 185]. Large amounts of annotated data facilitate the training of deep neural networks to estimate the object pose [70, 103, 141, 185]. However, (manual) annotations are expensive to generate and might be biased [145, C5]. Moreover, this type of approach limits operations to *known* objects. Approaches based on deep neural networks learn from large sets of annotated data with high-level object categories [105] using 3D models or depth data [103, 128, 141, 185, 187]. For example, DenseFusion [185] combines features obtained from RGB-D images and can handle occlusions and inaccurate segmentation. Pixel-wise Voting Network (PVNet) [141] estimates the pose of occluded or truncated objects with an uncertainty-driven Perspective-n-Point, learning a vector-field representation to localise a sparse set of 2D keypoints and their spatial uncertainty. Normalized Object Coordinate Space (NOCS) [187] uses a normalised object coordinates space formulation that jointly estimates the 6 DoF pose and the dimensions (in the form of a 3D bounding box) of a novel object (*i.e.* an object whose shape was not seen during training). Although deep neural networks estimate the 6 DoF object pose quite accurately, their training requires large amount of data usually annotated only for the high-level object category, containing images and/or *known* dense 3D models [70, 128, 141, 185, 187]. For example, PoseCNN [205], DenseFusion [185], SegOPE [70] and PVNet [141] evaluate only on objects with high-quality 3D models and good visibility in depth [205]. Table 2.1

Table 2.1: Comparison of markerless methods for localising in 3D and estimating the dimensions of textureless objects. Note that Saxena et al. [163] localise the grasping points in 3D instead of the object. KEY: Ref., reference; N-3D, no 3D object model; N-D, no depth; HLC, known high-level category; Loc., object localisation in 3D; and Dim., object dimensions estimation in 3D. ■ dimensions given by the 3D model.

Ref.	**Method**	**Assumptions**			**Tasks**		**Transparency**
		N-3D	N-D	HLC	Loc.	Dim.	
[163]	Saxena	✓	✓		✓		✓
[103]	DeepIM		✓		✓	■	
[128]	StoCS			✓	✓	■	
[141]	PVNet		✓	✓	✓	■	
[185]	DenseFusion			✓	✓	■	
[70]	SegOPE		✓	✓	✓	■	
[187]	NOCS	✓		✓	✓	✓	
Sec. 3.5	**LoDE**	✓	✓	✓	✓	✓	✓

compares relevant works, and comprehensive reviews of object pose estimation can be found in [70, 141, 185, 187]. With previously unseen objects, inference-based approaches localise them in 3D using only partial knowledge, such as dimensions, estimated through uni-modal [39] or multi-modal sensing [151], or by estimating the 6-DOF of textured objects by solving the Perspective-N-Point problem with known 2D-3D correspondences [169][98]. However, typical objects with textureless, reflective or transparent materials, like those in our benchmark, hinder reliable correspondences [97]. Moreover, depth sensors [187] cannot be applied with sufficient accuracy [163] or speed for a smooth handover [119].

2.2 Tracking

Tracking can be defined as localising an object in all frames of a video sequence [112]. Tracking approaches perform temporal association of object hypothesis to estimate trajectories while compensating for miss-detections and rejecting false-positive detections. A comprehensive literature review in generic object tracking is presented and discussed in this section.

Tracking can be categorised on the basis of different criteria including the number of objects, reference system, number of sensors, initialisation mode, processing mode and mathematical methodology. Table 2.2 shows an overview of the proposed tracking categorisation.

2.2.1 Number of objects

The number of objects to be tracked can be used as another criteria to categorise tracking methods either as *Single Object Tracking* (SOT) or *Multiple Object Tracking* (MOT). SOT works [80,

Table 2.2: Tracking categorisation according to the number of objects, reference system, number of sensors, initialisation mode, processing mode and mathematical methodology.

Number of objects	Single Multiple
Reference system	Image plane World coordinates
Number of sensors	Single Multiple
Initialisation mode	Detection free Tracking-by-detection
Processing mode	Online Buffered Offline
Mathematical methodology	Probabilistic Deterministic

214] mainly focus on building robust appearance models or motion models to cope with scale, rotations and illumination variations. MOT methods [15, 149, 217] additionally need to assign an identity to each object and maintain it throughout the time and deal with further challenges including inter-object occlusions, initialisation and termination of trajectories, unknown number of objects, similar appearance among objects or interaction among them.

2.2.2 Reference system

Similarly to detectors, trackers can also be classified according to the reference system used. Tracking can be classified according to the reference system used: *image plane* or *world coordinates*. Most of the current approaches work on the image plane reference system mainly for simplicity reasons [30, 38, 80]. Tracking in the world coordinates can potentially provide more accurate estimations and effective occlusion handling [140]. When the 3D vertical component is avoided, assuming planar ground, objects are considered to move on the ground plane and occlusions can be handled more robustly [149].

2.2.3 Sensors

Tracking can be performed with data captured by a *single sensor* or *multiple sensors*. Single-sensor (e.g. camera) tracking is an old problem that has been studied in depth and lot of works have been published [80, 115, 117, 138, 181]. Multiple state-of-the-art surveys have been published in the last decades covering these works and analysing in detail the multiple challenges present in this field [112, 203, 213]. These challenges are still open research problems such as: drastic appearance variations, similar appearance among objects, occlusions or moving cameras among others. Multi-sensor tracking allows to deal with more challenging scenarios where single-sensor tracking cannot cope with such as tracking over wide areas (greater than the field of view of a single camera) or crowded scenes. Multi-sensor scenario [27, 86, 121, 130, 179] has multiple sensors spread on the scene. When multiple static cameras are used for tracking, it is known as camera network scenario. Tracking using multiple sensors provides several advantages compared to single-sensor tracking such as: potentially more accurate tracking, robustness to sensor failures and scalability [161]. Multi-sensor tracking adds several challenges including camera calibration, topology of the network, drastic illumination and appearance variations. The case where multiple moving sensors are used provides more flexibility since the field of view of each sensor is dynamic and might be adapted according to the necessities of the scenario. However, the difficulty of the challenges in this scenario increases and includes new ones such as inter-camera tracking, fusion, communication and synchronisation. RGB cameras are considered as the default sensor in this thesis.

Sensors can capture information of different nature including optical/visual, acoustic, thermal or electrical signals. A single sensor can capture homogeneous information such as RGB [118, 149], infrared [22, 59, 214], thermal [99, 100] or monochrome [93]. Sensors of different nature can be collaboratively used for tracking. For example, RGB-D captures RGB and depth information [59] or RGB-W captures RGB and wireless signals [9] and both signals are used for performing tracking.

2.2.4 Initialisation

Methods can be grouped into two categories according to how to initialise trajectories: *Tracking-by-detection* (TBD) and *Detection Free* (DF). In TBD, *object hypothesis*, also known as detections or observations, are first detected in each frame and then temporally associated into trajectories [9, 126, 217]. Object hypotheses approaches can be divided into two groups: *object-detection based* and *motion-detection based.* When objects are people, a multitude of detectors have been proposed. Object detectors may search for trained intensity patterns such as [44, 47, 54]. These intensity patterns can be the colour gradient (e.g. edges) [44, 54], the spatial arrangement of colours (e.g. textures) [196], colour attributes [85] or having a high response to a pool of filters (deep learning) [212]. Detection fusion aims to combine candidate object position generated by diverse detectors in order to blend different designs and to boost detection reliability. Motion-based detectors model the background (e.g. applying frame difference) to then detect object hypothesis on the motions detected on the foreground. However, DF requires a manual initialisation of a fixed number of objects in the first frame, then the method tracks these fixed number of objects in the next frames [80, 118, 214]. As a summary, on one hand TBD approaches can deal with varying number of objects and, initialisation and termination of trajectories but are limited to a specific type of object and its performance is closely related with the object detector's performance. On the other hand, DF approaches cannot cope with varying number of objects, as manual initialisation is required.

2.2.5 Processing mode

Trajectories can be generated *offline* [184], buffered [147] or *online* [217]. Offline trackers use both past and future detections and can therefore better cope with miss-detections using long-term re-identification [167]. While offline methods can theoretically obtain global optimal solution generally providing a higher tracking accuracy compared to online approaches, they generate trajectories with delay and cannot be used for time constrained applications such as autonomous navigation of robots [53], and time-critical applications such as object re-identifications in large-scale camera networks [61]. The global optimum is usually formulated as a particular type of optimisation problem. For example, Bipartite Graph Matching [167], Dynamic Programming [201], Min-cost Max-flow Network Flow [221] or Conditional Random Fields [127, 209]. There is a subgroup of offline trackers known as buffered, also known as short-latency, that collects detections within a time window to estimate trajectories with a delay with respect to the current time instant. Online trackers estimate the object state at each time instant as detections are produced. Online trackers are suitable for real-time applications but suffer from shortage of observation. Online tracking can be performed by means of frame-by-frame detection associations (e.g. Hungarian algorithm [90]) with adaptive features based on the surrounding scene complexity [172]. Ambiguous association zones are identified based on the proximity of objects within these zones. Features such as appearance and motion can be used for object discrimination. Alternatively, detection association between consecutive frames can be achieved with Markov Decision Processes (MDPs) [204]. MDPs define the state evolution (trajectory) in terms of actions and rewards received based on these actions. In order to quantify the rewards, the MDPs need to learn the parameters. Features used in this context are optical flow, appearance, position and bounding box size. To overcome object occlusions, online trackers can use (estimated) physically plausible paths through occluded regions in order to re-assign the identity after an occlusion [149]. When the number of objects in a scene is large, occlusions are more likely to occur and motion in previous time steps can help tracking [24, 89]. In case of miss-detections, online trackers may rely on predictive models to continue tracking until a matching detection is found [149]. An object motion model can be designed based on the optical flow of the scene and can be used to predict the objects' position under the constrained motion of a crowd. There exists some relations between processing mode and initialisation mode. DF methods require to compute the state of the objects sequentially (i.e. in an online fashion) as the video stream is generated.

2.2.6 Mathematical methodology

Trackers can be divided into *probabilistic* and *deterministic* according to the mathematical methodology used. In probabilistic tracking [116, 156, 159, 181], object states are represented with probabilistic distributions that are being updated as the current observations are generated. The estimation of the states might vary among different executions over the same input data. Tracking based on deterministic optimisation [15, 167, 201, 202] collects the observations in an image sequence and some relations among them are built based on their appearance, spatial and temporal similarities. Then the tracking problem becomes an optimisation problem where the estimations are based on deterministic optimisation resulting in constant results over different trials. There exists some relations between mathematical methodology and processing mode. Most online approaches estimate the state of the objects in a probabilistic way while offline approaches always follow deterministic optimisation in the calculation of the states.

2.2.7 Discussion

To conclude, tracking is a well studied field but multiple challenges remain open research problems. We present in this thesis, tracking methodologies for *single* and *multiple* sensors. Most possible applications based on tracking including statistics, crowd analysis or surveillance need to obtain information from *multiple objects*, which can also be applicable to single-object tracking. The number of objects can vary in time since objects might enter and exit the field of view, consequently initialisation and termination processes are required as the number of objects at each time is unknown. Therefore, *tracking-by-detection* is the most common solution for MOT since this approach allows to automatically initialise new objects in each frame. As discussed earlier in this section, when objects are people or cars, multitude of pre-trained detectors are available. Applications become interesting in this scenario when the processing can be done in real time. Therefore only *online* methods are suitable for these scenarios that usually adopt *probabilistic inference* for the state estimation.

2.3 Object motion prediction

Object motion predictors model the dynamics of objects and estimate their future position on the image plane or in 3D. The ability to predict the motion of objects across frames is important for applications such as video coding [63, 74, 153], social-aware autonomous robots [110, 198], action recognition [186, 195], lifelogging [31], and object tracking [16, 34, 39, 93, 120]. In particular, an accurate object motion prediction is important in MOT applications since it allows to predict the potential position of objects in future frames which can be critical in presence of occlusions or low frame rate videos. Object motion prediction become essential in applications that require online MOT as decisions need to be made just with current information and cannot rely on future measurements and where other important tracking features such as appearance are not available (e.g. sports or insect tracking).

Moreover, cameras are becoming smaller due to technology miniaturisation and now they are installed in ground robots, drones and wearable devices. Likewise, power batteries need to become smaller in size to fit within the devices, thus efficient use of battery capacity becomes an essential point for achieving a desired trade-off between performance power and battery duration. In the MOT scenario, accurately predicting object trajectories in the future tens of frames can save computation workload as only one computation at the present time can be enough to accurately predict the object location for the next tens of frames. Therefore, accurate object motion predictors can potentially save battery life, a crucial point in mobile devices [162].

2.3.1 Static camera

It is a common practise to assume that objects move smoothly between frames, i.e. at constant velocity and/or with certain process noise [33, 165, 218]. A linear motion model performs sufficiently well with static cameras [33, 165, 218] for objects such as pedestrians and cars that approximately follow linear trajectories in the real world when captured by a camera at high frame rate. In this modelling, the velocity of objects in the next frame is the same as the one in the previous frames with certain independent process noise as

$$(u,v)_k = (u,v)_{k-1} + (\bar{u},\bar{v})_{k-1}\Delta_k + \mathbf{N}_k^{(u,v)} \tag{2.1}$$

where $(\bar{u},\bar{v})_{k-1}$ is the estimated velocity, Δ_k is the time lapsed between consecutive frames and $\mathbf{N}_k^{(u,v)}$ is the process noise that models the unknown changes that might happen over time to the location of the object (e.g. camera motions). The velocity is modelled similarly as

$$(\bar{u},\bar{v})_k = (\bar{u},\bar{v})_{k-1} + \mathbf{N}_k^{(\bar{u},\bar{v})}, \tag{2.2}$$

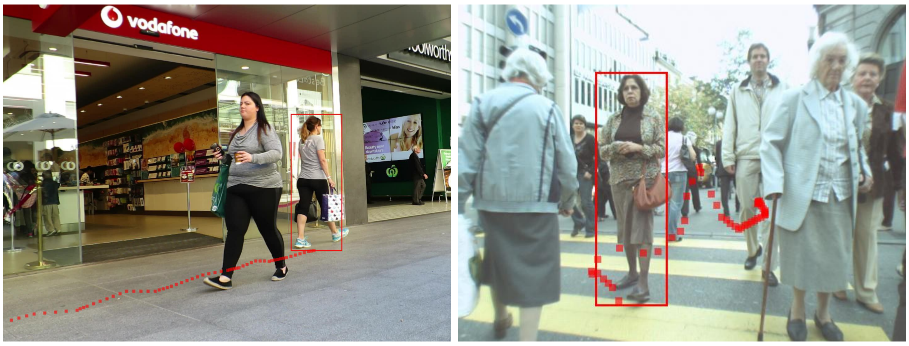
where $(\bar{u},\bar{v})_{k-1}$ is the estimated velocity at time $k-1$ and $\mathbf{N}_k^{(\bar{u},\bar{v})}$ is the process noise that models the unknown changes that might happen over time to the velocity of the object (e.g. accelerations).

These models have been proved to perform well in static cameras [33, 165, 218] as objects such as pedestrians or cars use to follow linear trajectories in the real world.

This type of modelling, known as linear motion prediction (LMP), is the most commonly used in the tracking literature because of its simplicity and good accuracy in static-camera scenarios. However, it is inadequate for moving cameras. Fig. 2.1 depicts a graphical example where two pedestrians (feet position marked in red) move on a straight line in the real world but the perceived motion on the image plane drastically differs from a straight line. This is because Fig. 2.1a is recorded from a static camera whereas Fig. 2.1b is recorded from a moving camera. It is clearly visible that LMP models cannot accurately model moving-camera scenarios, even that the objects move linearly on the real world. This occurs because LMP models cannot estimate the object motion as this is masked with the one of the camera. Therefore, LMP can potentially fail/drift in presence of object direction changes, unexpected object motions, reduced frame rate videos or moving cameras.

2.3.2 Moving camera

In sequences where the camera is moving, the motion of an object is challenging to estimate as its motion is combined with that of the camera and both, object and camera, potentially move simultaneously and independently. In practice, tracking drifts occur when LMP is used in moving cameras. Multiple methods propose to estimate the camera motion for improving the accuracy



(a) (b)

Figure 2.1: Moving pedestrians recorded from (a) static camera and (b) moving camera. Both pedestrians are following a (similar) linear trajectory in the real world but the apparent motion on the image planes is not linear when the camera moves. The bounding box indicates the location of the person in the current time instant. Red solid squares indicate the location of the object in each time instance in the last 2 seconds. Source images from [94].

Table 2.3: Object motion predictors. KEY: Ref, reference; CS, coordinate system; A, approach; L, linear; NL, non linear; CM, robust to camera motion; FS, robust to frame skipping; NC, works without camera calibration; NS, not scene specific; and FO, objects can move freely in the scene.

Type	**Ref**	**Strategy**	**CS**	**A**	**CM**	**FS**	**NC**	**NS**	**FO**
Data driven	[180]	learns typical motion patters using clustering	2D	NL			✓	✓	✓
	[8]	accounts for person-to-person interactions with LSTMs	2D	NL			✓	✓	✓
	[144]	accounts for person-to-person interactions and obstacles with LSTMs	2D	NL			✓	✓	✓
	[19]	handles long-term occlusions with LSTM	2D	NL			✓	✓	✓
Model based	[7]	probabilistic multimodal approach	2D	NL			✓	✓	✓
	[208]	online learning for prediction of objects and groups of objects	2D	NL			✓	✓	✓
	[207]	pedestrian trajectory prediction	2D	NL			✓	✓	✓
	[131]	Brownian model	2D	L			✓	✓	✓
	[218]	Markov Chain Monte Carlo data association	2D	L			✓	✓	✓
	[165]	linear modelling with Gaussian noise	2D	L			✓	✓	✓
	[34]	linear modelling with Gaussian noise	2D	L			✓	✓	✓
	[115]	accounts for perspective and frame-rate	2D	L			✓	✓	✓
	[C7]	accounts for perspective and frame-rate	2D	L			✓	✓	✓
	[120]	frame registration (aerial video) for camera motion estimation	2D	L	✓		✓		
	[101]	homography (aerial videos) camera motion estimation	2D	L	✓		✓		
	[16]	ground plane prediction (homography) for camera motion estimation	2D	L	✓				
	[39]	geometry priors (requires an RGB-D camera) for camera motion estimation	3D	L	✓			✓	✓
	[93]	frame-registration (aerial videos) for camera motion estimation	2D	L	✓		✓		
	Sec. 4.3.1	decouples global and object motion (homography)	2D	L	✓	✓	✓	✓	✓

in their application task. Indeed, the estimation of the camera motion is an important source of information for multiple computer vision applications including: simultaneous localisation and mapping [51, 132], action recognition [186, 195] and multiple object tracking [16, 39, 93, 120]. Next, we overview the most relevant related works for object motion prediction.

Predictors may be specifically designed for static [8, 180] or moving cameras [101, 120]; may assume that objects move on a common ground plane [93]; or may need camera calibration information [16, 39]. We can identify two main classes of predictors, namely data-driven and model-based predictors (see Table 2.3).

Data-driven predictors learn patterns from (large amounts of) training data [8, 19, 144, 180]. In structured environments, a clustering-based method can be used to predict the motion pattern of people [180]. A Long Short-Term Memory (LSTM) can predict the position and scale of objects [19]; whereas multiple LSTMs can learn person-to-person interactions to predict the position of people [8, 144]. Model-based predictors [7, 207, 208] may rely on no assumptions on object motion [131] or may assume that objects maintain a certain velocity, which is learned over recent observations [34, 165, 218]. First or higher order Markov models can be used to model motion [7, 34], or an independent noise component can account for velocity variations (i.e. accelerations) [165, 218]. Acceleration and noise can be modelled based on the camera-object distance [115, C7], frame rate [115, C7], or physics [16]. Moreover, non-linear motion patterns can be learned online with a hierarchical association of tracklets [208].

The motion predictors mentioned above are applicable to static cameras only. There exist other model-based approaches that account for both object and global (camera) motion [16, 39, 93, 101, 120]. *Object motion* is the observed motion of an object on the image plane when the camera is stationary. *Global motion* is the observed motion of an object on the image plane when, in the 3D world, the object is stationary and the camera moves. Global motion information may be extracted from the coherent motion of the background by considering moving objects as outlier motions, for example if they move on a common plane [16, 93, 101, 120]. Model-based predictors for moving cameras may use transformations among planes across consecutive frames [16] or 3D models that account for interacting objects using global and object motion, geometric constraints and a reversible jump Markov chain Monte Carlo particle filter [39].

Camera pose is another constraint that can be used by predictors: some methods are applicable only to top-down looking cameras mounted on a drone when the scene can be considered planar [93, 101, 120]. This assumption simplifies the explicit decoupling of camera and object motion, which is obtained by estimating a frame-by-frame transformation (e.g. homography or registration) and then by subtracting camera motion from observed motion. To conclude, methods and datasets, such as the trajectory Forecasting Benchmark [23], often consider only static or top-down looking cameras on high-altitude stationary drones and are therefore limited to specific scenarios.

2.4 Multi-object tracking

MOT, also known as multiple target tracking, is an interesting field included in the general tracking problem as seen in the previous section. MOT can be employed to automatically process large quantity of videos for video analysis and annotation [146]. When objects are people or cars, a large amount of highly accurate detectors have been developed for those specific type of objects. Thus, TBD is a commonly used framework to tackle the tracking problem of this type of objects. TBD in MOT estimates object states by temporally associating detections, also known as measurements or observations, and assigning identities based on appearance-spatio-temporal relationships while compensating for miss-detections and rejecting false-positive detections. MOT faces multiple challenges including short- and long-term occlusions, initialisation and termination of trajectories, unknown number of objects, similar appearance among objects, interaction among objects, drastic object appearance variations, potentially moving cameras or dynamic backgrounds.

In general, the main components of MOT are *observation model* and *motion model*. Observation model measures the similarities between trajectories and detections including appearance, motion, interaction and exclusion modelling, and occlusion handling. Whilst motion model studies the transition of the states across time.

2.4.1 Appearance modelling

Appearance modelling is considered as a vital cue in visual MOT since it can be highly discriminative in certain scenarios. Selecting the right features is a critical point in tracking. In an ideal case, the set of features should unequivocally identify each object, as well as distinguish it from others, in the chosen feature space. Appearance modelling can be seen as a two-step process: *visual representation* and *statistical measuring*. Visual representation is how to describe the visual characteristics of an object and is visually based on single cues or in multiple cues. Certain features are optical flow [12], covariance matrix [148], feature points [35], gradient based features [44], depth [129] or colour [135]. Statistical measuring is needed after the object has been visually represented via features. It can be defined as the computation of similarity (or dissimilarity) among different objects.

2.4.2 Interaction model

Interaction model, also known as mutual motion model, analyses the relation of an object to other spatially close objects. The speed and direction of an object is influenced by its neighbours. For example, a car moving on a road will be influenced by other cars on the road, known as *social force model* [68]. The systematic temporal changes of the preferred velocity of an object are described by a vectorial quantity that can be interpreted as a social force that represents the effect of the environment on the behaviour of the object. When a crowd of objects moves in a scene, each of them unconsciously follows others' patterns and guides others at the same time in order to avoid collisions, known as *crowd motion pattern model* [69]. A motion pattern is defined as a group of flow vectors that are part of the same physical process of motion pattern. This is accomplished by first detecting the representative modes (sinks) of the motion patterns, followed by construction of super tracks, which are the collective representation of the discovered motion patterns.

2.4.3 Exclusion model

Exclusion model is a component for the MOT problem that can be considered as two independent constraints: *detection-level exclusion* and *trajectory-level exclusion*. Detection-level exclusion claims that two detections in the same time step cannot correspond to the same trajectory [127]. The detection-level exclusion can be modelled by defining a cost term that penalises if two simultaneous detections at time t are assigned to the same trajectory with a cost if they are distant. Another way is to construct an exclusion graph [91] to capture the constraint that detections at the same time should have different labels and then calculate the Laplacian matrix to this graph and maximise the label error regarding the exclusion [111]. Trajectory-level exclusion claims that two trajectories cannot be observed by a single detection [127]. The trajectory-level exclusion modelling can be defined as an energy function that considerably generates large costs when two detections are spatially close [15].

2.4.4 Occlusion handling

Occlusion handling is one of the most challenging tasks present in MOT. Occlusions can lead to identity switches or trajectory fragmentation. Next, we discuss several approaches that have been proposed to tackle this problem. *Part-to-whole* techniques are commonly used and they are based on the assumption that part of the object is still visible during occlusions. Part based models allow to understand which parts of the object are occluded and to estimate the state of the whole object by only observing the visible parts [210]. A double-person detector with different levels of occlusions is trained in [176, 177]. *Buffer-and-recover* techniques accumulate observations when occlusion happens. When occlusion ends, object states are recovered based on the stored observations and states [129]. These approaches might produce delays on the decisions as they require to wait certain frames to take a decision, thus they are not suitable for online tracking. More recently, other solutions have been proposed to solve occlusions. For example, [149] proposes to only rely on geometric information to efficiently overcome detection failures and short-term occlusions. A collaborative tracking approach for addressing inter-occlusions is proposed in [206]. Moreover, performing 3D tracking within multi-part detectors to tackle this MOT challenge is proposed in [200]. This type of approaches can potentially better cope with challenging occlusions as objects can be spatially more distant in the 3D world system than as they appear in the 2D (i.e. image plane) re-projection.

2.4.5 Multi-object trackers

In this section, we show and critically analyse some related works to MOT and specially MOT with moving cameras. Possegger [149] proposed an online approach that assumes that the observation that object detectors primarily fail if objects are significantly occluded. Maintaining the identities of the objects is one of the principal purposes in MOT. This is very challenging in presence of multiple occlusions, illumination variations and moving cameras. The spatio-temporal evolution of occluded regions, detector reliability, and object motion prediction are exploited to robustly handle miss-detections trough occlusions. No appearance information is used just relying on spatial distances. This is an interesting work that obtains promising results with respect to state-of-the-art MOT methods. However, this approach is only applicable to the static camera scenario. Yu [217] proposed a high-performance detection and deep-learning-based appearance features showing a significant improvement in MOT results in both online and offline settings. The tremendous popularisation of Neural Networks in the Computer Vision community in the last years contributed to the creation of more accurate object detections and appearance representations that are two important pillars in the MOT problem. The appearance representation is based on a similar network to GoogLeNet [173] trained with nearly 119 K from 19,835 identities. The online tracking version is implemented with a KF [81] and Hungarian algorithm [90] whereas the offline version is based on K-Dense Neighbours [106]. Despite remarkable results in benchmarks such as MOT2016 [125], the approach cannot cope with occlusions in an online way.

An uncalibrated monocular camera is used in [40] by Choi to jointly estimate object trajectories, corresponding 2D/3D temporal trajectories in the camera reference system as well as the camera parameters (pose, focal length, etc) in a probabilistic formulation. An interaction (attraction and repulsion) model is presented to model multiple 2D/3D trajectories in space-time and handle situations where objects are occluded. Camera extrinsic parameters (location and orientation) are estimated using sparse feature points as additional observations inspired in Visual Simultaneous Localisation and Mapping procedures such as [45, 171] or direct methods such as Direct Sparse Odometry [50]. This allows to better estimate the motion model of the objects and better predict the estimation of the object locations. To this end, multiple assumptions are set in order to reduce the required number of camera parameters for allowing a 3D re-projection. For example, all objects rest on the ground plane and the camera only moves forwards.

From moving cameras, specifically from aerial views, different features in the videos can be observed: (i) size of the objects is considerably smaller, thus common appearance features are not reliable, (ii) frame rate is usually lower and (iii) object density is likely to be higher w.r.t the non-aerial scenario. Therefore, solutions are also different than the usual MOT scenario. Most works in the literature rely on DF approaches that rely on motion-based detection after background modelling. Aerial Tracking Learning Detection is presented in [118] introducing compensation in the camera motion and algorithmic modifications for combining appearance and motion cues for detection and tracking. TLD [80] considers both appearance and motion features for SOT handling short-term occlusions and maintaining the object in long sequences. However, TLD cannot cope with the problems present in aerial images such as frequent pose changes, scale and illumination variations and the low resolution images, noise and jitter introduced by the camera motion [118]. TLD is extended to multiple objects using multi-thread processing but limited to a maximum of three objects due to complexity issues and for not compromising the real time performance. Lankton [93] proposed a similar work focusing their contributions on decoupling the camera and object motion. In cases where the camera is not fixed due to pans and tilts in pan-tilt-zoom cameras or due to moving platforms as in the unmanned aerial vehicle (UAV) scenario they claim that camera motion must be taken into account. A tracking system estimates the camera motion by performing frame registration [84] and the resulting signal is used into

Table 2.4: Literature review on MOT methods with a single camera comparing sensor types, initialisation mode, processing mode, filtering and association, reference system, the use of appearance and camera calibration and whether the methods are robust to moving cameras (i.e. motion model robust to camera motion, MMRCM) and occlusions. Key: TLD, Tracking-Learning-Detection; KF, Kalman Filter; UKF, Unscented Kalman Filter; PF, Particle Filter; PHD-PF, Probability Hypothesis Density Particle Filter; KDN, K-Dense Neighbours; HDNS, Hierarchical dense neighbourhoods searching; MHT, Multi Hypothesis Tracking; DGM, Dynamic Graphical Model; EM, Energy Minimisation; IMU, Inertial measurement unit; and GNN, Global Nearest Neighbour. Methods with * indicate that perform SOT.

Ref	Sensor type	Initialisation		Tracking			Reference system		Observation model		
		TBD	**DF**	**Online**	**Filtering**	**Association**	**Image**	**World**	**Appearance modelling**	**MMRCM**	**Occlusion handling**
[118]	RGB		✓	✓	TLD	-	✓		✓	✓	
[93]	Gray		✓	✓	KF	Mean-shift	✓		✓	✓	
[217]	RGB	✓				KDN	✓		✓		
[199]	RGB	✓				HDNS	✓	✓	✓		✓
[30]	RGB	✓				MHT	✓		✓		
[38]	RGB	✓				DGM	✓		✓		
[15]	RGB	✓				EM		✓	✓		
[217]	RGB	✓		✓	KF	Hungarian	✓		✓		
[22]	RGB-D	✓		✓	UKF	GNN		✓	✓		
[149]	RGB	✓		✓		Hungarian	✓				✓
[40]	RGB	✓		✓	PF	Hungarian	✓	✓	✓	✓	✓
[142, 143] *	RGB + IMU		✓	✓	TLD	-	✓		✓	✓	
Sec. 4.2	RGB	✓		✓	PHD-PF	Hungarian		✓	✓	✓	✓

the KF [81] for the estimation of the object states. Detections are generated from a mean-shift localisation scheme [42]. Pestana [142, 143] proposes to perform visual tracking of a single object with a DF approach from an UAV. Their architecture allows the user to specify a target object in the image that the UAV has to follow from an approximate constant distance. Inertial measurement unit and ultrasound altitude sensor data are used to update the control algorithm.

Table 2.4 summaries the related work in MOT with a single sensor according to the previously described classification. We introduce and discuss the *Probability hypothesis density particle filter* (PHD-PF) [115, 116, 181] in the next section as it is the framework where we base the tracking contributions in this thesis.

2.5 The Probability Hypothesis Density Particle Filter

The PHD-PF framework is an effective filter for online state estimation, which can cope with clutter, spatial noise and miss-detections [115, 116]. The PHD filter estimates the state of multiple objects by building a positive and integrable function over a multi-dimensional state whose integral approximates the expected number of objects [116, 181]. The posterior distribution can

Table 2.5: Literature review on multi-object tracking methods based on Probability Hypothesis Density comparing the filtering and data association approaches and whether detection classification and perspective dependency is used.

Ref	**Filtering**		**Data Association**		**Other characteristics**	
	PHD	**PF**	**Clustering**	**Hungarian**	**Detection classification**	**Perspective dependency**
[116]	✓					
[117]	✓					
[181]	✓	✓				
[138]	✓	✓	✓	✓		
[115]	✓	✓	✓	✓		
Sec. 4.2	✓	✓		✓	✓	✓

be computed based on a Bayesian recursion that leverages the set of (noisy) detections and it is approximated using Sequential Monte Carlo for computational efficiency via a set of weighted random samples (particles) [181]. This approximation is known as the PHD Particle Filter (PHD-PF) and involves four main steps [115]: the *prediction* of particles over time; the *update* of the weights of the particles based on new detections; the *resampling* step to avoid that only few particles monopolising the whole mass; and *state estimation*. However, no information about the identity of the objects is present but to perform MOT, consistent identification of the objects is required. PHD filters need an additional mechanism to provide object identity information (i.e. data association). Aiming to address the lack of identities, various approaches have been published: (i) clustering after resampling the particles [115], (ii) keeping a separate tracker for each object and then perform "peak-to-track" association [104], (iii) combining clustering techniques with the introduction of hidden identifiers to the samples of the PHD [138, 139, 181]. Table 2.5 summaries the related PHD-based trackers where the different approaches are compared according to filtering, data association and other characteristics such as detection classification and perspective dependency. Note that [116, 117, 181] do not perform any type of data association, therefore no identities are provided with the estimated states.

Let us consider a Bayesian framework where the posterior probability distribution, $p(\mathbb{X}_k|\mathbb{Z}_{1:k})$, can be recursively estimated as [116]:

$$\underbrace{p(\mathbb{X}_k|\mathbb{Z}_{1:k})}_{\text{Posterior}} \propto \underbrace{p(\mathbb{Z}_k|\mathbb{X}_k)}_{\text{Observation Model}} \int \underbrace{p(\mathbb{X}_k|\mathbb{X}_{k-1})}_{\text{Prediction Model}} \underbrace{p(\mathbb{X}_{k-1}|\mathbb{Z}_{1:k-1})}_{\text{Previous Posterior}} d\mathbb{X}_{k-1},$$

where $p(\mathbb{X}_k|\mathbb{X}_{k-1})$ is the prediction model that defines the evolution of the state over time and the observation model and $p(\mathbb{Z}_k|\mathbb{X}_k)$ is the observation model that quantifies the likelihood of the states to generate the detections.

This Bayesian inference can be solved using the PHD-PF that approximates the posterior probability distribution with L_k particles (i.e. samples) where each of them has an associated weight π_k^i describing its importance towards the estimation of the states. Specifically, the PHD-PF distribution is approximated as

$$p(\mathbf{x}_{k-1}|\mathbb{Z}_{1:k-1}) \approx \sum_{i=1}^{L_k} \pi_{k-1}^i \delta\left(\mathbf{x}_{k-1} - \dot{\mathbf{x}}_{k-1}^i\right), \tag{2.3}$$

where $\delta(\cdot)$ is the Kronecker's delta function and $\dot{\mathbf{x}}_{k-1}^i$ is the i-th particle. As a filtering method, the PHD-PF does provide *no* information about the identity of the objects [117]. To perform as a (multi-object) tracker, the PHD-PF requires the use of additional mechanisms such as data association to introduce the concept of object identity.

We base our contributions described in this chapter in the PHD-PF proposed in [115]. The PHD-PF involves four main steps: the *prediction* of particles over time; the *update* of the weights of the particles based on new detections; the *resampling* to avoid that only few particles monopolising the whole estimation; and *state estimation*.

2.5.1 Prediction model

The prediction model models the motion of the objects over time and can be modelled as

$$\mathbf{x}_k^i = f_p(\mathbf{x}_{k-1}^i) + \mathbf{N}_k, \tag{2.4}$$

where $f_p(\cdot)$ is the object motion function that describes the temporal evolution of the object and $\mathbf{N}_k$ is the noise matrix that accounts for unmodelled factors.

2.5.2 Observation model

The observation model estimates the importance of each particle towards the estimation of the state as

$$\pi_k^i = f_o(\pi_{k-1}^i, \mathbb{Z}_k) \tag{2.5}$$

where $f_o(\cdot)$ is the observation function (also known as likelihood function).

2.5.3 Resampling and state estimation

The resampling step modifies the particle weights so that particles with larger weights are maintained and replicated, whereas those with lower weights are removed. Specifically, we use the multi-stage resampling [115], which uses multiple stages depending on the frame where the particles were created. This ensures that particles recently created for newly appearing targets, and likely to have a lower weight than older ones, are not removed. Finally, the states at time k is estimated as in Eq. 2.3. For further details in the PHD-PF baseline, please refer to [115].

Various works have been published aiming to address the lack of identities in the PHD-PF: (i) clustering after resampling the particles [115], (ii) keeping a separate tracker for each object and then performing 'peak-to-track' association [104], (iii) combining clustering techniques with the introduction of hidden identifiers to the samples of the PHD [138, 139, 181]. Clustering methods might introduce estimation errors and might require to know a priori the number of objects. Initialising a separate tracker for each object might be computationally expensive. We present in Sec. 4.2 how we perform MOT, keeping identity information, within the PHD-PF framework.

2.6 Performance measures

We refer as ground truth (GT) to the set of manual annotations for a given dataset defined as $\tilde{\mathbb{X}} = \{\tilde{\mathbf{x}}_{k,\lambda} : \lambda \in \Lambda_k; k = 0, ..., K_\lambda - 1\}$ composed of $|\Lambda_k|$ objects that exist during K_λ frames length each. Let an annotated object $\tilde{\mathbf{x}}_{k,\lambda}$ be defined as

$$\tilde{\mathbf{x}}_{k,\lambda} = (\tilde{u}_{k,\lambda}, \tilde{v}_{k,\lambda}, \tilde{w}_{k,\lambda}, \tilde{h}_{k,\lambda}), \tag{2.6}$$

where $\tilde{u}_{k,\lambda}$, $\tilde{v}_{k,\lambda}$ are the horizontal and vertical coordinates of the object (e.g. object centre) and $\tilde{w}_{k,\lambda}$ and $\tilde{h}_{k,\lambda}$ are the width and height of object λ at frame k.

As mentioned earlier, for simplicity in the notation, we avoid writing the sub-scripts of the components of the states in the remaining parts of the thesis.

2.6.1 Object localisation

The localisation performance considers true positive (TP), false positive (FP) and false negative (FN). These measures quantify the discrepancy between the tracking estimate and the GT. To specify each of them, we first define the intersection over union (IOU) as:

$$\mathrm{IOU} = \frac{\mathrm{overlap(estimation,\ \ GT)}}{\mathrm{union(estimation,\ GT)}}, \tag{2.7}$$

where overlap$(\cdot)$ is a function that computes the overlapped area, in the image plane, between an estimation and GT; and union$(\cdot)$ is a function that computes the union area, in the image plane, between an estimation and GT. We define a TP as a correct detection (IOU ≥ 0.5), a FP as a wrong detection (IOU < 0.5) and a FN as an annotation not detected. Upon these measures, we calculate the precision as

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = \frac{\mathrm{TP}}{\#\ \mathrm{detections}}, \tag{2.8}$$

the recall as

$$R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = \frac{\mathrm{TP}}{\#\ \mathrm{annotations}}, \tag{2.9}$$

and F1-Score as

$$\text{F1-score} = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P}+\mathrm{R}}, \tag{2.10}$$

where # indicates *number of.*

2.6.2 Dimensions estimation

For computing the performance in estimating the dimensions of an object, we compute the absolute error between the estimated and annotated width and height for a given object as

$$\lambda_k^w = |\tilde{w}_k - w_k| \tag{2.11}$$

and

$$\lambda_k^h = |\tilde{h}_k - h_k|. \tag{2.12}$$

2.6.3 Object motion prediction

The prediction accuracy is computed as the squared error between the predicted object locations and the manually annotated ones for the λ-th object at time k as

$$\mu_{k,\lambda} = |(\tilde{u}_{k,\lambda}, \tilde{v}_{k,\lambda}) - (u_{k,\lambda}, v_{k,\lambda})|_2^2, \tag{2.13}$$

where $(\tilde{u}_{k,\lambda}, \tilde{v}_{k,\lambda})$ and $(u_{k,\lambda}, v_{k,\lambda})$ are the point-wise annotation and estimation of the centre of the object and $|\cdot|_2$ indicates the l_2-norm.

2.6.4 Tracking

We use the CLEAR metrics [26, 83] for the evaluation of the tracking results since these are the standard in the MOT community. The measures are Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), False Alarm per Frame (FAF), Mostly Tracked objects (MT), Mostly Lost objects (ML) [102], Fragmented trajectories (Frag), False Positives (FP), False Negatives (FN) and Identity Switches (IDS). The measures quantify the discrepancy between the tracking results and the GT. MOTA is defined as

$$\text{MOTA} = 1 - \sum_{k=1}^{K} (\text{FN}_k + \text{FP}_k + \text{IDS}_k), \tag{2.14}$$

where FN_k, FP_k and IDS_k are the number of false negatives, false positives and identity switches for all objects at frame k, respectively. We consider the same definitions of FN and FP as in Sec. 2.6.1. And IDS is defined as the change in identity of a trajectory as defined by [94]. MOTP is defined as:

$$\text{MOTP} = \frac{1}{\sum_{k=1}^{K} c_k} \sum_{\lambda \in \Lambda_k} \sum_{k=1}^{K_\lambda} d_{k,\lambda}, \tag{2.15}$$

where c_k is the number of matches between the estimated state and the GT and $d_{k,\lambda}$ is the IOU between the estimated state, $\mathbf{x}_{k,\lambda}$, and its associated GT annotation, $\tilde{\mathbf{x}}_{k,\lambda}$, at time k. For further details of the tracking metrics, we refer to the reader to [26, 83, 94].

2.7 Discussion

This chapter overviewed existing works in the topic of object localisation and dimensions estimation, and object tracking from images and videos. Also, we presented the performance measures that are commonly used to evaluate these tasks.

Regarding object localisation and dimensions estimation, we found out that the majority of existing algorithms use either markers and a motion capture system or a full knowledge of the object (e.g. a 3D model) for performing these tasks. Existing solutions obtain accurate and efficient localisation when objects that are previously seen by the algorithms are used. However, the localisation of unseen objects, for which its prior knowledge is null or limited, remains a challenging and open research problem.

Regarding the tracking literature review, we showed the divisions in which tracking can be classified, focusing on the principal components of MOT and presenting some of the most relevant related works. MOT methods usually work in the image plane and have recently focused their efforts in improving the discriminative power of the appearance features. The performance of these methods would be seriously compromised in scenarios where the appearance of the objects is less reliable such as small resolution objects (e.g. objects from a high-altitude UAV) or very similar-looking objects (e.g. a group of ants or people with the same clothes). Occlusion handling is still a very challenging issue, that is even more difficult in online methods, and after numerous published works is still an open research problem. Besides, the motion of the camera is commonly obviated in tracking-by-detection approaches even though it is a very important feature to consider for robustly modelling the motion of the objects, specially in moving cameras. We focused on object motion prediction, which is an important component of trackers, that aim to estimate the future location of objects only by observing their previous location. We focused on existing algorithms that predict the motion of moving objects from both static and moving cameras. We showed that a large amount of predictors do not consider the camera motion which make them inaccurate when the camera is mounted on a moving platform such as a robot. This can potentially produce tracking drifts. Existing prediction models that consider the motion of the camera often rely on strong assumptions on the objects (e.g. objects lie on the same common plane), on the scene (e.g. scene planarity), or use additional sensors (e.g. depth cameras).

Chapter 3

Detection

In this chapter, we first describe a standard vision calibration procedure and we introduce an audio-visual calibration to enable the use of vision and audio signals in the same reference system (Sec. 3.1); then, we describe a novel method for object localisation and dimensions estimation (Sec. 3.2); next, we describe a multi-modal algorithm for human-to-robot handover of unseen objects (Sec. 3.3); besides, we introduce the CORSMAL containers dataset (Sec. 3.4); then, we show the experimental validation of the proposed methods (Sec. 3.5). Finally, we draw some conclusions (Sec. 3.6).

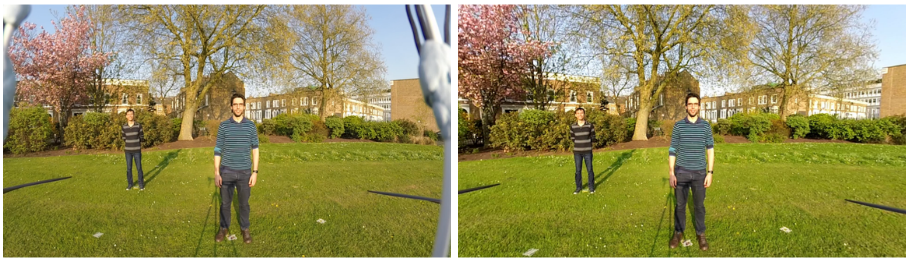
3.1 Calibration

Calibration is the procedure of preparing a device for its correct (or advanced) use. This section is divided in vision calibration (Sec. 3.1.1) and audio-visual calibration (Sec. 3.1.2).

3.1.1 Vision calibration

Vision calibration (also known as camera calibration) can be used to compensate for the deformation produced by the camera lens and to infer the real-world location of objects from the camera, Vision calibration is composed of two steps: intrinsic and extrinsic calibration. Intrinsic calibration refers to the estimation of a camera lens parameters (also known as intrinsic parameters or resectioning). We will employ these parameters for correcting the camera lens distortion and to determining the location of the camera in 3D.

To do so, we use an existing and standard camera calibration named pinhole camera model to



(a) (b)

Figure 3.1: Image captured by the camera mounted on the drone (a) before and (b) after image undistortion.

estimate the intrinsic and distortion parameters of the camera [67, 222]. We record a calibration video of a calibration board (e.g. checkerboard) at different locations and poses with respect to the camera and then estimate the camera parameters with standard libraries such as OpenCV or MATLAB. The radial and tangential lens distortion parameters are represented by ξ and the intrinsic matrix parameters are defined as

$$\mathbf{K} = \begin{bmatrix} F & s & u_0 \\ 0 & F & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

where F is the camera focal length (assuming squared pixels) measured in pixels, u_0 and v_0 indicate the location of the principal point (optical centre) in the image, and s is the skew axis coefficient.

The parameter ξ is used to undistort an image as

$$\widetilde{\mathbf{I}}_k = \mathcal{B}(\mathbf{I}_k, \xi), \tag{3.2}$$

where $\mathcal{B}(\cdot)$ represents the undistortion procedure [67], $\mathbf{I}_k$ and $\widetilde{\mathbf{I}}_k$ denote an image frame before and after undistortion, respectively. An example is given in Fig. 3.1 illustrating an image before and after the undistortion procedure. In the distorted image (Fig. 3.1a) we can observe a large distortion that is more clear on the corners on the image. It can be seen that in the undistorted image (Fig. 3.1b) the real-world lines maintain their linearity on the image thanks to the undistortion procedure.

The extrinsic calibration consists of estimating the 3D pose of the camera with respect to a reference system (e.g. the calibration board). The extrinsic parameters are composed of a rotation matrix and a translation vector that encodes the pose of the camera with respect to the reference. This can be used for 3D-2D projections as well as advanced multi-view applications. We obtain the extrinsic parameters, $\mathbf{E}$, using standard methods [64]. To do so, we record a calibration image $\mathbf{I}$ of the scene where a calibration board is visible. Then, we estimate the rotation and translation of the camera with respect to the the calibration board using standard libraries such as OpenCV or MATLAB.

3.1.2 Audio-visual calibration

The audio-visual calibration procedure consists of two steps: *audio-visual temporal alignment* to synchronise audio visual signals and *audio-visual geometrical alignment* that will align the estimations of both modalities. This allows to leverage both signals concurrently.

Let's have a sensing platform composed of a microphone array and a camera at the centre of the microphone array. The audio and video acquisition systems that work independently, where each one has its own processing unit, feature representation and coordinate system. We thus need to calibrate the microphone array and the camera so that the features from the audio and video streams can be jointly exploited.

The audio and video acquisition systems might work independently and start recording at different times. To jointly exploit audio and vision modalities it is needed to temporally align the signals. Assuming that the camera has an integrated microphone, we estimate the unknown time offset, δ_{av}, between the microphone array and the audio signal from the camera by manually matching (maximising the correlation between both signals) the signals with a calibration sound (e.g. clapping). This procedure can be also done using automatic methods [20].

Once both streams are temporally aligned, the geometrical alignment associates audio and video events in a unified coordinate system (Fig. 3.2). The 3D position P of a real-world object is projected on the image plane, where it is denoted as $\mathbf{p}$. Let θ_a and θ_v be the angles, on a 2D horizontal plane, of the object with respect to the microphone array and the camera. When an object emits a sound, its direction of arrival (DoA) can be estimated either from the microphone-array signals, θ_a, or from the visual signal, θ_v. Since the microphone array and the video camera have their own coordinate systems, to infer the DoA of the sound from the corresponding object in the image we need to know the relationship between θ_a and θ_v. In practice the centers of the microphone array O_M and the camera O_C are not perfectly aligned in Fig. 3.2.

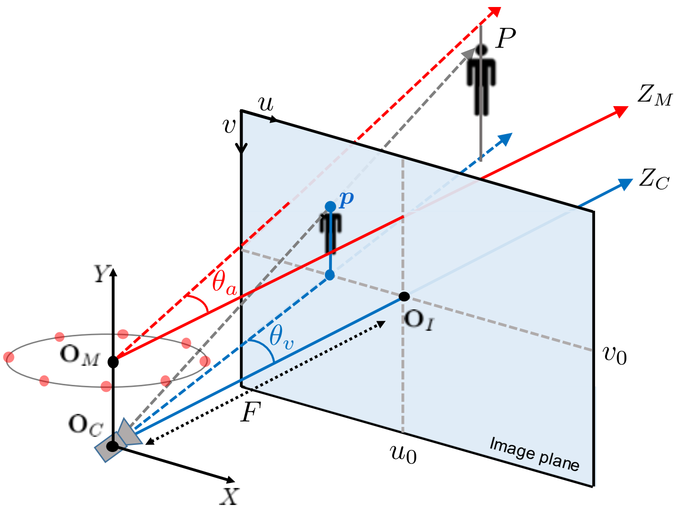


Figure 3.2: The 3D coordinate systems for the microphone array, (X,Y,Z_M), and the camera, (X,Y,Z_C). The centers of the microphone array and camera are $\mathbf{O}_M$ and $\mathbf{O}_C$, respectively; $\mathbf{O}_I = (u_0, v_0)$ is the principal point (centre) of the image; and F is the focal length of the camera. The sound source P is projected onto the image plane as $\mathbf{p}$ with visual angle θ_v. The audio angle from the sound source P to the array is θ_a.

We thus represent the relationship between θ_a and θ_v as

$$\theta_a = a_1\theta_v + a_2, \tag{3.3}$$

where $\mathbf{a} = [a_1, a_2]^{\mathrm{T}}$ are unknown constants. Note that we assume here that the sound source is co-located when perceived by visual and audio means (e.g. the voice of a person comes from their mouth), and that the environment generates negligible reverberations. To estimate a_1 and a_2, we do an audio-visual calibration recording where a speaker produces a sound at L different locations with both the microphone array and the camera while the drone is muted. As an example, let us use the sound from the location Q. For the audio, the DoA of the sound, θ_a^Q, can be estimated from the microphone signal with the SRP-PHAT algorithm [192]. For the video, we manually label the sound emitting point (i.e. the mouth of the speaker) in the image, $\mathbf{p}_Q = (u_Q, v_Q)$, and then estimate its DoA as

$$\theta_v^Q = \arctan\frac{u_Q}{F}. \tag{3.4}$$

We thus estimate a set of DoAs of the speaker from the audio as $\theta_a = [\theta_a^1, \ldots, \theta_a^L]^{\mathrm{T}}$ and from the video as $\theta_v = [\theta_v^1, \ldots, \theta_v^L]^{\mathrm{T}}$. The vector of parameters $\mathbf{a}$ is then estimated from θ_a and θ_v using least-square fitting.

Given the parameters ξ, $\mathbf{K}$ and $\mathbf{a}$, we can calibrate the audio and video signals, allowing their combined use. In this thesis, we use the described audio-visual calibration to allow the ground-truth annotation of the AVQ dataset (Sec. 4.4) and, therefore, to enable the evaluation of sound source tracking (Sec. 4.6.4). Furthermore, we use this calibration procedure for performing multi-modal localisation and speech enhancement of multiple sound sources from an unmanned aerial vehicle [152]. However, as this work has a strong component of audio processing, which lies outside the scope of this thesis, it is not described within the thesis. Please check the reference for further details on this work.

3.2 Multi-view object localisation and dimensions estimation

The 3D localisation of an object and the estimation of its properties, such as shape and dimensions, are challenging under varying degrees of transparency and lighting conditions of the objects. In this section, we propose LoDE (Localisation and Object Dimensions Estimator) a method for jointly localising container-like objects and estimating the dimensions of objects with transparent materials using two calibrated RGB cameras whose poses are known. Under the assumption of vertical circular symmetry, we estimate the dimensions of an object by sampling at different heights a set of sparse circumferences with iterative shape fitting and image re-projection to verify the sampling hypotheses in each camera using semantic segmentation masks.

3.2.1 Object localisation

LoDE localises the 3D centroid of the object from 2D centroids estimated from semantic segmentation masks. As many containers such as cups, drinking glasses and bottles have a circular symmetry along their vertical axis, LoDE is initialised with a set of circumferences sampled around the 3D centroid at different heights. Then, the algorithm iteratively fits to the object by reducing the radius of the circumferences until each circumference is verified within the object mask in each camera.

We propose an iterative algorithm to estimate the shape of an object and, as by-product, its dimensions, assuming the object to be circular symmetric with respect to its vertical axis. As also the location in 3D of the object is unknown, we combine multi-view projective geometry [64] with an iterative 3D-2D shape fitting to achieve the objective (see Fig. 3.3).

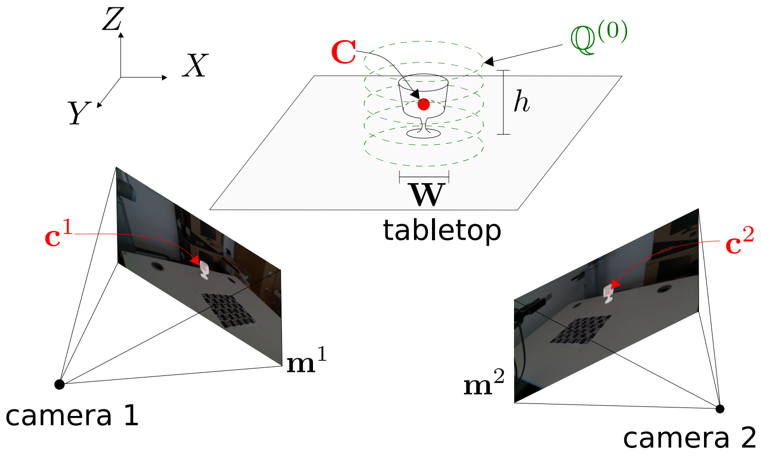


Figure 3.3: Two cameras capture an object from different viewpoints. Given only RGB images and camera poses, we estimate the width w and height h of the object without relying on 3D object models, depth information, or markers. The proposed method, LoDE, localises the object centroid in 3D, $\mathbf{X}$ from the 2D centroids, $\mathbf{x}^1$ and $\mathbf{x}^2$, estimated on the segmented images, $\mathbf{m}^1$ and $\mathbf{m}^2$, and then samples a set of sparse 3D points, $\mathbb{Q}^{(0)}$, belonging to circumferences centred at the centroid location and at different heights, to fit the object shape with an iterative 3D-2D algorithm.

Let $\mathbf{I}^c$, $c \in \{1,2\}$, represent the images (defined as in Eq, 1.2) from two calibrated cameras (see Sec. 3.1.1) that observe the object from different viewpoints. In this section we omit the time index k from the notation as the detection happens for a single image in a specific k. Let $\mathbf{E}^c$ be the 3D pose of each camera whose calibration is modelled by the intrinsic parameters $\mathbf{K}^c$, consisting of focal length and principal point.

We first detect the object in $\mathbf{I}^c$ with semantic segmentation,

$$\mathcal{D} : \mathbb{N}^{W \times H \times C} \to \{0,1\}^{W \times H}, \tag{3.5}$$

where $\mathbf{m}^c = \mathcal{D}(\mathbf{I}^c) \in \{0,1\}^{W,H}$ a binary feature map representing the segmented object.

We estimate the 2D centroid $\mathbf{c}^c$ of the segmented object with the intensity centroid method [160] through the definition of the moments within a local image area. Then, the centroid in 3D is computed by triangulating the two 2D centroids:

$$\mathbf{C} = \tau(\mathbf{c}^1, \mathbf{c}^2, \mathbf{E}^1, \mathbf{E}^2, \mathbf{K}^1, \mathbf{K}^2), \tag{3.6}$$

where $\tau(\cdot)$ is the triangulation operation [64].

3.2.2 Object dimensions estimation

To estimate the shape of the object, we initialise the algorithm with a cylindrical shape around its estimated 3D centroid that iteratively fits the object shape as observed by the cameras. Each iteration i creates L circumferences of radius $r^{(i)}$, centred at the estimated object 3D location $\mathbf{C}$ and with varying height z_l, $l = 1, \dots, L$. We represent the set of circumferences as

$$\mathbb{C}^{(i)} = \{(r_l^i, z_l, v_l)\}_{l=1}^{L}, \tag{3.7}$$

where $v_l \in \{0, 1\}$ indicates whether a circumference lies within the object mask of both cameras. For each circumference l, we also sample a set of N sparse 3D points,

$$\mathbb{Q}_l^{(i)} = \{\mathbf{Q}_{n,l}^{(i)} = (x_{n,l}, y_{n,l}, z_l)\}_{n=1}^{N}, \tag{3.8}$$

and the set of all 3D points is $\mathbb{Q}^{(i)} = \{\mathbb{Q}_l^{(i)}\}_{l=1:L}$. We project the 3D points onto the image of both cameras as

$$\mathbf{u}_{n,l}^{c} = \psi(\mathbf{Q}_{n,l}^{(i)}, \mathbf{E}^c, \mathbf{K}^c), \tag{3.9}$$

where $\psi(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the projection function [64]. Then, we verify if all the points belonging to circumference l, $\mathbb{Q}_l^{(i)}$, lie within the object mask of both cameras,

$$\eta = \sum_{n=1}^{N} \mathbf{m}^1(\mathbf{u}_{n,l}^1) + \mathbf{m}^2(\mathbf{u}_{n,l}^2), \tag{3.10}$$

and if the condition is satisfied (*i.e.* $\eta = 2N$), we set the corresponding flag as converged, *i.e.* $v_l = 1$.

For iteration $i+1$, we decrease the radius $r_l^{(i+1)}$ and re-create the 3D circumference points, $\mathbb{Q}^{(i+1)}$. Points with $v_l = 1$, are not re-sampled. This iterative 3D-2D shape fitting terminates when either all $v_l = 1$ or $r_l^{i+1} < \rho$, where ρ is the minimum radius that circumferences are created. Fig. 3.4 shows as example three iterations of the shape fitting for a transparent drinking glass. Additional visual results are available online[1].

When the shape fitting is finalised, we can estimate the dimensions of the object. We select

[1] Visual results of LoDE: `https://corsmal.eecs.qmul.ac.uk/CORSMAL_Containers_LoDE.html`

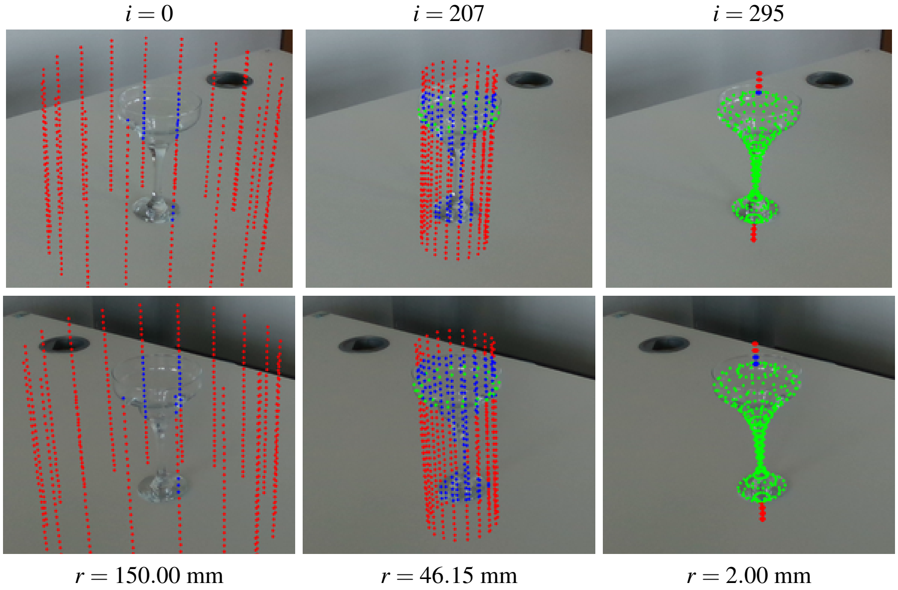


Figure 3.4: Initialisation, sampled iteration and convergence of the 3D-2D shape fitting of a drinking glass (top: left camera, bottom: right camera). Legend: i iteration number, r radius of the circumference, ● projected points lying outside the segmentation mask, ● projected points lying inside the segmentation mask and ● projected points whose circumference fits the shape of the object (inside the segmentation mask of both cameras).

among the valid circumferences $\mathbb{V} = \{(r_l, z_l, v_l) | v_l = 1\} \subset \mathbb{C}$, the one with the largest radius r^* and the ones with maximum and minimum heights, $\acute{z}$ and $\grave{z}$, respectively. The estimated object width is $\tilde{w} = 2r^*$ and the object height is $\tilde{h} = \acute{z} - \grave{z}$.

Note that while our method may resemble fitting approaches using Active Contour Models for segmenting images or surfaces in 3D via energy minimisation [37, 82, 175], LoDE fits the shape of an object in 3D exploiting already segmented images with a resampling-verification strategy.

3.3 Multi-modal human-to-robot handover of unseen objects

In this section, we propose a multi-modal baseline, based on the object localisation and dimensions estimation algorithm proposed in Sec. 3.2, to perform human-to-robot handovers of unseen objects with a real robotic arm.

The real-time estimation through vision of the physical properties of objects manipulated by humans is important to inform the control of robots for performing accurate and safe grasps of objects handed over by humans. However, estimating the 3D pose and dimensions of previously unseen objects using only RGB cameras is challenging due to illumination variations, reflective surfaces, transparencies, and occlusions caused both by the human and the robot. In this paper

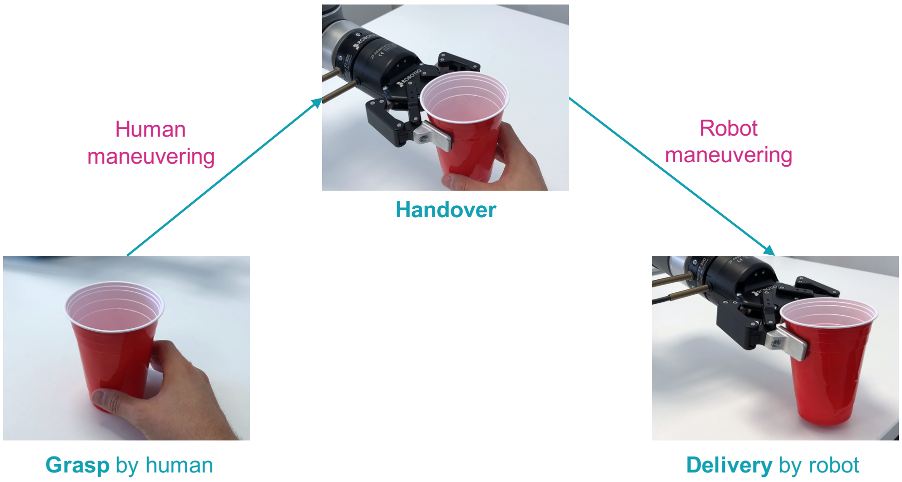


Figure 3.5: The main phases of the human-to-robot handover of a cup.

we present a benchmark for dynamic human-to-robot handovers that do not rely on a motion capture system, markers, or prior knowledge of specific objects.

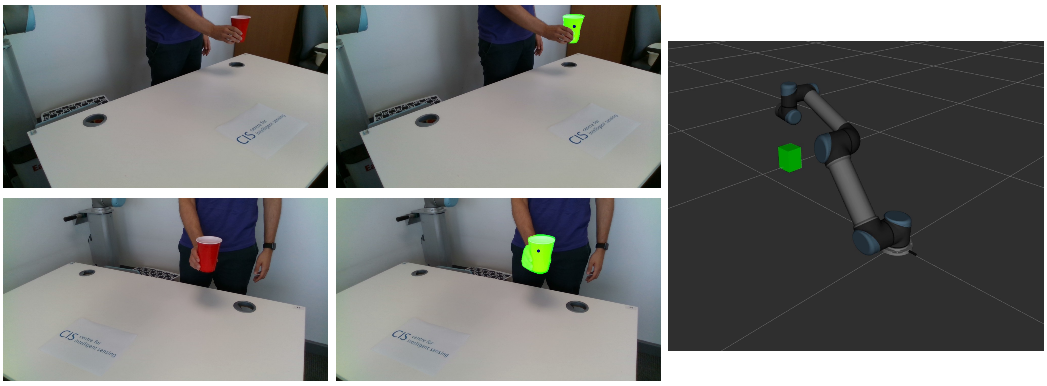
Humans manipulate daily objects made of a wide variety of materials, thus making safe and accurate handover to robots a challenging task, especially for previously unseen objects whose dimensions, stiffness, and weight are unknown. When humans are about to receive previously unseen objects from others, they first estimate the properties of the objects through vision and then use tactile and force feedback to improve this estimation. Even though the manipulation capabilities of robots exceed those of humans in terms of accuracy, their capabilities in terms of perception and dexterity still fall way behind those of humans. Open challenges include accurately estimating the object pose while the human moves the object, selecting the most appropriate grasping regions that will not harm the human, and predicting the filling of containers, such as boxes and cups, to estimate their mass and stiffness.

A fluid and efficient human-to-robot handover is the result of the combination of perception and control [119]. To this end, the pose of the object to hand over has to be tracked through vision to facilitate the planning and reaching of the predicted handover location, and the subsequent grasping and delivering of the object to a desired location.

Next we introduce a multi-modal baseline that observe the manipulation of an object by a human and infers how to safely perform human-to-robot handovers of unseen objects (Fig. 3.5). Note that the baseline has no prior information about the specific object used for the task.

We localise and estimate the dimensions of the object as described in Sec. 3.5.

Also, to deal with changes in appearance and to produce a smooth trajectory of the centroid



(a) (b) (c)

Figure 3.6: Sample segmentation and tracking result: (a) input frames (top: Camera 1, bottom: Camera 2); (b) 2D tracking result (green mask) and re-projected centroid (blue point); (c) visualisation of the 3D location and dimensions of the cup (green) estimated through vision.

for the control of the robotic subsystem, we track the cup over time [193] with a state-of-the-art SOT, $\mathcal{T}(\cdot)$, that operates in each camera independently and is automatically initialised with the smallest up-right rectangle containing all pixels of the mask $\mathbf{m}_0^c$, generated by the semantic segmentation used in Sec. 3.5. We remind that c indicates the camera index. Similarly to the semantic segmentation, the tracker produces for each frame a binary mask for the object, $\mathbf{m}_k^c = \mathcal{T}(\mathbf{m}_{k-1}^c, \mathbf{I}_k^c)$, which we use to estimate the centroid of the cup in 3D with the intensity centroid method [160] and triangulation (Eq. 3.6). Fig. 3.6b shows two sample masks produced by the tracker for Camera 1 and Camera 2 in S2 and the projected 2D centroids.

The baseline is instantiated in two different setups, S1 and S2, in different laboratories. The vision baseline is the same in both setups. Regarding the robotic control, in S1, we employ a task-space control of the robotic manipulator, and prediction/inference of the human intention, and in S2 we use a simpler robotic control with standard motion planning libraries[2]. Please refer to [J1] for further description on the robotic control.

3.4 The CORSMAL containers dataset

We collect a dataset[3] composed of images using 23 containers for liquids: 5 cups, 9 drinking glasses and 9 bottles (see Fig. 3.7). These objects are made of plastic, glass or paper, with different degrees of transparency and arbitrary shapes. The dataset contains 3 objects that do not have circular symmetry, *e.g.* object 6 (diamond glass), object 16 (amaretto bottle) and object 20

[2] MoveIt: `https://moveit.ros.org/`

[3] The CORSMAL containers dataset: `https://corsmal.eecs.qmul.ac.uk/CORSMAL_Containers.html`

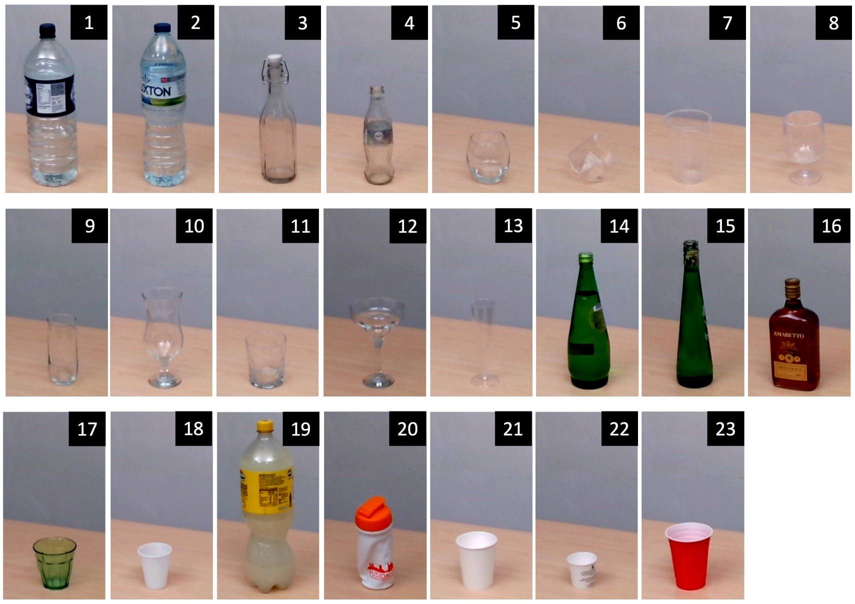


Figure 3.7: Objects in the CORSMAL container dataset. Objects 1 to 13 (transparent); 14 to 18 (translucent); 19 to 23 (opaque). Note that crops are taken from images acquired with the same camera view.

(deformed water-bottle).

We placed each object on a table and we acquired RGB, depth and stereo infrared (IR) images (1280×720 pixels) with two Intel RealSense D435i cameras, located approximately 40 cm from the object. RGB and depth images are spatially aligned. The cameras are calibrated and localised with respect to a calibration board. We acquired the images in two room setups to vary the lighting and background conditions. The first setup is an *office* with natural light from a window and objects placed on a table of size 160x80 cm and height 82 cm. The second setup is a *studio*-like room with no windows, where we used either ceiling lights or artificial studio-like lights to illuminate a table of size 60x60 cm and height 82 cm.

To acquire multiple images of the same object under different backgrounds, we capture data with the tabletop uncovered and then covered with two different tablecloths. We collected in total 207 configurations that are combinations of objects (23), backgrounds (3) and lighting conditions (3), resulting in 414 RGB images, 414 depth images and 828 IR images. We manually annotated the maximum width and height of each object with a digital calliper (0-150 mm, ±0.01 mm) and a measuring tape (0-10 m, ±0.001 m).

3.5 Experimental validation

In this section, we first evaluate multi-view object detection and dimensions estimation (Sec. 3.5.1) and then we show the validation of the multi-modal human-to-robot handover of unseen objects method (Sec. 3.5.2).

3.5.1 Multi-view object localisation and dimensions estimation

We validate the approach in the CORSMAL containers dataset that contains objects with different shapes and degrees of transparency, under varying lighting conditions and backgrounds (Sec. 3.4). The objects contained in the dataset are shown in Fig. 3.7. To quantify the success in localising the objects we employ the recall metric (Eq. 2.9) and for the accuracy of the estimations we compute the absolute error when estimating the largest width of the object (Eq. 2.11) and its height (Eq. 2.12).

We compare LoDE with a state-of-the-art method and two baselines, which do not require 3D object models and can estimate object dimensions: NOCS [187], a DNN-based approach that uses RGB-D data; a baseline that uses segmentation on RGB-D data (SegDD) and our approach applied to a stereo infrared camera with narrow-baseline on a single device (LoDE-IR). SegDD partially replicates the initial part of several existing DNN-based approaches, by using semantic segmentation and then back-projecting in 3D the pixels belonging to the object of interest, using the distance estimation of the depth image. The dimensions of the object are estimated from the most external points along the x-axis and y-axis, respectively (camera coordinate system). Note that while LoDE is multi-view, NOCS, SegDD and LoDE-IR are single-view. Thus, we report the results of single-view methods as the concatenation of the results from the two cameras used in the setup. We do not compare with other approaches for 6 DoF pose estimation, (*e.g.* DenseFusion [185]), or 3D Object Detection, (*e.g.* FrustumNet [150]), as they require the exact 3D model of each object which is not the case of study of this work.

Regarding the implementation details. For the semantic segmentation, SegDD, LoDE-IR and LoDE adopts Mask-RCNN [65] trained on the Common Objects in Context dataset [105] of which we consider the classes *cup*, *wine glass*, *bottle* and *vase*. For both LoDE-IR and LoDE, we set $L = 500$ circumferences, separated by 1 mm on height and composed of $N = 20$ points each (18° between point pairs) and we sample the radius of the circumferences, r, across iterations with the following schedule: $150.0, 149.5, \dots, 1.5, \rho$ (mm), with a minimum circumference

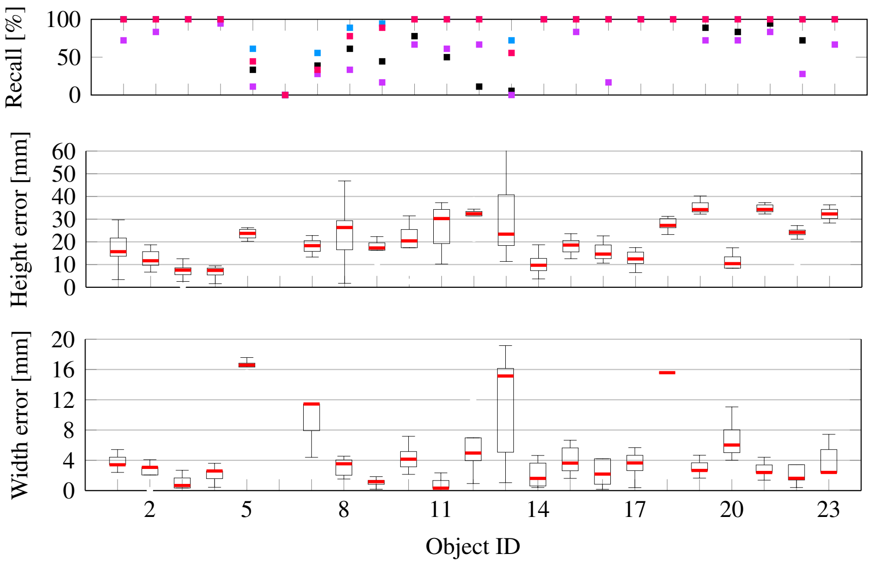


Figure 3.8: Detection recall (R) of all methods and errors for each dimension using LoDE for each object of the CORSMAL container dataset, across all backgrounds and lighting conditions. Note the different scale of the y-axis. Legend: NOCS [187] ■, SegDD 🟦, LoDE-IR 🟪 and LoDE 🟥.

radius of $\rho = 1.0$ mm to fit the shape of objects that have a thin stem, (*e.g.* object 12, margarita glass, or object 8, plastic wine glass).

Fig. 3.8 shows the statistics (median, min, max, 25 percentile and 75 percentile) of the dimensions error of our approach for each object across all the background and lighting variations. LoDE accurately estimates the width of most of the objects with an error less than 20 mm and with small variations across the configurations. Objects 5 (juice glass), 7 (beer cup), 13 (champagne flute) and 18 (small white cup) are the least accurate cases, where the median error is larger than 10 mm. LoDE is less accurate in estimating the object height with the errors varying between $\sim$10 mm and $\sim$40 mm. This larger inaccuracy is due to the perspective on the image plane, as circumferences at lower/higher height than the real one are re-sampled with smaller radius to fit within the object masks. Objects 1 (bottle of water), 8 (plastic wine glass), 11 (rum glass) and 13 (champagne flute) show larger variations across configurations than other objects. As width and height are estimated independently, there is no correlation between the two dimensions. While LoDE localises most of the objects across all the configurations (100% Recall), there are some challenging cases, such as objects 5 (juice glass), 7 (beer cup) and 13 (champagne flute), where the Recall is below 60%. Note that the champagne flute is not localised by NOCS and LoDE-IR. As expected, the most challenging case for all methods is object 6 (diamond glass) that is never detected through the semantic segmentation due to the high level of transparency

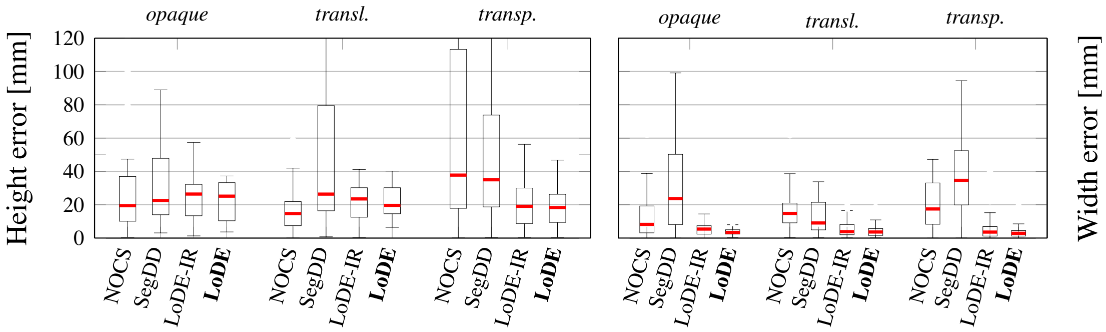


Figure 3.9: Estimation error of height and width for opaque, translucent and transparent objects.

and the unusual shape. Moreover, NOCS and LoDE-IR have lower Recall than LoDE for most of the transparent glasses/cups (*e.g.* objects 5 to 13) and the small cups (objects 18 and 22).

Fig. 3.9 compares the methods under varying degrees of transparency, such as opaque, translucent and transparent. The error is computed only for the cases where the object is successfully localised. As previously observed for LoDE, we can observe even here that all methods estimate the width more accurately than the height. The top-down perspective of the cameras makes the segmentation treat different parts of the object as one and consequently affects the height estimation when back-projecting in 3D via depth map or triangulation, or projecting for circumference verification. SegDD is more inaccurate for both translucent and transparent objects, with large variations especially in the height, due to the inaccuracies of the depth maps, while NOCS is mostly sensitive to transparent objects when localised. LoDE-IR and LoDE, instead, estimate the dimensions with a median error less than 30 mm despite the object transparency. However, NOCS and SegDD are more accurate in estimating the height for opaque objects.

Fig. 3.10 shows the Recall in localising the objects and the error in estimating the height and width dimensions, across all the configurations. As previously observed, LoDE outperforms NOCS and SegDD obtaining 2.6 mm and 10.6 mm more accurate height estimations, and 11.2 mm and 22.9 mm more accurate width estimations comparing their medians, respectively, with a smaller standard deviation. LoDE also outperforms LoDE-IR in both height and width estimations; furthermore, LoDE has a 25% detection recall higher than LoDE-IR at similar dimension error. Although both LoDE and SegDD uses Mask-RCNN, the detection recall of LoDE is slightly lower than SegDD, as LoDE considers the two views simultaneously, while SegDD works on each view individually.

Fig. 3.5.1 compares the results for one opaque and one transparent cup, one opaque and one

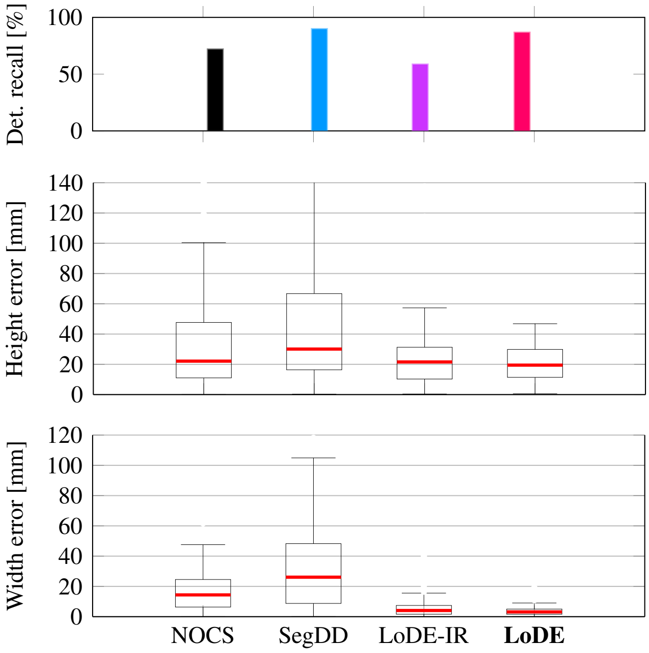


Figure 3.10: Detection recall and dimension estimation error. Legend: NOCS [187] ⬛, SegDD 🟦, LoDE-IR 🟪 and LoDE 🟥.

translucent bottle, and two transparent drinking glasses under different backgrounds and lighting conditions. All methods accurately estimate the dimensions of the opaque cup (object 22). While SegDD, LoDE-IR and LoDE fails to localise object 5 (juice glass) under natural lighting, NOCS estimates an inaccurate bounding box. However, NOCS fails to localise two transparent objects (objects 7 and 12). SegDD shows large inaccuracies for object 12 (margarita glass), object 7 (beer cup), and object 20 (deformed bottle), while LoDE-IR fails for object 20 and object 15 (translucent bottle). LoDE obtains less accurate results with non-symmetric objects (*e.g.* object 20) and under challenging lighting (last three columns), but successfully estimates transparent objects such as object 12 (margarita glass).

3.5.2 Multi-modal human-to-robot handover of unseen objects

We validate the baseline by performing a set of handovers with different objects, fillings, subjects, grasp types and handover types.

The setup consists of a robotic arm with a gripper, a table, cups and filling, and two cameras. The setup in S1 uses a KUKA LBR iiwa 7-DOF manipulator (14 kg payload)[4] and a task-space control of the robotic manipulator, and prediction/inference of the human intention. The setup in

[4] `https://www.kuka.com/en-ch/products/robotics-systems/industrial-robots/lbr-iiwa`

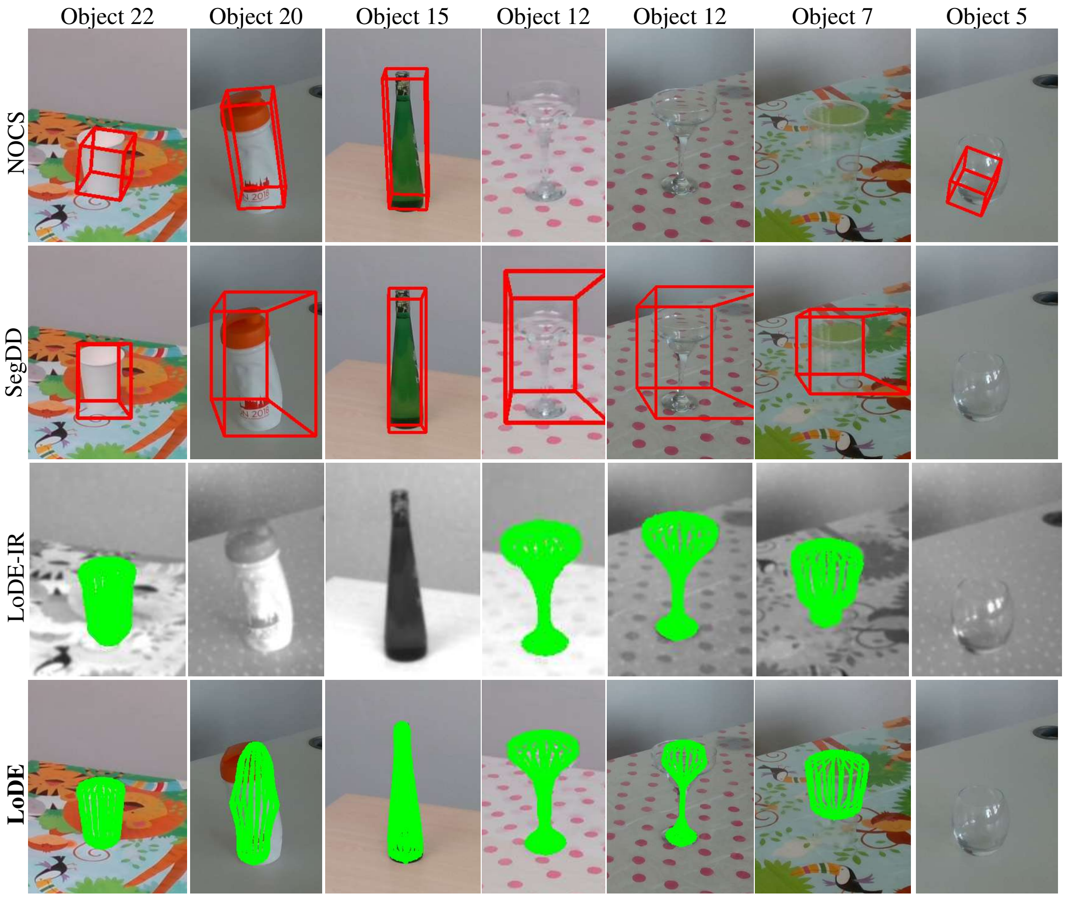


Figure 3.11: Sample results for objects with varying transparency, backgrounds and lighting. Fourth and fifth columns correspond to the same object and background but different lighting (artificial and natural, respectively).

S2 uses a UR5 6-DOF manipulator (5 kg payload)[5] and a simpler robotic control with standard motion planning libraries[6]. Both robots are equipped with a Robotiq 2F-85 2-finger gripper[7]. Both setups use two cameras (Intel RealSense D435), located at 40 cm from the arm. The cameras view the centre of the table and record RGB sequences at 30 Hz with a resolution of 1280×720. The cameras are synchronised, calibrated and localised with respect to a calibration board. Fig. 3.6a shows an example of the RGB images captured from setup S2. The baseline estimates the centroid and the dimensions of the cup in 3D at 18 Hz on an Intel i7-7700 CPU @ 3.60GHz, 16GB RAM, and a GeForce GTX 1060 6GB GPU.

Subjects are instructed to handover the cup naturally, standing opposite to the robot across the table, which should be covered with a white tablecloth. The subject grasps the cup from a pre-defined location at the centre of the table and carries the cup to one of the three approximate

[5] https://www.universal-robots.com/products/ur5-robot/

[6] MoveIt https://moveit.ros.org/

[7] https://robotiq.com/products/2f85-140-adaptive-robot-gripper

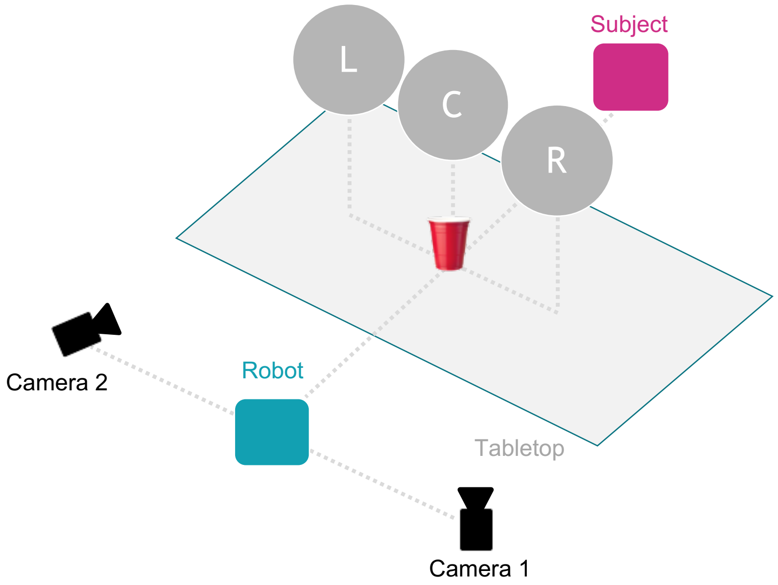


Figure 3.12: The setup for the benchmark and the three handover locations (left, L; centre, C; right, R), which are defined to be reachable by the robotic arm and comfortable for the subjects.

handover locations above the table: left, centre, and right with respect to the robot (Fig. 3.12). As one cannot explicitly set these locations, assuming the subjects are executing the handover naturally (i.e. with no intention to help or make it difficult for the robot), these locations can be roughly constrained by the reachability of the arm (i.e. $40\% - 50\%$), such that they are reachable by the robotic arm, but also comfortable for the subjects to perform the handover naturally. After the handover, the robot places the object on the table at a pre-defined location.

We select four objects that are challenging for both perception and robotics due to the high variability of their physical properties (e.g. material, shape, texture and mass). Specifically, we select four drinking cups (see Fig. 3.13): Cup 1 with high deformability and medium transparency; Cup 2 with average deformability and low transparency; Cup 3 with average deformability and high transparency; and Cup 4, the plastic wine glass from the YCB object database [36], which is not deformable and has high transparency. Moreover, we vary the mass and stiffness of each cup by filling it with rice, which is easy to purchase and - unlike liquids - harmless for the hardware in case of spilling. Table 3.1 summarises the properties of the cups and their filling.

Each subject shall perform the handover with three different grasps in random order: Grasp 1 from the bottom of the cup, Grasp 2 from the top, and Grasp 3 naturally. While Grasps 1 and 2 make it easier for the robot to grasp the cup, in Grasp 3 the hand of the subject may cover most of the surface of the cup, considerably reducing the candidate grasping regions (see Fig. 3.13).

In summary, in the experiments presented in this chapter, we account for 192 *configurations*

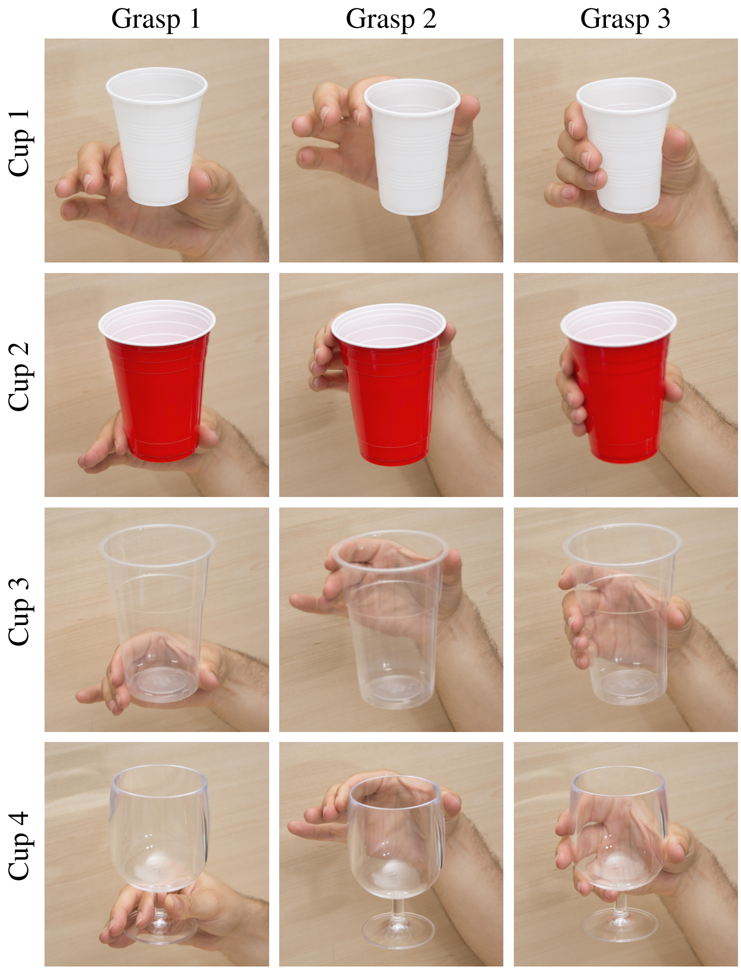


Figure 3.13: The cup and human-grasp types for the benchmark.

Table 3.1: The properties of the four cup types and the filling amount for the benchmark.

Properties	**Unit**	**Cup 1**	**Cup 2**	**Cup 3**	**Cup 4**
Deformability	-	High	Medium	Medium	None
Transparency	-	Medium	Low	High	High
Width at the top	cm	7.2	9.7	9.9	8.0
Width at the bottom	cm	4.3	6.1	6.5	6.5
Height	cm	8.2	12.1	13.6	13.5
Weight	g	2.0	10.0	9.0	134.0
Volume	ml	179.0	497.0	605.0	354.0
Filling amount	ml	125.0	400.0	450.0	300.0

in S1 and 48 in S2, that are decomposed as: 4 cup types, 2 filling amounts, 4 subjects (S1) and 1 subject (S2), 2 grasp types (grasp 3 is discarded as our baseline cannot deal with such a case), and 3 handover locations.

We quantify the performance of the vision subsystem as the AE of the estimation of the physical properties of the object that include object width at top and bottom and object height. We consider the distance between the position of the centre of the base of the cup at the end of the

Table 3.2: Vision subsystem results. Values indicate mean and standard deviation of the absolute error across 192 configurations in Setup 1 (S1) and 48 different configurations in Setup 2 (S2). Results for delivery accuracy are only computed when the handover has been successfully completed.

Measure	Units	Setup	
		S1	S2
Width at top	mm	9.12 ± 6.25	4.54 ± 5.20
Width at bottom	mm	9.60 ± 9.49	4.42 ± 5.73
Height	mm	20.53 ± 24.05	15.44 ± 1.95
Delivery	cm	2.88 ± 1.51	2.88 ± 1.30

task with respect to the initial position, to evaluate the detection accuracy as well as the overall completion of the task. We propose further measures to further evaluate the vision and robotic subsystems, and the completion of the overall task. However, we do not report them here as they lie outside of the scope of this thesis. For further details, please check [J1].

The results of the vision subsystem are shown in Table 3.2. We observe an estimation of the widths below 9.6 mm. For the cups' height the mean error is of 20.5 mm (in S1) and 15.44 mm (in S2).

Regarding the whole human-to-robot handover task, the proposed multi-modal system obtains a success rate of 75.0% in S1 and 72.9% in S2, where the success rate is defined as the ratio between successful handovers and the total number of configurations. A successful handover is considered when the cup is grasp by the robot and delivered back to the table in an upright position.

Fig. 3.14 shows aggregated scores[8] by cup, filling, and grasp type as well as handover location. Unlike Grasp 1 and Grasp 2, the baseline does not allow the performance of safe handovers when the subject holds the cup with a natural grasp (Grasp 3). This is due to the large size of the gripper we use: advanced perception algorithms should estimate the pose of the cup and human hand to detect suitable grasp regions to inform grasp planning. Note also that the performance is similar for the three handover locations, but varies for different cups, with the plastic wine glass (Cup 4) being the most challenging. Overall, the scores suggest that there is substantial room for improvement in developing perception and control algorithms to perform seamless human-to-robot handovers.

Fig. 3.15 shows two successful and one unsuccessful handovers with different grasp types

[8] The definition of the scores is available in [J1]

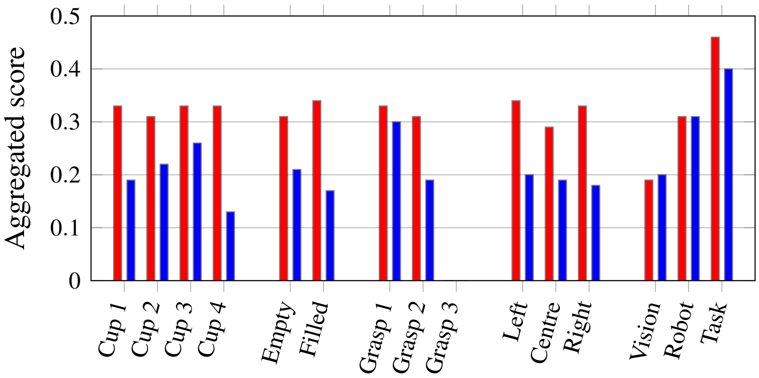


Figure 3.14: Scores averaged by cup, fillings, and grasp type; handover location; and subsystem for setup S1 (🟥) and setup S2 (🟦).

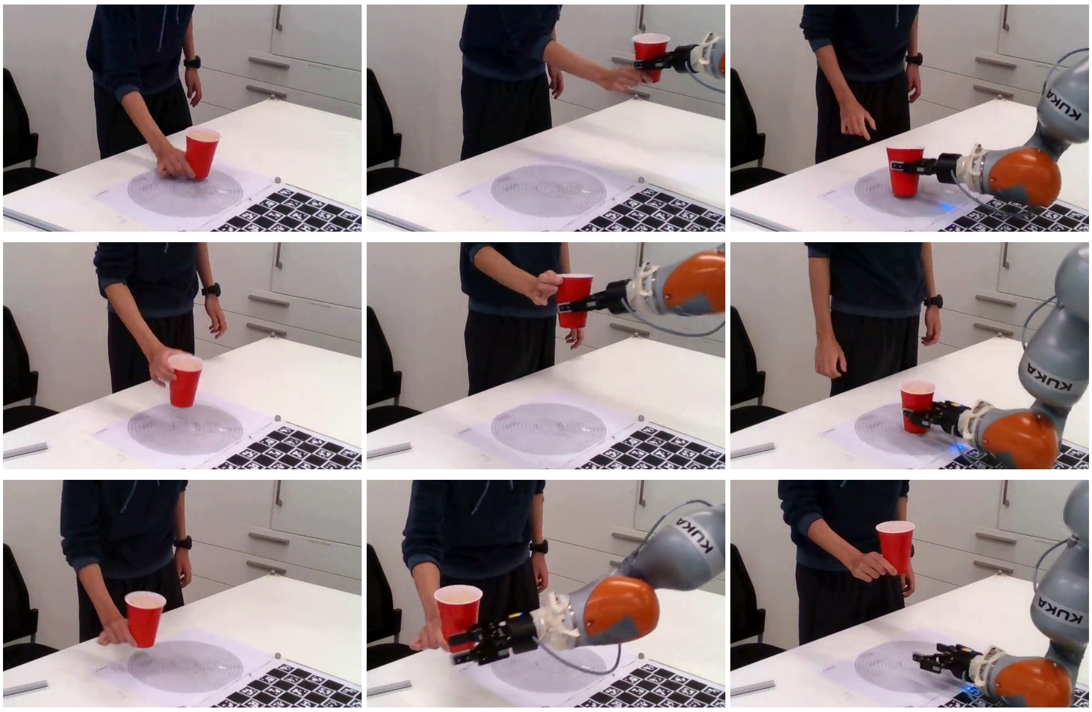

(a) (b) (c)

Figure 3.15: Sample handovers from setup S1, viewed from Camera 1, for Cup 2, full with Grasp 1 (first row), with Grasp 2 (second row), and with Grasp 1 (third row). (a) Human manipulation. (b) Handover. (c) Robot manipulation (third row: a failed handover).

and handover locations. Visual results of handovers[9] and the multi-modal baseline code[10] are available available.

[9] Handover visual results: `http://corsmal.eecs.qmul.ac.uk/benchmark.html#baseline`

[10] Baseline code: `https://github.com/CORSMAL/Benchmark`

3.6 Summary

This chapter presented an audio-visual calibration procedure and a method for localising and estimating the dimensions of unseen objects. First, we introduced an audio-visual calibration procedure that enables the combined use of audio and visual clues for different tasks such as ground-truth annotation of sound sources (from video) or the concurrent use of audio-visual signals for performing tasks such as localisation of sound sources or audio enhancement [152]. Also, we proposed LoDE, a method for localising and estimating the dimensions of container-like objects with circular symmetric shape, without relying on depth information, markers, or 3D models. LoDE uses an iterative multi-view 3D-2D shape fitting algorithm, verifying the model on the object image masks of two wide-baseline cameras. We also collected a dataset of containers-like objects with different degrees of transparency, and under varying lighting conditions and backgrounds. Experiments showed that LoDE has an object localisation success ratio of 86.96% and an average error less than 2 cm when measuring object height and under 0.5 cm when measuring the width of an object. However, this method is not suitable for objects that are not symmetric with respect to their vertical axis.

In addition to that, we presented a multi-modal baseline composed of a vision and a robotic subsystem for performing human-to-robot handover of unseen objects. Specifically, we combined semantic segmentation with object tracking in a two-stage algorithm that first estimates the 3D object location using multi-view projective geometry and, then, it estimates the height and width of the object with an iterative algorithm. Moreover, we proposed a benchmark[9] in [J1] to evaluate dynamic human-to-robot handovers in scenarios without motion capture systems, markers, or prior object models. We considered previously unseen objects (drinking cups), whose physical properties are subject to transformations, such as deformability due to the grasp, or different stiffness and filling amounts. We hope that the release of the benchmark, as well as the baseline code will encourage the community to participate in this benchmarking effort for creating advanced multi-modal algorithms for this task.

The work presented in this Chapter has been conducted in collaboration with multiple researchers: Sections 3.2, 3.4, and 3.5.1 with A. Xompero, and A. Modas; and Sections 3.3, and 3.5.2 with K. Chatzilygeroudis, A. Modas, N. Ferreira Duarte, and A. Xompero. I acknowledge that I am the principal contributor of the work presented in this Chapter and that is not present in any prior thesis.

Chapter 4

Tracking

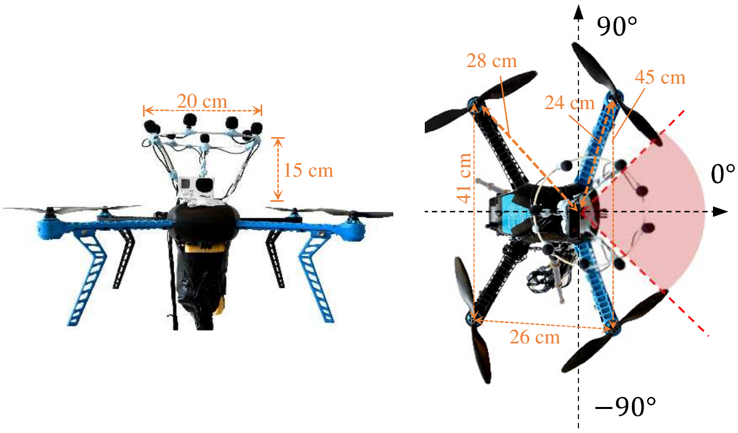
In this chapter, we first propose a method to perform single-object tracking from sound sources (Sec. 4.1); next, we introduce a method to perform multi-object tracking from a static camera (Sec. 4.2) where we include a perspective-dependent approach to perform object motion prediction; then, we present a multi-object tracker for moving cameras (Sec. 4.3) where we introduce a global-motion aware object motion prediction. Besides, we introduce a novel self-collected dataset for evaluation of SOT from audio signals (Sec. 4.4); then, we experimentally validate the proposed methods (Sec. 4.6); and, finally, we draw some conclusion (Sec. 4.7). Note that the tracking methodologies described in this chapter are based on the PHD-PF introduced in Sec. 2.5.

4.1 Single-object tracking

In this section, we perform single-object tracking of a sound source from a UAV equipped with a microphone array and a camera (see Fig. 4.1) using only audio signals. Note that the camera is only used to annotate the ground truth source location.

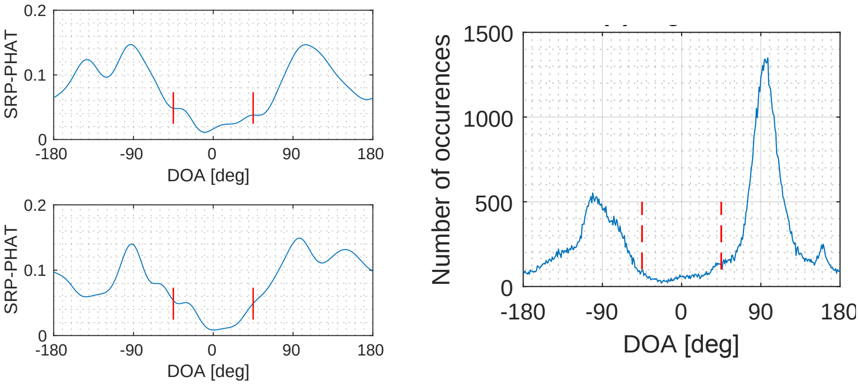
We employ existing audio processing techniques [190] to generate estimations (i.e. detections) of the DoA of the sound source. The audio processing generates a spatial confidence function, $\rho_k(\theta)$, that contains multiple noisy peaks corresponding to the sound source as well as a strong ego-noise produce by the UAV. Selecting the location with the highest intensity, as done in [189], may lead to erroneous results. We solve this problem with two steps: peak detection and tracking.

The ego-noise mainly consists of the sound emitted from the motors and the propellers. The



(a) (b)

Figure 4.1: The multi-rotor drone with an 8-microphone circular array and, for tracking performance evaluation, a camera mounted at the centre of the array. (a) Front view and (b) top view. The noiseless sector is indicated with a red shadowed area.



(a) (b)

Figure 4.2: Localisation results using ego-noise only. (a) SRP-PHAT functions at two random frames. (b) Histogram of the DoA estimates at individual time-frequency bins for a 30-second ego-noise segment. The noiseless sector [-45°, 45°] is indicated with red lines.

motor sound can be interpreted as point sources whose directions are static with respect to the position of the microphones. The ego-noise is generated by the motor, as well as the rotating blades. When the microphone array is placed at the front of the body of the drone (see Fig. 4.1a), the ego-noise tends to arrive from the side closer to the motors (the back of the array) thus creating a sector with lower ego-noise (the front of the array). Fig. 4.2a shows the SRP-PHAT functions computed at two random frames (each 2048-sample long) [192], where four peaks, corresponding to the four motors, can be observed. Fig. 4.2b shows the histogram of the local DoA estimation at individual time-frequency bins [189]. The histogram has lower values in the sector $[-45°, 45°]$. We name this sector, where we presume that an object sound can be more easily detected, as the *noiseless sector* [190].

The sound source direction is considered to be the peak with highest confidence in the noiseless sector and it is calculated as

$$\theta_k = \underset{\theta \in [\text{-}45^\circ, 45^\circ]}{\arg\max}\, \rho_k(\theta), \tag{4.1}$$

where $\rho_k(\theta)$ is the spatial confidence function of time block k. To track θ_k while filtering out the noisy spatial confidence function received at each block k, we use a particle filter [17, 181]. Note that we consider in this section single object tracking, therefore no data association is needed for retrieving the object identities. As in Eq 1.4, the i-th particle state is defined as

$$\dot{\theta}_k^i = (\theta, \bar{\theta})^T, \tag{4.2}$$

where θ is the estimated sound direction and $\bar{\theta}$ is the angular velocity at block k. Unlike in multiple-object tracking, we do not write in the notation the identity as only one identity is considered. Each particle has an associated weight, π_k^i, that informs how well a particle represents the actual location of the object. As previously discussed, the particle filter typically consists of four steps: prediction, update, state estimation and resampling.

We define the prediction model as

$$\dot{\theta}_k^i = G_k \dot{\theta}_{k-1}^i + \mathbf{N}_k, \tag{4.3}$$

where G_k is an affine transformation defined as

$$G_k = \begin{pmatrix} 1 & \Delta_k \\ 0 & 1 \end{pmatrix}, \tag{4.4}$$

where Δ_k is the time difference between audio blocks; and, $\mathbf{N}_b$ is a noise process defined as

$$\mathbf{N}_k = \begin{pmatrix} \mathcal{N}(0;\sigma) \\ \mathcal{N}(0;\bar{\sigma}) \end{pmatrix}, \tag{4.5}$$

where $\mathcal{N}(0;\sigma)$ and $\mathcal{N}(0;\bar{\sigma})$ are Gaussian functions with mean 0 and standard deviation σ and $\bar{\sigma}$, respectively.

Given the observed sound source direction (Eq. 4.1), the update step calculates the weights of the particles as

$$\pi_k^i = \frac{1}{\sqrt{2\pi}\sigma_u} \mathrm{e}^{-\frac{\left(\theta_k - \dot{\theta}_k^i\right)^2}{2\sigma_u^2}}, \tag{4.6}$$

where σ_u is the standard deviation that accounts for the sensing noise. Next, the state estimation step calculates the DoA of the sound as

$$\theta_k = \sum_{i=1}^{L} \pi_k^i \cdot \dot{\theta}_k^i, \tag{4.7}$$

where L is the number of particles.

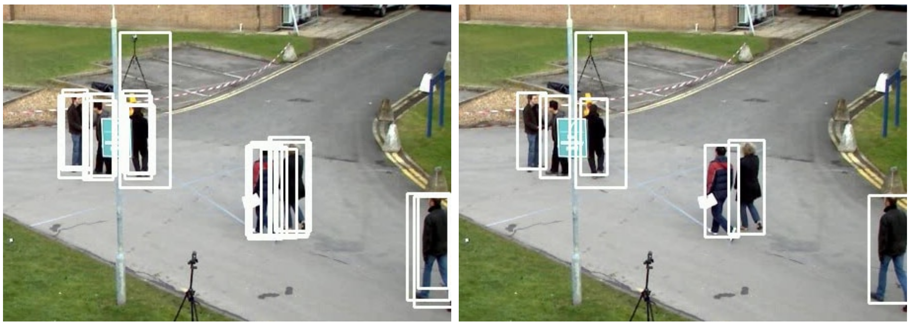
Finally, a resampling step discards particles with very low weight and duplicates particles with higher weight [115]. Therefore, the tracker outputs the estimated DoA for each of K time blocks, giving $(\theta_1, \ldots, \theta_k, \ldots, \theta_K)$.

4.2 Multi-object tracking from a static camera

As discussed in Sec. 2.5, PHD-PF does not assign identity information to the estimated states. Therefore, to work with multiple objects simultaneously we introduce in this section the Early Association Probability Hypothesis Density Particle Filter (EA-PHD-PF) that based on the PHD-PF introduces a mechanism, named Early association (EA), to perform tracking of a time-varying and unknown number of multiple objects within the PHD-PF framework without relying in clustering techniques.

4.2.1 Multi-detector fusion

Let's consider an *over-detection* process in which a (large) set of detections $\mathbb{Z}_k^*$ is generated at each time step k. This set of detections can be generated by running multiple detectors in parallel, by changing the operational parameters of a detector, or by a combination of these approaches. The set of detections is likely to generate multiple overlapping detections for a single object, potentially contain false positives, and likely to not contain false negatives (Fig. 4.3a). In order to manipulate detections generated by different detectors, whose confidences might be defined in different ranges, we normalise the confidences to $[0,1]$ range in a pre-processing step. We divide the confidence of each detection by the 99^{th} percentile of the confidences of all detections generated by the considered detector in a training dataset and, then, truncate the confidences to 1.



(a) (b)

Figure 4.3: Example of multi-detector fusion at frame 43 (crop) in PETS09-S2L1. (a) Initial object detections $\mathbb{Z}_k^*$; (b) fused detections $\mathbb{Z}_k$. Source images from [94].

Algorithm 1 Non-maximum suppression [44, 54]. Magenta indicates differences with respect to Algorithm 2.

```
Require: $(\mathbb{Z}_k^*, \mathbb{S}_k^*)$   ▷ Input tuple of detections and associated confidences sets
  $\mathbb{Z}_k \Leftarrow \emptyset$   ▷ Initialise output set of detections as an empty set
  $\mathbb{S}_k \Leftarrow \emptyset$   ▷ Initialise output set of associated confidences as an empty set
  if $|\mathbb{Z}_k^*| > 0$ then
    $(\mathbf{z}_k^{i*}, \mathbf{s}_k^{i*})_{i=1}^{|\mathbb{Z}_k^*|} \Leftarrow o((\mathbb{Z}_k^*, \mathbb{S}_k^*))$   ▷ Sort detections by decreasing confidence
    $\mathbb{I} \Leftarrow \{1, \ldots, |\mathbb{Z}_k^*|\}$   ▷ Initialise set of detection indexes
    while $|\mathbb{I}| > 0$ do   ▷ Iterate while $\mathbb{I}$ contains indices
      $i \Leftarrow \min(\mathbb{I})$   ▷ Select the smallest available index (i.e., the index of the non-used detection with highest confidence)
      for $j \in \mathbb{I}$ do   ▷ Iterate for all remaining indices in $\mathbb{I}$
        if $\mathrm{IOU}(\mathbf{z}_k^{i*}, \mathbf{z}_k^{j*}) > \tau_f$ then   ▷ If detections $i$ and $j$ has an IOU larger than $\tau_f$,
          $\mathbb{I} \Leftarrow \mathbb{I} \setminus \{j\}$   ▷ the index $j$ is removed from $\mathbb{I}$, the cardinality of the set decreases. The detection $j$ is suppressed. Note that when $j = i$, IOU=1, the index is removed and the detection is always used as output (see next line)
        end if
      end for
      $\mathbb{Z}_k \Leftarrow \mathbb{Z}_k \bigcup \{\mathbf{z}_k^{i*}\}$   ▷ Add detection $i$ to the set of output detections
      $\mathbb{S}_k \Leftarrow \mathbb{S}_k \bigcup \{\mathbf{s}_k^{i*}\}$   ▷ Add confidence $i$ to the set of output confidences
    end while
  end if
  return $(\mathbb{Z}_k, \mathbb{S}_k)$
```

KEY –
$o(\cdot)$: function that sorts detections and confidences by decreasing confidence;
$IOU(\cdot,\cdot)$: function that computes the intersection over union between two detections;
τ_f: overlap threshold to suppress detections.

Prior to use detections as input by tracking algorithms, detections are commonly filtered out in a pre-processing step. A common pre-processing step is the Non-Maximum Suppression (NMS) [44, 54], which maintains the most confident detections while removing the overlapping and less confident ones. NMS, described in Algorithm 1, receives as input the (large) set of detections $\mathbb{Z}_k^*$, their corresponding normalised confidences $\mathbb{S}_k^*$, and an overlap threshold τ_f. NMS order the detections by decreasing confidence, and then, iteratively removes detections with lower confidences and with intersection over union (IOU) larger than τ_f. As output, NMS generates a set of filtered detections $\mathbb{Z}_k$, which is a subset of the input ones, and their corresponding confidences $\mathbb{S}_k$. NMS is effective at removing overlapping detections; however, NMS discards information that might be valuable for the tracking process when multiple detectors are used. For instance, when detections $\mathbb{Z}_k^*$ are generated by multiple independent detectors, an object detected by a larger number of detectors might indicate a higher likelihood to be a true positive detection, than an object detected by a smaller number of detectors. This information is not leveraged by NMS.

Algorithm 2 Proposed detection fusion. Magenta indicates differences with respect to Algorithm 1.

```
Require: $(\mathbb{Z}_k^*, \mathbb{S}_k^*)$    ▷ Input tuple of detections and associated confidences sets
  $\mathbb{Z}_k \Leftarrow \emptyset$    ▷ Initialise output set of detections as an empty set
  $\mathbb{S}_k \Leftarrow \emptyset$    ▷ Initialise output set of corresponding confidences as an empty set
  if $|\mathbb{Z}_k^*| > 0$ then
    $(\mathbf{z}_k^{i*}, \mathbf{s}_k^{i*})_{i=1}^{|\mathbb{Z}_k^*|} \Leftarrow o((\mathbb{Z}_k^*, \mathbb{S}_k^*))$    ▷ Sort detections by decreasing confidence
    $\mathbb{I} \Leftarrow \{1, \dots, |\mathbb{Z}_k^*|\}$    ▷ Initialise set of detection indexes
    while $|\mathbb{I}| > 0$ do    ▷ Iterate while $\mathbb{I}$ contains indices
      $\breve{\mathbb{Z}}_k \Leftarrow \emptyset$    ▷ Initialise set of detections to fuse as an empty set
      $\breve{\mathbb{S}}_k \Leftarrow \emptyset$    ▷ Initialise set of confidences to fuse as an empty set
      $i \Leftarrow \min(\mathbb{I})$    ▷ Select the smallest available index (i.e., the index of the non-used
  detection with highest confidence)
      for $j \in \mathbb{I}$ do    ▷ Iterate for all remaining indices in $\mathbb{I}$
        if IOU$(\mathbf{z}_k^{i*}, \mathbf{z}_k^{j*}) > \tau_f$ then    ▷ If detections $i$ and $j$ has an IOU larger than $\tau_f$,
          $\mathbb{I} \Leftarrow \mathbb{I} \setminus \{j\}$    ▷ the index $j$ is removed from $\mathbb{I}$, the cardinality of the
  set decreases, and the detection $j$ is considered for *fusion*. Note that when $j = i$, IOU=1, the
  index is removed and the detection is always considered for fusion (see next line)
          $\breve{\mathbb{Z}}_k \Leftarrow \breve{\mathbb{Z}}_k \bigcup \{\mathbf{z}_k^{j*}\}$    ▷ Add detection $j$ to the set of detections to fuse
          $\breve{\mathbb{S}}_k \Leftarrow \breve{\mathbb{S}}_k \bigcup \{\mathbf{s}_k^{j*}\}$    ▷ Add confidence $i$ to the set of confidences to fuse
        end if
      end for
      $U \Leftarrow f(\breve{\mathbb{Z}}_k)$    ▷ Compute number of detectors contributing to the detection to fuse
      $\mathbb{Z}_k \Leftarrow \mathbb{Z}_k \bigcup \{\frac{1}{\sum_{n=1}^{|\breve{\mathbb{Z}}_k|} \breve{\mathbf{s}}_k^n} \sum_{n=1}^{|\breve{\mathbb{Z}}_k|} \breve{\mathbf{s}}_k^n \breve{\mathbf{z}}_k^n\}$    ▷ Fuse detection as the linear combination of detections
  with confidences as weights, and add to the set of output detections
      $\mathbb{S}_k \Leftarrow \mathbb{S}_k \bigcup \{\frac{U}{D} \frac{1}{|\breve{\mathbb{Z}}_k|} \sum_{n=1}^{|\breve{\mathbb{Z}}_k|} \breve{\mathbf{s}}_k^n\}$    ▷ Compute the confidence of the fused detection as an
  average, and add to the set of output confidences
    end while
  end if
  return $(\mathbb{Z}_k, \mathbb{S}_k)$
```

KEY –
$o(\cdot)$: function that sorts detections and confidences by decreasing confidence;
$IOU(\cdot, \cdot)$: function that computes the intersection over union between two detections;
τ_f: overlap threshold to suppress detections;
$f(\cdot)$: function that computes the number of detectors that contributed towards creating a fused detection;
D: total number of detectors.

To use this information, we propose a fusion algorithm that, unlike NMS, *fuses* overlapping detections instead of filtering them out, and uses the information of the number of detectors detecting each single object to produce enhanced confidence scores. The proposed fusion algorithm, described in Algorithm 2, receives the same input that NMS algorithm does, the (large) set of detections and their associated normalised confidences. Unlike NMS, the proposed fusion algorithm iteratively *fuses* high-confidence detections with lower-confidence ones when their IOU is larger than τ_f. New fused detections are generated as a weighted linear combination of the set of overlapping detections, $\breve{\mathbb{Z}}_k$, where the weights are the detections confidence. The detection confidence of a new fused detection is calculated as the average of the confidences of the detections to be fused normalised by $\frac{U}{D}$, where U is the number of detectors contributing to the new fused detection, and D is a constant equal to the total number of detectors. This normalisation rewards fused detections detected by a larger number of detectors, which we consider more likely to be true positives, and penalises fused detections detected by a smaller number of detectors, which we consider more likely to be false positives. Note that as $U \leq D$, the confidence of fused detections remains in the $[0,1]$ range. We experimentally set $\tau_f = \frac{1}{3}$. Fig. 4.3b shows a sample result of the fused detections.

We validate the multi-detector fusion approach in Sec. 4.6.2.

4.2.2 Data association

Traditionally, data association links previous tracking states with the current ones [181]. This results in the need for a clustering method for determining the identity of each of the particles at each time [115], specially for new particles. Differently, we propose an *early association* (EA) that links detections to previous predicted tracking states. The main benefits of the early association is that the clustering method is not needed, as newly generated particles are straight away associated with a trajectory, thus they have an associated identity since they are born; and an important speed up in the computation as clustering is commonly computational expensive. We refer to this association as *early* as it is performed before the update and resampling stages, unlike [115, 181] that is performed after the clustering and resampling stages.

Considering the set of fused detections $\mathbb{Z}_k$, we classify the set into two subsets based on the detections confidence as: strong and weak detections. *Strong* detections $\mathbb{Z}_k^+ = \{\mathbf{z}_k^i : s_k^i \geq \tau_s\}$ are confident detections and more likely to be true positives. *Weak* detections $\mathbb{Z}_k^- = \{\mathbf{z}_k^i : s_k^i < \tau_s\}$ are less confident detections and potential false positives. The ratio between the number of

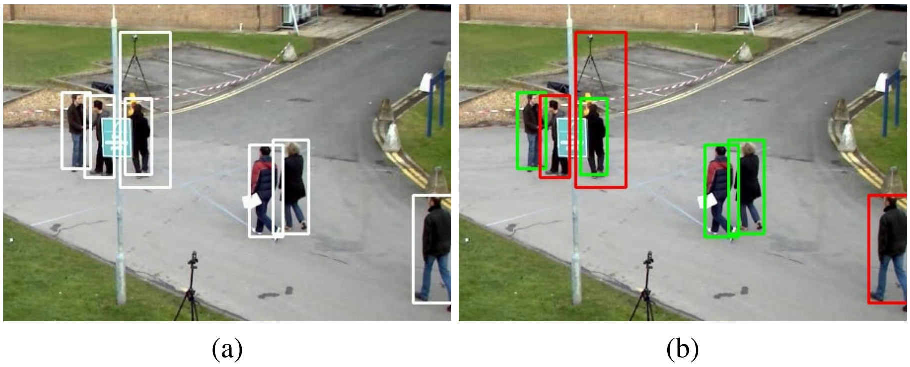


Figure 4.4: Example of strong and weak detections at frame 43 (crop) in PETS09-S2L1. (a) Fused detections $\mathbb{Z}_k$; (b) strong (green) $\mathbb{Z}_k^+$ and weak (red) $\mathbb{Z}_k^-$ detections after classification. Source images from [94].

false negatives and false positives can be varied accordingly to the problem's needs by tuning the confidence threshold τ_s. Fig. 4.4 shows a sample result of strong and weak detections. We use the concept of strong and weak detections to improve tracking performance. The idea is that strong detections can be highly trusted and, therefore, they can be used for critical tracking steps such as trajectory initialisation. However, weak detections are less reliable than the strong ones but can be still useful for maintaining the tracking of existing objects. Discarding weak detections will make the tracker to only rely on a limited number of strong detections and predictions, which can potentially produce drifts and, therefore, lost the track. Also, weak detections are useful to shorten the prediction time, maintaining a lower tracking uncertainty. Specifically, we associate the elements of the strong $\mathbb{Z}_k^+$ and weak $\mathbb{Z}_k^-$ set of detections to the predicted states using the Hungarian algorithm [90]. The association cost, $\omega_{k,\lambda}^j$, between a detection $\mathbf{z}_k^j$ and the predicted state, $\mathbf{x}_{k|k-1,\lambda}$, is

$$\omega_{k,\lambda}^j = d_l(\mathbf{z}_k^j, \mathbf{x}_{k|k-1,\lambda}) \cdot d_s(\mathbf{z}_k^j, \mathbf{x}_{k|k-1,\lambda}), \tag{4.8}$$

where $d_l(\cdot)$ and $d_s(\cdot)$ are the Euclidean distances between the position and bounding box size elements, respectively. Note that we multiply the normalised distances instead of averaging them, to penalise when they are dissimilar (e.g., when two objects are far from each other in the scene but appear close to each other on the image plane).

When a trajectory is not associated to any (strong or weak) detection, the state is estimated using existing particles only. When the trajectory is not associated to any detections for half a second, the state will be discarded before the EA and,therefore, the weight of its particles will gradually decrease toward zero.

EA enables the tracker to generate newborn particles that inherit the properties of its associated state (*inheritance*) or that produce a new identity (*initialisation*).

Inheritance. Strong detections $\mathbb{Z}_k^+$ generate J_k newborn particles to repopulate the area around existing states. The newborn particles are added to the L_{k-1} existing particles. In [115, 181], the newborn particles are created from a newborn importance function $p_k(\cdot)$ [115], which can be independently modelled from the estimated states as a Gaussian process:

$$\dot{\mathbf{x}}_{k,\lambda}^i \sim p_k(\dot{\mathbf{x}}_{k,\lambda}^i|\mathbf{z}_k^+) = \frac{1}{|\mathbb{Z}_k^+|} \sum_{\mathbf{z}_k^+ \in \mathbb{Z}_k^+} \mathcal{N}(\dot{\mathbf{x}}_{k,\lambda}^i; \mathbf{z}_k^+, \sigma_\lambda^n), \tag{4.9}$$

where $|\cdot|$ is the cardinality of a set, $\mathcal{N}(\cdot)$ is a Gaussian distribution and σ_λ^n is a constant covariance matrix. The covariance matrix can be dynamically updated based on parameters, such as detection size or video frame rate. Each newborn particle has an associated weight defined as

$$\pi_k^i = \frac{1}{J_k} \frac{\gamma_k(\dot{\mathbf{x}}_{k,\lambda}^i)}{p_k(\dot{\mathbf{x}}_{k,\lambda}^i|\mathbf{z}_k^+)}, \quad i = L_{k-1}+1, ..., L_{k-1}+J_k, \tag{4.10}$$

where $\gamma_k(\cdot)$ is the birth intensity, which is assumed to be constant when no prior knowledge about the scene is available [115]. Typically, J_k is chosen to have, on average, ρ particles per newborn object [181]. The process described in Eq. 4.9 could create newborn particles that are dissimilar from the corresponding state, as they are independently created.

Let $\hat{\mathbb{Z}}_k^+$ and $\hat{\mathbb{Z}}_k^-$ be the sets that contain, respectively, strong and weak detections that are *associated* to one of the predicted states, i.e. to an existing trajectory. Let $\hat{\mathbb{Z}}_k = \hat{\mathbb{Z}}_k^+ \cup \hat{\mathbb{Z}}_k^-$ be the set of detections that inherit the identity of the corresponding trajectories. We create newborn particles from $\hat{\mathbb{Z}}_k$ and inherit properties from their associated predicted states: the position and bounding box size are created from detections, whereas velocity and identity are inherited from the associated states.

We use a Gaussian mixture as an importance function to sample position and bounding box size elements from the detections as

$$\dot{\mathbf{x}}_{k,\lambda}^i \sim p_k(\dot{\mathbf{x}}_{k,\lambda}^i|\hat{\mathbf{z}}_k) = \frac{1}{|\hat{Z}_k|} \sum_{\hat{\mathbf{z}}_k \in \hat{Z}_k} \mathcal{N}(C\dot{\mathbf{x}}_{k,\lambda}^i; C\hat{\mathbf{z}}_k, C\Sigma_{k,\lambda}), \tag{4.11}$$

where

$$C = \begin{pmatrix} D & 0_{4\times2} \\ 0_{2\times4} & I_{2\times2} \end{pmatrix}, D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.12}$$

and

$$\Sigma_{k,\lambda} = (\mathbf{n}_{k,\lambda})^T, \tag{4.13}$$

where $\mathbf{n}_{k,\lambda} \in \mathbb{X}_k$ is a noise variable that varies for different times and identities generated from a Gaussian distribution as $n_{k,\lambda} \sim \mathcal{N}_{k,\lambda}(0;\sigma_{k,\lambda})$, with null mean and standard deviation $\sigma_{k,\lambda}$. The perspective distortion during the generation of newborn particles is modelled by $\Sigma_{k,\lambda}$. The standard deviations are modelled as a function of the dimensions of the bounding box detection. The values could be also learned.

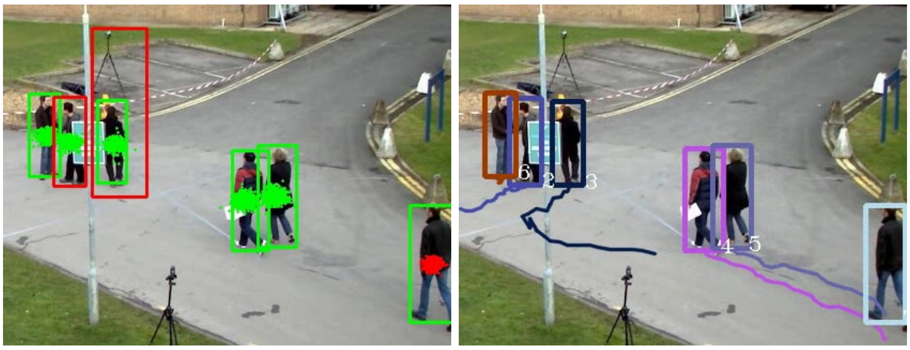
The new particles inherit the velocity and the identity from the trajectory that they are associated to as

$$\begin{aligned} \bar{u}^i_{k,\lambda} &= \bar{u}_{k-1,\lambda} + n^u_{k,\lambda}, \\ \bar{v}^i_{k,\lambda} &= \bar{v}_{k-1,\lambda} + n^v_{k,\lambda}, \\ \lambda_k &= \lambda_{k-1}, \end{aligned} \tag{4.14}$$

where $(\bar{u}_{k-1,\lambda}, \bar{v}_{k-1,\lambda})$ are the horizontal and vertical components of the estimated velocity of λ-th object. The weight of the newborn particles is calculated as in Eq. 4.10.

Initialisation While un-associated *weak* detections are discarded after EA, un-associated *strong* detections, $\check{\mathbb{Z}}_k = \mathbb{Z}^+_k \backslash \hat{\mathbb{Z}}^+$, initialise new trajectories. Newborn particles associated to a *new object* are generated in a limited volume of the state space around the un-associated strong detections. All particles generated from a given un-associated strong detection are created with the same new identity.

We do not consider re-identification, therefore, objects that exit and re-enter in the field of view of the camera are treated as new objects. The newborn importance function $p_k(\cdot)$ in Eq. 4.9 could be used to regulate where objects are more likely to enter in the scene [114]. Then, each



(a) (b)

Figure 4.5: Example of newborn particles initialising a new trajectory from an un-associated strong detection on the bottom right corner of the image at frame 43 (crop) in PETS09-S2L1 sequence. (a) Strong and weak detections indicated with green and red bounding boxes respectively; and existing and newborn particles indicated with green and red dots, respectively. (b) Tracking results where the identity of the objects is colour coded and the lines indicate previous estimated locations. Source images from [94].

detection in $\check{\mathbb{Z}}_k$ initialises a *new trajectory* and generates newborn particles using a Gaussian mixture as

$$\dot{\mathbf{x}}^i_{k,\lambda} \sim p_k(\dot{\mathbf{x}}^i_{k,\lambda}|\check{\mathbf{z}}_k) = \frac{1}{|\check{\mathbb{Z}}_k|} \sum_{\check{\mathbf{z}}_k \in \check{Z}_k} \mathcal{N}(\dot{\mathbf{x}}^i_{k,\lambda}; \check{\mathbf{z}}_k, \Sigma_{k,\lambda}), \tag{4.15}$$

where $\Sigma_{k,\lambda}$ it is defined as in Eq. 4.13 but the standard deviations are a function of the un-associated strong detection instead of the associated trajectory. The weights of the particles are calculated as in Eq. 4.10.

Fig. 4.5 shows a sample of (a) strong and weak detections, and new and existing particles; and (b) tracking results. The person entering on the field of view on the right is initialised because of the presence of an un-associated strong detection. The person with identity number 2 is detected by a weak detection which will be used in the update step. However, the weak detection that is a false positive is discarded because it is not associated to any predicted states. In Fig. 4.5b, green/red dots indicate existing/newborn particles.

4.2.3 Perspective-dependent object motion prediction

Let $\mathbb{X}_k$ be the set of all estimated states at time k whose elements are $\mathbf{x}_{k,\lambda} \in \mathbb{X}_k$ (Eq. 1.2). The elements of this set are obtained at each time step from the set of all existing particles $\dot{\mathbb{X}}_k$ whose elements are $\dot{\mathbf{x}}^i_{k,\lambda} \in \dot{\mathbb{X}}_k$, where $\dot{\mathbf{x}}^i_{k,\lambda}$ is the i-th particle.

Assuming that the motion of an object is independent from that of other objects, we propose a prediction model that propagates each particle $\dot{\mathbf{x}}^i_{k-1,\lambda}$ as

$$\dot{\mathbf{x}}^i_{k,\lambda} = G_{k,\lambda}\dot{\mathbf{x}}^i_{k-1,\lambda} + \mathbf{N}_{k,\lambda}, \tag{4.16}$$

where $G_{k,\lambda}$ is an affine transformation defined as

$$G_{k,\lambda} = \begin{pmatrix} A_{k,\lambda} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & B_{k,\lambda} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \end{pmatrix}, \tag{4.17}$$

where $\mathbf{0}_{2\times 2}$ and $\mathbf{I}_{2\times 2}$ are the zero and identity matrices of dimensions 2×2, respectively; $\mathbf{N}_{k,\lambda} \in \mathbb{X}_k$ is a noise variable, where each component $n^l_{k,\lambda} \sim \mathcal{N}_{k,\lambda}(0;\cdot)$ is sampled from an independent Gaussian variable of zero mean and standard deviation proportional to the bounding box dimensions of the state in the previous frame. As an object moving at constant velocity produces a smaller apparent displacement on the image plane when it is farther from the camera, we propose to improve the model by considering the effect of foreshortening. To this end, we model $\mathbf{N}_{k,\lambda}$ as a function of the distance from the camera, which is inversely proportional to the bounding box dimensions. We assume here that every object has the same dimensions than all the others in the real world. Specifically, we set the standard deviation of the noise for the horizontal and vertical components to be proportional to the width and height of the estimated state $\mathbf{x}_{k,\lambda}$.

In addition to the above, object acceleration variations, noisy detections and camera motion may generate erroneous predictions. To account for this, instead of relying only on the previous time step [115], we average the past M states over a longer time interval $k \in [k-M, k-1]$. Therefore, the proposed object motion prediction dynamically updates the position and velocity via the average velocity in the previous M frames, where A_k and B_k are defined as:

$$A_{k,\lambda} = \begin{pmatrix} 1 & \frac{u_{k-M|k-1,\lambda}}{u_{k,\lambda}} \\ 0 & \frac{u_{k-M|k-1,\lambda}}{\bar{u}_{k,\lambda}} \end{pmatrix}, \quad B_{k,\lambda} = \begin{pmatrix} 1 & \frac{v_{k-M|k-1,\lambda}}{v_{k,\lambda}} \\ 0 & \frac{v_{k-M|k-1,\lambda}}{\bar{v}_{k,\lambda}} \end{pmatrix}, \tag{4.18}$$

where $u_{k-M|k-1,\lambda}, v_{k-M|k-1,\lambda}$ are the average horizontal and vertical velocities of the estimated state $\mathbf{x}_{k,\lambda}$, respectively, whose values are computed as

$$\left(u_{k-M|k-1,\lambda}, v_{k-M|k-1,\lambda}\right) = \frac{1}{M}\sum_{k\in[k-M,k-1]} \left(u_{k,\lambda}, v_{k,\lambda}\right), \tag{4.19}$$

where $M = \min\left(M_{k,\lambda}, M_{max}\right)$, $M_{k,\lambda}$ is the number of time steps since the object $\mathbf{x}_{k,\lambda}$ was initialised and M_{max} the maximum number of considered time steps.

The weights of the particles, π^i_k, are not modified during the prediction step, therefore

$$\pi^i_{k|k-1} = \pi^i_{k-1}, \qquad i = 1, ..., L_{k-1}, \tag{4.20}$$

where L_{k-1} is the number of existing particles at $k-1$.

4.2.4 Perspective-dependent update, resampling and state estimation

Let the set of particles that share the same identity be $\dot{\mathbb{X}}_{k,\lambda} = \{\dot{\mathbf{x}}^i_{k,\lambda} : \lambda = \lambda_j\}$. After new detections are generated, the particles are re-weighted using the new detections at time k [115, 117, 182]. The particle weights are *updated* as

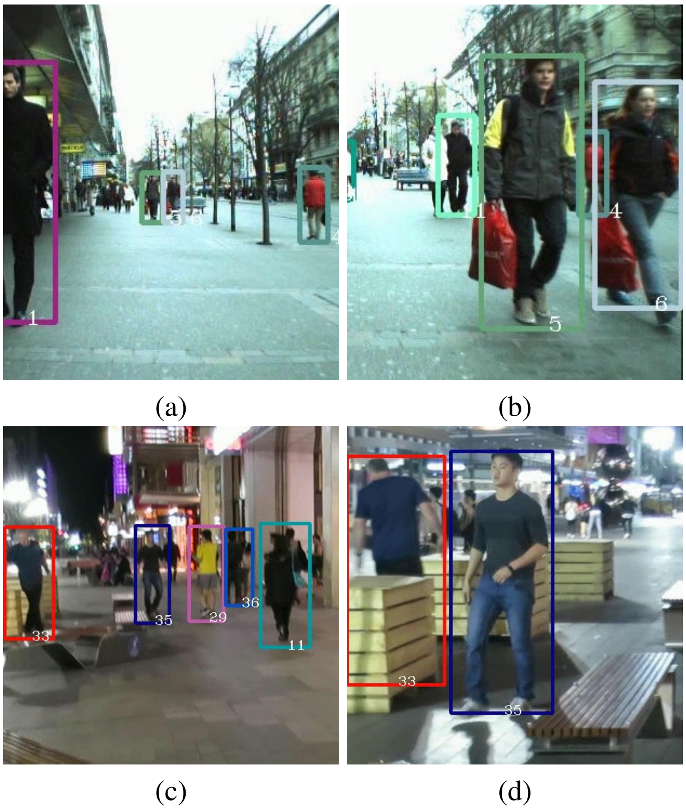
$$\pi^i_k = \pi^i_{k-1}\, p(\mathbf{z}^j_k | \dot{\mathbf{x}}^i_{k,\lambda}), \tag{4.21}$$

where $p(\mathbf{z}^j_k | \dot{\mathbf{x}}^i_{k,\lambda})$ is the likelihood of each particle component defined as

$$p(\mathbf{z}^j_k | \dot{\mathbf{x}}^i_{k,\lambda}) \propto e^{-\frac{|\mathbf{z}^j_k - \dot{\mathbf{x}}^i_{k,\lambda}|^2_2}{2(\sigma_{k,\lambda})^2}}, \tag{4.22}$$

where $e(\cdot)$ indicates the exponential function. Note that as detections do not have velocity components, they are discarded from this calculation. The standard deviations for the object size, $\sigma^w_{k,\lambda}$ and $\sigma^h_{k,\lambda}$, are constant, whereas the standard deviations for the object location, $\sigma^u_{k,\lambda}$ and $\sigma^v_{k,\lambda}$, are a function of the detection width to account for the foreshortening and a function of the video frame rate to account for frame rate variations. Unlike [115, 138] where the components of $\sigma_{k,\lambda}$ are defined as constant values, we define them as a time-variant variable that regulates the score based on the similarity between the components of the particles and detections (i.e. the particles of an object far from the camera will be less spread than those of a closer object due to the perspective). Fig. 4.6 shows examples of the use of the proposed perspective-dependent approach.

After the update step, *resampling* helps on avoiding the degeneracy problem [17]. The standard multinomial resampling [17, 181] splits particles proportionally to their weights, frame-by-frame independently. Because newborn particles have in general a lower weight than existing



(a) (b)

(c) (d)

Figure 4.6: Examples of tracking under perspective changes at frames 32 and 102 (crops) in ETH-Bahnhof, and at frames 178 and 375 (crops) in ADL-Rundle-8. Objects 5 and 6 (see (a), (b)) and objects 33 and 35 (see (c), (d)) are correctly tracked despite considerable perspective changes. Source images from [94].

particles, new objects may not be initialised due to repetitive deletion of their particles during resampling. To allow newborn particles to grow over time and reach a comparable weight to that of existing particles, newborn particles are resampled independently from existing particles using a multi-stage multi-nomial resampling step [115]. Finally, each state $\mathbf{x}_{k,\lambda}$ is estimated as the average of all resampled particles sharing the same identity:

$$\mathbf{x}_{k,\lambda} = \frac{1}{|\dot{\mathbb{X}}_{k,\lambda}|} \sum_{\dot{\mathbf{x}}^i_{k,\lambda} \in \dot{\mathbb{X}}_{k,\lambda}} \dot{\mathbf{x}}^i_{k,\lambda}, \tag{4.23}$$

where $\dot{\mathbb{X}}_{k,\lambda}$ is the set of particles sharing the same identity λ.

4.3 Multi-object tracking from a moving camera

Tracking multiple independently-moving objects with a camera mounted on moving agents such as robots or worn by people is becoming increasingly common. However, most online trackers rely on simple motion models that may be inaccurate in these scenarios as they solely consider the motion of the objects but disregard that of the camera. This is broadly discussed in Sec. 2.3. To address this problem, we present a real-time multi-object tracker based on the EA-PHD-PF (introduced in Sec. 4.2) with a prediction model that disentangles the motion of objects from that of the camera.

While offline trackers can use future observations to solve the association problem, online trackers rely on past observations only. To predict the likely location of objects in future frames, online trackers use motion models [16, 39, 93, 120]. However, the motion of an object can be masked by the global motion of the camera, thus making the prediction task very challenging. Moreover, miss-detections increase the difficulty in accurately predicting future object locations.

Predictors can model the motion of an object with random noise (Brownian model) [131] or with linear predictions based on its past motion [34, 115, 165, 218]. With static cameras, the linearity assumption may hold and linear predictors perform accurately; when objects and camera move independently, drifts often happen. Higher frame rates can alleviate this problem within short temporal windows [62], but require a higher energy budget, which is undesirable for robotic and wearable cameras.

With static cameras, prediction models can use deep learning [8, 19, 144]. The location and scale can be predicted with a LSTM [19], by modelling person-to-person interactions [8] and person-to-person interactions considering static objects present in the environment [144]. With moving cameras, methods require additional information, such as camera parameters [16, 39] and additional sensors (e.g. depth sensor) [39], or strong constraints on the type of application scenario, such as using high-altitude drones recording a top-down view [16] or having all objects sharing a common plane [16, 93, 101, 120]. A desirable prediction model for moving cameras should instead be agnostic to the type and location of objects as well as of the scene type.

In this section, we propose an online multi-object tracker with a novel model that accurately predicts the motion of objects despite the motion of the camera (see Fig. 4.7). The model is informed by global motion estimation and separates the object motion from that of the camera. We use as a base tracker for the model the Early Association Probability Hypothesis Density Particle Filter [C7].

4.3.1 Global motion estimation

Predicting the motion of objects captured by a moving camera is important for vision tasks. In this section, we present an accurate model to forecast the position of moving objects by disentan-

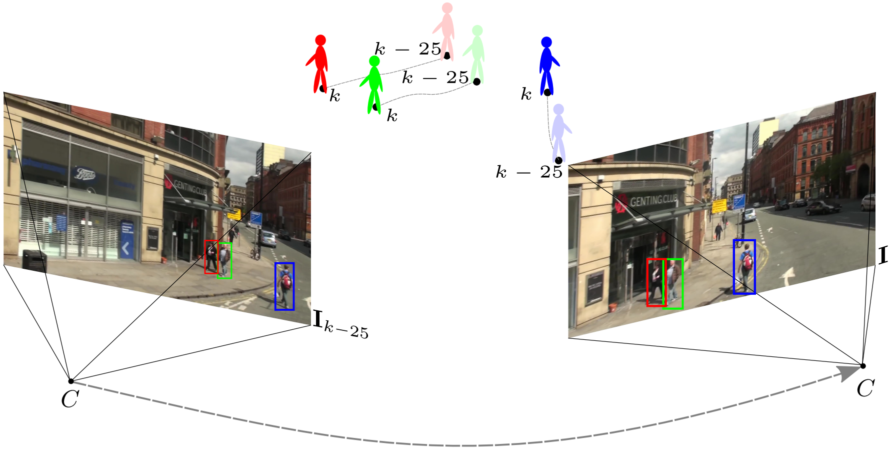


Figure 4.7: Sample multi-object tracking result (bounding boxes) using the proposed prediction model that facilitates maintaining object identities when camera and objects move independently, and without using any appearance information. Note that C is the camera and $\mathbf{I}_k$ is the frame at time k.

gling global and object motion without the need of camera calibration or planarity assumptions. Our predictor uses past observations to model the motion of objects online by selectively tracking a spatially balanced set of keypoints and estimating scene transformations between pairs of frames.

Cameras can move with 6 degrees of freedom, three translation-wise and three rotation-wise, as shown in Fig. 4.8. Let's consider a camera installed on a moving platform looking to an object placed straight in front of the camera 10 meters away from it. Considering a real robot, let's consider that it can translate up to 30 km/h and rotate up to 180 degrees/s [1] and let's assume that the objects remains static. For instance, if a camera records at 30 fps while the robot moves, the camera can translate up to 0.28 meters and rotate up to 6 degrees on the ground plane (i.e. along the y axis) within a frame interval. Let's consider that the camera translates at maximum translation speed towards the object during the time of a frame; which will produce a null pixel displacement on the image plane. Now, let's consider that the camera yaws (i.e. rotates along the Z axis at maximum rotational speed on during the time of a frame, which will produce a displacement of 157 pixels on the image plane. Fig. 4.9 shows examples on how the translation and rotation of the camera affects to the observed image plane location of a static object located at 10 meters straight in front of the camera. As can intuitively be imagined, it can be seen that rotations produce larger image plane displacements than translations in the mentioned scenario. A real example of this effect is shown in Fig. 4.10 where, as expected, the apparent displacement of a person in a translation-dominant video produces smaller displacement on the image plane

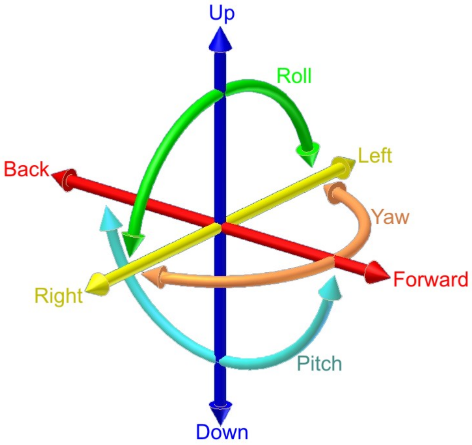


Figure 4.8: The six degrees of freedom of a camera: forward/back, up/down, left/right, yaw, pitch and roll. Image from [2].

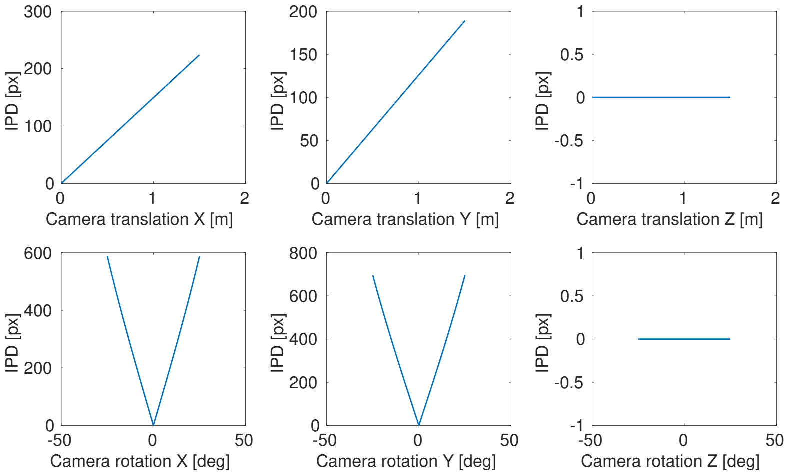


Figure 4.9: The effect of translation and rotation of the camera on the observed image plane location of a static object. In these scenarios, where the object is located 10 meters away from the camera on its y axis and with common camera settings, camera rotation has a greater effect than translation. KEY: IPD, image plane displacement.

than in a rotation-dominant video. Therefore, we assume that in our object scenario, translation between consecutive frames can be neglected and consider camera motions as purely rotational.

Without loss of generality, a projective transformation, usually known as homography [174], $\mathbf{H}_{k|k-1}$, relates points in frame $\mathbf{I}_{k-1}$ with their correspondences in the frame $\mathbf{I}_k$ (i.e. the same world point but imaged in another frame) in three different scenarios: (i) pure rotation, (ii) pure zoom and (iii) any motion in the 6 DoF with planar surfaces.

As discussed previously, we assume that the camera translation effect between consecutive frame is negligible. Therefore, in this work we make use of the first homography use case, where any pixel in frame $k-1$ is related by an homography transformation with its corresponding pixel in frame k. Note that no planar assumption is needed and no camera parameters are required.

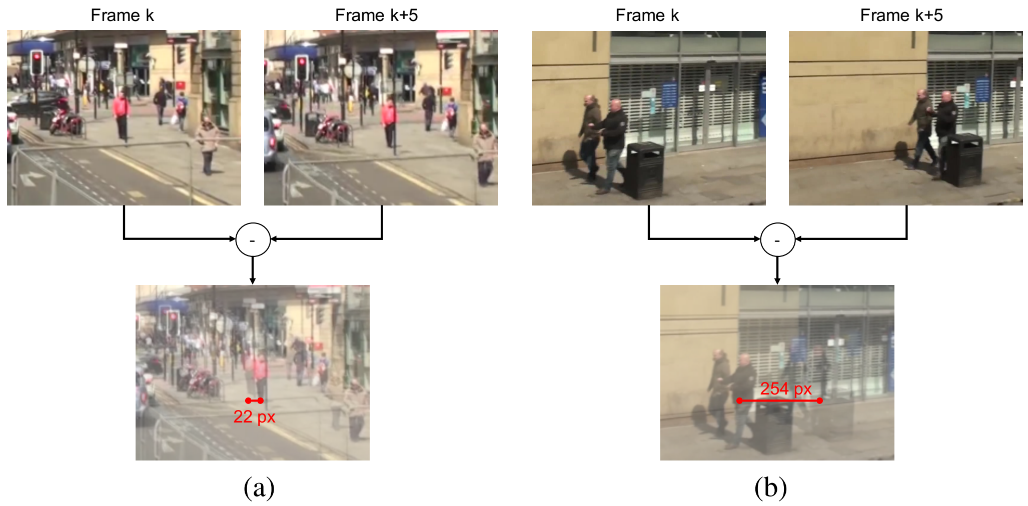


Figure 4.10: Samples from MOT17-13 showing the effect on the image plane when the camera undergoes a dominant (a) translation motion and (b) rotation motion. Note that the person in red in (a) moves 22 pixels as the camera just perform a translation while the person in black in (b) moves 254 pixels, more than an order of magnitude of difference for the same temporal difference and similar depth to the object. Frames k and $k+5$ are chosen for the sake of clarity in the example. Source images from [125].

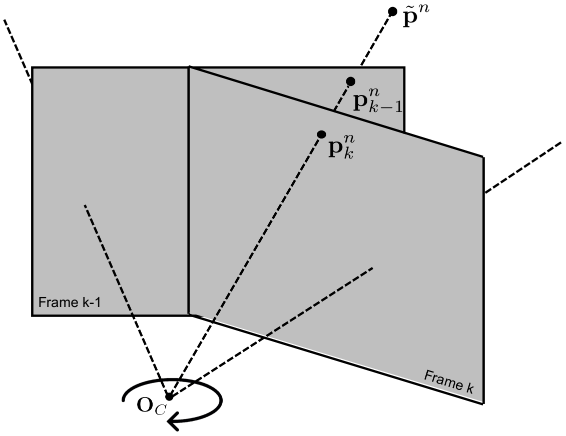


Figure 4.11: Two consecutive frames of a video sequence, where a pure camera rotation occur. A static real-world point $\tilde{\mathbf{p}}^n$ is visible in both images and imaged as $\mathbf{p}_{k-1}^n$ and $\mathbf{p}_k^n$. These two points (and the rest of the points visible in the two views) are related by an homography transformation $\mathbf{H}_{k|k-1}$.

Fig. 4.11 shows two consecutive frames whose camera undergoes a pure rotation. Points between both frames are related as $\mathbf{p}_k^n = \mathbf{H}_{k|k-1}\mathbf{p}_{k-1}^n$ where $\mathbf{p}_k^n = (u, v)$ is a generic point n in the image plane at time k. Homogeneous coordinates are considered and $\mathbf{H}_{k|k-1}$ is a 3 by 3 homography matrix between frames at time $k-1$ and k.

Let's consider to have a set of detections $\mathbb{Z}_k$ as defined in Eq. 1.1 in the form of points (i.e. discarding the dimension components) that represent potential objects of interest on the image plane.

Let $\mathbf{I}_k$ be a frame at time k and $\mathbf{x}_k = (u, v, 1)$ be the position, in homogeneous coordinates, of an object on the image plane, where u and v are the horizontal and vertical position of the centre of the object, respectively. Our goal is to determine the future $T_F \geq 1$ states $\mathbf{x}_{k:k+T_F}$, given the $T_P \geq 2$ past observations.

To facilitate the prediction of the object location, we decompose the motion observed on the image plane seen from a moving camera into two components: the global and the object motion. A graphical representation of this decomposition is depicted in Fig. 4.12. The *global motion* can be inferred from the coherent motion of the background [101]. The proposed object motion predictor models the global background motion between two frames, $\mathbf{I}_{k-2}$ and $\mathbf{I}_{k-1}$, with a homography, $\mathbf{H}_{k-1|k-2}$, assuming that, in ground-level vision, the global motion between consecutive frames can be approximated with a camera rotation (i.e. translation effects are negligible). For the prediction, we assume that the global motion between $k-1$ and k is similar to that between $k-2$ and $k-1$, and thus $\mathbf{H}_{k|k-1} \approx \mathbf{H}_{k-1|k-2}$. Therefore, given $\mathbf{x}_{k-1}$, the position of an object in $\mathbf{I}_{k-1}$, if the camera moves and the object is static in the real world, the predicted position of the same object in $\mathbf{I}_k$ is

$$\mathbf{x}_k = \frac{1}{\alpha_k} \mathbf{H}_{k-1|k-2} \mathbf{x}_{k-1}, \tag{4.24}$$

where $\alpha_k = \langle \mathbf{h}_3, \mathbf{x}_{k,\lambda} \rangle$ is a normalisation factor; $\mathbf{h}_3$ is the third row of $\mathbf{H}_{k-1|k-2}$ and $\langle \cdot, \cdot \rangle$ is the dot product. To estimate $\mathbf{H}_{k-1|k-2}$, we selectively detect and track keypoints on the background across frames. Fig. 4.13 summarises the proposed homography estimation procedure.

Let the set of keypoints detected in $\mathbf{I}_{k-2}$ be

$$\mathbb{P}_{k-2} = \{\mathbf{p}^n_{k-2} = (u, v)\}, \tag{4.25}$$

where u and v are the horizontal and vertical coordinates of keypoint n on the image plane. As implementation decision, we use *good features to track* [166] as keypoint detector, as it provides a good trade-off between speed and accuracy, and discard keypoints on moving objects based on a mask defined by the results of a people detector, extended by a margin to account for imprecisions on the detections.

We then track keypoints in $\mathbf{I}_{k-1}$ using a sparse iterative version of the Lucas-Kanade optical

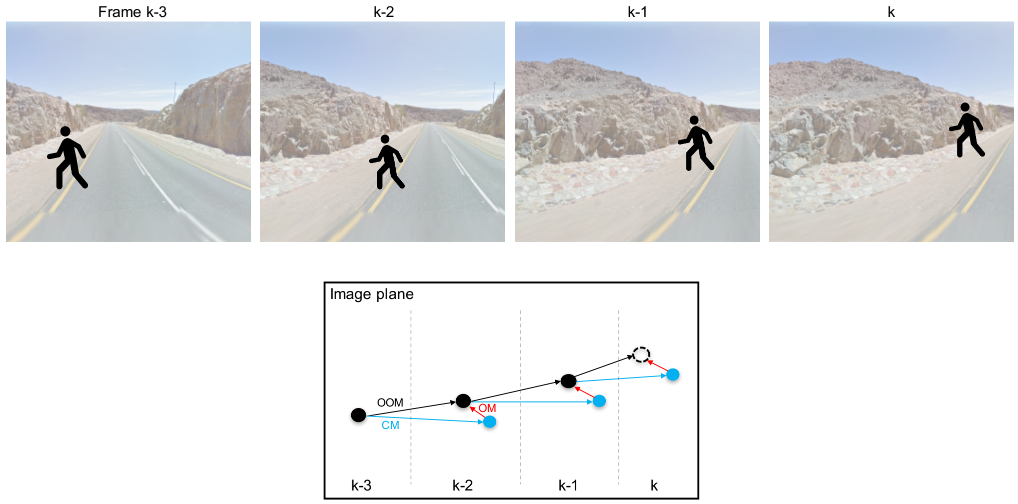


Figure 4.12: Sample representation to depict the observed object motion (OOM), camera motion (CM) and object motion (OM) concepts used in this section. In the first row, we show four consecutive frames from time $k-3$ to k captured from a moving camera that is rotating towards the left while a person is moving up and to the left with respect to the camera pose. In the second row, we show the motion decomposition. Points indicate location of the person and arrows indicate the displacement (motion) of the person between consecutive frames on the image plane. Black points indicate the position of the person on the image plane. Blue points indicate where the person would be if the person wouldn't have moved. The dashed black point indicates the estimated location of the person in the frame k. Black arrows are the OOM, blue arrows correspond to the CM and red arrows are the OM. In our model, we propose to decompose the OOM into CM and OM which allows us to learn the OM and then predict the location of the person in future frames. Note that the person is actually moving up and to the left with respect to the camera, as red arrows show, but the OOM is clearly to the right, as black arrows show)

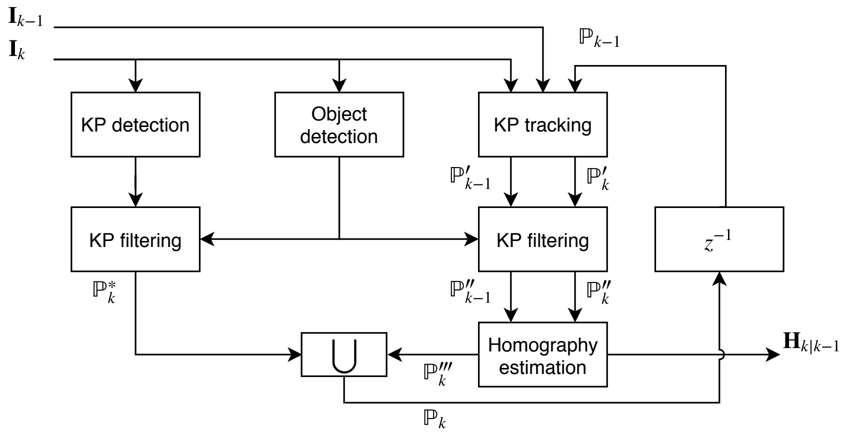


Figure 4.13: The proposed homography estimation pipeline.

flow in pyramids [32]. Tracking generates two sets of matched keypoints: $\mathbb{P}'_{k-2}$ and $\mathbb{P}'_{k-1}$, with $|\mathbb{P}'_{k-2}| = |\mathbb{P}'_{k-1}|$ where $|\cdot|$ is the cardinality of a set. Note that keypoints lost during tracking are removed from both sets. The filtered sets of matched keypoint pairs, $\mathbb{P}''_{k-2}$ and $\mathbb{P}''_{k-1}$, are obtained after masking the keypoints that lie on potentially moving objects (see Fig. 4.14).

As the number of tracked keypoints decreases over time due to occlusions, tracking errors and corresponding 3D points exiting the field of view of the camera, we maintain a balanced spatial distribution of keypoints [186] by dividing the frame into $N_u \times N_v$ equally-sized cells and triggering a keypoint detection process in cells with fewer than N_m keypoints. When a keypoint detection process is triggered, new keypoints are detected, $\mathbb{P}^{(i,j)}_{k-1}$, only outside the filtering mask; where $i \in [1, N_u]$ and $j \in [1, N_v]$ are the horizontal and vertical indices of the cells. The new set of detected keypoints is $\mathbb{P}^{*}_{k-1} = \bigcup_{i,j} \mathbb{P}^{(i,j)}_{k-1}$, where $\bigcup$ is the union operator.

We then compute the homography, $\mathbf{H}_{k-1|k-2}$, with the filtered and matched keypoints, $\mathbb{P}''_{k-2}$ and $\mathbb{P}''_{k-1}$, by calculating the transformation that relates the position of the keypoints in these sets using random sample consensus (RANSAC) [56, 174]. RANSAC generates the set of inlier keypoints, $\mathbb{P}'''_{k-1}$, by discarding matched keypoints that follow a different (homographic) transformation (see Fig. 4.14). Finally, the newly detected keypoints, $\mathbb{P}^{*}_{k-1}$, are added to the set of inlier keypoints as $\mathbb{P}_{k-1} = \mathbb{P}'''_{k-1} \cup \mathbb{P}^{*}_{k-1}$, which will be used for the next frame.

We show a visual representation of a estimated $\mathbf{H}_{k-1|k-2}$ in Fig. 4.15.

4.3.2 Global-motion aware object motion prediction

When the global motion produced by the camera motion has been estimated (e.g. as described in Sec. 4.3.1), we estimate the *object motion* (velocity) over the past T_P observed positions as

$$\bar{\mathbf{x}}_{k-1|k-T_P} = \frac{1}{T_P - 1} \sum_{i=0}^{T_P-2} \left(\mathbf{x}_{k-i-1} - \mathbf{H}_{k-i-1|k-i-2} \mathbf{x}_{k-i-2} \right). \tag{4.26}$$

Assuming that the motion in the 3D world of the object and the one of the camera are similar in the near future, we iteratively predict the object position as

$$\mathbf{x}_k = \frac{1}{\alpha_k} \mathbf{H}_{k-1|k-2} \mathbf{x}_{k-1} + \bar{\mathbf{x}}_{k|k-T_P}. \tag{4.27}$$

Next, we integrate the proposed object motion prediction in the probabilistic tracking framework presented in Sec. 4.2. The proposed prediction model decomposes the observed motion of an object in two independent components, global and object motion, for each particle as

$$\dot{\mathbf{x}}^{i}_{k,\lambda} = \underbrace{\mathbf{G}_{k-1|k-2,\lambda}\, \mathbf{x}^{i}_{k-1,\lambda}}_{\text{Global Motion}} + \underbrace{\mathbf{T}_{k|k-1}\, \mathbf{x}^{i}_{k-1,\lambda}}_{\text{Object Motion}} + \mathbf{N}_{k,\lambda}, \tag{4.28}$$

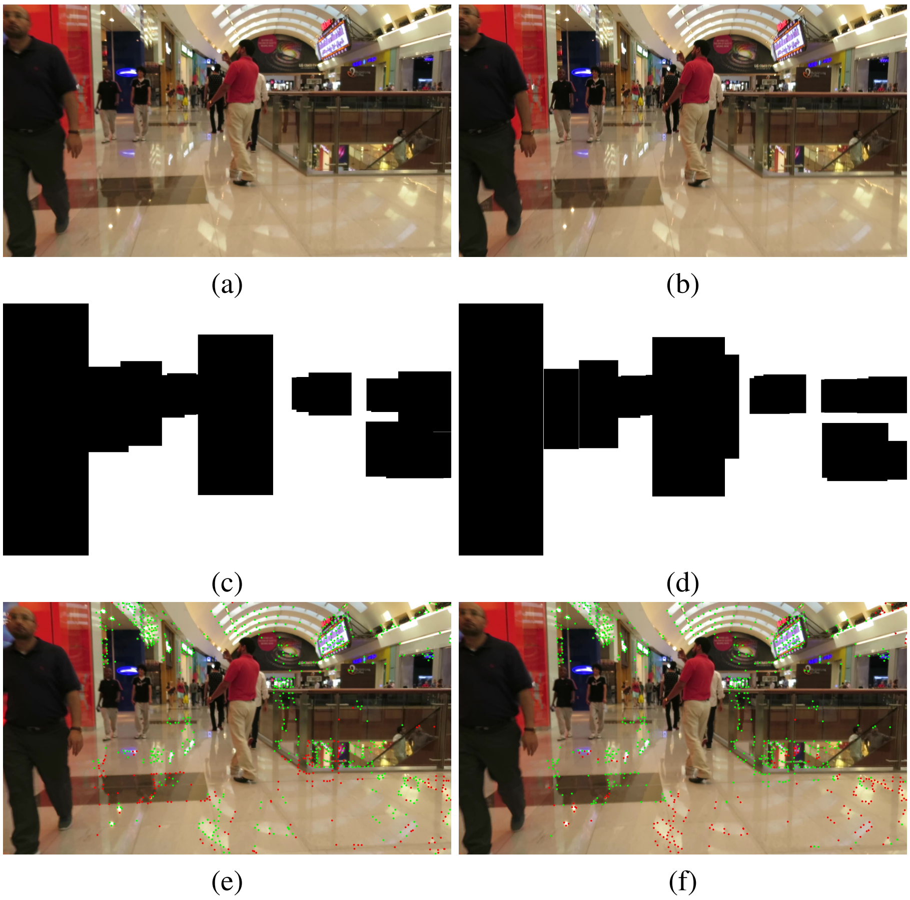


Figure 4.14: Visualisation of the main steps towards the proposed camera motion estimation for MOT17-11 sequence. (a-b) Input frames, (c-d) generated object mask and (e-f) tracked key-points. Key-point colour coded for inliners (green) and outliers (red) in the RANSAC scheme. Homography between frames is calculated on the inlier points drawn in sub-figures e and f. Note that the matching between keypoints is not represented in the figure. Images (a) and (b) from [125].

where $\mathbf{N}_{k,\lambda}$ is the noise matrix and $\mathbf{x}^{i}_{k,\lambda}$ is the i-th particle [17, C7].

The global motion matrix, $\mathbf{G}_{k-1|k-2,\lambda}$, describes the temporal evolution of the object on the image plane when the object is stationary in the 3D world and is defined as:

$$\mathbf{G}_{k-1|k-2,\lambda} = \begin{bmatrix} \frac{1}{\alpha_{k-1,\lambda}}\mathbf{H}_{k-1|k-2} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times2} \\ \mathbf{0}_{2\times3} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \end{bmatrix}, \tag{4.29}$$

where $\mathbf{H}_{k-1|k-2} \in \mathbb{R}^{3\times3}$ is a homography transformation that approximates the global motion between $k-1$ and k with that between $k-2$ and $k-1$; $\alpha_{k-1,\lambda} = \langle \mathbf{h}_3, (u,v,1)\rangle$ is a normalisation factor ($\mathbf{h}_3$ is the third row of $\mathbf{H}_{k-1|k-2}$ and $\langle\cdot\rangle$ is the dot product); and $\mathbf{0}$ and $\mathbf{I}$ are the zero and identity matrices whose dimensions are indicated in the subscript. We estimate the homography

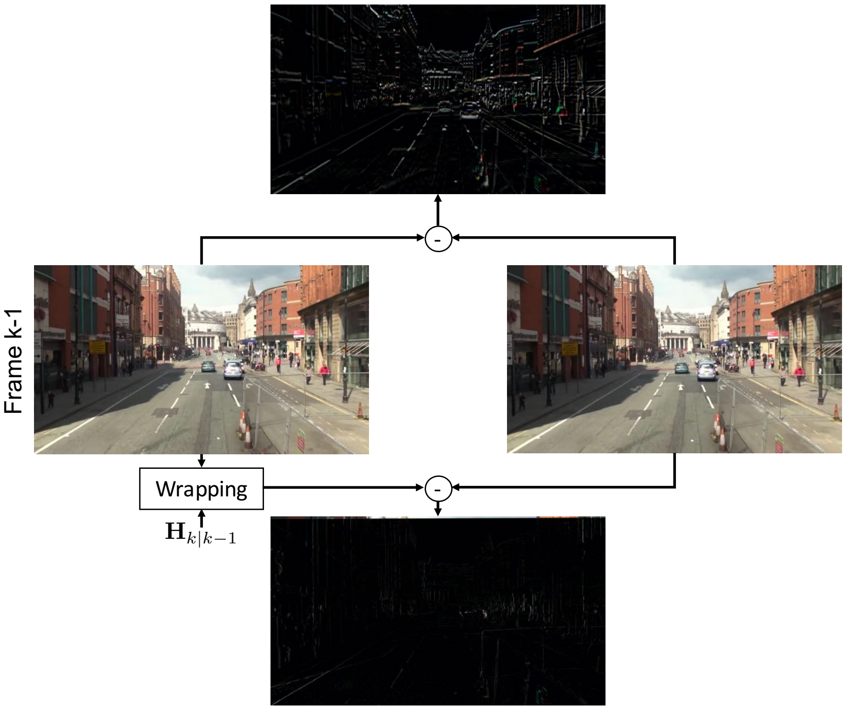


Figure 4.15: Qualitative homography quality assessment. Two consecutive frames at time $k-1$ and k are in the middle of the figure. On the top, the direct difference between the frames shows motions. On the bottom, the frame at $k-1$ is wrapped (i.e. rectified using the homography) and the difference between the wrapped version of the frame at $k-1$ and the frame at k show mostly no motions. Source images from [125].

transformation by detecting and tracking keypoints on the background as described earlier in this section.

The object motion matrix, $\mathbf{T}_{k|k-1}$, describes the temporal evolution of the object on the image plane when the camera is stationary and is defined as:

$$\mathbf{T}_{k|k-1} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{A} & \mathbf{0}_{3\times2} \\ \mathbf{0}_{2\times3} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \end{bmatrix}, \tag{4.30}$$

where

$$\mathbf{A} = \begin{bmatrix} \Delta_k & 0 \\ 0 & \Delta_k \\ 0 & 0 \end{bmatrix}, \tag{4.31}$$

and Δ_k is the time interval between two consecutive frames.

We account for unmodelled factors with the noise matrix, $\mathbf{N}_{k,\lambda}$, where each of its component is a value sampled from a Gaussian noise with zero mean and standard deviation σ^*. These parameters are commonly selected by estimating the uncertainties of each state component in a training dataset but can also be fixed [165], a function of the video frame rate [115], a function to the object size [C7] or online learned. We set the standard deviations for the dimensions of the object state, σ^w and σ^h, as constants, and the standard deviations for the object location and its velocity, σ^u, σ^v, $\sigma^{\bar{u}}$ and $\sigma^{\bar{v}}$, as function of the object size on the image plane to account for the foreshortening, as discussed previously in Sec. 4.2, (i.e. an object closer to the camera has higher uncertainty on its location compared than another object that is farther from it) and as a function of the video frame rate to account for frame rate variations [115] (i.e. an object seen by a lower frame-rate video has higher uncertainty on its location compared to when seen by a higher frame-rate video).

The posterior can be approximated as a weighted average of the particles describing the object λ, with weights π_k^i [17] (Eq. 4.23). The observation model estimates the importance of each particle towards the estimation of the state. This importance is encoded in the particle weight that, assuming that the state components are independent from each other, we update as in Eq. 4.21. We then resample the particles so that those with larger weights are maintained and replicated, whereas those with lower weights are removed. Specifically, we perform multi-stage resampling [115], which uses multiple stages depending on the frame where the particles were created. This ensures that particles recently created for newly appearing objects, and likely to have a lower weight than older ones, are not removed.

4.4 The audio-visual quadcopter dataset

We present an audio-visual dataset recorded outdoors from a UAV quadcopter. The dataset includes a scenario that is suitable for the evaluation of sound source localisation and sound enhancement with up to two static sources, and a scenario for source localisation and tracking with a moving sound source. These sensing tasks are made challenging by the strong and time-varying ego-noise generated by the rotating motors and propellers. The dataset was collected using a small circular array with 8 microphones and a camera mounted on the quadcopter. The camera view was used to facilitate the annotation of the sound-source positions and can also be used for multi-modal sensing tasks. For the annotation of the dataset, we used the audio-visual calibration proposed in Sec. 3.1.2. We made the dataset available to the research community[1].

Audio-visual sensing from a quadcopter is of interest for applications such as search and rescue, human-drone interaction and multimedia broadcasting [21, 72, 73, 79, 215]. However, the quality of sounds recorded from a quadcopter is poor due to the strong and time-varying ego-noise generated by the rotating motors and propellers, which cause extremely low signal-to-noise ratios, e.g. smaller than -15 dB [168, 190]. Moreover, the movement of the quadcopter itself and natural wind further complicate the analysis of sounds emitted by sources in the environment.

A number of microphone-array algorithms have been proposed to address these challenges for sound source localisation [43, 78, 79, 134, 136, 137] and enhancement to extract object sounds masked by the strong ego-noise [18, 43, 71, 152, 188, 190, 215]. As quadcopters are generally equipped with an onboard camera, audio-visual processing methods can use the visual information to facilitate the localisation of the sound source and the enhancement of the sound of object sources [152, C6].

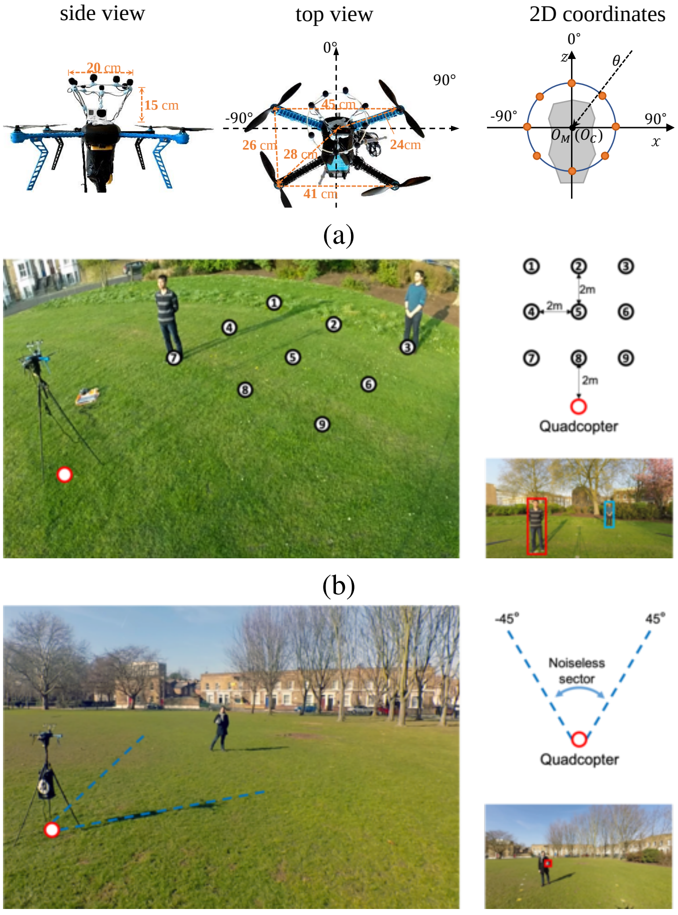
Indoor datasets with multi-channel sound recordings captured from a UAV platform are becoming available for sound source localisation [113] and sound enhancement [133]. DREGON was captured with an 8-channel cube-shape microphone array mounted on a Mikrokopter drone [113]. AIRA-UAS was captured with an 8-channel circular microphone array mounted on three types of drones, a DJI Matrice 100, a 3DR Solo and a Parrot Bebop 2 [133]. These two datasets were collected indoors to facilitate the annotation using external positioning systems.

In this section we present AVQ, the first annotated outdoor *Audio-Visual* dataset from a *Quadcopter* drone. The dataset can be used for audio-visual and audio-only tasks such as sound enhancement, sound source localisation and tracking. We use an 8-element microphone array mounted on a quadcopter to record sounds in the environment as well as a camera to allow multi-modal tasks. The dataset consists of two subsets that capture up to two *static* sound sources emitting sound in front of the drone; and a *moving* sound source (see Fig. 4.16).

We placed the recording prototype on a tripod at a height of 1.8 m in a park to record speech as sound played by a loudspeaker carried by a person walking in front of the UAV (Fig. 4.17).

The dataset consists of the $\mathcal{S}1$ and $\mathcal{S}2$ subsets, with natural and composite scenarios (see Table 4.1). $\mathcal{S}1$ includes up to two sound sources at fixed locations, whereas $\mathcal{S}2$ includes a moving sound source. In the *natural* scenario, the object sound and the ego-noise are recorded simultane-

[1] `http://cis.eecs.qmul.ac.uk/projects/avq/`



(a)

(b)

(c)

Figure 4.16: The AVQ recording setup. (a) Side and top view of the audio-visual sensing platform; and 2D-coordinate system. O_M and O_C denote the centres of the microphone array and of the camera in the 2D plane, respectively. (b) Recording environment for Subset $\mathcal{S}1$: two people talk from nine locations. (c) Recording environment for Subset $\mathcal{S}2$: a loudspeaker is carried by a person walking in front of the drone. The left and right panels of (b) and (c) show the overall scene and the view from the onboard camera, respectively.

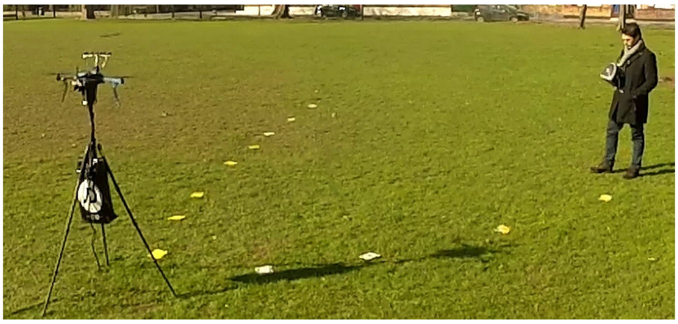


Figure 4.17: Experimental setup. A loudspeaker is carried by a person who walks in front of the drone that is placed on a tripod. The noiseless sector is indicated with yellow marks on the ground.

ously. In the *composite* scenario, the object sound and the ego-noise are recorded separately, thus allowing one to evaluate the performance at different input signal-to-noise ratios (SNRs) and to

Table 4.1: AVQ dataset: specifications. KEY: Sub, Subset; Seq, Sequence; Mod., Modality; Dur, Duration; VG, Video ground-truth; VAD, voice activity detection; A, Audio-only; AV, Audio-visual; EO, ego-noise only; SO, speech only; MIX, mixture; cons, constrained area; uncons, unconstrained area; and misc, miscellaneous.

Sub	Seq	Mod	Dur	VG	VAD	Type	Drone	Source
$\mathcal{S}1$	seq1	A	120s			EO	constant (50%)	/
	seq2	A	120s			EO	constant (100%)	/
	seq3	A	40 s			EO	constant (150%)	/
	seq4	AV	797s	✓	✓	SO	muted	2 sources 9 locations
	misc	microphone location, AV calibration parameters						
$\mathcal{S}2$	seq1	A	210s			EO	constant (100%)	/
	seq2	A	214s			EO	dynamic	/
	seq3	AV	215s	✓	✓	SO	muted	cons.
	seq4	AV	217s	✓	✓	SO	muted	uncons.
	seq5	AV	303s	✓	✓	MIX	constant (100%)	cons.
	seq6	AV	271s	✓	✓	MIX	constant (100%)	uncons.
	seq7	AV	258s	✓	✓	MIX	dynamic	cons.
	seq8	AV	249s	✓	✓	MIX	dynamic	uncons.
	misc	microphone location, AV calibration parameters						

compute the output SNR after processing [191].

In $\mathcal{S}1$, two people (the sound sources) talk at nine predefined locations in front of the drone (Fig. 4.16 b). The distance between these locations and the drone varies between 2 m and 6 m. We record only composite scenarios, i.e. the clean speech and the ego-noise are recorded separately. When recording the ego-noise, the quadcopter operates at 50%, 100% or 150% of the power level of the hovering state. When recording speech, the two people talk in turns for about 40 s each and then move to the next location. Video frame samples are shown in Fig. 4.18a. In $\mathcal{S}2$, a loudspeaker (the sound source) playing speech is carried by a person (Fig. 4.16 c). As the relative location of the microphone array and the motors and propellers is fixed, the ego-noise tends to arrive from the side closer to the motors (back side of the array), thus creating a sector with lower ego-noise (the front of the array). This allows us to identify a *noiseless sector* $[-45°, 45°]$ where an object sound can be more easily detected [190]. We record natural and composite scenarios. The drone operates either with a constant hovering power or with a time-varying power between 50% and 150% of the hovering state. The loudspeaker moves either in a *constrained* area (inside the noiseless sector) or in an *unconstrained* area (in front of the drone). The distance between the loudspeaker and the drone varies between 2 m and 6 m. Each sequence lasts for about 3 minutes. Video frame samples of the sound source in the noiseless sector (Fig. 4.18b) and outside the noiseless sector (Fig. 4.18c).

Table 4.1 summarises the specifications of the AVQ dataset. The audio is in WAV format

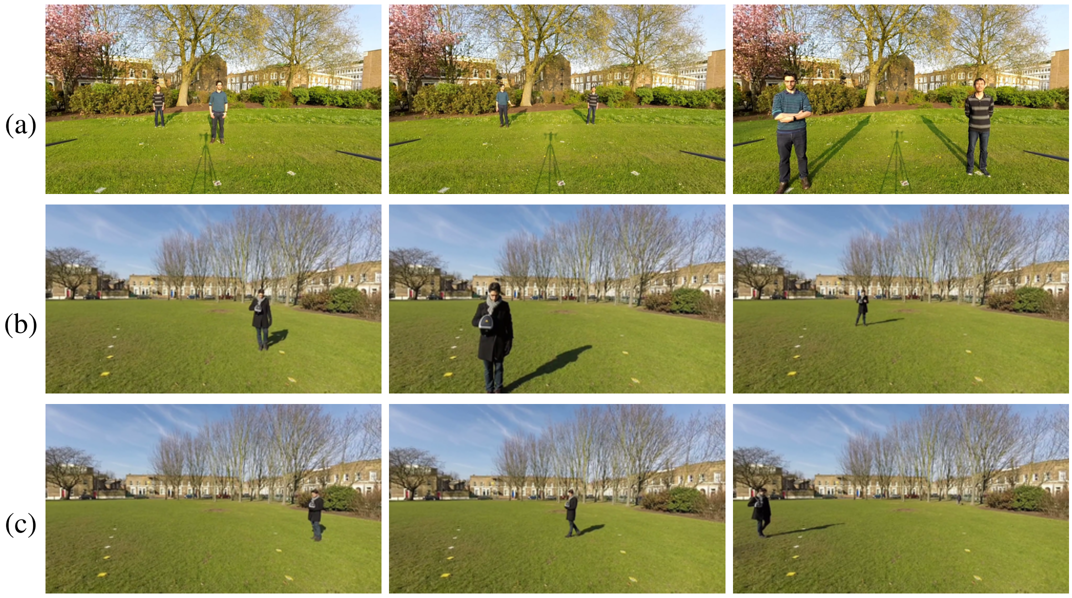


Figure 4.18: Sample frames of (a) $\mathcal{S}1$-seq4, with two alternatively-talking sound sources; (b) $\mathcal{S}2$-seq5 with the sound source moving in a constrained manner; and (c) $\mathcal{S}1$-seq6 with the sound source moving in an unconstrained manner.

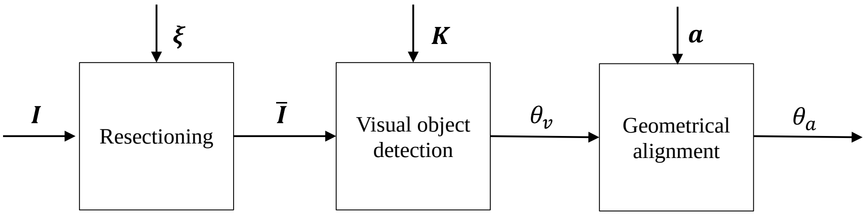
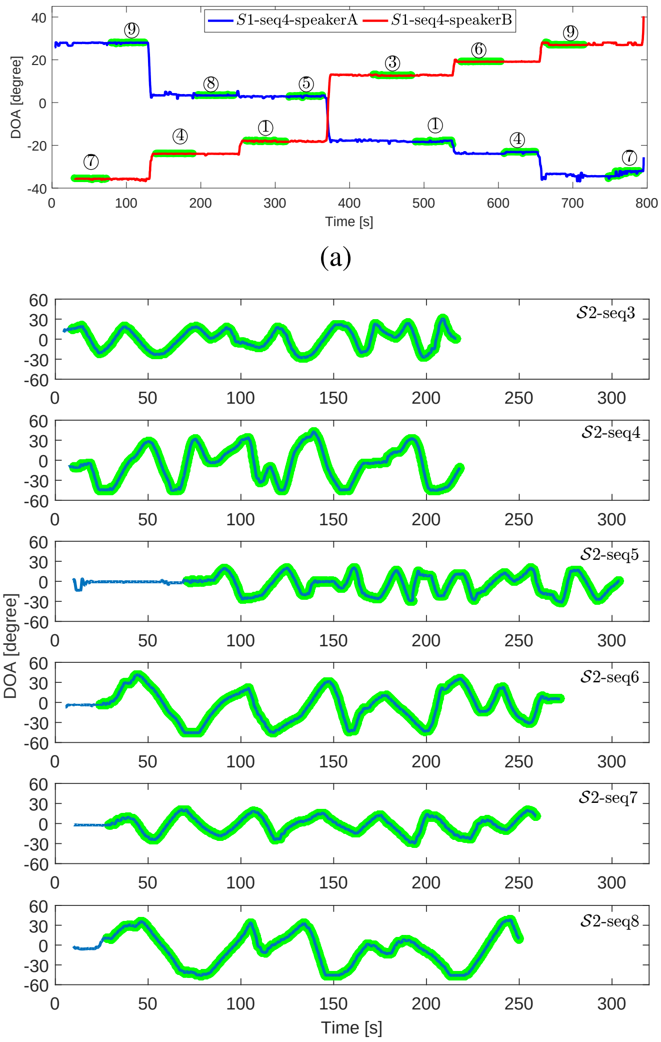


Figure 4.19: Audio-visual calibration. We undistort the original image $\mathbf{I}$ using the lens distortion parameter ξ (Eq. 3.2) and detect visual objects in the undistorted image $\widetilde{\mathbf{I}}$. From the location of the visual object we compute the visual angle θ_v using the camera intrinsic parameter $\mathbf{K}$ (Eq. 3.4) and convert to the audio angle θ_a using the calibration parameter $\mathbf{a}$ (Eq. 3.3).

with sampling rate 44.1 kHz. The video is in MP4 format with frame rate 30 fps, resolution 1920×1080 for $\mathcal{S}1$ and 1280×720 for $\mathcal{S}2$, and wide field of view (i.e. 70 vertical degrees and 120 horizontal degrees before undistortion). The total duration of the recordings is about 50 minutes.

To obtain the ground-truth locations of the sound source, we make use of the onboard camera. The video ground truth of the sound source locations makes it possible to evaluate source localisation and tracking performance in both natural and composite scenarios. We calibrate the audio and vision streams as described in Sec. 3.1. Fig. 4.19 illustrates the calibration steps.

When the multi-modal streams are calibrated, for $\mathcal{S}1$ we use a person detector [154] and for $\mathcal{S}2$ we use a visual marker to assist the loudspeaker detection (see the example in the right panels of Fig. 4.16b and c). Fig. 4.20a and b depict the video ground-truth locations, θ_v, and the voice



(a)

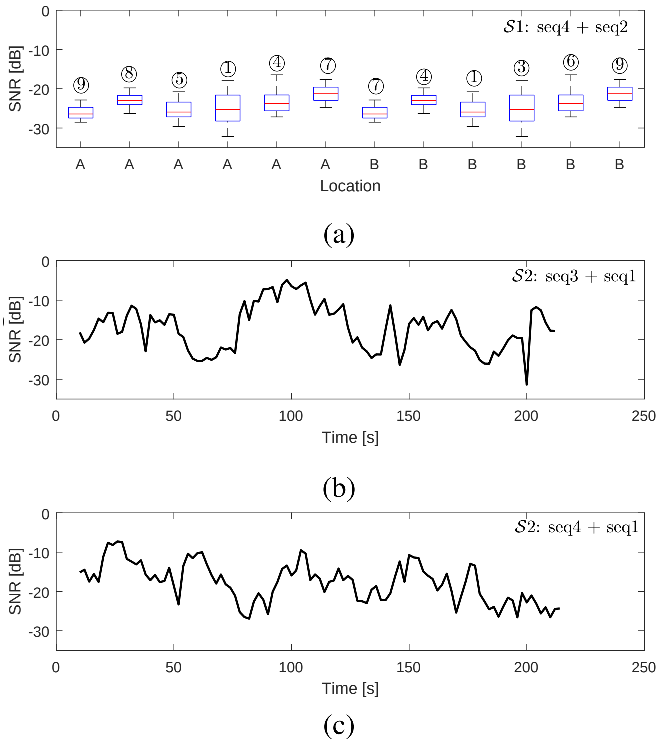
(b)

Figure 4.20: AVQ dataset: video ground-truth trajectory of the sound sources. Green thick lines: voice activity periods; circled numbers: sound source locations shown in Fig. 4.16b. (a) Subset $\mathcal{S}1$. (b) Subset $\mathcal{S}2$.

activity detector (VAD) information of the sound source for $\mathcal{S}1$ and $\mathcal{S}2$, respectively.

Fig. 4.21 illustrates the input SNR of some composite sequences generated by mixing clean speech with the ego-noise (when the drone is operating at hovering state). The SNR over a segment is defined as the power ratio between the clean speech and the ego-noise [191].

We use this dataset in Sec. 4.6.4 to evaluate the proposed SOT. Also, we use this dataset to perform localisation and enhancement of multiple sound sources in [152].



(a)

(b)

(c)

Figure 4.21: The SNR of composite sequences generated by mixing a clean speech with the ego-noise (drone operating at hovering state). (a) $\mathcal{S}1$: seq4 + seq2. (b) $\mathcal{S}2$: seq3 + seq1. (c) $\mathcal{S}2$: seq4 + seq1.

4.5 Confidence intervals for tracking performance scores

In this section, we first describe existing procedures for annotating the ground truth of a dataset (Sec. 4.5.1); then, we introduce a novel idea for estimating the confidence intervals of the annotations for a given evaluation metric and dataset (Sec. 4.5.2); besides, we show how the uncertainty in the annotations impacts on tracking benchmarks (Sec. 4.5.3).

4.5.1 Ground-truth annotations

As discussed in Sec. 2.6, tracking performance is commonly evaluated using discrepancy measures (e.g. CLEAR metrics [26]) that compare the tracking results with respect to a set of manual annotations, known as ground truth (GT), generated by for example selecting the coordinates and size of a rectangle to define an object on the image plane.

The generation of GT for MOT is needed per each object of interest in each frame of the video sequence. Ideally all frames should be annotated (Full Ground Truth) by multiple raters and a consolidated GT should be generated after analysis of their annotation results [183]. This become a tedious and time-consuming task when the volume of data to be annotate increases.

Table 4.2: Datasets for Multiple target tracking, their annotation tool and rules (including key-frame based annotation and linear interpolation). KEY: NA, not available. *, ADL-Rundle-xxx and Venice-xxx annotation generated with VATIC [183].

Ref	Dataset	Annotation tool	Key-frame based	Linear Interpolation
[94]	MOTB15*	VATIC	NA	✓
[125]	MOTB16	private	NA	✓
[5]	MOTB17	private	NA	✓
[55]	PETS2009**	NA	✓	✓
[3]	CAVIAR	CaviarGui		
[60]	KITTI	Mechanical Truck	NA	NA
[4]	i-LIDS	VIPER	✓	✓
[52]	ETH**	NA	✓	✓
[13, 14]	TUD**	NA	✓	✓

For example, the MOTB16 has almost 300k annotated objects. This task can become infeasible in some large scale datasets where millions of trajectories need to be annotated [10]. To speed-up the annotation process, only a set of *key frames* is generally manually annotated, while the annotation of the remaining frames is derived by interpolation (Interpolated Ground Truth).

Most of state-of-the-art datasets have been annotated and linearly interpolated for reducing the human workload. Table 4.2 shows MOT datasets and how their GT have been annotated. MOTB15 [94] uses already existing GT for part of the videos and they annotated using VATIC annotation tool 6 new videos from ADL-Rundle and Venice datasets. They manually annotate certain key frames and the intra-key frames are automatically annotated using a interpolation technique not unveil in the publication. MOTB16 [125] and MOTB17 uses a private annotation tool and do not provide any further information about their annotation policy. The GT for other well-known datasets such as PETS2009 [55], ETH [52] and TUD [13, 14] is available in [124]. Their GT uses linear interpolation to estimate the annotations between manually annotated key frames. CAVIAR dataset [3] has been manually annotated frame by frame with CaviarGui and therefore it is assumed that no interpolation has been applied. Lastly, i-LIDS dataset [4] has been manually annotated with ViPER [123] at certain key frames and the rest of annotations have been linearly interpolated as well.

Dataset annotation for MOT is a very time consuming and tedious task. State-of-the-art datasets have commonly hundred of thousands of annotations and the requirement of large-scale annotated data for deep neural network algorithms make essential the use of annotation tools.

Several works have been implemented to improve the annotation efficiency while maintaining

Table 4.3: State-of-the-art annotation schemes classified on base to the annotation shape options, interpolation techniques available and robustness to camera motions. Key: P, point; R, rectangle; OR, oriented rectangle; PO, polygon; C, circle; E, ellipse; RCM, robust to camera motion; and KTRA, keyframe-based tracking for rotoscoping and animation.

Ref	Name	Object shape						Interpolation					RCM
		P	R	OR	PO	C	E	2D Lin.	3D Lin.	Track.	Lear.	Optimis.	
[123]	ViPER	✓	✓	✓	✓	✓	✓	✓					
[6]	KATRA				✓					✓			
[219]	LabelMe				✓				✓				✓
[11]	FlowBoost		✓								✓		
[183]	VATIC		✓					✓		✓	✓	✓	
[28]	iVAT		✓		✓		✓	✓			✓		
[29]	ViTBAT	✓	✓				✓	✓					

the accuracy of the annotations. These works can be classified regarding the required human-interaction as *manual*, *semi-automatic* and *automatic*. Manual approaches require the annotation of all object at all frames one by one. Semi-automatic methods allow to the annotator (human who do the manual annotations) to decrease the number of required annotation by using multiple interpolation methods such as *linear interpolation*, *tracking*, *learning* or *optimisation* learning methods [28, 29, 123, 183, 219]. Automatic annotation approaches require the annotation of each object on one/multiple frame(s) and relies on automatic methods, as previously mentioned, to complete the rest of annotations [28]. Existing annotation tools are summarised in Table 4.3. Next, we briefly discuss their features focusing on their interpolation approach.

The most common approach is the semi-automatic annotation using 2D linear interpolation. These approaches require to manually annotate some key frames that can be selected in base to certain patterns such as arbitrary annotator decisions or equally-spaced key frames [183]. Intra-keyframe annotations are then generated by linearly interpolating the manual annotations [28, 29, 123, 183]. These approaches are extremely useful as they enormously relief the workload for the annotator while maintaining a good accuracy in most of the cases [183]. This make feasible the annotation of large-scale datasets. Moreover, these approaches are very convenient for some special cases such as occlusions, where the objects need to be annotated but the direct annotation is unfeasible to be annotated as the object is not visible (or fully visible). Most of the state-of-the-art annotation tools use this idea due to its simplicity and low computational complexity.

VIPER [123] can be considered as the first annotation system optimised for spatial labelling that uses linear interpolation on a key-frame-basis approach. This tool has been widely used in the literature due to its simplicity and flexibility as it allows to annotate several object shapes including point, rectangle, oriented rectangles, polygon, circle and ellipse. Similar works have been proposed focusing on particularities on the annotation such as ViTBAT [29], that focuses on the efficient individual and group state annotation whilst annotating their behaviour, or iVAT [28], that proposes key-frame semi-automatic annotation as well as a completely automatic mode that requires the annotation of each object on a frame that will be used as a template to estimate the object location in the remaining frames using a cascade of boosted classifiers. The non-linear projective transformation from 3D world to the 2D image plane produces that the simplification made on these works cannot cope with some object motions. Yuen et al [219] model the linear interpolation in the 3D world instead of on the image plane. This novel interpolation method allows to achieve more accurate object annotation compared with 2D linear interpolation for objects that can be considered to move linearly in 3D world (e.g. person or car moving on a straight line). Moreover, this work performs an image stabilisation process to better compensate the motion of moving cameras.

To cope with object motions that exhibit non-linear motions such as turning cars or flying basketballs, other methods have been proposed to fill the intra-keyframe annotations based on tracking, learning or optimisation. VATIC annotation tool proposed in [183] to annotate equally-spaced key frames and to generate missing annotations via linear interpolation, learning object templates, constrained tracking or efficient optimisation. Agarwala et al [6] propose a semi-automatic annotation technique by joining automatic tracking annotation with user interaction for rectification. Ali et al [11] presented a tool that can annotate videos from a sparse set of keyframe annotations by alternatively training an appearance-based detector and a convex time-based regularisation. As a drawback, these methods are computationally more expensive than performing linear interpolation and are prone to typical computer vision challenges such as occlusions, camera motion, motion blur or similar appearance of objects.

Most of the available MOT datasets provide the annotation for all objects in the dataset [55, 94, 125]. The annotation task is done using private [94, 125] or public annotation tools [28, 29, 183].

Concepts discussed in this section such as linear interpolation for semi-automatic annotation of MOT datasets are further discussed in the next section where we propose a method for automatically quantifying the inaccuracies in annotation.

4.5.2 Confidence intervals estimation

The objective evaluation of trackers quantifies the discrepancy between tracking results and a manually annotated ground truth. As generating ground truth for a video dataset is tedious and time-consuming, often only keyframes are manually annotated. The annotation between these keyframes is then obtained semi-automatically, for example with linear interpolation. This approximation has two main undesirable consequences: first, interpolated annotations may drift from the actual object, especially with moving cameras; second, trackers that use linear prediction or regularise trajectories with linear interpolation unfairly gain a higher tracking evaluation score. This problem may become even more important when semi-automatically annotated datasets are used to train machine learning models. To account for these annotation inaccuracies for a given dataset, we identify objects whose annotations are interpolated and propose a simple method that analyses existing annotations and produces a confidence interval to complement tracking scores. These confidence intervals quantify the uncertainty in the annotation and allow us to appropriately interpret the ranking of trackers with respect to the chosen tracking performance score.

Let $\tilde{\mathbb{X}}$ be the GT annotation for a generic dataset defined as in Eq. 2.6. $\tilde{\mathbb{X}}$ may contain manual and interpolated annotations, and can be decomposed as $\tilde{\mathbb{X}} = \tilde{\mathbb{X}} \cup \hat{\mathbb{X}}$, with $\tilde{\mathbb{X}} \cap \hat{\mathbb{X}} = \emptyset$, where $\tilde{\mathbb{X}} = \{\tilde{\mathbf{x}}_{k,\lambda}\}$ contains manually annotated objects and $\hat{\mathbb{X}} = \{\hat{\mathbf{x}}_{k,\lambda}\}$ contains automatically generated annotations, created for example through interpolation. If all the annotations are produced manually, then $\hat{\mathbb{X}} = \emptyset$ and $\tilde{\mathbb{X}} = \tilde{\mathbb{X}}$. We term $\tilde{\mathbb{X}}$ manual ground truth (MGT) and $\hat{\mathbb{X}}$ interpolated ground truth.

Let us assume that linear interpolation was used to generate $\tilde{\mathbb{X}}$ for a dataset. Our aim is to identify the elements of $\tilde{\mathbb{X}} = \bigcup_{\lambda \in \Lambda_k} \tilde{\mathbb{X}}_\lambda$. These elements are likely to have non-zero acceleration for all the components of the state:

$$\tilde{\mathbb{X}}_\lambda = \left\{ \tilde{\mathbf{x}}_{k,\lambda} : \mathbf{z}_{uu}, \mathbf{z}_{vv}, \mathbf{z}_{ww}, \mathbf{z}_{hh} \neq 0 : k = 0 \ldots \tilde{K}_\lambda - 1 \right\}, \tag{4.32}$$

where $\mathbf{z}_{ii}$ is the second partial derivative (i.e. acceleration) of component i, and $\tilde{K}_\lambda = |\tilde{\mathbb{X}}_\lambda| \leq K_\lambda$ is the cardinality of $\tilde{\mathbb{X}}_\lambda$.

From the identified $\tilde{\mathbb{X}}_\lambda$ we generate interpolated versions, $\mathbb{X}_\lambda^\beta$, with different decimation factors, $\beta \geq 2$, through a decimation-interpolation procedure:

$$\mathbb{X}_{\lambda}^{\beta} = \left\{ \mathbf{x}_{k,\lambda}^{\beta} = \tilde{\mathbf{x}}_{i\beta,\lambda} + j\Delta_{i,\lambda} : k = 0 \dots \tilde{K}_{\lambda} - 1; i = 0, \dots, \left\lfloor \frac{\tilde{K}_{\lambda}}{\beta} - 1 \right\rfloor; j = 0 \dots \beta - 1 \right\}, \tag{4.33}$$

where $\Delta_{i,\lambda} = \frac{\tilde{\mathbf{z}}_{(i+1)\beta,\lambda} - \tilde{\mathbf{z}}_{i\beta,\lambda}}{\beta}$ and $\lambda \in \Lambda_k$. Note that $\mathbb{X}_{\lambda}^{\beta}$ is composed of manual annotations (when $j = 0$) and linearly interpolated annotations (when $j \neq 0$); its cardinality is equal to that of $\tilde{\mathbb{X}}_{\lambda}$, i.e. $|\mathbb{X}_{\lambda}^{\beta}| = |\tilde{\mathbb{X}}_{\lambda}| = \tilde{K_{\lambda}}$; and $\lfloor\,\rfloor$; is the floor operator that rounds a number to the smaller integer.

To empirically estimate the uncertainty introduced by $\mathbb{X}_{\lambda}^{\beta}$ when evaluating tracking results, we consider a generic tracking performance measure, $s(\cdot,\cdot)$, which allows us to compare $\mathbb{X}_{\lambda}^{\beta}$ against its corresponding $\tilde{\mathbb{X}}_{\lambda}$ as:

$$\alpha_{s,\beta} = \frac{1}{|\Lambda_k|} \sum_{\lambda \in \Lambda_k} s(\tilde{\mathbb{X}}_{\lambda}, \mathbb{X}_{\lambda}^{\beta}). \tag{4.34}$$

The value of $\alpha_{s,\beta}$ allows us to define the confidence interval for $s(\cdot,\cdot)$: a tracker with the same tracking results as a $\mathbb{X}_{\lambda}^{\beta}$ should subtract (up to) $\alpha_{s,\beta}$ to the final score; whereas a tracker with the same tracking results as a given MGT should add (up to) $\alpha_{s,\beta}$ to the final score, if it is assessed using $\mathbb{X}_{\lambda}^{\beta}$.

To apply the confidence interval on the performance scores for a specific dataset, we *map* $\alpha_{s,\beta}$ to the amount of interpolation detected in the dataset.

As an example, let us quantify the amount of linearly interpolated annotations in MOTB15 [94] and MOTB16 [125]. Using the definition in Eq. 4.32, the MOTB16 training dataset results in having 39.7% of linearly interpolation annotations (Table 4.4). This approximately corresponds to a decimation factor of $\beta = 3$. We assume that the test dataset and the training dataset have a similar amount of linearly interpolated annotations, as the same annotation policy was used for both datasets [125]. MOTB16 uses three times more interpolation than MOTB15. In MOTB16, static-camera videos have a higher percentage of interpolated annotations than moving-camera videos. Surprisingly, in MOTB15 moving-camera videos have a higher percentage of interpolated GT annotation than static-camera videos.

To visualise the GT drifts caused by interpolation, we annotated frame-by-frame object 32 in MOTB16-02 (static camera) and object 4 in MOTB16-10 (moving camera), totalling 685 annotations. We refer to these annotations as *ideal* GT. Fig. 4.22 shows object 32 of MOTB16-02 (top) and object 4 of MOTB16-10 (bottom), and compares the MGT (red) against its decimated

Table 4.4: Percentage of linearly interpolated annotations detected in the MOTB15 and MOTB16 datasets using Eq. 4.32.

Camera motion	**MOTB15**	**MOTB16**
Static	6.2	52.7
Moving	17.0	12.7
Overall	11.0	39.7

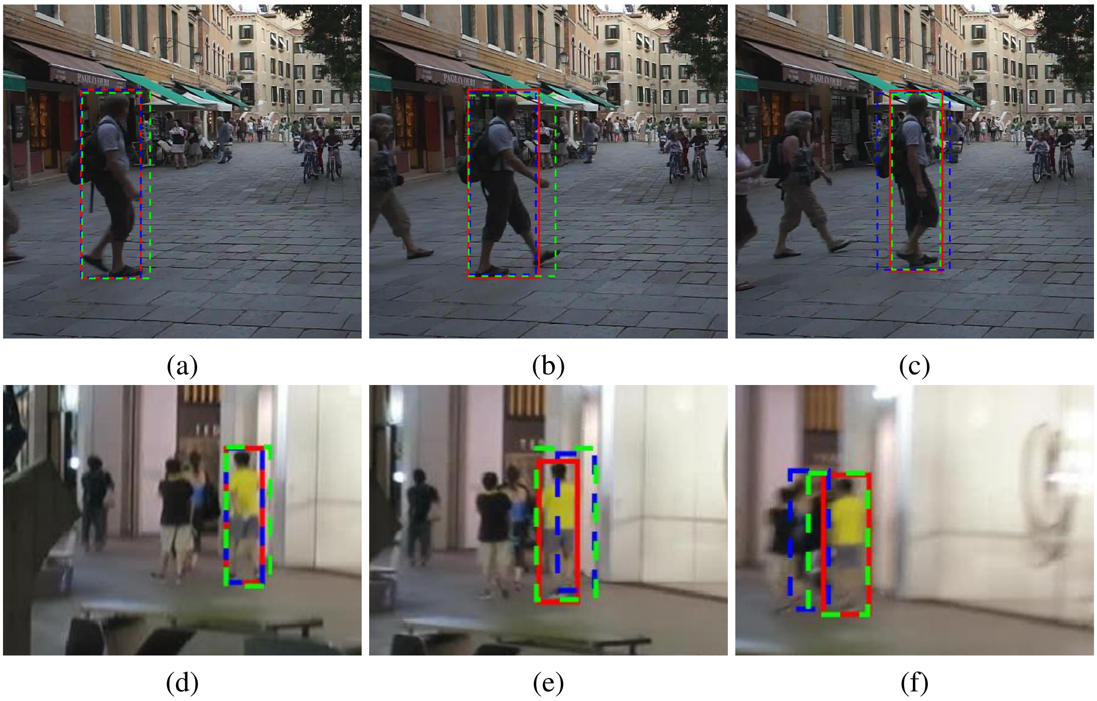
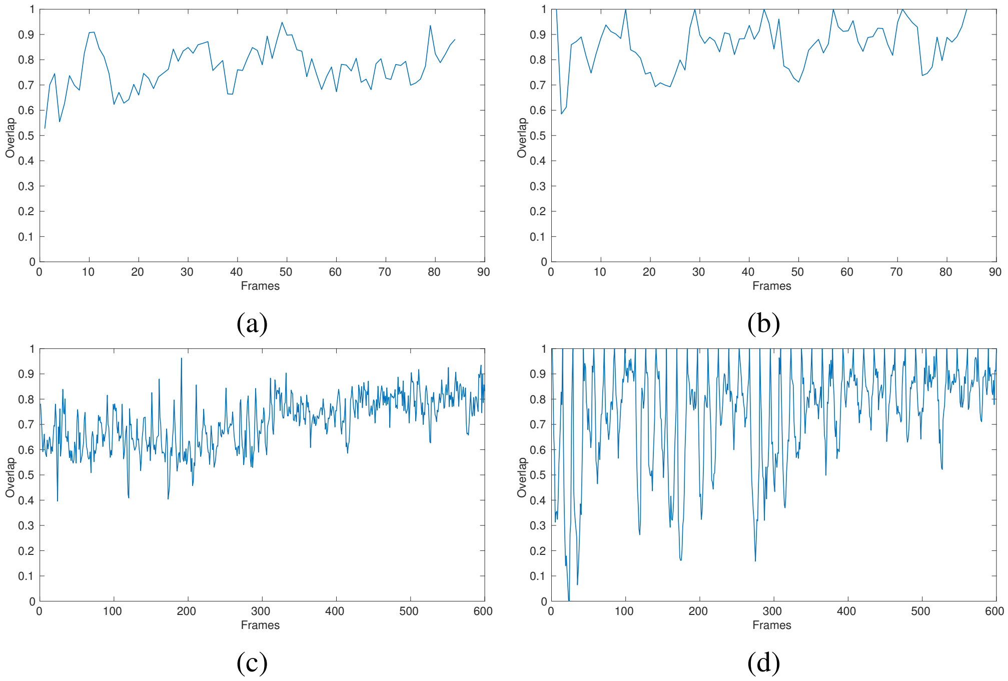


Figure 4.22: Comparison of manually annotated GT (red rectangle), interpolated generated GT with $\beta = 15$ (blue), and MOTB GT (green) for object 32 of MOTB16-02 (static camera) at frames 331, 338 and 352 (a-c) and for object 4 of MOTB16-10 (moving camera) at frames 1, 8 and 12 (d-f). Source images from [125].

version, $\mathbb{X}_{\lambda}^{\beta}$, with decimation factor $\beta = 15$ (blue) and the GT provided with the dataset (green), MOTB GT. The more noticeable drift occurs when the camera moves (Fig. 4.22e-f).

We quantify the overlap that the ideal GT produces against its interpolated versions and MOTB GT. We calculate the overlap between a manual annotation, $\tilde{\mathbf{x}}_{k,\lambda}$, and $\mathbf{x}_{k,\lambda}^{\beta}$, as: $\omega_k^{\lambda} = \frac{\tilde{\mathbf{x}}_{k,\lambda} \cap \mathbf{x}_{k,\lambda}^{\beta}}{\tilde{\mathbf{x}}_{k,\lambda} \cup \mathbf{x}_{k,\lambda}^{\beta}}$.

The effect of different interpolated GT versions on tracking evaluation scores is shown in Fig. 4.23. In the static-camera example (Fig. 4.23a-b), the overlap decreases moderately up to 0.5 (i.e. 50%). In the moving-camera example (Fig. 4.23c-d), the overlap decreases up to having frames with *no overlap* between interpolated GT versions and the ideal GT.



(a)

(b)

(c)

(d)

Figure 4.23: Comparison of overlaps between GTs for MOTB16-02 static-camera video (top row) and for MOTB16-10 moving-camera video (bottom row). Left column: overlap between ideal GT and MOTB GT. Right column: overlap between MGT and its interpolated generated version ($\beta = 15$).

4.5.3 Impact on ranking trackers

In this section, we analyse the impact of the interpolated GT on the ranking of trackers in a specific benchmark, MOTB16 [125]. In order to account for the uncertainty introduced by the interpolated annotation for a given dataset, we define confidence intervals to complement performance scores.

We first identify frames and objects for which no interpolation is used in the publicly available GT, $\tilde{\mathbb{X}}$, thus generating $\tilde{\mathbb{X}}$. Then, we generate multiple interpolated versions, $\hat{\mathbb{X}}_\beta = \bigcup_{\lambda \in \Lambda_k} \hat{\mathbb{X}}^\lambda_\beta$, with different decimation factors, $\beta \in \{3,6,9,12\}$. Finally, we compare each $\mathbb{X}^\beta_\lambda$ against the MGT, $\tilde{\mathbb{X}}$, using the specific performance scores to define the confidence intervals to be applied to each score (Eq. 4.34).

As discussed in Sec. 2.6.4, we use Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) as performance scores. MOTA and MOPT are defined in Eq. 2.14 and 2.15.

For MOTA we define: $s(\cdot,\cdot) = 100 - MOTA$. Likewise for MOTP. Table 4.5 shows that in the static-camera example (object 32 in MOTB16-02) both interpolated versions obtain full MOTA, whereas MOTP considerably decreases due to interpolation, due to its direct relation

Table 4.5: Evaluation tracking score with a GT annotated frame-by-frame (*ideal* GT) against its interpolated generated GT, $\mathbb{X}_{\lambda}^{\beta}$, with decimation factor ($\beta = 15$), and the MOTB GT annotation.

Camera motion	**Sequence**	**GT**	**MOTA**	**MOTP**
Static	MOTB16-02-id32	$\mathbb{Z}_{\beta=15}^{\lambda=32}$	100	85.70
		MOTB	100	76.91
Moving	MOTB16-10-id4	$\mathbb{Z}_{\beta=15}^{\lambda=4}$	72.00	69.23
		MOTB	97.33	70.36

Table 4.6: Uncertainties in MOTA and MOTP for MOTB16 produced by different decimation factors (β).

β	MOTA confidence	MOTP confidence
3	0.22	3.14
6	0.56	8.68
9	3.74	13.41
12	11.27	17.05

with the overlap. In the moving-camera example (object 4 of MOTB16-10) MOTA and MOTP differ from the ideal result. For example, when object 4 of MOTB16-10 is evaluated with the ideal GT obtains a MOTA of 72 (penultimate row in Table 4.5). This means that no tracker with MOTA results closer than $28 = 100 - 72$ to another tracker can be confidently said to outperform the other based on MOTA.

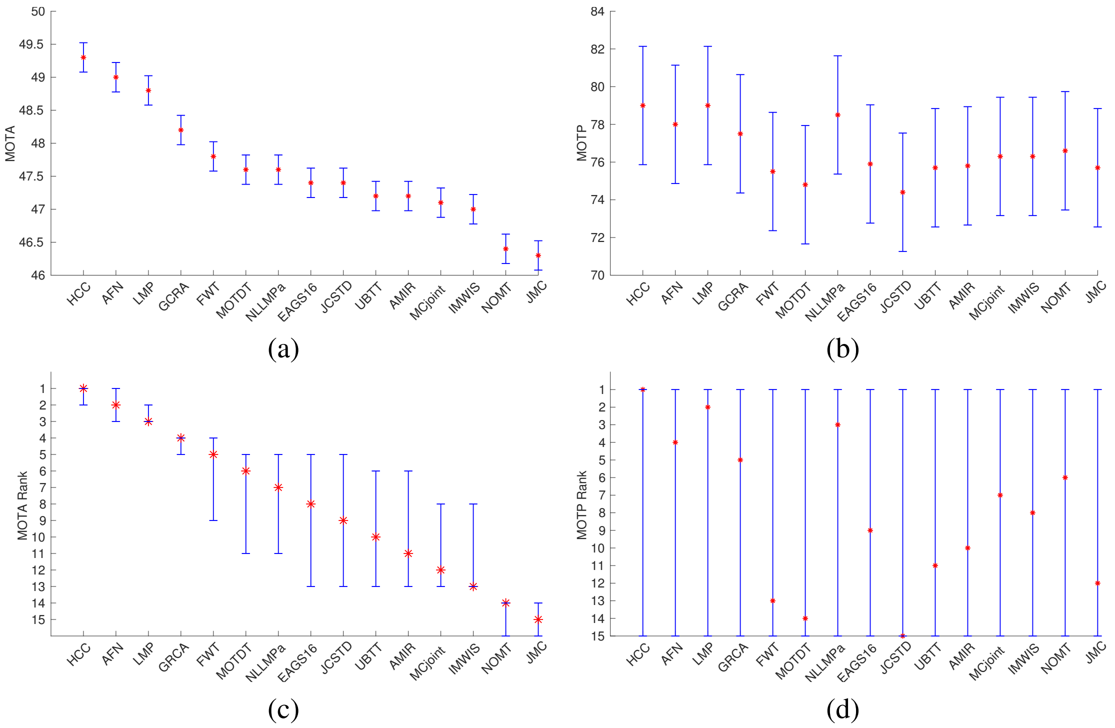
Table 4.6 shows how the confidence on the performance score varies with different decimation factors, β. In order to characterise MOTA and MOTP through the overlap only, we assume for simplicity that the number of IDS for all trackers is null.

To conclude, let us consider the TOP-15 trackers sorted by MOTA that use public detections on the MOTB16 test dataset. For this dataset, the estimated *MOTA uncertainty* is 0.22 and *MOTP uncertainty* is 3.14 (first row Table 4.6). These values of $\alpha_{s,\beta}$ for the two scores allow us to analyse the impact of the public GT annotation of MOTB16.

Fig. 4.24a-b shows MOTA and MOTP results obtained for each tracker[2], whereas Fig. 4.24c-d shows the ranking of trackers based on MOTA and MOTP. The estimated MOTA and MOTP confidence intervals are shown as bars. Note that MOTP is very sensitive to GT interpolation as small overlap variations influence the measure (Eq. 2.15). MOTA is instead less sensitive as it depends on the number of false positives and false negatives, which vary only when the overlap becomes smaller than 50%.

In summary, no TOP-15 tracker can be confidently assigned to a specific rank in the MOTB16

[2] `https://motchallenge.net/results/MOT16/` Last accessed on 27th June 2020.



(a)

(b)

(c)

(d)

Figure 4.24: TOP-15 performing trackers using public detections in MOTB16 according to MOTA measure. Red crosses indicate the MOTB Challenge measure (top row) and rank (bottom row). Blue bars translate the confidence interval (top row) into the ranking uncertainty (bottom row).

benchmark, as neighbouring trackers are within their MOTA uncertainty ranges. Moreover, there is no significant difference among any of the TOP-15 trackers in terms of MOTP, i.e. even the first and fifteenth tracker cannot be confidently ranked relative to each other in terms of MOTP.

4.6 Experimental validation

4.6.1 Experimental setup

For the evaluation of this chapter, we use two datasets named multiple object tracking benchmark (MOTB) dataset and the self-collected audio-visual quadcopter (AVQ) dataset. The former one is used for the experimental validation of multi-detector fusion (Sec. 4.6.2), object motion prediction described in (Sec. 4.6.3) and multi-object tracking from both static (Sec. 4.6.5) and moving (Sec. 4.6.6) cameras The latter one is used for the experimental validation of single-object tracking (Sec. 4.6.4). Next, we introduce both datasets.

The multiple object tracking benchmark datasets

The release of the MOTB15 [94], MOTB16 [125] and MOTB17 [5] have facilitated the improvement of the state of the art in MOT in the recent years. These datasets include several frame rates,

Table 4.7: Statistics for MOTB15 [94], MOTB16 [125] and MOTB17 [5] datasets.

Dataset	Characteristics	MOTB15	MOTB16	MOTB17
Training	Length (frames)	5,503	5,316	5,316
	Identities	500	517	546
	Annotations	39,905	110,407	112,297
	Average density	7.3	20.8	21.1
Test	Length (frames)	5,783	5,919	5,919
	Identities	721	759	785
	Annotations	61,440	182,326	188,076
	Average density	10.6	30.8	31.8

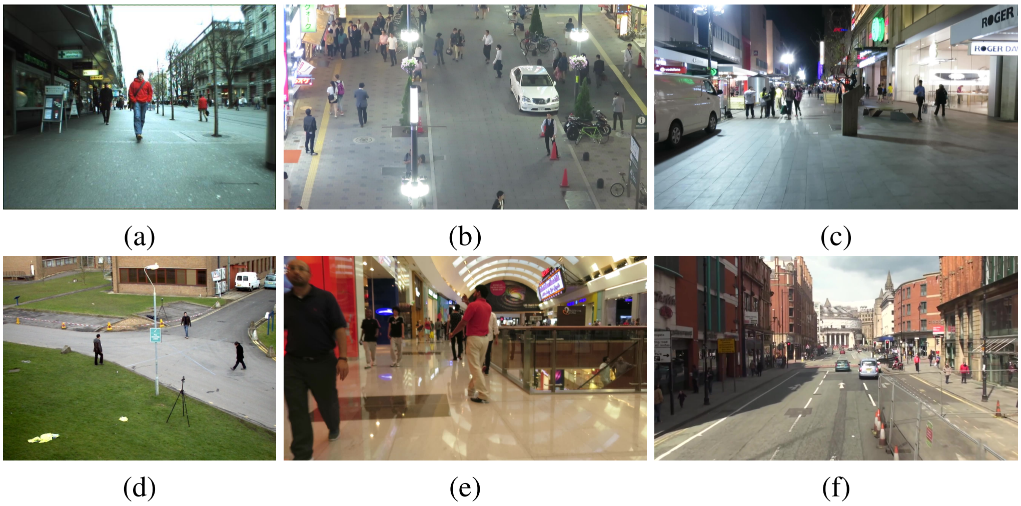
pedestrian densities, illuminations and point of views creating a challenging dataset to push up the state of the art toward a more general trackers.

The datasets include videos, annotations in the form of bounding boxes of objects (persons) and set of detections. Also, a leaderboard exists where all trackers are ranked for comparison. All this allows the fair comparison of different trackers as the same videos, detections and metrics are used. Table 4.7 shows some characteristics of the training and test datasets. MOTB15 contains 22 challenging video sequences (11 training, 11 test) in unconstrained environments, filmed with both static and moving cameras. MOTB16 contains 14 challenging video sequences (7 training, 7 test) in unconstrained environments filmed with both static and moving cameras. Each dataset provides a set of *public detections* and the authors encourage users of the benchmark to use the set of public detections, so that different trackers can be fairly compared to each other. MOTB17 contains the same videos as MOTB16 but MOTB17 includes three new set of detections. Tracking and evaluation are done in image coordinates. All sequences have been annotated with high accuracy, strictly following a well-defined protocol. Fig. 4.25 shows some sample frames of MOTB15 and MOTB16-17 datasets.

The MOTB datasets are suitable for evaluating object detection and object tracking. They are suitable for this thesis since they are composed of heterogeneous sequences with static and moving cameras in different environments.

4.6.2 Multi-detector fusion

In the specific implementation presented here, we use 4 different detectors: Aggregate Channel Features (ACF) [47] trained on INRIA (ACFI) and Caltech (ACFC) datasets, Discriminatively Trained Deformable Part Models (DTDPM) [54] and Scale dependant Pooling (SDP) [211]. The


(a) (b) (c)

(d) (e) (f)

Figure 4.25: Examples of the first frame of two sequences of MOTB15 [94] and four sequences of MOTB16 [125] training dataset. (a) ETH-Bahnhof, (b) MOTB16-04, (c) MOTB16-10, (d) PETS09-S2L1, (e) MOTB16-11 and (f) MOTB16-13.

combined detection has position and bounding box size equal to the weighted average of the position and bounding box size of the detections that contributed to the combination. We compare the performance of the proposed multi-detector fusion against the individual detections, the public detections available in MOTB15 challenge (ACF-MOT) [47], the fusion using an open framework for Combined Pedestrian Detection (CPD) [170] and a baseline where Non-Maximum Suppression is used to fuse the detections. We refer to the results of the proposed method as multi detector fusion (MDF). This experiment is performed in the MOTB15 training dataset.

Fig. 4.26 presents precision-recall curves generated by varying detection confidence thresholds. Results show that MDF outperforms all the individual detectors and fusion frameworks. MDF detects a larger number of correct objects while maintaining a low-rate of false positives. This motivates our choice of having a weighted fusion method that spatially fuses detections based on their confidence score as opposed to non-maximum suppression that selects existing detections without accounting for spatial occurrences.

Using the results shown in Fig. 4.26, we set the threshold in the detection confidence that provides the best F-Score for each method and show their performance in Table 4.8. In this table we can observe that our fusion method largely outperforms non-maximum suppression and CPD in terms of F-Score. MDF achieves competitive results for all videos. This provides us a good means for the generation of reliable trajectories as the tracker will help filter out additional false positives and further increase the recall by recovering weak detections during tracking.

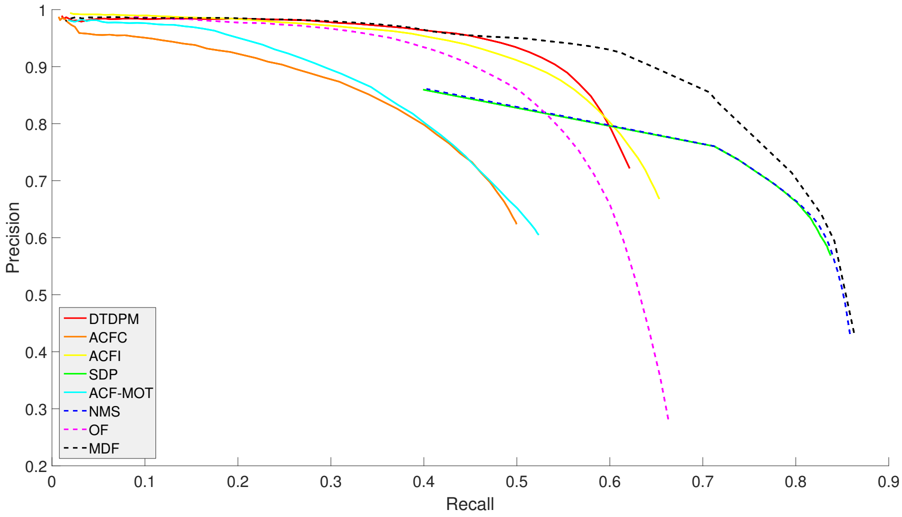


Figure 4.26: Detection results on MOTB15. Precision and Recall curves obtained with each individual detector (DTDPM [54], ACF-C [47], ACF-I [47], SDP [211] and ACF-MOT [47]), and fusion methods non-maximum suppression, CPD [170] and MDF on MOTB15 training sequences.

4.6.3 Object motion prediction

In this section we present the experimental validation of the global-motion aware object motion prediction presented in Sec. 4.3.2. The evaluation in this section is performed as a standalone module, i.e. only prediction is evaluated and no tracking is involved. As performance measure, we employ the mean squared error as described in Sec. 2.6.3.

We compare the proposed method, GM, with six predictors; we quantify robustness to noisy observations and to frame rate reduction; and we discuss the benefits of GM for multiple object tracking. We compare GM against a predictor based on a Long Short-Term Memory (LSTM) [19]; two state-of-the-art predictors based on $\mathbf{x}_{k+1} = \mathbf{x}_k + \bar{\mathbf{x}}_{k|k-T_P+1}$, namely a linear motion predictor (LP) [165] where

$$\bar{\mathbf{x}}_{k|k-T_P+1} = \frac{1}{T_P - 1} \sum_{i=0}^{T_P-2} (\mathbf{x}_{k-i} - \mathbf{x}_{k-i-1}), \tag{4.35}$$

and an exponentially weighted motion predictor (EM) [7] where

$$\bar{\mathbf{x}}_{k|k-T_P+1} = \frac{1}{\sum_{i=0}^{T_P-2} (\varepsilon)^i} \sum_{i=0}^{T_P-2} (\varepsilon)^i (\mathbf{x}_{k-i} - \mathbf{x}_{k-i-1}), \tag{4.36}$$

with $\varepsilon = 0.95$; a linear regressor (LR) where $\mathbf{x}_{k+1} = \mathbf{m}\mathbf{x}_k + \mathbf{b}$, with $\mathbf{m}$ and $\mathbf{b}$ learned online for each object from its past positions; a homography-based (SH) method [101] and, as reference, a

Table 4.8: Detection results obtained with individual detectors DTDPM [54], ACF-C [47], ACF-I [47], SDP [211] and ACF-MOT [47]), and fusion methods CPD [170], non-maximum suppression (NMS) and our MDF on the MOTB15 training dataset. Colour-coded cells indicate the best (darker gray) and the second best (lighter gray) scores for the F-Score. KEY: M, metric; P, Precision; R, Recall; and F, F-Score.

Sequence	**M**	**Individual**					**Fusion**		
		DTDPM	**ACF-C**	**ACF-I**	**SDP**	**ACF-MOT**	**CPD**	**NMS**	**MDF**
ADL-Rundle-6	P	81.63	79.57	71.30	96.44	63.41	65.60	96.44	93.94
	R	52.25	50.55	60.65	67.12	54.18	56.92	67.12	64.66
	F	63.71	61.82	65.54	79.15	58.43	60.95	79.15	76.60
ADL-Rundle-8	P	84.49	63.06	70.64	65.63	49.31	70.39	65.64	76.69
	R	57.91	42.65	54.92	80.52	43.99	64.20	80.55	71.93
	F	68.72	50.88	61.79	72.32	46.50	67.15	72.34	74.23
ETH-Bahnhof	P	83.46	82.99	84.61	93.48	72.80	88.16	93.48	90.78
	R	66.14	52.16	73.91	69.50	57.08	50.10	69.50	78.79
	F	73.80	64.06	78.90	79.73	63.99	63.90	79.73	84.36
ETH-Pedcross2	P	83.86	91.47	95.93	97.36	73.14	91.35	97.36	97.90
	R	42.22	11.59	24.45	64.46	12.21	33.74	64.46	46.83
	F	56.16	20.57	38.96	77.56	20.92	49.28	77.56	63.36
ETH-Sunnyday	P	80.43	80.94	86.31	93.41	84.50	89.94	93.41	91.01
	R	75.91	19.88	71.96	76.85	40.72	66.33	76.85	87.38
	F	78.11	31.93	78.49	84.33	54.95	76.35	84.33	89.16
KITTI-13	P	65.13	35.43	49.85	69.81	50.29	69.30	69.81	60.12
	R	60.71	64.37	71.91	66.95	46.61	50.05	66.95	78.04
	F	62.84	45.70	58.88	68.35	48.38	58.13	68.35	67.92
KITTI-17	P	92.52	58.46	81.86	94.90	82.12	90.24	94.90	84.75
	R	56.91	68.03	69.82	73.79	54.60	56.78	73.79	76.73
	F	70.47	62.88	75.36	83.02	65.59	69.70	83.02	80.54
PETS09-S2L1	P	92.24	86.65	93.17	94.98	79.59	86.00	95.05	97.07
	R	88.67	84.62	92.99	86.24	91.14	71.59	86.32	91.91
	F	90.42	85.63	93.08	90.40	84.97	78.14	90.48	94.42
TUD-Campus	P	89.81	77.64	88.48	99.26	75.51	94.07	99.26	99.24
	R	54.04	69.64	66.30	74.65	61.84	66.30	74.65	72.70
	F	67.48	73.42	75.80	85.21	67.99	77.78	85.21	83.92
TUD-Stadtmitte	P	96.33	66.12	96.08	98.69	77.90	97.05	98.69	99.67
	R	74.83	59.26	78.46	78.11	72.58	73.88	78.11	79.41
	F	84.23	62.50	86.38	87.20	75.15	83.89	87.20	88.40
Venice-2	P	86.53	56.01	80.30	42.92	57.04	75.18	42.92	71.67
	R	40.13	51.65	51.02	77.66	51.94	51.63	77.66	65.71
	F	54.84	53.74	62.39	55.29	54.37	61.22	55.29	68.56
Overall	P	85.02	69.92	80.71	73.54	65.34	79.06	73.55	85.62
	R	57.89	47.00	59.76	73.94	49.94	54.72	73.96	70.60
	F	68.88	56.21	68.67	73.74	56.61	64.68	73.75	77.39

static-object prior-knowledge method (SP) with $\mathbf{x}_{k+1} = \mathbf{x}_k$. Moreover, we compare with GMG, a variation of GM that assumes that all objects lie on that common ground plane, similarly to [101], by masking pixels that are in the estimated ground plane that is defined as the convex hull between the bottom corners of the detections and the bottom corners of the frame.

To compare the methods on the accuracy of their prediction, we use the past T_P ground-truth positions of an object to predict its future T_F positions. If λ is the object index, the ground truth annotations, $\hat{\mathbb{X}} = \left\{\tilde{\mathbf{x}}_{k,\lambda} | \forall \lambda, k\right\}$, of the training dataset includes both static and moving objects. We quantify the prediction error as described in Eq. 2.6.3.

For the good-features-to-track detector [166] and for the sparse tracker [32] we use the default parameters of the OpenCV implementation (version 3.4.1): 50 as maximum number of corners, 0.01 as quality level and 10 pixels as minimum distance for the detector; and 21×21 as window size, three maximum levels of the pyramid and 0.001 as minimum eigenvalue threshold for the tracker. We calculate the homography from the set of correspondent keypoints with the OpenCV implementation and default parameters. The margin in the masking is $c = 0.05$. The minimum number of keypoints per cell is $N_m = 20$. The mask is defined by people detected with SDP [211].

We use three publicly available datasets: MOTB15 [94], MOTB16 [125] and MOTB17 [5]. These datasets are composed of sequences recorded from moving and static cameras. For the first three experiments, we build a *training*, *validation* and *testing* dataset from the MOTB training sequences aiming to balance the number of annotations between static and moving cameras and do not use the same video in different subsets. The *training dataset* accounts for 44% of the dataset and it is composed of ADL-Rundle-8, ETH-Bahnhof, ETH-Sunnyday, KITTI-13, KITTI-17, PETS09-S2L1, TUD-Campus, TUD-Stadtmitte, MOTB16-04, MOTB16-10, MOTB17-04 and MOTB17-10. The *validation dataset* accounts for 16% of the dataset and it is composed of MOTB16-05, MOTB16-09, MOTB17-05 and MOTB17-09. The *testing dataset* accounts for 40% of the dataset and it is composed of Venice-2, MOTB16-02, MOTB16-11, MOTB16-13, MOTB17-02, MOTB17-11, MOTB17-13.

Table 4.9 compares the prediction accuracy of GM against that of the other algorithms. LP, EM, LR and LSTM accumulate prediction errors when the prediction time is long ($T_F = \{10, 20, 30\}$) for any T_P, except for $T_P = 2$ where they obtain competitive results. These methods respectively obtain an average prediction error of 55.4, 52.0, 67.2 and 65.1 pixels when predicting over 30 frames and having observed 30 past frames. GM obtains consistently the best result

Table 4.9: Prediction error on moving-camera sequences (average and its standard deviation on the testing dataset). KEY: T_P, number of past observed positions; T_F, number of future positions to predict; SP, static object prior knowledge; LP, linear prediction; EM, exponentially-weighted prediction; LR, linear regressor; SH, simple homography-based predictor; GMG, proposed global motion with ground masking; GM, proposed global motion prediction; and *, at least an order of magnitude larger (and therefore not reported). The lower the number the better the performance. Best and second best performing methods are shown with dark and light grey, respectively. Values indicate the mean squared error in pixels.

T_P	T_F	**SP**	**LP**	**EM**	**LR**	**LSTM**	**SH**	**GMG**	**GM**
2	1	7.3 (8.2)	2.1 (4.2)	2.1 (4.2)	2.1 (4.2)	6.1 (7.0)	2.8 *	2.2 (4.2)	2.2 (4.2)
	10	35.0 (47.4)	13.7 (19.6)	13.7 (19.6)	13.7 (19.6)	27.3 (36.9)	69.9 *	16.0 (26.4)	13.0 (16.8)
	20	60.3 (81.2)	31.2 (40.5)	31.2 (40.5)	31.2 (40.5)	47.4 (62.9)	194.6 *	31.0 (42.6)	25.1 (31.9)
	30	80.5 (106.1)	50.7 (62.2)	50.7 (62.2)	50.7 (62.2)	64.8 (82.7)	292.7 *	46.1 (61.3)	37.5 (46.0)
10	1	7.2 (8.2)	2.8 (3.4)	2.7 (3.4)	5.2 (5.8)	5.3 (6.5)	10.9 *	3.5 (5.2)	2.9 (3.7)
	10	35.0 (46.8)	15.4 (18.7)	14.9 (18.3)	17.6 (20.2)	25.0 (35.5)	55.4 *	11.5 (18.8)	9.5 (14.2)
	20	59.9 (79.4)	32.0 (38.0)	31.3 (37.5)	34.1 (39.2)	44.0 (60.5)	140.3 *	19.2 (30.4)	16.0 (24.2)
	30	79.4 (101.8)	49.7 (57.6)	49.0 (57.0)	51.9 (58.8)	59.9 (78.2)	199.8 *	26.7 (39.8)	22.3 (33.0)
20	1	7.2 (8.2)	3.6 (3.8)	3.3 (3.7)	12.1 (11.5)	5.6 (6.5)	11.1 *	3.6 (5.6)	3.0 (3.9)
	10	34.9 (46.2)	18.4 (21.7)	17.1 (20.4)	26.1 (26.8)	27.1 (35.9)	63.4 *	11.1 (18.6)	9.4 (15.1)
	20	59.3 (76.8)	35.7 (41.5)	33.8 (39.7)	42.8 (45.5)	49.6 (57.2)	145.4 *	17.8 (28.8)	15.2 (24.8)
	30	78.8 (99.3)	52.9 (60.7)	50.8 (58.7)	59.8 (64.2)	68.6 (76.3)	200.5 *	24.5 (37.5)	20.6 (32.3)
30	1	7.2 (8.2)	4.0 (4.3)	3.5 (3.9)	19.7 (18.5)	5.9 (6.4)	11.7 *	3.6 (5.3)	3.1 (4.2)
	10	34.6 (45.3)	20.3 (23.7)	18.1 (21.5)	34.2 (33.4)	28.2 (34.8)	53.6 *	11.3 (18.6)	9.7 (16.2)
	20	59.2 (76.0)	38.0 (44.3)	35.1 (41.1)	50.8 (51.7)	49.4 (59.1)	193.3 *	18.0 (29.1)	15.5 (25.9)
	30	78.8 (99.1)	55.4 (65.0)	52.0 (60.8)	67.2 (71.1)	65.1 (81.4)	247.5 *	24.6 (37.9)	20.8 (33.3)

followed by GMG except for ($T_P = \{2, 10\}, T_F = 1$) where they are slightly outperformed by LP and EM. In general, the larger T_P (the more past frames are observed), the lower the prediction error, with a reduction of the improvement with $T_P = \{20, 30\}$. SH has large prediction errors due to the lack of a constraint for maintaining a spatial distribution of keypoints and of a masking procedure to eliminate outlier local motions.

Fig. 4.27 shows the object prediction accuracy in the presence of Gaussian noise of varying standard deviation in the past T_P observed object positions. While SP, LR and LSTM are more robust to noise in relative terms, GMG and GM outperform the rest in absolute values when the standard deviation of the noise is lower than 20.

Fig. 4.28 shows the prediction errors when the frame rate is reduced. To constrain the temporal observation when the video frame rate decreases, we select T_P and T_F as $T_P = \lceil \frac{T'_P \kappa}{\gamma} \rceil$ and $T_F = \lceil \frac{T'_F \kappa}{\gamma} \rceil$ where $\lceil \cdot \rceil$ is the ceiling function that rounds a number up to the nearest larger integer, T'_P and T'_F are the past/future number of seconds to observe/predict and κ is the original frame rate of the video. When $\gamma = 1$ (50% frame rate reduction), GM has the lowest absolute prediction error with an error reduction of 41% with respect to the next best-performing methods (LP and EM), and 58% with respect to the subsequent next best-performing methods (LR and SH). When $\gamma = 4$ the error reduction is of 10% with respect to the next best-performing methods (LP

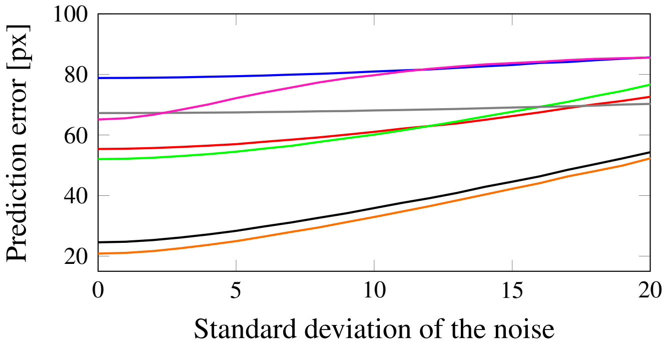


Figure 4.27: Average prediction error in all moving-camera videos of the testing dataset for the next $T_F = 30$ frames when the observations over the past $T_P = 30$ frames are contaminated by Gaussian noise of varying standard deviation. For better visualisation, not showing SH as its error is at least an order of magnitude larger. Values indicate the mean squared error. KEY – px: pixels; SP ▬, LP ▬, EM ▬, LR ▬, LSTM ▬, GMG ▬ and GM ▬.

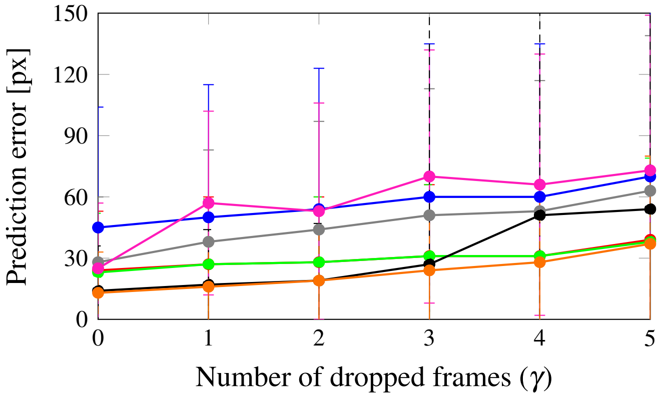


Figure 4.28: Prediction errors (average and its standard deviation) in all moving-camera videos of the testing dataset for the next 0.5 s when observing their positions in the past 0.5 s. For better visualisation, not showing SH as its error is at least an order of magnitude larger. Values indicate the mean squared error. KEY – px: pixels, SP ▬, LP ▬, EM ▬, LR ▬, LSTM ▬, GMG ▬ and GM ▬.

and EM), and 47% with respect to the subsequent next best-performing methods (LR and SH). EM and LP perform similarly (3% and 5% larger prediction error) to GM when $\gamma = 5$ as GM accumulate global motion estimations errors over time. Using only 25% of the original frame rate ($\gamma = 3$), GM obtains comparable prediction accuracy than LP, EM, LR and LSTM at the original frame rate ($\gamma = 0$). These results indicate that the proposed method allows one to reduce the camera acquisition rate while still obtaining a lower prediction error compared to other algorithms.

Next, we compare the processing speed of the methods under comparison and the proposed method with no code optimisation in the testing dataset. All experiments are executed with an Intel i7 microprocessor and 16GB of RAM. The proposed method achieves average processing speed faster than 28 frames per second. Methods that do not use image processing (SP, LP, EM, LR and LSTM) compute the predictions in less than 1 millisecond per frame. SH works at an average of 44 frames per second.

4.6.4 Single-object tracking

We evaluate the tracking performance by comparing the trajectory estimated by the sound source tracker, which is updated every half-block interval ($W/2$), with the ground-truth trajectory generated from the video of the on-board camera. We measure the tracking error, bounded at 180°, at the video frame rate (30 Hz), and calculate the mean and standard deviation across the whole trajectory.

We compare particle filtering (PF) with, as baseline methods, median filtering (MF) and no filtering (NF) on the localisation at individual blocks. MF updates the localisation at the b-th block as the median value, $\mathbb{M}(\cdot)$, of the localisation results across a sequence of W_p blocks:

$$\hat{\theta}_{\mathrm{MF},b} = \mathbb{M}\left(\tilde{\theta}_{b-W_p+1}, \cdots, \tilde{\theta}_b\right), \tag{4.37}$$

where W_p is predefined constant. NF uses the localisation at the b-th block without any processing:

$$\hat{\theta}_{\mathrm{NF},b} = \tilde{\theta}_b. \tag{4.38}$$

We use four block sizes, $W \in \{0.5, 1, 2, 3\}$ s, and in each block we use a STFT of size 1024 and 50% overlap. We set the search area as $[-180°, 180°]$ with an interval of 1°, i.e. $D = 361$. We set the noiseless sector as $[-45°, 45°]$. For the particle filter we set $N = 1000$, and we use a grid-search approach in a training dataset to select a different set of parameters for each block size: for $W = 0.5$ s, $\sigma_p = 3.5°$, $\sigma_u = 10°$, and $W_p = 8$; for $W = 1$ s, $\sigma_p = 5.5°$, $\sigma_u = 4.5°$, and $W_p = 4$; for $W = 2$ s, $\sigma_p = 7.5°$, $\sigma_u = 3.5°$, and $W_p = 2$; for $W = 3$ s, $\sigma_p = 9.5°$, $\sigma_u = 3°$ and $W_p = 1$. $\sigma_{\bar{p}} = 0.05$ for all setups. Unless otherwise specified, $W = 2$ s in the comparisons.

For audio localisation, we compare the performance of two out-of-the-box audio processing approaches. We compare the localisation results obtained by the time-frequency (TF) and the steered response power (SRP) approaches [189], with $W = 1$ s, in the composite sequence recorded in $\mathcal{S}1$. In the presence of noise, SRP is only able to detect the ego-noise only (i.e.,

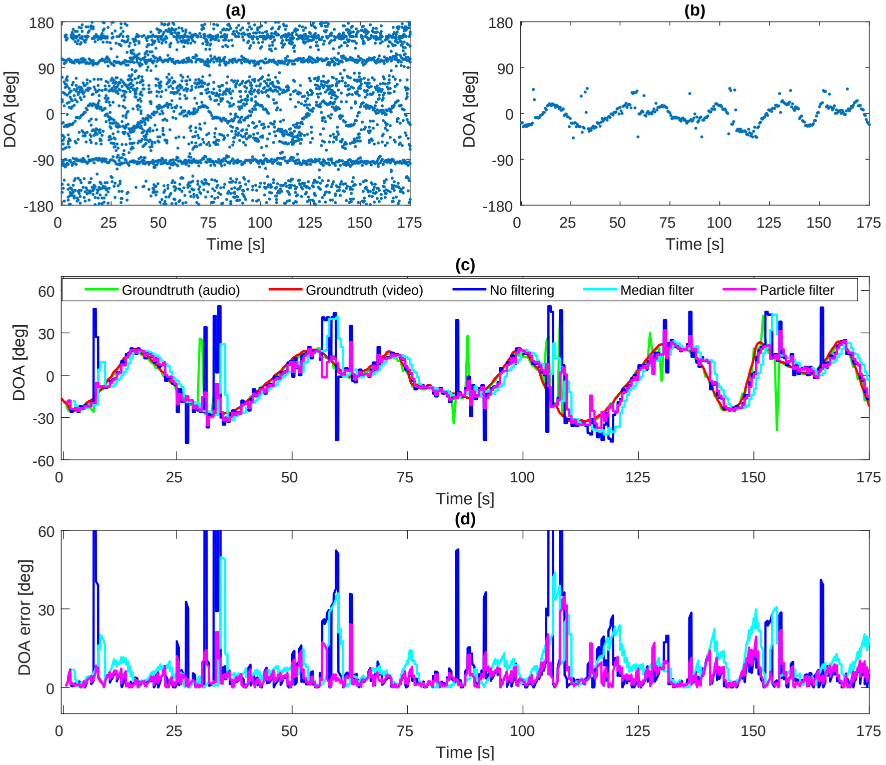


Figure 4.29: Tracking results on the composite sequence in $\mathcal{S}1$, with $W = 1$ s. (a) Original peak detection result. (b) Proposed peak detection result. (c) Trajectories generated by different trackers. (d) Tracking errors obtained by different trackers.

noise produced by the propellers), while TF is able to detect those plus the speech. Therefore, we employ the TF approach for the rest of the experiments.

Fig. 4.29 shows intermediate tracking results based on the confidence map. Fig. 4.29a depicts the original peak detection results, where we retain 10 peaks per processing block in the whole circular area [-180°, 180°]. The confidence map contains considerable noise but the trajectory of the speech can still be observed. Fig. 4.29b depicts the proposed peak detection results, where only one peak is detected in the noiseless sector [-45°, 45°]. The proposed method can remove the spurious peaks in Fig. 4.29a effectively. Fig. 4.29c depicts the ground-truth trajectory of the sound source, the trajectory of the clean speech, the trajectory from detection without filtering (i.e. Fig. 4.29b, the tracking results with MF and PF. All the three trackers (NF, MF, and PF) can capture the trajectory of the moving sound source well. Fig. 4.29d depicts the tracking errors; PF has the smallest variations. The mean (standard deviation) localisation errors by NF, MF, and PF are 5.8°(9.0°), 5.5°(5.5°) and 4.5°(4.3°), respectively. The proposed peak detection method can produce good localisation results, with large errors only in a few blocks. The tracker further improves the localisation accuracy, with PF slightly outperforming MF.

Table 4.10: Localisation errors on the composite sequence in $\mathcal{S}1$ with different block sizes. Each cell shows the mean (standard deviation) error in degrees. The best result is indicated in grey.

W (s)	No filtering	Median filtering	Particle filtering
0.5	10.6 (15.4)	7.1 (8.9)	6.0 (5.4)
1	5.8 (9.0)	5.5 (5.5)	4.3 (4.5)
2	4.7 (4.7)	5.8 (4.7)	4.7 (3.9)
3	6.2 (4.9)	6.2 (4.9)	6.5 (5.0)

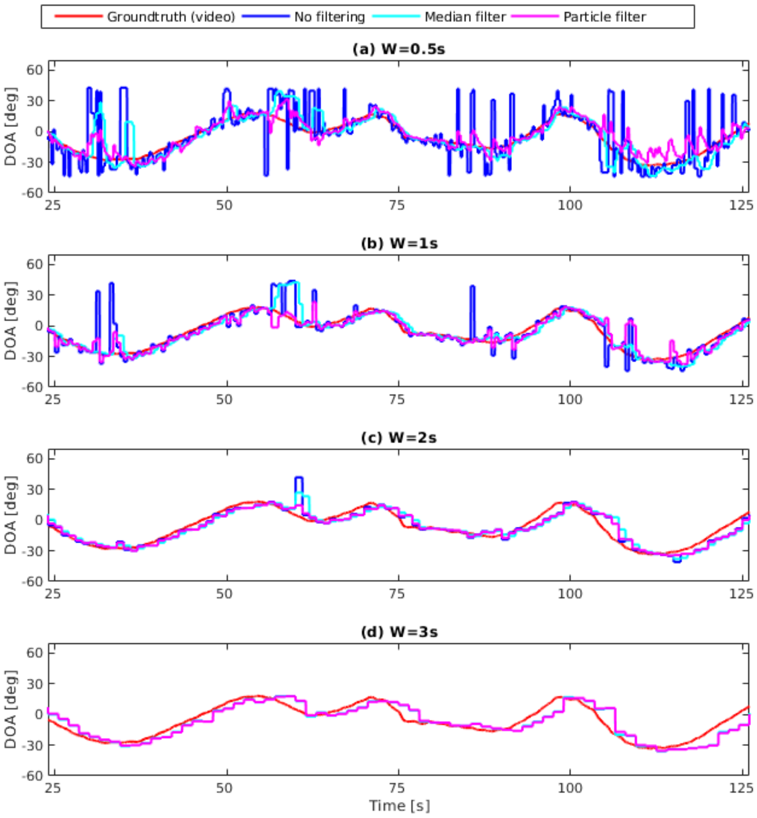
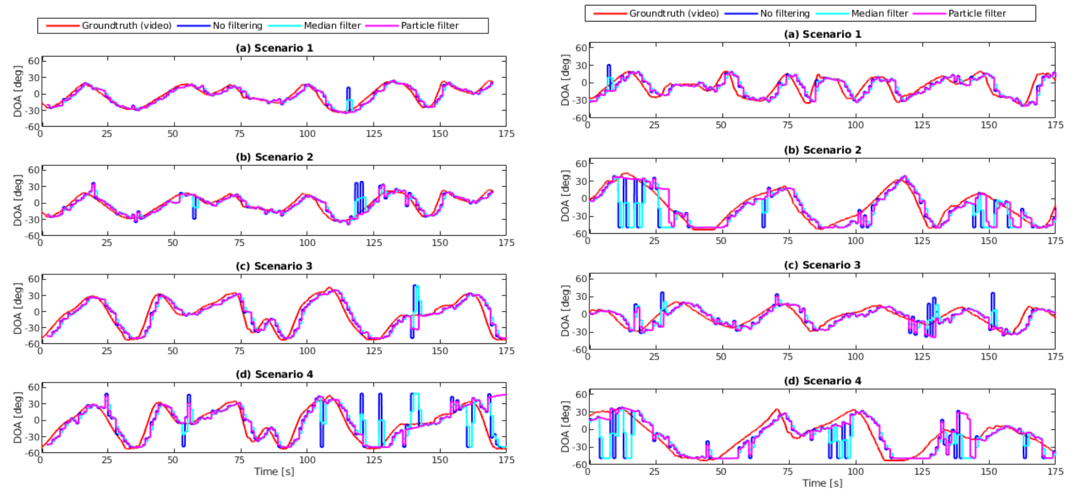


Figure 4.30: Tracking results on the composite sequence in $\mathcal{S}1$ with different block sizes $W \in \{0.5, 1, 2, 3\}$ s.

Table 4.10 shows the localisation error for each tracker with different block sizes. For all trackers, the accuracy improves with the block size until $W = 1$ s, slightly changes with $W = 2$ s, and then drops with $W = 3$ s. Fig. 4.30 compares tracking results for different block sizes. The larger the block size, the less noisy the localisation results and the smoother the trajectory. However, the larger the block size, the longer the tracking delay, which increases the localisation error (see Fig. 4.30a and Fig. 4.30d for $W = 0.5$ s and 3 s, respectively).

Table 4.11 shows the localisation errors on the four scenarios in the composite and the natural datasets. Fig. 4.31a and Fig. 4.31b depict the tracking results for these two datasets. As expected, the trackers perform better in $\mathcal{S}1$ and $\mathcal{S}2$ (loudspeaker moves inside the noiseless sector) than in



(a) (b)

Figure 4.31: Tracking results for the four scenarios in the (a) composite and (b) natural dataset.

Table 4.11: Localisation errors in the four scenarios of the composite ($\mathcal{C}$) and natural (N) dataset. Each cell shows the mean (standard deviation) error in degrees. The best result is indicated in grey.

Dataset	Scenario	No filtering	Median filtering	Particle filtering
$\mathcal{C}$	$\mathcal{S}1$	3.5 (4.7)	4.6 (4.5)	3.8 (3.9)
	$\mathcal{S}2$	4.3 (7.8)	4.9 (5.7)	4.4 (4.5)
	$\mathcal{S}3$	7.4 (9.0)	9.3 (9.0)	8.2 (7.8)
	$\mathcal{S}4$	8.0 (17.6)	10.9 (13.8)	8.4 (21.2)
N	$\mathcal{S}1$	8.7 (7.5)	9.9 (7.8)	9.1 (7.4)
	$\mathcal{S}2$	8.8 (8.4)	8.7 (6.5)	8.3 (6.5)
	$\mathcal{S}3$	14.7 (18.9)	15.4 (16.0)	10.3 (8.8)
	$\mathcal{S}4$	16.4 (19.5)	16.4 (16.7)	11.5 (9.3)

$\mathcal{S}3$ and $\mathcal{S}4$ (loudspeaker moves freely in front of the drone), because the source localisation performance degrades when the speaker moves outside the noiseless sector. The hovering power of the drone does not greatly affect the tracking performance, as shown by the similar performance in $\mathcal{S}1$ and $\mathcal{S}2$, and in $\mathcal{S}3$ and $\mathcal{S}4$. The composite dataset and natural dataset do not show large differences in localisation error. Video tracking results available online[3].

4.6.5 Multi-object tracking from a static camera

We validate the proposed Early Association Probability Hypothesis Density Particle Filter (EA-PHD-PF) and compare it against state-of-the-art online tracking methods on the MOTB15 and MOTB16 benchmark datasets [94, 125].

[3] Video tracking results available at `https://www.youtube.com/watch?v=zJapzQllr_M`

We use the *public detections* provided by the MOT benchmark and our *private detections* produced by combining detections from state-of-the-art person detectors (description in Sec. 4.2.1). We refer to the tracker using the public detections from MOT benchmark as EA-PHD-PF(Pub) and to the tracker using the private detections as EA-PHD-PF(Priv).

We allow associations to be made between detections and predicted states only if their intersection over union is larger than $\tau_a = 1/3$. The parameter that controls when a trajectory will not seek for more detections is $V = \lceil f \rceil / 1s$, where f is the frame-rate of the video sequence. The parameter that controls the maximum possible number of frames to consider in the prediction model is $M_{max} = \lceil f/2 \rceil / 1s$.

We select the parameters of our method on the MOTB15 and MOTB16 training datasets via parameter search and then use these parameters in MOTB15 and MOTB16 testing sequences, respectively. For the set of public detections $\tau_s = 0.39$ in MOTB15 and $\tau_s = 0.20$ in MOTB16. For the set of private detections $\tau_s = 0.35$ in both datasets[4]. The number of particles per object, ρ, is set to 500. We set the standard deviation values used for the prediction, update and newborn particle generation as a function of the bounding box size. Using the annotations of the MOTB15 and MOTB16 training dataset, the standard deviation for the λ-th tracking state at time k is

$$\sigma_{k,\lambda} = \mathbf{x}_{k,\lambda} std \left(\left\{ \frac{d^2 \tilde{\mathbf{x}}_{k,\lambda}}{dk^2} \right\}_{\forall k,\lambda} \right), \tag{4.39}$$

where $std(\cdot)$ is the standard deviation, $\frac{d^2(\cdot)}{dk^2}$ is the second derivative that quantifies the noise in the variation of the annotation $\tilde{\mathbf{x}}_{k,\lambda}$. We use the bounding box size at time k, w and h, in order to adapt the noise to the scale of the bounding box.

For tracking evaluation we use the performance metrics discussed in Sec. 2.6. Table 4.12 compares the tracking results of our proposed method using both public and private detections with the TOP-6 online trackers submitted to the MOTB15 and MOTB16 benchmark[5]. The upper part of the table (MOTB15) shows that EA-PHD-PF(Priv) outperforms the following submissions FOMT, LKDAT_CNN, MOT_DL, AMPL, GM_2015 and MDP_SubCNN, in terms of MOTA and MOTP. The number of FN and the ML percentage are overall lower than the other trackers. This is due to the ability of EA-PHD-PF(Priv) to robustly perform state estimation, exploiting weak

[4] Larger values of τ_s reduce the number of false positives and lead to a more conservative initialisation of the trajectories.

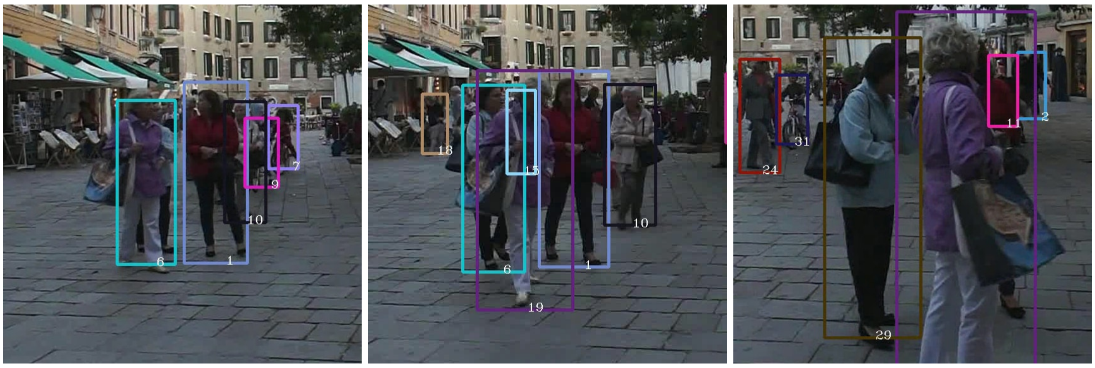
[5] Last accessed on 27th June 2020.

Table 4.12: Online tracking results on the MOTB15 (TOP-6 trackers) and on the MOTB16 (TOP-6 trackers) test datasets. Non-referenced results correspond to anonymous submissions. Colour-coded cells indicate the best (darker grey) and the second best (lighter grey) scores for the F-Score. Trackers can employ detections (Det) public (Pub) or private (Priv). Public detections are released by the challenge and are common to all trackers. Private detections are generated by each participant and are different to each tracker. KEY: P, Precision; R, Recall; and F, F-Score.

Dataset	Tracker	Det	MOTA	MOTP	FAF	MT (%)	ML (%)	FP	FN	IDS	Frag	Hz
MOTB15	FOMT	Priv	53.0	74.8	1.2	32.7	14.6	6,974	20,776	1,143	2,043	16.0
	LKDAT_CNN[6]	Priv	49.3	74.5	1.0	20.8	28.4	6,009	24,550	563	1,155	1.2
	MOT_DL	Priv	49.1	73.9	1.5	35.4	25.0	8,488	22,281	511	1,390	3.9
	AMPL	Priv	48.9	75.3	1.8	28.3	24.3	10,676	20,292	401	1,072	5.1
	GM_2015	Priv	48.2	74.0	1.6	33.6	20.0	9,385	21,318	1,110	1,841	19.3
	MDP_SubCNN [204]	Priv	47.5	74.2	1.5	30.0	18.6	8,631	22,969	628	1,370	2.1
	EA-PHD-PF [C7]	Pub	22.3	70.8	1.4	5.4	52.7	7,924	38,982	833	1,485	12.2
	EA-PHD-PF [C7]	Priv	53.0	75.3	1.3	35.9	19.6	7,538	20,590	776	1,269	11.5
MOTB16	POI [217]	Priv	66.1	79.5	0.9	34.0	20.8	5,061	55,914	805	3,093	9.9
	AMPL	Priv	63.7	78.4	1.7	36.1	21.6	10,171	55,322	714	1,538	4.5
	MHTWG	Priv	61.2	79.2	1.3	30.2	22.4	7,897	61,289	1,570	2,683	30.0
	KFILDAwSDP	Priv	57.3	77.5	2.6	24.6	25.3	15,682	60,252	1,873	2,664	2.2
	MDPNN16	Pub	43.8	75.5	0.6	12.4	40.7	3,501	98,193	723	2,036	1.0
	olCF [87]	Pub	43.2	74.3	1.1	11.3	48.5	6.651	96.515	381	1.404	0.4
	EA-PHD-PF [C7]	Pub	38.8	75.1	1.4	7.9	49.1	8,114	102,452	965	1,657	11.8
	EA-PHD-PF [C7]	Priv	52.5	78.8	0.7	19.0	34.9	4,407	81,223	910	1,321	12.2

detections without relying on the prediction only when (strong) detections are missing. The relatively higher number of IDS compared to the other methods is due to the fact that we rely only on the position and size of the bounding box inferred from the detections and *we are not using any appearance models* to discriminate nearby objects in this evaluation. Spawning objects are not considered. Identity switches are more likely in crowded scenes where occlusions often happen, as shown in Fig. 4.32. The bottom part of Table 4.12 (MOTB16) shows that EA-PHD-PF(Priv) currently is at half-rank in terms of MOTA achieving competitive results in MOTP, FP and fragmentations metrics. This behaviour is mostly due to the unbalance on the detectors used in each algorithm. The number of IDS is higher than AMPL and olCF, because the features they use are better able to discriminate objects. The results using public detections rank our tracker EA-PHD-PF(Pub) at the end of the TOP-6 online trackers along with olCF since the detection performance of the public detections are not comparable with the private detections provoking a big amount of FN and then a relatively low MOTA.

Fig. 4.33 compares sample tracking results using public (a-d) and private (e-h) detections. We can observe along the first row how the object firstly initialised as 115 is then reinitialised, lost and reinitialised again due to the high number of FN in the public dataset. However, the (same)



(a) (b) (c)

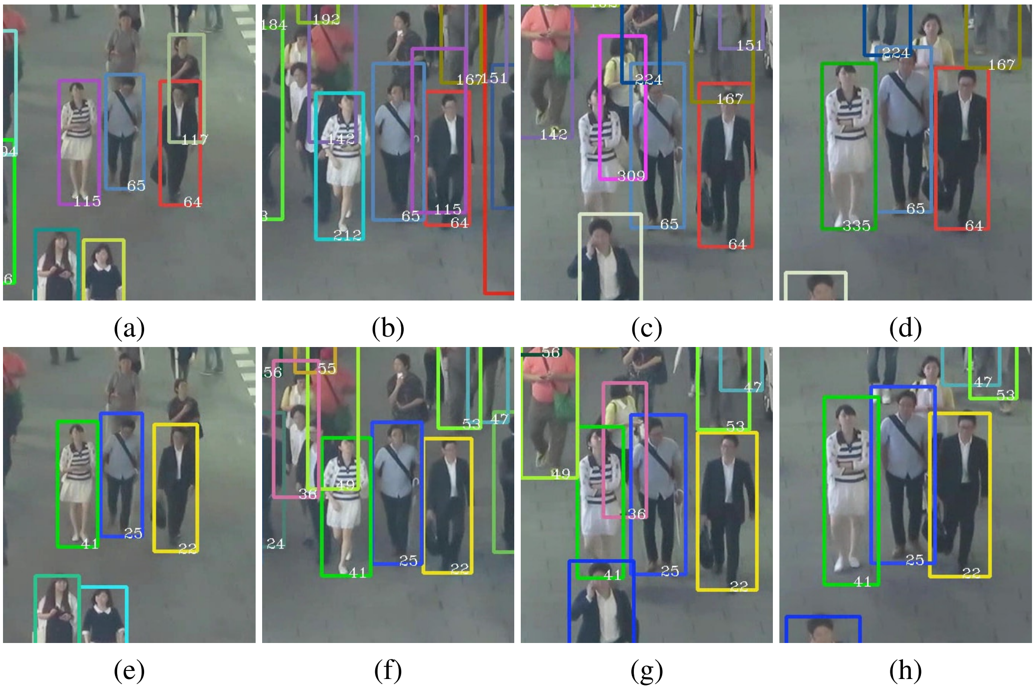
Figure 4.32: Examples of tracking under multiple occlusions at frames 22, 97 and 261 (crops) in Venice-1/MOTB16-01 using EA-PHD-PF(Priv). (a) Object 6 is correctly tracked while it occludes another object. (b) The occluded object becomes visible and trajectory 6 drifts towards it; object 6 is reinitialised as object 19 at frame 22. (c) Intermittent detections cause object 6 to be reinitialised as object 29 at frame 97.

object firstly initialised as 41 in the second row is correctly tracked along the whole sequence. We can observe the presence of false-positive trajectories (i.e. green, purple and red bounding boxes in Fig. 4.33b). These false-positive trajectories are difficult to remove because they are caused by persistent false-positive detections appearing for a few consecutive frames and the confidence scores of those detections are as high as those of true positive detections. With EA-PHD-PF(Priv) these detections are filtered out without adding any false-negative trajectories.

Visual tracking results can be found online athttps://motchallenge.net.

In order to evaluate the effect of each contribution, we compare the tracking performance of the proposed method with respect to versions where each contribution is independently omitted. In particular, four different evaluations have been tried. The results are shown in Table 4.13. The first row shows the tracking results of our method (EA-PHD-PF) in the MOTB15 training dataset [94] where our three main contributions are present. Strong and weak detections are used as described in Section 4.2.1, Early Association is used as explained in Sec. 4.1 and perspective dependency is applied as explained in Sec. 4.2.3. Weak detections are discarded, second row, and all used as strong detections, third row. In the forth row, clustering followed by association is performed instead of using the Early Association approach, similarly to [115]. Lastly in the last row, fixed noises are applied to the prediction, update and sampling processes instead of using the proposed perspective dependency approach.

The proposed method, first row in Table 4.13, achieves the best MOTA and IDS results, which are the most relevant metrics in MOT, and at the same time that gets very competitive results in



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.33: Examples of tracking at frames 240, 461, 645 and 717 (crops) in MOTB16-03 using public (first row) and private (second row) detections. (a-d) The object identified as 115 is reinitialised multiple times due to occlusions and lack of detections. (e-h) The (same) object, identified as 41, is correctly tracked. Source images from [125].

Table 4.13: Ablation studies comparing the tracking performance with and without strong and weak detections, Early Association and prediction dependency in MOTB15 training dataset. The best result for each metric indicated in grey. Key: EA, Early Association; and PD, prediction dependent. * indicates that all detections are considered to be strong.

Detections		EA	PD	MOTA	MOTP	FAF	MT (%)	ML (%)	FP	FN	IDS	Frag	Hz
Strong	Weak												
✓	✓	✓	✓	56.11	78.53	0.92	42.40	27.20	5,042	12,266	205	444	11.5
✓		✓	✓	45.33	78.64	0.58	27.40	36.20	3,192	18,000	625	785	12.1
*		✓	✓	19.81	77.67	4.65	68.60	7.00	25,577	5,914	510	790	8.3
✓	✓		✓	48.60	74.80	0.50	22.40	32.40	2,597	17,083	848	955	6.6
✓	✓	✓		55.59	78.08	1.01	44.40	27.40	5,534	11,954	232	479	11.8

the rest of metrics. As expected, when weak detections are not used (second row), challenging detections are missed and then FN and IDS drastically increases. However, if weak detections are treated as strong detections (i.e. there is not classification of detections and all are used as strong detections), third row, a large number of trajectories are initialised, provoking a small number of FN but with a large number of FP and IDS. Using clustering instead of the proposed EA obtain about 8 less MOTA points and the processing speed is almost double. Lastly, in the last row, removing the perspective dependency achieves similar results to the proposed method but slightly decreases the MOTA and MOTP results since objects at different distances to the

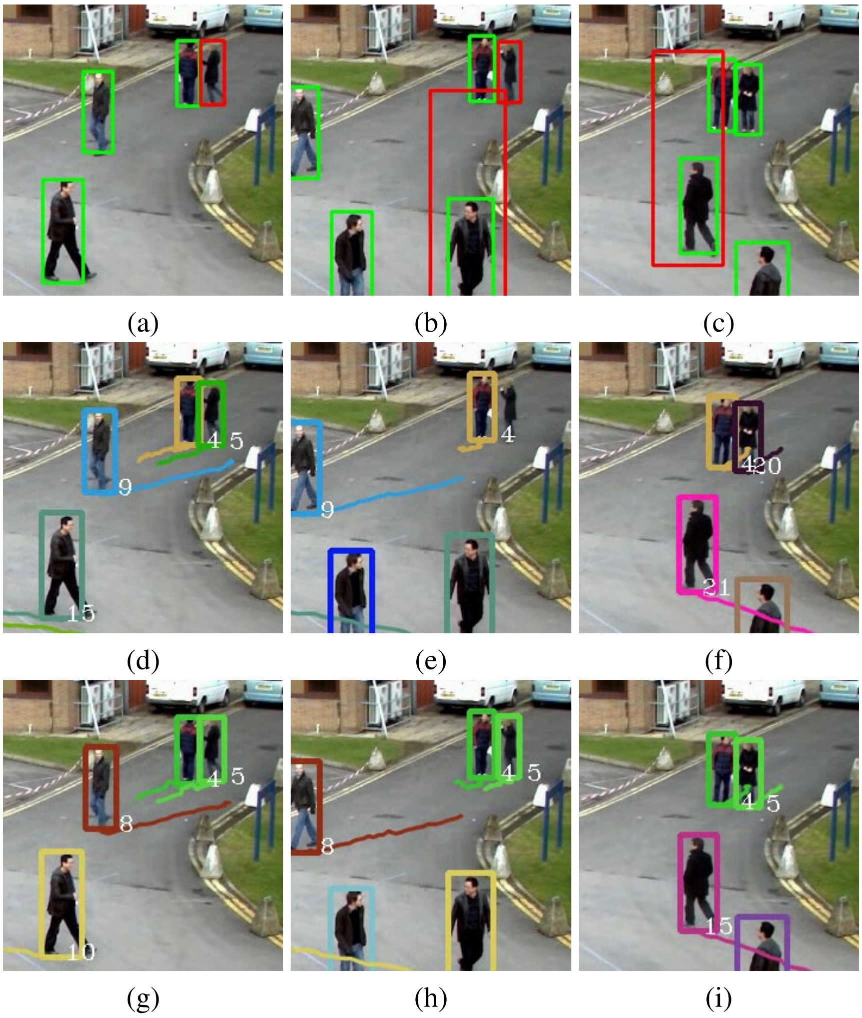


Figure 4.34: Examples of tracking at frames 178, 193 and 240 (crops) in PETS09-S2L1 (not) using weak detections. (a-c) Strong (green) and weak (red) detections. (d-f) Without using weak detections object 5 is lost and a new trajectory is later initialised with identity 20. (g-i) Using weak detections object 5 is correctly tracked. Source images from [125].

camera cannot cope with the same fixed noise.

Fig. 4.34 shows the benefit of weak detections. Without weak detections, miss-detections produce false negative trajectories and identity switches (Fig. 4.34, second row). When weak detections are used, the objects are correctly tracked (Fig. 4.34, third row).

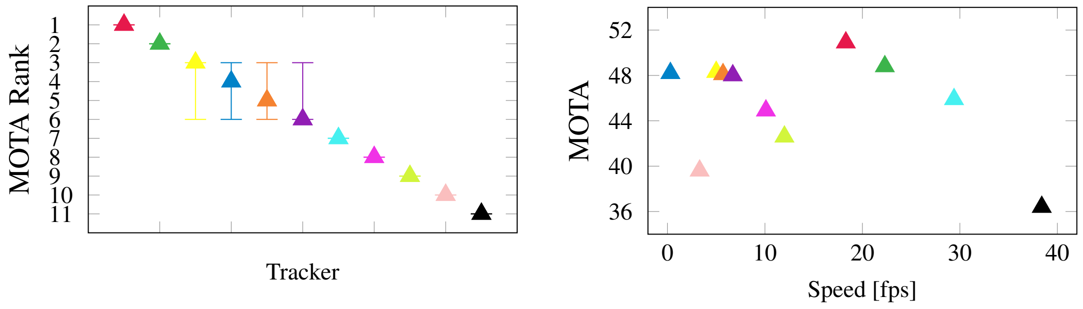
We have presented a variation of the PHD-PF framework, named EA-PF-PF, that effectively uses strong (high-confidence) and weak (low-confidence) detections for performing online multi-object tracking. The states are labelled via an efficient early association strategy that allows the avoidance of the often required clustering stage. Perspective information is used in the EA-PHD-PF framework within the prediction, update and generation of newborn particles enabling the tracker for scale variations of the objects. The proposed object motion model named as perspective-dependent linear motion prediction is a variation of a traditional linear motion prediction object motion model that can be used to accurately predict the location of a given tracked object when certain linear assumptions are meet. In our object application with videos similar to MOTB, the motion of objects can be considered as linear when recorded by static cameras. However, when the objects of interest are observed from moving cameras, the proposed motion model is not appropriate and the tracking prediction may drift. An example of tracking drift due to linear motion prediction in moving cameras can be seen in the first row of Fig. 4.36.

4.6.6 Multi-object tracking from a moving camera

We perform the evaluation in the Multiple Object Tracking Benchmark 2017 (MOTB17) [5] dataset which provides three sets of object detections generated by: DTDPM [54], FRCNN [157] and SDP [211]. We use these detections as input for the tracker. As evaluation score, we use Multiple Object Tracking Accuracy (MOTA) [26] and Multiple Object Tracking Precision (MOTP) [26]. As discussed in Sec. 4.5, the ground-truth annotations in this dataset are generated using semi-automatic techniques, therefore, the evaluation metrics contain uncertainties that are estimated to be 0.225 for MOTA and 5.678 for MOTP [C5]. In other words, one tracker cannot be guaranteed to outperform another on a certain metric when their metrics do not differ more than twice the confidence interval value. We run the tracker five times and report the average and standard deviation results.

The parameters of the global motion estimators are set as in the reference papers. The MOTB17 dataset is divided in training and testing sets. We select the tracking parameters that achieve the most accurate results in the training dataset by brute force and we fix them for all experiments. The standard deviation for the prediction and observation models are set to: $\sigma_p^u = \sigma_p^v = \frac{\Delta_k \cdot w}{18}$, $\sigma_p^{\bar{u}} = \sigma_p^{\bar{v}} = \frac{\Delta_k \cdot w}{36}$, $\sigma_p^w = \sigma_p^h = 5$, $\sigma_o^u = \sigma_o^y = \frac{\Delta_k \cdot w}{12}$ and, $\sigma_o^w = \sigma_o^h = 10$. The time interval between frames is considered constant and, therefore, we set $\Delta_k = 1$. As for the strong/weak detection threshold, τ_s [C7], we use $\tau_s = 0.2$ for DTDPM, $\tau_s = 0.95$ for FRCNN and $\tau_s = 0.8$ for SDP. The number of particles per object is $\rho = 200$.

We compare our tracker against eight state-of-the-art online trackers, which can be clustered in three groups regarding their focus: appearance, dynamics, or others. Trackers that focus on improving the *appearance* modelling are: MOTDT [108], HAM-SADF [216], AM-ADM [96], FPSN [95], and GM-PHD-KCF [92]. These trackers propose different deep-learning-based approaches to learn an appearance representation to be able to discriminate objects to each other



(a)

(b)

Figure 4.35: Comparison of online trackers in MOTB17 test dataset. (a) MOTA rank and its confidence interval [C5] and (b) comparison between MOTA and execution speed. KEY – fps: frames per second, MOTDT [108] ▲, HAM-SADF [216] ▲ , DMAN [223] ▲ , AM-ADM [96] ▲ , PHD-GSDL [57] ▲ , FPSN [95] ▲ , EA-PHD-PF [C7] ▲ , GM-PHD-KCF [92] ▲ , GM-PHD [49] ▲ , proposed with linear prediction (LP) [165] ▲ and proposed with global motion (GM) (Sec. 4.3.1) ▲.

with the aim of improving tracking performance in challenging situations such as occlusions or crowded scenes. Trackers that focus on improving the *dynamic* modelling, e.g. DMAN [223], propose another deep-learning-based model that focuses on hard negative distractors using both spatial and temporal attention mechanisms. The last group is formed by PHD-GSDL [57] and GM-PHD [49], which are based in the PHD filter framework. The former focuses on improving the modelling of the target birth with a novel gating mechanism; and the latter one is a used as PHD-based tracker baseline.

Fig. 4.35a shows the MOTA Ranking in MOTB17 test dataset for all available online trackers (we do not show tracking precision rankings as MOTP confidence interval indicates that this metric is not informative to rank these trackers in this dataset - the proposed tracker also achieves state-of-the-art MOTP rank). Regarding MOTA and the confidence intervals, we can confidently say that the proposed tracker outperforms all trackers under comparison but MOTDT [108], which uses a deeply learned appearance representation of the objects, slightly outperforms the proposed tracker with a MOTA 1.1 higher but 4 fps slower. Fig. 4.35b shows the results when comparing accuracy and execution speed (the higher, the more accurate; the further to the right, the faster). Considering both measures, the best performing trackers are MOTDT and the proposed one. The rest of the trackers are slower and less accurate except for GM-PHD (9.2 lower MOTA but 70% faster) and the proposed method with LP as predictor (2.9 lower MOTA but 30% faster). Visual tracking results with LP as predictor are available online[7].

To examine the contribution of the global motion module, we integrate within the proposed

[7] `https://motchallenge.net/tracker/PHD_LMP`

Table 4.14: The object predictors under comparison. KEY: Ref, reference; TM, considers object motion; CM, considers camera motion; NS, not scene specific; and FT, no assumptions on the object location.

Ref	Prediction model	TM	CM	NS	FT
BM [131]	Brownian			✓	✓
LP [165]	Linear	✓		✓	✓
SH [19]	Planar homography	✓	✓		
GM [C4]	Global motion	✓	✓	✓	✓

tracker four different prediction models, including two global motion estimators: a homography-based method (SH) [19], a global motion-aware camera motion (GM) [C4], a Brownian model (BM) [131], a linear predictor (LP) [165] that does not account for camera motion and we refer to it as EA-PHD-PF (Sec. 4.3). Table 4.14 summarises the prediction models under comparison. SH and GM estimate the global motion, $\mathbf{H}_{k-1|k-2}$, and is provided to the tracker in Eq. 4.27. Prediction methods such as SH and GM outperforms deep-learning-based prediction models in this scenario [C4], therefore we do not compare against deep-learning-based predictors in our experiments.

Fig. 4.36 shows tracking results with LP (first row) and GM (second row) in a moving camera sequence (MOTB17-13). The camera heavily jaws producing large object motions in the image plane. While LP wrongly forecasts the object locations producing multiple false trajectory initialisations and identity switches, GM accurately predicts the motion of the objects and maintains the correct object identities with no false positive initialisation. Visual tracking results are available online[8].

Table 4.15 shows the results when modifying the input set of detections (DTDPM [54], FR-CNN [157] or SDP [211]) and the global motion estimators (BM [131], LP [165], SH [19] and GM [C4]) on the training dataset. The detector is the component that most affects tracking accuracy. GM enables the most accurate and precise tracking obtaining on average an increment of 9.0 and 3.2 MOTA points, and 3.0 and 0.5 MOTP points, when comparing with BM and LP, respectively, in moving cameras. Among the alternatives that consider the global motion, SH performs worse than GM due to a more inaccurate global motion estimation. The image processing techniques used to estimate the global motion impacts the execution speed of the tracker. The fastest tracker is when BM and LP are used (i.e. no global motion is needed to be estimated) achieving up to 58.0 fps while SH and GM obtain 29.7 and 22.5 fps, respectively (using DTDPM

[8] Tracking results in MOTB17 dataset with public detections.

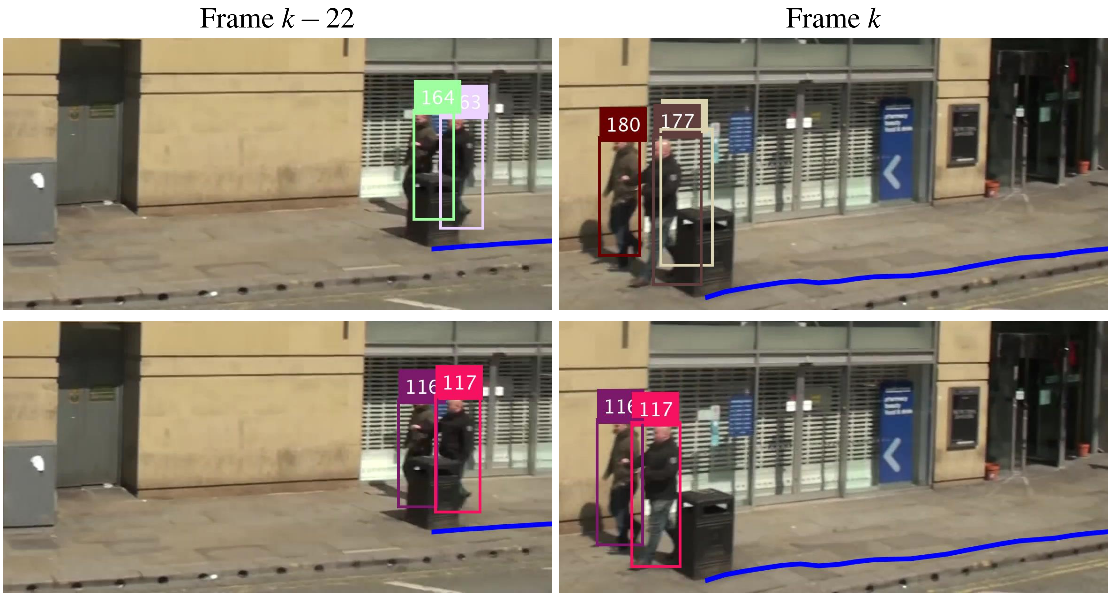


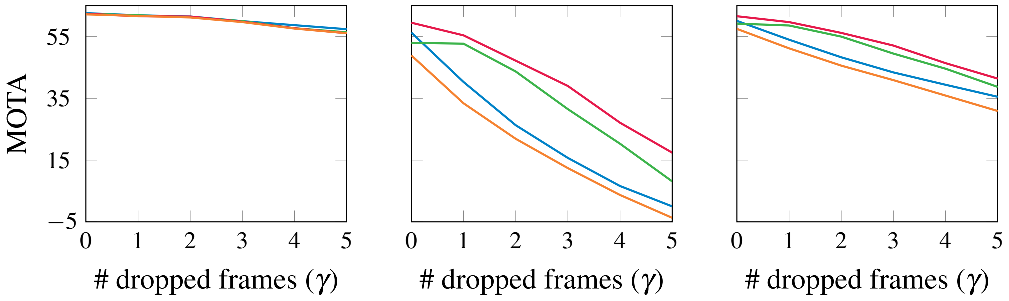
Figure 4.36: Sample tracking results in MOTB17-13 sequence while camera jaws clockwise with two predictors: linear prediction (top row) and the proposed global motion (bottom row). Colours and numbers in bounding boxes indicate object identity. Blue line indicates the past locations of a static object (the rubbish bin) and it is drawn to visualise the camera motion. Note that global motion predictor allows the tracker to maintain the correct object identities (second row). Source images from [125].

Table 4.15: Tracking performance (average and standard deviation) comparison for different detectors and prediction models on MOTB17 training dataset. KEY, PM, prediction model; M, moving-camera sequences only; C, complete dataset; BM, Brownian model [131]; LP, linear prediction [165]; SH, homography-based global motion [101]; and GM, global motion estimation [C4]. The higher the score, the better the performance. Best and second best performing methods per detector and type of camera motion are shown in dark and light grey, respectively.

Detector	**PM**	**MOTA**		**MOTP**	
		M	**C**	**M**	**C**
DTDPM	BM	30.3 (0.1)	34.8 (0.1)	74.2 (0.1)	76.3 (0.1)
	LP	33.1 (0.3)	36.1 (0.2)	75.0 (0.1)	76.5 (0.1)
	SH	31.8 (0.6)	35.5 (0.2)	74.5 (0.1)	76.3 (0.1)
	GM	35.5 (0.1)	37.0 (0.1)	75.3 (0.1)	76.5 (0.1)
FRCNN	BM	42.6 (0.1)	46.8 (0.2)	79.2 (0.1)	84.3 (0.2)
	LP	49.5 (0.3)	49.6 (0.2)	83.5 (0.1)	86.5 (0.1)
	SH	46.5 (0.2)	48.4 (0.1)	82.2 (0.0)	86.0 (0.2)
	GM	52.9 (0.2)	50.9 (0.1)	84.0 (0.1)	86.6 (0.3)
SDP	BM	48.9 (0.3)	59.9 (0.4)	78.7 (0.1)	83.0 (0.1)
	LP	56.5 (0.4)	62.6 (0.1)	81.0 (0.0)	83.7 (0.0)
	SH	52.6 (0.6)	61.7 (0.2)	80.4 (0.0)	83.3 (0.1)
	GM	60.3 (0.3)	64.7 (0.2)	81.6 (0.1)	83.9 (0.1)

as detector).

Fig. 4.37 shows the tracking accuracy when downsampling the video frame rate by keeping



(a) (b) (c)

Figure 4.37: Tracking performance with reduced video frame rate (γ) in (a) static-camera videos, (b) moving-camera videos, and (c) the complete MOTB17 training dataset (both static- and moving-camera); for four prediction models: Brownian model (BM) [131] —, linear prediction (LP) [165] —, homography-based global motion (SH) [101] — and global motion estimation (GM) [C4] —. Standard deviations are less than one point. KEY: #, number of.

one frame and skipping the next γ (e.g. $\gamma = 1$ indicates 50% frame rate reduction) for different prediction models. Fig. 4.37(a) indicates that the tracking performance in static cameras is similar regardless the prediction model. However, when the camera moves Fig. 4.37(b), GM consistently obtains the highest tracking accuracy for any γ. For instance, when $\gamma = 3$ the propose prediction method outperforms BM, LP and SH by 26.6, 23.3 and 5.4 MOTA points, respectively. The proposed prediction formulation, when using SH and GM as global motion estimators, allows the tracker to reduce the video frame rate by 75% ($\gamma = 3$) while maintaining 60% and 66% of its original tracking accuracies, whereas BM and LP are only able to maintain 25% and 28% of its original tracking accuracies in moving-camera sequences. Moreover, GM allows the tracker to drop half of the frames while outperforming BM and SH, and achieving comparable results than LP.

Fig. 4.38 shows visual tracking results with LP (first column) and GM (second column). In the first video (first two rows), the camera translates forward and rotates significantly; the LP cannot cope with this complex motion and most of the object identities are switched. However, the GM is able to maintain the identity for most objects. In the second video (third and fourth rows), the camera only translates forward producing small apparent motions in the image plane; both prediction models are able to accurately track most of the objects. In the third video (last two rows), the camera undergoes a strong clockwise jawing producing large motions in the image plane which results in multiple false trajectory initialisations when LP is used but GM can cope with this challenging scenario.

In summary, experiments show that the proposed tracker outperforms most of the trackers

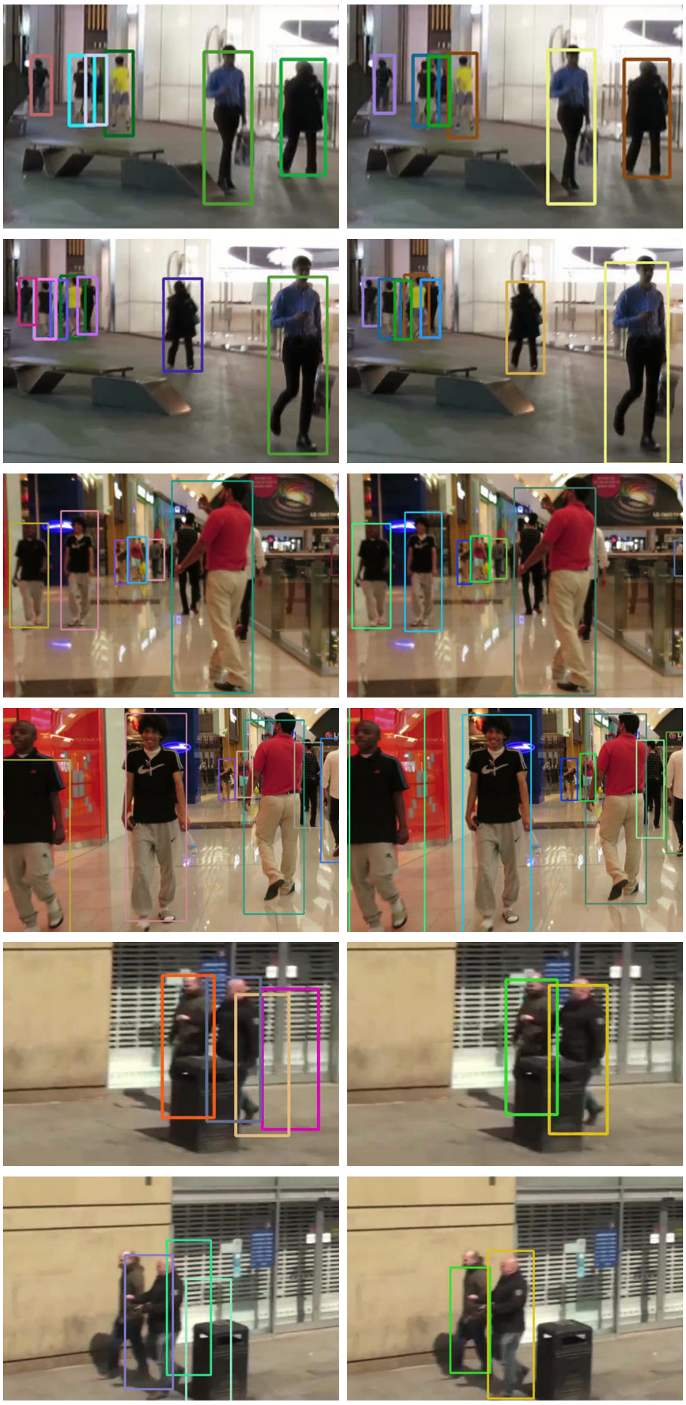


Figure 4.38: Comparative tracking results with LP prediction method (first column) and with GM (second column). Colour code indicates the object identity. GM is able to maintain the object identities in most of the cases in the three scenarios under strong camera motion. Source images from [125].

in terms of accuracy and speed (Fig. 4.35). Specially, we showed that the proposed prediction model excels in the presence of moving cameras (Fig. 4.15). Also, we showed that the proposed tracker can be used to intentionally decrease the frame rate of videos, with the aim of reducing the number of operations, and thus consumption power, and still maintain higher levels of accuracy than existing alternatives (Fig. 4.37).

4.7 Summary

This chapter presented an audio-based algorithm that tracks a sound source from a UAV, a vision-based multi-object tracker for static and moving cameras, and an object motion prediction that is able to forecast the location of moving objects from a moving camera.

We introduced a probabilistic-based tracker that is able to track a moving sound source from a noisy UAV by combining time-frequency filtering, peak detection, and particle filtering. We presented a self-collected audio-visual dataset recorded outdoors with an 8-microphone circular array and a camera mounted on a quadcopter that enable researches for the creation of new audio-visual sensing problems in the presence of strong ego-noise. We experimentally validated the effectiveness of the proposed tracker that obtains tracking errors under 11.5 degrees.

We presented EA-PHD-PF, an online multi-object tracker that exploits strong and weak detections in a Probability Hypothesis Density Particle Filter framework. Strong detections are used for trajectory initialisation and tracking. Weak detections are used for tracking existing objects only to reduce the number of false negatives without increasing the false positives. Moreover, we presented a method to perform early association between trajectories and detections, which eliminates the need for a clustering step for labelling. Also, we exploited perspective information in prediction, update and newborn particle generation. Experimental results showed that our tracker, without using object appearance information, outperforms alternative online trackers in the Multiple Object Tracking 2015 and 2016 benchmark datasets in terms tracking accuracy, false negatives and speed. The tracker works at an average speed of 12 fps.

Lastly, we introduced GM, an object motion predictor that is aware of the global camera motion. The proposed predictor does not require camera calibration or the presence of planar surfaces. GM considerably reduces the prediction error compared to the state-of-the-art predictors when predicting over 30 frames in the future. Moreover, GM outperforms state-of-the-art predictors when processing fewer frames, thus allowing one to intentionally reduce the video frame rate and hence the energy consumption, an important aspect for vision tasks with moving cameras. Finally, we proposed a formulation that allows the integration of GM in EA-PHD-PF framework. We validated the proposed tracker on the Multiple Object Tracking Benchmark 2017 dataset and compared against state-of-the-art trackers. Experimental results showed that the use of GM improves tracking accuracy by up to 3.8 MOTA points when compared to traditional linear predictors. Moreover, the proposed tracker obtains comparable tracking results to using a linear prediction model but by processing only half of the frames, which is a desirable property for tracking with resource-constrained platforms. The proposed tracker obtains competitive results with respect to published trackers in both accuracy and speed.

The work presented in Sections 4.1, and 4.4 has been conducted in collaboration with L. Wang. L. Wang contributed developing the audio processing techniques, which are not described in this thesis. We use the output of such as techniques as input of the methods presented in this Chapter. I acknowledge that I am the principal contributor of the work presented in these sections and that is not present in any prior thesis.

Chapter 5

Conclusion

In this chapter, we first summarise the achievements and limitations of the works of this thesis (Sec. 5.1) and then, expose future research directions upon the work of this thesis (Sec. 5.2).

5.1 Summary of achievements and limitations

The main achievements of this thesis are related to the problems of object localisation and dimensions estimation, and online tracking.

Regarding the first problem, we proposed LoDE, which is able to localise and accurately estimate the dimensions of unseen objects without the need for markers or prior information of the objects such as 3D models or of the object's appearance. While existing methods, which require RGB-D sensors, obtain detection recall of 72.2%, the proposed LoDE, which only requires RGB sensors, achieves a detection rate of 87.0%. In addition to higher detection rates, the proposed method outperforms state-of-the-art alternatives regarding accuracy in estimating both the height and width of the objects (Fig. 3.10). Interestingly, LoDE is robust to the object transparency, whereas the performances of state-of-the-art methods deteriorate when the transparency of the objects increases (Fig. 3.9). We employed LoDE together with a robotic arm in a multi-modal system. This enabled the performance of the challenging human-to-robot handover of unseen drinking cups. To achieve this, the system overcame multiple challenges, mainly the limited prior knowledge of the objects, high-precision requirements to both localise and estimate the dimensions of the object in 3D, and a fast computational speed to perform dynamic handovers (i.e. natural handovers). Regarding *limitations*, LoDE is not capable of operating with objects with a different geometry than the assumed one (symmetric with respect to its vertical axis and with a circular base) as can be seen in Fig. 3.5.1 for Object 20; or when the object is in a different position than upright. In addition, the method completely relies on the performance of the semantic segmentation module that is used as input. In the case that the success rate on localising objects by the semantic segmentation module is limited (e.g. 58.94% when images are infrared), LoDE cannot work and thus is unable to estimate the shape/dimensions of the object at all. Regarding the robotic application, LoDE does not provide any information of the location of the human hand, nor on the available object grasp points, both of which are desirable for advanced and safer handovers.

Regarding the second problem of online tracking, we proposed a probabilistic method, EA-PHD-PF, that is versatile as we validated in a variety of settings including single and multiple objects, with static and moving cameras, and using audio and video signals as input. EA-PHD-PF obtains state-of-the-art tracking results while *not* using appearance information of the objects. This has two main benefits. First, even though the code lacks optimisation, the tracker performs at high speed ($\approx$ 12 fps) compared with other trackers (see last column of Table 4.12). Second, when objects are detected, the proposed tracker is invariant to their appearance thus the tracker is a suitable option for scenarios where the appearance is not informative (e.g. very similar objects such as a group of bees or people dressing with the same cloths). However, when appearance information is useful (e.g. well-lit videos containing objects with a distinguishable appearance), trackers that use appearance information might outperform our tracker in terms of accuracy and IDS, but at the cost of being slower [204, 217] (Table 4.12). The proposed online tracker is composed of six main parts that address different problems:

Multi-detector fusion (Sec. 4.2.1) enables the use of over-populated detection sets that are likely to have no false negatives but multitude of false positive and repeated detections on actual objects; the confidence-based threshold can then be tuned to vary the ratio between false positives and false negatives, as is appropriate for the application task. The proposed multi-detector fusion outperforms independent detectors and other fusion approaches (e.g. NMS) in terms of higher precision for the same level of recall (Fig. 4.26). This over-populated detection stage can be the basis for other uses as we will see in the next point. Regarding *limitations*, the fusion requires the generation of an over-populated set (e.g. using multiple detectors), which might be not computationally efficient, unless code parallelisation and optimisation is performed.

Strong and weak detections (Sec. 4.2.1 and 4.2) allow for the use of an over-populated detection set. The confidence-based initialisation and inheritance processes (Sec. 4.2) produce an important tracking accuracy increment while maintaining a similar tracking precision (Fig. 4.34 and Table 4.13). Specifically, a notable reduction in the number of IDS ($\approx 70\%$) and in the number of trajectory fragmentations ($\approx 56\%$) with a similar performance speed can be observed. Regarding *limitations*, the strong/weak classification might generate non-optimal decisions as it relies only on the confidence provided by the object detector, which might be inaccurate.

The *Early Association* (Sec. 4.2.2) allows the PHD-PF framework to include the identity estimation by performing the data association between predicted trajectories and detections, instead of associating estimated trajectories with previous trajectories. The replacement of the clustering by the early association produces two main benefits: (i) the tracking accuracy improves by more than 7.5 points and the tracking precision improves by more than 3.5 points; and (ii) the speed of the tracker improves by more than 40% as it allows removing the time-consuming clustering procedure (Table 4.13). Regarding *limitations*, experimental results show that the EA slightly increases the number of false positives, mainly caused by the wrong initialisation of new trajectories. This is mostly due to the fact that EA associates detections with trajectories and, as detections only have location information (i.e. velocity is not available as it is in the trajectories), the EA module is unable to use velocity information in the association cost function (Eq. 4.8). The lack of velocity during the association produces that in the presence of objects intersecting to each other, the discriminability between the objects is very poor as they are very close to each other in the decision space, which in the absence of speed and appearance is only composed of location and dimensions of objects.

The *perspective-dependent object motion prediction* (Sec. 4.2.3) provides more accurate predictions in static cameras and with changes in scale of the objects. This allows both a reduction in the number of false positives that, when perspective is not considered, create wrong new trajectory initialisations; and reduces the number of IDS, while maintaining the speed (Table 4.13). A *limitation* of this model is the strong assumption of that every object of a specific class has the same dimensions than all the other objects in the real world, which is often not true (e.g. people or cars are often of different size).

The *global-motion aware object motion prediction*, GM (Sec. 4.3.2), enables the prediction of the location of moving objects when they are observed from a moving camera without the need for camera calibration, additional sensors or strong assumptions on the object or on the scene. GM outperforms all considered alternatives in terms of prediction accuracy when predicting longer than one frame in the future (Table 4.9). In fact, GM obtains an accuracy improvement larger than 60% when predicting the location of objects 30 frames in the future. Also, GM performs in real-time, can handle noisy observations (Fig. 4.27) and can work accurately in videos with a reduced frame rate (Fig. 4.28). In addition to this, we showed that the use of linear motion prediction models in tracking produces large tracking drift when the camera moves (Fig. 4.36 and 4.38, and Table 4.9). Using GM as a prediction module within the EA-PHD-PF shows an important improvement in both tracking accuracy and precision (Table 4.15). Regarding *limitations*, GM might not work properly if the camera is mounted in a platform that can move at high speed (e.g. a drone), as the assumptions of pure rotation might not hold. Also, in this scenario, the fast speed of the camera is likely to capture images with high motion blur, which will make our method incapable of tracking keypoints; and, therefore, the estimation of the global motion produced by the camera motion won't be estimated accurately or estimated at all.

The *confidence intervals for tracking performance scores* quantifies the bias that datasets annotated using semi-automatically linear-based interpolation techniques causes when evaluating trackers. We showed that this type of annotations undeservedly benefit trackers that use or learn to use linear prediction models. This problem is particularly acute with moving cameras (Table 4.5). We proposed a protocol to account for this uncertainty that calculates the confidence interval for a given evaluation score and dataset using only information extracted from the already existing ground truth annotations. As *limitation*, the proposed method assumes that when interpolations are used in the annotation of a dataset, these are done by linear means. When a dataset is annotated using other type of interpolation (e.g. polynomial), the method won't be able to estimate the confidence intervals. However, if one crafts a technique to detect those type of interpolation, this can be integrated into the formulation in a straight forward manner.

The *audio-visual* calibration procedure (Sec. 3.1.2) allows algorithms to collaboratively use audio-visual for advanced applications or to enable the annotation of datasets from video signals to evaluate audio-based algorithms (Sec. 4.6.4). As *limitation*, the audio-visual calibration/inference will only work when the estimate DOA of the sound source is accurate. For instance, when the environment has strong reverberations (e.g. in an indoors environment) the estimated DOA might fail, thus making the calibration/inference procedure to fail.

Finally, we designed, collected and made publicly available two datasets: the CORSMAL containers dataset that is suitable for evaluating object localisation and dimensions estimation (Sec. 3.4), and the AVQ dataset that enables the evaluation of sound source localisation and audio enhancement in very low SNR conditions (Sec. 4.4). Also, we proposed a novel benchmark to evaluate dynamic human-to-robot handovers in scenarios without motion capture systems, markers, or prior object models [J1]. These works, besides being useful for the evaluation of the contributions of this thesis, we expect them to be the basis of other research groups for the development of more advanced algorithms.

5.2 Future work

Challenges remain open on how to design efficient and accurate algorithms for object localisation and dimensions estimation, which can be applied to unseen objects; and object tracking that can work online. We provide future directions for our work to encourage research in this area.

Regarding the problem of object localisation and dimensions estimation, algorithms that use strong prior information of objects such as 3D models of the objects, which is a common practise until the date, produce accurate results even with the presence of clutter and strong occlusions and are fast when a GPU is available. We believe this problem can be considered mostly solved. Therefore, we would like to encourage researchers to drift their focus towards the novel and unresolved problem of using unseen object for which prior information is limited or null. Initial works have been already proposed with this focus [163, 187, 220], also, an interesting research path of interest can be considering LoDE and relaxing the assumption on the geometry of the object to make it suitable for objects of any shape. Another interesting possible direction is to extend the localisation and dimensions estimation to estimate the full 6 DoF pose of the object. Also, designing algorithms that can work with only one narrow-stereo baseline can be of great interest, for example for robotic applications, where is desirable to install a limited amount of sensors and to be embedded within the robot so that the working setup does not require to be altered by installing additional sensors. We briefly explored this within the thesis (see LoDE-IR in Sec. 3.5.1) where we use an IR narrow-stereo baseline, however the semantic segmentation approach reduced its performance with infrared inputs. Therefore, it can be interesting research to explore the training of semantic segmentation models with IR images and the use of transfer learning to obtain enhanced semantic segmentation results in this setup. Other related research directions related with robotics are the estimation of the grasp points of the objects, estimation of the location of the human hand during human manipulation of objects. All the above, can produce a large impact in the robotics field allowing robots, for the first time, to perform advanced human-robot interactions while providing safer environments.

Regarding the problem of online multiple object tracking, we envision two main research directions. The first one is related to the use of the observations received by detectors. The concept of strong and weak detections can be extended to propose solutions that learn and/or adaptively tune the confidence-based threshold according to the needs of the application task such as, for example, image quality, object's dimensions or camera resolution. Investigating the integration of the strong and weak detections concept within the EA-PHD-PF might be of interest. The second one relates to one of the main causes of tracking failures in online tracking, the unresolved challenge of occlusions. One research direction is within the early association strategy, which can be improved by creating an advanced association cost to enhance the association accuracy and, specially, to reduce the number of FP initializations in the presence of object occlusions. The association cost in the EA framework could be improved by leveraging the object estimated velocity; to do this, it is needed the development of advanced detection algorithms that infer the velocity or direction of the objects from a single frame. For instance, when objects are people, this could be attempted by inferring the speed and velocity from the skeleton pose of the person. Another research direction is the design of advanced object motion predictors that can inform the tracker with more accurate predictions. This can be used to improve the uncertainty created by short-term occlusions. As discussed in this thesis, it is of great importance to accurately predict in moving cameras. To achieve this, existing SLAM algorithms could be used together with online tracking approaches for obtaining enhanced tracking results. However, when cameras move fast (e.g. mounted in UAV) motion blur happens, making existing methods that use RGB cameras to drastically reduce their performance or even to fail. In these cases, the use of event cameras which are not affected by motion blur is an interesting research area [58]. For scenarios where long-term occlusions or objects exiting and re-entering the field of view happen, a research direction is the use of re-identification mechanisms or the use of multiple sensors. The latter one allows one to cover a larger visible area and to observe objects under different views, which can potentially handle occlusions.

Bibliography

[1] Kobuki specifications. http://kobuki.yujinrobot.com/about2/. Accessed: 29-11-2019.

[2] Six degrees of freedom. https://en.wikipedia.org/wiki/Six_degrees_of_freedom. Accessed: 29-11-2019.

[3] Caviar dataset. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/, January 2004. Accessed: 29-11-2019.

[4] Imagery library for intelligent detection systems (i-lids). www.ilids.co.uk, 2012. Accessed: 29-11-2019.

[5] Multiple object tracking benchmark 2017. https://motchallenge.net/data/MOT17/, 2017. Accesed: 29-11-2019.

[6] A. Agarwala, A. Hertzmann, D. Salesin, and S. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics*, 23(3):584–591, 2004.

[7] V. Akbarzadeh, C. Gagne, and M. Parizeau. Target trajectory prediction in PTZ camera networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Portland, USA, June 2013.

[8] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 2016.

[9] A. Alahi, A. Haque, and L. Fei-Fei. RGB-W: When vision meets wireless. In *Proceedings of the IEEE International Conference on Computer Vision*, Araucano Park, Chile, December 2015.

[10] A. Alahi, V. Ramanathan, and L. Fei-Fei. Socially-aware large-scale crowd forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2211–2218, 2014.

[11] K. Ali, D. Hasler, and F. Fleuret. Flowboost - appearance learning from sparsely annotated video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, USA, June 2011.

[12] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Proceedings of the European Conference on Computer Vision*, Marseille, France, October 2008.

[13] M Andriluka and S. RotS. Roth. Schiele. Monocular 3d pose estimation and tracking by detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, June 2010.

[14] M Andriluka and S. RotS. Roth. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, June 2010.

[15] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, June 2011.

[16] J. Arrospide, L. Salgado, M. Nieto, and R. Mohedano. Homography-based ground plane detection using a single on-board camera. *IET Intelligent Transport Systems*, 4(2):149–160, 2010.

[17] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, August 2002.

[18] Y. Hioka B. Yen and B. Mace. Improving power spectral density estimation of unmanned aerial vehicle rotor noise by learning from non-acoustic information. In *Proceedings International Workshop Acoustic Signal Enhancement*, Tokyo, Japan, September 2018.

[19] M. Babaee, Z. Li, and G. Rigoll. Occlusion handling in tracking multiple people using RNN. In *Proceedings of the IEEE International Conference on Image Processing*, Athens, Greece, October 2018.

[20] Sophia Bano and Andrea Cavallaro. Vicomp: composition of user-generated videos. *Multimedia Tools and Applications*, 75(12):7187–7210, June 2016.

[21] M. Basiri, F. Schill, P. U. Lima, and D. Floreano. Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Vilamoura-Algarve, Portugal, October 2012.

[22] F. Basso, M. Munaro, S. Michieletto, E. Pagello, and E. Menegatti. Fast and robust multi-people tracking from rgb-d data for a mobile robot. In *Intelligent Autonomous Systems*, Jeju Island, Korea, June 2012.

[23] S. Becker, R. Hug, W. Hbner, and M. Arens. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. *arXiv:1805.07663*, 2018.

[24] A. Bera and D. Manocha. REACH - realtime crowd tracking using a hybrid motion model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, USA, May 2015.

[25] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. *CoRR*, abs/1903.05625, 2019.

[26] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):246–309, May 2008.

[27] B. Bethke, M. Valenti, and J. How. Cooperative vision based estimation and tracking using multiple uavs. In *Advances in Cooperative Control and Optimization: Proceedings of the International Conference on Cooperative Control and Optimization*, Berlin, Heidelberg, January 2007.

[28] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini. An interactive tool for manual, semi-automatic and automatic video annotation. *Computer Vision and Image Understanding*, 131:88–99, 2015.

[29] T. A. Biresaw, T. Nawaz, J. Ferryman, and A. I. Dell. Vitbat: Video tracking and behavior annotation tool. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Colorado Springs, USA, August 2016.

[30] S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, January 2004.

[31] M. Bolanos, M. Dimiccoli, and P. Radeva. Toward storytelling from visual lifelogging: An overview. *IEEE Transactions on Human-Machine Systems*, 47(1):77–90, 2017.

[32] J. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[33] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[34] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, September 2011.

[35] G. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, USA, June 2006.

[36] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar. Yale-CMU-Berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.

[37] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the IEEE International Conference on Computer Vision*, Cambridge, USA, June 1995.

[38] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.

[39] W. Choi, C. Pantofaru, and S. Savarese. A general framework for tracking multiple people from a moving camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1577–1591, July 2013.

[40] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *Proceedings of the European Conference on Computer Vision*, Crete, Greece, September 2010.

[41] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, July 2011.

[42] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[43] G. Ferrin D. Salvati, C. Drioli and G. L. Foresti. Beamforming-based acoustic source localization and enhancement for multirotor UAVs. In *Proceedings European Signal Processing Conference*, Rome, Italy, September 2018.

[44] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, June 2005.

[45] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[46] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking. *arXiv preprint arXiv:1905.09304*, 2019.

[47] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, August 2014.

[48] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 2019.

[49] V. Eiselein, D. Arp, M. Ptzold, and T. Sikora. Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Genoa, Italy, September 2012.

[50] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *CoRR*, abs/1607.02565, 2016.

[51] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular slam. In *Proceedings of the European Conference on Computer Vision*, Zurich, Switzerland, September 2014.

[52] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A Mobile Vision System for Robust Multi-Person Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1831–1846, 2009.

[53] A. Ess, K. Schindler, B. Leibe, and L. Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14):1707–1725, December 2010.

[54] P. F. Felzenszwalb, R.B. Girshick, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, September 2010.

[55] J. Ferryman and A. Shahrokni. PETS2009: Dataset and challenge. In *International Workshop on Performance Evaluation of Tracking and Surveillance*, Snowbird, USA, December 2009.

[56] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[57] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi. Particle phd filter based multiple human tracking using online group-structured dictionary learning. *IEEE Access*, 6:14764–14778, 2018.

[58] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *arXiv preprint arXiv:1904.08405*, 2019.

[59] S. Gao, Z. Han, D. Doermann, and J. Jiao. Depth structure association for rgb-d multi-target tracking. In *Proceedings of the IEEE Conference on Pattern Recognition*, Stockholm, Sweden, August 2014.

[60] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, June 2012.

[61] S. Gong, M. Cristani, S. Yan, and C.C. Loy. *Person Re-Identification*. 2014.

[62] F. Hamed H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, Honolulu, USA, October 2017.

[63] A. Hakeem, K. Shafique, and M. Shah. An object-based video coding framework for video sequences obtained from static cameras. In *Proceedings of the ACM International Conference on Multimedia*, Singapore, November 2005.

[64] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. 2003.

[65] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October 2017.

[66] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[67] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.

[68] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, May 1995.

[69] M. Hu, S. Ali, and M. Shah. Learning motion patterns in crowded scenes using motion flow field. In *Proceedings of the IEEE Conference on Pattern Recognition*, Tampa, USA, December 2008.

[70] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 2019.

[71] T. Ishiki and M. Kumon. Design model of microphone arrays for multirotor helicopters. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Hamburg, Germany, October 2015.

[72] A. Finzi J. Cacace, R. Caccavale and V. Lippiello. Attentional multimodal interface for multidrone search in the alps. In *Proceedings of the IEEE International Conference Systems, Man and Cybernetics*, Budapest, Hungary, October 2016.

[73] K. Y. Zhai J. R. Cauchard and J. A. Landay. Drone and me: an exploration into natural human-drone interaction. In *Proceedings ACM International Joint Conference Pervasive and Ubiquitous Computing*, Osaka, Japan, September 2015.

[74] S. Jehan-Besson, M. Barlaud, and G. Aubert. An object based motion method for video coding. In *Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[75] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *CoRR*, abs/1907.09408, 2019.

[76] K. Nagira T. Otsuka K. Itoyama K. Nakadai K. Furukawa, K. Okutani and H. G. Okuno. Noise correlation matrix estimation for improving sound source localization by multirotor uav. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Tokyo, Japan, 2013.

[77] M. Kumon K. Hoshiba, K. Nakadai and H. G. Okuno. Assessment of music-based noise-robust sound source localization with active frequency range filtering. *J. Robotics Mechatronics*, 30(3):426–435, 2018.

[78] M. Wakabayashi T. Ishiki M. Kumon Y. Bando D. Gabriel K. Nakadai K. Hoshiba, K. Washizaki and H. G. Okuno. Design of uav-embedded microphone array system for sound source localization in outdoor environments. *Sensors*, (11):1–16, 2017.

[79] H. G. Okuno K. Nakadai, M. Kumon. Development of microphone-array-embedded uav for search and rescue task. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Vancouver, Canada, September 2017.

[80] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, July 2012.

[81] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions ASME, Journal of Basic Engineering*, 82:35–45, March 1960.

[82] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[83] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, February 2009.

[84] Y. Keller and A. Averbuch. Fast motion estimation using bidirectional gradient methods. *IEEE Transactions on Image Processing*, 13(8):1042–1054, August 2004.

[85] F.S. Khan, R.M. Anwer, J. van de Weijer, A.D. Bagdanov, M. Vanrell, and A.M. Lopez. Color attributes for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, June 2012.

[86] S. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of the European Conference on Computer Vision*, Graz, Austria, May 2006.

[87] H. Kieritz, S. Becker, W. Hübner, and M. Arens. Online multi-person tracking using integral channel features. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2016.

[88] S. Kim, A. Shukla, and A. Billard. Catching objects in flight. *TRO*, 30(5):1049–1065, October 2014.

[89] L. Kratz and K. Nishino. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):987–1002, May 2012.

[90] H.W. Kuhn and B. Yaw. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[91] A. Kumar and C. De Vleeschouwer. Discriminative label propagation for multi-object

tracking with sporadic appearance features. In *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia, December 2013.

[92] T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Lecce, Italy, August 2017.

[93] S. Lankton and A. Tannenbaum. Improved tracking by decoupling camera and target motion. In *Proceedings SPIE*, February 2008.

[94] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, April 2015.

[95] S. Lee and E. Kim. Multiple object tracking via feature pyramid siamese networks. *IEEE Access*, 7:8181–8194, 2019.

[96] S. Lee, M. Kim, and S. Bae. Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures. *IEEE Access*, 6:67316–67328, 2018.

[97] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

[98] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[99] A. Leykin and R. Hammoud. Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications*, 21(4):587–595, 2010.

[100] A. Leykin, Y. Ran, and R. Hammoud. Thermal-visible video fusion for moving target tracking and pedestrian classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007.

[101] S. Li and D. Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *Association for the Advancement of Artificial Intelligence*, 2017.

[102] Y. Li, C. Huang, and R. Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, June 2009.

[103] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. DeepIM: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.

[104] Lin Lin, Yaakov Bar-Shalom, and Thiagalingam Kirubarajan. Data association combined with the Probability Hypothesis Density Filter for multitarget tracking. In *Proceedings SPIE*, August 2004.

[105] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.

[106] H. Liu, X. Yang, L. Latecki Jan, and S. Yan. Dense neighborhoods on affinity graph. *International Journal of Computer Vision*, 98(1):65–82, 2012.

[107] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul W. Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *CoRR*, abs/1809.02165, 2018.

[108] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, San Diego, USA, 2018.

[109] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[110] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras. People tracking with human motion predictions from social forces. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2010.

[111] W. Luo, B. Stenger, X. Zhao, and T. Kim. Automatic topic discovery for multi-object tracking. In *The Association for the Advancement of Artificial Intelligence*, 2015.

[112] W. Luo, X. Zhao, and T. Kim. Multiple object tracking: A review. *CoRR*, abs/1409.7618, 2014.

[113] V. Miguet M. Strauss, P. Mordel and A. Deleforge. Dregon: dataset and methods for uav-embedded sound source localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Madrid, Spain, October 2018.

[114] E. Maggio and A. Cavallaro. Learning scene context for multiple object tracking. *IEEE Transactions on Image Processing*, 18(8):1873–1884, August 2009.

[115] E. Maggio, M. Taj, and A. Cavallaro. Efficient multitarget visual tracking using random finite sets. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1016–1027, August 2008.

[116] R. Mahler. A theoretical foundation for the Stein-Winter Probability Hypothesis Density (PHD) multitarget tracking approach. In *Proceedings of MSS National Symposium on Sensor and Data Fusion*, San Diego, USA, June 2002.

[117] R. Mahler. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007.

[118] V. Malagi and K. Rangarajan. Multi-object tracking in aerial image sequences using aerial tracking learning and detection algorithm. *Defence Science Journal*, 66(2):122–129, 2016.

[119] Andrea H. Mason and Christine L. MacKenzie. Grip forces when passing an object to a partner. *Experimental Brain Research*, 163(2):173–187, May 2005.

[120] G. Máttyus, C. Benedek, and T. Szirányi. Multi target tracking on aerial videos. 2010.

[121] R. Mazzon and A. Cavallaro. Multi-camera tracking using a multi-goal social force model. *Neurocomputing*, 100:41–50, 2013.

[122] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard. A human-inspired controller for fluid human-robot handovers. In *Proceedings IEEE-RAS International Conference Humanoid Robots*, Santa Monica, USA, November 2016.

[123] David Mihalcik and David Doermann. The design and implementation of viper. *University of Maryland*, 2003.

[124] A. Milan. Ground-truth annotations for several tracking datasets. `http://www.milanton.de/data.html`. Accessed: 29-11-2019.

[125] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016.

[126] A. Milan, S. Roth, and K. Schindler. Continuoflying animals us energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):58–72, January 2014.

[127] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Portland, USA, June 2013.

[128] C. Mitash, A. Boularias, and K. E. Bekris. Robust 6d object pose estimation with stochastic congruent sets. In *Proceedings of the British Machine Vision Conference*, Newcastle, United Kingdom, September 2018.

[129] D. Mitzel, E. Horbert, A. Ess, and B. Leibe. Multi-person tracking with sparse detection and continuous segmentation. In *Proceedings of the European Conference on Computer Vision*, Crete, Greece, August 2010.

[130] R. Mohedano and N. García. Simultaneous 3d object tracking and camera parameter estimation by bayesian methods and transdimensional mcmc sampling. In *Proceedings of the IEEE International Conference on Image Processing*, Brussels, Belguim, September 2011.

[131] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *ICRA*, 2002.

[132] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, October 2015.

[133] J. Martinez-Carranza O. Ruiz-Espitia and C. Rascon. Aira-uas: an evaluation corpus for audio processing in unmanned aerial system. In *Proceedings International Conference Unmanned Aircraft Systems*, Dallas, USA, June 2018.

[134] T. Ohata, K. Nakamura, T. Mizumoto, T. Taiki, and L. Nakadai. Improvement in outdoor sound source detection using a quadrotor-embedded microphone array. In *Proceedings of*

the IEEE/RSJ International Conference on Intelligent Robot and Systems, Chicago, USA, September 2014.

[135] K. Okuma, A. Talenghani, and N. De Freitas. A boosed particle filter: multitarget detection and tracking. In *Proceedings of the European Conference on Computer Vision*, Prague, Czech Republic, May 2004.

[136] K. Okutani, T. Yoshida, K. Nakamura, and K. Nakadai. Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Vilamoura-Algarve, Portugal, October 2012.

[137] P. Mohapatra P. Misra, A. A. Kumar and P. Balamuralidhar. Aerial drones with location-sensitive ear. *IEEE Commun. Mag.*, (7):154–160, 2018.

[138] 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*, December 2005.

[139] K. Panta, B. N. Vo, Sumeetpal Singh, and Arnaud Doucet. Probability Hypothesis Density filter versus multiple hypothesis tracking. In *Proceedings of SPIE*, August 2004.

[140] Y. Park, V. Lepetit, and W. Woo. Multiple 3d object tracking for augmented reality. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2008.

[141] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. PVNet: Pixel-wise voting network for 6DoF pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 2019.

[142] J. Pestana, J. Sanchez-Lopez, P. Campoy, and S. Saripalli. Vision based gps-denied object tracking and following for unmanned aerial vehicles. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2013.

[143] J. Pestana, J. Sanchez-Lopez, S. Saripalli, and P. Campoy. Computer vision based general object following for gps-denied multirotor unmanned vehicles. In *American Control Conference*, 2014.

[144] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena. A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Brisbane, Australia, May 2018.

[145] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *ICRA*, Stockholm, Sweden, May 2016.

[146] F. Poiesi and A. Cavallaro. *Multi-target tracking in video*, chapter 19, pages 561–579. 2014.

[147] F. Poiesi and A. Cavallaro. Tracking multiple high-density homogeneous targets. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(4):623–637, April 2015.

[148] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, USA, June 2006.

[149] H. Possegger, T. Mauthner, P.M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, USA, June 2014.

[150] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, June 2018.

[151] X. Qian, A. Brutti, O. Lanz, M. Omologo, and A. Cavallaro. Multi-speaker tracking from an audio-visual sensing device. *IEEE Transactions on Multimedia*, 2019.

[152] A. Cavallaro R. Sanchez-Matilla, L. Wang. Multi-modal localization and enhancement of multiple sound sources from a micro aerial vehicle. In *Proceedings of the ACM International Conference on Multimedia*, Mountain View, USA, October 2017.

[153] RM. Rajakaruna, WA. Fernando, and J. Calic. Application-aware video coding architecture using camera and object motion-models. In *Proceedings of International Conference on Industrial and Information Systems*, Rupnagar, India, June 2011.

[154] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 2016.

[155] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, July 2017.

[156] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[157] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[158] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. In *Proceedings of the European Conference on Computer Vision*, 2016.

[159] M. Rodriguez, I. Laptev, J. Sivic, and JY. Audibert. Density-aware person detection and tracking in crowds. In *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona, Spain, November 2011.

[160] P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, February 1999.

[161] A. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa. Object detection, tracking and recognition for multiple smart cameras. *Proceedings of the IEEE*, 96(10):1606–1624, 2008.

[162] J. C. SanMiguel and A. Cavallaro. Energy consumption models for smart camera networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2661–2674, December 2017.

[163] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[164] S. Saxena, F. Bremond, M. Thonnat, and R. Ma. Crowd behavior recognition for video

surveillance. In *Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems*, Juan-les-Pins, France, October 2008.

[165] K. Shafique, Mun Wai Lee, and N. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Alaska, USA, June 2008.

[166] Jianbo Shi et al. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 1994.

[167] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Rhode Island, USA, June 2012.

[168] G. Sinibaldi and L. Marino. Experimental analysis on the noise of propellers for small uav. *Appl. Acoust.*, (1):79–88, 2015.

[169] I. Skrypnyk and D. G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *IEEE/ACM International Symp. Mixed Augmented Reality*, November 2004.

[170] F. De Smedt and T. Goedeme. Open framework for combinated pedestrian detection. In *Proceedings of Computer Vision, Imaging and Computer Graphics Theory and Applications*, Berlin, Germany, March 2015.

[171] P. Smith, I. Reid, and A. Davison. Real-time monocular slam with straight lines. In *Proceedings of the British Machine Vision Conference*, Edinburgh, Scotland, September 2006.

[172] F. Solera, S. Calderara, and R. Cucchiara. Learning to divide and conquer for online multi-target tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.

[173] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, June 2015.

[174] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[175] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[176] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia, December 2013.

[177] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. *International Journal of Computer Vision*, 110(1):58–69, 2014.

[178] J-M. Valin, F. Michaud, J. Rouat, and D. Létourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Las Vegas, USA, October 2003.

[179] D. Varga, T. Szirányi, A. Kiss, L. Spórás, and L. Havasi. A multi-view pedestrian tracking method in an uncalibrated camera network. In *Proceedings of the IEEE International Conference on Computer Vision*, Araucano Park, Chile, December 2015.

[180] D. Vasquez and T. Fraichard. Motion prediction for moving objects: a statistical approach. In *ICRA*, New Orleans, USA, April 2004.

[181] B.-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. In *Proceedings of Information Fusion*, Queensland, Australia, July 2003.

[182] B. N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245, October 2005.

[183] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.

[184] B. Wang, G. Wang, K.L. Chan, and L. Wang. Tracklet association with online target-specific metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, USA, June 2014.

[185] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. DenseFusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 2019.

[186] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, 119(3):219–238, September 2016.

[187] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 2019.

[188] L. Wang and A. Cavallaro. Ear in the sky: Ego-noise reduction for auditory micro aerial vehicles. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Colorado Springs, USA, 2016.

[189] L. Wang and A. Cavallaro. Time-frequency processing for sound source localization from a micro aerial vehicle. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, New Orleans, USA, March 2017.

[190] L. Wang and A. Cavallaro. Acoustic sensing from a multi-roto drone. *IEEE Sensors Journal*, (11):4570–4582, 2018.

[191] L. Wang, Gerkmann, and S. Doclo. Noise power spectral density estimation using maxnsr blocking matrix. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1493–1508, September 2015.

[192] L. Wang, T. K. Hon, J. D. Reiss, and A. Cavallaro. An iterative approach to source counting and localization using two distant microphones. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(6):1079–1093, June 2016.

[193] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 2019.

[194] T. Wang, C. Yang, F. Kirchner, P. Du, F. Sun, and B. Fang. Multimodal grasp data set: A novel visual-tactile data set for robotic manipulation. *International Journal Advanced Robotic Systems*, 16(1):1–10, January 2019.

[195] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3–19, 2013.

[196] X. Wang, T.X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[197] X. Wang, X. Ma, and W.E.L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian model. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 31(3):539–555, March 2009.

[198] Y. Wang and A. Cavallaro. Active visual tracking in multi-agent scenarios. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Lecce, Italy, September 2017.

[199] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, USA, June 2014.

[200] C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3d scene understanding with explicit occlusion reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, USA, June 2011.

[201] J.K. Wolf, A.M. Viterbi, and G.S. Dixon. Finding the best set of K paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2):287–296, March 1989.

[202] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, November 2007.

[203] L. Xi, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Hengel. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, 4(4):58:1–58:48, October 2013.

[204] Y. Xiang, A. Alahi, and S. Savarese. Learning to Track: Online Multi-Object Tracking by Decision Making. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.

[205] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, Pittsburgh, USA, June 2018.

[206] J. Xiao and M. Oussalah. Collaborative tracking for multiple objects in the presence of inter-occlusions. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(2):304–318, February 2016.

[207] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, USA, June 2011.

[208] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, June 2012.

[209] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, June 2012.

[210] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Proceedings of the European Conference on Computer Vision*, Florence, Italy, October 2012.

[211] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 2016.

[212] Y. Yang, G. Shu, and M. Shah. Semi-supervised learning of feature hierarchies for object detection in a video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Portland, USA, June 2013.

[213] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(13):1–45, December 2006.

[214] A. Yilmaz, K. Shafique, and M. Shah. Target tracking in airborne forward looking infrared imagery. *Image and Vision Computing*, 21(7):623–635, 2003.

[215] S. Yoon, S. Park, Y. Eom, and S. Yoo. Advanced sound capturing method with adaptive noise reduction system for broadcasting multicopters. In *IEEE International Conference on Consumer Electronics*, Las Vegas, USA, 2015.

[216] Y. Young-Chul, B. Abhijeet, Y. Kwangjin, and J. Moongu. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. *CoRR*, abs/1805.10916, 2018.

[217] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan. POI: Multiple object tracking with high performance detection and appearance feature. In *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, October 2016.

[218] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.

[219] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[220] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Qu Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3750–3757, 2018.

[221] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, USA, June 2008.

[222] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

[223] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.

[224] Z. Zou, Z. Shi, Y. Guo, and J. Ye. Object detection in 20 years: A survey. *CoRR*, abs/1905.05055, 2019.