Super-Resolution Approaches for Depth Video Enhancement

Kassem Al-Ismaeil

PhD-FSTC-2015-42
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 07/09/2015 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Kassem AL ISMAEIL
Born on 25 January 1981 in Aleppo, (Syria)

SUPER-RESOLUTION APPROACHES FOR DEPTH VIDEO ENHANCEMENT

Dissertation defense committee

Dr. Björn Ottersten, dissertation supervisor
Professor, Université du Luxembourg

Dr-Ingénieur Holger Voos, Chairman
Professor, Université du Luxembourg

Dr. Stefano Berretti, Vice Chairman
Professor, Università di Firenze

Dr. Cédric Demonceaux
Professor, Université de Bourgogne

Dr. Bruno Mirbach,
Algorithm Group Leader, IEE S.A.

To my uncle
Farouk Tabour.
May you rest in peace.

Abstract

Sensing using 3D technologies has seen a revolution in the past years where cost-effective depth sensors are today part of accessible consumer electronics. Their ability in directly capturing depth videos in real-time has opened tremendous possibilities for multiple applications in computer vision. These sensors, however, have major shortcomings due to their high noise contamination, including missing and jagged measurements, and their low spatial resolutions. In order to extract detailed 3D features from this type of data, a dedicated data enhancement is required. We propose a generic depth multi–frame super–resolution framework that addresses the limitations of state-of-the-art depth enhancement approaches. The proposed framework does not need any additional hardware or coupling with different modalities. It is based on a new data model that uses densely upsampled low resolution observations. This results in a robust median initial estimation, further refined by a deblurring operation using a bilateral total variation as the regularization term. The upsampling operation ensures a systematic improvement in the registration accuracy. This is explored in different scenarios based on the motions involved in the depth video. For the general and most challenging case of objects deforming non-rigidly in full 3D, we propose a recursive dynamic multi–frame super-resolution algorithm where the relative local 3D motions between consecutive frames are directly accounted for. We rely on the assumption that these 3D motions can be decoupled into lateral motions and radial displacements. This allows to perform a simple local per–pixel tracking where both depth measurements and deformations are optimized. As compared to alternative approaches, the

results show a clear improvement in reconstruction accuracy and in robustness to noise, to relative large non-rigid deformations, and to topological changes. Moreover, the proposed approach, implemented on a CPU, is shown to be computationally efficient and working in real-time.

Acknowledgements

First and foremost, I would like to express my special appreciation and thanks to my advisor Professor Björn Ottersten for his advice, guidance, and support on both levels professional and personal. It was a great honor and privilege to be part of his SIGCOM Research Group. All my interactions with the group have been very constructive especially with members of the Computer Vision Team.

I would like to also express my gratitude to my industrial supervisor Dr. Bruno Mirbach for his invaluable comments and feedback which helped in always relating my work to real-world applications. In addition, I thank all team members at the Advanced Engineering Department at IEE for the fascinating discussions and support. I owe a lot to the amazing Thomas Solignac, Senior Computer Vision Engineer at IEE, for being extremely generous by sharing with me some of his exceptional experience. I would like to tell him "Merci beaucoup mon ami".

Furthermore, I would like to express my gratitude to Professor Holger Voos for serving on my PhD Supervisory Committee and for accepting to chair the PhD Defense Committee. I also thank Professor Stefano Berretti and Professor Cédric Demonceaux for accepting to evaluate this work as Vice Chairman and Member, respectively, of the Defense Committee.

This thesis would not be the same without my mentor, friend and wife Dr. Djamila Aouada. I am forever grateful for her continuous guidance, ideas, and patience which were my main source of encouragement throughout these years. I would like her to know that she is my model of an exceptional researcher and scientist.

Finally, my greatest thanks and appreciation go to my family; my wife, my parents, my siblings, Koala, and Dabdoob, for their huge love, patience, and support.

This thesis has been funded by the Fonds National de la Recherche (FNR), Luxembourg, under the code C11/BM/1204105/FAVE as part of the CORE project FAVE.

Contents

Notation

In this thesis, matrices are denoted by boldface, uppercase letters, $\mathbf{M}$, and vectors are denoted by boldface, lowercase letters, $\mathbf{v}$. Scalars are denoted by italic letters, *e.g.,* x, K, α. The following mathematical notation will be used:

$\mathbf{M}^T$	transpose of matrix $\mathbf{M}$
$\mathbf{M}^{-1}$	inverse of matrix $\mathbf{M}$
$\mathbf{M}\uparrow$	upsampling of matrix $\mathbf{M}$
$\mathbf{M}^{(\ell)}$	matrix $\mathbf{M}$ at level or iteration ℓ
$\mathbf{I}_n$	identity matrix of dimension n by n
$\mathbf{0}$	matrix of zeros
$\mathbb{1}_r$	vector of length r whose elements are equal to 1
$\|\mathbf{v}\|_2$	L_2 norm of vector $\mathbf{v}$
$\|\mathbf{v}\|_1$	L_1 norm of vector $\mathbf{v}$
$\mathbf{v}^{\mathtt{i}}$	the $\mathtt{i}$–th element of vector $\mathbf{v}$
$\overline{\mathbf{v}}$	registered version of vector $\mathbf{v}$
$\hat{\mathbf{v}}$	estimate of $\mathbf{v}$
$x \to \infty$	x tends to infinity
$\arg\min$	the minimizing argument
$\mathtt{p}$	pixel position
$\mathtt{p}_t^{\mathtt{i}}$	pixel position at time t under index $\mathtt{i}$
$\mathcal{Z}$	depth surface
$\frac{df}{dx}$	derivative of f with respect to x
$\frac{\partial f}{\partial x}$	partial derivative of f with respect to x
$p(\cdot)$	probability density function
$\mathrm{tr}(\mathbf{M})$	trace of $\mathbf{M}$

$\mathrm{cov}(\mathbf{v})$	covariance of $\mathbf{v}$
$\mathrm{var}(\mathbf{v})$	variance of $\mathbf{v}$ defined as $\mathrm{tr}(\mathrm{cov}(\mathbf{v}))$
$\mathrm{sign}(\cdot)$	sign function
$\mathrm{div}_{\mathbf{v}}(\cdot)$	divergence with respec to $\mathbf{v}$

Abbreviations

AWGN	Additive White Gaussian Noise
BF	Bilateral Filter
BTV	Bilateral Total Variation
CAD	Computer Aided Design
CPU	Central Processing Unit
eS&A	enhanced Shift & Add
GPU	Graphics Processing Unit
HR	High Resolution
ICP	Iterative Closest Point
i.i.d	independent and identically distributed
JBU	Joint Bilateral Upsampling
LR	Low Resolution
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MRF	Markov Random Field
MSE	Mean Square Error
PSF	Point Spread Function
PSNR	Peak Signal to Noise Ratio
PyrME	Pyramidal Motion Estimation
RAM	Random Access Memory
RecUP-SR	Recursive Upsampling for Precise Super–Resolution
RMSE	Root Mean Square Error
S&A	Shift & Add
SISR	Single Image Super–Resolution
SL	Structured Light

SR	Multi–Frame Super–Resolution
SNR	Signal to Noise Ratio
SURE	Stein's Unbiased Risk Estimate
SurfUP-SR	Surface Upsampling for Precise Super–Resolution
ToF	Time-of-Flight
UP-SR	Upsampling for Precise Super–Resolution
VBSR	Variational Bayesian Super–Resolution

List of Figures

LIST OF FIGURES

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation and scope

Sensing using 3D technologies, structured–light (SL) cameras or time–of–flight (ToF) cameras [3, 4], has seen a revolution in the past years where sensors such as the Microsoft Kinect version 1 and 2 are today part of accessible consumer electronics [5]. Examples of these cameras are shown in Figure 1.1. The ability of these sensors in directly capturing depth videos in real–time has opened tremendous possibilities for applications in gaming, robotics, surveillance, health care, etc. These sensors, unfortunately, have major shortcomings due to their high noise contamination, including missing and jagged measurements, and their low spatial resolutions. This makes it impossible to capture detailed 3D features indispensable for many 3D computer vision algorithms. The face data in Figure 1.2 is an example of such challenging raw depth measurements. Running a traditional face recognition algorithm on this type of data would result in a very low recognition rate [14, 15, 16]. Some solutions have been proposed in the literature for recovering these details. Most of the work proposed to enhance the resolution of this data has been based on fusion with high resolution (HR) images acquired with a second camera, e.g., 2D camera [10, 17, 18], stereo camera [19, 20], or both 2D and stereo cameras [21]. These multi–modality methods provide solutions with undesired texture copying artifacts in addition to being highly dependent on parameter tuning. Moreover, using an additional camera requires dealing with data mapping and synchronization issues. Also, due to cost,

(a) D-IMager (b) SR4000 (c) CamboardNano

(d) 3D MLI (e) Kinect v1 (f) Kinect v2

Figure 1.1: Example of depth Cameras. (a) D-IMager by Panasonic – (160 × 120) pixels [1], (b) SwissRanger SR4000 ToF camera by MESA Imaging – (176 × 144) pixels [2], (c) CamboardNano ToF camera by PMD – (160 × 120) pixels [3], (d) 3D MLI Sensor ToF camera by IEE – (56 × 61) pixels [4], (e) Kinect v1 structured light camera by PrimeSense – (640 × 480) pixels [5], (f) Kinect v2 ToF camera by PrimeSense – (640 × 480) pixels [5].

in many applications it is not possible to use additional imaging chips or optical components.

In order to obtain higher quality depth videos without additional hardware, one may think of using concepts from multi–frame image reconstruction. The key idea is to compensate for the limitations of the imaging system by fusing multiple frames captured with the same system. When the reconstruction involves resolution enhancement, we talk about multi–frame super–resolution (SR) [22]. Its goal is to recover an HR image from a set of LR images captured with the same camera by exploring the deviations between these LR images and a reference frame. SR approaches have been largely explored in 2D imaging. Recently, few attempts have been carried out in order to extend 2D SR algorithms to static depth scenes. The extension of these algorithms to depth data is not straightforward due to the textureless nature of depth data. In [9], a dedicated preprocessing has been proposed to achieve depth SR from a single image but

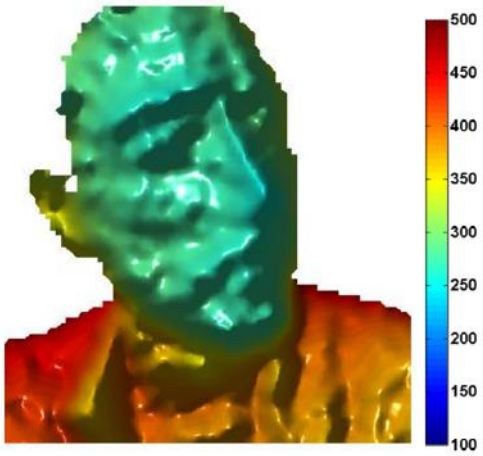


Figure 1.2: 3D plotting of a low resolution depth frame captured using a ToF camera. The 3D point cloud is created by back projecting the depth values using the camera intrinsic parameters. Distance units on the colored bar are in mm.

using a large database for training. In the more challenging case of dynamic depth scenes, this extension is even more difficult due to artifacts caused by fast motions.

The scope of this thesis is to define a new generic depth SR framework that addresses the limitations of state-of-art depth enhancement approaches and that is adapted to the properties of depth data. Our objective is to use this framework for the simpler case of static depth scenes, and most importantly to extend it to the challenging case of a depth video with freely moving objects. In what follows we review the challenges specific to each case.

1.1.1 Enhancement of static depth scenes

A scene is considered static if the motion from one frame to another is global where the frames could be seen as slightly different perspectives of the same scene. In this case, the SR estimation is usually solved numerically by iterative methods starting from an initial image followed by an optimization step. The quality of the initial image remains a point of weakness for most existent methods. This image may be obtained by an operation known as *Shift & Add* (*S&A*) [7, 23]

which includes a filling procedure based on the global relative motion of the considered LR images. In 2008, Schuon et al. proposed to apply the *S&A* method on depth data in a multi–frame setup [24]. While this work showed that SR may be used successfully on depth scenes without any training, it is still not a practical solution because of the large number of required LR observations. Indeed, for a weak diversity in the motions available in the LR depth images, the initial HR image suffers from undefined pixels. As a practical remedy, it is often necessary to capture a relatively large number of LR frames in order to increase the diversity in motion. An extended version of [24] has been proposed by the same authors in [25] by defining a new cost function dedicated to depth data. Both approaches, in [24] and [25], do not solve the limitation on the number of required frames inherent to classical *S&A*; thus, they remain not practical solutions. Another initialization method is by aligning the LR measurements on an HR grid and interpolating the missing points. A promising method in this category is the work by Babacan et al. referred to as *variational Bayesian* SR (*VBSR*) [26]. In the case of textureless depth data, such interpolation induces erroneous values and flying pixels that are difficult to attenuate. Recently, Newcombe et al. have presented the *KinectFusion* algorithm for real–time 3D reconstruction and interaction using a moving Microsoft Kinect camera [27]. This algorithm is based on a dense Iterative Closest Point (ICP) tracking algorithm and hence HR depth observations are required for the algorithm to converge.

1.1.2 Enhancement of dynamic depth scenes

Using multiple frames to recover depth details has been successful in the case of static scenes or scenes with a global rigid motion [24, 25, 27]. Since these methods and their immediate derivatives, the real challenge that the research community has been facing is extending the multi–frame depth enhancement concept to scenes with non-rigid deformations. It is these scenes that we will refer to as dynamic depth scenes.

There have been few attempts to handle single object scanning under relative

small non-rigidities by replacing a global rigid registration with a non–rigid alignment [28, 29, 30]. These techniques, however, cannot handle large deformations, and are not very practical for real–time applications. Real–time non–rigid reconstruction approaches have been achieved with the help of a template which is first acquired then used for tracking of non–rigidities with a good flexibility [31, 32]. Recently, Afzal et al. have proposed the first non–rigid version of *KinectFusion* algorithm named *KinectDeform* [33] and later on its extension [34]. This method does not require any template, and similarly to *KinectFusion*, provides an enhanced smoother reconstruction over time with the addition of handling non–rigid deformations in the scene. *KinectDeform* has been successfully tested on an Asus Xtion Pro Live camera [35], equivalent to Microsoft Kinect structured light version 1. It cannot, however, perform well on lower resolution, noisier ToF cameras such as the PMD CamboardNano [3]. Indeed, its registration module requires denser raw acquisitions. *DynamicFusion* [36] is another recent non–rigid version of *KinectFusion*. Thanks to a GPU implementation, it has been tested on a Kinect camera in real–time. However, its reconstruction accuracy has not been evaluated, and it has only been validated visually. Moreover, it builds on the assumption of having only one moving object in the scene. In addition, its reported limitations are its sensitivity to complex scenes and scenes with changes in topology. Also, similarly to *KinectDeform*, one may suspect *DynamicFusion* not to be able to perform well on a lower resolution noisier ToF camera.

1.2 Objectives and contributions

The objective of this thesis is to address the aforementioned limitations of both static and dynamic depth enhancement methods under the SR framework. The main contributions are listed below:

1. **Upsampling for a robust depth SR:** In order to be able to deploy multi–frame SR algorithms in practice, specifically *S&A*, without requiring a very high number of observed LR frames, we improve the initial estimation of the HR frame. To that end, we propose a new data model resulting from densely upsamled LR observations and leading to a median estimation.

This new formulation solves the problem of undefined pixels. Moreover, we show the impact of upsampling in increasing sub–pixel accuracy and reducing the rounding error of motion vectors. This allows to improve the performance of pyramidal motion estimation in the context of SR. As a consequence, it increases the motion diversity within a small number of observed frames, making the enhancement of depth data more practical. Quantitative experiments run on the Middlebury dataset [37] show that our method outperforms state-of-art techniques in terms of accuracy and robustness to the number of frames and to the noise level.

This work has been published in [38] and [39] and some extended parts are under review in [40].

2. **Dynamic depth SR for non–rigid deformations:** Most depth SR methods available in the literature are dedicated to static depth scenes. None of them has addressed the enhancement of dynamic depth scenes with non–rigidly deforming objects. In this thesis, we propose the *UP-SR* algorithm, which stands for *Upsampling for Precise Super–Resolution*, as the first dynamic multi–frame depth video SR algorithm that can enhance depth videos containing non–rigidly deforming scenes without any prior assumption on the number of moving objects they contain or on the topology of these objects. These advantages are possible thanks to a direct processing on depth maps without using connectivity information inherent to meshing as used in subsequent methods, namely, *KinectDeform* [33, 34] and *DynamicFusion* [36]. The *UP-SR* algorithm is based on a data model that uses densely upsampled, and cumulatively registered versions of the observed LR depth frames. With the proposed cumulative motion estimation, a high registration accuracy is achieved between non–successive upsampled frames with relative large motions. A statistical performance analysis is derived in terms of mean square error (MSE) explaining the effect of the number of observed frames and the effect of the SR factor at a given noise level. We evaluate the accuracy of the proposed algorithm theoretically and experimentally as function of the SR factor, and the level of contamination with

noise. Experimental results on both real and synthetic data show the effectiveness of the proposed algorithm on dynamic depth videos as compared to state-of-art methods.

This work has been published in [13] and [41]. An extended version is currently under review [40].

3. **Real–time recursive SR for dynamic depth scenes with non–rigid deformations:** Although the *UP-SR* algorithm is able to handle dynamic depth scenes containing multiple moving objects, it is limited to lateral motions as it only computes 2D dense optical flow and does not account for the full motion in 3D, known as scene flow, or the 2.5D motion, known as range flow. It consequently fails in the case of radial deformations. Moreover, it is not practical because of a heavy cumulative motion estimation process applied to a number of frames buffered in the memory. Thus, we define a new recursive dynamic multi–frame SR algorithm, *recUP-SR*, which improves over the *UP-SR* algorithm by keeping its advantage and solving its limitations – not considering 3D motions and using an inefficient cumulative motion estimation. The key idea is by directly accounting for the relative local 3D motions between consecutive frames. We rely on the assumption that these 3D motions can be decoupled into lateral motions and radial displacements. This allows to perform a simple local per–pixel tracking. The geometric smoothness is subsequently added using a multi–level L_1 minimization with a bilateral total variation (BTV) regularization. The performance of this method is thoroughly evaluated on both real and synthetic data. As compared to alternative approaches, the results show a clear improvement in reconstruction accuracy and in robustness to noise, to relative large non–rigid deformations, and to topological changes. Moreover, the proposed approach, implemented on a CPU, is shown to be computationally efficient and working in real–time.

This work has been published in [42] and its extended version is currently under review [43].

4. **Bilateral filter evaluation based on exponential kernels:** Our work, similarly to the *S&A* approach, is based on two steps; first a blurred estimation where the data fusion happens followed by a deblurring step. The deblurring represents an important element in adding the geometrical smoothness to the blurred estimated solution. Throughout this thesis, we adopt a regularized L_1 optimization in the deblurring phase. We use the BTV as a regularization term [7]. This choice is motivated by the fact that the properties of a bilateral filter, namely, noise reduction while preserving edges, is now established as an appropriate method for depth data processing [9, 27, 44]. This filter is commonly used with Gaussian kernel functions without real justification. The choice of the kernel functions has a major effect on the filter behaviour. We propose to use exponential kernels with L_1 distances instead of Gaussian ones. We derive Stein's Unbiased Risk Estimate (SURE) to find the optimal parameters of the new filter and compare its performance with the conventional one. We show that this new choice of the kernels has a comparable smoothing effect but with sharper edges due to the faster, smoothly decaying kernels. We further propose a multi–level version of the L_1 optimization with a BTV regularization in a similar fashion as in [45, 46, 47]. This process leads to effectively deblurring the intermediate blurred solution while keeping fine details without over–smoothing.

   This work has been published in [48], and part of it is currently under review [43].

5. **Improved face recognition using LR depth cameras:** Enhancing depth data captured with cost-effective depth sensors should have an important impact on their deployment in real–world applications.
   We choose face recognition as one such application and propose to tailor our SR framework for the enhancement of 3D facial data. A new algorithm is proposed. It is called *SurfUP-SR* which stands for *Surface Upsampling for Precise Super–Resolution*. Considering a face as a surface in 3D, we reformulate the *UP-SR* algorithm on a 3D point cloud instead of its original formulation on a depth image. This reformulation allows an efficient

implementation, and leads to a largely enhanced 3D face reconstruction. It hence improves the 3D face recognition rate while using cost–effective LR depth cameras. In addition, we provide a tool for an automatic 3D face reconstruction from data acquired with a PMD CamboardNano camera [3]. Experimental evaluation of *SurfUP-SR* using a real LR 3D face dataset has been carried out. Obtained results show an efficient enhancement in the resolution and the quality of the raw faces. Moreover, *SurfUP-SR* is shown to decrease the 3D reconstruction error, and most importantly to increase the 3D face recognition rate.

This work has been published in [15].

1.3 Publications

JOURNALS

1. **K. Al Ismaeil**, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten, "Real–Time Enhancement of Dynamic Depth Videos with Non–Rigid Deformations". IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2015. (Under review)

2. **K. Al Ismaeil**, D. Aouada, B. Mirbach, and B. Ottersten, "Enhancement of Dynamic Depth Scenes by Upsamplig for Precise Super–Resolution (*UP–SR*)". International Journal in Computer Vision and Image Understanding (CVIU), Springer, 2015. (Under review)

CONFERENCES

1. **K. Al Ismaeil**, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten, "Real-Time Non-Rigid Multi-Frame Depth Video Super-Resolution", IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 8-16, 2015 – **Best Paper Award**

2. D. Aouada, **K. Al Ismaeil**, B. Ottersten, "Patch-based Statistical Performance Analysis of Upsampling for Precise Super-Resolution", 10th Inter-

national Conference on Computer Vision Theory and Applications. (VISAPP), pp. 186-193, 2015.

3. D. Aouada, **K. Al Ismaeil**, K. Kedir, B. Ottersten, "Surface UP-SR for an Improved Face Recognition Using Low Resolution Depth Cameras", 11th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 107-112, 2014.

4. **K. Al Ismaeil**, D. Aouada, B. Mirbach, and B. Ottersten, "Dynamic Super Resolution of Depth Sequences with Non-Rigid Motions", 20th IEEE International Conference on Image Processing (ICIP), pp. 660-664, 2013.

5. **K. Al Ismaeil**, D. Aouada, B. Mirbach, and B. Ottersten, "Depth Super-Resolution by Enhanced Shift and Add", 15th International Conference in Computer Analysis of Images and Patterns (CAIP), pp. 100-107, 2013.

6. **K. Al Ismaeil**, D. Aouada, B. Mirbach, and B. Ottersten, "Mutli-Frame Super-Resolution by Enhanced Shift & Add", 8th IEEE International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 171-176, 2013.

7. **K. Al Ismaeil**, D. Aouada, B. Mirbach, and B. Ottersten, "Bilateral Filter Evaluation Based on Exponential Kernels", 21st IAPR International Conference on Pattern Recognition (ICPR), pp. 258-261, 2012.

PUBLICATIONS NOT INCLUDED IN THE THESIS

1. H. Afzal, **K. Al Ismaeil**, D. Aouada, F. Destelle, B. Mirbach, B. Ottersten, "KinectDeform: Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects", 11th IEEE International Conference on 3D Vision Workshops (3DVW), pp. 1-7, 2014.

2. A. Habed, **K. Al Ismaeil**, D. Fofi, "A New Set of Quartic Trivariate Polynomial Equations for Stratified Camera Self-calibration under Zero-Skew and Constant Parameters Assumptions", 12th European Conference on Computer Vision (ECCV), pp. 710-723, 2012.

1.4 Thesis outline

The organization of this dissertation is as follows:

- **Chapter 2:** In order to understand the properties and challenges of the considered depth data, necessary background on cost–effective depth cameras, their principle, and characteristics are given. In addition, backgrounds on motion estimation and multi–frame SR are reviewed.
- **Chapter 3:** A practical and robust multi–frame SR method for static scenes with global lateral motions is presented. We explain the source of the limitations of the initialization step in the *S&A* algorithm and give our solution.
- **Chapter 4:** The *UP-SR* algorithm is presented as our first proposed solution for dynamic depth scenes. Its cumulative motion estimation is decribed and evaluated experimentally. A statistical performance analysis for *UP-SR* is then derived.
- **Chapter 5:** The *UP-SR* algorithm is reformulated in a recursive manner for the sake of real–time applications. The resulting *recUP-SR* is presented as our new solution to handle radial deformations in addition to lateral ones. Qualitative and quantitative experimental evaluations are presented and discussed.
- **Chapter 6:** A comparison of the performance of the bilateral filter using Gaussian and exponential kernels is given. Furthermore, the SURE risk function for a bilateral filter using exponential kernels is derived in order to find the filter optimal parameters.
- **Chapter 7:** The *UP-SR* algorithm is reformulated as a new approach on 3D point clouds and incorporated in a 3D face reconstruction pipeline. The impact of this enhancement is illustrated experimentally for 3D face reconstruction and validated for 3D face recognition using real data.
- **Chapter 8:** Concluding remarks and perspectives on future work building on the contributions of this thesis are discussed.

Chapter 2

Background

Chapter 2 reviews the basic concept of multi-frame SR and formulates the general problem of this work by laying down the model and assumptions common to all the parts of the thesis.
This work addresses the enhancement of depth videos captured with cost–effective depth sensors. There are two components that guide the design of the algorithms described in subsequent chapters. These components are the properties of the data as captured by the sensors and the motions in the scene. We therefore introduce the working principles of the considered sensors and their characteristics. Finally, we review the concepts of motion estimation in 2D and in 3D.

2.1 Multi–frame super–resolution

2.1.1 General data model

As discussed in Chapter 1, multi–frame SR may be thought of as an alternative solution to enhancing the quality of depth images acquired with a cost–effective depth camera. Indeed, this image processing concept has already been explored in the case of other optical imaging systems [22, 49, 50] where additional hardware cannot be added because of cost or implementational issues. The goal of multi–frame SR is to reconstruct a higher quality image from a set of low quality acquisitions captured with the same optical device. It is casted as an inverse problem; so we start by first defining the data model which is assumed to be

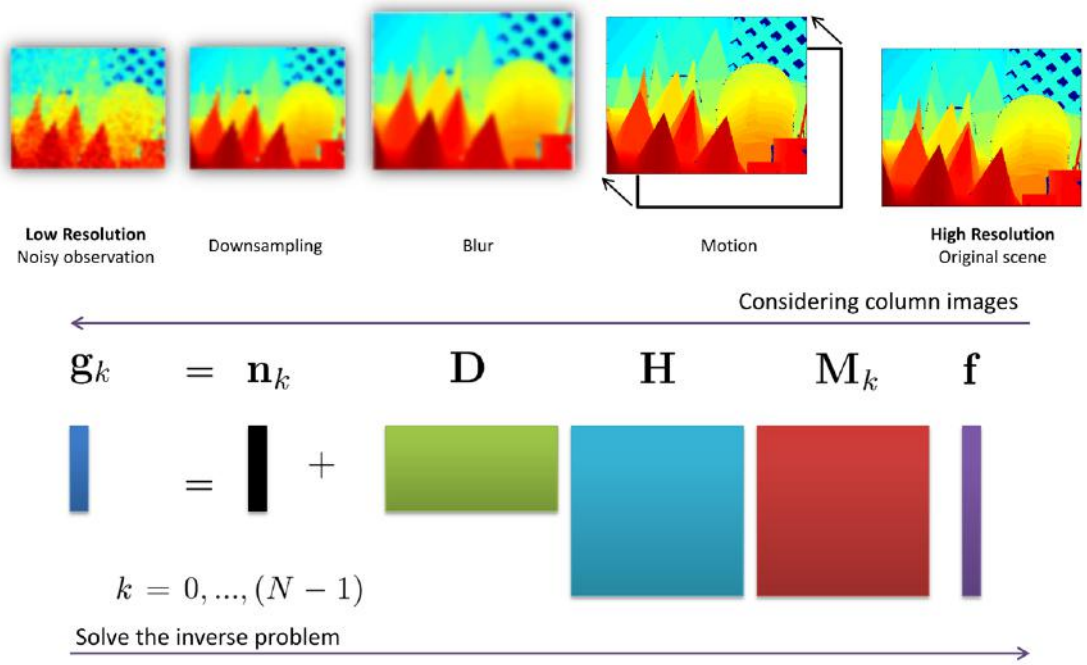


Figure 2.1: Illustration of the multi–frame super–resolution data model.

linear as follows.

$$\mathbf{g}_k = \mathbf{L}_k\mathbf{f} + \mathbf{n}_k, \quad k = 0, \cdots, N-1, \tag{2.1}$$

where $\mathbf{g}_k$ is the k^{th} observed measurement, $\mathbf{L}_k$ represents the imaging system, and $\mathbf{n}_k$ is a random additive noise. The HR image that we want to estimate is $\mathbf{f}$. All images are represented as column vectors following a lexicographic ordering. The unknown image $\mathbf{f}$ is assumed to be of size $(n \times 1)$ while the size of $\mathbf{g}_k$ is $(m \times 1)$ with $n = r^2 \cdot m$. The factor r is known as the *SR factor* and corresponds to the targeted increase in spatial resolution.

To estimate $\mathbf{f}$, a cost function $C(\cdot)$ has to be defined based on some definition of closeness between the estimate and the measurements, also called *fidelity*. The simplest cost function that has been traditionally adopted is the sum of least squares where the L_2 norm of residuals is to be minimized [11], as follows

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmin}}\, C(\mathbf{f}) = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{k=0}^{N-1} \|\mathbf{L}_k\mathbf{f} - \mathbf{g}_k\|_2^2. \tag{2.2}$$

When the noise $\mathbf{n}_k$ is zero mean white Gaussian, this solution is equivalent to the Maximum Likelihood (ML) estimate that maximizes the conditional probability

of the observations given the original image, i.e., $p(\mathbf{g}_1, \cdots, \mathbf{g}_N/\mathbf{f})$ [11, 51]. The solution in (2.2) is however not a stable one because SR is an ill-posed problem with the system being often under-determined, i.e., $N < r^2$. An undesired consequence of this is the amplification of noise in the final solution. This is why imposing some prior information on the solution by means of regularization is necessary to reach a stable SR solution. The cost function $C(\cdot)$ would therefore have an additional term such that

$$C(\mathbf{f}) = \sum_{k=0}^{N-1} \|\mathbf{L}_k\mathbf{f} - \mathbf{g}_k\|_2^2 + \lambda\Gamma(\mathbf{f}), \tag{2.3}$$

where $\Gamma(\cdot)$ is the regularization function, or *penalty* function, that imposes the prior information on $\mathbf{f}$, and λ is a parameter that controls the strength of the penalization.

The model of the imaging system $\mathbf{L}_k$ dictates the computational complexity of minimizing (2.3), and the performance of the resulting solution. Typically, the matrix $\mathbf{L}_k$ is modelled as follows

$$\mathbf{L}_k = \mathbf{DHM}_k, \tag{2.4}$$

with $\mathbf{M}_k$ being an $(n \times n)$ matrix corresponding to the geometric motion between $\mathbf{f}$ and $\mathbf{g}_k$. The optical blur is modelled by the point spread function (PSF) of the camera represented by the $(n \times n)$ space and time invariant blur matrix $\mathbf{H}$ which is block circulant. The sampling process is modelled by the downsampling matrix $\mathbf{D}$ of dimension $(m \times n)$. This model is illustrated in Figure 2.1.

By assuming that the motion is translational, the matrix $\mathbf{M}_k$ is consequently block circulant and $\mathbf{M}_k$ and $\mathbf{H}$ become commutative [52]. As a result, the SR estimation may be decomposed into two subtasks as detailed next in Section 2.1.2.

2.1.2 Two-step estimation

Given that $\mathbf{M}_k\mathbf{H} = \mathbf{HM}_k$, the data model in (2.1) can be rewritten as

$$\mathbf{g}_k = \mathbf{DM}_k\mathbf{z} + \mathbf{n}_k, \quad k = 0, \cdots, N-1, \tag{2.5}$$

with $\mathbf{z} = \mathbf{Hf}$ being a blurred version of $\mathbf{f}$. Estimating $\mathbf{f}$ may thus be decomposed into two main steps: 1) Estimation of the blurred HR image $\mathbf{z}$ by fusion and/or

interpolation. 2) Deblurring by an iterative optimization in order to find the final estimate of $\mathbf{f}$. Such a two-step estimation has been adopted by multiple state-of-art approaches as it helps a computationally more efficient implementation [7, 23, 52, 53, 54].

In [23], Farsiu et al. have shown that the assumption of a Gaussian additive noise $\mathbf{n}_k$ is not the closest to reality. Instead, it should be more of a heavy tailed distribution, and specifically a Laplacian distribution is considered to be a better candidate. As a result, the first step of estimating the blurred HR image $\mathbf{z}$ follows from (2.5) as

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmin}} \sum_{k=0}^{N-1} \|\mathbf{D}\mathbf{M}_k\mathbf{z} - \mathbf{g}_k\|_1. \tag{2.6}$$

where an L_1 norm is used in place of the L_2 norm in (2.2). This, in turn, has shown to provide a robust blurred solution as presented in [7, 11, 23]. When accompanied with the proper zero-filling and motion compensation, the operation in (2.6) is known as *Shift & Add* (*S&A*).
Afterwards, comes the deblurring step which is in general similar to (2.3) with a slight change in the fidelity term such that

$$C(\mathbf{f}) = \|\mathbf{H}\mathbf{f} - \hat{\mathbf{z}}\|_1 + \lambda\Gamma(\mathbf{f}). \tag{2.7}$$

In addition to playing the role of deblurring, the operation in (2.7) helps in recovering desired properties in the final image $\hat{\mathbf{f}}$ through an appropriate choice of the regularization function $\Gamma(\cdot)$. In [25], Schuon et al. proposed a regularization term tailored for depth data, leading to a new depth–dedicated SR method referred to as *LidarBoost*. The aim of *LidarBoost* is to preserve areas with a smooth geometry. To that end, it implements a regularization term that is a function of spatial gradients approximated with finite differences. The original *LidarBoost* uses an L_2–norm of weighted depth gradients. In order to better accommodate the needs of detailed 3D object scanning, Cui et al. proposed a new version of *LidarBoost* where the regularization term is set to be an anisotropic non–linear function of gradients [55]. In both cases, however, the initial HR is obtained by means of averaging, which is not appropriate for sensing cluttered depth scenes. In Section 2.2, we review the properties of depth videos as captured by cost–effective depth sensors.

2.2 Cost–effective depth sensing

In less than a decade, depth sensors have become accessible devices at very affordable prices. The best example of such sensors is the Microsoft Kinect version 1 [5] based on the structured–light (SL) principle, or more recently its newest version based on the time–of–flight (ToF) principle. These two technologies fall under the category of active sensing. Indeed, they both rely on an active source that illuminates the scene of interest. In order to understand the properties and challenges of the data captured by these cameras, we review below their respective working principles.

2.2.1 Working principles

- **Structured–light cameras:** SL cameras are composed of a projector, e.g., a near infra–red (NIR) laser projector, and an intensity camera, e.g., a monochrome CMOS camera. The projector emits a specially designed light pattern on the scene of interest. The camera then sees a deformed pattern depending on the geometry of the scene. In the illustration of Figure 2.2, the example of a simple pattern in the form of a straight line is projected. The camera perceives a straight line when the projection is on a flat wall. It perceives a deformed curve when the projection is on a more complex shape such as a cylinder. From these deformations, the depth measurement can be extracted with the knowledge of camera intrinsic parameters.

- **Time–of–flight cameras:** ToF cameras are composed of an active NIR projector and an optical sensor. The projector illuminates the scene with a phase modulated signal. The optical sensor then captures the intensity of the reflected signal as well as its phase. At the level of each pixel, the difference in phase between the emitted and the received signals is used to calculate the time it took for the signal to travel from the sensor to the scene. This time can finally easily be converted to the corresponding distance, i.e., depth measurement, using the speed of light. Figure 2.3 illustrates the ToF working principle.

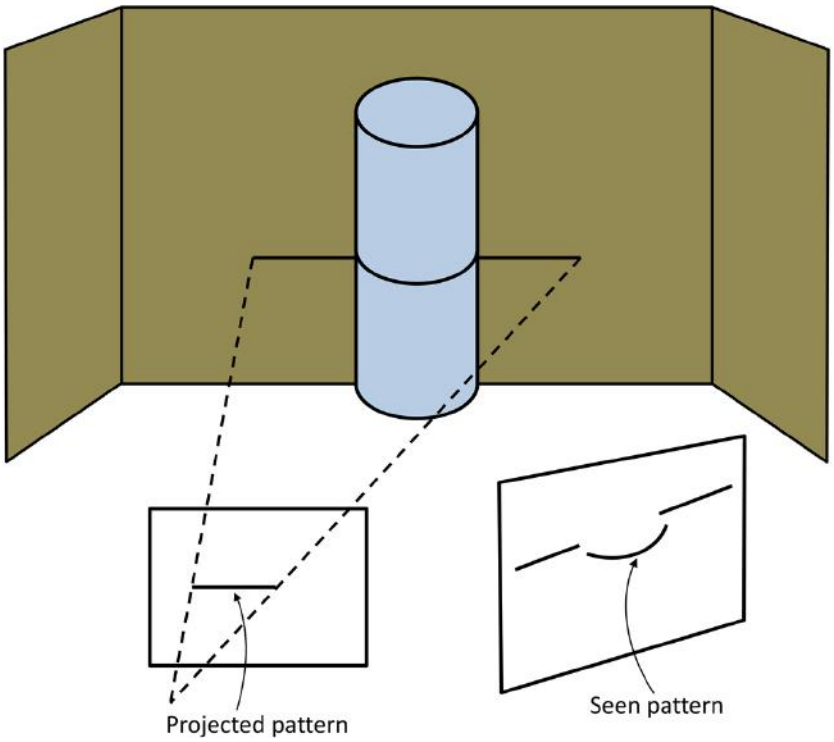


Figure 2.2: Illustration of the working principle of a structured–light camera. (Reproduced from [6]).

2.2.2 Data properties

The following are the different challenges presented by depth data captured with a ToF or an SL depth camera [56, 57, 58].

1. **Systematic depth errors:** Systematic errors are due to the low spatial resolution of cameras. These errors are perceived mainly for objects that are far from the camera. This results in imprecise depth measurements. Also, because of the active illumination in both SL and ToF sensors, the random photon shot noise, which is the dominant category of noise, increases with the increase of the distance from the scene. In the case of ToF cameras, there are additional errors caused by approximations of the emitted sinusoidal signal or approximations of the demodulation function.

2. **Non–systematic depth errors:** Long exposure times cause over saturation which in the case of SL cameras makes it difficult to detect light

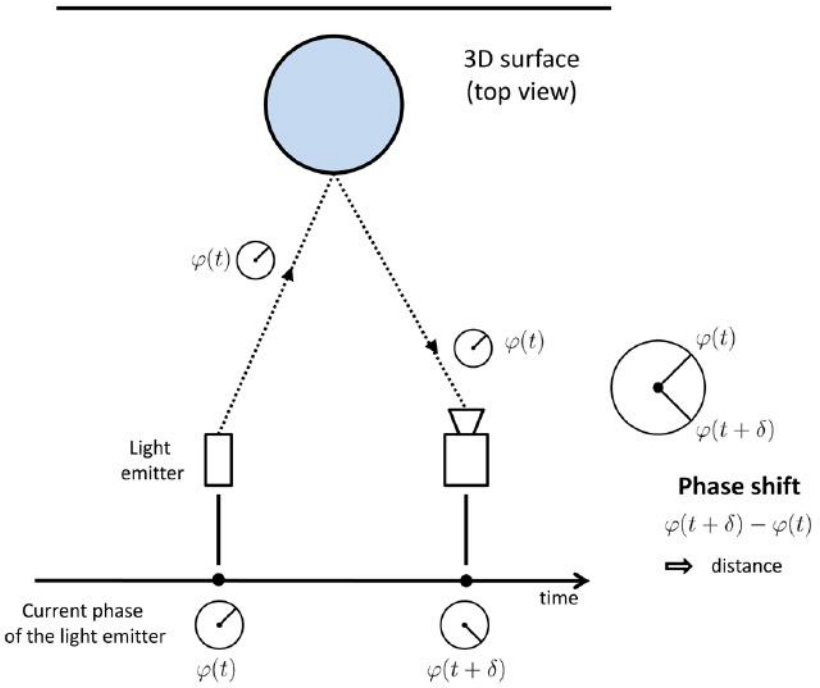


Figure 2.3: Working principle of a time–of–flight camera. (Reproduced from [6]).

patterns, and hence difficult to use outdoors. ToF cameras are equipped with filters for background light suppression that make them relatively more robust to background light. This light, nonetheless, causes unwanted noisy pixels. ToF and SL sensors also suffer from inhomogeneous depth values especially on object boundaries. They are called flying pixels for ToF–acquired data resulting from a mixture of signals reflected from surfaces at different depths, usually a mixture of foreground and background values. In the case of SL cameras, occlusions may occur on boundaries where no light reaches causing invalid pixels.

3. **Motion errors:** Moving objects cause special artifacts in the form of altered depth measurements. For SL cameras, this is due to errors in the detected pattern. These errors are larger for faster motions. In the case of the ToF technology, multiple acquisitions, usually four, are used to find the phase difference and assign a depth value to one pixel. Motions may cause mismatches between these acquisitions which leads to errors. More

specifically these errors have been classified in [59] as follows:

- Artifacts from lateral motions which result from mixing background and foreground phase values at the boundary of moving objects.
- Artifacts from radial motions which are caused by phase changes for an object moving radially and hence having a varying depth.
- Artifacts from texture changes which occur for objects of varying reflectivity leading to changes in phase while no depth changes actually occur.

2.3 Motion estimation

In order to compensate for the effect of motion, an accurate motion estimation has to be achieved. Radial motions in the depth direction combined with lateral motions constitute the motion in 2.5D or the so–called *range flow* [60]. This type of motion is often encountered in depth videos. We review below its concept.

A time–varying depth surface $\mathcal{Z}$ may be viewed as a mapping of a pixel position $\mathtt{p}_t^{\mathtt{i}} = (x_t^{\mathtt{i}}, y_t^{\mathtt{i}})$ on the sensor image plane, at a time instant t, and defined as follows:

$$\begin{aligned} \mathcal{Z}: \quad \mathbb{R}^2 \times \mathbb{N} \quad &\to \mathbb{R} \\ \mathtt{p}_t^{\mathtt{i}} \quad &\mapsto \mathcal{Z}(x_t^{\mathtt{i}}, y_t^{\mathtt{i}}). \end{aligned} \tag{2.8}$$

The value $\mathcal{Z}(x_t^{\mathtt{i}}, y_t^{\mathtt{i}})$ corresponds to the $\mathtt{i}^{\text{th}}$ element of the depth image $\mathbf{z}_t$ written in lexicographic vector form, that we will denote in what follows as $\mathbf{z}_t^{\mathtt{i}}$. The deformation of the surface $\mathcal{Z}$ from $(t-1)$ to t takes the point $\mathtt{p}_{t-1}^{\mathtt{i}}$ to a new position $\mathtt{p}_t^{\mathtt{i}}$. It may be expressed through the derivative of $\mathcal{Z}$ following the direction of the 3D displacement resulting in a range flow $(u_t^{\mathtt{i}}, v_t^{\mathtt{i}}, w_t^{\mathtt{i}})$ where the radial displacement in the depth direction $w_t^{\mathtt{i}} = \frac{d\mathbf{z}_t^{\mathtt{i}}}{dt}$ is added as the third component to the lateral displacement $\mathtt{m}_t^{\mathtt{i}} = (u_t^{\mathtt{i}}, v_t^{\mathtt{i}})$ where $u_t^{\mathtt{i}} = \frac{dx_t^{\mathtt{i}}}{dt}$ and $v_t^{\mathtt{i}} = \frac{dy_t^{\mathtt{i}}}{dt}$.

Applying the depth constraint to motion, we find the range flow constraint as first proposed in [60] and later used in [61, 62, 63, 64]. It is defined as follows:

$$u_t^{\mathtt{i}} \frac{\partial \mathbf{z}_t^{\mathtt{i}}}{\partial x_t^{\mathtt{i}}} + v_t^{\mathtt{i}} \frac{\partial \mathbf{z}_t^{\mathtt{i}}}{\partial y_t^{\mathtt{i}}} - w_t^{\mathtt{i}} + \frac{\partial \mathbf{z}_t^{\mathtt{i}}}{\partial t} = 0. \tag{2.9}$$

The range flow equation (2.9) is usually used in a variational framework to estimate the range flow $(u_t^{\mathtt{i}}, v_t^{\mathtt{i}}, w_t^{\mathtt{i}})$. However, estimating a dense range flow, i.e., a three dimensional vector for each point $\mathtt{p}_t^{\mathtt{i}}$, for $\mathtt{i} = 1, \cdots, n$, is still computationally complex and not achievable in real-time, at least, not with a sub–pixel accuracy [65]. Using RGB-D depth cameras has allowed a multi-modal approach for range flow estimation by adding a standard 2D *optical flow* constraint applied on available intensity images $\mathbf{a}_{t-1}$ and $\mathbf{a}_t$ such that:

$$u_t^{\mathtt{i}} \frac{\partial \mathbf{a}_t^{\mathtt{i}}}{\partial x_t^{\mathtt{i}}} + v_t^{\mathtt{i}} \frac{\partial \mathbf{a}_t^{\mathtt{i}}}{\partial y_t^{\mathtt{i}}} + \frac{\partial \mathbf{a}_t^{\mathtt{i}}}{\partial t} = 0, \tag{2.10}$$

where $\mathbf{a}_t^{\mathtt{i}}$ denotes the $\mathtt{i}^{\text{th}}$ element of the intensity image $\mathbf{a}_t$. A global energy functional combining (2.9) and (2.10), regularized with some smoothness condition, is then optimized [64, 66].

Chapter 3

Robust Depth SR for Static Scenes

We enhance the *Shift & Add* super–resolution approach to increase the resolution of depth data using a set of low resolution images related by global relative motion. In order to be able to deploy such a framework in practice, without requiring a very high number of observed low resolution frames, we propose a new data model that leads to a median estimation from densely upsampled low resolution frames, hence, solving the problem of undefined pixels and increasing the motion diversity within a small number of observed frames.

3.1 Introduction

SR is a common technique used to recover an HR reference image from a set of observed LR images subject to errors due to the optical acquisition system such as noise and blurring, and to deviations from the reference image due to relative motion. The past two decades have witnessed several contributions on SR for static scenes [7, 67, 68, 69, 70, 71, 72]. Most of the proposed methods are dedicated to a simple translational or affine motion. As presented in [73], these algorithms are numerically limited to small global motions even for an increased number of LR frames. Most SR techniques start with constructing the initial HR grid with sub–pixel accuracy by combining the LR frames by interpolation.

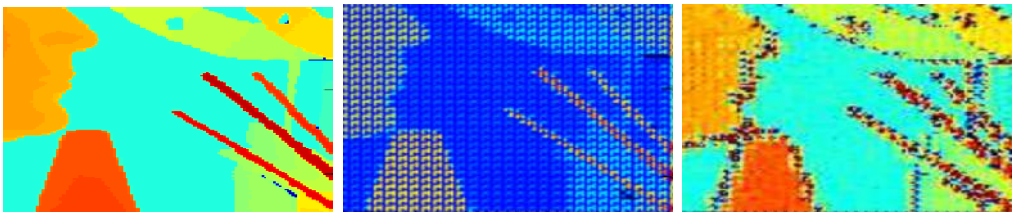
These methods work effectively when a sufficient number[1] of LR images contain slightly different perspectives of the scene. It is critical to start with an initial HR image that is as accurate as possible. The initial image may be obtained by an operation commonly referred to as *Shift & Add* (*S&A*) [7] which includes a filling operation based on the motion of the considered LR images. Another method is by aligning the LR measurements on an HR grid and interpolating the missing points, the most successful method is the variational Bayesian SR (*VBSR*) [26]. Once an initial HR image is designed, it is refined with an optimization process by minimizing a given cost function to finally reach the desired HR image. The main drawback of these methods is that the quality of the initial HR image is restricted to a specific range of motions related to the SR factor. Indeed, a weak motion diversity among the LR frames leads to undefined pixels in the initial HR image resulting in artifacts in the final solution and a strong deterioration of the SR performance. As a solution to this, example–based SR algorithms have been proposed [9], and their combinations with classical multi–frame SR [74]. Such algorithms rely on a heavy learning phase, and assume that images carry some redundancies.

In this work we propose to release the limitations on scale and the number of required frames of classical SR algorithms without prior assumptions on the data and without engaging in an additional learning stage. Our method is based on an accurate registration of frames to the reference frame resulting in an enhanced *S&A* algorithm. Our strategy consists in using the efficient pyramidal optical flow estimation starting from LR frames upsampled up to the SR factor. This is followed by a pixel–wise median operation which guarantees that no undefined pixels appear in the initial HR image and it is further refined by a selective optimization.

3.2 Background

Let $\mathbf{f}$ be an HR depth image in the form of a column vector of length n and let $\mathbf{g}_k$, $k = 0, ..., (N-1)$, be N observed LR depth images, where each LR image is

[1] Note that this number is bounded as proven in [73].



(a) Original HR (b) *S&A* [7] (c) *VBSR* [26]

Figure 3.1: Unreliable and flying pixels in the initial estimate $\hat{\mathbf{z}}$ using state-of-the-art SR methods (Red colors are the closest objects and green colors are the furthest ones.)

a column vector of length m, such that $n = r^2 \cdot m$, with r being the SR factor. Every frame $\mathbf{g}_k$ may be viewed as an LR noisy and deformed realization of $\mathbf{f}$ caused by the depth acquisition system at the k-th acquisition. Using a two–step estimation as presented in Section 2.1.2, the estimate $\hat{\mathbf{f}}$ is found as the result of minimizing (2.7). Starting with an accurate initial blurred estimate $\hat{\mathbf{z}}$ has a strong impact on the final solution. The classical *S&A* approach [7] defines $\mathbf{z}$ by first setting it to a zero vector of size $(n \times 1)$.

Then, all LR images $\mathbf{g}_k$ are used to update its pixel values. To that end, given a reference LR image $\mathbf{g}_0$ chosen as the closest one to the target HR image $\mathbf{f}$, the global translational motions $\mathbf{M}_k$ between each image $\mathbf{g}_k$ and $\mathbf{g}_0$ are estimated for $k = 1, \cdots, (N-1)$. The estimated motions $\hat{\mathbf{M}}_k$ are used to register all LR images $\mathbf{g}_k$ with respect to the reference image $\mathbf{g}_0$. The resulting registered images $\overline{\mathbf{g}}_k$ are defined as:

$$\overline{\mathbf{g}}_k = \hat{\mathbf{M}}_k \mathbf{g}_k. \tag{3.1}$$

These images are then grouped into S sets clustered based on their relative motions $\mathtt{m}_k$. The frames contained in one set are fused by median filtering resulting in one LR image $\tilde{\mathbf{g}}_i$ per motion $\mathtt{m}_i$, with $1 \leq i \leq S \leq N$. Each frame is then used to update the pixels of $\mathbf{z}$ at a given position $\mathtt{p}$ as follows:

$$\mathbf{z}\left(r \circ (\mathtt{p} + \mathtt{m}_i)\right) = \tilde{\mathbf{g}}_i(\mathtt{p}), \tag{3.2}$$

where we define $\circ$ as an upsampling by r of a lexicographic position. This operation is known as zero–filling in the *S&A* approach. We note that for a successful filling, there should be enough motion diversity in the considered LR frames, i.e.,

a sufficient number of sets S. Indeed, in order to further update the zero pixels in $\mathbf{z}$, an additional $(r \times r)$ median filtering is applied. Given that the median filter's breakdown point is $\frac{1}{2}$, a meaningful filling that does not leave pixels undefined is achieved if the following condition is satisfied:

$$\text{round}\left(\frac{r^2}{2}\right) \leq S. \tag{3.3}$$

This condition is, however, not always satisfied. We show the effect of undefined pixels in $\mathbf{z}$ caused by classical *S&A* in Figure 3.1(b). A similar phenomenon is observed using interpolation-based initialization such as *VBSR* [26] as seen in Figure 3.1(c), suggesting that interpolation is not an adequate solution to remove undefined pixels. Moreover, it creates additional artifacts on depth data such as jagged values on edges.

The problem of undefined pixels often occurs in practice. It is dealt with by restricting the SR factor to low values, e.g., $r = 2$, and by taking a relatively large number of frames, e.g., $N > 30$, thus indirectly attempting to satisfy inequality (3.3). This in turn, limits the practical use of SR algorithms. In Section 3.3, we propose to increase the motion diversity S by upsampling the LR frames in order to give more freedom in the choice of r without having to increase the number of observations N.

3.3 Enhanced pyramidal motion

In the SR problem, a highly accurate motion estimation with a $\pm\frac{1}{2}$ sub–pixel accuracy at the HR level is desired. This corresponds to a sub–pixel accuracy of $\pm\frac{1}{2r}$ at the LR level. To reach this objective, two ways may be considered:

1. Tuning the parameters of the chosen optical flow algorithm until the desired accuracy is reached, then multiplying the LR motion vectors by the SR factor r;

2. Upsampling the LR frames prior to estimating motion.

The main disadvantage of the former solution is that full knowledge of the used optical flow algorithm and its parameters is needed. In addition, modifying the

parameters in order to increase the accuracy requires increasing the number of iterations in the optical flow related optimization process. On the other hand, the latter solution could be seen as a more systematic option. The choice between these two solutions is totally based on the targeted application. Either ways, the registration has to be done at the upsampled level in order to attenuate the rounding error of motion vectors.

In this work, we propose to follow the second option, and to upsample the observed LR images even before registering them. We further detail the advantages of this approach in the context of pyramidal motion estimation (PyrME) [75, 76]. Indeed, PyrME is the principle followed by most optical flow algorithms used in the SR framework. PyrME uses the pyramidal strategy to increase sub–pixel accuracy and robustness to large motions as compared to estimating motions directly from observed frames. In what follows, we describe PyrME as it is currently used. Then, we present how we further improve its performance in the context of the SR problem. Let $\mathtt{m}_k = (u_k, v_k)$ be the motion vector between a frame $\mathbf{g}_k$ and the reference frame $\mathbf{g}_0$ at a given target point $\mathtt{p}$. This motion vector is estimated by minimizing the following error:

$$\xi(\mathtt{m}_k) = \sum_{\mathtt{q}=\mathtt{p}-\mu}^{\mathtt{p}+\mu} \|\mathbf{g}_0(\mathtt{q}) - \mathbf{g}_k(\mathtt{q}+\mathtt{m}_k)\|_2^2. \tag{3.4}$$

This error is calculated within an integration disc of radius μ, which corresponds to the largest motion that can be detected within this framework. The center of this disc is represented by the target pixel position $\mathtt{p}$. A small value of μ increases the sub–pixel motion accuracy while a large value is preferable in order to increase robustness to large motions. PyrME was proposed as a trade–off solution for these conflicting characteristics. The main idea is to follow a coarse to fine strategy that progressively downsamples the images $\mathbf{g}_k$ and $\mathbf{g}_0$ starting from the bottom of the pyramid. These images are downsampled by a factor $2^{(\ell)}$ in the dyadic case, where ℓ indicates the pyramidal level, $\ell = 0, \cdots, L$. Considering two consecutive levels ℓ and $(\ell - 1)$, the downsampling process may be defined as follows:

$$\mathbf{g}_k^{(\ell)}(\mathtt{p}) = \mathbf{g}_k^{(\ell-1)}(2\mathtt{p}) \quad s.t. \quad \mathbf{g}_k^{(0)} = \mathbf{g}_k, \quad \forall k. \tag{3.5}$$

In fact, the number of the pyramidal levels L is directly related to the considered minimum size of the downsampled image at the highest level of the pyramid. Let us define this minimum size as $(d \times d)$ pixels. Then, we may define the maximal number of pyramidal levels as:

$$\frac{\sqrt{m}}{2^L} = d \;\; \Rightarrow \;\; L = \log_2\left(\sqrt{m}\right) - \log_2\left(d\right). \tag{3.6}$$

Starting from the top of the pyramid, the motion is first estimated from the images of lowest resolution, i.e. at the highest level $\ell = L$, before progressively going back down to the images of highest resolution, i.e., at the initial level $\ell = 0$. At each level ℓ, the motion $\mathtt{m}_k^{(\ell)}$ between the two images $\mathbf{g}_k^{(\ell)}$ and $\mathbf{g}_0^{(\ell)}$ consists of an initial estimate $\omega_k^{(\ell)}$ and a residual motion $\phi_k^{(\ell)}$. The initial estimate $\omega_k^{(\ell)}$ is obtained from the preceding level $(\ell + 1)$ such that $\omega_k^{(\ell)} = 2 \cdot \mathtt{m}_k^{(\ell+1)}$, and initially set to zero at the level $\ell = L$. The two images $\mathbf{g}_k^{(\ell)}$ and $\mathbf{g}_0^{(\ell)}$ are then pre–registered using the initial motion vector. This pre–registration step reduces the process of finding the optimal motion $\mathtt{m}_t^{(\ell)}$ to finding the optimal residual motion. The estimation of the optimal residual motion is then defined by the following minimization:

$$\phi_k^{(\ell)} = \underset{\nu}{\operatorname{argmin}} \sum_{\mathtt{q}=\mathtt{p}-\mu}^{\mathtt{p}+\mu} \|\mathbf{g}_0^{(\ell)}(\mathtt{q}) - \mathbf{g}_k^{(\ell)}(\mathtt{q} + \omega_k^{(\ell)} + \nu)\|_2^2. \tag{3.7}$$

The optimal motion at level ℓ is then defined as $\mathtt{m}_k^{(\ell)} = \omega_k^{(\ell)} + \phi_k^{(\ell)}$. In order to have a high sub–pixel resolution accuracy, a small neighbourhood disc of radius μ is considered in the refinement operation defined in (3.7). By repeating the operation in (3.7) for all the levels of the pyramid, the finest motion vector is obtained at $\ell = 0$ defining $\mathtt{w}_k$ as:

$$\mathtt{m}_k := \mathtt{m}_k^{(0)} = \omega_k^{(0)} + \phi_k^{(0)}. \tag{3.8}$$

We may also express this motion using the refined residuals at all levels as follows:

$$\mathtt{m}_k = \sum_{\ell=0}^{L} 2^{(\ell)} \phi_k^{(\ell)}. \tag{3.9}$$

The maximal pixel motion vector that can be detected at any level ℓ is restricted by the initial motion vector from the preceding level and the radius of the neighbourhood disc μ in (3.7). By considering all the refined residuals as in (3.9), the

maximal overall pixel motion that can be detected at the level $\ell = 0$ by PyrME is within a maximum radius of:

$$\mu_{\max} = G(L) \times \mu \quad \text{with} \quad G(L) = 2^{(L+1)} - 1. \tag{3.10}$$

From (3.10), we see that the maximal motion is controlled by the gain $G(L)$ and the radius of the neighbourhood disc μ. The gain $G(L)$ is a function of the height L of the pyramid. By considering a small μ while increasing the number of pyramidal levels, PyrME may estimate large motions up to $\mu_{\max}$; hence, verifying the robustness property in addition to the accuracy one.
In the context of the SR problem, our target is to increase the resolution of the LR images up to the resolution of the final HR images with size $(\sqrt{n} \times \sqrt{n})$ pixels. By increasing the resolution, we thus increase the number of pyramidal levels. This gives us a natural way to further improve the performance of PyrME by upsampling the LR frames up to the SR factor r prior to any motion estimation. This upsampling step directly impacts the two properties of PyrME :

- *Robustness:*
  The upsampling step leads to changing the size of the pyramid base and hence changing the starting point in PyrME. These changes result, in turn, to an increased pyramidal height $L \uparrow^r$ by $\log_2(r)$ which results in a new and higher gain $G(L \uparrow^r)$:

  $$G(L \uparrow^r) = r \cdot G(L) + (r-1), \quad \text{with } r > 1. \tag{3.11}$$

  The result in (3.11) shows that, in the SR context, the robustness to large motions for PyrME, may further be enhanced with a new larger gain $G(L \uparrow^r)$.

- *Accuracy:*
  By increasing the resolution with a factor r, the initial motion vector at the new level can be estimated from $\mathtt{m}_k^{(0)}$ in (3.8) as $\omega_k^{(-\log_2(r))} = r \cdot \mathtt{m}_k^{(0)}$. Hence,

the optimal refined final motion can be further defined as:

$$\begin{aligned} \mathtt{m}_k := \mathtt{m}_k^{(-\log_2(r))} &= \omega_k^{(-\log_2(r))} + \phi_k^{(-\log_2(r))} \\ &= r \cdot (\omega_k^{(0)} + \phi_k^{(0)}) + \phi_k^{(-\log_2(r))}. \end{aligned} \tag{3.12}$$

By back projecting the newly refined motion in (3.12) to the original resolution at the level $\ell = 0$, we have:

$$\mathtt{m}_k^{(0)} = \omega_k^{(0)} + \phi_k^{(0)} + \frac{\phi_k^{(-\log_2(r))}}{r}. \tag{3.13}$$

Comparing (3.8) and (3.13), we find an increase in accuracy of $\delta \mathtt{w}_k(r) = \frac{1}{r} \cdot \phi_k^{(-\log_2(r))}$. This confirms the result in [77] which shows that higher image resolutions help in increasing the accuracy of motion estimation. We note that the advantage of upsampling for PyrME saturates when a certain accuracy increase is reached, i.e., $\lim_{r\to\infty} \delta \mathtt{m}_k(r) = 0$.

3.4 Dense upsampling

Following the result in Section 3.3, we use the enhanced PyrME and follow an upsampling strategy as a starting point for a new improved SR algorithm. We define the r–times upsampling of the observed LR image $\mathbf{g}_k$ as $\mathbf{g}_k \uparrow = \mathbf{U}\mathbf{g}_k$, where $\mathbf{U}$ is an $(n \times m)$ upsampling matrix.

Due to the specific properties of depth data, classical interpolation–based methods, such as bicubic interpolation, cannot be used as they lead to flying pixels and to blurring effects especially for boundary pixels. Thus, the upsampling $\mathbf{U}$ has to be dense, which is also known as nearest neighbour upsampling. For our problem, it is defined by the following matrix:

$$\mathbf{U} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q} \end{bmatrix}, \tag{3.14}$$

where $\mathbf{0}$ is a zero matrix, and $\mathbf{Q}$ represents the blocks of $\mathbf{U}$ of size $(\sqrt{n}r \times \sqrt{m})$. The dense upsampling implies that

$$\mathbf{Q} = \left[\underbrace{\mathbf{P}^T, \cdots, \mathbf{P}^T}_{r \text{ times}} \right]^T, \tag{3.15}$$

where T denotes the matrix transpose, and $\mathbf{P}$ is a matrix of size $(\sqrt{n} \times \sqrt{m})$ such that:

$$\mathbf{P} = \begin{bmatrix} \mathbb{1}_r & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbb{1}_r & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbb{1}_r \end{bmatrix} \quad \text{with} \quad \mathbb{1}_r = \underbrace{[1, \cdots, 1]}_{r \text{ times}}^T. \tag{3.16}$$

We assume in what follows that the upsampling matrix $\mathbf{U}$ is the transpose of the downsampling matrix $\mathbf{D}$. Their product $\mathbf{UD} = \mathbf{A}$ gives another block circulant matrix $\mathbf{A}$ that defines a new blurring matrix $\mathbf{B} = \mathbf{AH}$. The matrix $\mathbf{A}$ is actually a block diagonal matrix with the square matrix $\mathbf{QQ}^T$ repeated $\sqrt{m}$ times on its diagonal. Considering that $\mathbf{B}$ and $\mathbf{M}_k$ are block circulant matrices, we have $\mathbf{BM}_k = \mathbf{M}_k\mathbf{B}$. As a result, the initialization described in Section 3.2 gets modified where a new blurred HR image $\mathbf{z} = \mathbf{Bf}$ is to be estimated first.

3.5 Proposed algorithm

Considering (2.1) and (2.4), the classical data model for a static scene is given as:

$$\mathbf{g}_k = \mathbf{DHM}_k\mathbf{f} + \mathbf{n}_k, \quad k = 0, \cdots, N-1, \tag{3.17}$$

We assume that the additive noise $\mathbf{n}_k$ follows a white multivariate Laplace distribution as it has been shown to better fit the SR problem as compared to a Gaussian noise model [23]. This distribution is defined as follows:

$$p(\mathbf{n}_k) = \prod_{i=1}^{m} \frac{\sqrt{2}}{2\sigma} \exp\left(-\frac{\sqrt{2}|\mathbf{n}_k(i)|}{\sigma}\right), \tag{3.18}$$

where $\frac{\sigma}{\sqrt{2}}$ is a positive Laplace scale factor leading to the diagonal covariance matrix $\mathbf{\Sigma} = \sigma^2\mathbf{I}_m$, with $\mathbf{I}_m$ being the identity matrix of size $(m \times m)$.
By left multiplying (3.17) by $\mathbf{U}$ we find:

$$\mathbf{g}_k \uparrow = \mathbf{M}_k\mathbf{Bf} + \mathbf{Un}_k, \qquad k = 0, ..., (N-1). \tag{3.19}$$

In addition, similarly to [78], for analytical convenience, we assume that all pixels in $\mathbf{g}_k \uparrow$ originate from pixels in $\mathbf{f}$ in a one to one mapping. Therefore, each row in $\mathbf{M}_k$ contains 1 for each position corresponding to the address of the source

pixel in $\mathbf{f}$. This bijective property implies that the matrix $\mathbf{M}_k$ is an invertible permutation. Following the result in Section 3.3, its estimate $\hat{\mathbf{M}}_k$ is obtained from upsampled LR frames $\mathbf{g}_k \uparrow$, $k = 0, \cdots, (N-1)$. The corresponding registrations to the reference $\mathbf{g}_0 \uparrow$ are performed as

$$\mathbf{g}_k \uparrow = \hat{\mathbf{M}}_k \overline{\mathbf{g}}_k \uparrow . \tag{3.20}$$

Given (3.20), by left multiplying (3.19) by $\hat{\mathbf{M}}_k^{-1}$, we find

$$\overline{\mathbf{g}}_k \uparrow = \mathbf{B}\mathbf{f} + \boldsymbol{\nu}_k, \qquad k = 0, ..., (N-1). \tag{3.21}$$

This finally leads to a new simplified SR data model which is analogous to a classical image denoising problem using multiple observations, specifically

$$\overline{\mathbf{g}}_k \uparrow = \mathbf{z} + \boldsymbol{\nu}_k, \quad k = 0, \cdots, N-1, \tag{3.22}$$

where $\boldsymbol{\nu}_k = \hat{\mathbf{M}}_k^{-1}\mathbf{U} \cdot \mathbf{n}_k$ is an additive noise vector of length n. The permutation $\hat{\mathbf{M}}_k^{-1}$ only reorders the elements of $\mathbf{n}_k$ while $\mathbf{U}$ leads to replicating each element r^2 times. This results in a new $(n \times n)$ covariance matrix with a non–diagonal structure $\tilde{\mathbf{\Sigma}} = \hat{\mathbf{M}}_k^{-1}\mathbf{U}\mathbf{\Sigma}\mathbf{D}\hat{\mathbf{M}}_k$. For simplicity of analysis, we will however assume an independent and identically distributed (i.i.d.) Laplace random vector with $\tilde{\mathbf{\Sigma}} = \sigma^2\mathbf{I}_n$. The error due to this simplification is a blurring effect that should be largely reduced in the deblurring step.

Given the data model (3.22), the two steps of initialization and deblurring are described below.

Step 1: Initialization

The log–likelihood function associated with (3.22) becomes

$$\begin{aligned} \ln p(\overline{\mathbf{g}}_0 \uparrow, \cdots, \overline{\mathbf{g}}_{(N-1)} \uparrow \mid \mathbf{z}) = \\ = \ln \left(\prod_{k=0}^{N-1} \frac{\sqrt{2}}{2\sigma} \exp\left(-\frac{\sqrt{2}\|\overline{\mathbf{g}}_k \uparrow - \mathbf{z}\|_1}{\sigma} \right) \right) \\ = -N \ln \frac{\sigma}{\sqrt{2}} - \frac{\sqrt{2}}{\sigma} \sum_{k=0}^{N-1} \|\mathbf{z} - \overline{\mathbf{g}}_k \uparrow\|_1. \end{aligned} \tag{3.23}$$

Maximizing (3.23) with respect to $\mathbf{z}$, we obtain

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \sum_{k=0}^{N-1} \|\mathbf{z} - \overline{\mathbf{g}}_k \uparrow\|_1, \tag{3.24}$$

which corresponds to the pixel–wise median estimator, i.e., $\hat{\mathbf{z}} = \text{med}_k\{\overline{\mathbf{g}}_k \uparrow\}_{k=0}^{N-1}$. The non–zero initialization in (3.24) relaxes the condition in (3.3), thus solving the problem of undefined pixels. In order not to fall under the same artifacts as those present with interpolation–based SR approaches, see Figure 3.1(c), it is necessary to perform the filling operation from registered and clustered LR images as in (3.2). Indeed, the values from LR frames remain more reliable sources of information than the ones due to upsampling. They are further processed by a $(r \times r)$ median filtering to smooth out noisy depth pixels. We point out that the higher accuracy in the estimation of $\mathtt{w}_k$ shown in Section 3.3 leads to a higher discrimination between motions, and results in a higher diversity S and a better update of the pixel values in $\mathbf{z}$ as compared to the case of classical *S&A*. In our algorithm, it is more accurate to refer to this operation as *initialization update* rather than filling.

Step 2: Deblurring

The deblurring given in (2.7) is slightly modified where the blur $\mathbf{H}$ is replaced by $\mathbf{B}$. Moreover, a diagonal weighting matrix $\mathbf{\Lambda}$ is used to assign a weight to each pixel position. This weight is proportional to the number of measurements used in the update of the corresponding pixels during Step 1. In our work, we adopt the robust bilateral total variation as a regularization term Γ_{BTV} as defined in [7], and solve the following optimization:

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\text{argmin}} \Big(\mathbf{\Lambda} \|\mathbf{B}\mathbf{f} - \hat{\mathbf{z}}\|_1 + \lambda \Gamma_{BTV}(\mathbf{f}) \Big). \tag{3.25}$$

The matrix $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal corresponds to the elements of a weighting matrix $\mathbf{W}$ of the same size as the HR image $\mathbf{f}$. The matrix $\mathbf{W}$ assigns a weight to each pixel in $\mathbf{f}$ such that its contribution in the optimization (3.25) is proportional to the number of measurements used in initializing its value in $\mathbf{z}$ during Step 1.

For an efficient optimization, we propose to further use $\mathbf{W}$ as a mask that, similarly to [79], only selects a subset of pixels to contribute in the optimization. In our framework, this selection actually improves the estimation by only accounting for reliable pixels, those that were updated, and setting to zero the effect of the pixels that were not updated during the initialization of $\mathbf{z}$. We illustrate the effect of this new selective optimization in Section 3.6.1. The proposed SR algorithm for static scenes is an enhanced version of the classical *S&A*, that we will refer to as *eS&A*. It is summarized in Algorithm 3.1.

Algorithm 3.1 ***eS&A*:** Robust super–resolution for static scenes

1. Choose the reference frame $\mathbf{g}_0$.

for k, *s.t.*, $k \in [0, N-1]$,

do

2. Compute $\mathbf{g}_k \uparrow$ using (3.14).
3. Estimate the registration matrices $\hat{\mathbf{M}}_k$ using enhanced PyrME.
4. Compute $\overline{\mathbf{g}}_k \uparrow$ using (3.20).

end do

end for

5. Find $\hat{\mathbf{z}}$ by applying a median estimator (3.24).
6. Update $\hat{\mathbf{z}}$ using (3.2).
7. Deduce $\hat{\mathbf{f}}$ by deblurring with (3.25).

end for

3.6 Experimental results

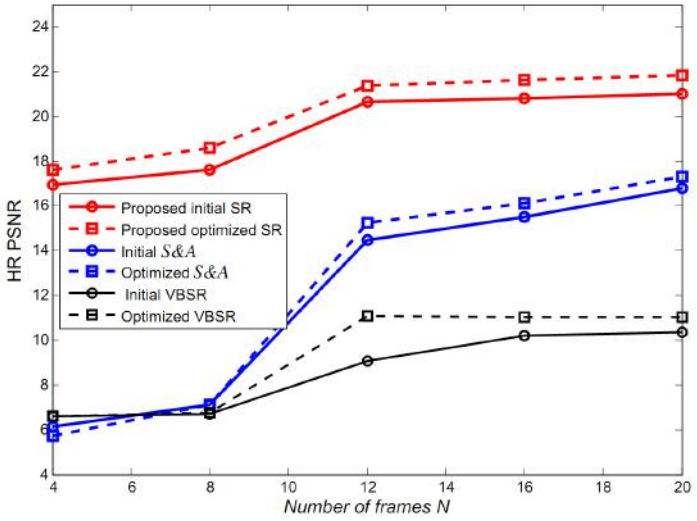
We compare the performance of the proposed *eS&A* algorithm described in Algorithm 3.1 with the two state-of-art methods, that are currently, to the best of our knowledge, the best performing SR algorithms, namely, *S&A* [7], and *VBSR* [26]. We tested these methods using the software provided in [80] and [81]. As these methods have been originally proposed for 2D data, we first evaluate *eS&A* on static 2D scenes then move to static depth scenes.

3.6.1 Static 2D scenes

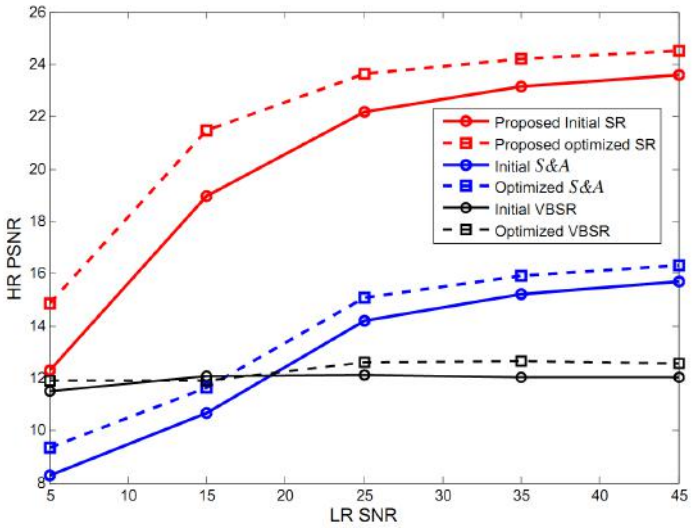
Starting with the HR image "EIA" [82], we generate LR images by downsampling with a factor $r = 4$, and simulating a (3×3) Gaussian PSF with a standard deviation of 0.4, and further degrading by additive white Gaussian noise (AWGN). We evaluate the robustness of the proposed algorithm against two parameters: number of considered LR images N, and image contamination with noise using the signal to noise ratio (SNR). We measure the quality of the estimated HR image using peak signal to noise ratio (PSNR) defined as: $\text{PSNR} = 10\log_{10}\frac{m\times n}{\|\mathbf{f}-\hat{\mathbf{f}}\|_2}$. Figure 3.2(a) shows the average PSNR for 100 different noise realizations, and N progressively increasing from 4 to 20. In order to evaluate for relatively large global motions, translation parameters are generated randomly between 0 and 9 pixels. Note that in [7], smaller motions have been used which explains the difference with the result presented in this work using the same *S&A* algorithm. In the overdetermined case where $N \geq r^2$, and for small motions, both methods *eS&A* and *S&A* give comparable results.

We first provide the results of the non–zero initialization step using the proposed *eS&A* (solid lines). Then we give the final results obtained after applying the selective optimization (dashed lines) whose starting point is the output of the previous step. To avoid any increase in computational cost, upsampled frames can be registered using an approximation by upscaling corresponding LR motion vectors. Note that for a fair comparison, we use the same set of parameters in the optimization step for both the proposed method and the method in [7]. From Figure 3.2(a), it is clear that the proposed method provides significant improvements as compared to existing methods for any choice of N, even as small as 4 images. This observation holds for both the initial estimation and for the iteratively optimized solution. Note that the initial estimate considerably outperforms *VBSR* and *S&A*, initial and optimized solutions.

Figure 3.3 illustrates an example of an HR estimated image using 8, 12, and 20 observed LR images. Due to the condition (3.3), it is not surprising to see the artifacts caused by the undefined pixels, where the number of images is not sufficient to cover the motion range. Moreover, it is clear that the proposed method provides the best visually enhanced HR images as seen in Figure 3.3 (c), (f), and



(a)



(b)

Figure 3.2: PSNR for different SR methods applied on (75×75) LR frames with $r = 4$ (a) for increasing N, (b) for increasing SNR levels.

(i) with sharper edges compared to other methods.

Next, we conduct a second round of experiments to evaluate the performance of the proposed *eS&A* at different noise levels. We use the same 12 frames generated previously and further degrade them by AWGN with SNR of 5, 15, 25, 35, and

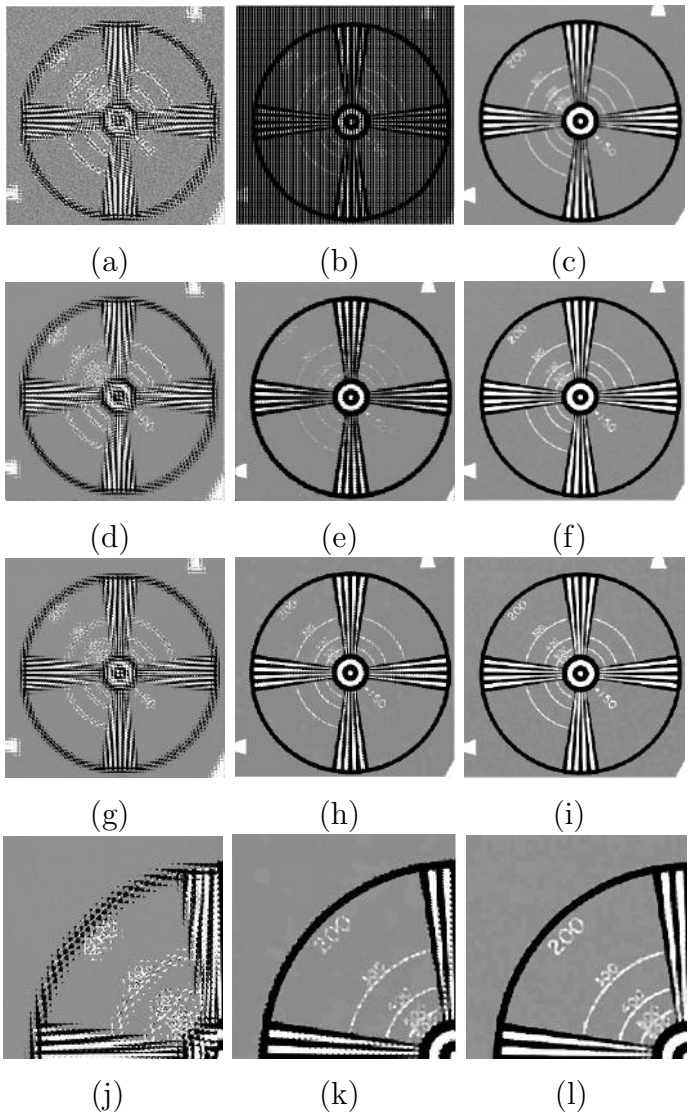
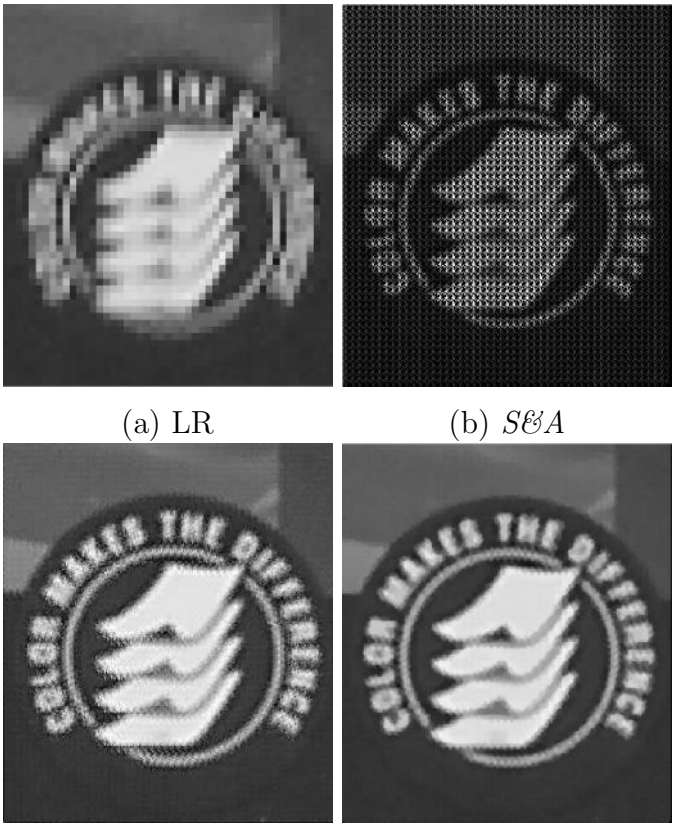


Figure 3.3: Results of different SR methods applied to a (75×75) LR sequence of a static scene with $r = 4$ and different frame numbers. *VBSR* for (a) $N = 8$, (d) $N = 12$, (g) $N = 20$, and by *S&A* for (b) $N = 8$, (e) $N = 12$, (h) $N = 20$ and by proposed *eS&A* for (c) $N = 8$, (f) $N = 12$, (i) $N = 20$.

45 dB. One may note that for a fair comparison we use this number of frames as it guarantees an initial HR image without undefined pixels for all methods (see Figure 3.3(d), (e), and (f)). Mean PSNR values of 100 different noise realizations are plotted in Figure 3.2(b) showing that the proposed method provides the



(a) LR (b) *S&A*

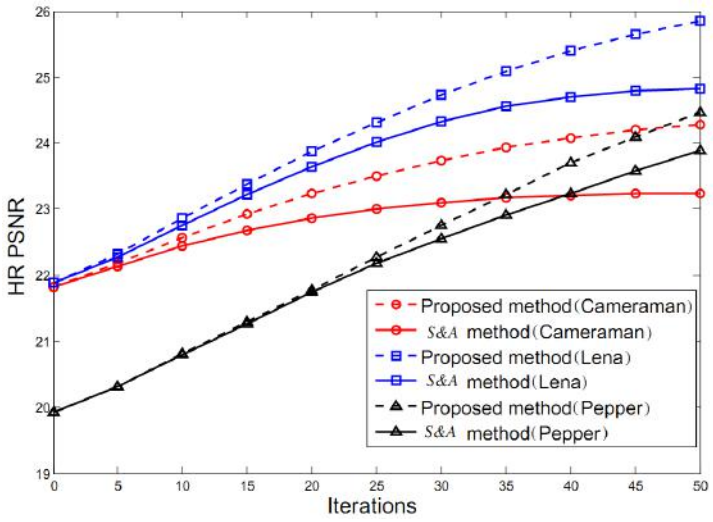
(c) *VBSR* (d) *eS&A*

Figure 3.4: Results of different SR methods with $r = 6$ applied on real data of a static scene, (b) *S&A*, (c) *VBSR*, and (d) proposed method *eS&A*.

best results among discussed SR methods across all noise levels. It is important to note that the optimization of initial estimates ensures a final result that is consistently more robust to noise.

Similar results and conclusions were obtained using real data. We consider a set of 20 LR images of resolution (57×49) pixels of the *disk* dataset [82]. Figure 3.4 presents the SR results of *disk* images for an SR factor $r = 6$. It is visually clear that the proposed method provides better results with sharper edges and less ringing artifacts than other methods in addition to solving the undefined pixels problem.

Finally, in order to illustrate the effect of the proposed selective optimization as compared to the optimization proposed in [7], we ran an experiment on three



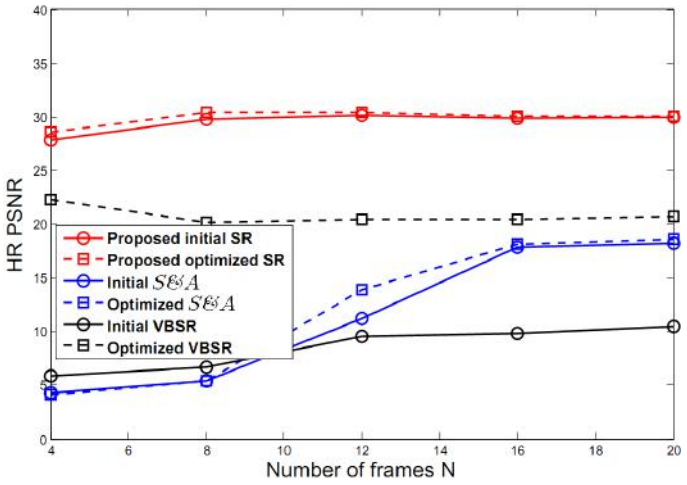
(a)

Figure 3.5: PSNR per iteration of the proposed selective optimization in *eS&A* against *S&A* [7] using 11 (96×96) LR images with SNR = 25dB and SR scale factor $r = 5$.

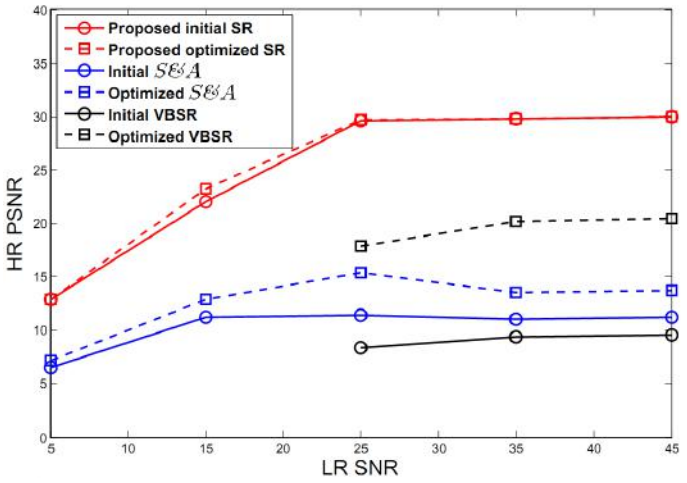
initial HR images by performing 50 iterations. Initial images are obtained by applying *eS&A* with $r = 5$ on three different sequences. Each sequence consists of 11 (96 × 96) LR frames further degraded by AWGN with SNR = 25dB. As shown in Figure 3.5, we may see that the proposed optimization method results in an increase in $PSNR$ as compared with [7]. Moreover, the number of processed pixels decreases and varies from an image to another depending on the number of selected pixels (e.g., 82944 pixels (36 %) and 73728 pixels (32%) processed pixels per iteration for the "Cameraman" and "Lena" images, respectively. In contrast, the objective function in [7] processes all pixels with a minimum weight of value 1 for unreliable pixels (e.g., for a (480×480) image, the number of processed pixels are 230400 pixels per iteration).

3.6.2 Static depth scenes

To evaluate the performance of the proposed *eS&A* on static depth scenes, similarly to Section 3.6.1, we test its robustness on synthetic and real depth images against two parameters: number of considered LR images N, and image con-



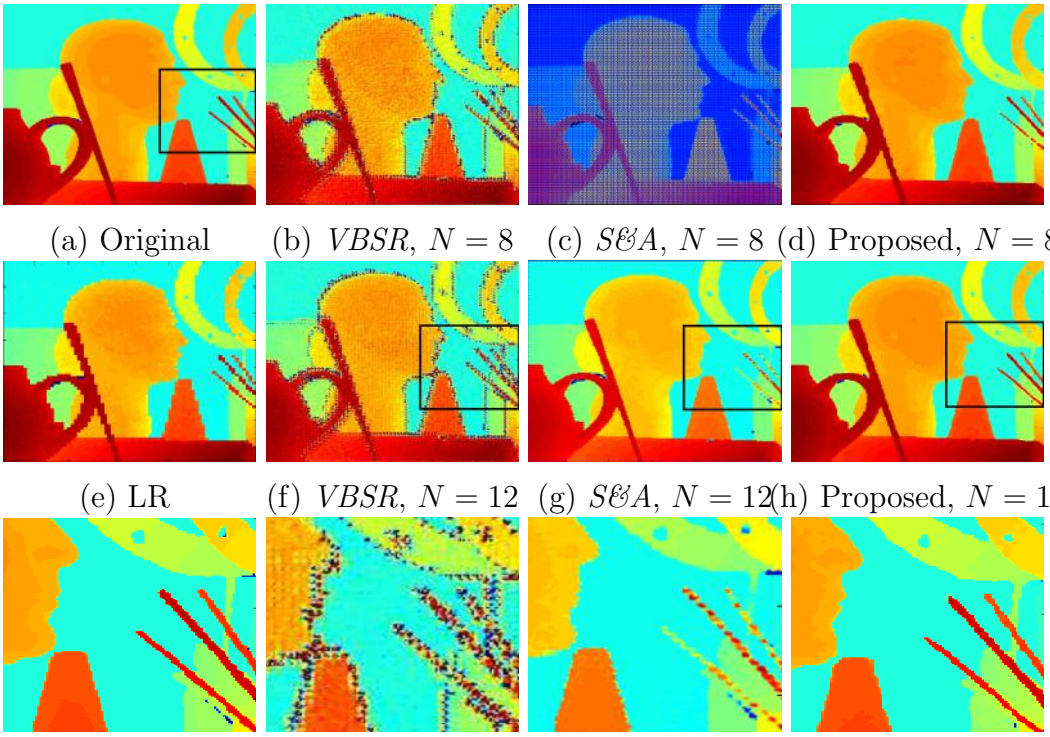
(a) Different N values



(b) Different input SNR levels

Figure 3.6: Mean PSNR values for different SR methods applied to a (75×75) LR sequence of a static depth scene with $r = 4$.

tamination with noise measured by SNR. Each time, we compare the classical *S&A* [7], and *VBSR* [26]. First, we run Monte–Carlo simulations on synthetic sequences of a static scene subjected to a randomly generated global motion. These sequences were created by downsampling the HR image "ART" from the Middlebury dataset [37] with a factor $r = 4$, and PSF using a Gaussian function with a standard deviation of 0.4, and further degrading them by AWGN. For a fixed noise level corresponding to SNR = 45dB, and 100 different realizations,



(a) Original (b) *VBSR*, $N = 8$ (c) *S&A*, $N = 8$ (d) Proposed, $N = 8$

(e) LR (f) *VBSR*, $N = 12$ (g) *S&A*, $N = 12$ (h) Proposed, $N = 12$

(i) (j) (k) (l)

Figure 3.7: Results of different SR methods on a static ToF depth scene with different frame numbers ($N = 8, N = 12$) and SR factor of $r = 4$.)

Figure 3.6(a) shows the average PSNR for N progressively increasing from 4 to 20 frames. It is clear that the proposed method outperforms both *S&A* and *VBSR* across different numbers of LR frames. This difference is even more noticeable for very low values of N, which illustrates the practicality of the proposed *eS&A* method.

Next, we run another round of experiments to evaluate the performance of *eS&A* across different noise levels. A sequence of 12 LR depth images of size (75×75) pixels was used. It was generated in the same way as in the previous experiment, and further degraded by AWGN with SNR of 5, 15, 25, 35 and 45dB. Figure 3.6(b) shows that the proposed method is consistently more robust to noise.

Furthermore, the textureless property of depth images combined with dense upsampling boost the performance of the proposed initial HR frame estimation,

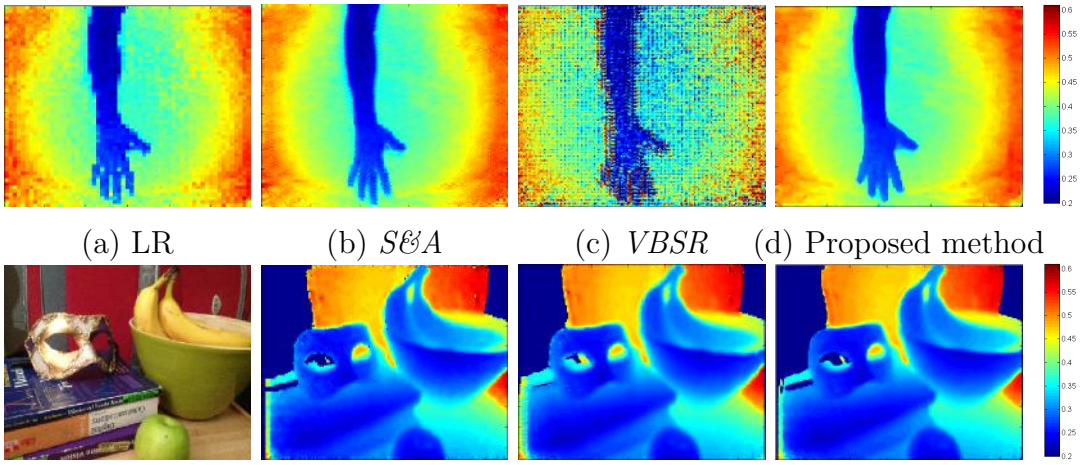
even for a very high noise level, e.g., SNR = 5dB, leading to comparable results before and after optimization with (3.25) as shown, respectively, with the dashed and continuous red lines in Figure 3.6(a) and Figure 3.6(b). This result suggests that the non–zero initialization may be considered as a standalone approach in the case of depth data as it does not deviate much from the assumptions related to the data model in (3.21).

We give, in Figure 3.7, an example of an HR estimated image of "ART" using 8 and 12 LR images in the first and second rows, respectively. Due to the condition (3.3), it is not surprising to see the artifacts caused by undefined pixels where the number of images is not sufficient to cover the motion range. Moreover, as seen in Figure 3.7(d),(h), it is clear that *eS&A* method provides the best visually enhanced HR depth images with sharper edges as compared to the results of *S&A* and *VBSR*.

Finally, we teste the proposed *eS&A* on two real depth sequences which are very short. The first sequence contains 8 LR depth images acquired using an IEE MLI ToF camera of resolution (56×61) pixels [4]. The second sequence contains 5 LR frames acquired using a PMD CamBoard nano of resolution (120×165) pixels [3]. Considering an SR factor of 4, the final results are given in Figure 3.8, clearly showing that for these practical cases with a small number of frames N, the proposed method nicely super–resolves the LR frames by preserving edges and details while *S&A* and *VBSR* fail due to undefined pixels. Note that for the sake of practical deployment, to avoid any additional computational cost in the proposed method, the motion estimation from upsampled LR frames may be approximated by upscaling the corresponding LR motion vectors.

3.7 Conclusion

We presented a new enhanced *S&A* algorithm which improves the quality of the initialization of the HR image in the context of the SR problem. The proposed algorithm is based on upsampling LR images before registering them. We demonstrated that this new approach for SR provides a more accurate motion estimation and registration. Thanks to a dense upsampling, this algorithm is shown to perform well on 2D and on depth data. Experimental results with both synthetic



(a) LR (b) *S&A* (c) *VBSR* (d) Proposed method

(e) 2D Image (f) LR (g) *S&A* (h) Proposed method

Figure 3.8: Results of different SR methods on real LR ToF short sequences.

and real images demonstrate that the proposed algorithm gives results superior to existing state-of-art methods such as classical *S&A* and *VBSR* under various conditions; low number of input LR images, and different noise levels. In addition to being robust, the proposed approach showed that it can be reliably used as an initial guess for SR algorithms. Further optimizing this initialization ensures a strong resilience to noise without additional computational cost. Finally and based on the work proposed in this chapter we have developed a software tool for HR 3D face reconstruction using an LR depth camera, see Appendix B.

Chapter 4

Depth SR for Dynamic Scenes

Chapter 4 presents an SR algorithm for dynamic depth scenes. The proposed solution can handle scenes containing one or more moving objects even non-rigidly without prior assumptions on their shape, and without training. The proposed algorithm referred to as *Upsampling for Precise Super-Resolution* (*UP-SR*) is based on a new data model that uses densely upsampled, and cumulatively registered versions of the observed LR frames. It is these two key components, together, that constitute the working principle of *UP-SR*. While we have described the concept of upsampling in Chapter 3, we herein focus on the concept of cumultative motion estimation.

4.1 Introduction

SR techniques have been largely explored in 2D imaging. We have seen their extension to depth data in Chapter 3. Further extending them to dynamic depth scenes with moving objects is another challenge mainly given the additional depth artifacts caused by motion as summarized in Section 2.2.2. Indeed, we recall that fast motions and surface reflectivity of objects in the scene create invalid pixels and the so–called flying pixels; thus, making most existent 2D SR algorithms fail when directly applied on dynamic depth videos.

Even before talking about dynamic depth data, dynamic scenes in general are challenging scenarios. They require the local motion of moving objects to be computed accurately. They, hence, may face the problem of self–occlusions especially

in the case of non–rigidly moving objects. This difficulty arises in depth videos, but also for 2D sequences [83, 84, 85, 86]. Most of the methods in the literature are limited due to strong assumptions on the shape and number of moving objects. Hardie et al. [83] proposed to restrict the application of SR to a masked area inside a segmented moving object. In [84], Farsiu et al. proposed to combine SR with tracking by using a Kalman filter, and van Eekeren et al. proposed in [85] to use bilinear interpolation. Boundary pixels around the moving object consequently can not be super–resolved, which leads to severe artifacts especially in the case of multiple moving objects in the scene. As a solution, van Eekeren et al. proposed a new algorithm that solves that, and enhances the resolution of boundary pixels up to the final desired HR [86]. This method is, however, computationally heavy and based upon strong assumptions wherein the background is super–resolved first using one of the well–known SR methods for static scenes [7, 26, 72]. Thus, this method is restricted to the SR factor and also to the size of the moving object. Indeed if, for example, the size of the moving object covers more than half of the LR image through the sequence, the method fails. To obtain a complete HR image, a second step of super–resolving the moving foreground is needed. This step assumes a single moving object with polygonal boundaries and a simple linear motion. In addition to these strong assumptions, a segmentation process is applied, followed by tracking to link the most similar object in each frame to a so–called reference object. For this reason, the enhancement of the resolution of dynamic depth scenes has been so far mostly based on fusion with higher resolution 2D data that has to be simultaneously captured [10, 18]; thus, requiring a perfect alignment, synchronization, and mapping of the 2D and depth images, and assuming the correspondence of edges on the two modalities. These methods may be computationally efficient, but unfortunately they frequently suffer from artifacts caused by the heuristic nature of the enforced statistical model, mainly copying the intensity texture of 2D images to depth images.

4.2 Problem formulation

The aim of dynamic SR algorithms is to estimate a sequence of HR images $\{\mathbf{f}_{t_0}\}$ of size $(\sqrt{n} \times \sqrt{n})$ from observed LR sequences. The dynamic SR problem can be

simplified by reconstructing one HR image at a time, $\mathbf{f}_{t_0}$, for $t_0 \in \mathbb{N}$ using an LR sequence $\{\mathbf{g}_t\}_{t_0-N+1}^{t_0}$ of length N, where each LR image $\mathbf{g}_t$ is of size $(\sqrt{m} \times \sqrt{m})$ pixels, with $\sqrt{n} = r \cdot \sqrt{m}$, where r is the SR factor, such that $r \geq 1$. Note that for the sake of simplicity, and without loss of generality, we assume squared images. Every image $\mathbf{g}_t$ may be viewed as an LR noisy and deformed realization of $\mathbf{f}_{t_0}$ at the acquisition time t, with $t \leq t_0$. Rearranging all images in lexicographic order, i.e., column vectors of lengths n for $\mathbf{f}_t$, and m for $\mathbf{g}_t$, we consider the following data model:

$$\mathbf{g}_t = \mathbf{DHM}_{t_0}^t \mathbf{f}_{t_0} + \mathbf{n}_t, \quad t \leq t_0, \tag{4.1}$$

where $\mathbf{D}$ is a matrix of dimension $(m \times n)$ that represents the downsampling operator, and which we assume to be known and constant over time. The system blur is represented by the time and space invariant matrix $\mathbf{H}$. The vector $\mathbf{n}_t$ is an additive Laplacian noise at time t, as justified in [7, 23]. The matrices $\mathbf{M}_{t_0}^t$ are $(n \times n)$ matrices corresponding to the geometric motion between the considered HR image $\mathbf{f}_{t_0}$ and the observed LR image $\mathbf{g}_t$ prior to its downsampling.
Based on the data model in (4.1), and using an L_1 norm between the observations and the model, the Maximum Likelihood (ML) estimate of $\mathbf{f}_{t_0}$ is obtained as follows:

$$\hat{\mathbf{f}}_{t_0} = \arg\min_{\mathbf{f}_{t_0}} \sum_{t=t_0-N+1}^{t_0} \|\mathbf{DHM}_{t_0}^t \mathbf{f}_{t_0} - \mathbf{g}_t\|_1. \tag{4.2}$$

Using the same approach as in [7, 87], we consider that $\mathbf{H}$ and $\mathbf{M}_{t_0}^t$ are block circulant matrices. Therefore: $\mathbf{HM}_{t_0}^t = \mathbf{M}_{t_0}^t \mathbf{H}$. The minimization in (4.2) can then be decomposed into two steps; initialization by estimating the blurred HR image $\mathbf{z}_{t_0} = \mathbf{Hf}_{t_0}$, followed by a deblurring step to recover $\hat{\mathbf{f}}_{t_0}$. In what follows, we assume that $\mathbf{g}_t$ is simply the noisy and decimated version of $\mathbf{z}_t$ without any geometric warp. We may thus write $\mathbf{M}_t^t = \mathbf{I}_n, \forall t$, $\mathbf{I}_n$ being the identity matrix of size $(n \times n)$, hence, $\mathbf{M}_{t_0}^t \mathbf{z}_{t_0} = \mathbf{z}_t = \mathbf{Hf}_t$. This operation can be assimilated to registering $\mathbf{z}_{t_0}$ to $\mathbf{z}_t$. We draw attention to the fact that in the case of static SR, instead of a sequence, a set of observed LR images is considered, i.e., there is no order between frames. Such an order becomes crucial in dynamic SR because the estimation of motion, based on the optical flow paradigm, happens between consecutive frames only. An accurate dynamic SR estimation is consequently

highly dependent on the accuracy of estimating the registration matrices between consecutive frames $\mathbf{M}_t^{t-1}$, as well as the motion between non–consecutive frames $\mathbf{M}_{t_0}^t$ with $t < t_0 - 1$.

In Section 4.3, we present our strategy for a cumulative estimation of the non–consecutive motion matrices $\mathbf{M}_{t_0}^t$, leading to the final proposed *UP-SR* algorithm.

4.3 Novel reduced SR data model

Following the result in Section 3.3, we use the enhanced PyrME and follow an upsampling strategy as a starting point for a new improved SR algorithm. As shown in Section 3.3, upsampling the observed LR images $\mathbf{g}_t$ prior to any operation should lead to a more accurate and robust motion estimation, which enhances the registration of frames. We define the resulting r–times upsampled image as $\mathbf{g}_t \uparrow = \mathbf{U} \cdot \mathbf{g}_t$, where $\mathbf{U}$ is an $(n \times m)$ upsampling matrix. Due to the specific properties of depth data, the upsampling matrix $\mathbf{U}$ has to correspond to a dense upsampling as defined in Section 3.4.

4.3.1 Cumulative motion estimation

Most of optical flow approaches, including the proposed enhanced PyrME, work under the assumption of small motions. Thus, by considering the frames which are far from the reference frame at t_0, high registration errors are introduced as compared to the errors introduced by frames that are closer to t_0. Further frames are therefore considered as outliers. To tackle this problem, we propose a new registration method. This method is based on a cumulative motion estimation where we use the temporal information provided by intermediary frames between the reference frame and the frame under consideration.
Each two consecutive upsampled frames $\mathbf{g}_t \uparrow$ and $\mathbf{g}_{t+1} \uparrow$ in the sequence are related as follows:

$$\mathbf{g}_{t+1} \uparrow = \mathbf{M}_t^{t+1} \mathbf{g}_t \uparrow + \boldsymbol{\delta}_{t+1}, \tag{4.3}$$

where $\boldsymbol{\delta}_{t+1}$ represents the innovation which is assumed to be negligible. We apply the enhanced PyrME strategy described in Section 3.3 to estimate the

local motion $\mathbf{M}_t^{t+1}$ for all the pixel positions $\mathtt{p}$. By so doing we obtain a dense optical flow.

$$\hat{\mathbf{M}}_t^{t+1} = \arg\min_{\mathbf{M}} \Psi\left(\mathbf{g}_{t+1}\uparrow, \mathbf{g}_t\uparrow, \mathbf{M}\right), \tag{4.4}$$

where Ψ is a dense optical flow–related cost function, in the simplest case based on local mean squared errors as in (3.4). The motion from $\mathbf{g}_t\uparrow$ to $\mathbf{g}_{t+1}\uparrow$ is computed in a similar way; thus, leading to the registration of $\mathbf{g}_t\uparrow$ to $\mathbf{g}_{t+1}\uparrow$ as follows:

$$\overline{\mathbf{g}}_t^{t+1}\uparrow = \hat{\mathbf{M}}_t^{t+1}\mathbf{g}_t\uparrow. \tag{4.5}$$

The main target is to define $\overline{\mathbf{g}}_t^{t_0}\uparrow$, which represents the registered version of $\mathbf{g}_t\uparrow$ to the reference $\mathbf{g}_{t_0}\uparrow$ by using all the registered upsampled images $\overline{\mathbf{g}}_t^{t+1}\uparrow$, as defined in (4.5), for $t < t_0$, see Figure 4.1. This approach is similar to the concept proposed in [88], with an additional improvement where we further reduce the cumulated motion error by recomputing $\hat{\mathbf{M}}_t^{t+1}$ using the already registered frame $\overline{\mathbf{g}}_{t-1}^{t}\uparrow$ as follows:

$$\hat{\mathbf{M}}_t^{t+1} = \arg\min_{\mathbf{M}} \Psi\left(\mathbf{g}_{t+1}\uparrow, \overline{\mathbf{g}}_{t-1}^{t}\uparrow, \mathbf{M}\right). \tag{4.6}$$

We prove by induction (see Appendix A) the following registration equation for non–consecutive frames:

$$\overline{\mathbf{g}}_t^{t_0}\uparrow = \hat{\mathbf{M}}_t^{t_0}\mathbf{g}_t\uparrow = \underbrace{\hat{\mathbf{M}}_{t_0-1}^{t_0}\cdots\hat{\mathbf{M}}_t^{t+1}}_{(t_0-t)\text{ times}}\cdot\mathbf{g}_t\uparrow, \tag{4.7}$$

where

$$\hat{\mathbf{M}}_t^{t_0} = \hat{\mathbf{M}}_{t_0-1}^{t_0}\cdots\hat{\mathbf{M}}_t^{t+1}. \tag{4.8}$$

Note that due to the high noise level in depth raw data, we apply a preprocessing step with a bilateral filter before motion estimation. The bilateral filter is only used in the preprocessing step while the original depth data is mapped in the registration step and further used in the fusion process.

4.3.2 Proposed UP-SR algorithm

The classical data model for a dynamic scene is given in (4.1). The additive noise $\mathbf{n}_t$ follows a white multivariate Laplace distribution as it has been shown to

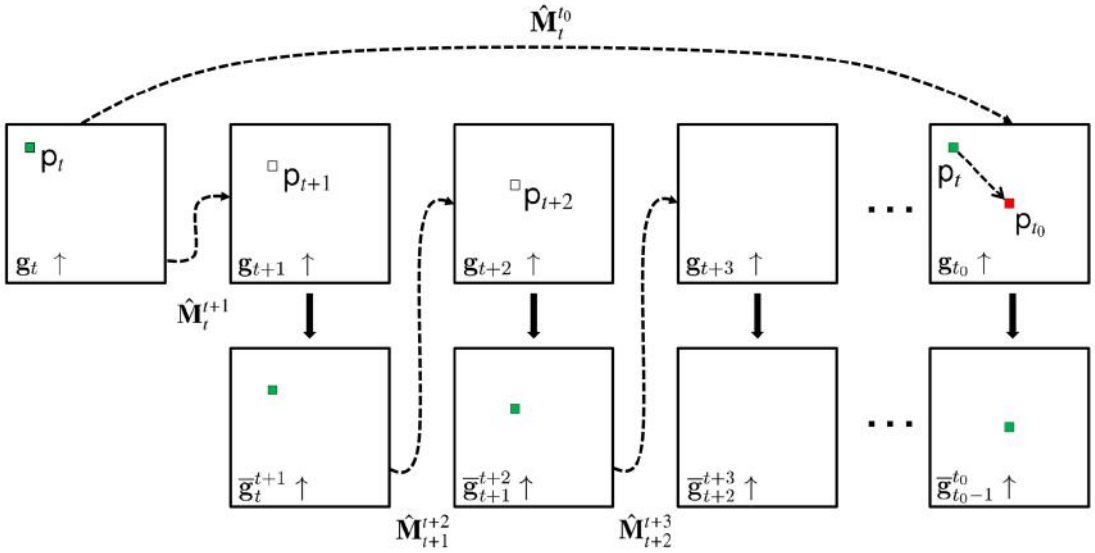


Figure 4.1: *UP-SR* Cumulative Motion Estimation: All intermediate registered upsampled depth frames are used to register the pixel $\mathbf{p}_t$ in frame $\mathbf{g}_t \uparrow$ to its corresponding pixel at the position $\mathbf{p}_{t_0}$ from the reference frame $\mathbf{g}_{t_0} \uparrow$ where $\mathbf{g}_t \uparrow$ and $\mathbf{g}_{t_0} \uparrow$ are non–consecutive upsampled frames.

better fit the SR problem as compared to a Gaussian noise model [7, 23]. This distribution is defined as follows:

$$p(\mathbf{n}_t) = \prod_{i=1}^{m} \frac{\sqrt{2}}{2\sigma} \exp\left(-\frac{\sqrt{2}|\mathbf{n}_t(i)|}{\sigma}\right), \tag{4.9}$$

where $\frac{\sigma}{\sqrt{2}}$ is a positive Laplace scale factor leading to the diagonal covariance matrix $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_m$, with $\mathbf{I}_m$ being the identity matrix of size $(m \times m)$.
Considering the reference frame $\mathbf{f}_{t_0}$, and by left multiplying (4.1) by $\mathbf{U}$, we find:

$$\mathbf{g}_t \uparrow = \mathbf{M}_{t_0}^t \mathbf{B} \mathbf{f}_{t_0} + \mathbf{U}\mathbf{n}_t, \qquad t < t_0. \tag{4.10}$$

In addition, similarly to [78], for analytical convenience, we assume that all pixels in $\mathbf{g}_t \uparrow$ originate from pixels in $\mathbf{f}_{t_0}$ in a one to one mapping. Therefore, each row in $\mathbf{M}_{t_0}^t$ contains 1 for each position corresponding to the address of the source pixel in $\mathbf{f}_{t_0}$. This bijective property implies that the matrix $\mathbf{M}_{t_0}^t$ is an invertible permutation, $[\hat{\mathbf{M}}_{t_0}^t]^{-1} = \hat{\mathbf{M}}_t^{t_0}$. Following the result in Section 3.3, and using the cumulative motion proposed in Section 4.3.1, the motion matrix $\hat{\mathbf{M}}_{t_0}^t$ is obtained

from upsampled LR frames $\mathbf{g}_t \uparrow$, $t = t_0 - N + 1, \cdots, t_0$, as in (4.8). Thus, the corresponding registrations to the reference $\mathbf{g}_{t_0} \uparrow$ are performed as

$$\mathbf{g}_t \uparrow = \hat{\mathbf{M}}_{t_0}^{t} \overline{\mathbf{g}}_t^{t_0} \uparrow . \tag{4.11}$$

Given (4.11), and by left multiplying (4.10) by $[\hat{\mathbf{M}}_{t_0}^{t}]^{-1}$, we find

$$\overline{\mathbf{g}}_t^{t_0} \uparrow = \mathbf{B}\mathbf{f}_{t_0} + \boldsymbol{\nu}_t, \qquad t < t_0. \tag{4.12}$$

This finally leads to a new simplified SR data model which is analogous to a classical image denoising problem using multiple observations, specifically

$$\overline{\mathbf{g}}_t^{t_0} \uparrow = \mathbf{z}_{t_0} + \boldsymbol{\nu}_t, \qquad t < t_0, \tag{4.13}$$

where $\boldsymbol{\nu}_t = \hat{\mathbf{M}}_t^{t_0}\mathbf{U} \cdot \mathbf{n}_t$ is an additive Laplacian noise vector of length n with mean zero and covariance $\tilde{\boldsymbol{\Sigma}} = \hat{\mathbf{M}}_t^{t_0}\mathbf{U}\boldsymbol{\Sigma}\mathbf{D}\hat{\mathbf{M}}_{t_0}^{t}$.
Given the data model in (4.13), the two steps of initialization and deblurring are described below.

Step 1: Initialization

The log–likelihood function associated with (4.13) becomes

$$\begin{aligned} \ln p(\overline{\mathbf{g}}_{t_0-N+1}^{t_0} \uparrow, \cdots, \overline{\mathbf{g}}_{t_0}^{t_0} \uparrow \mid \mathbf{z}_{t_0}) = \\ &= \ln \left(\prod_{t=t_0-N+1}^{t_0} \frac{\sqrt{2}}{2\sigma} \exp\left(- \frac{\sqrt{2}\|\overline{\mathbf{g}}_t^{t_0} \uparrow - \mathbf{z}_{t_0}\|_1}{\sigma} \right) \right) \\ &= -N \ln \frac{\sigma}{\sqrt{2}} - \frac{\sqrt{2}}{\sigma} \sum_{t=t_0-N+1}^{t_0} \|\mathbf{z}_{t_0} - \overline{\mathbf{g}}_t^{t_0} \uparrow\|_1. \end{aligned} \tag{4.14}$$

Maximizing (4.14) with respect to $\mathbf{z}_{t_0}$, we obtain

$$\hat{\mathbf{z}}_{t_0} = \arg\min_{\mathbf{z}_{t_0}} \sum_{t=t_0-N+1}^{t_0} \|\mathbf{z}_{t_0} - \overline{\mathbf{g}}_t^{t_0} \uparrow\|_1 \Rightarrow \hat{\mathbf{z}}_{t_0} = \mathrm{med}_t\{\overline{\mathbf{g}}_t^{t_0} \uparrow\}_{t=t_0-N+1}^{t_0}. \tag{4.15}$$

In fact, the equation in (4.15) represents a temporal pixel–wise median filter med_t, which constitutes the fusion step in the *UP-SR* algorithm. Taking the median filter as a temporal filter solves the problem of invalid pixels caused by depth sensors [59], and guarantees that no flying pixels are generated, such erroneous pixels are caused, in classical SR methods [25, 55], by averaging background and foreground pixels.

Step 2: Deblurring

In this work, we adopt Maximum A Posteriori (MAP) estimation using the robust bilateral total variation (BTV) as a regularization term as defined in [7]. This choice is motivated by the fact that the properties of a bilateral filter, namely, noise reduction while preserving edges, is now established as an appropriate method for depth data processing [9, 27, 38]. The BTV regularization is defined as follows:

$$\Gamma_{BTV}(\mathbf{f}_{t_0}) = \sum_{i=-l}^{i=l} \sum_{j=-l}^{j=l} \alpha^{|i|+|j|} \parallel \mathbf{f}_{t_0} - \mathbf{S}_x^i \mathbf{S}_y^j \mathbf{f}_{t_0} \parallel_1 . \tag{4.16}$$

The matrices $\mathbf{S}_x^i$ and $\mathbf{S}_y^j$ are shifting matrices that shift $\mathbf{f}_{t_0}$ by i, and j pixels in the horizontal and vertical directions, respectively. The scalar $\alpha \in]0, 1]$ is the base of the exponential kernel which controls the speed of decay [48].
The final solution is:

$$\hat{\mathbf{f}}_{t_0} = \underset{\mathbf{f}_{t_0}}{\operatorname{argmin}} \Big(\|\mathbf{B}\mathbf{f}_{t_0} - \mathbf{z}_{t_0}\|_1 + \lambda \Gamma_{BTV}(\mathbf{f}_{t_0}) \Big), \tag{4.17}$$

where λ is the regularization parameter. The *UP-SR* algorithm is summarized in Algorithm 4.1.

Because of the complexity of dynamic scenes with moving objects, the choice of the order of the reference frame $\mathbf{g}_{t_0}$ with respect to the frames used to super–resolve it plays a major role. Since we use a temporal median filter in fusing the registered depth frames, taking $\mathbf{g}_{t_0}$ to be in the middle is a natural choice to estimate the corresponding HR depth image $\mathbf{f}_{t_0}$.

4.4 Statistical performance analysis

In this section we derive the performance of the *UP-SR* algorithm in terms of mean square error (MSE) for a fixed noise level. This derivation helps in better understanding the effect of the number of frames N and the effect of the SR factor r on the performance of the *UP-SR* algorithm. In [89, 90], there have been some attempts to derive the asymptotic limits of SR. However, these attempts do not take into account the bias of an SR estimator, which is always part of an

Algorithm 4.1 ***UP-SR*:** Upsampling for Precise Super-Resolution

for t_0,

1. Choose the reference frame $\mathbf{g}_{t_0}$.

for t, *s.t.*, $t_0 - N + 1 \leq t \leq t_0$,

do

2. Compute $\mathbf{g}_t \uparrow$ using (3.14).
3. Estimate the registration matrices $\hat{\mathbf{M}}_t^{t_0}$ using (4.8).
4. Compute $\overline{\mathbf{g}}_t^{t_0} \uparrow$ using (4.7).

end do

end for

5. Find $\hat{\mathbf{z}}_{t_0}$ by applying a temporal median estimator (4.15).
6. Estimate $\hat{\mathbf{f}}_{t_0}$ by deblurring using (4.17).

end for

image reconstruction process [91]. Moreover, a Gaussian noise model is usually assumed while *UP-SR* exploits an additive Laplacian noise model [23]. Taking into account the considered problem, we propose to adapt the affine bias model of [92] based on a representation with patches, which leads to an approximation of the *UP-SR* bias. This bias is related to two main factors, namely, the error due to gradient–based motion estimation [91], and to the SR factor r. Few assumptions are introduced for simplicity of analysis but we will show that they hold in the experimental evaluation, both quantitatively and qualitatively.

Thanks to the new data model proposed in (4.13), we look into the performance of the median estimator $\hat{\mathbf{z}}_{t_0}$ as defined in (4.15) in terms of MSE. Let us define $\mathrm{tr}(\cdot)$ and $\mathrm{cov}(\cdot)$ to be the trace and the covariance functions, respectively. Then, the MSE may be decomposed into two parts; the $\mathrm{bias}(\cdot)$, and the variance $\mathrm{var}(\cdot)$, defined for a given vector $\mathbf{z}$ as $\mathrm{var}(\mathbf{z}) = \mathrm{tr}\left(\mathrm{cov}(\mathbf{z})\right)$. By considering a known ground truth $\mathbf{f}_{t_0}$, we may then express the MSE as follows:

$$\mathrm{MSE}\left(\hat{\mathbf{z}}_{t_0}, \mathbf{f}_{t_0}\right) = \mathrm{var}(\hat{\mathbf{z}}_{t_0}) + \|\mathrm{bias}\left(\hat{\mathbf{z}}_{t_0}\right)\|^2. \tag{4.18}$$

4.4.1 Bias computation

Chatterjee and Milanfar have proposed in [92] an affine bias model for image denoising. The processing is done on patches, thus making the model in [92] local. We have shown in Section 4.3 how the SR problem can be formulated as a denoising problem (4.13). We may therefore apply the model in [92] after some modifications to fit the estimation in (4.15).
We decompose the ground truth image $\mathbf{f}_{t_0}$ into n patches $\{\mathbf{q}_{t_0}(i), i = 1, \cdots, n\}$ where each patch $\mathbf{q}_{t_0}(i)$ is of size $(r \times r)$ pixels and centered at the pixel $\mathbf{f}_{t_0}(i)$. Similarly, the frames $\overline{\mathbf{g}}_t^{t_0} \uparrow$ are decomposed into n overlapping patches $\{\mathbf{p}_t(i), i = 1, \cdots, n\}$. In fact, the estimation in (4.15) corresponds to the process of locally selecting the element with the highest ranking among the N patches at the same position $\{\mathbf{p}_t(i), t = t_0 - N + 1, \cdots, t_0\}$. Let $\mathbb{E}(\cdot)$ be the expectation operator, and $\mathbf{I}_r$ the identity matrix of size $(r \times r)$. By considering two frames at different times t and t', we may calculate the local bias per patch as explained in [41] as follows:

$$\text{bias}\left(\hat{\mathbf{q}}_{t_0}(i)\right) = \mathbf{S}_i \mathbf{q}_{t_0}(i) + \mathbf{u}_i, \tag{4.19}$$

with

$$\mathbf{S}_i = \left(\mathbb{E}\left(\mathbf{W}_{t_0}^{t'}(i)\right) - \mathbf{I}_r\right) \mathbf{q}_{t_0}(i),$$

and

$$\mathbf{u}_i = \mathbb{E}\left(\mathbf{W}_{t_0}^{t'}(i)\boldsymbol{\eta}_{t_0}(i) + \mathbf{w}_{t_0}^{t'}(i)\right),$$

where $\mathbf{W}_{t_0}^{t'}(i)$ and $\mathbf{w}_{t_0}^{t'}(i)$ are the sub–block of $\hat{\mathbf{M}}_{t_0}^{t'}$ centered at position i, and the local innovation directly related to cumulated innovations defined in (4.3), respectively. The vector $\boldsymbol{\eta}_{t_0}(i)$ represents the patch measurement error due to noise and to blur. The final bias is then defined as:

$$\|\text{bias}\left(\hat{\mathbf{z}}_{t_0}\right)\|^2 = \sum_{i=1}^{n} \|\text{bias}\left(\hat{\mathbf{q}}_{t_0}(i)\right)\|^2. \tag{4.20}$$

In the simple case where the average motion per patch and its innovation $\mathbf{w}_{t_0}^{t'}(i)$ are close or equal to zero, the per–patch bias term becomes $\mathbb{E}\left(\boldsymbol{\eta}_t(i)\right)$. This bias is in fact due to the effects of the per–patch blur and to noise. The statistical

properties of the noise are the same as those of $\boldsymbol{\nu}_t$. The blur effect is due to the (r^2-1) pixels per patch generated by the upsampling step. Assuming that they induce a fixed mean error ρ, the total bias may be simplified as follows:

$$\|\text{bias}\left(\hat{\mathbf{z}}_{t_0}\right)\|^2 = \sum_{i=1}^{n} \|\mathbb{E}\left(\boldsymbol{\eta}_t(i)\right)\|^2 = n \cdot (r^2-1)\rho^2. \tag{4.21}$$

We can see in (4.21) that, for $r = 1$, the estimation becomes unbiased. This is due to the fact that there is no blur caused by the upsampling process. Generally, the bias term is data dependent because of $\mathbf{q}_{t_0}(i)$ in (4.19). It also depends on the SR factor r, and the local motions and noise. From (4.21), we conclude that the bias is proportional to the squared SR factor r^2 and to the image size n.

4.4.2 Variance computation

Assuming that the noise $\boldsymbol{\nu}_t$ follows an i.i.d. n–multivariate Laplace distribution, we may write: $\text{var}(\hat{\mathbf{z}}_{t_0}) = \text{tr}\left(\text{cov}(\hat{\mathbf{z}}_{t_0})\right) = n \cdot \text{var}\left(\hat{\mathbf{z}}_{t_0}(i)\right)$, $i = 1, \cdots, n$. Therefore, we may define the variance as [93]

$$\text{var}\left(\hat{\mathbf{z}}_{t_0}(i)\right) = 2\sigma^2 f(N), \quad i = 1, \cdots, n, \tag{4.22}$$

where for N even,

$$f(N) = \frac{4N!}{\left(\left(\frac{N-1}{2}\right)!\right)^2} \left(\frac{1}{2}\right)^{\frac{N+1}{2}} \sum_{k=0}^{\frac{N-1}{2}} \frac{\binom{\frac{N-1}{2}}{k}\left(-\frac{1}{2}\right)^k}{(N+1+2k)^3}, \tag{4.23}$$

and for N odd,

$$f(N) = \frac{N!}{\left(\frac{N}{2}\right)!\left(\frac{N}{2}-1\right)!}\left(\frac{1}{2}\right)^{\frac{N}{2}}\left(\frac{1}{N^3}\left(\frac{1}{2}\right)^{\frac{N}{2}} + \sum_{k=0}^{\frac{N}{2}-1}\binom{\frac{N-1}{2}}{k}\left(-\frac{1}{2}\right)^k \frac{7N^2+8N(k+1)+4(k+1)^2}{N^2(N+2k+2)^3}\right). \tag{4.24}$$

Our model assumes that the effect of overlapping patches is expressed in the bias term. Thus, the variance is independent of r, which corresponds to the simple denoising operation where no SR is involved and $r = 1$. It is proportional to the noise variance σ^2 and to the number of measurements N. The Cramèr Rao bound corresponding to the variance in (4.22) is equal to $\frac{\sigma^2}{2N}$. Thus, for a very long sequence, where N tends to ∞, the variance $\text{var}(\hat{\mathbf{z}}_{t_0})$ tends to 0.

4.5 Experimental results

In order to evaluate the performance of the *UP-SR* algorithm, we start by separately looking at the impact of the two key components, upsampling and cumulative motion estimation, designed to handle the motion of freely moving and deforming objects in depth LR videos. Then, we provide a quantitative evaluation comparing with state-of-art approaches by testing on synthetic data with ground truth. We give qualitative examples using the same synthetic data in addition to real data acquired in a laboratory environment. Finally, for different SR factors and varying noise levels, we compare the obtained results to the theoretical analysis given in Section 4.4.

4.5.1 Upsampling and motion estimation

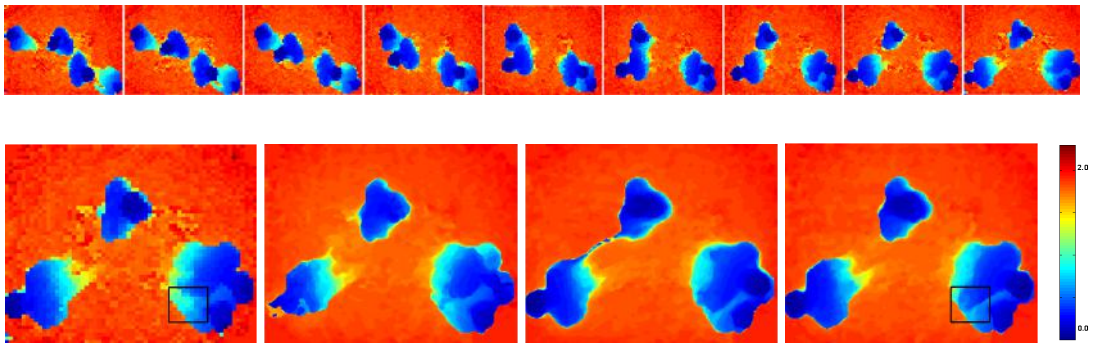
To demonstrate the effect of the upsampling step on the motion estimation process, we conduct the following experiment. We consider the "Art" depth image from the Middlebury dataset [37]. We shift it with one pixel in both x and y directions at the resolution $r = 1$. As a result, the corresponding motion vector at a given scale $r = R$ is $\mathsf{w}^{L\uparrow R} = (R, R)$ pixels, which represents the ground truth motion. In this experiment, we take $R = 8$. Next, we estimate motion vectors for different SR factors, i.e., r varying from 1 to R. These vectors are further upscaled with the factor $\frac{R}{r}$ in order to be compared with the motion ground truth $\mathsf{w}^{L\uparrow R}$. The error of the estimated motion is calculated as follows: $\epsilon_r = \|\frac{R}{r} \cdot \mathsf{w}^{L\uparrow r} - \mathsf{w}^{L\uparrow R}\|_2$. The obtained results are shown in Table 4.1. They clearly support our claim where the error decreases by a factor of $\frac{1}{r}$ by increasing the SR factor r. We can see that estimating motion from upsampled images with the factor $r = R$ is more accurate than upscaling the estimated motion from the lowest level with $r = 1$.

4.5.2 Cumulative registration

To illustrate the effectiveness of the cumulative registration proposed in Section 4.3.1, we consider a challenging case of four persons moving with a large motion in different directions. The used setup is an LR ToF camera, the 3D

	r=1	r=2	r=4	r=6	r=8
ϵ_r (pixels)	0.51	0.25	0.13	0.08	0.06
Gain in accuracy (%)	0%	50%	75%	84%	88%

Table 4.1: Errors ϵ_r between estimated motions upscaled with a factor of $(\frac{R}{r})$ with $r = 1, ..., R$, and estimated motions from upsampled frames with a resolution factor $R = 8$.



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

Figure 4.2: *UP-SR* results with $r = 4$ using different registration techniques of a dynamic scene with four persons moving in different directions. The sequence consists of 9 LR (56×61) depth images. (a) Last frame in the LR sequence. (b) *UP-SR* without cumulative motion. (c) *UP-SR* with cumulative motion upscaled from LR frames. (d) *UP-SR* with the proposed cumulative motion from upsampled frames. The largest measured depth in this scene is 2.5 m.

MLI [4], mounted in the ceiling and looking at the scene from the top. One of the LR frames is shown in Figure 4.2 (a). We apply the *UP-SR* algorithm on this sequence using three different registration techniques, namely, non–cumulative registration, cumulative registration using the upscaled motion vectors estimated from LR frames, and the proposed cumulative registration using the estimated motion from upsampled LR frames. The corresponding results are shown in Figure 4.2 (b), (c), and (d), respectively. They show the superiority of the third technique over the first two techniques, which confirms the advantage of using the proposed cumultative motion estimation. We note, nevertheless, interesting limitations in the case, for example, of intersecting or touching objects, as can be seen within the bounding boxes in Figure 4.2 (d) and Figure 4.5 (d). This is

due to the textureless nature of depth images which may cause two objects to be allocated to the same depth value, and hence makes them wrongly appear as one object.

4.5.3 Qualitative comparison

We use the "Samba" dataset available in [8], which provides a real sequence of a 3D dynamic scene with HR ground truth, Figure 4.3 (e). We downsample a sub–sequence of 9 LR frames with a scale factor $r = 4$. The obtained LR sequence is of resolution (256×147) pixels. This sequence is degraded with additive Laplacian noise with σ varying from 0 to 100 mm. The created LR noisy depth sequence is then super–resolved. In order to visually evaluate the performance of *UP-SR*, we plot in 3D the super–resolved results of the "Samba"–generated sequence for the noise level of $\sigma = 30$ mm. As expected, the *UP-SR* algorithm provides a better result by keeping the fine details as compared to the bicubic interpolation and to the patch–based SISR methods. By zooming on the face part and plotting the 3D error map, it is clear that *UP-SR* gives the closest result as compared to the ground truth, see Figure 4.3 for more details.

Using the same setup of the LR ToF camera mounted in the ceiling at a 2.5m height, we captured an LR depth video of two persons sitting on chairs sliding in two different directions. A sequence of 9 LR depth images, of size (56×61) pixels, was super–resolved with an SR factor $r = 5$ using bicubic interpolation, 2D/depth fusion [10], dynamic S&A [11], patch–based SISR [9], and the proposed *UP-SR*. Visual results for one frame are given in Figure 4.4 (b), (c), (d), (e), and (f), respectively. Obtained results show that bicubic interpolation and dynamic S&A fail on depth data mainly on boundary pixels, while the result of the 2D/depth fusion suffers from strong 2D texture copying on the final super–resolved depth frame as shown in Figure 4.4 (c). We can see the results of SISR in Figure 4.4 (e), where the inaccuracies are also observed especially on objects' boundaries. We show in Figure 4.4 (f) the result of the *UP-SR* algorithm where we obtained clear sharp edges in addition to an efficient removal of noisy pixel values. This is mostly due to the proposed sub–pixel motion estimation combined with an accurate cumulative registration leading to a successful temporal fusion of the sequence.

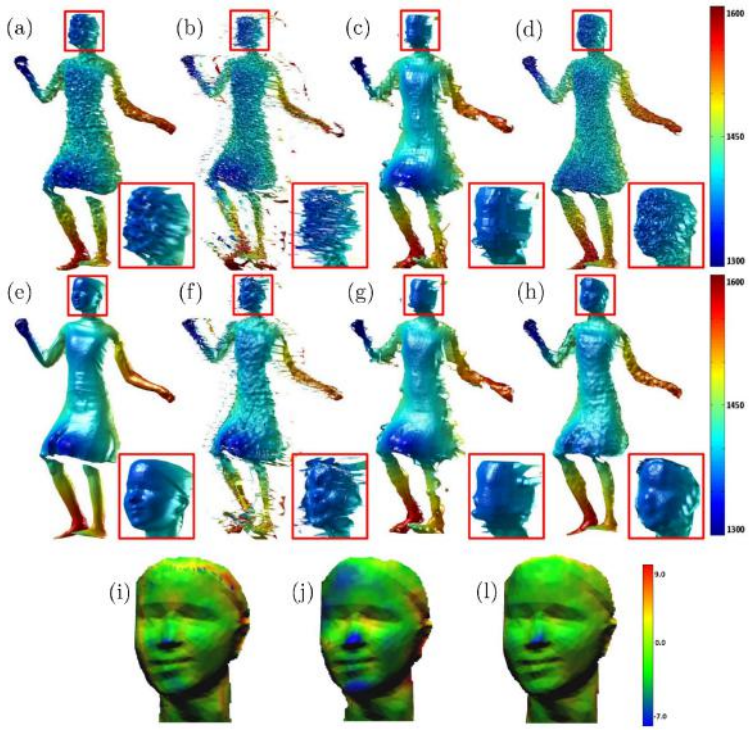


Figure 4.3: 3D results of different SR methods applied on the "Samba" sequence [8]. (a) LR noisy input. (b) Bicubic interpolation. (c) Patch–based SISR [9]. (d) *UP-SR*, initial estimate. (e) Ground truth. (f) Deblurred bicubic. (g) Deblurred patch–based SISR. (h) Deblurred *UP-SR*. Third row represents the 3D error maps for: (i) Bicubic. (j) Patch–based SISR. (l) Proposed *UP-SR*. We can see that the obtained error using the the proposed *UP-SR* (l) is quite small as compared to other methods where the bicubic interpolation leads to noisy depth measurements in addition to the flying pixels represented by the yellow and orange collors in the 3D error map in (i). The obtained results using the patch–based SISR is quite smooth and lead to removing fine details, and hence, resulting in large 3D reconstruction errors, see blue patches in (j). The depth is measured in mm.

Similar results are observed in Figure 4.5 by testing the different methods on the challenging case of the sequence of four moving persons.

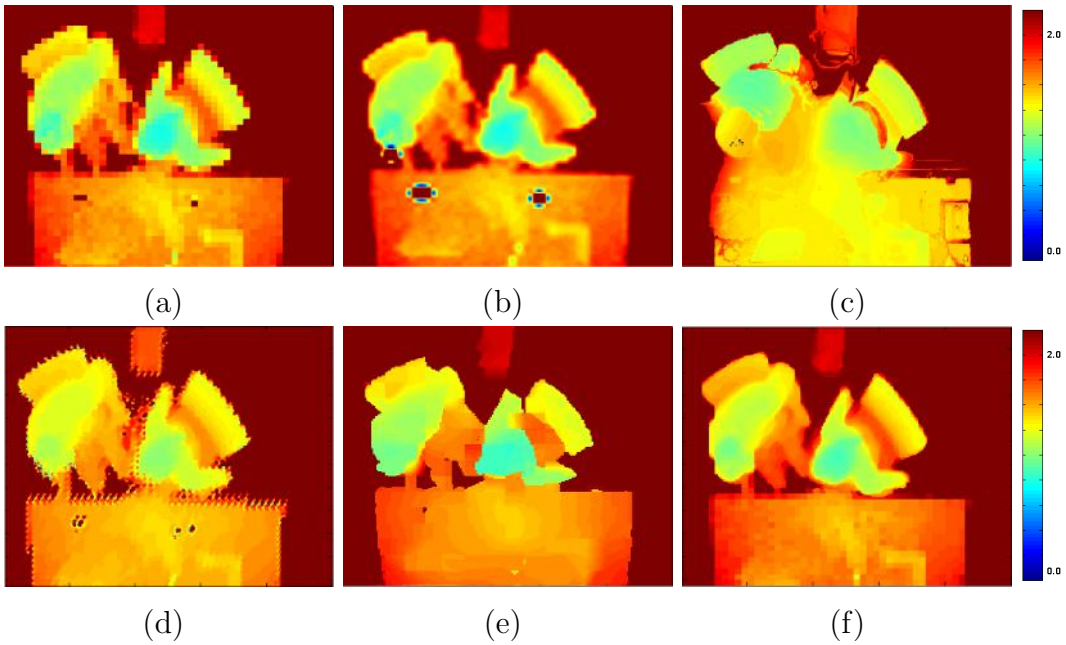


Figure 4.4: Moving chairs sequence: comparison of the results for different SR methods with SR factor of $r = 5$: (a) Last frame of 9 LR (56×61) depth images. (b) Bicubic interpolation of the last depth frame in the sequence. (c) 2D/depth fusion [10]. (d) Dynamic S&A [11]. (e) SISR S&A [9]. (f) Proposed *UP-SR*.

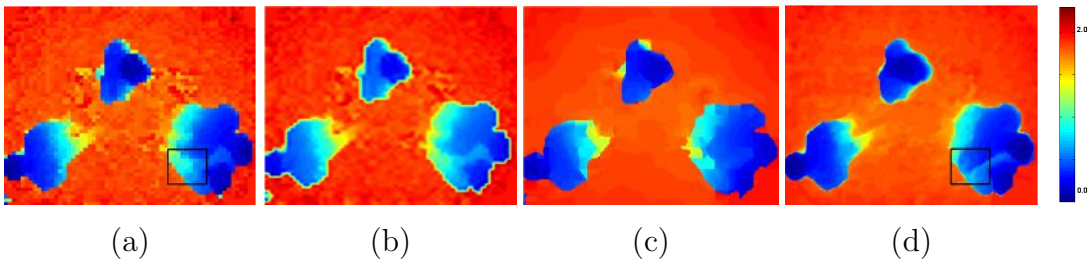


Figure 4.5: Comparison of the results for different SR methods with SR factor of $r = 4$. These methods are applied on a dynamic sequence of four persons with fast motion in different directions. (a) Last frame of LR (56×61) depth images. (b) Bicubic interpolation of the last depth frame in the sequence. (c) SISR [9]. (d) Proposed *UP-SR*.

4.5.4 Quantitative comparison

We provide a quantitative evaluation of the proposed *UP-SR* algorithm as compared to two methods, namely, the conventional bicubic interpolation and the

patch–based single image SR (SISR) given in [9]. We start with the "Samba" dataset, where the previously created LR noisy depth sequences are super–resolved using these methods and the proposed method. We compare the obtained results at two levels, initial and deblurred using the deblurring step proposed in Section 4.3. For the deblurring step we use an exhaustive search to find the best optimization parameters corresponding to the smallest 3-D reconstruction error. The quantitative results are reported in Figure 4.6. As expected, by applying the conventional bicubic interpolation method directly on depth images, a large error in the reconstructed HR depth image is obtained. This error is mainly due to flying pixels around object's boundaries, Figure 4.3 (b). Thus, for a fair

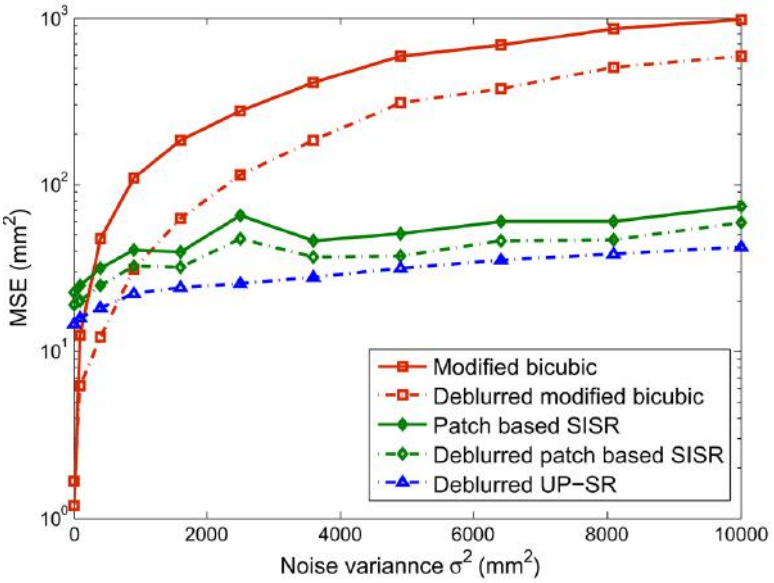


Figure 4.6: MSE at different noise levels for different SR methods applied to an LR depth sequence created from the "Samba" dynamic data [8], with $r = 4$ and $N = 9$.

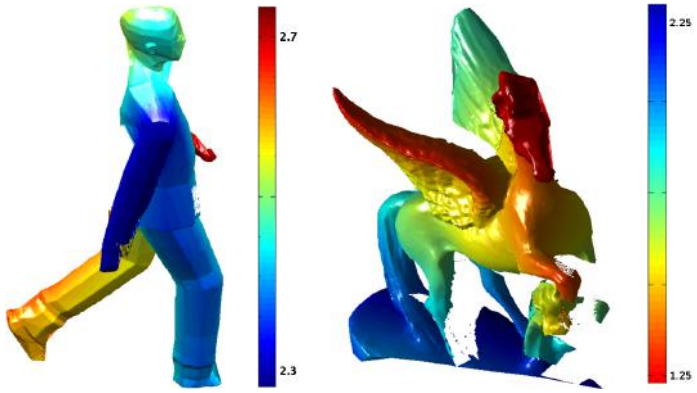
comparison we run another round of experiments using a modified bicubic interpolation, where we remove all flying pixels by defining a fixed threshold. Yet, the 3D reconstruction error remains relatively high. This is due to the fact that bicubic interpolation does not profit from the temporal information provided by the sequence. Only in the case of one moving object and a very low noise level (less than 10 mm) the modified bicubic interpolation may be considered as shown

by the red solid line in Figure 4.6. The performances of SISR, original and deblurred, are given in green lines, solid, and dashed, respectively. SISR can be seen to be robust to noise as its performance is stable even for high noise levels. The addition of the deblurring step of *UP-SR* improves the MSE of the original SISR algorithm. The result of the proposed *UP-SR* algorithm is shown with a blue dashed line. Its MSE is the lowest among all the tested methods, and is also shown to be robust across all noise levels. This result can be explained by the fact that SISR is a patch–based method where no temporal information is used in recovering the fine details even after applying a deblurring step. In contrast, the good quality of the *UP-SR* results is obtained thanks to the temporal fusion using the pixel–wise median filtering after a cumulative registration. This fusion plays a major role in attenuating the temporal noise and represents an appropriate process to deal with the problem of flying pixels. Moreover, the spatial deblurring step leads to further adding a smoothing effect while keeping sharp edges, hence, recovering fine details.

4.5.5 Statistical performance analysis

In order to illustrate the statistical analysis of the *UP-SR* algorithm with quantitative evaluation, we set up the following experiment. We use the publicly available toolbox V-REP [94] to create synthetic data with fully known ground truth for both dynamic and static scenes, Figure 4.7. (a), and Figure 4.7. (b), respectively. Three depth cameras with the same field of view are fixed at the same position. These cameras are of different resolutions, namely, 512^2, 256^2, and 128^2 pixels. They are used to capture three sequences for each subject. These sequences are further degraded with additive Laplacian noise with σ varying from 0 mm to 60 mm. Each sequence is super-resolved using *UP-SR* by considering 9 successive frames.

Starting with the static case, the corresponding MSE performance of the initialization step and the second deblurring step of *UP-SR* are reported in Figure 4.8 in solid and dashed lines, respectively. In the simple case where $r = 1$, the SR problem is merely a denoising one where the ground truth is estimated from 9 noisy measurements. In other words, the objective is not to increase resolution,



(a) (b)

Figure 4.7: Ground truth data used for the statistical performance analysis.

and hence there is no blur due to upsampling. Indeed, as seen in Figure 4.8, the solid red line overlaps with the dashed-dotted black line which corresponds to the theoretical variance for the odd case obtained using (4.24). A non-zero bias is found for $r = 2$ and $r = 4$ where the corresponding blue and green solid lines are above the theoretical variance. This suggests a correlation between motion and upsampling blur as expressed by the vector $\mathbf{u}_i$ in (4.19). We note an increased bias for a larger SR factor r. This is justified by a larger blur effect due to the dense upsampling and to motion. Finally, the dashed lines in Figure 4.8 confirm the performance enhancement after applying the optimization in (4.17); thus, ensuring an effective deblurring. We used an exhaustive search to find the best parameters for Γ_{BTV}. These quantitative results can be appreciated visually in Figure 4.10 where the noise level is fixed at $\sigma = 30\ mm$. The effective resolution enhancement, with a SR factor of $r = 4$, and denoising power of *UP-SR* for a static depth scene is seen in 3-D in Figure 4.10 (i). The average RMSE in 3-D is shown in Figure 4.10 (l).

In the dynamic case a similar behaviour has been observed with some differences related to the local motion estimation and data type. We can see that even for the simple case with $r = 1$ a non-zero bias from the theoretical variance is found for both the initial and optimized results, represented by the solid and dashed red lines in Figure 4.9, respectively. This bias is mainly due to the error caused by the self-occlusion. In the case of low resolution with $r = 2$ and $r = 4$,

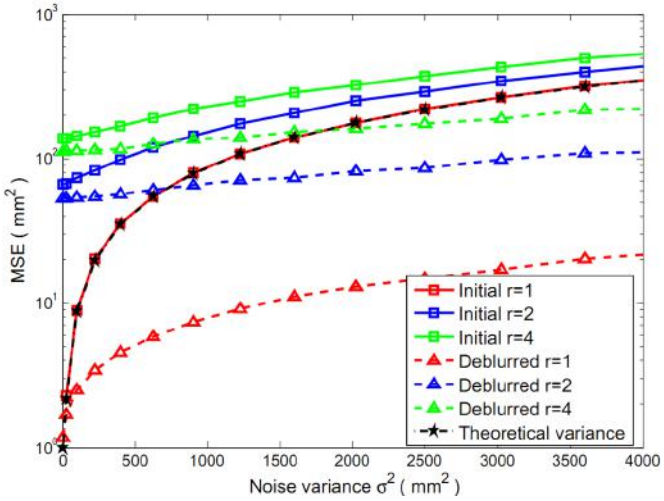


Figure 4.8: *UP-SR* MSE versus noise variance for a static scene.

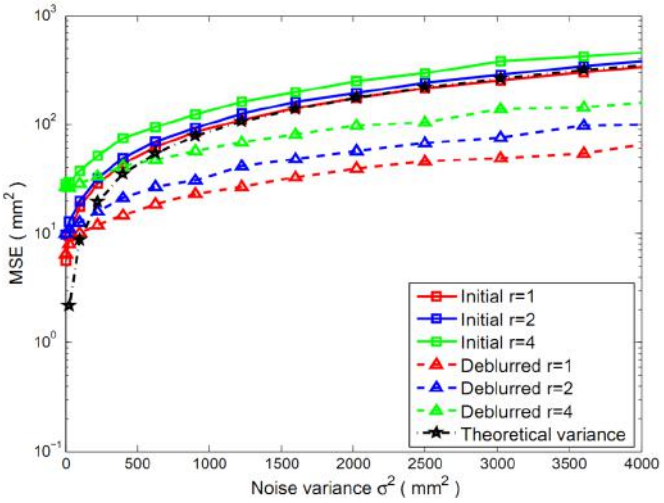


Figure 4.9: *UP-SR* MSE versus noise variance for a dynamic scene.

we can see that the non-zero bias in Figure 4.9 follows the same behaviour similar to the static case but with less shifting from the theoretical variance, especially for low noise levels as can be seen in the corresponding blue and green solid lines. This is directly related to the data type. Whereas, in the dynamic case we use a CAD object Figure 4.7. (a) with less details than the one used for the static case Figure 4.7. (b). Therefore, the downsampling process has more effect on the static object and leads to a larger loss in details, hence a larger bias.

4.6 Conclusion

A new multi-frame super-resolution algorithm for dynamic depth scenes has been proposed. It has been shown to be effective in enhancing the resolution of dynamic scenes with one or multiple non-rigidly moving objects. The proposed algorithm relies on two main components; first, an enhanced motion estimation based on a prior upsampling of the observed low resolution depth frames up to the super-resolution factor. Second, it uses a cumulative motion estimation accurately relating non-consecutive frames in the considered depth sequence, even for relatively large motions. In addition, the multi-frame super-resolution problem has been reformulated defining a simplified data model which is analogous to a classical image denoising problem with additive Laplacian noise, and using multiple observations. This has led to a median initial estimate, further refined by a deblurring operation using a bilateral total variation as the regularization term. For a thorough understanding of the impact of the different parameters, namely, number of observed frames N and the super-resolution factor r, a statistical model for the proposed approach in terms of MSE has been derived. One important conclusion is that the blur effect is due to both upsampling, motion and occlusions. Extensive evaluations using synthetic and real data have been carried out, showing the consistent good performance of the proposed approach in full correspondence with the derived theoretical statistical model.

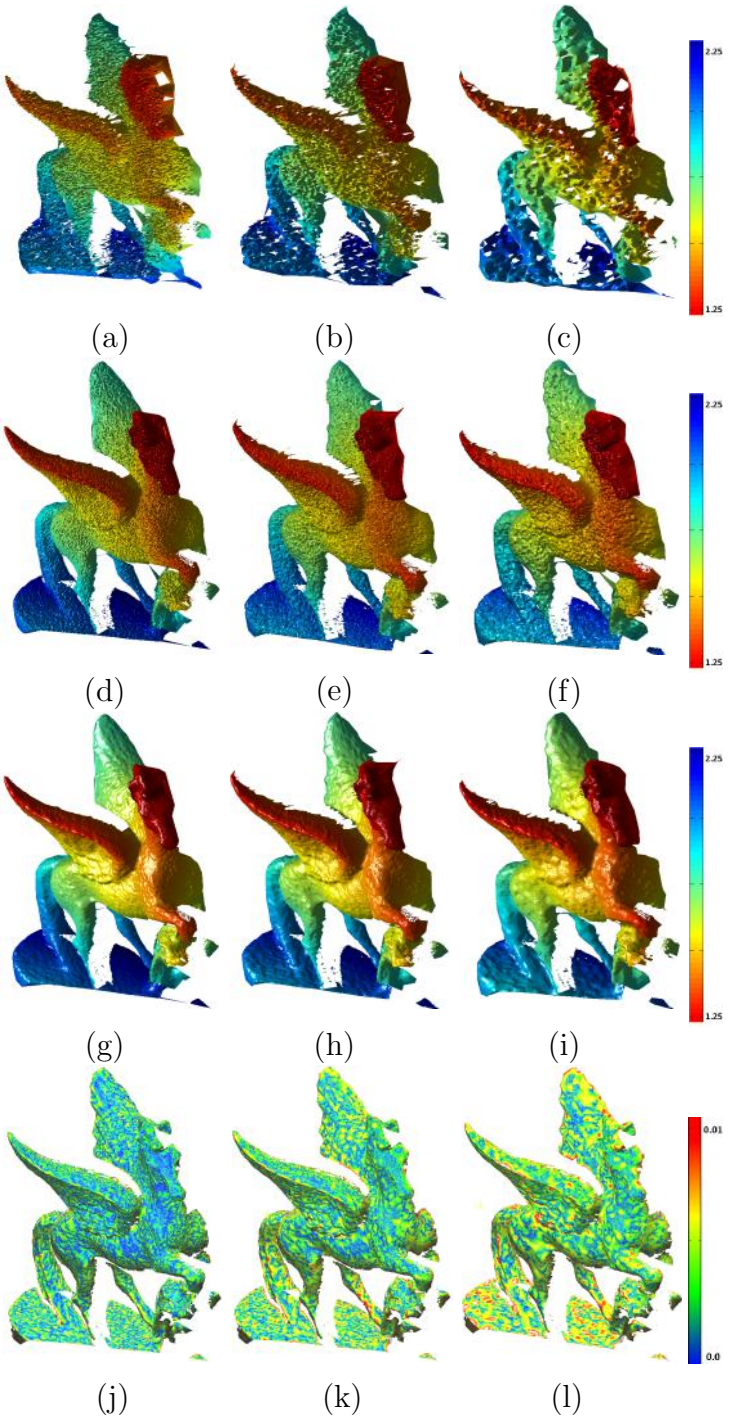


Figure 4.10: Statistical performance analysis of *UP-SR* for static depth scenes. First, second and third columns correspond respectively to $r = 1$, $r = 2$, and $r = 4$ where (a), (b) and (c) are the noisy LR observations; (d), (e), and (f) are the result of the Initial of *UP-SR*; (g), (h), and (i) are the result of deblurring step of *UP-SR*. The corresponding error maps as compared with the ground truth Figure 4.7. (b) are given in (j), (k), and (l).

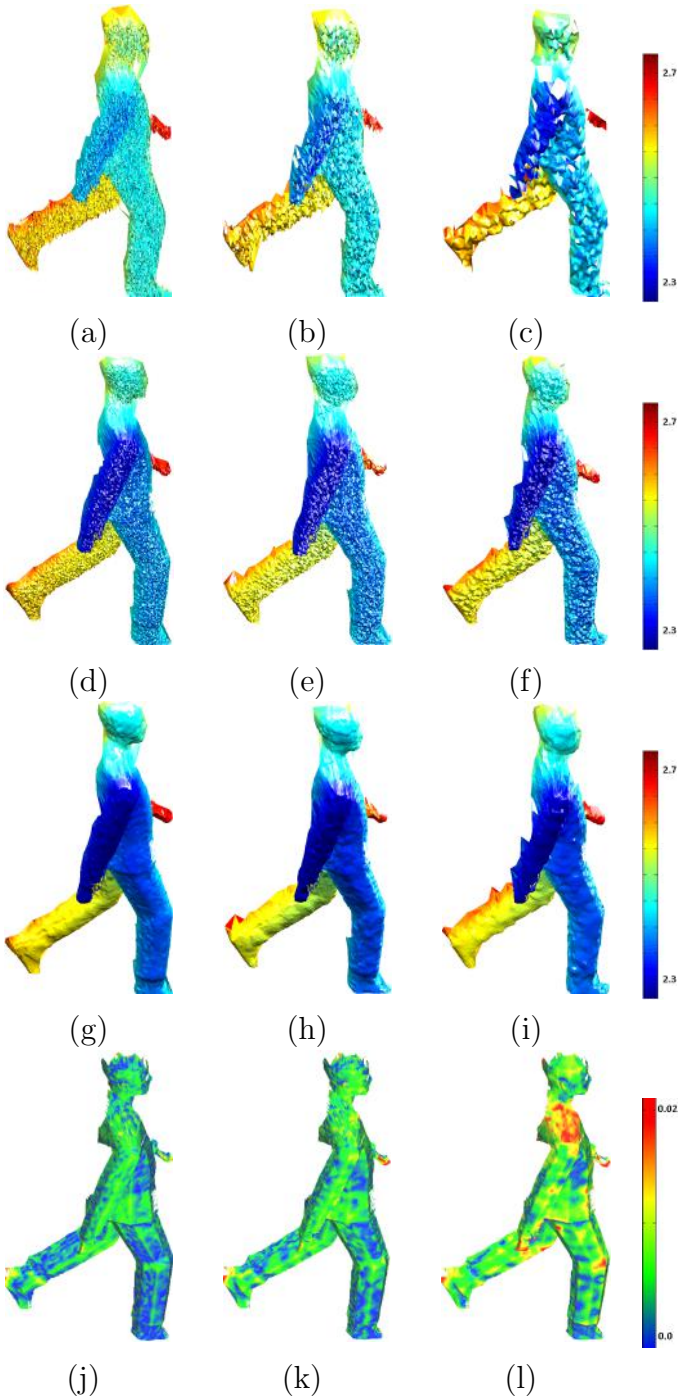


Figure 4.11: Statistical performance analysis of *UP-SR* for dynamic depth scenes. First, second and third columns correspond respectively to $r = 1$, $r = 2$, and $r = 4$ where (a), (b) and (c) are the noisy LR observations; (d), (e), and (f) are the result of the initialization step of *UP-SR*; (g), (h), and (i) are the result of the deblurring step of *UP-SR*. The corresponding error maps as compared with the ground truth Figure 4.7. (a) are given in (j), (k), and (l).

Chapter 5

Recursive Depth SR for Dynamic Scenes

Chapter 5 presents a dynamic multi–frame super–resolution algorithm which enhances low resolution dynamic depth videos containing freely non–rigidly moving objects. Existent methods are either limited to rigid objects, or restricted to global lateral motions. The proposed algorithm in Chapter 4, in addition, handles local lateral motions but still discards radial displacements. We herein address these shortcomings by accounting for non–rigid displacements in 3D. In addition to 2D optical flow, we estimate the depth displacement, and simultaneously correct the depth measurement by Kalman filtering. This concept is incorporated efficiently in a multi–frame super–resolution framework. It is formulated in a recursive manner that ensures an efficient deployment in real–time.

5.1 Introduction

In Chapter 4, we proposed the *UP-SR* algorithm as a multi–frame SR algorithm for dynamic depth scenes. This algorithm is, however, limited to lateral motions, and fails in the case of radial deformations. Moreover, it is not practical due to a heavy cumulative motion estimation process applied to a certain number of frames buffered in the memory. Alternatively, a recursive formulation may be thought of as in [95] where an iterative SR was proposed based on a block affine motion model resulting in a relatively efficient processing. This, however, is not

applicable to non–lateral motions. Earlier attempts for recursive SR approaches have proposed to use a Kalman filter formulation [78, 84, 96, 97, 98]. These methods work only under two conditions: constant translational motion between low resolution frames which represents the system motion model (i.e. transition matrix), and intensity consistency assumption between each pair of images in the video sequence. In the case of dynamic depth videos, these assumptions are not always valid. Indeed, for such videos, individual pixel motions have to be tracked through the video. A local motion model such as a dense 2D optical flow as in [13] is not sufficient, it is necessary to account for the full 3D motion in the SR reconstruction, known as scene flow, or the 2.5D motion, known as range flow.

For a reduced complexity we herein propose to approximate range flow by estimating radial motions on top of the 2D optical flow. Moreover, we propose a recursive depth multi–frame SR algorithm by using multiple Kalman filters. To ensure efficiency, we propose to treat a video as a set of one–dimensional signals. By so doing, we show that we reach an approximation of range flow; which enables us to take radial deformations into account in the SR estimation. To adequately preserve the smoothness properties of the depth surface, and remove noise and blur without over smoothing, we propose to use a multi–level version of the iterative bilateral total variation regularization given in [7]. In summary, the contribution of this chapter is a new mutli–frame depth SR algorithm which has the following properties: 1) Recursive, hence, suitable for real–time applications. 2) Robust to radial motions without explicitly computing range flow. 3) Accurate depth video reconstruction thanks to the proposed multi–level iterative bilateral regularization. An overview of the proposed algorithm is shown in Figure 5.1.

5.2 Background and problem formulation

Let us consider an LR video $\{\mathbf{g}_t\}$ acquired with a depth sensor. The captured scene is assumed to be dynamically and non–rigidly deforming without any assumption on the number of moving objects. Each LR observation $\mathbf{g}_t$ is represented by a column vector of length m corresponding to the lexicographic ordering of frame pixels. The objective of depth SR is to reconstruct an HR depth video $\{\mathbf{f}_t\}$ using $\{\mathbf{g}_t\}$, where each frame $\mathbf{f}_t$ is of length n with $n = r^2 \times m$ such that $r \in \mathbb{N}^*$

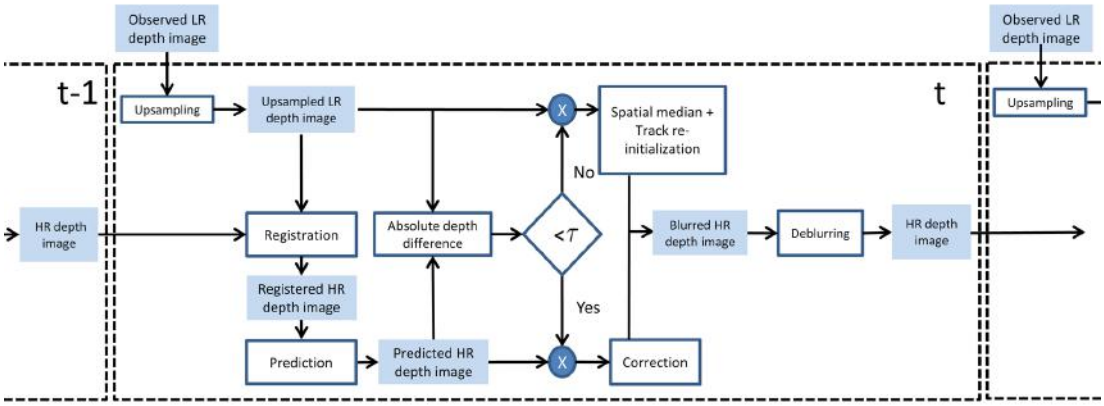


Figure 5.1: Flow chart of the proposed multi–frame depth super–resolution algorithm for dynamic depth videos containing one or multiple non–rigidly deforming objects.

is the SR scale factor. In the classical multi–frame depth SR problem, in order to reconstruct a given frame $\mathbf{f}_{t_0} \in \{\mathbf{f}_t\}$, also known as the reference frame, the N preceding observed LR frames are used.

An LR observation $\mathbf{g}_t$ is related to the reference frame through the following data model:

$$\mathbf{g}_t = \mathbf{DHM}_{t_0}^t \mathbf{f}_{t_0} + \mathbf{n}_t, \qquad t_0 \geq t, \tag{5.1}$$

where $\mathbf{D}$ is a known constant downsampling matrix of dimension $(m \times n)$. The system blur is represented by the time and space invariant matrix $\mathbf{H}$. The $(n \times n)$ matrices $\mathbf{M}_{t_0}^t$ correspond to the motion between $\mathbf{f}_{t_0}$ and $\mathbf{g}_t$ before downsampling. The vector $\mathbf{n}_t$ is an additive white noise at time instant t. Without loss of generality, both $\mathbf{H}$ and $\mathbf{M}_{t_0}^t$ are assumed to be block circulant commutative matrices. As a result, the estimation of $\mathbf{f}_{t_0}$ may be decomposed into two steps; estimation of a blurred HR image $\mathbf{z}_{t_0} = \mathbf{H}\mathbf{f}_{t_0}$, followed by a deblurring step to recover $\hat{\mathbf{f}}_{t_0}$. The *UP-SR* solution proposed in Chapter 4 computes a cumulative motion $\hat{\mathbf{M}}_t^{t_0}$ through the estimation of intermediate dense lateral motions $\hat{\mathbf{M}}_t^{t+1}$ between consecutive frames. The information in $\hat{\mathbf{M}}_t^{t+1}$ is equivalent to finding the horizontal and vertical displacements in pixels $u_t^{\mathtt{i}}$ and $v_t^{\mathtt{i}}$, respectively, for each pixel position $\mathtt{p}_t^{\mathtt{i}} = (x_t^{\mathtt{i}}, y_t^{\mathtt{i}}), \mathtt{i} = 1, \cdots, n$. In the continuous case, these displacements correspond to the lateral motions $\mathtt{m}_t^{\mathtt{i}} = (u_t^{\mathtt{i}}, v_t^{\mathtt{i}})$ where $u_t^{\mathtt{i}} = \frac{dx_t^{\mathtt{i}}}{dt}$ and $v_t^{\mathtt{i}} = \frac{dy_t^{\mathtt{i}}}{dt}$.

Radial displacements in the depth direction, often encountered in depth sequences, are not handled. In order to address this problem, it is important to consider and incorporate the full 3D motion per pixel or the 2.5D version of dense optical flow [60], known as range flow, in the *UP-SR* framework (Section 2.3).

In this work we propose to decouple the estimation of lateral local motions $\mathtt{m}_t^{\mathtt{i}}$ from the estimation of the radial displacement $w_t^{\mathtt{i}}$. This is in order to reduce complexity, but also in order to introduce a probabilistic framework that allows us to recursively estimate $w_t^{\mathtt{i}}$ and the corrected depth value at the same point. We propose to use (2.10) to find $\mathtt{m}_t^{\mathtt{i}}$ first. Then, we proceed to estimate $w_t^{\mathtt{i}}$ under a probabilistic framework where we account for radial motion uncertainties.

5.3 Proposed approach

The proposed depth video enhancement approach is based on an extension of the *UP-SR* algorithm. As our goal is a real–time processing, the major difference resides in replacing the cumulation of N frames in *UP-SR* for processing a reference frame at time t_0, by a recursive processing that only considers two consecutive frames at $(t-1)$ and t where the current frame is to be enhanced each time. We denote the i^{th} element of a lexicographically ordered vector image $\mathbf{x}$ as $\mathbf{x}^{\mathtt{i}}$. The measurement model per pixel for each current frame may then be defined by setting $t_0 = t$ in (4.13), resulting in

$$\tilde{\mathbf{z}}_t^{\mathtt{i}} := [\mathbf{g}_t \uparrow]^{\mathtt{i}} = \mathbf{z}_t^{\mathtt{i}} + [\mathbf{n}_t \uparrow]^{\mathtt{i}} \qquad \forall t, \tag{5.2}$$

where $\mathbf{n}_t \uparrow = \mathbf{U}\mathbf{n}_t$, with $\mathbf{U}$ being the upsampling matrix defined in Section 3.4. In this work, the additive noise $[\mathbf{n}_t \uparrow]^{\mathtt{i}}$ is assumed to be zero mean Gaussian with the variance σ_n^2, i.e., $[\mathbf{n}_t \uparrow]^{\mathtt{i}} \sim \mathcal{N}(0, \sigma_n^2)$.
The problem at hand is then to estimate $\mathbf{z}_t^{\mathtt{i}}$ given a noisy measurement $\tilde{\mathbf{z}}_t^{\mathtt{i}}$ and an enhanced noise–free depth value $\mathbf{z}_{t-1}^{\mathtt{i}}$ estimated at the preceding iteration.
The time–deforming depth scene is viewed as a dynamic system where the state of each pixel is defined by its depth value and radial displacement. These states are estimated dynamically over time using a Kalman filter. The *UP-SR* dynamic model in (4.3) is directly used to characterize the dynamic system and introduce

the uncertainties of depth measurements and radial deformations in one probabilistic framework. The proposed recursive approach, that we will refer to as *RecUP-SR*, is summarized in the flow chart of Figure 5.1. The main steps are described in what follows.

5.3.1 Lateral registration

In order to be able to carry a per–pixel processing, essential for handling non–rigid deformations, one needs to properly align these pixels between consecutive frames. This is achieved by registration through 2D dense optical flow that estimates the lateral motion between the intensity images $\mathbf{a}_{t-1}$ and $\mathbf{a}_t$.
In the case of RGB-D cameras, these images are provided directly. Mapping and synchronization have to be ensured, though, as in [64] and [66].
In the case of ToF cameras, the provided intensity images, known as amplitude images, can not be used directly. Their intensity values differ significantly depending on the camera integration time and on the distance of the scene from the camera; hence, not verifying the optical flow assumption of brightness consistency. Thus, in order to guarantee an accurate registration, it is necessary to apply a standardization step similar to the one proposed in [12] prior to motion estimation, see Figure 5.2.

If intensity images are not available, for example when using synthetic data, the 2D optical flow can be directly estimated using LR raw depth images, but after a denoising step (e.g. using a bilateral filter). We note that this denoising should only be used in the preprocessing step. The original raw depth data is the one to be mapped in the registration step.
In all cases, as for *UP-SR*, we register the upsampled versions of the LR images after upscaling the motion vectors estimated from the LR images. We define the registered depth image from $(t-1)$ to t as $\bar{\mathbf{z}}_{t-1}^{t}$. Consequently, the radial displacement $w_t^{\mathbf{i}}$ may be initialized by the temporal difference between depth measurements, i.e.,

$$w_t^{\mathbf{i}} \approx \tilde{\mathbf{z}}_t^{\mathbf{i}} - [\bar{\mathbf{z}}_{t-1}^{t}]^{\mathbf{i}}. \tag{5.3}$$

This first approximation of $w_t^{\mathbf{i}}$ is an initial value that requires further refinement directly accounting for the system noise. We propose to do that using a per–pixel

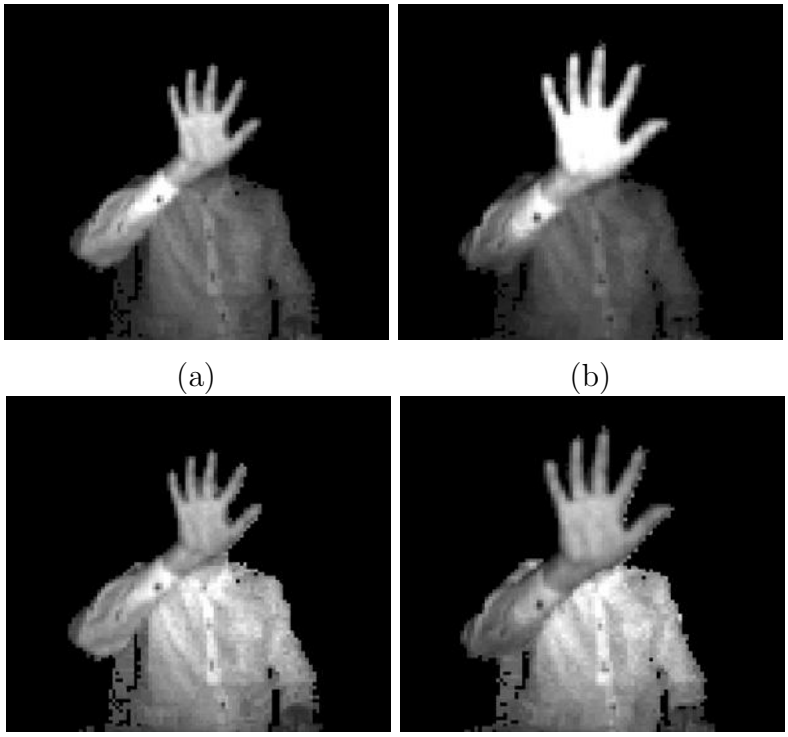


Figure 5.2: Correcting amplitude images using a standardization step [12]. (a) and (b) show the original amplitude images for a dynamic scene containing a hand moving towards the camera where the intensity (amplitude) values differ significantly depending on the object distance from the camera. The corrected amplitude images for the same scene are presented in (c) and (d), where the intensity consistency is preserved.

tracking with a Kalman filter as detailed in Section 5.3.2.

5.3.2 Refinement by per–pixel tracking

According to the definition of image pixel registration, we have $\mathbf{z}^{\mathtt{i}}_{t-1} := [\bar{\mathbf{z}}^{t}_{t-1}]^{\mathtt{i}}$. The dynamic model follows from (4.3) as

$$\mathbf{z}^{\mathtt{i}}_{t} = \mathbf{z}^{\mathtt{i}}_{t-1} + \boldsymbol{\mu}^{\mathtt{i}}_{t} \qquad \forall t, \tag{5.4}$$

where $\boldsymbol{\mu}_t$ is a noisy version of the innovation $\boldsymbol{\delta}_t$ first introduced in the *UP-SR* dynamic model in (4.3). Whereas in *UP-SR* this innovation is neglected, in

RecUP-SR it is assimilated to the uncertainty considered in the dynamic model. In this work, we assume a constant velocity model with an acceleration $\gamma_t^{\mathbf{i}}$ following a Gaussian distribution $\gamma_t^{\mathbf{i}} \sim \mathcal{N}(0, \sigma_a^2)$. As a result, the noisy innovation may be expressed as:

$$\boldsymbol{\mu}_t^{\mathbf{i}} = w_{t-1}^{\mathbf{i}} \Delta t + \frac{1}{2} \gamma_t^{\mathbf{i}} \Delta t^2. \tag{5.5}$$

The dynamic model in (5.4) can then be rewritten as:

$$\begin{cases} \mathbf{z}_t^{\mathbf{i}} = \mathbf{z}_{t-1}^{\mathbf{i}} + w_{t-1}^{\mathbf{i}} \Delta t + \frac{1}{2} \gamma_t^{\mathbf{i}} \Delta t^2 \\ w_t^{\mathbf{i}} = w_{t-1}^{\mathbf{i}} + \gamma_t^{\mathbf{i}} \Delta t \end{cases}. \tag{5.6}$$

Considering the following state vector:

$$\mathbf{s}_t^{\mathbf{i}} = \begin{pmatrix} \mathbf{z}_t^{\mathbf{i}} \\ w_t^{\mathbf{i}} \end{pmatrix}, \tag{5.7}$$

where both the depth measurement and the radial displacement are to be filtered, (5.6) becomes:

$$\mathbf{s}_t^{\mathbf{i}} = \mathbf{K} \mathbf{s}_{t-1}^{\mathbf{i}} + \boldsymbol{\gamma}_t^{\mathbf{i}}, \tag{5.8}$$

with $\mathbf{K} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$, and $\boldsymbol{\gamma}_t^{\mathbf{i}} = \gamma_t^{\mathbf{i}} \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix}$ is the process noise which is white Gaussian with the covariance

$$\mathbf{Q} = \sigma_a^2 \Delta t^2 \begin{pmatrix} \Delta t^2/4 & \Delta t/2 \\ \Delta t/2 & 1 \end{pmatrix}. \tag{5.9}$$

Using standard Kalman equations, the prediction is achieved as

$$\begin{cases} \hat{\mathbf{s}}_{t|t-1}^{\mathbf{i}} = \mathbf{K} \mathbf{s}_{t-1|t-1}^{\mathbf{i}}, \\ \hat{\mathbf{P}}_{t|t-1}^{\mathbf{i}} = \mathbf{K} \mathbf{P}_{t-1|t-1}^{\mathbf{i}} \mathbf{K}^T + \mathbf{Q}. \end{cases} \tag{5.10}$$

The error in the prediction of $\hat{\mathbf{s}}_{t|t-1}^{\mathbf{i}}$ is corrected using the observed measurement $\tilde{\mathbf{z}}_t^{\mathbf{i}}$. This error is considered as the difference between the prediction and the observation, and weighted using the Kalman gain matrix $\mathbf{G}_{t|t}^{\mathbf{i}}$ which is calculated as follows:

$$\mathbf{G}_{t|t}^{\mathbf{i}} = \hat{\mathbf{P}}_{t|t-1}^{\mathbf{i}} \mathbf{b}^T \left(\mathbf{b} \hat{\mathbf{P}}_{t|t-1}^{\mathbf{i}} \mathbf{b}^T + \sigma_n^2 \right)^{-1}, \tag{5.11}$$

such that the observation vector is $\mathbf{b} = (1, 0)^T$. The corrected state vector $\mathbf{s}_{t|t}^{\mathbf{i}} = \begin{pmatrix} \mathbf{z}_{t|t}^{\mathbf{i}} \\ w_{t|t}^{\mathbf{i}} \end{pmatrix}$ and corrected error covariance matrix $\mathbf{P}_t^{\mathbf{i}}$ are computed as follows:

$$\begin{cases} \mathbf{s}_{t|t}^{\mathbf{i}} = \hat{\mathbf{s}}_{t|t-1}^{\mathbf{i}} + \mathbf{G}_{t|t}^{\mathbf{i}} \left(\tilde{\mathbf{z}}_t^{\mathbf{i}} - \mathbf{b} \hat{\mathbf{s}}_{t|t-1}^{\mathbf{i}} \right), \\ \mathbf{P}_{t|t}^{\mathbf{i}} = \hat{\mathbf{P}}_{t|t-1}^{\mathbf{i}} - \mathbf{G}_{t|t}^{\mathbf{i}} \mathbf{b} \hat{\mathbf{P}}_{t|t-1}^{\mathbf{i}}. \end{cases} \tag{5.12}$$

This per–pixel filtering is extended to all the depth frame resulting in n Kalman filters run in parallel. Each filter tracks the trajectory of one pixel. At this level, pixel trajectories are assumed to be independent. The advantage of the processing per pixel is to keep all the required matrix inversions for (2×2) matrices. The burden of traditional Kalman filter–based SR as in [78] will consequently be avoided. Moreover, for a recursive multi–frame SR algorithm, instead of using the video sequence of length N to recover one frame, we use the preceding recovered frame $\hat{\mathbf{f}}_{t-1}$ to estimate $\mathbf{f}_t$ from the current upsampled observation $\mathbf{g}_t \uparrow$.

Furthermore, in order to separate background from foreground depth pixels, and tackle the problem of flying pixels, especially around edges, we define a condition for track re-initialization. This condition is based on a fixed threshold τ such that:

$$\begin{cases} \text{Continue the track} & \text{if } |\tilde{\mathbf{z}}_t^{\mathbf{i}} - \hat{\mathbf{z}}_{t|t-1}^{\mathbf{i}}| < \tau; \\ \text{New track \& spatial median} & \text{if } |\tilde{\mathbf{z}}_t^{\mathbf{i}} - \hat{\mathbf{z}}_{t|t-1}^{\mathbf{i}}| \geqslant \tau. \end{cases}$$

The choice of the threshold value τ is related to the type of the used depth sensor and the level of the sensor–specific noise.

The assumption of independent trajectories leads to blurring artifacts, and requires a corrective step to bring back the correlation between neighbouring pixels from the original depth surface $\mathcal{Z}$. To that end, we use an L_1 minimization where we propose a multi–level iterative BTV regularization as detailed in Section 5.3.3.

5.3.3 Multi–level iterative bilateral TV deblurring

Similarly to the *UP–SR* algorithm, $\mathbf{f}_t$ is estimated in two steps; first, finding a blurred version $\hat{\mathbf{z}}_t$, which is the result of Section 5.3.2. Then the deblurring takes place to recover $\hat{\mathbf{f}}_t$ from $\hat{\mathbf{z}}_t$. To that end, we apply the following deblurring framework:

$$\hat{\mathbf{f}}_t = \underset{\mathbf{f}_t}{\operatorname{argmin}} \Big(\|\mathbf{B}\mathbf{f}_t - \hat{\mathbf{z}}_t\|_1 + \lambda \Gamma(\mathbf{f}_t) \Big), \tag{5.13}$$

where λ is a regularization parameter that controls the amount of regularization needed to recover the original blur and noise–free frame. We choose to use a BTV regularizer [7] in order to enforce the properties of bilateral filtering on the final solution [48, 99, 100]. It is a filter that has been shown to perform well on depth

data [18, 101, 102]. Indeed, it is a filter that smoothes an image while preserving its sharp edges based on pixel similarities in both the spatial and in the intensity domains. The BTV regularizer is defined as:

$$\Gamma(\mathbf{f}_t) = \sum_{\mathtt{i}=-\mathtt{I}}^{\mathtt{i}=\mathtt{I}} \sum_{\mathtt{j}=-\mathtt{J}}^{\mathtt{j}=\mathtt{J}} \alpha^{|\mathtt{i}|+|\mathtt{j}|} \parallel \mathbf{f}_t - \mathbf{S}_x^{\mathtt{i}} \mathbf{S}_y^{\mathtt{j}} \mathbf{f}_t \parallel_1 . \tag{5.14}$$

The matrices $\mathbf{S}_x^{\mathtt{i}}$ and $\mathbf{S}_y^{\mathtt{j}}$ are shifting matrices which shift $\mathbf{f}_t$ by $\mathtt{i}$, and $\mathtt{j}$ pixels in the horizontal and vertical directions, respectively. The scalar $\alpha \in]0, 1]$ is the base of the exponential kernel which controls the speed of decay.

Minimizing the cost function in (5.13) has shown to give good results in *UPSR* [41]; however, unless all the parameters are perfectly chosen, which is a challenge in itself, the final result can end up being a denoised and deblurred version of $\mathbf{f}_t$, which is also over–smoothed. This issue has been addressed by iterative regularization in the case of denoising [45, 46, 47, 103], and in the more general case of deblurring [104].

In the same spirit, we use an iterative regularization where we propose to focus on the choice of the regularization parameter λ. Specifically, our deblurring method consists in running the minimization (5.13) multiple times where the regularization strength is progressively reduced in a dyadic way. We define, thus, a multi–level iterative deblurring with a BTV regularization such that the solution at level l is

$$\hat{\mathbf{f}}_t^{(l)} = \underset{\mathbf{f}_t^{(l)}}{\operatorname{argmin}} \Big(\|\mathbf{B}\mathbf{f}_t^{(l)} - \mathbf{f}_t^{(l-1)}\|_1 + \frac{\lambda}{2^l} \Gamma(\mathbf{f}_t^{(l)}) \Big), \text{ with } \mathbf{f}_t^{(0)} = \hat{\mathbf{z}}_t. \tag{5.15}$$

Combined with a steepest descent numerical solver, the proposed solution is described by the following pseudo–code: The parameter β is a scalar which represents the step size in the direction of the gradient, $\mathbf{I}$ is the identity matrix, and $\operatorname{sign}(\cdot)$ is the sign function. The parameter L is the number of levels considered, and K is the number of iterations for one level.

We note that the correct formulation of the problem at the beginning of Section 5.3 is to use the final deblurred depth value $\mathbf{f}_{t-1}^{\mathtt{i}}$ obtained as a solution of (5.15) instead of $\mathbf{z}_{t-1}^{\mathtt{i}}$.

Algorithm 5.1 Multi-level iterative bilateral total variation deblurring.

for $l = 1, \cdots, L$

for $k = 1, \cdots, K$

$$\hat{\mathbf{f}}_t^{(l,k)} = \hat{\mathbf{f}}_t^{(l,k-1)} - \beta \left\{ \mathbf{B}^T \text{sign}\left(\mathbf{B}\hat{\mathbf{f}}_t^{(l,k-1)} - \mathbf{z}_t \right) + \frac{\lambda}{2^l} \sum_{\mathtt{i}=-\mathtt{I}}^{\mathtt{i}=\mathtt{I}} \sum_{\mathtt{j}=-\mathtt{J}}^{\mathtt{j}=\mathtt{J}} \alpha^{|\mathtt{i}|+|\mathtt{j}|} \left(\mathbf{I} - \mathbf{S}_y^{-\mathtt{j}} \mathbf{S}_x^{-\mathtt{i}} \right) \text{sign}\left(\hat{\mathbf{f}}_t^{(l,k-1)} - \mathbf{S}_x^{\mathtt{i}} \mathbf{S}_y^{\mathtt{j}} \hat{\mathbf{f}}_t^{(l,k-1)} \right) \right\}$$

end for

$\mathbf{z}_t \longleftarrow \hat{\mathbf{f}}_t^{(l,K)}$

end for

5.4 Experimental results

In this section, we test the proposed *RecUP-SR* algorithm on different LR depth videos where we evaluate its performance using: (i) synthetic depth videos with a known ground truth, and (ii) real depth videos of dynamic scenes with non–rigid deformations captured by a ToF camera (PMD camboard nano [3]). The synthetic data is used in order to provide a quantitative evaluation as compared to state–of–art methods as well as a full understanding of the contribution of the intermediate steps of *RecUP-SR* on the quality of the final result. The tested examples vary from simple scenes with one moving object to a more complex cluttered scenes containing multiple moving objects with non–rigid deformations.

5.4.1 Evaluation on synthetic data

We evaluate the performance of the *RecUP-SR* algorithm at different levels. First, we show how it is efficient in filtering both the depth value as well as the radial displacement and hence the corresponding velocity. Then, we compare the accuracy of the reconstructed 3D super–resolved scene with state–of–art results. The comparison is done by back–projecting the reconstructed HR depth images to the 3D world using the camera matrix and calculating the 3D Root Mean Squared Error (RMSE) of each back–projected 3D point cloud with respect to the ground truth 3D point cloud. Finally, we show the importance of the contribution of

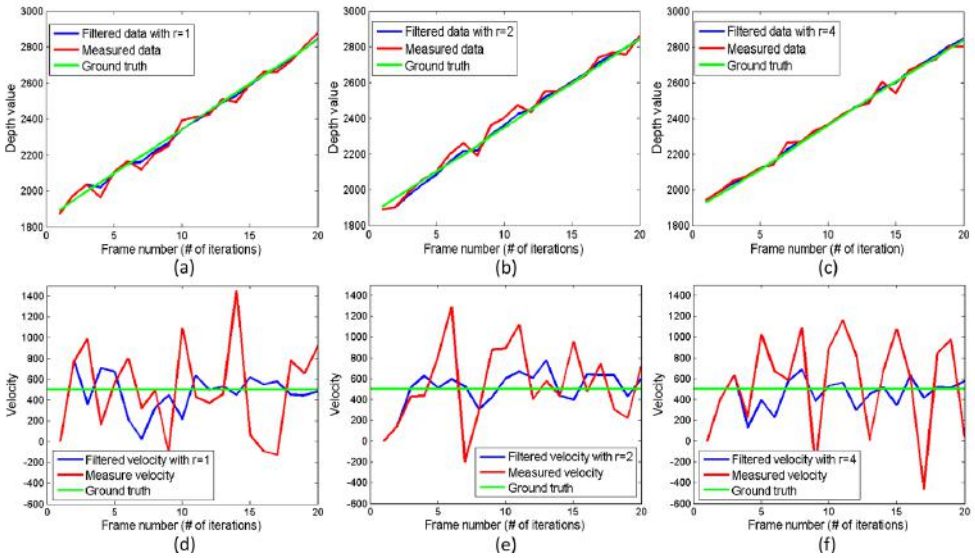


Figure 5.3: Tracking results for different depth values randomly chosen from the super–resolved sequences with different SR scale factors $r = 1, r = 2$, and $r = 4$, are plotted in (a), (b), and (c), respectively. The corresponding filtered velocities are shown in (d), (e), and (f), respectively.

each step of the proposed *RecUP-SR* algorithm in improving the quality of the final result.

5.4.1.1 Filtering of depth measurements and radial displacements

We start with a simple and fully controlled scene containing one 3D object moving radially with respect to the camera. The considered object in this experiment is a synthetic hand. A sequence of 20 depth frames is captured with a 5 cm difference between each two successive frames, and $\Delta t = 0.1$ seconds. The generated sequence is downsampled with a scale factor of $r = 2$, and $r = 4$, and further degraded with additive Gaussian noise with a standard deviation σ varying from 10 to 80 mm. We then super–resolve the obtained LR noisy depth sequences by applying the proposed algorithm with a scale factor of $r = 1$, $r = 2$, and $r = 4$. Obtained results show that by increasing the scale factor r, a higher 3D error is introduced as seen in Figure 5.4. In the simple case where $r = 1$, the SR problem is merely a denoising one, and hence there is no blur due to upsampling. In

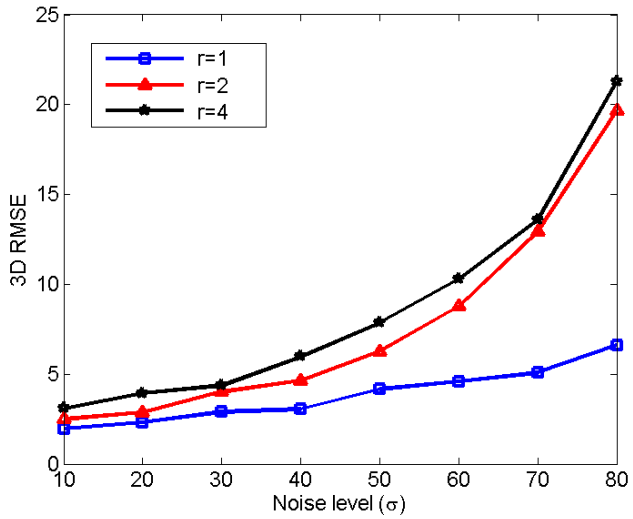


Figure 5.4: 3D RMSE in mm of the super–resolved hand sequence using the proposed method with different SR scale factors. Increasing the SR factor leads to a higher 3D reconstruction error. This is due to the blurring effects of the upsampling process and the lower resolution of the used LR depth sequence as compared to the one used with $r = 1$.

contrast, by increasing the SR factor r, more blurring effects occur leading to a higher 3D error.

Furthermore, in order to evaluate the quality of the filtered depth data and the filtered velocity, we randomly choose one pixel $\mathbf{p}_t^{\mathbf{i}}$ from each super–resolved sequence with $r = 1$, $r = 2$, and $r = 4$, and a fixed noise level for $\sigma = 50\ mm$. For each one of these pixels, we track the corresponding enhanced depth value $\mathbf{f}_t^{\mathbf{i}}$ and the corresponding enhanced velocity $\frac{\Delta \mathbf{f}_t^{\mathbf{i}}}{\Delta t}$ through the super-resolved sequence. In Figure 5.3 (a), (b), and (c), we can see how the depth values are filtered (blue lines) as compared to the noisy depth measurements (red lines) for all scale factors. Similar behaviour is observed for the corresponding filtered velocities in Figure 5.3 (d), (e), and (f).

5.4.1.2 Comparison with state–of–art methods

In order to compare our algorithm with state-of-art methods, we use a more complex scene with a highly non–rigidly moving object. We use the publicly available

Table 5.1: 3D RMSE in mm for the super–resolved dancing girl sequence using different SR methods. These methods are applied on LR noisy depth sequences with two noise levels. The SR scale factor for this experiment is $r = 4$.

	$\sigma = 25mm$				$\sigma = 50mm$			
	Arm	Torso	Leg	Full body	Arm	Torso	Leg	Full body
Bicubic	10.5	7.5	8.9	8.8	25.2	14.9	13.1	16.5
SISR	**9.0**	5.6	8.4	6.6	14.1	6.9	9.6	9.7
UP-SR	22.2	15.6	9.3	15.9	29.7	17.4	12.8	23.5
Proposed	9.6	**3.6**	**7.5**	**6.3**	**9.9**	**4.8**	**8.1**	**9.5**

"Samba" [8] data. This dataset provides a real sequence of a full 3D dynamic dancing lady scene with high resolution ground truth. This sequence contains both non–rigid radial motions and self–occlusions, represented by arms and legs movements, respectively. We use the publicly available toolbox V-REP [94] to create from the "Samba" data a synthetic depth sequence with fully known ground truth. We choose to fix a depth camera at a distance of 2 meters from the 3D scene. Its resolution is 1024^2 pixels. The camera is used to capture the depth sequence. Then, similarly to the previous set-up, we downsample the obtained depth sequence with $r = 4$ and further degrade it with additive Gaussian noise with standard deviation σ varying from 0 to 50 mm. The created LR noisy depth sequence is then super–resolved using state-of-art methods, the conventional bicubic interpolation, *UP-SR* [13], *SISR* [9], and the proposed *RecUP-SR*. Table 5.1 reports the 3D reconstruction error of each method at different noise levels. The comparison is done at two levels: (i) Different parts of the reconstructed 3D body, namely, arm, torso, and the leg, and (ii) full reconstructed 3D body. As expected, by applying the conventional bicubic interpolation method directly on depth images, a large error is obtained. This error is mainly due to the flying pixels around object boundaries. Thus, we run another round of experiments using a modified bicubic interpolation, where we remove all flying pixels by defining a fixed threshold. Yet, the 3D reconstruction error is still relatively high across all noise levels, see Table 5.1. This is due to the fact that bicubic interpolation does not profit from the temporal information provided by the sequence. We observe in Table 5.1 that the proposed method provides, most of the time, better results as compared

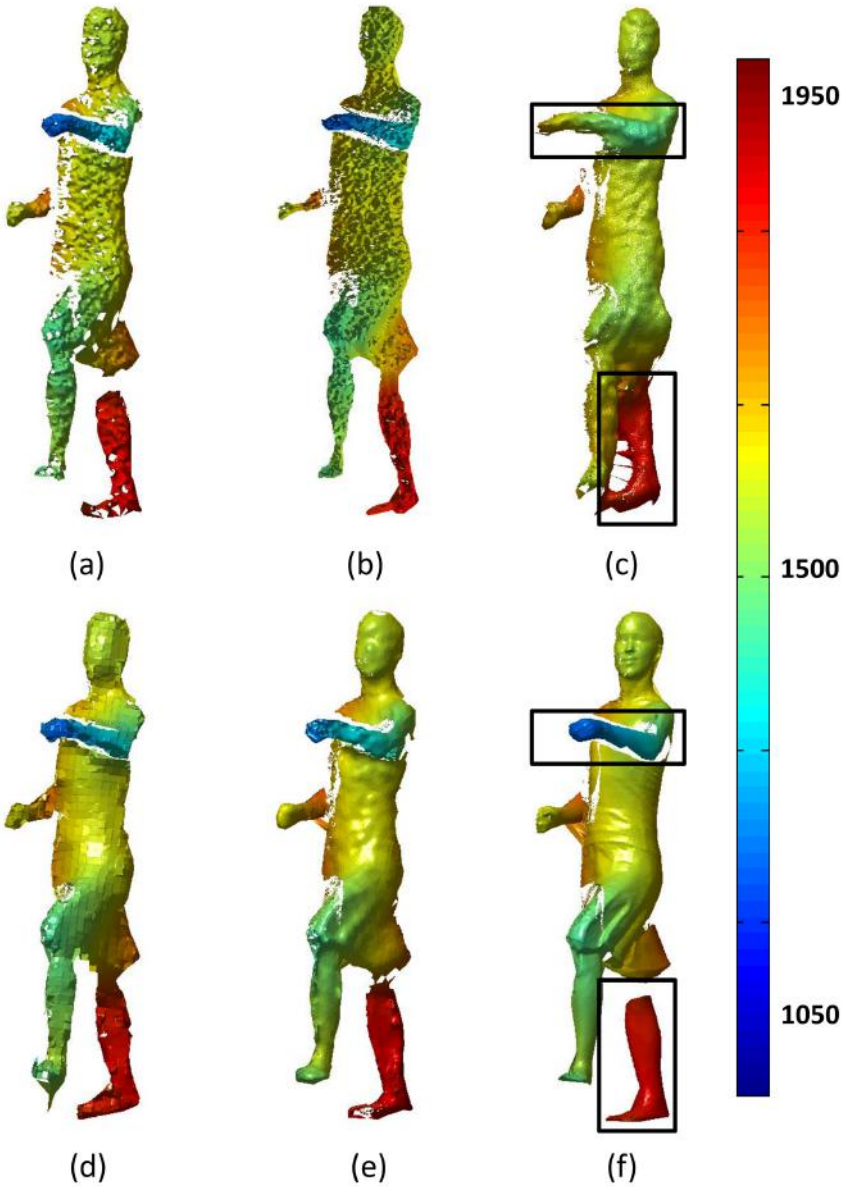


Figure 5.5: 3D Plotting of one super–resolved depth frame with $r = 4$ using: (b) bicubic interpolation, (c) Patch based single image SR (*SISR*) [9], (d) *UP-SR* [13], (e) Proposed *RecUP-SR* algorithm with $[L = 3, K = 7, \lambda = 2.5]$. (a) 3D plotting of one LR noisy depth frame. (f) 3D ground truth. Distance units on the coloured bar are in mm.

to state-of-art algorithms. In order to visually evaluate the performance of the proposed *RecUP-SR* algorithm, we plot the super–resolved results of the dancing girl sequence in 3D. We show the results for the sequence at $\sigma = 30\ mm$. We note that *RecUP-SR* outperforms state-of-art methods by keeping the fine details (e.g. the details of the face) as can be seen in Figure 5.5 (e). Note that the *UP-SR* algorithm fails in the presence of radial movements and self–occlusions, see black boxes in Figure 5.5 (c). In contrast, the *SISR* algorithm can handle these cases, but cannot keep the fine details due to its patch–based nature, see Figure 5.5 (d). In addition, a heavy training phase is required.

5.4.1.3 Evaluation of the effects of different steps

In order to better understand the contribution of each step of the proposed algorithm, we consider the "Facecap" data [105] which is a simple scene of a real 3D face sequence with non–rigid deformations. We use a similar setup to the one used with the "Samba" dataset by fixing a camera at a distance of 0.7 meter from the 3D face. We create a new synthetic depth sequence of the moving face. Then, we downsample the obtained depth sequence with $r = 4$ and further degrade it with additive Gaussian noise with standard deviation σ varying from 0 to 20 mm. The obtained LR noisy depth sequence is then super–resolved with $r = 4$ using: 1) Kalman filter, 2) spatial deblurring, and 3) the proposed *RecUP-SR* algorithm. In the deblurring process, two different techniques are considered, one–level deblurring and the proposed multi–level deblurring. The accuracy of the reconstructed 3D face sequences is measured by calculating the 3D RMSE. In Figure 5.6, we report the obtained results for the super–resolved LR noisy depth sequence with $\sigma = 10\ mm$. We see how the Kalman filter attenuates the noise gradually and hence decreasing the 3D RMSE for an increased number of frames (black solid line). We notice that, in the presence of a non-smooth motions, the constant velocity filtering model needs few number of iterations (frames) before converging which affects the reconstruction quality of the super–resolved depth frame. For example, due to the up and down non–smooth and fast motions of the eye brows between frame number 20 to 25, the per–pixel temporal filtering is not converged yet, and hence the 3D error increases for a few number of frames

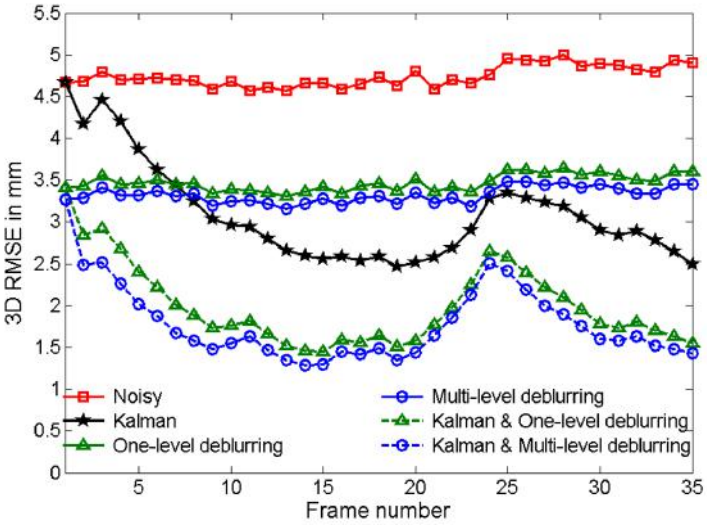


Figure 5.6: Effects of applying different steps separately and combined on a sequence of 35 LR noisy depth frames with $\sigma = 10\ mm$. The combination of the Kalman filter with the spatial multi–level deblurring provides the best performance in reducing the 3D RMSE.

before decreasing again Figure 5.6 (Black solid line).

By considering the deblurring step alone without engaging in the per–pixel temporal filtering process, we can see that the 3D RMSE is almost constant throughout the sequence as shown in Figure 5.6 (solid blue and green lines). This can be explained by the fact that there is no engagement of temporal information. Instead only a spatial filtering is applied at each frame independently of each other. Finally, by looking at the obtained results in Figure 5.6, we find that the best performance is achieved by combining the spatial and the temporal filters (blue and green dashed lines), with an advantage of using the proposed multi–level deblurring approach over the one–level conventional deblurring approach. Note that an intensive search is applied to find the best deblurring parameters which lead to the smallest 3D RMSE error. This combination, in fact, constitutes the key component of the proposed algorithm. In Figure 5.7, we show the physical effects of the previously discussed cases by plotting the corresponding 3D super–resolved results of the last HR depth frame in the sequence. Starting from the first column, we show the LR noisy faces for different noise levels. The filtered

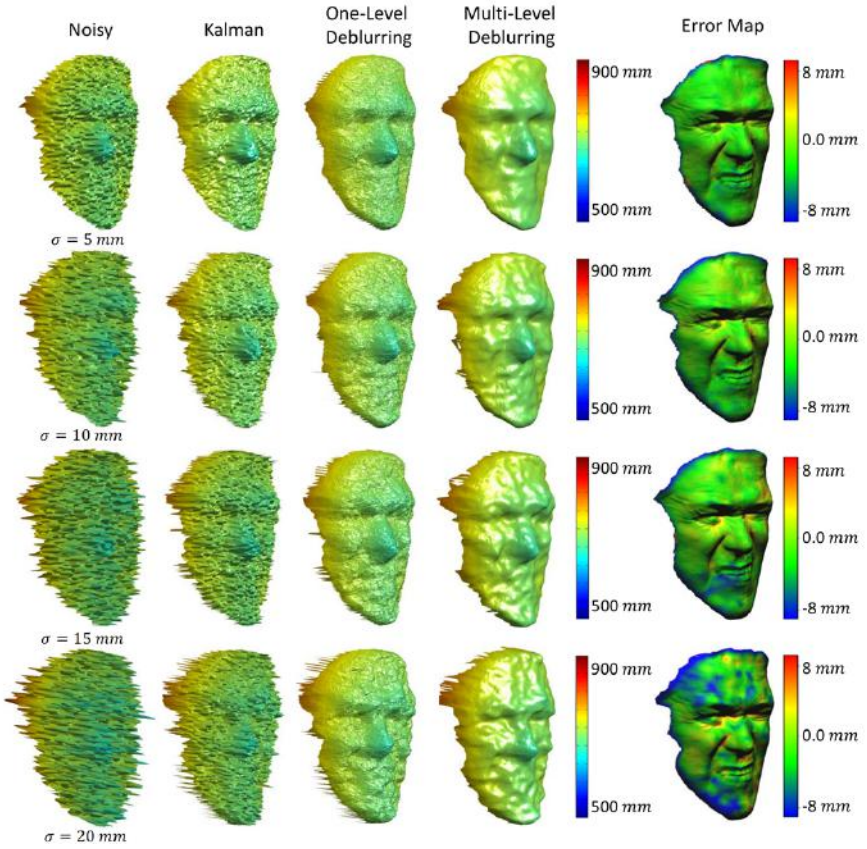


Figure 5.7: 3D plotting of (starting from left column): 1) LR noisy depth frames, 2) super–resolved depth frames with $r = 4$ using Kalman filter, 3) super–resolved depth frame with $r = 4$ using the proposed method with one–level deblurring step with $[L = 1, K = 25]$ 4) super–resolved depth frame with $r = 4$ using the proposed method with the proposed multi–level deblurring step with $[L = 5, K = 25]$, 5) Error map of comparing the obtained results in forth column with the 3D ground truth.

results using a per–pixel Kalman filtering are shown in the second column where we see how the noise has been attenuated. The results of the proposed algorithm using the one–level deblurring step, with $L = 1$ and $K = 25$, and the multi–level deblurring step, with $L = 5$ and $K = 5$, are plotted in the third and fourth columns, respectively. By visually comparing the obtained results, we find that the proposed algorithm with the multi–level deblurring process provides the best results and hence confirms the quantitative evaluation of Figure 5.6 (blue dashed

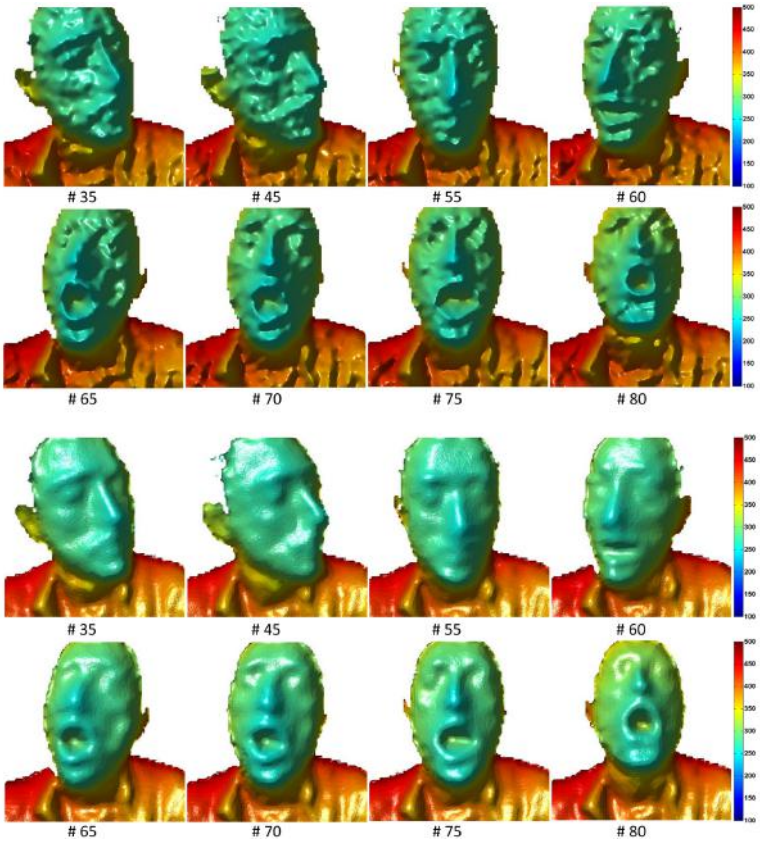


Figure 5.8: Results of applying the proposed algorithm on a real sequence captured by a LR ToF camera (120×160) pixels of a non–rigidly moving face. First and second rows contain a 3D plotting of selected LR captured frames. Third and fourth rows contain the 3D plotting of the super-resolved depth frames with $r = 4$. Distance units on the coloured bar are in mm. Full video available through this link.

line) where it provides the lowest 3D RMSE.

5.4.2 Evaluation on real data

We run the *RecUP-SR* on a different LR real depth sequences captured with a ToF camera (PMD camboard nano with resolution of 120 × 160 [3]). First, we start with a simple scene with one non–rigidly moving face. Then, we show the robustness of *RecUP-SR* to topology changes by testing it on more complex and cluttered scene containing multiple moving objects

Figure 5.9: Results of different super–resolution methods with a scale factor of $r = 4$ applied to a low resolution dynamic depth video captured with a ToF camera with a frame rate of 50 ms per frame. (a) Raw low resolution depth frame. (b) Bicubic interpolation. (c) Patch Based Single Image Super–Resolution (*SISR*) [9]. (d) Upsampling for Precise Super–Resolution (*UP-SR*) [13]. (e) Proposed algorithm. Distance units on the coloured bar are in *mm*.

5.4.2.1 One non–rigidly moving object

We test the proposed algorithm on a real LR depth sequence of a non–rigidly deforming face with large motions and local non–rigid deformations. We super–resolve this sequence using the proposed algorithm with an SR scale factor of $r = 4$. Obtained results are given in 3D in Figure 5.8. They visually show the effectiveness of the proposed algorithm in reducing noise, and further increasing the resolution of the reconstructed 3D face under large non–rigid deformations. Full video of results is available through this link. To visually appreciate these results as compared to state-of-art methods, we tested the bicubic, *UP-SR*, and *SISR* methods on the same LR real depth sequence. Obtained results show the superiority of the *RecUP-SR* as compared to other methods, see Figure 5.9.

In order to show the evolution of the tracking process through the time, we plot the filtered depth value of a randomly chosen tracked pixel Figure 5.12. The blue line shows the filtered trajectory of this pixel as compared to its row noisy measurement in red. Furthermore, we show in Figure 5.10 (b) how the raw radial depth displacement is noisy and ranges from -50 *mm* to -10 *mm* while in fact the real displacement of the face in this frame has to be smooth and homogeneous. By applying the proposed algorithm, the noisy displacement is refined to match the real homogeneous displacement of an approximate value

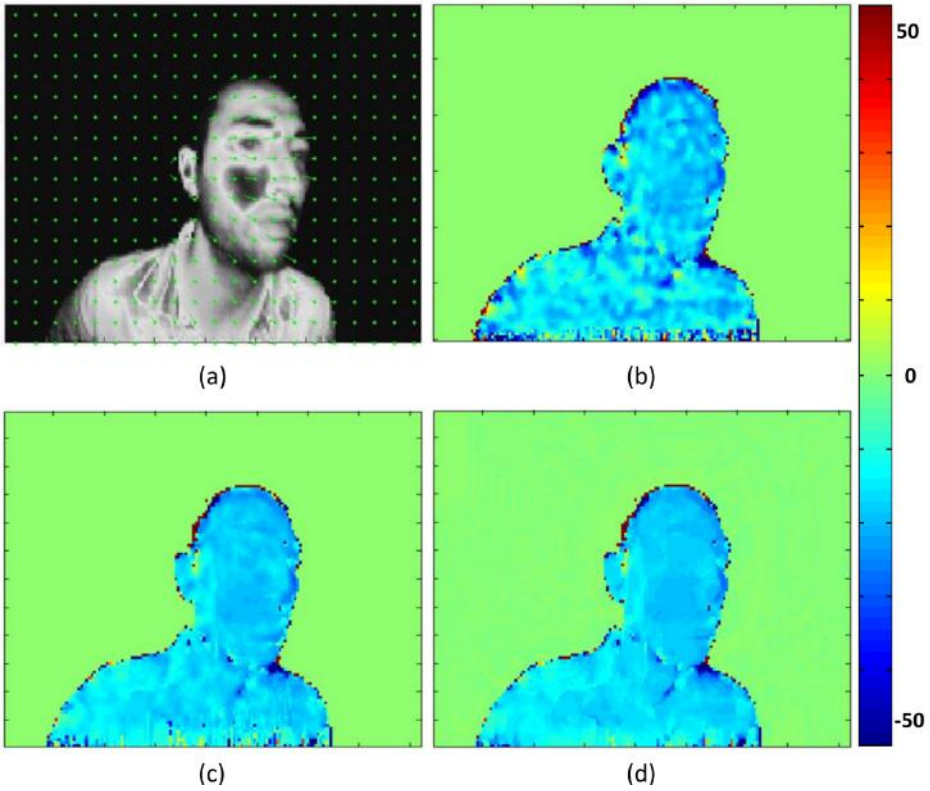


Figure 5.10: Radial depth displacement filtering. (a) 2D optical flow calculated from the normalized amplitude images. (b) Raw noisy depth radial depth displacement. (c) Filtered radial depth displacement using Kalman filter. (d) Filtered radial depth displacement using the proposed method. Unites in the coloured bar are in mm.

of -20 mm, see Figure 5.10 (d). We run another experiment on a second real sequence composed of 120 depth frames of a face moving with long hair causing strong self–occlusions. The goal of this experiment is to show how the tracking process is reinitialized in the self–occlusion case for all pixels representing the self–occluded area. We super–resolve the acquired sequence with a scale factor of $r = 6$. Obtained results are shown in Figure 5.11. It is interesting to see in the third row how the tracking life for each pixel is evolving through the time with stronger occlusions causing shorter tracking. For example, all pixels with the dark red colors in Figure 5.11 (d) have been appeared through the full sequence and no self–occlusion happened and hence the track continues. In contrast, for most

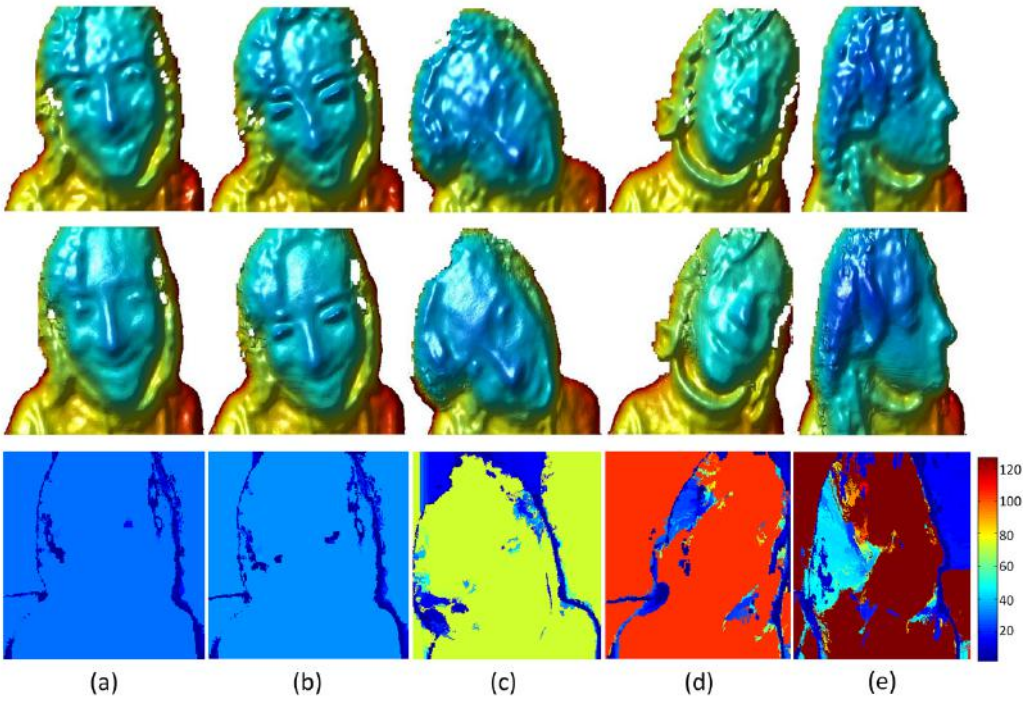


Figure 5.11: Results of applying the proposed algorithm on a real sequence captured by a LR ToF camera (120×160) pixels of a non–rigidly moving face. First and second rows contain a 3D plotting of selected LR captured frames and the 3D plotting of the super-resolved depth frames with $r = 6$, respectively. Third row shows the tracking life for each pixel through the sequence. Units of the coloured bar represents the tracking life (iterations).

of the boundary pixels the tracking process has been reinitialized (blue dark) and thus a spatial median filter is applied for these pixels.

5.4.2.2 Cluttered scene

Finally, we tested *RecUP-SR* on a cluttered scene of moving hands transferring a ball from one hand to another. This scene is quite complex where it contains multiple objects moving with non–rigid deformations, and self–occlusions with one hand passing over the second one. Moreover, the scene contains a challenging case of topology changes represented by hands touching each other and then separating. We note that a strong temporal filtering leads to a longer time for convergence in the case of self–occlusions or non–smooth motions. Similarly,

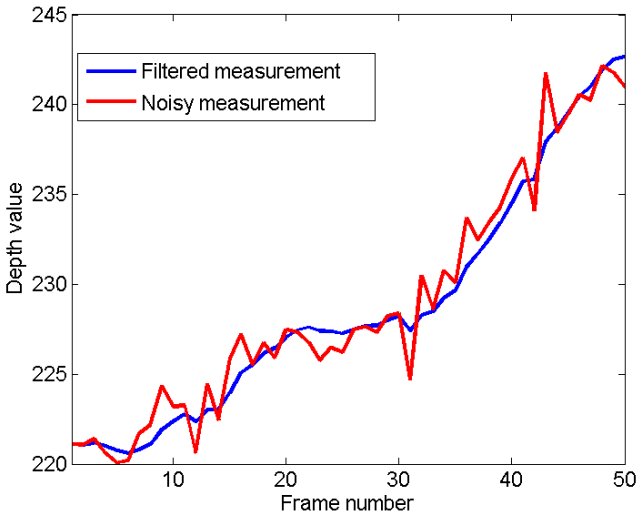


Figure 5.12: Filtered depth value profile of a tracked pixel through the super–resolved sequence of a real face, with SR scale factor of 4.

a strong spatial filtering leads to undesired over-smoothing effects and hence removing the fine details from the final reconstructed HR depth sequence. Thus, in order to handle such a scene, a trade–off between the temporal and spatial filtering has to be achieved. Obtained results in Figure 5.13 show the robustness of the proposed algorithm in handling this kind of scenes. Full video of results is available through this link.

5.4.2.3 Running time

The algorithm's run–time on all sequences acquired using the (PMD camboard nano [3]) with a SR scale factor of $r = 4$ is 50 ms per frame using a 2.2 GHz i7 processor with 4 Gigabyte RAM.

5.5 Conclusion

We have proposed a new algorithm to enhance the quality of low resolution noisy depth videos acquired with cost–effective depth sensors. This algorithm is designed to handle non–rigid deformations thanks to a per–pixel filtering that di-

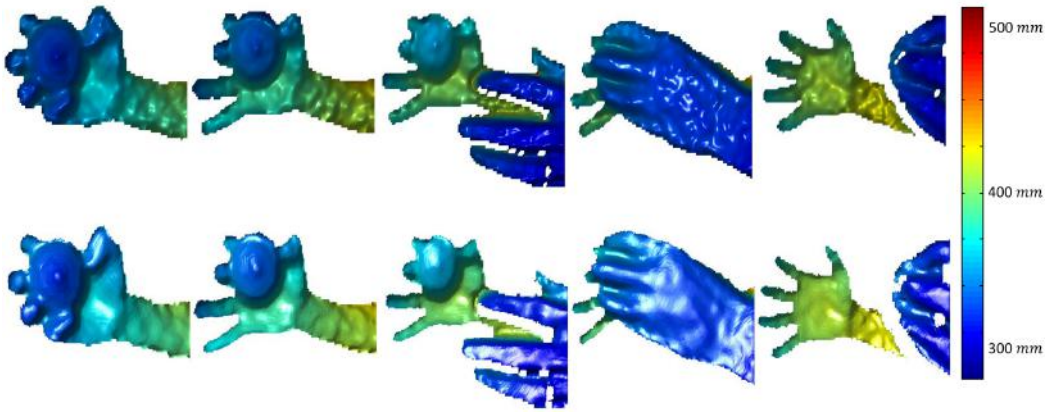


Figure 5.13: Results of applying the proposed algorithm on a real sequence captured by a LR ToF camera (120×160) pixels of a cluttered scene. First row contains a 3D plotting of selected LR captured frames. Second row contains a 3D plotting of the corresponding super–resolved depth frames with $r = 3$. Full video available through this link.

rectly accounts for radial displacements. It is formulated in a dynamic recursive way that allowed a computationally efficient real–time implementation on CPU. Moreover, as compared to state-of-art methods, the processing on depth maps while estimating the local motions in 3D has allowed to maintain a good robustness against topological changes and independence of the number of moving objects in the scene. In order to keep smoothness properties without losing details, each filtered depth frame is further refined using a multi–level iterative bilateral total variation regularization after filtering and before proceeding to the next frame in the sequence. In the case of self–occlusions, the proposed algorithm needs a few number of depth measurements before converging, which is not suitable for some applications. Our future work will focus on increasing robustness to self–occlusions.

Chapter 6

Evaluation of Bilateral Filtering

The well–known bilateral filter is used to smooth noisy images while keeping their edges. This filter is commonly used with Gaussian kernel functions without real justification. The choice of the kernel functions has a major effect on the filter behavior. We propose to use exponential kernels with L_1 distances instead of Gaussian ones. We derive Stein's Unbiased Risk Estimate to find the optimal parameters of the new filter and compare its performance with the conventional one. We show that this new choice of the kernels has a comparable smoothing effect but with sharper edges due to the faster, smoothly decaying kernels.

6.1 Introduction

Image denoising is a common image restoration procedure. The main challenge is to find a good image denoising technique that removes noise while preserving image features such as edges and texture. Over the past three decades, many algorithms have been proposed. One common approach is to use the bilateral filter (BF) [100]. This filter is a weighted average of the local neighborhood pixels. The weighting is based on the product of two kernel functions; one spatial using the distance between the location of the center pixel and the location of the neighboring pixels. The second kernel is radiometric, and uses the distance between the intensity of the center pixel and the intensity of the neighboring pixels. Each weighting kernel is controlled by a parameter determining its width. These kernels are commonly chosen to be Gaussian functions with mean zero.

Stein's Unbiased Risk Estimate (SURE) has been used to find the optimal widths of the Gaussians, i.e., their standard deviations [106], [107], [108]. The objective being to find a trade–off between image smoothing and edge preservation while minimizing SURE risk function, an estimator of the mean square error (MSE) between the noisy image and the filtered one.

As mentioned by Elad [99], as long as the kernel functions used in the BF are smoothly decaying and symmetric, they can be chosen in place of the Gaussian functions. However, very little work exists using bilateral filters with a different kernel. In [7], Farsiu et al. used an exponential kernel in their implementation of the BF, but no justification was given for this choice. It is clear that an adequate choice of the kernels may lead to a good filter performance. We further argue that a faster decaying kernel would ensure sharper edges while smoothing the rest of the image. The question is whether exponential kernels fall under this category. In order to answer this question we compare the performance of the BF using Gaussian kernels, that we refer to as $\mathrm{BF}_{\mathrm{Gauss}}$, and the BF using exponential kernels, that we refer to as $\mathrm{BF}_{\mathrm{exp}}$. We derive the SURE risk function for $\mathrm{BF}_{\mathrm{exp}}$ in order to find the filter optimal parameters. Our simulations show that for different levels of noise, $\mathrm{BF}_{\mathrm{exp}}$ consistently gives a lower or equal MSE and always provides a final image that is visually better. Given that $\mathrm{BF}_{\mathrm{exp}}$ and $\mathrm{BF}_{\mathrm{Gauss}}$ are computationally comparable, in view of our results, $\mathrm{BF}_{\mathrm{exp}}$ is at least similar to $\mathrm{BF}_{\mathrm{Gauss}}$.

6.2 Review of bilateral filtering

Let $\mathbf{x}$ be a noise–free vector image of length n degraded by added zero–mean white Gaussian noise $\mathbf{n}$ of variance σ^2 and of the same size. The observed corrupted image $\mathbf{y}$ is given by

$$\mathbf{y} = \mathbf{x} + \mathbf{n}. \tag{6.1}$$

The BF recovers the original image $\mathbf{x}$ by a nonlinear filtering that replaces the noisy intensity value $\mathbf{y}^{\mathtt{i}}$ at each pixel location $\mathtt{p}^{\mathtt{i}}$ with a weighted average of the

neighboring pixels $\mathtt{p}^\mathtt{j}$, such that:

$$\hat{\mathbf{x}}^\mathtt{i} = \frac{\sum_{\mathtt{p}^\mathtt{j} \in N_\mathtt{i}} f_S(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) f_R(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) \cdot \mathbf{y}^\mathtt{j}}{\sum_{\mathtt{p}^\mathtt{j} \in N_\mathtt{i}} f_S(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) f_R(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j})}, \tag{6.2}$$

where $N_\mathtt{i}$ denotes the neighborhood of the pixel position $\mathtt{p}^\mathtt{i}$. The weighting kernel $f_S(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j})$ is based on the distance between $\mathtt{p}^\mathtt{i}$ and $\mathtt{p}^\mathtt{j}$, and $f_R(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j})$ is based on a radiometric distance, i.e., the difference between the two pixel intensities $\mathbf{y}^\mathtt{i}$ and $\mathbf{y}^\mathtt{j}$. We write the final filtered image as $\hat{\mathbf{x}} = BF(\mathbf{y}, \boldsymbol{\theta})$ with $\hat{\mathbf{x}} = [\hat{\mathbf{x}}^\mathtt{i}]_{\mathtt{i}=1}^n$ and $\boldsymbol{\theta}$ being the vector containing the filter parameters. The two kernels have to verify two properties: 1) symmetry, and 2) smooth decay. Conventionally, these functions are taken as Gaussians with an L_2 norm (Euclidean distance) and parameterized by (λ_g, β_g) . That is $\mathrm{BF}_{\mathrm{Gauss}}$ is defined by:

$$\begin{cases} f_S(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) = \exp\left(-\dfrac{\| \mathtt{p}^\mathtt{i} - \mathtt{p}^\mathtt{j} \|_2^2}{2\lambda_g}\right) \\ f_R(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) = \exp\left(-\dfrac{|\mathbf{y}^\mathtt{i} - \mathbf{y}^\mathtt{j}|^2}{2\beta_g}\right) \end{cases} \tag{6.3}$$

Another choice for the kernels is the exponential function with an L_1 norm (Manhattan distance). The base of the exponential defines the width of the kernel and needs to be in the interval $]0, 1[$ to verify Property 2). The resulting $\mathrm{BF}_{\mathrm{exp}}$ is defined by:

$$\begin{cases} f_S(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) = a_e^{\|\mathtt{p}^\mathtt{i} - \mathtt{p}^\mathtt{j}\|_1} & = \exp\left(\| \mathtt{p}^\mathtt{i} - \mathtt{p}^\mathtt{j} \|_1 \cdot \ln a_e\right) \\ f_R(\mathtt{p}^\mathtt{i}, \mathtt{p}^\mathtt{j}) = b_e^{|\mathbf{y}^\mathtt{i} - \mathbf{y}^\mathtt{j}|} & = \exp\left(|\mathbf{y}^\mathtt{i} - \mathbf{y}^\mathtt{j}| \cdot \ln b_e\right) \end{cases} \tag{6.4}$$

with $0 < a_e, b_e < 1$. For the sake of comparison, we similarly define the bounded parameters of $\mathrm{BF}_{\mathrm{Gauss}}$ as $a_g = e^{-\frac{1}{2\lambda_g}}$, and $b_g = e^{-\frac{1}{2\beta_g}}$. Comparing the two filters $\mathrm{BF}_{\mathrm{Gauss}}$ and $\mathrm{BF}_{\mathrm{exp}}$ passes through comparing the two parameter vectors $\boldsymbol{\theta}_g = [a_g, b_g]^T$, and $\boldsymbol{\theta}_e = [a_e, b_e]^T$. We note that the main difference between the kernels is in the square in the exponent of the Gaussian kernels, that we will see in Section 6.4, plays a role in the difference in performance.

6.3 Parameter estimation for bilateral filtering

The quality of the denoised image $\hat{\mathbf{x}}$ is very dependent on the choice of the filter parameters, $\boldsymbol{\theta}$ in general. To optimally set these parameters, we use SURE as an unbiased estimator of the MSE, obtained from the observed noisy image $\mathbf{y}$. Indeed, the quality of the denoising technique is measured by:

$$\mathrm{MSE}(\hat{\mathbf{x}}) = \frac{1}{n} \parallel \mathbf{x} - \hat{\mathbf{x}} \parallel_2^2 . \tag{6.5}$$

An unbiased estimator of (6.5) is given in [106], and defined as the following SURE risk function:

$$R_{\boldsymbol{\theta}} = \frac{1}{n} \parallel \mathbf{y} - \hat{\mathbf{x}} \parallel_2^2 - \sigma^2 + 2\frac{\sigma^2}{n}\mathrm{div}_{\mathbf{y}}\left(\hat{\mathbf{x}}\right), \tag{6.6}$$

where $\mathrm{div}_{\mathbf{y}}\left(\hat{\mathbf{x}}\right)$ is the divergence of the denoising filter BF (e.g., $\mathrm{BF}_{\mathrm{Gauss}}$ or $\mathrm{BF}_{\mathrm{exp}}$) with respect to the observed image such that:

$$\mathrm{div}_{\mathbf{y}}\left(\hat{\mathbf{x}}\right) = \sum_{\mathtt{i}=1}^{n} \frac{\partial \hat{\mathbf{x}}^{\mathtt{i}}}{\partial \hat{\mathbf{y}}^{\mathtt{i}}}. \tag{6.7}$$

Finding the optimal $\boldsymbol{\theta}$ follows as: $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}}\ \hat{R}_{\boldsymbol{\theta}}$. In practice, the noise variance σ^2 can easily be estimated from the observed data.

In case of $\mathrm{BF}_{\mathrm{Gauss}}$, (6.6) is given in [106]. The divergence term in (6.6) plays a crucial role in the expression of SURE. In [109] we can see that the neighbouring pixels $\mathbf{y}^{\mathtt{j}}$ are considered as a constant with respect to the center pixel $\mathbf{y}^{\mathtt{i}}$. Thus, the divergence of the center of the denoising kernel over the entire image will be zero and the SURE will not give the exact estimation of the MSE, We herein give the derivation for the case of the proposed $\mathrm{BF}_{\mathrm{exp}}$. We first define $f_{SR}(\mathtt{p}^{\mathtt{i}}, \mathtt{p}^{\mathtt{j}}) =$

$a_e^{\|\mathbf{p}^i-\mathbf{p}^j\|_1} b_e^{|\mathbf{y}^i-\mathbf{y}^j|}$, then:

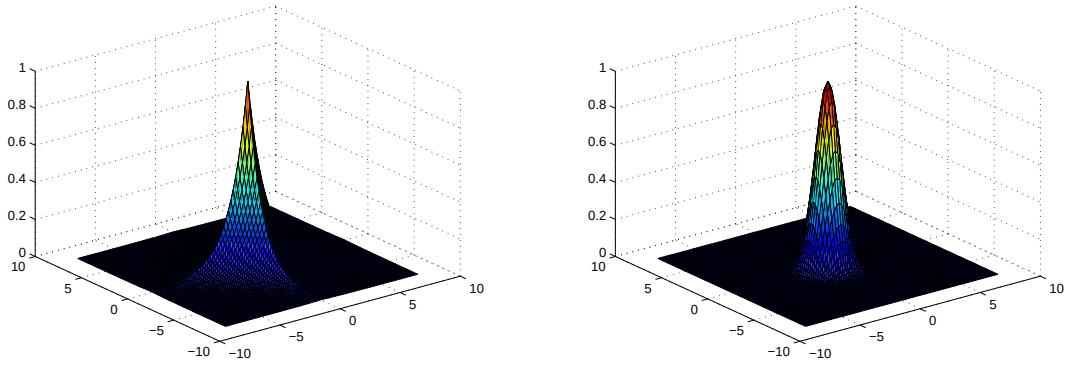
$$\begin{aligned}
\frac{\partial \hat{\mathbf{x}}^i}{\partial \mathbf{y}^i} &= \frac{\partial \left[\dfrac{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)\mathbf{y}^j}{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)} \right]}{\partial \mathbf{y}^i} \\
&= \ln(b_e) \left[\frac{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)\text{sign}(\mathbf{y}^i - \mathbf{y}^j)\mathbf{y}^j}{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)} \right] \\
&- \ln(b_e) \left[\frac{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)\text{sign}(\mathbf{y}^i - \mathbf{y}^j)}{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)} \right] \\
&\times \left[\frac{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)\mathbf{y}^j}{\sum_{\mathbf{p}^j \in N_i} f_{SR}(\mathbf{p}^i, \mathbf{p}^j)} \right],
\end{aligned}$$

where $\text{sign}(\cdot)$ is the sign function. We thus find the optimal $\boldsymbol{\theta}_e$ that ensures the best possible denoising using BF_{exp}. Similarly we find the optimal $\boldsymbol{\theta}_g$ that ensures the best possible denoising using BF_{Gauss}.

6.4 Comparison of the two bilateral filters

Both exponential and Gaussian kernel functions are symmetric and smoothly decaying functions as depicted in Figure 6.1. However, the decay of the exponential kernel is faster which should achieve sharper edges.

BF is about finding a trade–off between the parameters; spatial and radiometric. These parameters, $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_e$, control the kernels decay. Small parameter values give a simple uniform non–adaptive filtering which is known to degrade the image edges, and large values reduce the smoothing effect. As illustrated in Figure 6.2(b), the optimized radiometric parameters, both b_g and b_e, are almost the same for both kernels. On the other hand, the spatial parameters shown in



(a) Exponential kernel (b) Gaussian kernel

Figure 6.1: Exponential and Gaussian kernels.

Figure 6.2(a) of the Gaussian kernel decrease by increasing the noise level compared to the exponential. Thus, the exponential kernel leads to sharper edges Figure 6.5(d) than the Gaussian kernel illustrated in Figure 6.5(c).

6.5 Experimental results

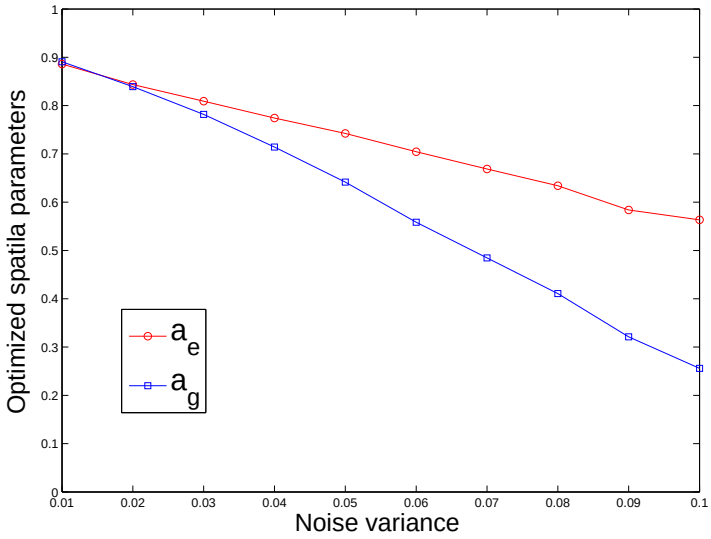
In our experiments we illustrate the performance of the bilateral filter using the proposed kernel compared to the standard Gaussian kernel. First, we run a Monte–Carlo simulation over 50 normalized noisy images by adding white Gaussian noise with a noise variance varying from 1% to 10% corresponding to the range from 10 dB to 20 dB. At each noise level, we denoise the images by a bilateral filter with the proposed kernel and the standard Guassian kernel. The spatial and radiometric smoothing parameters for both kernels were optimized based on the SURE approach. In Figure 6.3, the average RMSE for both kernels is illustrated, where we can see that the proposed $\mathrm{BF}_{\mathrm{exp}}$ performs better than the standard $\mathrm{BF}_{\mathrm{Gauss}}$ for this "Cameraman" example. Moreover, the proposed kernel shows it superiority over the standard Gaussian where it leads to a visual improvement in denoising results as shown in Figure 6.5.

Next, we test our algorithm on a 1D signal by adding a noise with a variance of $\sigma^2 = 5\%$. As illustrated in Figure 6.4, the exponential kernel $\mathrm{BF}_{\mathrm{exp}}$ illustrated

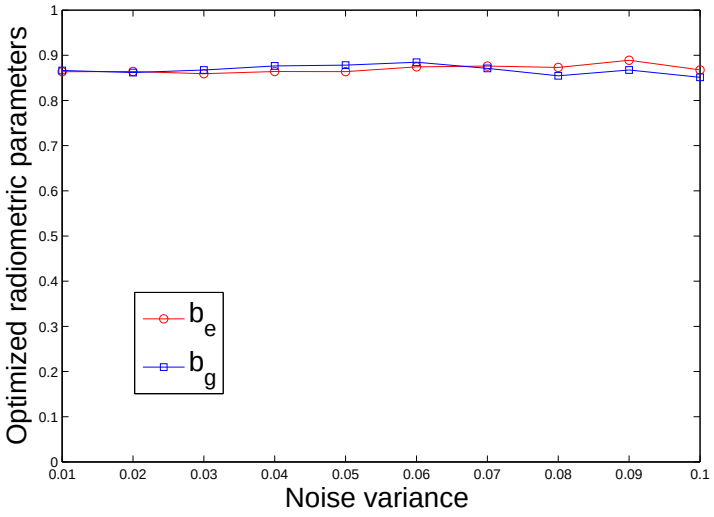
in blue, gives a result that is closer to the original noise–free signal, confirming its better performance.

6.6 Conclusion

Tomasi and Manduchi have proposed the bilateral filter as a noise removal algorithm for images. In this work we have proposed to use the exponential kernel as an alternative to the standard Gaussian commonly used by the community. We verified that the proposed kernel is numerically better than the standard Gaussian for image denoising. Moreover, we showed that the optimum spatial and radiometric parameters provided by the exponential kernel lead to a better trade–off between blurring and denoising, thus suppressing noise while preserving edges.



(a)



(b)

Figure 6.2: Optimized kernel parameters for increasing noise variance (%): (a) spatial, (b) radiometric. Experiment applied on the Cameraman image

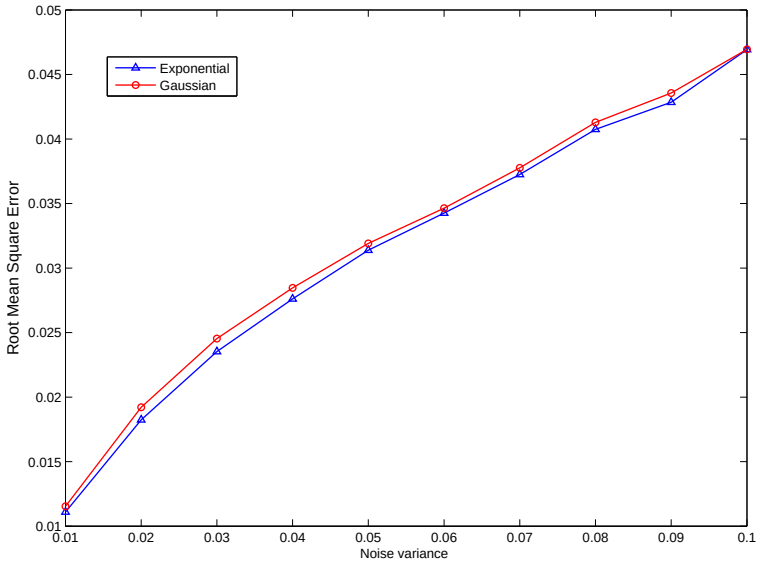


Figure 6.3: RMSE of the bilateral filter using exponential and Gaussian kernels for increasing noise variance (%). Experiment applied on the Cameraman image.

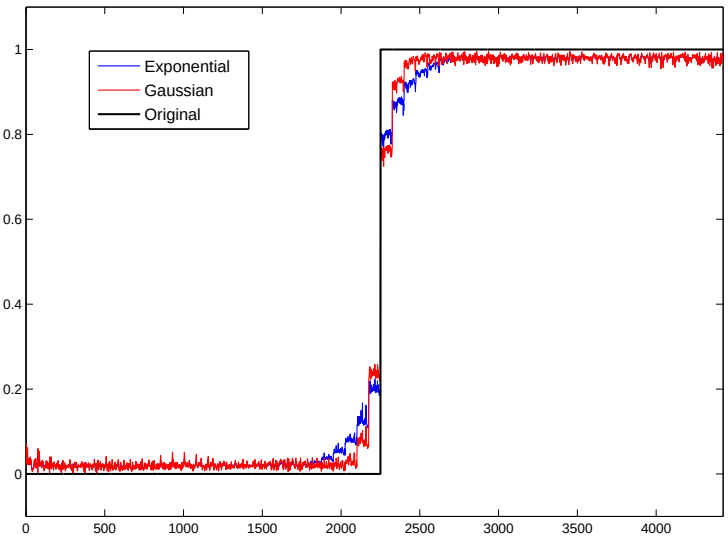


Figure 6.4: Illustration on denoising a 1D signal. See the text for explanation.

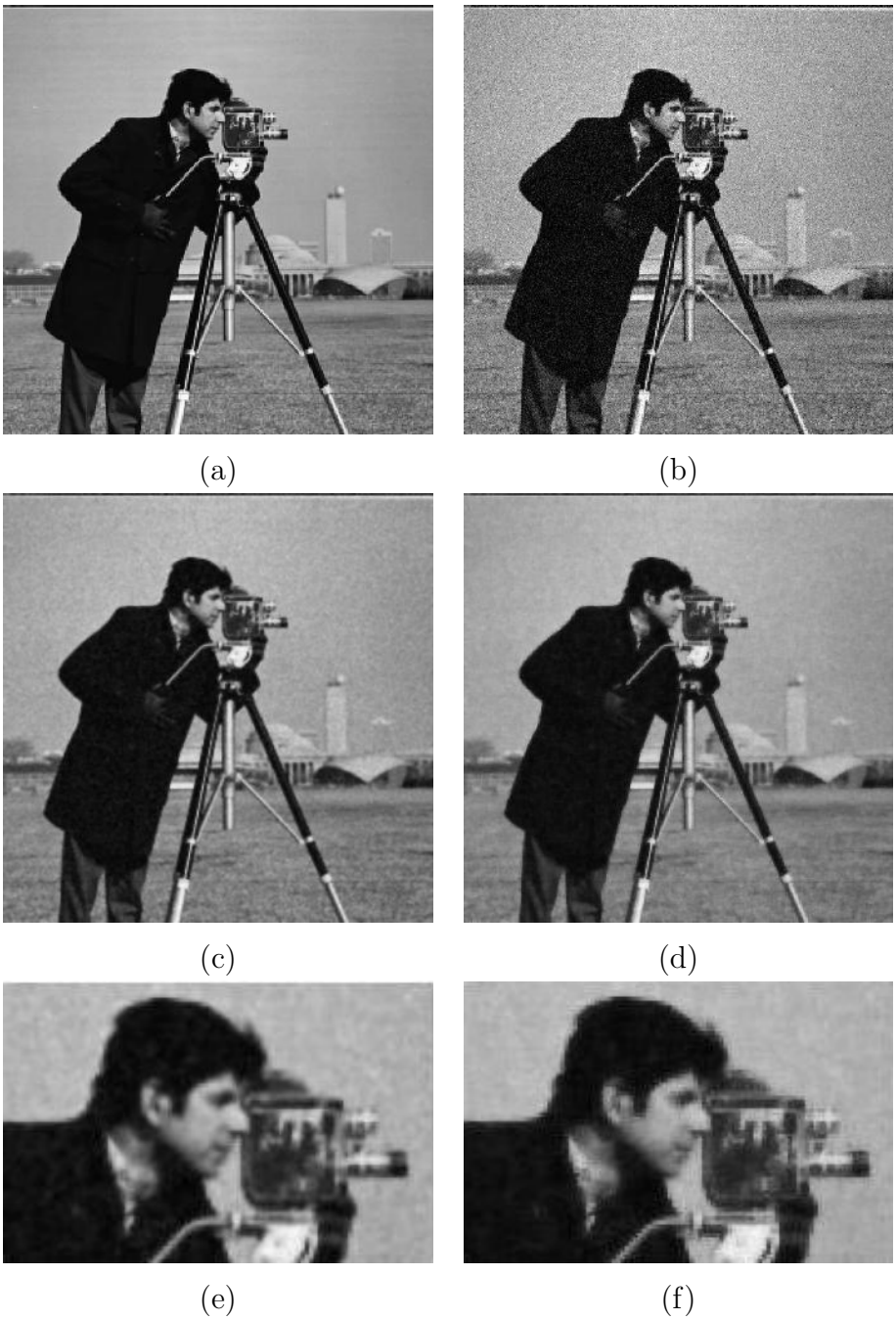


Figure 6.5: Denoising example: (a) original image, (b) noisy image (σ=0.08), (c)-(d) denoised images using Gaussian and exponential kernels, respectively.(e)-(f) zoomed patch for $\mathrm{BF}_{\mathrm{Gauss}}$ and $\mathrm{BF}_{\mathrm{exp}}$, respectively .

Chapter 7

Enhanced 3D Face Reconstruction and Recognition

We address the limitation of low resolution depth cameras in the context of face recognition. Considering a face as a surface in 3D, we reformulate the *UP–SR* algorithm as a new approach on three dimensional points. This reformulation allows an efficient implementation, and leads to a largely enhanced 3D face reconstruction. Moreover, combined with a dedicated face detection and representation pipeline, the proposed method provides an improved face recognition system using low resolution depth cameras. We show experimentally that this system increases the face recognition rate as compared to directly using the low resolution raw data.

7.1 Introduction

In the past ten to fifteen years, research on automatic face recognition has actively moved from 2D to 3D data mostly acquired using HR laser scanners. Multiple approaches have been developed for this kind of data. Until recently the race was about designing sensors to capture data with higher levels of details and higher resolutions [110]. Today much more affordable and less bulky depth cameras, with 3D capabilities, have become accessible. They are, however, of limited resolutions, and present a high level of noise. Some examples are the 3D MLI by IEE of resolution (56×64) [4], and the PMD camboard nano of resolution

(120×165) [3]. Because of their LR and the noisy nature of the acquired data, previously defined 3D face recognition algorithms are no longer ensured to be as effective [14].

The multi–frame super–resolution (SR) framework is an appropriate solution where it becomes possible to recover a higher resolution frame by fusing multiple LR ones. It has been successfully used in the case of 2D face images [111, 112]. Similar efforts have been undertaken for 3D facial data. In [113], a learning–based method has been proposed to directly find the mapping between an LR image and its corresponding HR image without using multiple frames. In [114], Peng et al. proposed to use facial features in a Maximum A Posteriori SR framework.

Depth facial data may also benefit from the SR framework. Recently, Berretti et al. proposed to use SR on facial depth images once back–projected in 3D, and defined the *superfaces* approach [14]. The SR algorithm they deployed is similar in principle to the initial blurred estimate provided in the *eS&A* algorithm proposed in Chapter 3 and extended in Chapter 4 to the dynamic case where the considered multiple realizations were ordered frames constituting a video sequence.This corresponds to the *UP–SR* approach whose key component is a prior upsampling of the observed data which is proven to enhance the registration of frames over time. In addition, it uses a bilateral total variation framework as a smoothness condition. In [115], a similar concept of temporal fusion was considered for 3D facial data enhancement. However, the increase in resolution was induced from temporal data cumulation without a real SR formulation or upsampling. Moreover, smoothness was ensured by bilateral filtering as a post processing operation and not included in the optimization objective function.

The contribution of this chapter is twofold; first, we reformulate *UP-SR* on 3D point clouds constituting the facial surface similarly to the work in [14]. However, by performing the deblurring phase of *UP-SR*, 3D face reconstruction results are maintained, if not enhanced. Second, we show experimentally that using these results for 3D face recognition clearly improves the recognition rate as compared to using raw LR acquisitions. This second contribution requires a full dedicated pipeline for automatic face acquisition from depth cameras. Moreover, level curves equidistant from the nose tip and radially sampled are considered as facial features for matching and comparison.

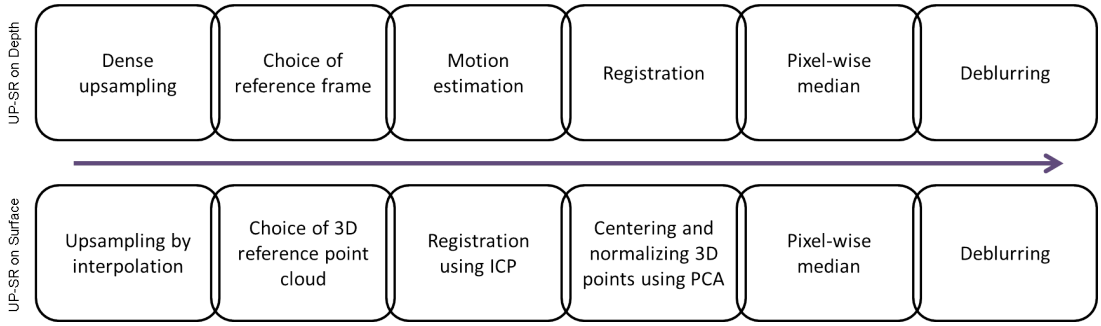


Figure 7.1: *UP-SR* steps on depth data and on a surface in 3D.

7.2 Background

In what follows, we review the *UP-SR* algorithm. We represent all images in lexicographic vector form. Let us consider an HR depth image $\mathbf{f}$ of size n, and N observed LR images $\mathbf{g}_k$, $k = 0, ..., (N-1)$, of size m, such that $n = r \cdot m$, where r is the SR factor. Every frame $\mathbf{g}_k$ is an LR noisy and deformed realization of $\mathbf{f}$ modeled as follows:

$$\mathbf{g}_k = \mathbf{DHM}_k\mathbf{f} + \mathbf{n}_k, \qquad k = 0, ..., (N-1), \tag{7.1}$$

where $\mathbf{M}_k$ is an $(n \times n)$ invertible matrix corresponding to the geometric motion between $\mathbf{f}$ and $\mathbf{g}_k$. We assume that $\mathbf{g}_0$ is the reference frame for which $\mathbf{M}_0 = \mathbf{I}_n$. The point spread function of the depth camera is modeled by the $(n \times n)$ space and time invariant blurring matrix $\mathbf{H}$. The matrix $\mathbf{D}$ of dimension $(m \times n)$ represents the downsampling operator, and the vector $\mathbf{n}_k$ is an additive noise at k which follows a white multivariate Laplace distribution of mean zero and covariance $\mathbf{\Sigma} = \sigma^2\mathbf{I}_m$, with $\mathbf{I}_m$ being the identity matrix of size $(m \times m)$.
One of the key components of *UP-SR* is to upsample the observed LR images prior to any operation. We define the resulting r–times upsampled image as:

$$\mathbf{g}_k \uparrow = \mathbf{U} \cdot \mathbf{g}_k, \tag{7.2}$$

where $\mathbf{U}$ is an $(n \times m)$ upsampling matrix. This allows to directly solve the problem of undefined pixels in the SR initialization phase. It also leads to a more accurate and robust estimation of the motion $\hat{\mathbf{M}}_k$ as it is now computed

between $\mathbf{g}_k \uparrow$ and $\mathbf{g}_0 \uparrow$. The following registration of frames to the reference is consequently enhanced:

$$\overline{\mathbf{g}}_k \uparrow = \hat{\mathbf{M}}_k^{-1} \mathbf{g}_k \uparrow . \tag{7.3}$$

Without loss of generality, both $\mathbf{H}$ and $\mathbf{M}_k$ are assumed to be block circulant matrices. Choosing the upsampling matrix $\mathbf{U}$ to be the transpose of $\mathbf{D}$, the product $\mathbf{UD} = \mathbf{A}$ defines a new block circulant blurring matrix $\mathbf{B} = \mathbf{AH}$. We have, therefore, $\mathbf{BM}_k = \mathbf{M}_k\mathbf{B}$. As a result, the estimation of $\mathbf{f}$ may be decomposed into two steps; estimation of a blurred HR image $\mathbf{z} = \mathbf{Bf}$, followed by a deblurring step. The data model in (7.1) becomes

$$\overline{\mathbf{g}}_k \uparrow = \mathbf{z} + \boldsymbol{\nu}_k, \qquad k = 0, ..., (N-1), \tag{7.4}$$

where $\boldsymbol{\nu}_k = \hat{\mathbf{M}}_k^{-1}\mathbf{U} \cdot \mathbf{n}_k$ is an additive noise vector of length n. Using an L_1–norm $\| \cdot \|_1$, the estimate of $\mathbf{z}$ using the corresponding Maximum Likelihood is

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \sum_{k=0}^{N-1} \| \mathbf{z} - \overline{\mathbf{g}}_k \uparrow \|_1. \tag{7.5}$$

The result in (7.5) is, by definition, the pixel–wise temporal median estimator $\hat{\mathbf{z}} = \text{med}_k\{\overline{\mathbf{g}}_k \uparrow\}$.

To recover $\hat{\mathbf{f}}$ from $\hat{\mathbf{z}}$, an iterative optimization is performed as a deblurring step. Considering a regularization term $\Gamma(\mathbf{f})$, chosen to be the bilateral total variation (BTV) given in [7], we find

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\text{argmin}} \Big(\|\mathbf{Bf} - \hat{\mathbf{z}}\|_1 + \lambda\Gamma(\mathbf{f}) \Big), \tag{7.6}$$

where λ is the regularization parameter. The *UP-SR* algorithm is given in Algorithm 4.1 and its pipeline is given in Figure 7.1.

7.3 Surface upsampling for precise super–resolution

The different steps in *UP-SR* as described in Section 7.2 may be directly applied on LR depth images $\mathbf{g}_k$ of faces as those illustrated in Figure 7.2 (a). The resulting

reconstructed face $\mathbf{f}$ is shown in Figure 7.2 (c). A dedicated software tool has been developed for automatic HR 3D face reconstruction using an LR depth frames acquired with an LR ToF camera, see Appendix B. While it is of higher resolution, it presents artifacts that we argue are caused by applying *UP-SR* on gridded depth data. To remedy these artifacts, we propose in what follows to back–project the $\mathbf{g}_k$ frames, $k = 0, \cdots, N-1$, to $\mathbb{R}^3$ using the intrinsic parameters of the camera used for the acquisition. We end up with N corresponding point clouds $\mathcal{G}_k = \{\mathtt{p}_k^{\mathtt{i}} = (x_k^{\mathtt{i}}, y_k^{\mathtt{i}}, z_k^{\mathtt{i}}) \in \mathbb{R}^3, \mathtt{i} = 1, \cdots, m\}$ as shown in Figure 7.2 (b).

The objective is now to reconstruct an HR point cloud $\mathcal{F} = \{\mathbf{q}_k^{\mathtt{i}} = (x_k^{\mathtt{i}}, y_k^{\mathtt{i}}, z_k^{\mathtt{i}}) \in \mathbb{R}^3, \mathtt{i} = 1, \cdots, n\}$ belonging to the surface $\mathcal{S}$ of the original face, i.e., $\mathcal{F} \subset \mathcal{S}$.

We adapt the algorithm in Algorithm 4.1 to point clouds, and define a modified version of the *UP-SR* algorithm that we refer to as *SurfUP-SR*.
The two main phases are maintained: 1) estimation of $\mathcal{Z}$, a blurred version of $\mathcal{X}$; 2) deblurring by optimization as in (7.6). The steps of upsamling and registration need to be adapted as described in the following sections. An illustration of differences between *UP-SR* and *SurfUP-SR* is given in Figure 7.1.

7.3.1 Surface upsampling

Assuming that the point cloud $\mathcal{G}_k$ is a sampling of a surface $\mathcal{S}_k$, the upsampling of $\mathcal{G}_k$ may be reformulated as a problem of interpolating the surface $\mathcal{S}_k$ from scattered points. The surface $\mathcal{S}_k$ may be defined implicitly by a function f as: $f(x, y, z) = 0, \quad \forall\, \mathtt{p} = (x, y, z) \in \mathcal{S}_k$, or equivalently by using the interpolant $\mathcal{P}_f$ as:

$$\mathcal{P}_f\,(x, y) = z, \qquad \forall\, \mathtt{p} = (x, y, z) \in \mathcal{S}_k. \tag{7.7}$$

The m points in $\mathcal{G}_k$ verify (7.7), hence they form a system of m equations, from which $\mathcal{P}_f$ may be defined. A solution using kernel regression has been proposed in [87]. An efficient GPU implementation has been given in [116]. We used the Matlab `scatteredInterpolant` function in our implementation. Once $\mathcal{P}_f$ is found, it is used to define $(r-1)\cdot m$ additional points belonging to $\mathcal{S}_k$ for chosen (x, y)–positions. As a result, a denser point cloud $\mathcal{G}_k \uparrow$ containing a total of n

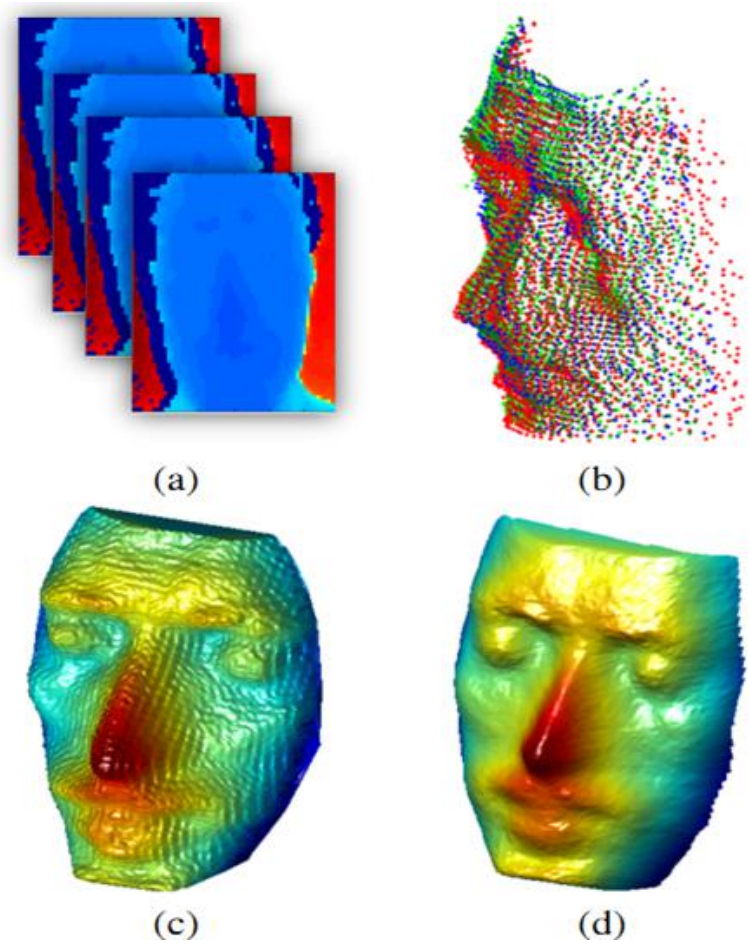


Figure 7.2: Face reconstruction with *UP-SR* using (a) depth images, (b) point clouds. The corresponding results are shown in (c) and (d), respectively.

points is found such that

$$\mathcal{G}_k \uparrow = \mathcal{G}_k \cup \{\mathtt{p}_k^{\mathtt{i}} = (x_k^{\mathtt{i}}, y_k^{\mathtt{i}}, z_k^{\mathtt{i}}) \in \mathbb{R}^3, \mathtt{i} = m+1, \cdots, n\}, \tag{7.8}$$

and $(x_k^{\mathtt{i}}, y_k^{\mathtt{i}}) \in [-1, 1] \times [-1, 1]$.

7.3.2 Surface registration

The motion estimation and registration steps in *UP-SR* are replaced by directly using classical 3D point cloud registration techniques. We use iterative closest points (ICP) to rigidly register each point cloud $\mathcal{G}_k \uparrow$ to the reference $\mathcal{G}_0 \uparrow$. This is

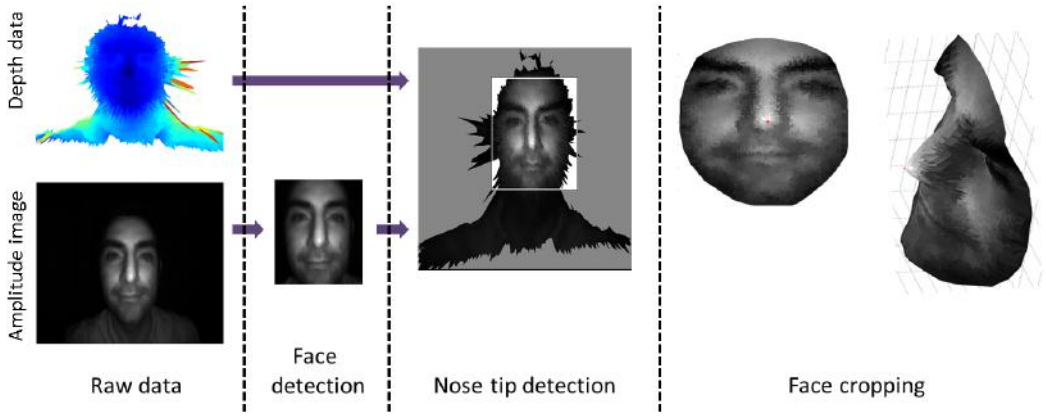


Figure 7.3: Preprocessing step of the facial acquisition pipeline using a depth camera.

done by estimating the optimal transformation parameters, namely, 3D rotation $\hat{\mathbf{R}}_k$, translation $\hat{\mathbf{t}}_k$, and global scaling factor $\hat{\alpha}_k$ that minimize the distance $Err(\cdot)$ between the transformed and the reference point clouds such that

$$[\hat{\mathbf{R}}_k, \hat{\mathbf{t}}_k, \hat{\alpha}_k] = \underset{\mathbf{R},\mathbf{t},\alpha}{\operatorname{argmin}} Err\left(\alpha \mathbf{R} \mathcal{G}_k \uparrow + \mathbf{t}, \mathcal{G}_0 \uparrow\right). \tag{7.9}$$

The registered point cloud $\overline{\mathcal{G}}_k \uparrow$ is then computed as:

$$\overline{\mathcal{G}}_k \uparrow = \hat{\alpha}_k \hat{\mathbf{R}}_k \mathcal{G}_k \uparrow + \hat{\mathbf{t}}_k. \tag{7.10}$$

With these modifications, the new *SurfUP-SR* algorithm is given in Algorithm 7.1. Its visual impact is shown in the example of Figure 7.2 (d).

7.4 Proposed face recognition pipeline

Our proposed pipeline is composed of three main stages: preprocessing of raw data, feature extraction and matching.

7.4.1 Preprocessing

The preprocessing step is an essential step in the design of a face recognition system as it affects the performance of the system significantly. We implement

Algorithm 7.1 ***SurfUP-SR*:** Surface Upsampling for Precise Super–Resolution.

1. Choose the reference frame $\mathcal{G}_0$.

for k, s.t., $k = 1, \cdots, N$,

do

2. Compute $\mathcal{G}_k \uparrow$ using (7.8).
3. Estimate $\hat{\mathbf{R}}_k$, $\hat{\mathbf{t}}_k$, and $\hat{\alpha}_k$ using ICP as in (7.9).
4. Compute $\overline{\mathcal{G}}_k \uparrow$ using (7.10).

end do

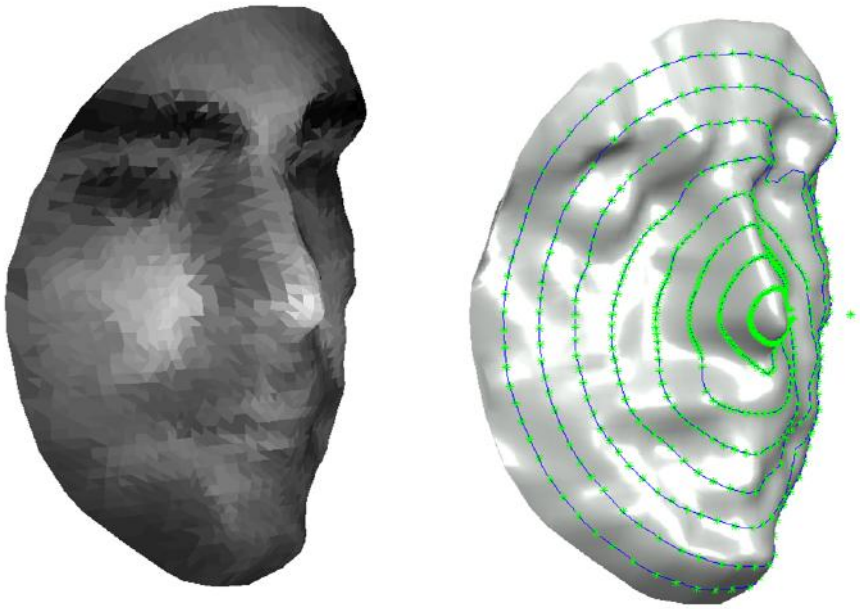
5. Find $\hat{\mathcal{Z}}$ by applying a median estimator (7.5).
6. Deduce $\hat{\mathcal{F}}$ by deblurring using (7.6).

end for

fast and efficient techniques to detect the face region and the nose tip for an effective segmentation and alignment. We apply a face detection algorithm on the amplitude or 2D image only, then we map the face region with the corresponding depth image to obtain the corresponding 3D facial region. In this work, the Viola–Jones [117] face detection algorithm is used for its computational efficiency and high detection rate. Once we detect the depth face region, we detect the nose tip represented by the point with the smallest depth value. The nose tip is used as a basic feature for our segmentation and alignment. Using a spherical cropping centered at the nose tip, we discard the ear, hair and part of the neck areas. Finally, the ICP registration is used for alignment.

7.4.2 Feature extraction

We use spherical curves and their radial discretization as features to represent each face. A spherical curve is obtained by intersecting the facial surface with a sphere. In order to have smoother and continuous curves, we apply the interpolation technique proposed in [118]. Spherical curves are discretized radially by slicing the spherical intersection curves using a plane that is parallel to the face normal and that intersects the spherical curves radially at uniform angles. Each face is represented by an indexed collection of $M \times L$ points in 3D, where M denotes the number of curves per sample face and L is the number of points in



(a) (b)

Figure 7.4: Feature extraction step using: (a) Observed LR 3D face with texture from amplitude or 2D images. (b) Extracted level curves.

each curve. We end up with a feature vector of size $M \times L \times 3$ for each face. An example of the extracted feature curves is shown in Figure 7.4.

7.4.3 Matching

The matching step aims to associate each probe 3D face to the the closest 3D face in the database by comparing their extracted features. The comparison is carried out by an appropriate distance measure on the space of the extracted feature curves. We choose the cosine distance in our experiments as we found it to be the best performing one. This is confirmed by the survey of Smeets et al. [119].

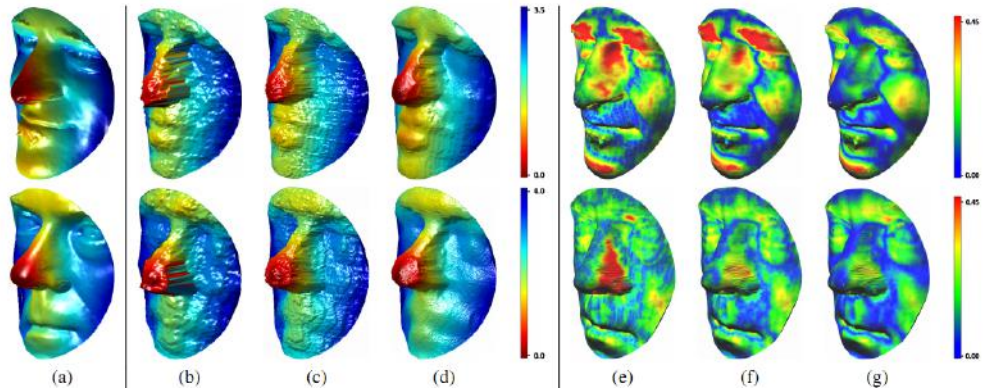


Figure 7.5: 3D face reconstruction results. (a) 3D laser scan ground truth. (b) One of the LR 3D faces. (c) Results of the *superfaces* algorithm. (d) Results of the proposed *SurfUP-SR* algorithm. (e) 3D error map corresponding to the 3D LR face. (f) 3D error map corresponding to the *superfaces* results. (g) 3D error map corresponding to the proposed *SurfUP-SR.*

.

7.5 Experimental part

We evaluate the performance of the proposed system for both 3D face reconstruction and recognition. First, to evaluate the quality of the reconstructed 3D faces, we use the publicly available *superfaces* dataset [120]. It has been acquired using the well known Kinect camera [35]. A sequence of 2D and depth images for 20 different subjects are provided. Moreover, an HR scanned version for each subject is available as ground truth. The dataset has only one realization for each subject which makes it not appropriate for recognition purposes. Thus, we built our real dataset using 10 subjects with two different realizations for each subject. The dataset is acquired using the PMD camboard nano time of flight camera with a resolution of (120×165) pixels [3].

7.5.1 Reconstruction

In order to evaluate the quality of the reconstructed faces, we use the above mentioned real dataset [120]. The faces in the depth frames are of low resolution due to the object distance from the camera. To improve its quality, we conduct

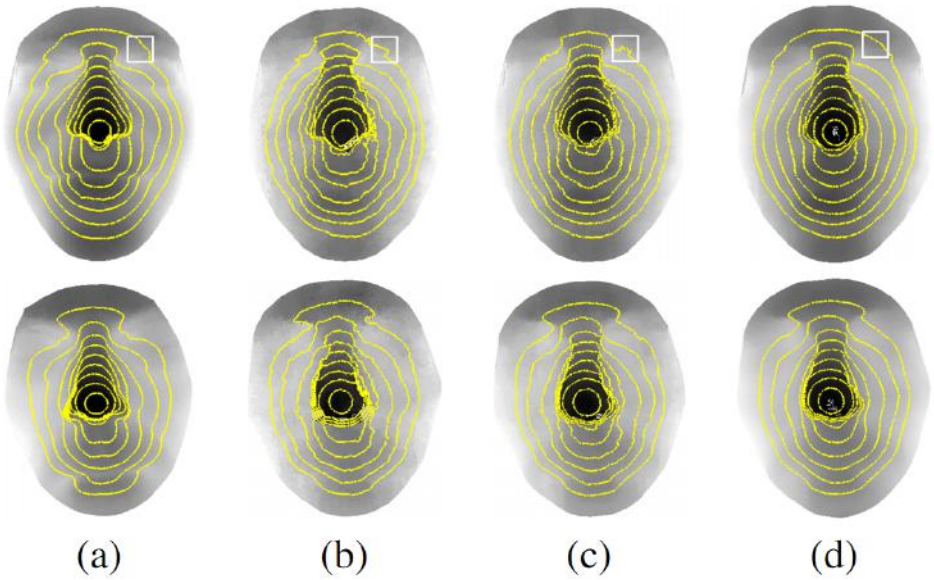
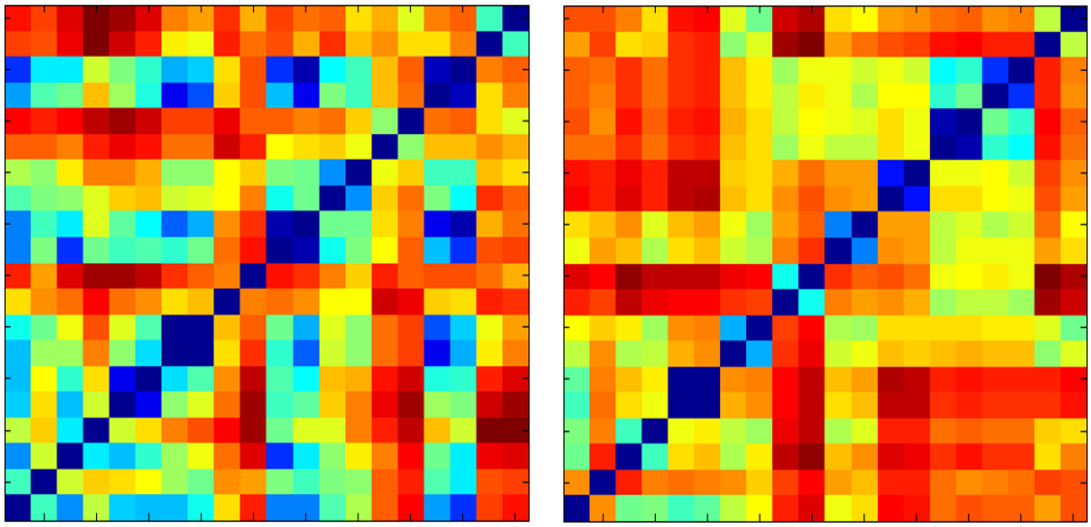


Figure 7.6: Extracted level curves from 3D faces for: (a) Ground truth. (b) LR. (c) *superfaces*. (d) *SurfUP-SR*.

the following test. We apply *SurfUP-SR*, and show the results for two subjects (01 and 19) using 5 LR frames. An LR frame for each subject is shown in Figure 7.5.(b), first and second rows, respectively. Obtained results show that the proposed algorithm provides a visually improved HR 3D faces as seen in Fig. 7.5.(d) as compared to the LR captured data Figure 7.5.(b). Moreover, our algorithm provides better visual results than the recently proposed *superfaces* algorithm [14], Figure 7.5. (c). This is due to the fact that *SurfUP-SR* includes an additional deblurring step. Our results are of sufficient quality for many applications such as 3D face recognition. In order to provide a quantitative evaluation, we measure the reconstruction error of *SurfUP-SR* and *superfaces* against the laser scanned ground truth. In Figure 7.5. (f) and (g), we may see the color–coded reconstruction error of the *superfaces* method [14] and *SurfUP-SR*, respectively. As expected, obtained results show that *SurfUP-SR* is at least as good as *superfaces* and sometimes better. Moreover, by taking a look to the error range bar in Figure 7.5, we note that in most areas the errors are below 0.5 cm.



(a) (b)

Figure 7.7: Confusion matrices. (a) Using the LR 3D observed faces. (b) Using the super–resolved 3D faces by the proposed *SurfUP-SR*.

7.5.2 Recognition

In order to test the impact of *SurfUP-SR* on a face recognition algorithm, we evaluate the performance of the pipeline presented in Section on the raw LR faces in our database. We then run the same pipeline on the super–resolved faces of our database. We may see in Figure 7.6 the enhancement incurred by *SurfUP-SR* on the quality of the extracted feature curves. Indeed, their extraction from LR faces leads to noisy curves. For the same subject, these curves become smoother and less noisy if extracted from super–resolved data. The quality of these curves directly affects the final result of the face recognition algorithm. The corresponding confusion matrices are given in Figure 7.7(a) and in Figure 7.7(b). We notice an improved recognition rate from 50% to 80% when super-resolving. This confirms the importance of having a higher resolution for an increased recognition rate and the effectiveness of the proposed *SurfUP-SR*.

7.6 Conclusion

In this chapter we proposed a new multi–frame super–resolution algorithm *SurfUP-SR* which improves 3D face recognition rate using low resolution, and cost–effective depth cameras. We reformulated the *UP-SR* algorithm on a 3D point cloud instead of its original formulation on a depth image. In addition, we provided a full automatic 3D face acquisition from depth cameras. Experimental evaluation of *SurfUP-SR* using a real low resolution 3D face dataset has been carried out. Obtained results show an efficient enhancement in the resolution and the quality of the captured low resolution 3D faces. Moreover, we showed the impact of the proposed algorithm in decreasing the 3D reconstruction error, and most importantly in increasing the 3D face recognition rate.

7. ENHANCED 3D FACE RECONSTRUCTION AND RECOGNITION

Chapter 8

Conclusions

In this thesis, we proposed a general depth multi–frame super–resolution framework that addresses the limitations of state-of-art depth enhancement approaches. The proposed framework does not need any additional hardware or coupling with different modalities. It is based on reformulating the classical super–resolution problem into an image denoising one by assimilating the effect of downsampling to a blur effect once multiplied by its reverse operation, that is, by upsampling. This has resulted in a robust median initial estimate, further refined by a deblurring operation using a bilateral total variation as the regularization term. Furthermore, we showed that the upsampling operation ensures a systematic improvement in the registration accuracy. It is chosen to be a dense nearest neighbor upsampling in the case of depth data.

This upsampling property has been explored in three cases, which has led to three algorithms. The considered cases can be described based on the motions involved in the video.

First, in the case of relative global lateral motions, the considered video or set of images describe a static depth scene. For this case, we have proposed the ***eS&A*** algorithm, presented in Chapter 3, which was our first attempt in using the upsampling property. In addition to increasing the registration accuracy with respect to the reference frame, *eS&A* naturally solves the problem of undefined pixels inherent to classical SR by a non-zero initialization of the estimate of the HR depth frame. While the original *S&A* algorithm [23] ensures robust reconstruction of static scenes, and was successfully extended to scanning of static

objects [25], it has been made more practical with the proposed modifications. Indeed, the *eS&A* method ensures at least the same performance as *S&A* with a lower number of observations.

Second, the case of local lateral motions was considered. This corresponds to dynamic scenes with objects non-rigidly deforming with a deformation assumed to be parallel to the image plane. For this case, we have proposed the ***UP-SR*** algorithm, described and analyzed in Chapter 4. The robustness of this algorithm to local deformations is obtained through the combination of the upsampling property with a cumulative motion estimation. The order among the observed frames becomes crucial in this case and helps in processing videos with relative large deformations between the first and last frame. This is as long as the motion between consecutive frames is small enough in order to verify the new data model in (4.13) with the combined blurring operator. Moreover, we noted that the *UP-SR* algorithm may present errors when different objects are touching over multiple frames, specifically more than half of the total number of considered frames in one sequence. Indeed, two objects may be wrongly assigned to the same object.

Third, in the more general case of local lateral and radial motions, objects can deform non-rigidly in full 3D. Consequently, the dense lateral motions estimated in *UP-SR* with optical flow had to be extended to dense range flow. The algorithm ***RecUP-SR*** was introduced as a solution in Chapter 5. It was designed with keeping its practicality in mind; thus, favoring an approximation of range flow by decomposing it into lateral motions and radial ones. Furthermore, this algorithm was formulated in a dynamic recursive way that allowed a computationally efficient real–time implementation on CPU. As compared to state-of-the-art methods, the processing on depth maps while approximating local 3D motions has allowed to maintain a good robustness against topological changes and independence of the number of moving objects in the scene. This property is a clear advantage over most recent methods that explicitly compute a flow in 3D and apply a processing on meshed point clouds [33, 36].

Supported by the experimental results on both synthetic and real data, we believe that the proposed SR framework opens new possibilities for computer vision applications using cost-effective depth sensors in dynamic scenarios with non-rigid motions.

We have looked into 3D face recognition as one possible application. We have developed a tool based on *UP–SR* for automatic 3D face reconstruction as presented in Appendix B. Furthermore, we have adapted *UP–SR* for a more robust and accurate reconstruction resulting in the ***SurfUP-SR*** algorithm presented in Chapter 7. It is based on reformulating *UP-SR* on a 3-D point cloud instead of its original formulation on a depth image. We showed that this improvement has an impressive positive impact on the final recognition rate of 3D faces captured with a cost-effective depth camera.

The thesis has addressed an important and timely problem in computer vision. The current results have triggered new research questions as described below:

- **Considering full 3D motions:** In our work we have progressively moved from considering global lateral motions, to local lateral motions, and finally to considering local lateral and filtered radial motions. This last model ensured a new level of robustness in handling objects with non-rigid deformations. One would expect to reach a further improved performance in case the true 3D motion is incorporated in the proposed SR. We have started investigating this direction by proposing a first tentative in [33] by defining the *KinectDeform* algorithm. More research is still required to have a practical version of *KinectDeform* that can work in real-time and that can handle sparse and noisy observations.

- **Considering extra prior information:** In the current work, we have started from scratch in studying dynamic depth videos and their enhancement. We therefore naturally started from relative simple models and assumptions on data properties and motion models. While current results are promissing, we see further potential enhancements, concretely in considering extra information as prior to be incorporated by regularization. One option is to consider extending the BTV regularization to a multi-modal one, where intensity is fused with depth data. We see this in the same spirit as the join bilateral upsampling concept already tested for depth data enhancement but this time it would be by deriving its total variational version. Another way to incorporate extra contextual information would be to consider learning combined with filtering, which would lead

to a combination of a single image SR framework with a multi-frame SR framework. This is expected to bring robustness to the challenging cases of large self-occlusions.

- **Exploiting results for pattern recognition:** In view of the current results, and after testing their usefulness for at least one application, i.e., 3D face recognition, we foresee a large line of work where cost–effective depth cameras can be deployed in many pattern recognition–based applications. We name, for example, gesture control, face expression recognition, and action recognition. In these applications the dynamic depth videos represent the first source of information. We note that in addition to benefiting from the enhanced geometrical features, we expect further exploitation of the enhanced motions in 2.5D.

Appendix A

Proof of the Cumulative Motion Estimation

We prove by induction the following $\zeta(n)$ statement:

$$\begin{cases} \mathbf{M}_{t_0-n}^{t_0}\mathbf{g}_{t_0-n}\uparrow &= \overline{\mathbf{g}}_{t_0-n}^{t_0}\uparrow, \\ \text{s.t. } \mathbf{M}_{t_0-n}^{t_0} &= \mathbf{M}_{t_0-1}^{t_0}\mathbf{M}_{t_0-2}^{t_0-1}...\mathbf{M}_{t_0-n}^{t_0-n+1} \end{cases} \cdots \zeta(n).$$

Let us consider that $\zeta(n-1)$ is true, i.e.

$$\begin{cases} \mathbf{M}_{t_0-(n-1)}^{t_0}\mathbf{g}_{t_0-(n-1)}\uparrow &= \overline{\mathbf{g}}_{t_0-(n-1)}^{t_0}\uparrow, \\ \text{s.t. } \mathbf{M}_{t_0-(n-1)}^{t_0} &= \mathbf{M}_{t_0-1}^{t_0}\mathbf{M}_{t_0-2}^{t_0-1}...\mathbf{M}_{t_0-(n-1)}^{t_0-(n-1)+1} \end{cases} \tag{A.1}$$

From (A.1) we have:

$$\mathbf{M}_{t_0-(n-1)}^{t_0}\mathbf{M}_{t_0-n}^{t_0-(n-1)} = \mathbf{M}_{t_0-n}^{t_0} \tag{A.2}$$

Base case: When $n = 1$ we have

$$\mathbf{M}_{t_0}^{t_0}\mathbf{g}_{t_0}\uparrow = \overline{\mathbf{g}}_{t_0}^{t_0}\uparrow, \tag{A.3}$$

and

$$\mathbf{M}_{t_0}^{t_0}\mathbf{M}_{t_0}^{t_0-1} = \mathbf{M}_{t_0}^{t_0-1}. \tag{A.4}$$

Both (A.3) and (A.4) are verified because $\mathbf{M}_{t_0}^{t_0} = I_n$. Then,
Induction step: We need to show that $\zeta(n-1) \Rightarrow \zeta(n)$.
Given two consecutive frames: $\mathbf{y}_{t_0-n}$ and $\mathbf{y}_{t_0-(n-1)}$, we have:

$$\mathbf{M}_{t_0-n}^{t_0-(n-1)}\mathbf{g}_{t_0-n}\uparrow = \overline{\mathbf{g}}_{t_0-n}^{t_0-(n-1)}\uparrow, \tag{A.5}$$

where

$$\hat{\mathbf{M}}_{t_0-n}^{t_0-(n-1)} = \arg\min_{\mathbf{M}} \Psi\left(\mathbf{g}_{t_0-(n-1)} \uparrow, \mathbf{g}_{t_0-n} \uparrow, \mathbf{M}\right). \tag{A.6}$$

Multiplying (A.5) by $\mathbf{M}_{t_0-(n-1)}^{t_0}$ we find

$$\mathbf{M}_{t_0-(n-1)}^{t_0}\mathbf{M}_{t_0-n}^{t_0-(n-1)}\mathbf{g}_{t_0-n} \uparrow = \mathbf{M}_{t_0-(n-1)}^{t_0}\overline{\mathbf{g}}_{t_0-n}^{t_0-(n-1)} \uparrow . \tag{A.7}$$

From (A.2) and (A.7) we have

$$\mathbf{M}_{t_0-n}^{t_0}\mathbf{g}_{t_0-n} \uparrow = \overline{\mathbf{g}}_{t_0-n}^{t_0} \uparrow .$$

Appendix B

Tool for Automatic 3D Face Reconstruction

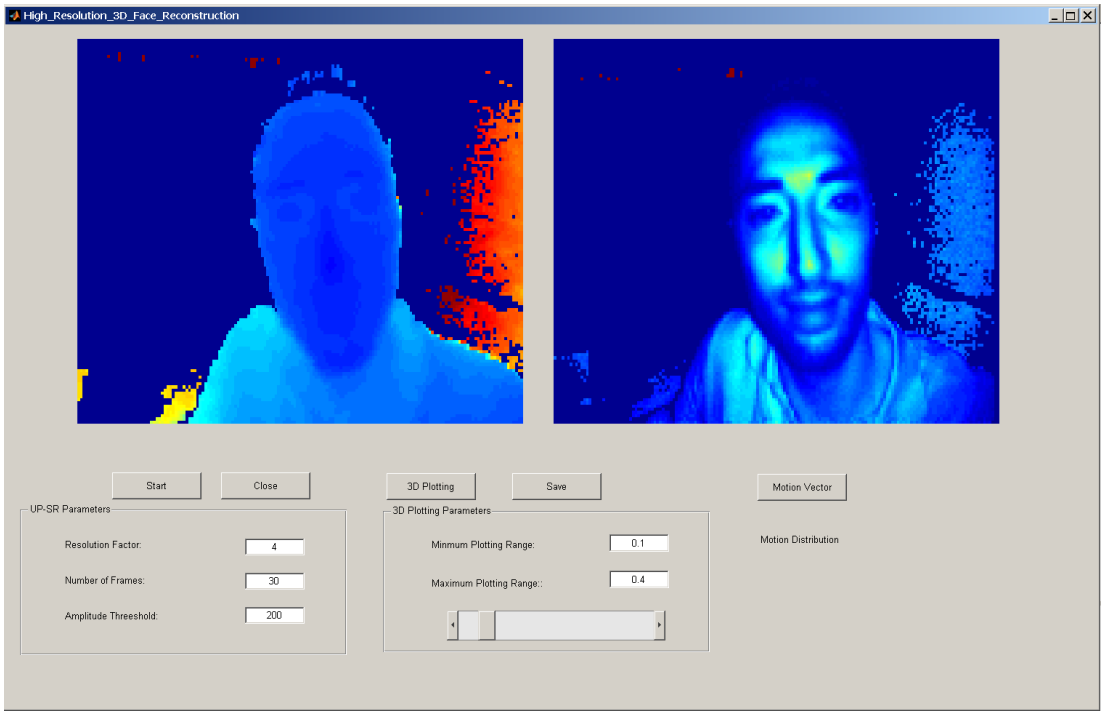


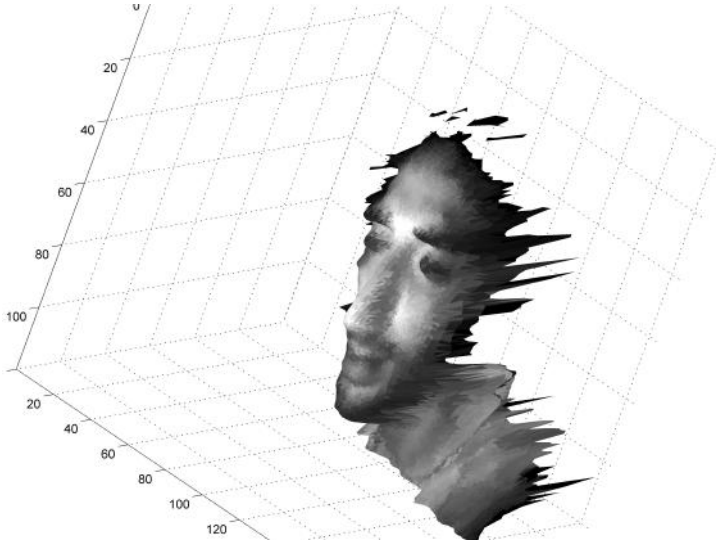
Figure B.1: Tool for automatic HR 3D face reconstruction from acquired low resolution depth images.

We have developed a software tool for automatic HR 3D face reconstruction

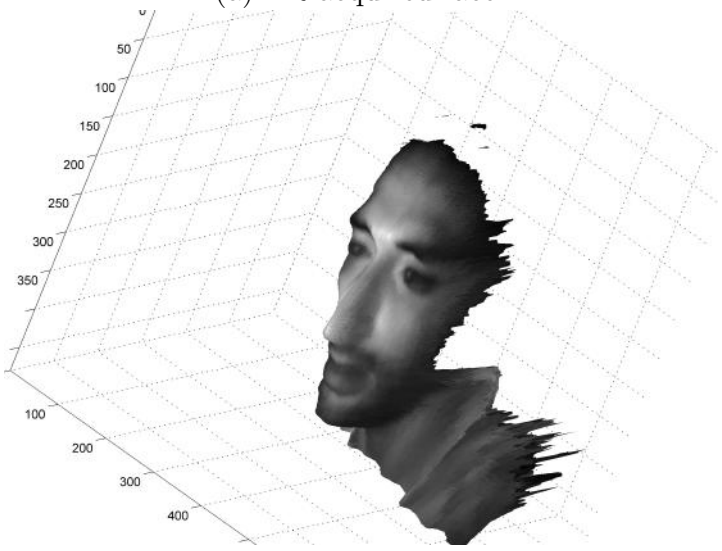
using an LR depth camera. The objective of this software tool is the implementation of the proposed SR algorithms in Chapter 3. A snapshot of the developed tool is shown in Figure B.1 where we show a sample of an acquired LR depth image and its corresponding amplitude image.

Some specific features of the software are:

- The input LR sequence is directly acquired using an LR depth camera.
- The user is able to specify the number of the acquired LR depth frames N.
- The user is able to specify the SR scale factor r.
- Motion estimation is done automatically by the software.
- The output is HR super–resolved depth frames which can be plotted in 3D and saved in .mat format.
- The software tool allows a direct mapping between the super–resolved depth frame and the corresponding super–resolved 2D (amplitude) image.
- The mapped super–resolved images can be plotted in 3D as illustrated in Figure B.2.



(a) LR acquired face



(b) HR super–resolved face

Figure B.2: HR 3D face reconstruction. (a) LR 3D Face acquired using the PMD camera. (b) Super–resolved 3D Face using the proposed method with 30 acquired LR frames.

Bibliography

[1] D-IMager. http://www2.panasonic.biz/es/densetsu/device/3DImageSensor/, 2015. xv, 2

[2] SwissRanger. http://enterprise.hptg.com/, 2015. xv, 2

[3] PMD technologies. siegen, germany. camboard nano,. http://www.pmdtec.com, 2015. xv, 1, 2, 5, 9, 42, 78, 86, 90, 104, 112

[4] 3D MLI. http://www.iee.lu/home-page., 2015. xv, 1, 2, 42, 57, 103

[5] Microsoft kinect camera. https://www.microsoft.com/en$-$us/kinectforwindows/, 2015. xv, 1, 2, 17

[6] F. Brunet. Contributions to parametric image registration and 3d surface reconstruction. In *European Ph.D. in Computer Vision, Université dAuvergne, Clérmont-Ferrand, France, and Technische Universitat Munchen, Germany*, 2010. xv, 18, 19

[7] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super-resolution. In *IEEE Transactions on Image Processing*, pages 1327–1344, 2004. xvi, 3, 8, 16, 23, 24, 25, 33, 34, 35, 38, 39, 40, 46, 47, 50, 52, 70, 76, 94, 106

[8] N.C. Beaulieu and S. Jiang. http://people.csail.mit.edu/drdaniel/mesh$_$animation/. xvii, 58, 59, 61, 81

[9] O. M. Aodha, N. Campbell, A. Nair, and G. Brostow. Patch based synthesis for single depth image super-resolution. In *European Conference on*

Computer Vision, pages 71–84, 2012. xvii, xix, xx, 2, 8, 24, 52, 58, 59, 60, 61, 81, 82, 87

[10] F. Garcia, D. Aouada, B. Mirbach, T. Solignac, and B. Ottersten. Real-time hybrid tof multi-camera rig fusion system for depth map enhancement. In *IEEE International Conference Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2011. xvii, 1, 46, 58, 60

[11] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. In *International Journal of Imaging Systems and Technology*, pages 47–57, 2004. xvii, 14, 15, 16, 58, 60

[12] M. Sturmer, J. Penne, and J. Hornegger. Standardization of intensity-values acquired by time-of-flight-cameras. In *IEEE Workshop on Computer Vision and Pattern Recognition*, pages 1–6, 2008. xviii, 73, 74

[13] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Dynamic super-resolution of depth sequences with non-rigid motions. In *IEEE International Conference on Image processing*, pages 660–664, 2013. xix, xx, 7, 70, 81, 82, 87

[14] S. Berretti, A. Del Bimbo, and P. Pala. Superfaces: A super-resolution model for 3d faces. In *5th Workshop on Non–Rigid Shape Analysis and Deformable Image Alignment*, pages 73–82, 2012. 1, 104, 113

[15] D. Aouada, K. Al Ismaeil, K. K. Idris, and B. Ottersten. Surface up-sr for an improved face recognition using low resolution depth cameras. In *11th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 107–112, 2014. 1, 9

[16] S. Berretti, P. Pala, and A. Del Bimbo. Face recognition by super-resolved 3d models from consumer depth cameras. In *IEEE Transactions on Information Forensics and Security*, pages 1436–1449, 2014. 1

[17] F. Garcia, D. Aouada, B. Mirbach, T. Solignac, and B. Ottersten. Unified multi-lateral filter for real-time depth map enhancement. In *Image and Vision Computing*, pages 26–41, 2015. 1

[18] Q. Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 1, 46, 77

[19] J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[20] J. Zhu, L. Wang, R. Yang, J. E. Davis, and Z. Pan. Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1400–1414, 2011. 1

[21] Q. Yang, K. Tan, B. Culbertson, and J. Apostolopoulos. Fusion of active and passive sensors for fast 3d capture. In *IEEE International Workshops Multimedia Signal Processing*, pages 69–74, 2010. 1

[22] P. Milanfar. Super-resolution imaging. In *CRC Press*, 2010. 2, 13

[23] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Robust shift and add approach to super-resolution. In *International Symposium on Optical Science and Technology*, pages 121–130, 2003. 3, 16, 31, 47, 50, 53, 117

[24] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. High-quality scanning using time-of-flight depth superresolution. In *IEEE Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2008. 4

[25] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *IEEE International Conference Computer Vision and Pattern Recognition*, pages 343–350, 2009. 4, 16, 51, 118

[26] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Variational bayesian super resolution. In *IEEE Transactions on Image Processing*, pages 984–999, 2011. 4, 24, 25, 26, 34, 40, 46

[27] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*, pages 984–999, 2011. 4, 8, 52

[28] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. In *IEEE Transactions on Visualization and Computer Graphics*, page 643650, 2012. 5

[29] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. In *ACM Transactions on Graphics Proceedings of ACM SIGGRAPH Asia*, pages 187:1–187:9, 2013. 5

[30] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. 3d scanning deformable objects with a single rgbd sensor. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5

[31] H. Li, J. Yu, Y. Ye, and C. Bregler. Real-time facial animation with on-the-fly correctives. In *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH*, 2013. 5

[32] M. Zollhofer, M. NieBner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. In *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH*, pages 156:1–156:12, 2014. 5

[33] H. Afzal, K. Al Ismaeil, D. Aouada, F. Destelle, B. Mirbach, and B. Ottersten. Kinectdeform: Enhanced 3d reconstruction of non-rigidly deforming objects. In *3DV Workshop on Dynamic Shape Measurement and Analysis*, 2014. 5, 6, 118, 119

[34] H. Afzal, D. Aouada, B. Mirbach, and B. Ottersten. View-independent enhanced 3d reconstruction of non-rigidly deforming objects. In *Proceedings of the 16th International Conference on Computer Analysis of Images and Patterns*, 2015. 5, 6

[35] http://www.primesense.com/, 2015. 5, 112

[36] R. Newcombe, D. Fox, and S. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5, 6, 118

[37] Middlebury dataset. http://vision.middlebury.edu/stereo/data/, 2015. 6, 40, 56

[38] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Depth super-resolution by enhanced shift & add. In *Proceedings of the 15th International Conference on Computer Analysis of Images and Patterns*, pages 100–107, 2013. 6, 52

[39] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Multi-frame super-resolution by enhanced shift & add. In *IEEE International Symposium on Image and Signal Processing and Analysis*, pages 171–176, 2013. 6

[40] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Enhancement of dynamic depth scenes by upsamplig for precise super–resolution (up–sr). In *International Journal in Computer Vision and Image Understanding, Springer, (Under review)*, 2015. 6, 7

[41] D. Aouada, K. Al Ismaeil, and B. Ottersten. Patch-based statistical performance analysis of upsampling for precise superresolution. In *10th International Conference on Computer Vision Theory and Applications*, 2015. 7, 54, 77

[42] K. Al Ismaeil, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten. Real-time non-rigid multi-frame depth video super-resolution. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2015. 7

[43] K. Al Ismaeil, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten. Real–time enhancement of dynamic depth videos with non–rigid deformations. In *IEEE Transactions on Pattern Analysis and Machine Intelligence. (Under review)*, 2015. 7, 8

[44] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. In *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH*, pages 950–953, 2003. 8

[45] W. Li, C. Zhao, Q. Liu, Q. Shi, and S. Xu. A parameter-adaptive iterative regularization model for image denoising. In *EURASIP Journal on Advances in Signal Processing*, 2012. 8, 77

[46] M. R. Charest, M. Elad, and P. Milanfar. A general iterative regularization framework for image denoising. In *40th Annual Conference on Information Sciences and Systems*, 2006. 8, 77

[47] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. In *Multi-scale Model Simulation*, pages 460–489, 2005. 8, 77

[48] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Bilateral filter evaluation based on exponential kernels. In *International Conference on Pattern Recognition*, pages 258–261, 2012. 8, 52, 76

[49] M. D. Robinson, S. J. Chiu, C. A. Toth, J. A. Izatt, J. Y. Lo, and S. Farsiu. New applications of super–resolution in medical imaging. In *Book chapter in Super–Resolution Imaging, CRC Press*, pages 383–412, 2010. 13

[50] F. Li, X. Jia, and D. Fraser. superresolution reconstruction of multispectral data for improved image classification. In *IEEE Geoscience and Remote Sensing Letters*, pages 689–693, 2009. 13

[51] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. In *IEEE Transactions on Image Processing*, pages 1646–1658, 1997. 15

[52] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space invariant blur. In *IEEE Transactions on Image Processing*, pages 1187–1193, 2001. 15, 16

[53] S. Lertrattanapanich and N. K. Bose. High resolution image formation from low resolution frames using delaunay triangulation. In *IEEE Transactions on Image Processing*, pages 1427–1441, 2002. 16

[54] M. C. Chiang and T. E. Boulte. Efficient super-resolution via image warping. In *Image and Vision Computing*, page 761771, 2000. 16

[55] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. Algorithms for 3d shape scanning with a depth camera. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1039–1050, 2013. 16, 51

[56] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. In *Sensors 2012*, pages 1437–1454, 2013. 18

[57] H. Sarbolandi, D. Lefloch, and A. Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. In *Computer Vision and Image Understanding, In Press*, 2015. 18

[58] M. Hansard, S. Lee, O. Choi, and R.P. Horaud. Time-of-flight cameras: Principles, methods and applications. In *Springer Briefs in Computer Science*, 2013. 18

[59] M. Lindner and A. Kolb. Compensation of motion artifacts for time of-flight cameras. In *Proceeding of Dynamic 3D Vision Workshop*, pages 16–27, 2009. 20, 51

[60] M. Yamamoto, P. Boulanger, J. A. Beraldin, and M. Rioux. Direct estimation of range flow on deformable shape from a video rate range camera. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 82–89, 1993. 20, 72

[61] H. Spies, B. Jahne, and J. Barron. Regularised range flow. In *European Conference in Computer Vision*, pages 785–799, 2000. 20

[62] H. Spies, B. Jahne, and J. Barron. Range flow estimation. In *Computer Vision and Image Understanding*, page 209231, 2002. 20

[63] H. Spies and J. Barron. Evaluating the range flow motion constraint. In *16th International Conference on Pattern Recognition*, pages 517–520, 2002. 20

[64] J. M. Gottfried, J. Fehr, and C. S. Garbe. Computing range flow from multi-modal kinect data. In *Advances in Visual Computing, Lecture Notes in Computer Science*, pages 758–767, 2011. 20, 21, 73

[65] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3d motion understanding. In *International Journal of Computer Vision*, pages 29–51, 2011. 21

[66] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *IEEE International Conference on Robotics and Automation*, pages 2276–2282, 2013. 21, 73

[67] R. Y. Tsai and T. S. Huang. Multiframe image restoration and registration. In *Advances in Computer Vision and Image Processing*, pages 317–339, 1984. 23

[68] N. K. Bose, H. C. Kim, and H. M. Valenzuela. Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframe. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 269–272, 1993. 23

[69] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. In *IEEE Transactions on Image Processing*, pages 1187–1193, 2001. 23

[70] M. Chiang and T. E. Boult. Efficient super-resolution via image warping. In *Image and Vision Computing*, pages 761–771, 2000. 23

[71] M. Irani and S. Peleg. Improving resolution by image registration. In *Graphical models and image processing*, pages 231–239, 1991. 23

[72] A. Zomet, A. Rav-Acha, and S. Peleg. Robust super-resolution. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 645–650, 2001. 23, 46

[73] L. Zhouchen and S. Heung-Yeung. Fundamental limits of reconstruction-based superresolution algorithms under local translation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 83–97, 2004. 23, 24

[74] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *IEEE International Conference on Computer Vision*, 2009. 24

[75] J. Y. Bouguet. Pyramidal implementation of the lukas kanade feature tracker. description of the algorithm. http://robots.stanford.edu/cs223b04/algo_tracking. 27

[76] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision*. 27

[77] L. Xu, J. Jia, and S. B. Kang. Improving sub-pixel correspondence through upsampling. In *Journal on Computer Vision and Image Understanding*. 30

[78] M. Elad and A. Feuer. Super-resolution reconstruction of continuous image sequence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 31, 50, 70, 76

[79] L. J. Karam, N. G. Sadaka, R. Ferzli, and Z. A. Ivanovski. An efficient selective perceptual-based super-resolution estimator. In *IEEE Transactions on Image Processing*. 34

[80] S. Farsiu. Resolution enhancement software: http://www1.idc.ac.il/toky/videoproc-07/projects/superres/, year = , month = , pages=,. 34

[81] S. Villena, M. Vega, D. Babacan, J. Mateos, R. Molina, and A. K. Katsaggelos. Super–resolution software, http://decsai.ugr.es/pi/superresolution/software.html. 34

[82] S. Villena, M. Vega, D. Babacan, J. Mateos, R. Molina, and A. K. Katsaggelos. Mdsp super resolution and demosaicing datasets. available: http://users.soe.ucsc.edu/~milanfar/software/ sr-datasets.html. 35, 38

[83] R. Hardie, T. Tuinstra, K. Barnard, J. Bognar, and E. Armstrong. High resolution image reconstruction from digital video with global and non-global scene motion. In *IEEE International Conference Image Processing*, pages 153–156, 1997. 46

[84] S. Farsiu, M. Elad, and P. Milanfar. Video-to-video dynamic super–resolution for grayscale and color sequences. In *Journal on Advances in Signal Processing*, 2006. 46, 70

[85] A. W. M. van Eekeren, K. Schutte, J. Dijk, D. J. D. Lange, and L. J. van Vliet. Super-resolution on moving objects and background. In *IEEE International Conference on Image Processing*, pages 2709–2712, 2006. 46

[86] A. W. M. van Eekeren, K. Schutte, and L. J. van Vliet. Multiframe super-resolution reconstruction of small moving objects. In *IEEE Transactions on Image Processing*, pages 2901–2912, 2010. 46

[87] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super–resolution without explicit sub–pixel motion estimation. In *IEEE Transactions on Image Processing*, pages 1958–1975, 2009. 47, 107

[88] F. Garcia, D. Aouada, B. Mirbach, T. Solignac, and B. Ottersten. Spatio-temporal tof data enhancement by fusion. In *IEEE International Conference on Image Processing*, pages 981–984, 2012. 49

[89] A. Rajagopalan and P. Kiran. Motion-free super-resolution and the role of relative blur. In *J. Opt. Soc. Amer.*, page 20222032, 2003. 52

[90] D. Robinson and P. Milanfar. Statistical performance analysis of super-resolution. In *IEEE Transactions on Image Processing*, pages 1413–1428, 2006. 52

[91] D. Robinson and P. Milanfar. Bias-minimizing filters for motion estimation. In *IEEE Asilomar Conference on Signals, Systems and Computers*, 2003. 53

[92] P. Chatterjee and P. Milanfar. Bias modeling for image denoising. In *IEEE Asilomar Conference on Signals, Systems and Computers*, pages 856–859, 2009. 53, 54

[93] N.C. Beaulieu and S. Jiang. Ml estimation of signal amplitude in laplace noise. In *IEEE Conference Global Telecommunications*, pages 1–5, 2010. 55

[94] N.C. Beaulieu and S. Jiang. http://www.k-team.com/mobile-robotics-products/v-rep. 62, 81

[95] V. Patanavijit and S. Jitapunkul. An iterative super–resolution reconstruction of image sequences using a lorentzian bayesian approach with fast affine block–based registration. In *International Conference on Wireless Communications and Mobile Computing*, pages 51–56, 2006. 69

[96] M. Elad and A. Feuer. Super-resolution restoration of an image sequence: adaptive filtering approach. In *IEEE Transactions on Image Processing*, pages 387–395, 1999. 70

[97] B. C. Newland, A. D. Gray, and D. Gibbins. Modified kalman filtering for image super-resolution: Experimental convergence results. In *The Ninth International Conference on Signal and Image Processing*, pages 58–63, 2007. 70

[98] J. Tian and K. K. Ma. A new state-space approach for super-resolution image sequence reconstruction. In *IEEE International Conference on Image Processing*, pages 881–884, 2005. 70

[99] M. Elad. On the origin of bilateral filter and ways to improve it. In *IEEE Transactions on Image Processing*, pages 1141–1151, 2002. 76, 94

[100] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*, page 836846, 1998. 76, 93

[101] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 2007. 77

[102] F. Garcia, D. Aouada, B. Mirbach, T. Solignac, and B. Ottersten. A new multi-lateral filter for real-time depth enhancement. In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 2011. 77

[103] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. In *IEEE Signal Processing Magazine*. 77

[104] A. Kheradmand and P. Milanfar. A general framework for regularized, similarity-based image restoration. In *IEEE Transactions on Image Processing*, pages 5136–5151, 2014. 77

[105] L. Valgaerts, C. Wu, A. Bruhn, H. P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. In *ACM Transactions on Graphics*, 2012. 83

[106] C. M. Stein. Estimation of the mean of multivariate normal distribution. In *The Annals of Statistics*, pages 1135–1151, 1981. 94, 96

[107] S. Ramani, T. Blu, and M. Unser. Monte–carlo sure: a black-box optimization of regularization parameters for general denoizing algorithm. In *IEEE Transactions on image processing*, pages 1540–1554, 2008. 94

[108] D. V. Ville and M. Kocher. Sure–based non-linear means. In *IEEE Transactions on image processing*, pages 2683–2690, 2001. 94

[109] H. Peng and R. Rao. Bilateral kernel parameter optimization by risk minimization. In *International Conference on Image Processing*, pages 3293–3296, 2010. 96

[110] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. In *ACM Transactions on Graphic*, pages 40:1–40:9, 2010. 103

[111] F. Lin, C. Fookes, V. Chandran, and S. Sridharan. Super-resolved faces for improved face recognition from surveillance video. In *Advances in Biometrics*, pages 1–10, 2007. 104

[112] C. Fookes, F. Lin, V. Chandran, and S. Sridharan. Evaluation of image resolution and super-resolution on face recognition performance. In *Journal of Visual Communication and Image Representation*, pages 75–93, 2012. 104

[113] S. Peng, G. Pan, and Z. Wu. Learning-based super-resolution of 3d face model. In *Proceedings of the IEEE International Conference on Image Processing*, page 382385, 2005. 104

[114] G. Pan, S. Han, Z. Wu, and Y. Wang. Super-resolution of 3d face. In *IEEE European Conference on Computer Vision*, page 389401, 2006. 104

[115] M. Hernandez, J. Choi, and G. Medioni. Laser scan quality 3-d face modeling using a low-cost depth camera. In *Proceedings of the IEEE European Signal Processing Conference*, pages 1995–1999, 2012. 104

[116] S. Cuomo, A. Galletti, G. Giunta, and A. Starace. Surface reconstruction from scattered points via rbf interpolation on gpu. In *Federated Conference on Computer Science and Information Systems*. 107

[117] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001. 110

[118] D. Aouada and H. Krim. Squigraphs for fine and compact modeling of 3-d shapes. In *IEEE Transactions on Image Processing*, pages 306–321, 2010. 110

[119] D. Smeets, P. Claes, J. Hermans, D. Vandermeulen, and P. Suetens. A comparative study of 3-d face recognition under expression variations. In *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, pages 710–727, 2012. 111

[120] Superfaces dataset. http://www.micc.unifi.it/vim/datasets/4d-faces/, 2015. 112