



University of Padova
Department of Information Engineering

Ph.D. School of Information Engineering
Information Science and Technology

26th Cycle

Disparity and Motion Estimation and their Application to 3D Reconstruction

Francesco Michielin

Advisor:
Prof. Giancarlo Calvagno

Ph.D. School Director:
Prof. Matteo Bertocco

Co-Advisor:
Ing. Piergiorgio Sartor

Course coordinator:
Prof. Carlo Ferrari

Academic Year 2013-2014

To Caterina

Contents

Abstract	ix
Sommario	xi
Acknowledgments	xiii
List of Acronyms	xv
List of Figures	xvii
List of Tables	xxi
1 Introduction	1
2 Real motion estimation	5
2.1 Introduction	5
2.2 Recursive search true motion estimation	6
2.3 Binarized cross correlation	8
2.4 Recursive search real motion estimation based on binarized cross correlation	11
2.5 Parallel motion estimation - MePar	12
2.6 Experimental results	13
2.7 Conclusions	18
3 Disparity Estimation	19
3.1 Introduction	19
3.2 Disparity estimation algorithms	21
3.2.1 Local stereo algorithms	21
3.2.2 Global stereo algorithms	21
3.2.3 Semi-global stereo algorithm	21
3.3 Simple Tree stereo algorithm	22
3.4 User assisted processing	23
3.4.1 Directional filtering	24
3.4.2 Directional binarization	25

3.4.3	Experimental results	26
3.5	Conclusions and future work	28
4	Hybrid System	31
4.1	Introduction	31
4.2	Time-of-flight cameras	32
4.2.1	CW ToF Cameras: typical distance measurement errors	34
4.3	ToF Cameras and Stereo System: comparison and combination	35
4.4	Ideal model for depth super-resolution	36
4.4.1	Joint bilateral filter	37
4.4.2	“Kim” Joint bilateral filter	37
4.4.3	Weighted joint bilateral filter	38
4.4.4	Experimental results	40
4.5	Real model for depth super-resolution	44
4.5.1	Camera rig	45
4.5.2	Calibration	45
4.5.3	Color images pre-processing	46
4.5.4	ToF pre-processing	46
4.5.5	Mesh mapping	48
4.5.6	Iterative depth super-resolution	49
4.5.7	Stereo refinement	51
4.5.8	Experimental results	52
4.6	Conclusions	56
5	Combined motion and disparity estimation	57
5.1	Introduction	57
5.2	Independent estimations	58
5.3	Dependent estimations: round trip check	60
5.4	Experimental results	63
5.4.1	Middlebury database for Optical Flow	63
5.4.2	KITTI Vision Benchmark Suite	66
5.5	Conclusions and future work	67
6	Application to 3D reconstruction	71
6.1	Introduction	71
6.2	System explanation	72
6.2.1	Camera calibration	73
6.2.2	CoMEDE and Optical Flow combination	73
6.2.3	Results	74
6.3	Multiple images	77
6.3.1	Color based segmentation	77

6.3.2	Correspondence Chaining	78
6.3.3	Results	78
6.4	ToF support	83
6.5	Conclusion and future work	85
7	Conclusions	87
	Bibliography	89
	List of Publications	97

Abstract

This thesis provides an overview of the research that was done along the three years of the Ph.D. studies in the field of motion and disparity estimations. Concerning the motion estimation, a new parallel algorithm, which is suitable for a GPU based implementation, and a novel matching criterion based on a cross correlation based on binarized input are proposed. The combination of these techniques significantly reduces the computational cost while maintaining comparable results with respect to a standard recursive search method. On the topic of disparity estimation, a semi-automatic processing to align the disparity edges with the object borders is presented. This takes advantage of a possible user interaction which is used to mark the wrongly estimated disparity jumps. This information is then integrated in the estimation process by means of a feedback loop in the smoothness constraint of the algorithm. The support of an active system for the purpose of geometry estimation is evaluated as well. Specifically the problem of a Time-of-Flight camera depth map super-resolution guided from a single or multiple color cameras is addressed. A considerable gain is achieved with respect to state of the art methods, in particular with an increasing level of noise. In the second part of the thesis the motion and disparity estimation problem is jointly formulated. A method that tries to solve the aperture problem and to obtain a more robust estimation by loosely coupling a set of independent estimations is proposed. Specifically, the different estimations are processed in a hierarchical fashion and between every iteration a consistency between the resulting displacements is calculated. This is used as a quality measurement and integrated in the following iteration processing. Numerical results show the effectiveness of this approach. Eventually the techniques that have been developed are employed in a possible application scenario, i.e., the 3D reconstruction based on dense correspondence estimation. In particular we show how the combination of a block based recursive search algorithm with a subsequent optical flow method permits to obtain an accurate estimation and consequently a very precise reconstruction.

Sommario

Questa tesi fornisce una visione d'insieme della ricerca effettuata durante i tre anni di dottorato nell'ambito della stima del moto e disparità. Per la stima del moto si propone un nuovo algoritmo adatto per un'implementazione su GPU ed un criterio di stima delle corrispondenze che calcola la correlazione in una sequenza precedentemente binarizzata. La combinazione di queste tecniche permette una significativa riduzione del costo computazionale ottenendo comunque risultati comparabili con un algoritmo ordinario di ricerca ricorsiva. Per la stima della disparità si propone invece un processamento semi-automatico per allineare i bordi della disparità con quelli degli oggetti. In particolare questo metodo usa una possibile iterazione con l'utente per definire le regioni non stimate correttamente. Questa informazione è poi integrata nell'algoritmo tramite un feedback nel vincolo di smoothness. Inoltre si è anche considerato il possibile supporto di un sistema attivo, più precisamente ci si è concentrati sull'aumento di risoluzione della depth map di una camera a tempo-di-volo guidato da una o più camere a colori. I risultati dimostrano un notevole guadagno rispetto all'attuale stato dell'arte, in particolar modo in caso di un alto livello di rumore. Nella seconda parte della tesi viene formulato in modo congiunto il problema della stima del moto e disparità. A questo riguardo viene proposto un metodo che cerca di risolvere il problema dell'apertura e di ottenere conseguentemente una stima più robusta mediante un accoppiamento lasco tra un insieme di stime indipendenti. Più precisamente le differenti stime vengono effettuate tramite un procedimento gerarchico e tra ogni iterazione viene calcolata la consistenza dei risultati. Questa rappresenta una stima della qualità ed è quindi integrata nella successiva iterazione permettendo un miglioramento dei risultati ottenuti. Infine le tecniche sviluppate precedentemente sono state applicate alla ricostruzione tridimensionale. In particolar modo viene mostrato come la combinazione di un algoritmo di ricerca ricorsiva con un metodo basato sull'optical flow permetta una stima densa e precisa e conseguentemente un'accurata ricostruzione.

Acknowledgments

First of all I want to thank Prof. Giancarlo Calvagno and Piergiorgio Sartor since both supervised me during this PhD. Even if in different ways I learnt a lot under their guidance. I am grateful to Oliver Erdler, since he always stimulated me to do my best, and to Yalcin Incesu for his precious advices and his patience. I am thankful to various colleagues and friends that shared with me their time, opinions and ideas: Marco Martin, Paolo Baracca, Matteo Canale, Michiele Caruso, Pietro Zanuttigh, Simone Milani, Chiara Masiero, Francesco Simmini, Martina Favaro, Carlo Dal Mutto, Fabio Dominio, Matteo Bassi, Enrico Cappelletto, Mauro Donadeo, Fabio Padovan, Andrea Rigoni, Christian Unruh, Shamus Donovan, Fabian Schaschek, Martin Fritz, Thimo Emmerich, Paul Springer, Jens Nausedat, Volker Freiburg, Roman Streubel, Alessandro Vianello, Oliver Wasenmueller. In particular I have to thanks Matthias Brueggemann and Bernd Krolla with which I had a really interesting and profitable collaboration. It is nice to work with professionals. I am very thankful to Caterina for her patience, her constant support, and for her will to improve ourself day by day. Last but not least I want to thank my family who have always supported and helped me to become the person I am today.

List of Acronyms

ME	Motion estimation
MVF	motion vector field
OF	optical flow
PC	phase correlation
RS	recursive search
ROI	region of interest
SAD	sum of absolute differences
CC	cross correlation
MV	motion vector
BCC	binarized cross correlation
BBD	binarized block difference
AE	angular error
EE	absolute flow endpoint error
CCS	camera coordinate system
WTA	winner take all
NCC	normalized cross correlation
DP	dynamic programming
MRF	Markov random field
UI	user interaction
MAE	mean absolute error
Out-Noc	percentage of erroneous pixels in non-occluded areas

ToF	time-of-flight
CCD	charge-coupled device
CW	continuous wave
LUT	look-up-table
LMP	Lux Media Plan
BF	bilateral filter
JBF	joint bilateral filter
WJBF	weighted joint bilateral filter
CoMEDE	combined motion and disparity estimation
RTC	round trip check
LVC	local vector consistency
TUDo	Technical University of Dortmund
DFKI	German Research Center for Artificial Intelligence
ICP	iterative closest point

List of Figures

2.1	Block base RS framework	7
2.2	Predictor selection in RS	8
2.3	Update star in the block based RS with SAD matching criteria	9
2.4	Image binarization	10
2.5	Binarized cross correlation matching	10
2.6	Block base RS framework with BCC as matching criterion	11
2.7	Comparison between SAD and BBD matching in RS	11
2.8	Predictor selection in MePar	12
2.9	Comparison between SAD and BBD matching	13
2.10	Results comparison of the RS and MePar based on the SAD and BCC matching criteria	14
2.11	Results comparison in RubberWhale sequence	16
2.12	Results comparison in Grove2 sequence	16
2.13	Results comparison in Grove3 sequence	17
2.14	Results comparison in Hydrangea sequence	17
3.1	Schematic representation of the 2D CCS and 3D CCS associated to the left and the right cameras.	19
3.2	Simple Tree framework	22
3.3	User assisted processing framework	23
3.4	Example of user interaction	24
3.5	Results of directional filtering	25
3.6	P_2 penalty calculation	26
3.7	Results comparison of the Simple Tree algorithm and UI	27
3.8	Results comparison for the Venus image	29
3.9	Results comparison for the Tsukuba image	29
3.10	Results comparison for the Barn1 image	29
3.11	Results comparison for the Poster image	29
4.1	Examples of ToF camera models	33
4.2	Multi-path effect	34

4.3	Motivation of flying pixels problem	34
4.4	Example of flying pixels problem	35
4.5	Three-sigma rule	39
4.6	Creation of the weighting factor α image	40
4.7	Visual comparison of the <i>aloe</i> scene super-resolution	41
4.8	Objective comparison of the <i>aloe</i> scene super-resolution	41
4.9	Visual comparison of the <i>baby1</i> scene super-resolution	42
4.10	Objective comparison of the <i>baby1</i> scene super-resolution	42
4.11	Visual comparison of the <i>wood2</i> scene super-resolution	43
4.12	Objective comparison of the <i>wood2</i> scene super-resolution	43
4.13	Depth super-resolution framework	44
4.14	Camera rig	45
4.15	Calibration pattern	46
4.16	ToF Preprocessing	46
4.17	Results of the ToF pre-processing	47
4.18	Flying pixel detection	47
4.19	Graphical representation of the mesh mapping	48
4.20	Mesh mapping results	49
4.21	Iterative super-resolution	50
4.22	Iterative super-resolution results	50
4.23	Stereo refinement	51
4.24	Stereo refinement results	51
4.25	Results of super-resolution on <i>Pyramid</i>	53
4.26	Results of super-resolution on <i>Elk</i>	54
4.27	Results of super-resolution on <i>Lion</i>	55
5.1	Block size in hierarchical estimation, from coarse to fine	59
5.2	CoMEDE framework	59
5.3	CoMEDE with RTC framework	60
5.4	Round trip check	61
5.5	Comparison of SAD and RTC	62
5.6	Selection of the predictors based on the RTC and LVC	62
5.7	Results comparison CoMEDE system	63
5.8	Results comparison in RubberWhale sequence	64
5.9	Results comparison in Grove2 sequence	64
5.10	Results comparison in Grove3 sequence	65
5.11	Results comparison in Hydrangea sequence	65
5.12	Results comparison CoMEDE system on KITTI	66
5.13	Results comparison in 023 sequence	68
5.14	Results comparison in 068 sequence	68

5.15	Results comparison in 095 sequence	69
5.16	Results comparison in 193 sequence	69
6.1	Stereo 3D reconstruction framework	72
6.2	Results comparison of CoMEDE system and OF	74
6.3	Results comparison in RubberWhale sequence	75
6.4	Results comparison in Grove2 sequence	75
6.5	Results comparison in two view 3D reconstruction	76
6.6	Results comparison in two view 3D reconstruction	77
6.7	Multiview 3D reconstruction framework	78
6.8	Datasets for the 3D reconstruction	79
6.9	3D reconstruction of the <i>Lion</i> dataset	80
6.10	3D reconstruction of the <i>Civetta</i> dataset	80
6.11	3D reconstruction of the <i>Lion</i> dataset	81
6.12	3D reconstruction of the <i>Civetta</i> dataset	82
6.13	Framework for the 3D reconstruction based on the ToF and cameras array .	84
6.14	Results comparison in ToF based 3D reconstruction	86

List of Tables

2.1	MVF convergence time (number of frames)	15
2.2	SAD and BCC comparison (number of operations)	15
4.1	Advantages and disadvantages of a stereo vision system and ToF cameras .	35
5.1	Aperture problem: advantages and disadvantages in the block size selection	58
6.1	Advantages and disadvantages of RS based and OF based approaches . . .	73
6.2	3D reconstruction of the <i>Lion</i> dataset	83
6.3	3D reconstruction of the <i>Civetta</i> dataset	83

Chapter 1

Introduction

In the recent years there has been considerable interest in the estimation of a scene geometry and the related objects motion. The estimation of the motion in an image sequence has been widely used in many techniques. Frame rate conversion, video compression, temporal noise reduction or temporal super-resolution correspond to only a minor part in a large set of applications. This heterogeneity led to the development of specific algorithms for different applications. Indeed in a video compression application, where the aim is to minimize the number of bits sent, the estimated vectors may also not correspond to the real motion of the objects. Vice versa in a frame rate conversion the true motion vectors are necessary to avoid the presence of artifacts and it should be considered that a simple brute force minimization of a matching criteria may not fulfill this constrain.

In the field of geometry estimation the stereo vision system has been extensively investigated for the last few decades. The aim is to detect the correspondences in a stereo image pair based on the assumption that the projection of a patch of the scene into the two cameras is similar. This is a problem that, despite to the copious effort spent by the researchers, has not been solved completely. In fact the underlying assumption may not be satisfied, e.g., in presence of non-Lambertian surfaces and occlusions, or, as for the real motion estimation, the similarity measurement may not derive the correct displacement, e.g., in case of different light conditions or texture-less regions.

Other possible approaches for the reconstruction of the scene geometry are formed by the class of active systems. These methods actively interact with the scene and are, in general, more robust than a stereo system. Moreover the combination of active and passive techniques have been recently investigated. In particular one of the most investigated topics is the combination of the Time-of-Flight range cameras with stereo vision systems, due to their complementary features.

Recently a large number of researchers started to consider the possibility to combine the recovering of the scene geometry with the associated objects motion. Indeed to improve the quality of the resulting 3D motion estimation in a stereo sequence it is important to couple the different estimations. This joint representation of motion and geometry is

then suitable for many applications in the field of computer vision, as for example the 3D reconstruction or the integration of view synthesis with frame rate conversion. Moreover even biomedical and automotive applications are possible.

This thesis wants to provide an overview of the research that has been done along the three years of the Ph.D. studies. In this period the problems of motion and disparity estimations were investigated. Initially the two estimations were investigated separately and then a joint formulation of the problem was derived. Moreover, in the purpose of geometry estimation, the support of a Time-of-Flight camera was also considered. Eventually the application of 3D reconstruction was used to show the quality of the obtained results. Specifically the thesis is organized as follows.

- In Chapter 2 we consider the motion estimation problem. In order to satisfy the demand of a faster approach we propose two modifications to the standard block based recursive search [1] that is typically implemented in actual consumer displays. The first contribution is related to a new matching criterion. This is a cross correlation between frames which are binarized in advance. Thanks to this solution the computational cost can be significantly reduced while an objective evaluation shows that we are able to obtain comparable results with respect to the standard recursive search method. A new parallel implementation that avoids the spatial recursion inside the estimation is also introduced. In particular the combination of the two contributions permits to obtain a fast algorithm suitable for a GPU implementation.
- In Chapter 3 we consider the disparity estimation problem. We propose an improvement of the Simple Tree stereo algorithm [2] based on a possible offline user interaction. In particular we analyze the algorithm highlighting the related weakness and in this context we formulate our contribution. Thanks to the a-priori knowledge of the user marks we apply a directional blurring along the object borders to facilitate a subsequent border detection. This is performed by means of an image binarization that combines the luminance and chrominance information. Finally the disparity edges are aligned to the object borders thanks to a feedback loop in the smoothness cost of the algorithm.
- In Chapter 4 we consider the possibility of an active system support in the field of geometry estimation. Specifically we use a Time-of-Flight range camera due to its complementary features with respect to the stereo vision system. For this purpose an algorithm for the super-resolution of a Time-of-Flight depth map driven by a high resolution RGB camera is presented. This is firstly developed in a controlled scenario which is then used to compare the proposed algorithm with state of the art methods. An objective evaluation proves the quality of the obtained results. In the second part of this Chapter the algorithm is adapted to a real data set. In particular the super-resolution is performed in an iterative fashion in order to obtain a scalable

and robust approach. This permits to achieve an accurate and high resolution depth map.

- In Chapter 5 the joint formulation of the motion and disparity estimations in a stereo sequence is considered. We introduce the concept of round trip check as a measure of the estimations quality. This is then used to loosely couple a set of independent estimations, constituted by state of the art of the block based recursive search methods, modifying their implicit smoothness constraint. The effectiveness of the proposed solution, which aims to solve the aperture problem and to obtain a more robust estimation, is proved with an objective evaluation on two different datasets.
- In Chapter 6 we propose a possible practical application of the algorithm developed in Chapter 5. We provide an overview on how a dense estimation can support a 3D reconstruction method. In particular the work presented is the outcome of a collaboration with the Technical University of Dortmund (TU Dortmund) and with the German Research Center for Artificial Intelligence (DFKI) of Kaiserslautern. In the first part we show how the combination of the algorithm developed in Chapter 5 with the optical flow of TU Dortmund outperforms the single estimations. This combination can be used to obtain a dense and accurate 3D reconstruction even from a single pair of images. Then the evolution of the system to multiple cameras is explained. We also compare the resulting reconstruction with the state of the art methods. Eventually we consider an initial integration of the super-resolution algorithm developed in Chapter 4 into the 3D reconstruction framework.
- In Chapter 7 we conclude the thesis with a summary of the main findings.

Chapter 2

Real motion estimation

2.1 Introduction

Motion estimation (ME) is a fundamental problem in image sequences processing. Many techniques, e.g., temporal noise reduction, temporal super-resolution and frame rate conversion, are based on a ME algorithm which calculates a motion vector field (MVF). Various approaches are available in literature but in general ME can be modeled as a selection of weighted error criteria. These are usually a combination of correlation methods with smoothness constraints. The main approaches can be divided into optical flow (OF) [3–5], phase correlation (PC) [6–9] and block based methods with recursive search (RS) [1,10–15]. OF approaches try to solve the ill-posed problem of finding a two component vector as a solution of a single differential equation imposing additional constraints. In particular the methods of Horn & Schunck [3] and Lucas & Kanade [4] solve the OF equation with a global and a local optimization, respectively. Unfortunately OF methods are not suitable for large displacements that are common in high definition content, and real time implementations are only available on standard definition resolution [5]. PC methods compute the shift between two image blocks by means of a linear phase term in the Fourier domain [7]. This corresponds to the normalized cross-correlation in the spatial domain and the associated peak provides the relative shift. The results are also not strongly affected by the presence of noise or luminance changes in the scene [6] but PC has the drawback that the block size has to be double of the largest movement that can be measured [8]. Indeed the Fourier transformation supposes an infinite periodic signal which is not true in case of an image region of interest (ROI). Due to its constraints the PC is commonly used to select an initial set of candidates for a subsequent RS algorithm. This allows to keep the update range low and to obtain a less noisy estimation that admits large displacements [8]. Block based RS motion estimations are typically implemented in actual consumer displays. These methods perform a recursive flow calculation. They compute the vector related to the actual block position minimizing a similarity function, usually the sum of absolute differences (SAD) in combination with different smoothness constraint

penalties, of a small set of candidate vectors based on spatio-temporal predictions. The motion changes are then located by means of a random set of updates applied to the candidates. The choices of the predictors, the updates range and the associated penalties can be seen as an implicit smoothness constraint integrated into the algorithm, e.g., a larger update scale may track larger flow variations or provide a faster convergence but has the drawback of a noisier vector flow estimation. While for the actual consumer displays the real time implementation of the RS can be considered solved by means of the hardware implementation, this is not true for a software based implementation. Different scanning techniques as in [15] have been considered to introduce a parallelism in the estimation and to exploit the multi-core processing of the modern CPU and GPU. In this Chapter we describe two novel methods to reduce the computational cost in a block based RS motion estimation. The first contribution combines the RS and PC approaches in the spatial domain by using a cross correlation (CC) as a matching criteria. This does not only solve the problem of the large block size selection of the PC but has the advantage that the size of the block to match and the search area can be different. In addition the computational complexity is kept low by means of an image binarization on which the CC can be implemented efficiently. The second contribution proposes to avoid the spatial recursion inside the RS scheme and to exploit only the temporal one. This allows a fully parallel estimation which can take advantage of the multi-core processing of a GPU. Moreover the combination of the two methods permits an improvement on the results as well as a reduction of the convergence time of the estimation without increasing the computational cost.

In particular in Section 2.2 we firstly revise the standard block based RS method. In Sections 2.3 and 2.4 we respectively describe the CC on binarized images and how this can be integrated in a RS approach. We present the parallel estimation in Section 2.5 and a overview of the experimental results in Section 2.6. Eventually in Section 2.7 we draw some conclusions.

The methods presented in this chapter were applied for two patents in [16,17]. Furthermore, parts of the work presented here were published in [14].

2.2 Recursive search true motion estimation

A block based RS motion estimation algorithm follows the framework shown in Figure 2.1. The image is partitioned into non-overlapping regions, called blocks. For every block a recursive assignment of vectors which minimize the combination of a similarity function with different smoothness constraint penalties is performed. Depending on the block scanning technique, different blocks may already possess a motion vector which refers to the current frame and iteration, while for other blocks this is not yet calculated. This is shown in Figure 2.2(a) in which the arrow describes the scanning order. The current block for which the motion vector (MV) has to be calculated is in white and the dark

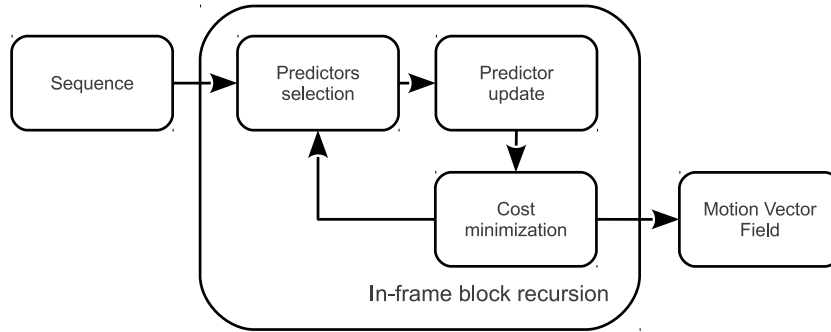


Figure 2.1: Block base RS framework: the image is subdivided in non-overlapping blocks and for every patch, depending on the scanning technique, a set predictors is selected. Each of these provides a MV value that is randomly updated with Gaussian distribution [13] forming a final set of candidates. Eventually the final MV for the current block is selected among this set minimizing a predefined cost function. The operation is recursively repeated for all the blocks in the frame.

or light gray are respectively the blocks for which the MV has already been or not been calculated for this frame and iteration. The dark gray blocks are usually referred as spatial predictors whereas the light gray as temporal predictors. Both the predictor types are used to permit the convergence to the correct MV for the current block. This recursion method is supported by the following assumptions [11]:

1. The block size is smaller than the size of an object.
2. The moving objects have spatial consistency.
3. The moving objects have inertia.

The first two assumptions validate the predictors recursion inside the estimation, the third supports the presence of the temporal predictor. Moreover due to the spatial consistency only a sub-set of the spatio-temporal predictors is necessary for convergence. Then the predictor scheme shown in Figure 2.2(b) is generally used. In the SAD based approach a random update with Gaussian distribution [13] is added to every predictor forming the final candidates (c_x, c_y) , see Figure 2.3. This permits to locate motion changes inside the frame and between consecutive frames. Eventually the motion vector (V_x, V_y) is computed minimizing over the complete candidates set the combination of the SAD value with different smoothness constraint penalties

$$(V_x, V_y) = \arg \min_{(c_x, c_y)} SAD(c_x, c_y) + P(c_x, c_y), \quad (2.1)$$

where

$$SAD(c_x, c_y) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |I(i, j, t) - I(c_x + i, c_y + j, t + 1)|. \quad (2.2)$$

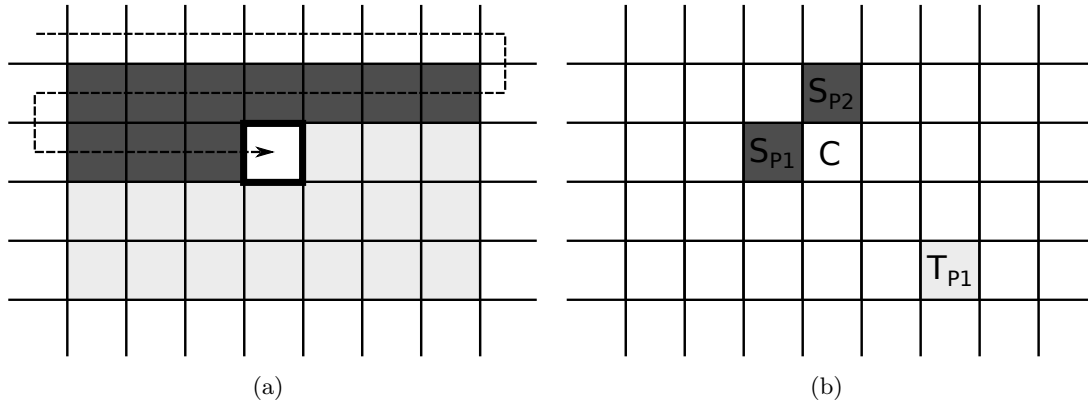


Figure 2.2: Predictor selection: (a) Recursion through all blocks in the vector field with the zig-zag scan. Every block can be classified as spatial predictor or temporal predictor. The first ones, in dark gray, have already a MV that refers to the current frame and iteration whereas the latter ones, in light gray, have a MV that belongs to a previous frame or iteration. (b) Based on the assumption that moving objects have spatial consistency only a subset of the predictors is necessary. This is the case of the classical RS predictor scheme with two spatial and one temporal predictors.

Usually a different set of penalties is considered depending on the candidate typology [1]:

$$P_{Spatial} < P_{Temporal} < P_{Update}. \quad (2.3)$$

This is used to favour the spatial predictors over temporal ones. Indeed the first ones refer to the current frame whereas the latter ones to a previous estimation. Moreover also the vectors related to the updates are penalized more to avoid a noisy MVF estimation and to be less affected by periodic patterns.

2.3 Binarized cross correlation

PC methods have several advantages. They provide a sub-pel estimation accuracy, they are quite robust to noise and illumination changes and the match is performed in the frequency domain with a point by point multiplication. Unfortunately the Fourier Transform introduces some spatial constraints. The same block size has to be used for the reference and the target blocks and the signal is supposed to be infinite and periodic. In fact the match in the frequency domain correspond to a circular-shift in the spatial domain. For this reason the largest reliable vector that can be measured correspond to half size of the block. In this case already half matched samples are invalidated from the samples at the other side of the block. Then PC methods can estimate reliable MVs only when objects are greater than their movement. This is not true in case of small moving objects in the scene and therefore a selection between the maximum measurable movement and the vector accuracy has to be chosen. This issue can be overcome by means of a CC. The

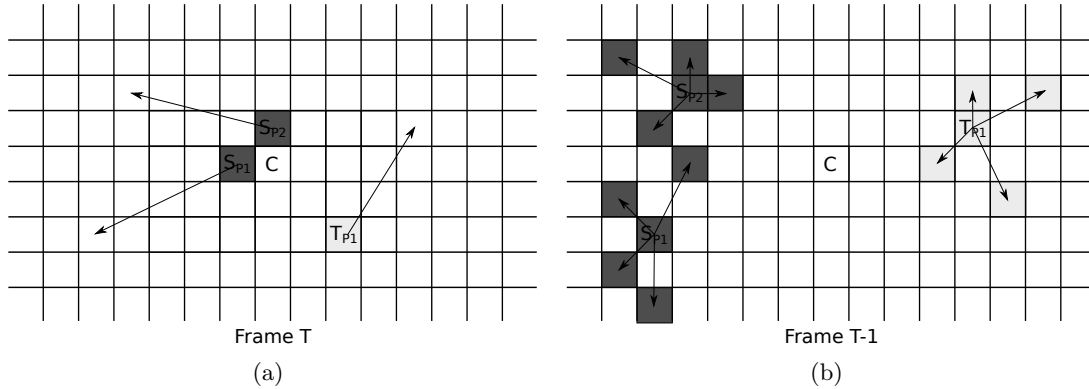


Figure 2.3: Update star in the block based RS with SAD matching criteria: (a) the spatial and temporal predictors of the current block, C, are selected. (b) In the SAD based match a random update with Gaussian distribution is added to the vector value of the predictors forming a set of final candidates.

advantages of the spatial domain are evident. The signal is not required to be infinite and periodic and the matching is then not impaired by circularly repeated samples. Moreover the reference block size and the search range in the target frame are now independent since it is possible to match a small window into a larger area. The disadvantage of the CC refers to the higher computational cost. A possible solution to reduce this effort is to apply the CC on binarized images. There are several techniques to binarize images. The method chosen in this thesis is a moving average window thresholding since it is not so much affected by blur of moving edges and by illumination changes. For every pixel position on the image the mean intensity value is calculated in a surrounding window:

$$m(x, y) = \frac{1}{(2w + 1)^2} \sum_{i=-w}^w \sum_{j=-w}^w I(x + i, y + j). \quad (2.4)$$

Then the binarized image is computed by comparing the pixel intensity value against the mean calculated in (2.4):

$$B(x, y) = \begin{cases} 0 & \text{if } I(x, y) < m(x, y) \\ 1 & \text{if } I(x, y) \geq m(x, y). \end{cases} \quad (2.5)$$

The result of the binarization process is shown in Figure 2.4. In particular Figure 2.4(a) shows the reference image in gray-scale value of the *Grove2* sequence from the Middlebury database [18] and Figure 2.4(b) shows the corresponding binarized image. The binarized cross correlation (BCC) method is summarized in Figure 2.5. In this figure a ROI from the binarized reference, a ROI from the binarized target frame of the *Grove2* sequence and the corresponding binarized correlation are shown. The red block of Figure 2.5(a) is searched in the red window of Figure 2.5(b). For every possible shift (s_x, s_y) over a predefined set (S_x, S_y) a binarized block difference (BBD) is calculated:

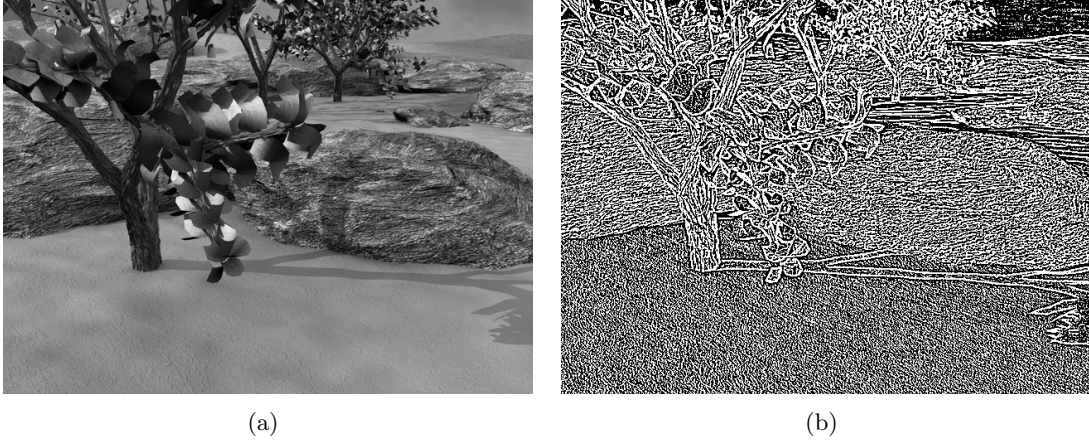


Figure 2.4: Image binarization: (a) Original reference frame in gray-scale, (b) Binarized reference frame with a 5×5 moving average threshold.

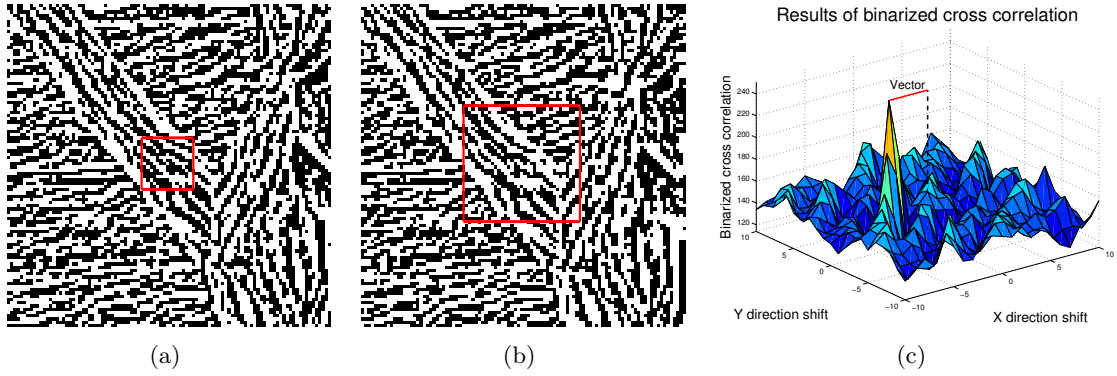


Figure 2.5: Binarized cross correlation matching: (a) ROI of the reference binarized image, the area inside the red rectangle correspond to the block to match, (b) ROI of the target binarized image, the area inside the red rectangle correspond to the search area and (c) Results of the binarized cross correlation, the offset between the peak and the center of the correlation surface is the selected vector.

$$BBD(s_x, s_y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B(i, j, t) \oplus B(s_x + i, s_y + j, t + 1). \quad (2.6)$$

This can be efficiently implemented with a sum of N^2 XOR operations, where N is the considered block size, between the binarized values of the reference block $B(i, j, t)$ and the binarized values of the target block $B(s_x + i, s_y + j, t + 1)$ shifted by the (s_x, s_y) position. Then the vector (V_x, V_y) , see Figure 2.5(c), is chosen selecting the shifting set (s_x, s_y) that maximizes value of $BBD(s_x, s_y)$:

$$(V_x, V_y) = \arg \max_{(s_x, s_y)} BBD(s_x, s_y). \quad (2.7)$$

2.4 Recursive search real motion estimation based on binarized cross correlation

The RS method and the BCC described in the previous Section can be combined as it is shown in Figure 2.6. In particular we propose a method that follows the recursion

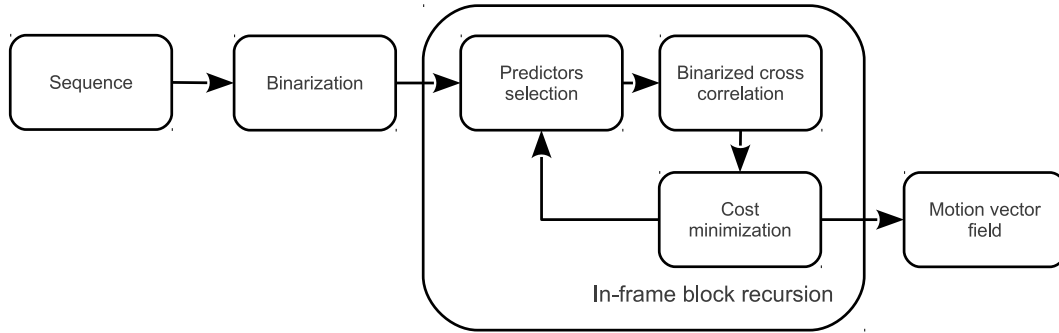


Figure 2.6: Block base RS framework with BCC as matching criterion: the image is in advance binarized and the update is substituted by a BCC.

predictor scheme already represented in Figure 2.2(b) but it applies to every predictor position the BCC presented in Section 2.3. Figure 2.7(b) shows in dark and light gray the areas that are now considered with this new matching criteria. It is possible to notice how a larger number of possible candidates can be tested but, thanks to the binary matching, the computational cost is even reduced. After the complete scanning a correlation surface, as the one shown in Figure 2.5(c), is computed for every predictor. Every point of this surface corresponds to a similarity measure between the different shifts in the target area and the reference block. For every predictor the shift that maximizes the BBD is considered (2.7) and the offset between the peak and the center of the correlation surface is the selected

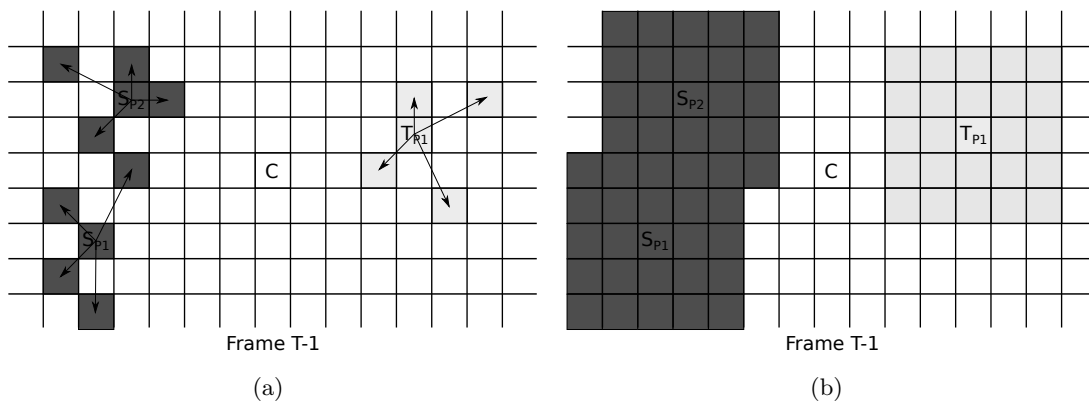


Figure 2.7: Comparison between SAD and BBD matching in RS: (a) In the SAD based match a random update with Gaussian distribution is added to the vector value of the predictors forming a set of final candidates. (b) In the BBD based match a BCC is performed for every predictor.

update that has to be added to the vector predictor. Eventually the combination of the predictor vector (V_{Px}, V_{Py}) with the previously selected update that maximize the BBD is chosen as final MV:

$$(V_x, V_y) = \underset{(V_{Px}+s_x, V_{Py}+s_y)}{\arg \max} BBD(V_{Px} + s_x, V_{Py} + s_y). \quad (2.8)$$

2.5 Parallel motion estimation - MePar

Due to the intra-frame recursion the RS algorithm can not be implemented efficiently in software. In fact the vector estimation for the current block relies on the already estimated blocks, which depends on the scanning technique. Then, in order to have a fully parallel algorithm, the spatial recursion has to be omitted and only MVs from the previous frame have to be considered. In this way all the blocks can be iterated independently and at the same time. Since no assumptions on the MVF can be done in advance and all the predictors belongs to the previous estimation we opted for the symmetrical predictor

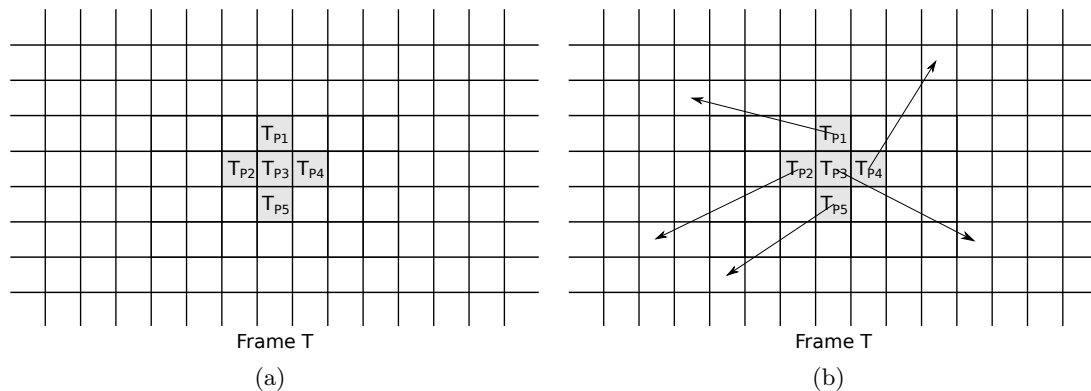


Figure 2.8: Predictor selection: (a) Predictor scheme in the proposed solution, since no intra-frame recursion is used, all the predictors have a MV that belongs to a previous frame or iteration, light gray. (b) Example of possible motion vectors.

scheme shown in Figure 2.8. Then the procedure follows the standard RS approach. A random update star is added to every predictor forming the final candidates set shown in Figure 2.9(a). From this set the most reliable MV is selected minimizing the SAD based similarity function (2.1). Due to the lack of intra-frame recursion the convergence time of the estimation clearly increases. To reduce this issue a higher number of predictors or updates may be considered. This has the drawback of an increased computational cost. In order to obtain a larger number of candidates while keeping a comparable computational cost (see Figure 2.9 and Table 2.2) we opted again for the BBD as matching criteria. Then the frame is in advance binarized (2.5) with a local moving average (2.4) and the final MV for each block is selected maximizing the BBD of equation (2.8). Eventually, in order to exploit the temporal consistency, at the end of each frame a vector projection of the MVF

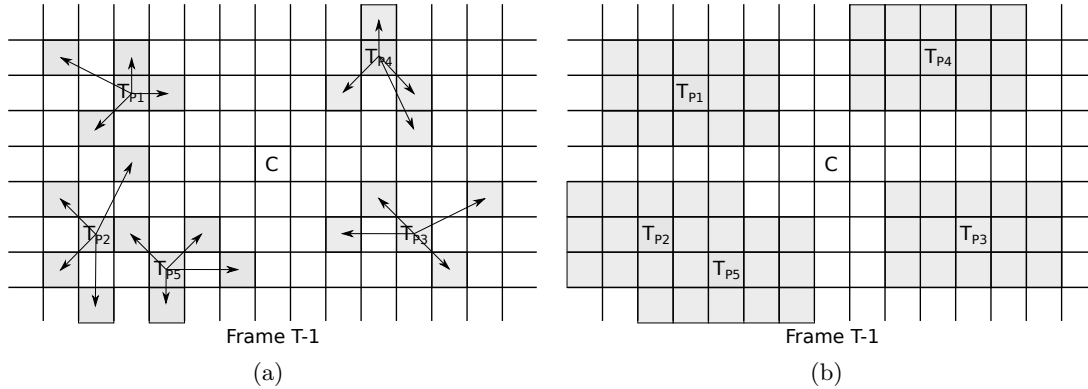


Figure 2.9: Comparison between SAD and BBD matching: the temporal predictors of the current block, C , are selected. (a) In the SAD based match a random update with Gaussian distribution is added to the vector value of the predictors forming a set of final candidates. (b) In the BBD based match a BCC is performed for every predictor.

is computed. In this way an initial estimation for the subsequent frame becomes available increasing the convergence speed.

2.6 Experimental results

In our experiments we compared the RS approach against our parallel ME and for both the estimators the two different matching criteria, based on SAD or BBD match, were also considered. In all the cases an 8×8 block size was adopted. Moreover, for both the SAD based estimations a totality of five updates per predictor and the predictor itself were evaluated as possible candidates. For the RS ME based on BCC a shift of ± 3 pixels in x direction and ± 1 pixels in y direction were considered, while for the parallel ME based on BCC a shift of ± 4 pixels in x direction and ± 2 pixels in y direction were adopted. Then a search window of 7×3 and 9×5 pixels, respectively were used. In particular the larger searching window for the parallel implementation was necessary to overcome the lack of the intra-frame recursion. The test was made on the Middlebury database [18] which provides a set of different sequences. In particular, except for the *Dimetrodon* sequence which is formed by only two frames, all the others are formed by a totality of eight frames and with an available ground truth MVF related to the fourth frame. An objective evaluation is then possible. This is measured with two different metrics: the angular error (AE) and the absolute flow endpoint error (EE) [18]. The AE provides a relative measure which penalizes less the errors for high motion measuring the angle between the ground truth vector (GT_x, GT_y) and the estimated one (V_x, V_y):

$$AE = \cos^{-1} \left(\frac{1 + V_x \times GT_x + V_y \times GT_y}{\sqrt{1 + V_x^2 + V_y^2} \sqrt{1 + GT_x^2 + GT_y^2}} \right). \quad (2.9)$$

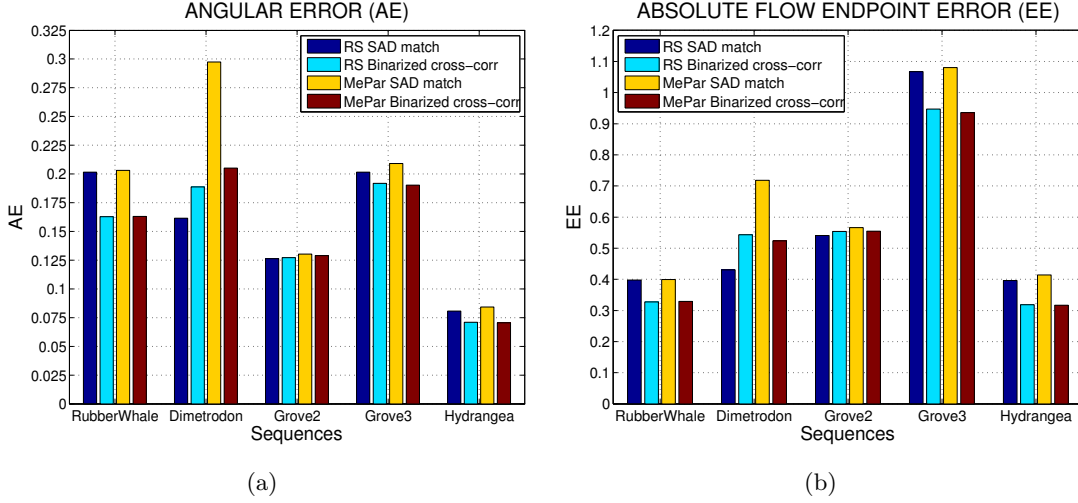





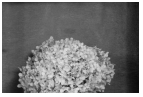

Figure 2.10: Results comparison of the RS and MePar based on the SAD and BCC matching criteria: (a) Angular error results, (b) Absolute flow endpoint error results.

This may over-penalize errors in regions of zero motion with respect to errors in regions of smooth motion. The EE tries to solve this issue penalizing the errors independently to the motion level:

$$EE = \sqrt{(V_x - GT_x)^2 + (V_y - GT_y)^2}. \quad (2.10)$$

The results are shown in Figure 2.10 for the AE and the EE respectively. Both the metrics present a similar trends. The results from the RS and the MePar based on SAD matching criteria are comparable, as well for the RS and the MePar based on BCC. Indeed by means of the BCC a more precise estimation along the objects borders is achieved, while the SAD match leads to a coarser estimation resolution in these areas. This is in particular clear in the *Rubber Whale* and *Grove3* sequences, see Figures 2.11 and 2.13. However the BCC has the drawback of a slightly more noisy estimation in the homogeneous regions. This was also expected because the proposed method (2.7) does not use penalties that regularize the SAD (2.1) based results in those areas. It is important to notice that the worst results for the parallel approach are obtained in the *Dimetrodon* sequence, for which only one estimation is not enough to reach a MVF convergence. In fact another important factor in the MV estimation is convergence time. This is the number of frames the estimator needs to provide a stable MVF. As already mentioned, the Middlebury dataset [18] provides the ground truth MVF only for a single frame. Then an objective evaluation of the convergence time for every sequence is not possible. To address this issue a subjective evaluation of the convergence time for each sequence is reported in Table 2.1. It is possible to notice that in general the MePar based on BCC matching criteria reduces the number of frames necessary to obtain a stable MVF in comparison to the same estimator configuration using a SAD criterion. However the omission of the intra-frame recursion in the proposed parallel method increases the convergence time.

Table 2.1: MVF convergence time (number of frames).

Method	Sequence				
					
	Dimetrodon	Grove 2	Grove 3	Hydrangea	Rubber Whale
RS (SAD)	1	1	1	1	1
RS (BBD)	1	1	1	1	1
MePar (SAD)	-	3	3-4	1	1
MePar (BBD)	1	2	3-4	1	1

This is definitely a problem that needs to be addressed, e.g., by means of a global motion estimator and multiple or hierarchical estimation approaches, and a trade-off between the convergence time and the computational cost should be considered. In Table 2.2 we also provide a computational effort comparison between the SAD and the BCC matches in term of 8-bit pseudo-code commands. We decided to not consider the cost of the binarization since it is comparable to the low pass filter that is usually performed in advance to a SAD based RS true motion estimation. For both the SAD based approaches and considering an 8×8 block size there are 64 subtractions (SUB), 64 absolute values (ABS) and 63 additions (ADD) per match. Considering the BCC, there are 8 XOR, which correspond to the 64 bit of the binarized block, 8 Count bit, which sum the number of the bits at 1 in one byte, and 7 additions per match. Because in the SAD match we perform 6 matches for every predictor, at the original predictor and at the 5 updates positions, we obtain a final number of operations per predictor of 1146. For the BCC in the RS ME we use a 7×3 searching window, while for the MePar ME a larger window of 9×5 is adopted to address the lack of intra-frame recursion. The total operations per predictor are then 483 and 1035, respectively. Finally it should be noticed that, since we process a binarized frame and less memory is required, the bandwidth constraints are also more relaxed.

Table 2.2: SAD and BCC comparison (number of operations).

Method	RS		MePar	
Match Type	SAD	BCC	SAD	BCC
Block Size	8×8	8×8	8×8	8×8
Operations	64 SUB 64 ABS 63 ADD	8 XOR (8bit) 8 Count bit 7 ADD	64 SUB 64 ABS 63 ADD	8 XOR (8bit) 8 Count bit 7 ADD
TOT per Block-Match	191	23	191	23
TOT per Predictor	1146	483	1146	1035

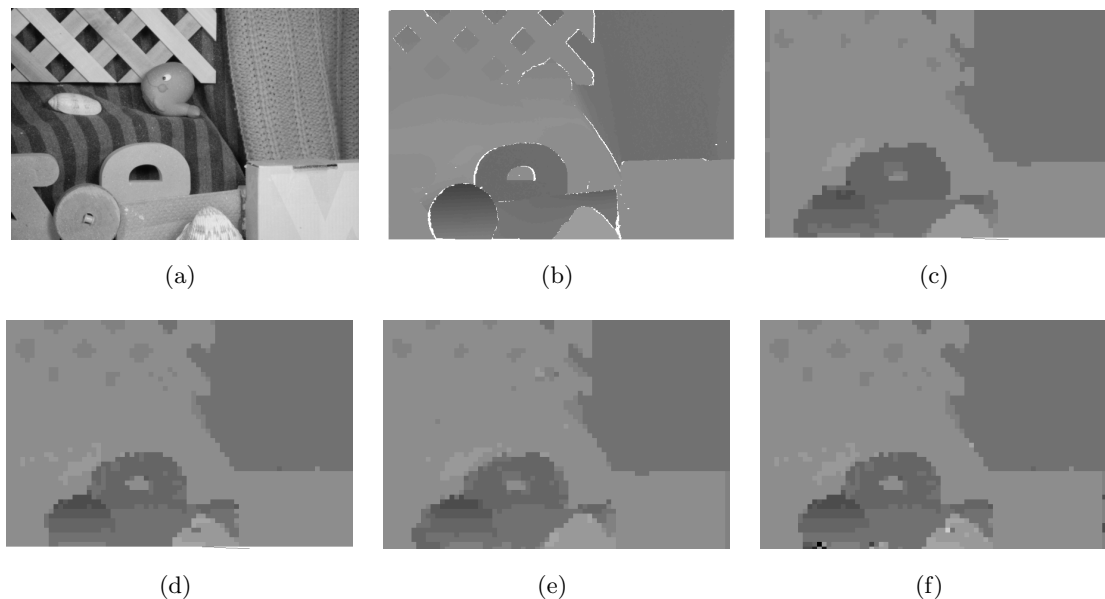


Figure 2.11: Results comparison in RubberWhale sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for RS with SAD matching, (d) MVF for RS with BBD matching, (e) MVF for MePar with SAD matching and (f) MVF for MePar with BBD matching.

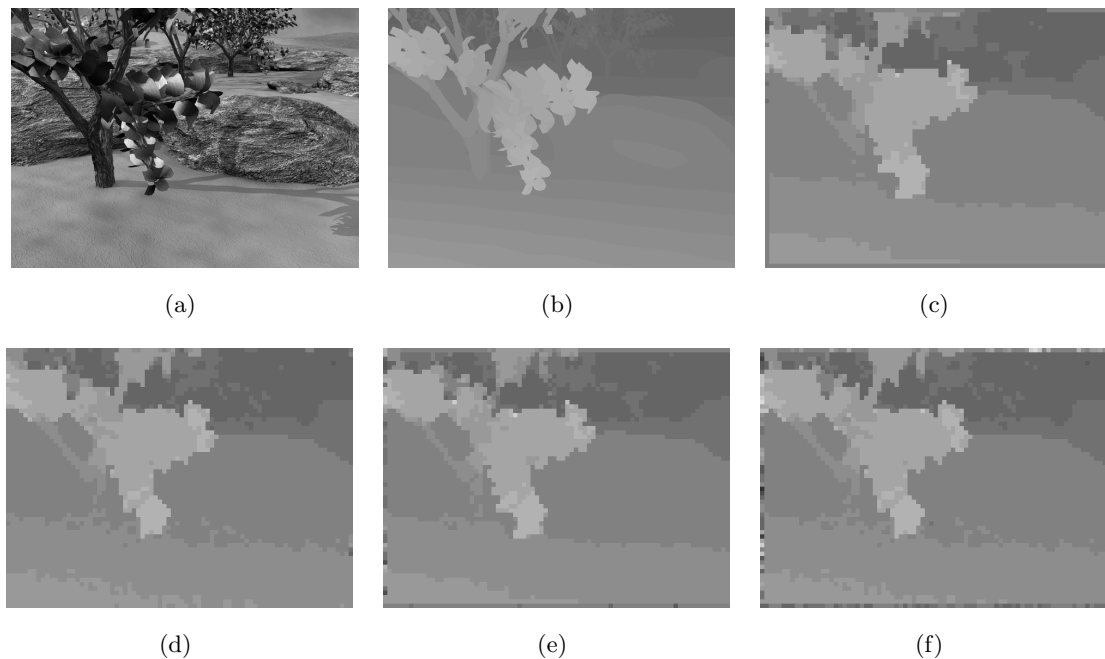


Figure 2.12: Results comparison in Grove2 sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for RS with SAD matching, (d) MVF for RS with BBD matching, (e) MVF for MePar with SAD matching and (f) MVF for MePar with BBD matching.

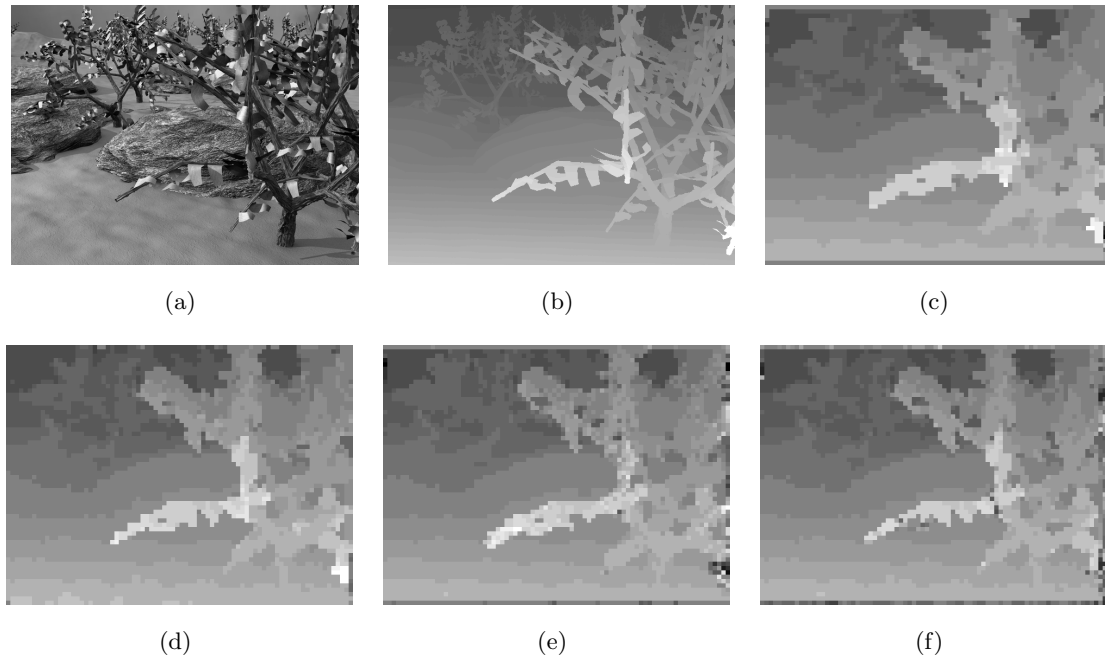


Figure 2.13: Results comparison in Grove3 sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for RS with SAD matching, (d) MVF for RS with BBD matching, (e) MVF for MePar with SAD matching and (f) MVF for MePar with BBD matching.

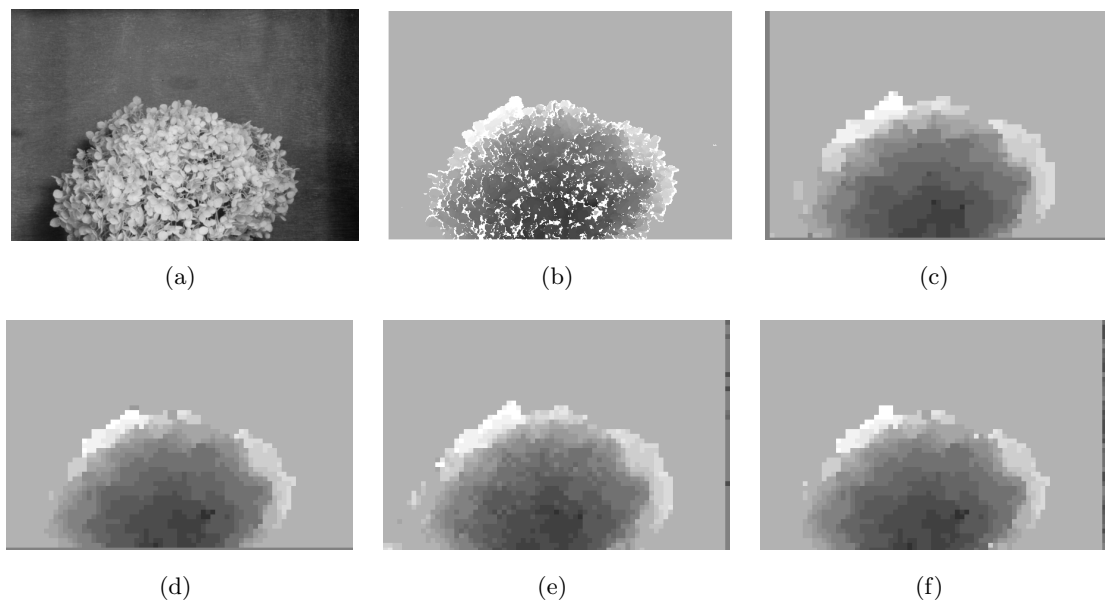


Figure 2.14: Results comparison in Hydrangea sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for RS with SAD matching, (d) MVF for RS with BBD matching, (e) MVF for MePar with SAD matching and (f) MVF for MePar with BBD matching.

2.7 Conclusions

In this Chapter a novel method for motion estimation was proposed. This avoids the intra-frame recursion typical of the RS systems to permit a parallel implementation. Moreover the combination with the BCC as matching criteria allows a faster convergence of the MVF whereas the computational complexity is kept low by means of an image binarization. The results show a comparable performance with respect to the classical RS SAD based approach with considerable advantages in terms of implementation due to the algorithm parallelism.

Chapter 3

Disparity Estimation

3.1 Introduction

Disparity estimation between a stereo image pair is a fundamental problem in the field of computer vision. With this process is possible to extract information of the three-dimensional structure of the scene. In particular there exists a relation between the distance of the object from the cameras (depth) and the difference between the position of the object in the two images (disparity). As it is shown in Figure 3.1 each camera has a 2D camera coordinate system (CCS) and a 3D CCS [19].

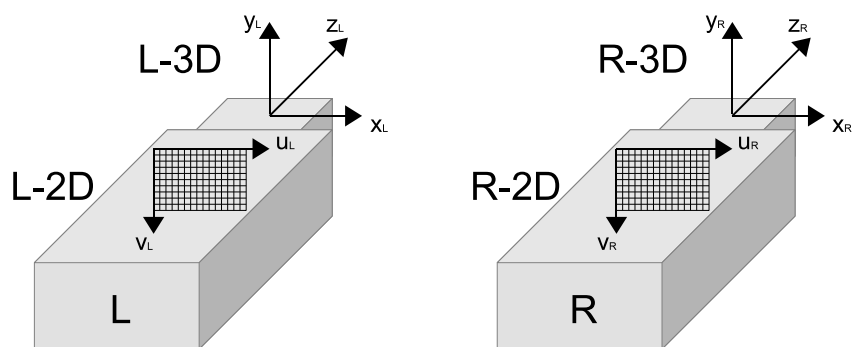


Figure 3.1: Schematic representation of the 2D CCS and 3D CCS associated to the left and the right cameras [19].

The 2D CCS illustrates the pixels coordinates in the image plane and the 3D CCS describes the positions of scene points with respect to the camera itself. The disparity then represents the difference in pixels between the position of an object in the left 2D CCS (L-2D CCS) and the right 2D CCS (R-2D CCS). The depth related to the object may be then obtained from the knowledge of the disparity [20] by means of a camera calibration procedure [21–23]. This is used to estimate the cameras *intrinsic* and the *extrinsic* parameters. The intrinsic parameters describe the transformation between the 3D CCS and the 2D CCS and are usually denoted in the matricial form as K_{CAM} . The

extrinsic parameters encode the transformation between the two different 3D CCSs and are composed by a rotation matrix R and a translation vector t . After the calibration, the two images are generally rectified in order to transform each image plane such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. The rectified images can be thought of as acquired by a new stereo pair obtained by rotating the original cameras in order to generate two coplanar image planes that are also parallel to the baseline. This simplifies the stereo matching algorithm since the search domain of corresponding pixels is restricted from a 2D problem (horizontal and vertical) to a 1D problem (usually horizontal). Eventually, the process of image rectification also compensates the projective distortion introduced by the camera lenses and the focal length difference between the stereo camera pair [20, 24]. Specifically, after the rectification process, a scene point $W = (x, y, z)$, expressed with respect to the L-3D CCS, that is visible from both the cameras, can be projected into the point w_L for the L-2D CCS and into the point w_R for the R-2D CCS. Due to the image rectification, the coordinate of the points w_L and w_R are respectively (u_L, v_L) and $(u_R, v_R) = (u_L + d, v_L)$, where d is the disparity. Moreover it is possible to prove [20] that the depth information z is inversely proportional to the corresponding disparity d ,

$$z = \frac{bf}{d}, \quad (3.1)$$

where b is the baseline in meters (i.e., the distance between the two camera nodal points) and f is the focal length in pixel. Then an object close to the camera (low depth z) has a high value of disparity d , whereas an object far from the camera (high depth z) has a low value of disparity. Moreover deriving equation (3.1) with respect to d the following relation can be obtained,

$$\Delta z \cong -\frac{bf}{d^2} \Delta d, \quad (3.2)$$

where Δd and Δz are respectively the disparity and depth resolutions. Eventually, combining equations (3.1) and (3.2) is possible to notice that the depth resolution decreases quadratically with respect to z ,

$$\Delta z \cong -\frac{z^2}{bf} \Delta d. \quad (3.3)$$

Once the disparity of all the points in a image is obtained, forming a disparity map, this can be easily transformed in a depth map by means of equation (3.1). It is fundamental to recall that equation (3.1) holds true only in case of rectified images. This Chapter assumes this particular situation to be true and it is organized as follow. In Section 3.2 the main categories of algorithm used for disparity estimation are presented. Section 3.3 introduces and evaluates a particular global algorithm for disparity estimation, the Simple Tree stereo algorithm [2]. A possible improvement to this algorithm is proposed in Section 3.4 where also an evaluation between the two methods is presented. Eventually the conclusions are drawn in Section 3.5.

The method presented in this chapter was applied for a patent in [25].

3.2 Disparity estimation algorithms

Disparity estimation is one of the most investigated topics in the field of computer vision. The aim is to detect the correspondences in a stereo pair image. As already mentioned, in case of rectified images the search domain of corresponding points can be reduced to a 1D problem. Indeed thanks to the rectification the epipolar lines coincide with the image scan-lines. A wide number of stereo algorithms are available in literature and they can be subdivided into three main classes *local*, *global* and *semi-global*. In particular the different types can be generalized as a trade-off between low computational complexity and robustness.

3.2.1 Local stereo algorithms

In local methods the disparity for every point is typically computed by maximizing a local similarity function in a winner take all (WTA) strategy. In general the SAD [26–28], the normalized cross correlation (NCC) [29], or the census transform [30] on the gray scale images are used as similarity functions [31]. Moreover in [32, 33] also the extension of the previous metrics into the RGB color space has been considered.

3.2.2 Global stereo algorithms

Global stereo algorithms compute the disparity D by imposing constraints on the whole image. In general they formulate the problem as a global energy cost minimization:

$$\arg \min_D (E_{data}(I_1, I_2, D) + E_{smooth}(D)). \quad (3.4)$$

The first term measure the data similarity between the two images I_1 and I_2 under the condition of a certain disparity. The smoothness term is used to regularize the estimation under the assumption that, in general, a disparity map presents smooth regions separated by sharp transitions along object borders. Many different methods have been developed to solve equation (3.4). This can be based on the combination of dynamic programming (DP) in different directions [2], or since the disparity map is commonly represented as a Markov random field (MRF) [34] which can be solved by means of belief propagation [35] and graph-cuts [36] techniques.

3.2.3 Semi-global stereo algorithm

The semi-global stereo algorithms follow the global formulation, but they limit the constrains only to a portion of the image. Examples of algorithms that perform this strategy are the one based on DP over scan-lines [37] and the one based on DP over scan-lines and diagonals [38]. Indeed the relaxed constraints reduce the computational cost.

Obviously if the two images are not rectified the estimation has to be applied in two dimensions. This quadratically increases the computational complexity especially for the global or semi-global stereo algorithms. Then, in case of high resolution images, the only feasible estimations relies in the local algorithms. In particular the bidimensional displacement estimation is then similar to the one that is used for the motion estimation, see Chapter 2. This particular case is treated more in details in Chapter 5.

3.3 Simple Tree stereo algorithm

The Simple Tree stereo algorithm is a quite accurate method for the disparity estimation which does not rely on a previous image segmentation. Specifically this can be subdivided in the three main steps depicted in Figure 3.2. The correlation between two

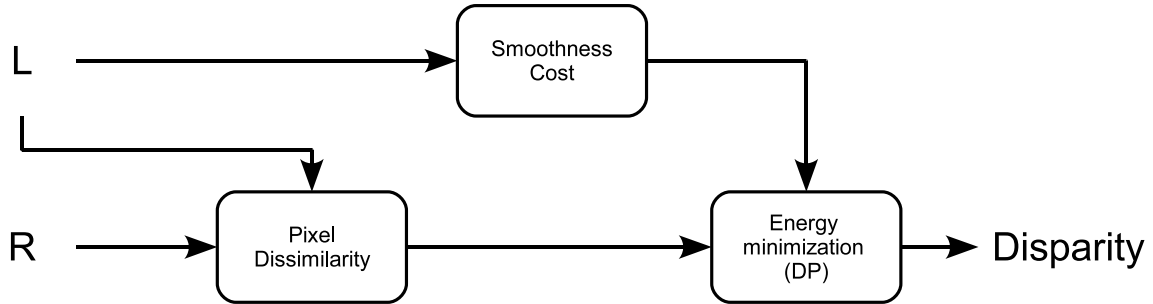


Figure 3.2: Simple Tree framework.

pixels is calculated with the Sampling-intensive measurement of Birchfield and Tomasi [39]. This measurement of pixel dissimilarity is quite accurate for homogeneous depth regions, but may fail along the object borders. To overcome this issue a smoothness cost function, a modified Potts model [2], is used. Then the two terms are combined in the following equation:

$$E(D) = \sum_{p \in I} m(p, d_p) + \sum_{(p,q) \in N} s(d_p, d_q). \quad (3.5)$$

Eventually the disparity D that minimizes the equation is selected. In particular the minimization is processed by means of DP over the combination of two trees structures, one vertical and one horizontal. This is the main contribution of the authors and for more details the reader is referred to the original publication [2]. What is important to notice is the modified Potts model used in the smoothness cost function. This provides a different set of penalties depending on the possible disparity jumps:

$$s(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ P_1 & \text{if } |d_p - d_q| = 1 \\ P_2 & \text{if } |d_p - d_q| > 1. \end{cases} \quad (3.6)$$

Specifically the penalty for disparity jumps that are larger than one pixel is driven by a threshold of a first derivative $|I_p - I_q|$ calculated in the RGB colorspace:

$$P_2 = \begin{cases} P_3 & \text{if } |I_p - I_q| < T \\ P_4 & \text{if } |I_p - I_q| \geq T. \end{cases} \quad (3.7)$$

This is based on the assumption that different colors are usually linked to different objects. Unfortunately, as it is shown in Figure 3.6(b), the first derivative, and consequently the penalty P_2 , is easily invalidated by noise or texture areas. A possible solution to this issue, which exploits the information provided by an external user, is presented in the following Section. For a more detailed explanation regarding the values used for the penalties the reader is again referred to the original publication [2].

3.4 User assisted processing

In high quality video processing, several applications work offline, requiring the user to annotate specific image conditions. In case of disparity estimation, the delivered vectors might not be perfect in quality, in the sense that they do not fit properly the objects. An user can approximately mark the borders of objects, where the vectors are too imprecise. In this context a second automatic process can take over from where the user left and bring the final quality to an acceptable level. In particular the framework for the proposed solution is shown in Figure 3.3. After an initial disparity calculation the user has to

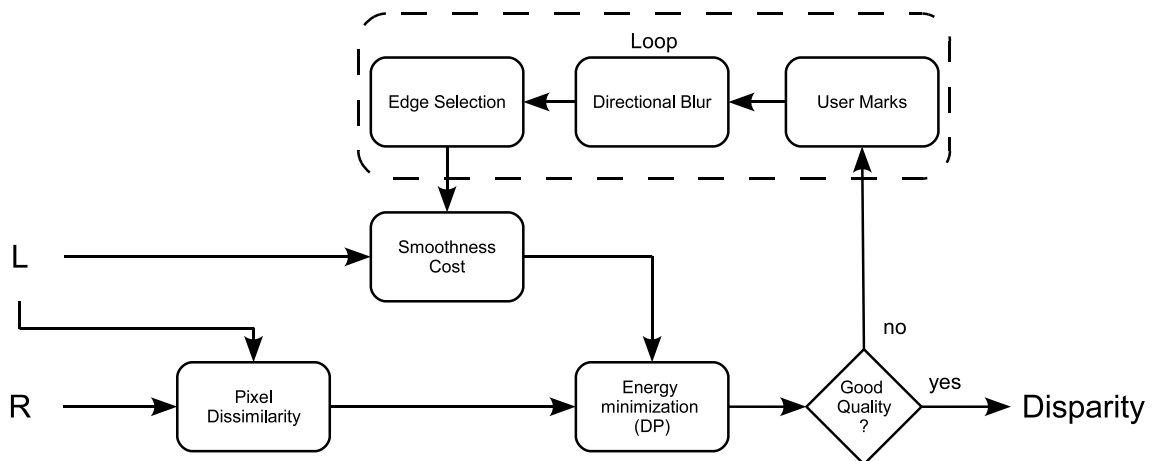


Figure 3.3: User assisted processing framework.

judge the quality of the results. This can be done by overlaying the disparity itself to the respective image as it is shown in Figure 3.4(a).

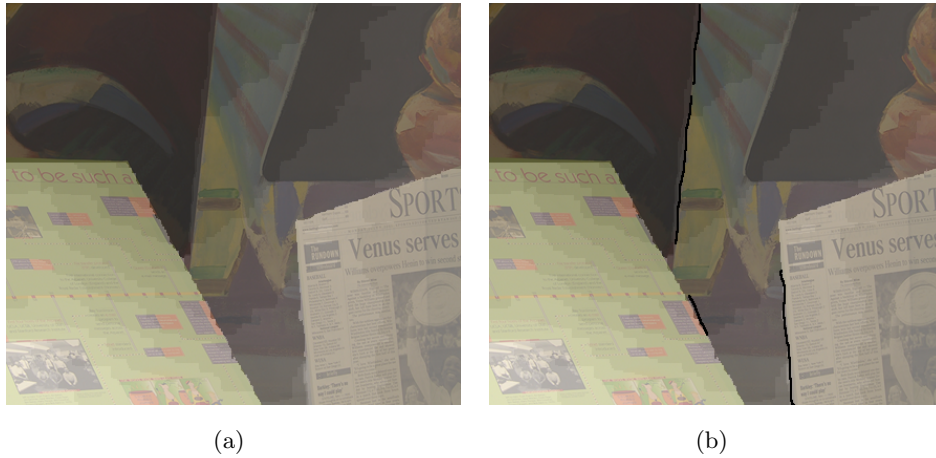


Figure 3.4: Example of user interaction: (a) Simple tree estimated depth overlaid to the image, (b) User marks (black).

Then the user can marks with some drawings the errors in the estimation, see Figure 3.4(b). This information can be used to apply a directional filtering which reduces noise and texture in the edge orthogonal direction. Eventually, the edge can be easier located and the smoothing constraints can be consequently updated.

3.4.1 Directional filtering

Sometimes the borders of an object do not correspond to the sharpest edge in the neighborhood. This happens very often in texture or noise areas where there is not a dominant direction. For this reason it should be possible to use the direction information provided by the user in order to filter the neighborhood of the user marks with a directional low-pass filter. This should increase the chance to detect the real object border. Specifically a directional smoothing with a Gaussian kernel is applied. This can be stretched along a predefined angle by means of the following equations [40]:

$$f(x, y) = A \exp \left(- \left(a(x - x_0)^2 + 2b(x - x_0)(y - y_0) + c(y - y_0)^2 \right) \right), \quad (3.8)$$

where

$$\left\{ a = \frac{\cos^2 \theta}{2\sigma_x^2} + \frac{\sin^2 \theta}{2\sigma_y^2}, b = \frac{\sin 2\theta}{4\sigma_x^2} - \frac{\sin 2\theta}{4\sigma_y^2}, c = \frac{\sin^2 \theta}{2\sigma_x^2} + \frac{\cos^2 \theta}{2\sigma_y^2} \right\}. \quad (3.9)$$

In equations (3.8) and (3.9) A is a normalization term, θ is the angle between the horizontal and the edge directions, σ_x and σ_y are the standard deviation in the direction of the filter and in the orthogonal direction respectively. It is important to notice that the term θ can also vary at every line. Thus the filter can really follow the objects boundaries. This is proved from the filtering results shown in Figure 3.5(b). It is possible to see how the sharpness of the borders is preserved while the surrounding textures and orthogonal structures are strongly smoothed.



Figure 3.5: Results of directional filtering: (a) Reference image, (b) Reference image after the application of the directional Gaussian filter.

3.4.2 Directional binarization

After the smoothing the image borders are detected. For this purpose an iterative row based binarization is performed. This can be summarized in the following steps. The region of the row for which the smoothing was applied is binarized and the binarization output is evaluated. When there is one and only one transition between the two segments in the considered interval the procedure ends. Indeed in this case the binarization can be considered stable. When this is not the case the first and the last values in the array are discarded and the binarization is reprocessed. Finally this procedure is repeated until a single transition is found or the interval reaches a minimum size. This is in particular visible in the lower right corner of Figure 3.6(a). In fact it is possible to see how the convergence size of the interval is reduced in correspondence to the black stripe for the title. The binarization is performed with two methods. The first uses the luminance information while the second one employs the chrominances components in the CIELab color space. In the first case the data are thresholded with their mean. Sometimes the luminance results too noisy or the edge can be easily found exploiting the color information. This can be represented by at least a couple of values, in our case the two chrominances (a, b) . Then the binarization has to be adapted to a bidimensional case. Since in the monodimensional situation the mean is used to minimize the variance of the set, we have to find the line which minimizes the variance of the distance between the point and line itself and that passes through the mean point:

$$\begin{cases} \frac{\partial}{\partial m} \left(\frac{\sum_i (d_i - d_m)^2}{N} \right) = 0 \\ y_m - mx_x - q = 0 \end{cases} \quad \text{where} \quad \begin{cases} (x_m, y_m) = \left(\frac{1}{N} \sum_i x_i, \frac{1}{N} \sum_i y_i \right) \\ d_m = \frac{1}{N} \sum_i d_i = \frac{1}{N} \sum_i \frac{y_i - mx_x - q}{\pm\sqrt{1+m^2}} \end{cases} \quad (3.10)$$

This results in a second order equation for m and the related q :

$$\left\{ \begin{array}{l} m^2 \left[-2x_m y_m + \frac{2}{N} \sum_i x_i y_i \right] + m \left[-2x_m^2 + 2y_m^2 + \frac{2}{N} \sum_i x_i^2 - \frac{2}{N} \sum_i y_i^2 \right] + \\ \quad + \left[2x_m y_m - \frac{2}{N} \sum_i x_i y_i \right] \\ q = y_m - m x_x. \end{array} \right. \quad (3.11)$$

Finally, the results of the two binarization procedures are combined by means of the final interval size. Indeed this gives an indication of the stability in the binarization procedure since a wider interval stabilizes the mean value. The identified edges with their corresponding convergence intervals are shown in Figure 3.6(a).

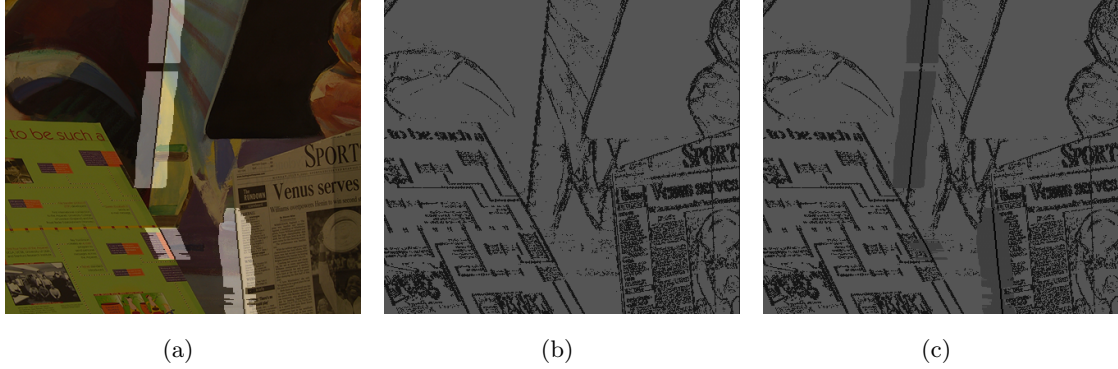
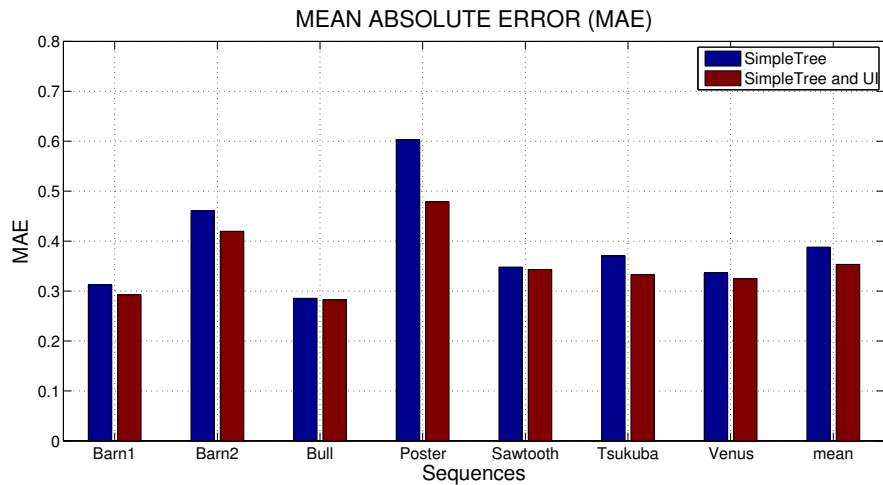


Figure 3.6: P_2 penalty calculation: (b) Based on the thresholded first derivative in the RGB colorspace, (c) Based on the thresholded first derivative in the RGB colorspace and the directional binarization.

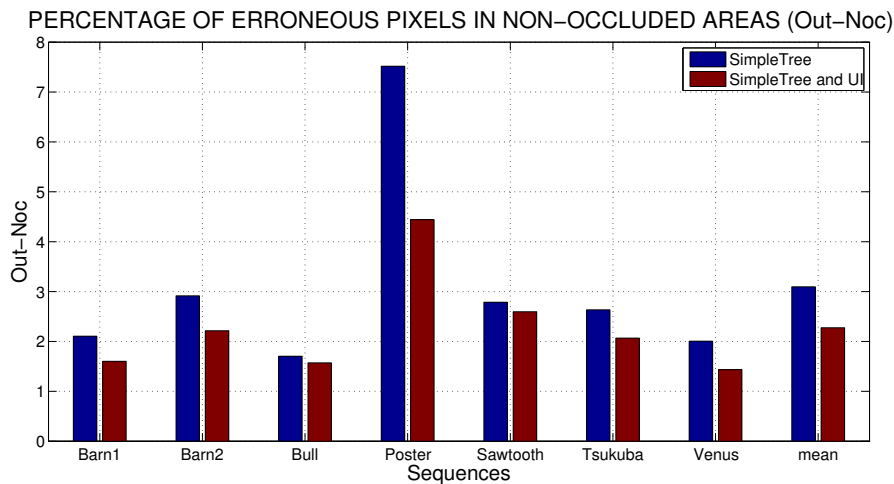
These are then used to update the calculation of the penalty P_2 in (3.7). Specifically at the border position the penalty is strongly reduced while for the surrounding convergence interval a higher value was set, see Figure 3.6(c). This permits to avoid the estimation of wrong disparity jumps in case of texture or noise close to the real depth discontinuities.

3.4.3 Experimental results

In our experiments we compared the Simple Tree approach against the modified version based on the bias of an user. The test was made on the 2002 Middlebury stereo database [41]. This provides a set of different stereo images and the related ground truth disparity. An objective evaluation is then measured by means of two different metrics: the mean absolute error (MAE), see Figure 3.7(a), and the percentage of erroneous pixels in non-occluded areas (Out-Noc), see Figure 3.7(b).



(a)



(b)

Figure 3.7: Results comparison of the Simple Tree algorithm and user interface (UI): (a) Mean absolute error, (b) Percentage of correctly estimated pixels, with threshold $th = 1$.

In particular for the Out-Noc a threshold of 1 pixel was set. Since the process was developed to refine the estimated disparity on the objects borders the MAE does not show a significative, even if visible, improvement. Indeed the MAE averages the results over the whole image and the object borders correspond to a minor portion of it. The Out-Noc metrics is more relevant for the evaluation. In fact the Simple Tree algorithm already provides noticeable results and a wrong disparity estimation around an object border easily produces an error which is higher than the set threshold. Moreover the advantage of the proposed refinement is also demonstrated in the visual comparison. Specifically, a subset of the tested images with the related ground truth and the estimated disparities are shown in Figures 3.8, 3.9, 3.10 and 3.11. Is possible to see how with a simple modification of the smoothness constraint the disparity jumps can be realigned with the object borders.

In the *Tsukuba* sequence some marks were applied along the head and in the lower part of the lamp. In both the *Barn1* and *Poster* sequences the border of the triangle was marked. For all the sequences the estimated disparity is more close to the ground truth.

3.5 Conclusions and future work

In this Chapter it was proposed a possible improvement to the Simple Tree stereo algorithm. In particular the algorithm was analyzed and the weak point was identified in the smoothness cost calculation. Indeed the penalty associated to a disparity jump is related to a fixed threshold of the first derivative in the the RGB colorspace. The first derivative is easily affected by noise. Then we proposed an user interaction (UI) based framework to semi-automatic improve the quality of the results. Specifically a directional filtering is performed to smooth the regions in the orthogonal direction to the user marks. This increases the chance to locate the correct border of a row based binarization that exploit the luminance or the chrominances information. Finally the smoothness cost is updated and the disparity is recalculated. Objective measurements show the effectiveness of the proposed method. It should also be noticed that the framework presented for the UI could be also suitable for a hierarchical refinement approach. The disparity jumps and the associated directions may be detected in a coarse scale estimation and then refined in a full size estimation.

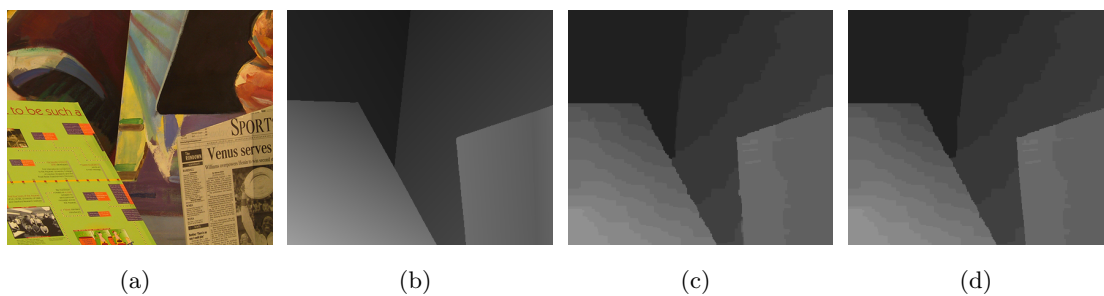


Figure 3.8: Results comparison with the Venus images: (a) Left image, (b) Ground truth disparity, (c) Simple tree estimated disparity, (d) Simple tree biased by user estimated disparity.

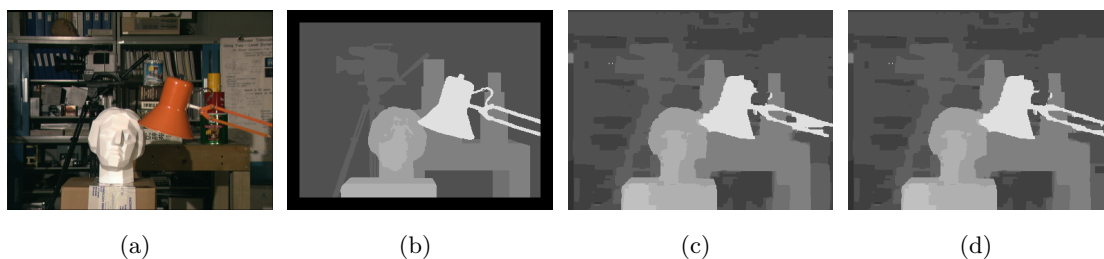


Figure 3.9: Results comparison with the Tsukuba images: (a) Left image, (b) Ground truth disparity, (c) Simple tree estimated disparity, (d) Simple tree biased by user estimated disparity.

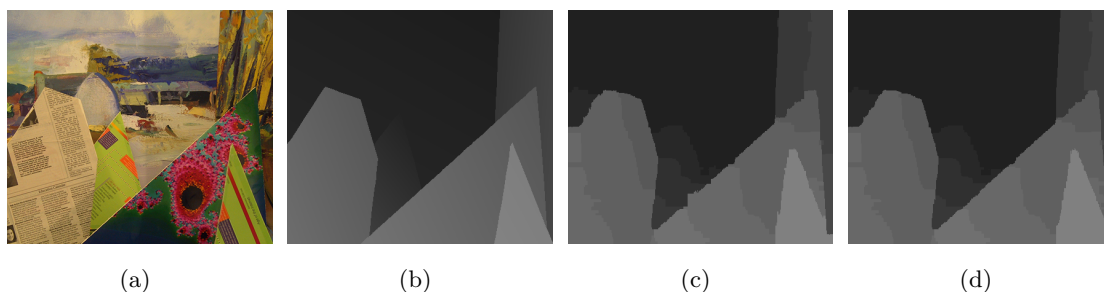


Figure 3.10: Results comparison with the Barn1 images: (a) Left image, (b) Ground truth disparity, (c) Simple tree estimated disparity, (d) Simple tree biased by user estimated disparity.

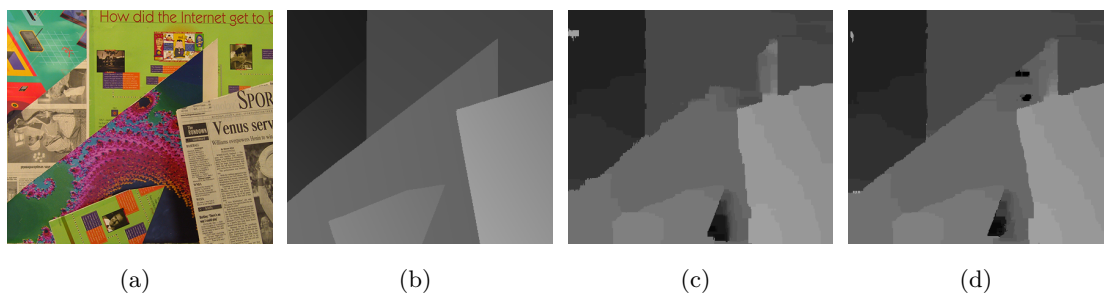


Figure 3.11: Results comparison with the Poster images: (a) Left image, (b) Ground truth disparity, (c) Simple tree estimated disparity, (d) Simple tree biased by user estimated disparity.

Chapter 4

Hybrid System

4.1 Introduction

The 3D analysis of a scene is one of the most important field in computer vision. Different techniques, e.g., object detection, position determination and 3D reconstruction, require a precise scene geometry estimation. A variety of methods are available for this purpose but these can be subdivided in two main classes: active and passive systems. Passive systems estimate the scene geometry without an interaction with the scene, this is the case of the disparity estimation in a stereo setup presented in Chapter 3. Active systems, e.g., *laser range scanners*, *structured light cameras* and *time of flight range cameras*, actively interact with the scene to reconstruct the related depth. This interaction could be based on the active triangulation principle (*laser range scanners*, *structured light cameras*) or on the time-of-flight (ToF) principle. In the latest period methods that fuse active and passive systems (hybrid systems) have been developed. In particular the combination of a ToF and standard cameras is one of the most interesting and investigated topic due to their complementary characteristics. ToF cameras are not sensitive to the scene peculiarity, but they provide only low resolution depth maps. Moreover the most significant artifact occurs at depth boundaries and in region with low infrared reflectance. In the stereo system a high resolution geometry estimation can be obtained, but the quality strongly depends on the scene. Accurate results are provided at the object boundaries while this is not possible for homogeneous regions. Therefore, a collaborative approach, which exploits the best characteristics of the two systems, may allow to reach a better quality of the final three-dimensional scene sensing. Many approaches have been proposed to fuse the information of one ToF camera with a single color camera [42–47] or with multiple cameras [19,48]. Finally also methods that adopts multiple ToF sensors and multiple color cameras have been considered [49]. When only a standard camera is adopted the geometry information is uniquely available from the ToF. Then the main purpose is to increase the low resolution of the ToF depth considering the related color information provided by the camera. In this case methods that adapt the bilateral filter technique [50]

have been proposed. In [46] a joint bilateral filter is adopted. This method performs well along the depth borders but may introduce artifacts in the reconstructed geometry that can be attributed to a not perfect correlation between color and depth data. The approach in [47] addresses the problem by means of a weighting factor in the filtering process to differentiate between homogeneous depth regions and object boundaries. In [43] the authors propose to combine the ToF and color camera informations in a cost volume with an iterative approach. Then at every step the resolution is increased and a bilateral filter is applied along the cost volume. Other methods exploit the geometry and color relation to perform a super-resolution driven by segmentation [42] or based on a Markov random field approach [44]. Finally in [45] a method to solve the problem of mapping between the ToF and standard camera image planes is presented. The authors propose a solution that combines a PMD ToF chip and a traditional camera charge-coupled device (CCD) in a single device by means of an optical splitter.

The aim of this Chapter is to exploit the information coming from a camera rig composed by a PMD ToF camera and three standard video-cameras in order to increase the low-resolution ToF depth up to the camera image resolution. An overview of the ToF principle and a comparison with the stereo system are treated in Sections 4.2 and 4.3 respectively. Then an ideal model, based on the Middlebury database [41], is proposed in Section 4.4. This simplified model is then used to compare a novel method against [46] and [47] thanks to the ground truth availability. This approach is then applied to a real case and Section 4.5 evaluates the problems that arise when some sequences are acquired with the available camera rig. In particular a complete scheme to obtain a super-resolved depth map is proposed. This is composed by a joint camera calibration of the camera rig, a preprocessing of the ToF depth to detect and to remove unreliable values and an iterative super-resolution approach based on the method developed for the ideal model. Eventually in Section 4.6 we draw some conclusions.

The methods presented in this Chapter were applied for a patent in [51]. Furthermore, parts of the work presented here has already been published by A. Vianello in his MSc thesis [24] for which the reader is referred for a more accurate description.

4.2 Time-of-flight cameras

Matricial ToF range cameras are relatively new active sensors which allow the acquisition of 3D point clouds at video frame rates. The most known ToF manufacturer are the *PMDTec* [52], *Mesa Imaging* [53], *SoftKinetic* [54] and *Microsoft* [55]. In particular images of the MESA SR4000 [53] and the PMD CamCube 3.0 [52] are shown in Figure 4.1. The ToF cameras provide a depth measurement that is based on the time-of-flight principle. The time-of-flight τ_d is the time that the light needs to cover the distance d from the light source to an object and from this object back to the camera. Then, considering

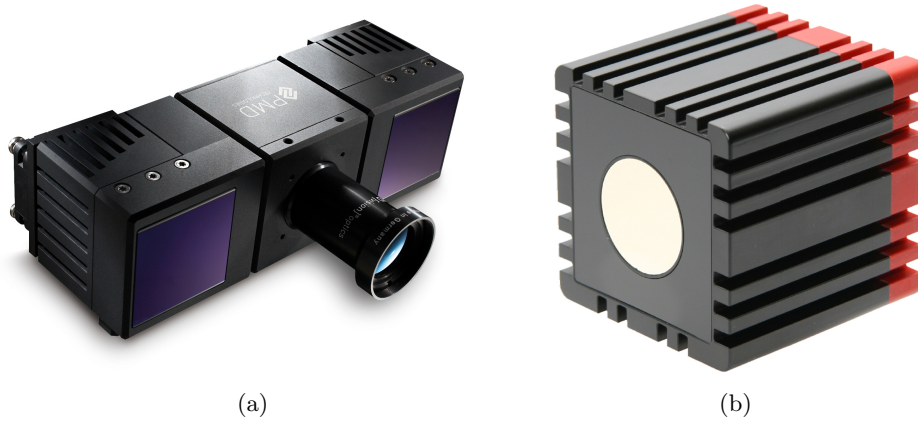


Figure 4.1: Examples of ToF camera models: (a) CamCube 3.0 from PMD Technologies GmbH, Germany [52], (b) ASR4000 from MESA Imaging AG, Switzerland [53].

c the light speed ($c = 3 \cdot 10^8 [m/s]$), τ_d can be computed as

$$\tau_d = \frac{2d}{c}. \quad (4.1)$$

Actually two types of ToF cameras are available: pulse-based and phase-based, also known as continuous wave (CW) ToF [56]. Since the available hardware is a PMD CamCube 3.0 only the latter ToF type will be considered. A CW ToF emits an IR optical signal $s_E(t)$ with amplitude A_E and modulated by a sinusoid of frequency f_{mod} :

$$s_E(t) = A_E[1 + \sin(2f_{mod} \cdot t)]. \quad (4.2)$$

The signal is reflected by the target scene surface and travels back to the camera sensor positioned near the emitter. Then the received signal $s_R(t)$ can be defined by the following equation:

$$s_R(t) = A_R[1 + \sin(2f_{mod} \cdot t + \Delta\phi)] + B_R. \quad (4.3)$$

The A_R is the attenuated amplitude due to the losses in the transmission path. Moreover the non-instantaneous propagation of the signal introduces a phase delay $\Delta\phi$ and the B_R is an intensity factor which mainly depends on the additional background light [57]. Equation 4.3 is in general simplified considering $A = A_R$ and $B = A_R + B_R$:

$$s_R(t) = A \sin(2f_{mod} \cdot t + \Delta\phi) + B. \quad (4.4)$$

The quantity A and B are important for SNR evaluation and they are usually referred as amplitude and intensity respectively despite they are both IR radiation amplitudes. The phase delay $\Delta\phi$ is instead used to estimate the depth d :

$$\Delta\phi = 2\pi f_{mod} \tau_d = 2\pi f_{mod} \frac{2d}{c} \Rightarrow d = \frac{c}{4\pi f_{mod}} \Delta\phi. \quad (4.5)$$

4.2.1 CW ToF Cameras: typical distance measurement errors

Many errors affect the estimation of the distance in a CW ToF. These can be generally subdivided in two main categories: random errors, e.g., *photo-shot noise*, *multipath effect*, *flying pixels*, and systematic errors, e.g., *harmonic distortion and phase wrapping*.

The photo-shot noise describes the arrivals of photons on the sensor. This is statistically characterized by a Poisson distribution. However the related estimation of the distance d distribution can be approximated [58] by a Gaussian with standard deviation

$$\sigma_d = \frac{c}{4\pi f_{mod} \sqrt{2}} \frac{\sqrt{B}}{A}. \quad (4.6)$$

The multipath effect occurs when the incident ray is reflected from a non-specular surface into multiple directions. This is represented in Figure 4.2 where the incident ray is shown

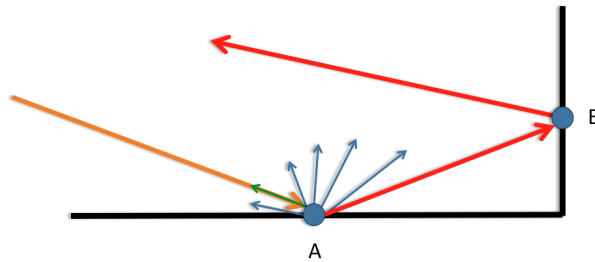


Figure 4.2: Multi-path effect: the emitted ray (orange) hit the surface at point A and is reflected in multiple directions (blue and red rays). The red ray reaches then B and travels back to the ToF camera affecting the distance measured at the sensor pixel relative to B.

in orange, the reflected ray in a perfect scenario in green, and the rays reflected in other possible directions in blue. In particular when one of the latter rays, as for the case of the red one, firstly hit others scene objects and then travel back to the ToF sensor the depth measurements are impaired by the combination of direct and indirect paths.

The flying pixels problem can be described by the effect shown in Figure 4.3. To each ToF

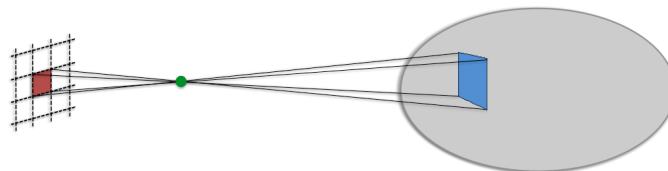


Figure 4.3: Motivation of flying pixels problem: to every ToF sensor pixel (red) is associated a finite size scene area (blue).

pixel sensor is associated a finite size area of the scene. Then the estimated depth is a convex combination between the different depth levels of the area. This is not a problem in case of regions with constant distance values but in case of an object boundary some artifacts becomes visible as it is shown in Figure 4.4.

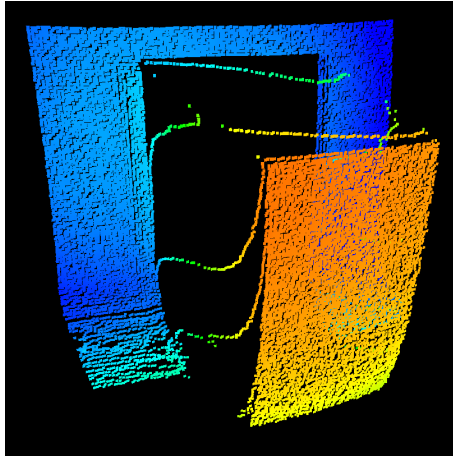


Figure 4.4: Example of flying pixels problem: the depth measurements at an object borders are a convex combination between the different depths levels.

The harmonic distortion relies on the sinusoids construction. In fact they are obtained applying a low-pass filtering on the squared wave-forms emitted by the LEDs. This produces a systematic offset which can be compensated or at least reduced by means of a look-up-table (LUT) correction.

The phase wrapping is associated to the phase shift estimation ambiguity. This leads to an estimation range limited to $[0, d_{MAX}] = \left[0, \frac{c}{2f_{mod}}\right]$. Then every object located at a distance $d > d_{MAX}$ will be estimated to a wrong depth $d' = \text{mod}(d, d_{MAX})$.

The problems considered in this section are only a minimal part of the errors that affect the CW ToF measurement. For a more complete explanation the reader is referred to [24] and in particular to [59].

4.3 ToF Cameras and Stereo System: comparison and combination

In Chapter 3 and in the previous Sections of this Chapter it was possible to deduce that the ToF cameras and the stereo vision system own complementary features. The advantages and the disadvantages of the two systems are in fact summarized in Table 4.1

Table 4.1: Advantages and disadvantages of a stereo vision system and ToF cameras: The two systems can be considered complementary.

	Stereo	ToF
Image resolution	✓	X
Depth discontinuities	✓	X
Flat areas	X	✓
Occlusions	X	✓
Depth resolution	$\Delta z \cong \frac{z^2}{bf} \Delta d$	$\Delta z \sim C$

In the ToF the image sensor resolution is relatively small (e.g., 200×200 in the PMD CamCube 3.0) and the flying pixels problem affects the estimation along depth discontinuities, see Figure 4.4. The stereo system provides a much higher image resolution (e.g., 1920×1080 in the Lux Media Plan (LMP) HD1200 [60]) but the results are scene dependent. A quite precise estimation may be obtained in textured and at the object borders regions, while the homogeneous areas decrease the stereo performance. Moreover the different cameras positions produce occlusion regions for which an estimation is not possible. On the other hand, the ToF performs best in areas without depth discontinuities and, since it is a monocular system, it does not suffer from occlusion problem. The depth accuracy of the ToF can be considered constant as first approximation (actually the noise slightly increase with the distance) along the working interval, while this exponentially decrease in a stereo estimation.

Due to their complementary features, a collaborative approach that take advantage of the strong points of each system can be developed. The first problem to be addressed in the fusion of a ToF with a stereo vision system is the great difference in term of image sensors resolution. In order to overcome this issue many techniques for super-resolution of the ToF depth map driven by a single or multiple cameras have been recently developed [19, 42–49]. Due to the lack of a real dataset with an associated ground truth, a simplified model of the ToF has been considered. This is presented in the following section.

4.4 Ideal model for depth super-resolution

To evaluate the precision of different super-resolution algorithms a simplified model of the ToF has to be established. For this purpose we used the Middlebury 2006 dataset [61]. This is a database with 21 pairs of rectified color images with an associated ground truth disparity obtained with structured light techniques. Moreover we decided to consider only the photon-shot noise and consequently an approximated Gaussian distribution as in [19]. Then a perfect mapping between ToF and video-camera images was also assumed. In this scenario the evaluation consists to recover a ground truth depth map starting from a down-sampled and noised version of it and the associated high resolution color images. In particular the down-sampled and noised depth map simulates the ToF estimation while the latter refers to the cameras images. An additive Gaussian noise with several levels of standard deviation was used, specifically $\sigma_N \in \{0, 0.5, \dots, 9.5, 10\} [cm]$. Then, the reconstruction quality was evaluated by computing the Mean Absolute Error (MAE) between the reconstructed depth D_R and the ground truth depth D_{GT} :

$$MAE = \frac{1}{N} \sum_{u,v} |D_R(u,v) - D_{GT}(u,v)|. \quad (4.7)$$

Finally the test was carried out with the methods proposed in [46], in [47], and with a novel algorithm which combines and extends the previous procedures and which will

be introduced in Section 4.4.3. In particular the filtering was computed after an initial upsampling with a random tiles approach [24] for which every depth value is randomly assigned to an $a \times a$ block, where a is the upsampling factor, creating an initial sparse high resolution depth map D_S .

4.4.1 Joint bilateral filter

The bilateral filter (BF) was introduced by Tomasi and Manduchi in [50] as a non-linear filter which combines two kernels, a spatial and an range one. The first depends on the distance to the central pixel position while the second weights the intensity differences. Thanks to the kernels combination this technique is effective for image denoising, achieving a reduction of the the unwanted artifacts while preserving the important image content, such as edges. Recently also the joint bilateral filter (JBF) has been introduced. This is an evolution of the BF in which the range kernel is not driven by the intensity differences in the image to be filtered but is driven by a second guidance image, from which the *joint* terminology derives. In [46] the authors apply the JBF for super-resolution purposes and in particular they increase the resolution of a depth map driving the process with the associated higher resolution color image:

$$JBF[D_S]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(\left| \tilde{I}_{\mathbf{p}} - \tilde{I}_{\mathbf{q}} \right|) D_{S,\mathbf{q}}. \quad (4.8)$$

In equation (4.8) \mathbf{p} and \mathbf{q} denote the pixel coordinates in the guidance image \tilde{I} and D_S respectively and $JBF[D_S]_{\mathbf{p}}$ is the up-sampled dense depth map obtained from the low resolution depth map $D_{S,\mathbf{q}}$. Moreover G_{σ_s} and G_{σ_r} are the spatial and the range Gaussian kernels, while $W_{\mathbf{p}}$ is a normalization factor. Finally S is the filter aperture.

4.4.2 “Kim” Joint bilateral filter

The direct application of JBF for depth super-resolution may be affected by artifacts, e.g., *texture copying*, in the reconstructed geometry. These can be attributed to erroneous assumption about the correlation of color and depth data. A depth map generally presents smooth regions separated by sharp transitions along object borders while a color image presents a much higher variety. In particular the presence of textures in the color image may invalidate the Gaussian intensity difference term $G_{\sigma_r}(\cdot)$ of equation (4.8). In this situation small weights are associated to pixels from the same object and with the same depth. To overcome this issue, in [47] the authors propose an up-sampling JBF (from now denoted with JBF_{KIM}) which includes a weighting parameter γ that depends on the

depth difference Δ_S :

$$JBF_{KIM}[D_S]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} \left[(1 - \gamma(\Delta_S)) G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) + \gamma(\Delta_S) G_{\sigma_r}(|\tilde{I}_{\mathbf{p}} - \tilde{I}_{\mathbf{q}}|) \right] D_{S,\mathbf{q}}. \quad (4.9)$$

This is used to give different weights to the range kernel $G_{\sigma_r}(\cdot)$, based on color intensity difference, and to the spatial kernel $G_{\sigma_s}(\cdot)$. More in details γ belongs to the interval $[0, 1]$ and it is defined as:

$$\gamma(\Delta_S) = \frac{1}{1 + e^{-\varepsilon(\Delta_S - \tau)}}, \quad (4.10)$$

where Δ_S is the difference between the maximum and the minimum depth values in the pixels neighborhood S , whereas ε and τ are two constant parameters. In case of a smooth depth region, as inside an object, γ is close to 0 and so the filter's response depends almost entirely on the Gaussian depth difference. This is indeed necessary to avoid the texture copying. Vice versa in an area with objects borders γ is close to 1 and the filter's response is biased more from the Gaussian color intensity difference.

4.4.3 Weighted joint bilateral filter

The proposed weighted joint bilateral filter (WJBF) extends and combines both the already presented procedures [46, 47]. The up-sampling of the low resolution depth map is driven by the high resolution color image with two different kernels, according to the area characteristics:

$$WJBF[D_S]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} \left[(1 - \alpha) G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_{r,FLAT}}(|\tilde{I}_{\mathbf{p}} - \tilde{I}_{\mathbf{q}}|) + \alpha G_{\sigma_{r,EDGE}}(|\tilde{I}_{\mathbf{p}} - \tilde{I}_{\mathbf{q}}|) \right] D_{S,\mathbf{q}}. \quad (4.11)$$

The homogeneous depth areas are processed with a JBF for which a relaxed range parameter $\sigma_{r,FLAT}$ is used to avoid the texture copying problem. At the depth boundaries, under the assumption that depth discontinuities usually coincide with color image intensity edges, a Gaussian color intensity difference term with a very selective range parameter $\sigma_{r,EDGE}$ is used to obtain sharp edges. Finally the weighting factor α attempts to solve the main issue of [47] in which the correspondent γ was depending on two constant thresholds.

Weighting factor

Recalling that the photon shot noise can be modeled as Gaussian [58], it can be shown [24] that the related standard deviation increases with the distance. This standard deviation provides a value of depth accuracy of the ToF depth. In particular in our model we decided to approximate the ToF noise as a combination of this Gaussian

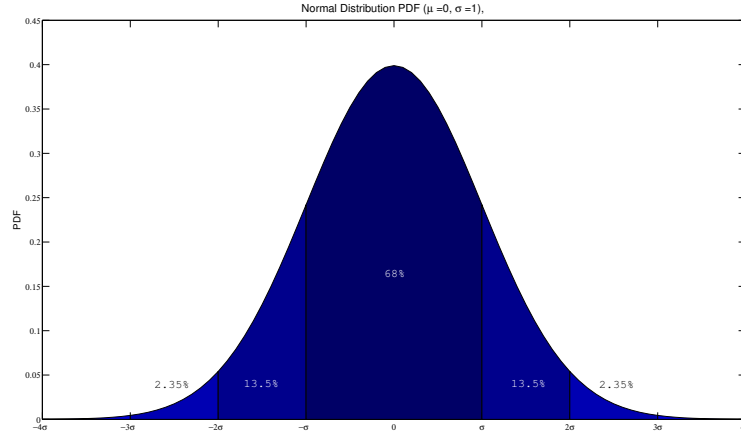


Figure 4.5: Three-sigma rule: most of the values are between $-\sigma$ and σ , almost all of them are no farther than 2σ from μ and there are virtually no observations farther than 3σ from μ [62].

contribution with an additional general source of noise Φ , *e.g.* flying pixels, multipath or phase wrapping.

$$NOISE_{ToF} = \mathcal{N}(0, \sigma_D) + \Phi. \quad (4.12)$$

In particular since we are in the *ideal model*, the additional contribution Φ is only produced by the depth jumps. Since the Gaussian noise distribution $\mathcal{N}(0, \sigma_D)$ at each distance can be calculated experimentally [24], we can assume to have this information. Then the *three-sigma rule* [62] can be applied to derive the weighting factor α . As it is shown in Figure 4.5, nearly all values lie within the interval of $\pm 3\sigma$ around the mean value μ . Specifically, around 68.27% of the values lie within $\pm\sigma$ of the mean, 95.45% within $\pm 2\sigma$ of the mean and 99.73% within $\pm 3\sigma$ of the mean. Then a local standard deviation in the sparse depth map can be estimated as following

$$\sigma_S = \sqrt{\frac{1}{N-1} \sum_{\mathbf{q} \in A} (\mathbf{q} - \bar{\mathbf{q}})^2}, \quad (4.13)$$

where $\bar{\mathbf{q}}$ is the mean value and N is the number of non zero values in the considered area A . For each standard deviation value σ_S of the standard deviation image Σ_S the weighting factor α is calculated by comparing σ_S with the standard deviation σ_N of the depth image Gaussian noise. When $\sigma_S < 2\sigma_N$ the factor Φ in Equation (4.12) can be considered negligible and the depth variation can be accounted due to the normal camera noise only, hence α is set to 0. Vice versa, if $\sigma_S > 5\sigma_N$ most likely this high variation is due to a real depth discontinuity, hence α is set to 1. Between $2\sigma_N$ and $5\sigma_N$ the weighting factor α has a linear behavior. Figure 4.6 shows the two steps for the weighting factor calculation from the sparse depth D_S .

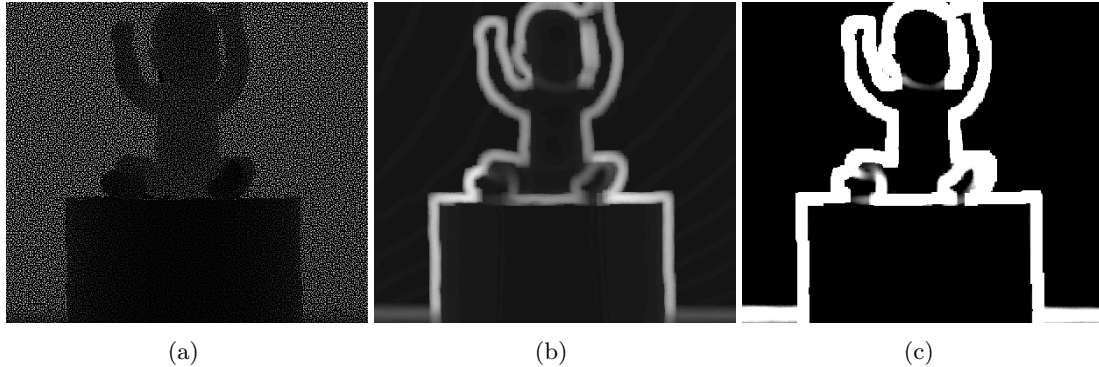


Figure 4.6: Creation of the weighting factor α image: (a) sparse depth D_S , (b) standard deviation image Σ_S , and (c) α image.

4.4.4 Experimental results

As already mentioned, the evaluation was performed on the Middlebury 2006 dataset [61] and with the *JBF* [46], the *JBF_{KIM}* [47] and the proposed *WJBF* method. In particular we report the results for the *aloe*, *baby1* and *wood2* images. The filters' parameters for the super-resolution are:

- **Joint Bilateral Filter:** Filter size 15×15 pixels, spatial parameter $\sigma_s = 5$ and range parameter $\sigma_r = 0.03$.
- **Kim Joint Bilateral Filter:** Filter size 15×15 pixels, spatial parameter $\sigma_s = 5$ and range parameter $\sigma_r = 0.03$. For the blending function of Equation (4.10), the parameter $\varepsilon = 0.5$ and τ is set to $15[cm]$.
- **Weighted Joint Bilateral Filter:** Filter size 15×15 pixels, spatial parameter $\sigma_s = 5$, range parameter for flat areas $\sigma_{r,FLAT} = 0.1$ and range parameter for edge areas $\sigma_{r,EDGE} = 0.03$. For the calculation of the sparse depth map standard deviation Σ_D the same window size of the filter is used. From Σ_D , the weighting factor α is computed as explained in Section 4.4.3 and when $\sigma_N = 0[cm]$ the thresholds are calculated using $\sigma_N = 0.5[cm]$.

Figures 4.7, 4.9, and 4.11 show the three scenes up-sampled using the three filters with four different levels of noise. From a visual inspection it can be appreciated how the proposed *WJBF* outperforms the other two filters. In the *JBF* based results the texture copying problem is quite evident and this becomes even more clear when the noise level increases. In similar manner, for high noise levels the *JBF_{KIM}* detect depth discontinuities in flat areas, hence it transfers textures from the guidance color image in the final up-sampled depth. With the *WJBF* the weighting factor α allows to distinguish between edge and flat areas, hence the texture copying problem is less noticeable than for the other two filters.

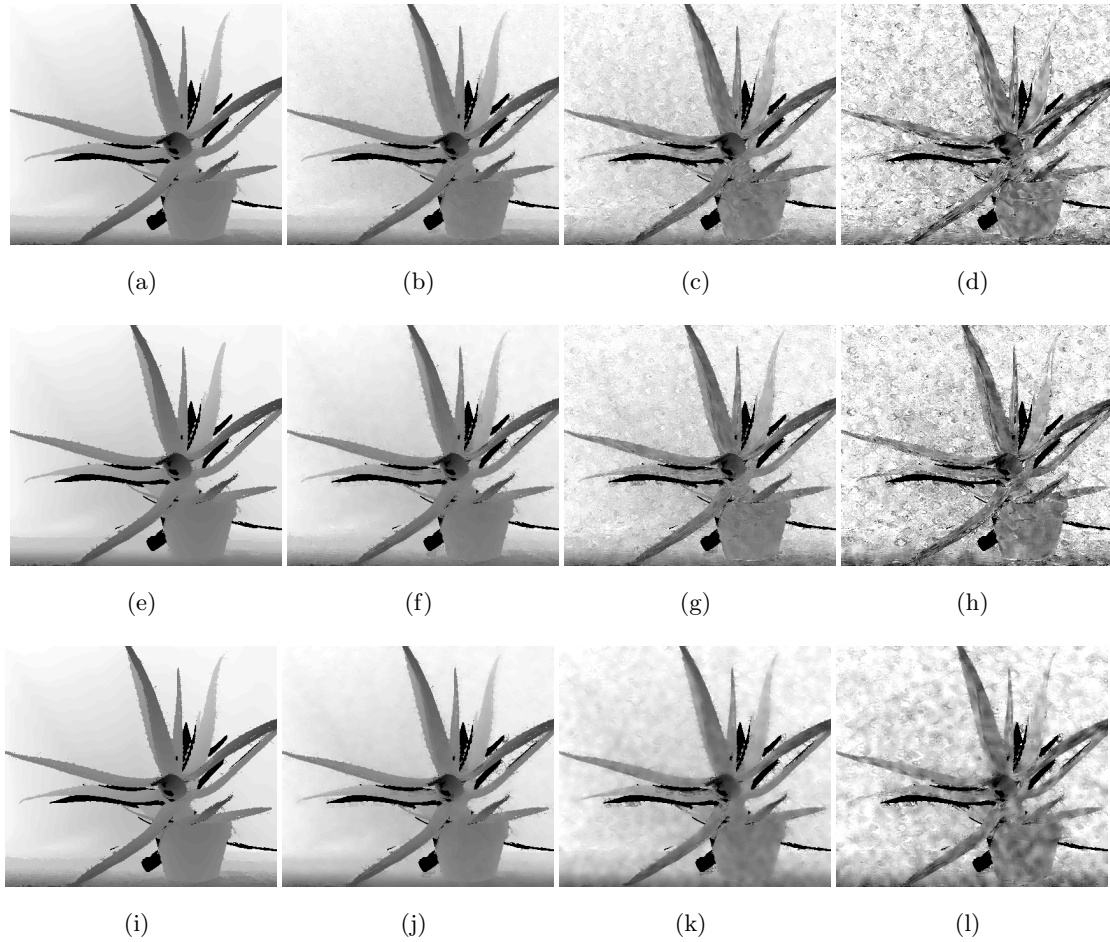


Figure 4.7: Visual comparison of the *aloe* scene super-resolution: (a) JBF $\sigma_N = 0[cm]$, (b) JBF $\sigma_N = 2[cm]$, (c) JBF $\sigma_N = 5[cm]$, (d) JBF $\sigma_N = 10[cm]$, (e) JBF Kim $\sigma_N = 0[cm]$, (f) JBF Kim $\sigma_N = 2[cm]$, (g) JBF Kim $\sigma_N = 5[cm]$, (h) JBF Kim $\sigma_N = 10[cm]$, (i) WJBF $\sigma_N = 0[cm]$, (j) WJBF $\sigma_N = 2[cm]$, (k) WJBF $\sigma_N = 5[cm]$, (l) WJBF $\sigma_N = 10[cm]$.

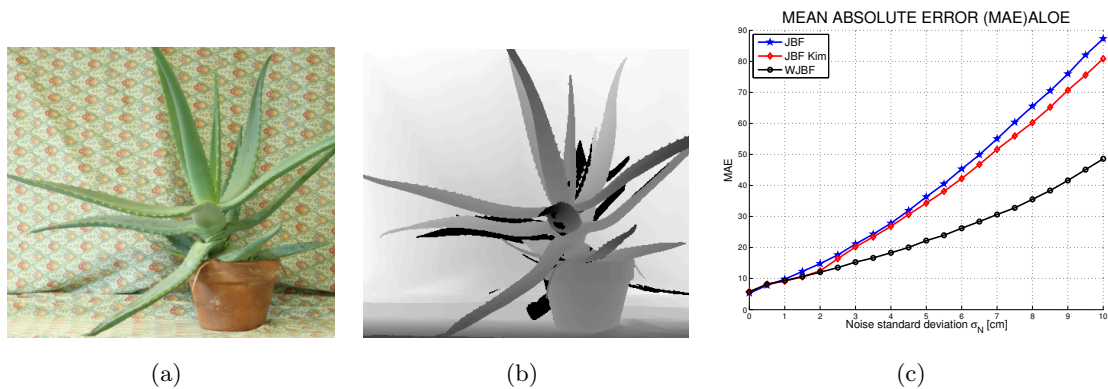


Figure 4.8: Objective comparison of the *aloe* scene super-resolution: (a) Image, (b) Depth ground truth, (c) MAE comparison.

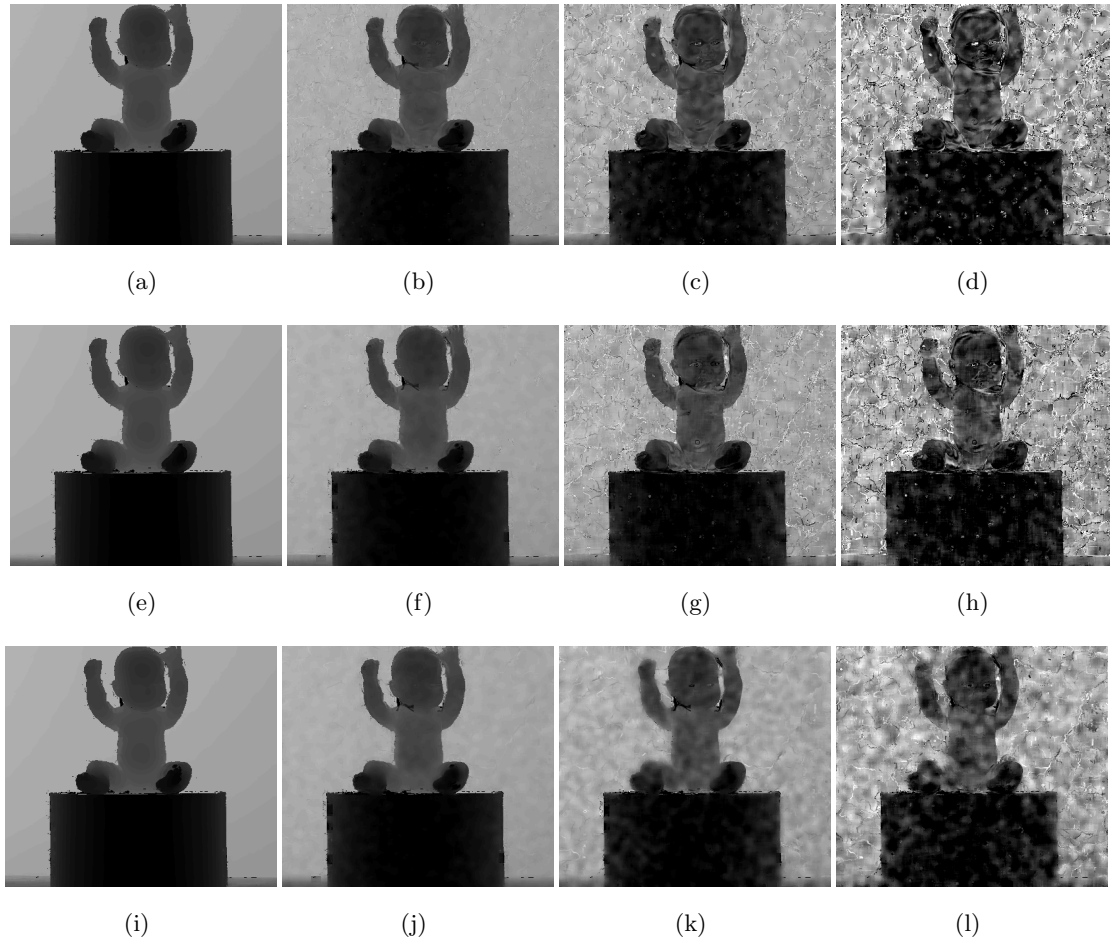


Figure 4.9: Visual comparison of the *baby1* scene super-resolution: (a) JBF $\sigma_N = 0[cm]$, (b) JBF $\sigma_N = 2[cm]$, (c) JBF $\sigma_N = 5[cm]$, (d) JBF $\sigma_N = 10[cm]$, (e) JBF Kim $\sigma_N = 0[cm]$, (f) JBF Kim $\sigma_N = 2[cm]$, (g) JBF Kim $\sigma_N = 5[cm]$, (h) JBF Kim $\sigma_N = 10[cm]$, (i) WJBF $\sigma_N = 0[cm]$, (j) WJBF $\sigma_N = 2[cm]$, (k) WJBF $\sigma_N = 5[cm]$, (l) WJBF $\sigma_N = 10[cm]$.

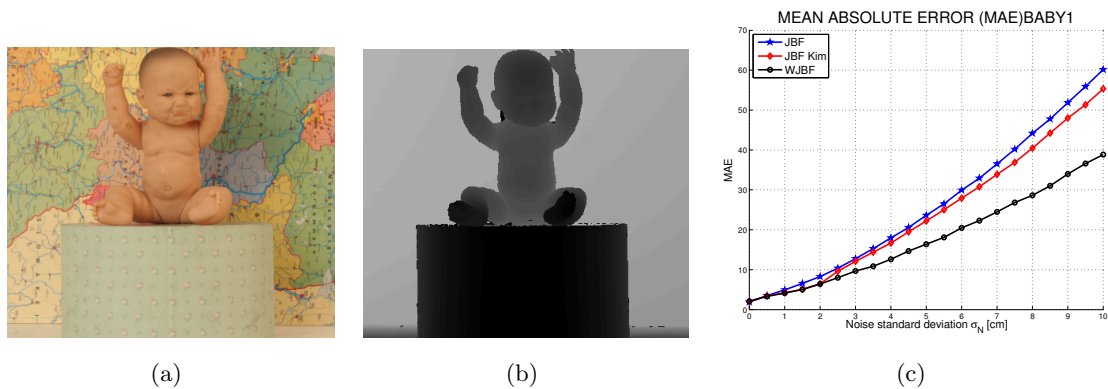


Figure 4.10: Objective comparison of the *baby1* scene super-resolution: (a) Image, (b) Depth ground truth, (c) MAE comparison.

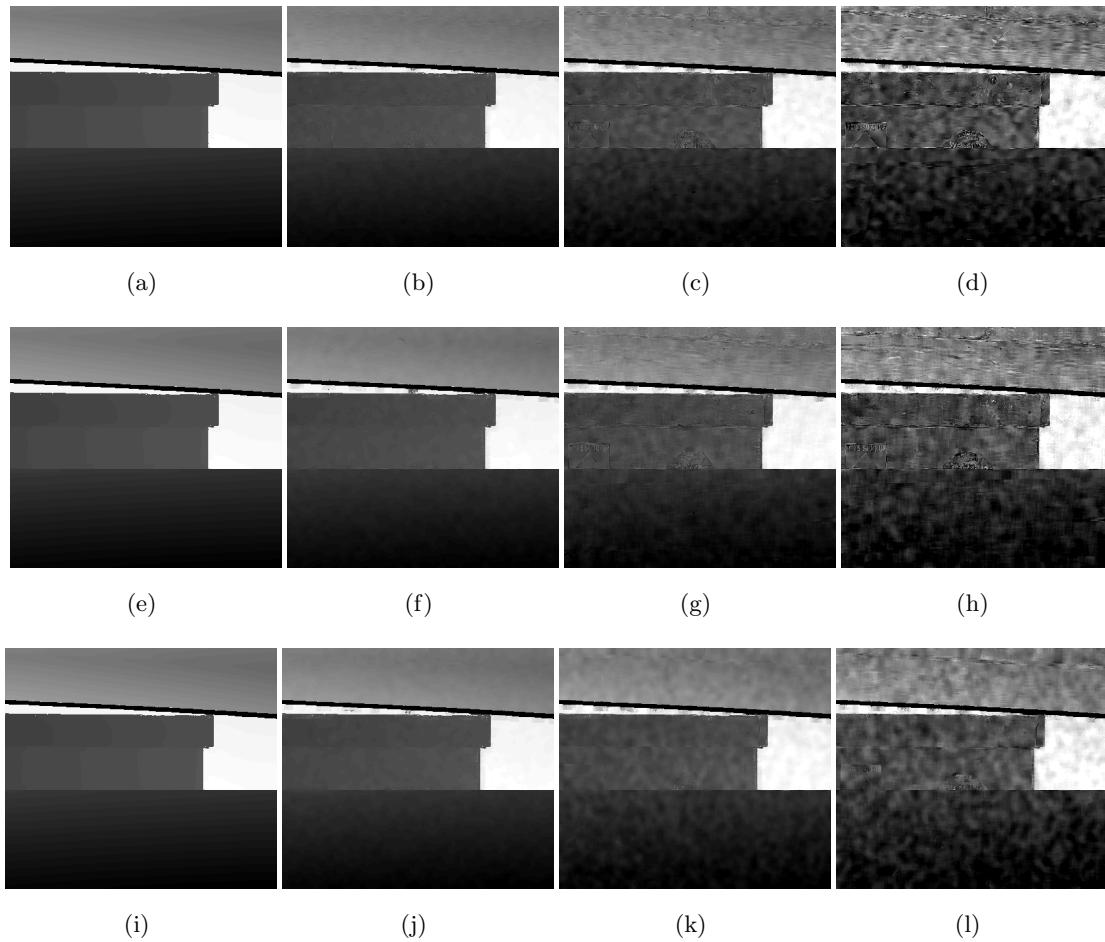


Figure 4.11: Visual comparison of the *wood2* scene super-resolution: (a) JBF $\sigma_N = 0[cm]$, (b) JBF $\sigma_N = 2[cm]$, (c) JBF $\sigma_N = 5[cm]$, (d) JBF $\sigma_N = 10[cm]$, (e) JBF Kim $\sigma_N = 0[cm]$, (f) JBF Kim $\sigma_N = 2[cm]$, (g) JBF Kim $\sigma_N = 5[cm]$, (h) JBF Kim $\sigma_N = 10[cm]$, (i) WJBF $\sigma_N = 0[cm]$, (j) WJBF $\sigma_N = 2[cm]$, (k) WJBF $\sigma_N = 5[cm]$, (l) WJBF $\sigma_N = 10[cm]$.

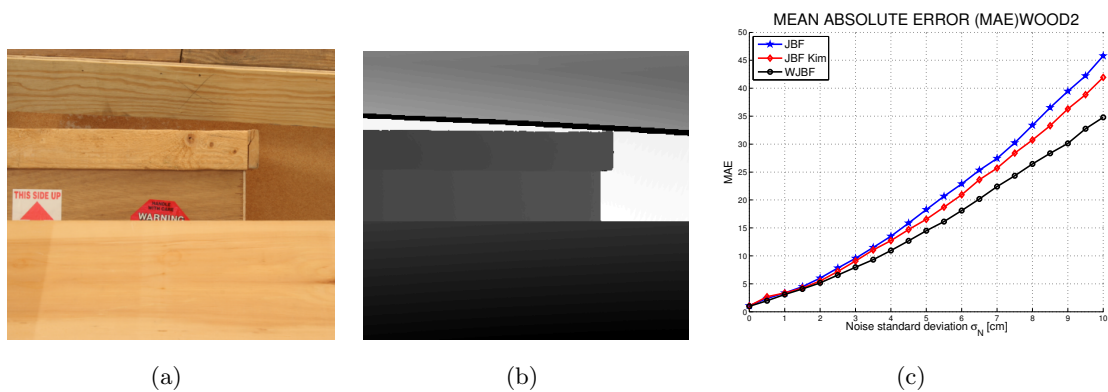


Figure 4.12: Objective comparison of the *wood2* scene super-resolution: (a) Image, (b) Depth ground truth, (c) MAE comparison.

For very high level of noise the results based on the WJBF method are affected by a slight blur along the objects border. This is a problem that could be solved with a tuning of the threshold parameters and with a more sophisticated selection of the weighting parameter α , as in [47], which presently has a linear behavior. Nevertheless this does not invalidate the objective evaluation based on the MAE (4.7) metrics as function of the noise standard deviation σ_N shown in Figures 4.8, 4.10, and 4.12. The curves behaviors show that the normal JBF always perform worse than the JBF_{KIM} and WJBF. Moreover, the latter two show almost the same results for the first levels of noise (i.e., until $\sigma_N = 2.5[cm]$). Then the WJBF outperforms the JBF_{KIM} thanks to its noise adaptivity and due to the fixed threshold τ in [47].

4.5 Real model for depth super-resolution

After the development and the evaluation of the super-resolution algorithms in the ideal model conditions, a *real model* has to be considered. For this purpose data from a ToF and video-cameras have to be analyzed. An overview of the framework that we adopted is shown in Figure 4.13. The first problem that has to be addressed is the difference between

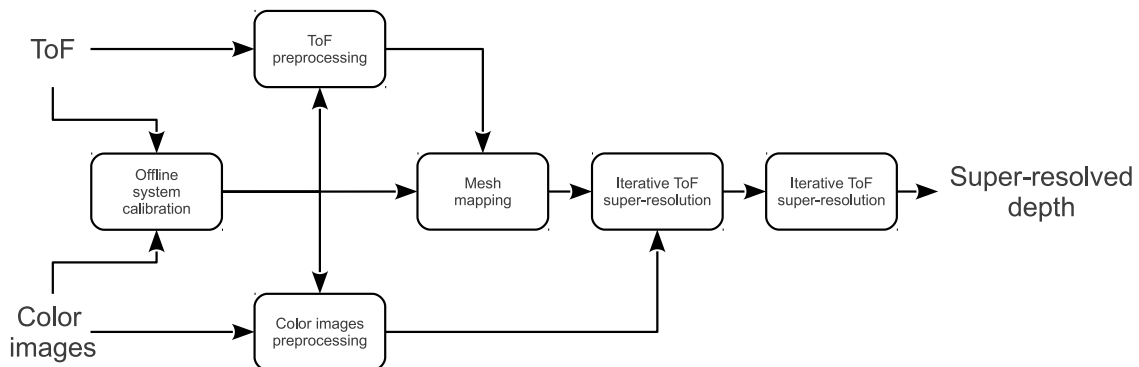


Figure 4.13: Depth super-resolution framework.

the image planes of the ToF and the color cameras and the lens distortion associated to each camera. Therefore a calibration procedure is needed. The intrinsic parameters are used to perform a ToF and color images preprocessing. Moreover in the ToF preprocessing the depth is filtered to reduce the ToF noise and also the flying pixels are detected. Then a mesh mapping to align the ToF based depth to the color image plane is applied using the extrinsic parameter and the ToF depth itself. Once the depth and the color image are superimposed the super-resolution can be performed. This is organized in an iterative fashion. The depth map is subsequently upsampled with the guidance of a subsampled version of the color image until the camera resolution is reached. Eventually, a post-processing stereo technique for a final super-resolved depth refinement is applied.

4.5.1 Camera rig

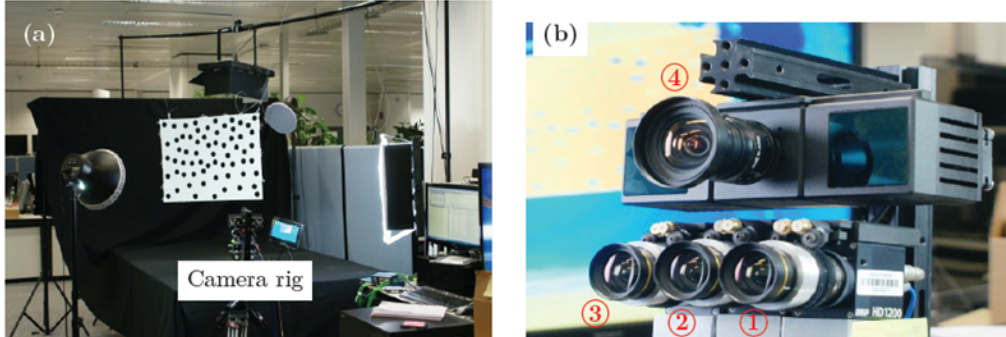


Figure 4.14: Camera rig: Scene in front of the camera rig (a), which is mounted on a tripod. The camera rig (b), composed by the ToF camera and the three standard cameras.

The available hardware for the depth super-resolution is shown in Figure 4.14(b). The rig is composed by the PMD CamCube 3.0 ToF camera indicated by ④, and three *Lux Media Plan (LMP) HD1200* [60] CMOS video-cameras indicated by ①, ②, ③. In particular the ToF camera is positioned on the top of the central standard camera ② and all the cameras are synchronized by a hardware trigger. The acquisition setup is shown in Figure 4.14 (a). The background and the desks are covered with black sheets and additional lights are used to illuminate the scene.

4.5.2 Calibration

Camera calibration refers to the problem of recovering the external and internal geometry of an optical acquisition device that are needed to describe the camera imaging process. The external geometry of a camera is defined by the rotation matrix \mathbf{R} and the translation vector \mathbf{t} , which relates the camera orientation and position to the world coordinate system. The internal geometry is described by the camera calibration matrix \mathbf{K} . This evaluates the geometric aberrations produced by the lens system of the camera. The method that was used for the calibration procedure is the SONY's toolbox from [21]. This procedure evolves the method proposed in [22] with the application of a 2.5D dot calibration pattern in substitution of the original one shown in Figure 4.15. This planar pattern, formed by 64 uniquely placed dots, is suited for the standard camera calibration but fails with low-resolution ToF camera amplitude images [21]. This is the reason for the application of a 2.5D pattern, where the dots are replaced with holes. The used pattern is visible in Figure 4.14 (a) and has a size of $800 \times 600[mm]$, whereas each hole has a diameter of $40[mm]$ to ensure that the IR-rays of the ToF camera illumination unit could pass through them. Thanks to this solution the calibration toolbox permit to calibrate a set of stereo cameras and a ToF camera simultaneously within subpixel and submillimeter accuracy. For more details the reader is referred to [21].

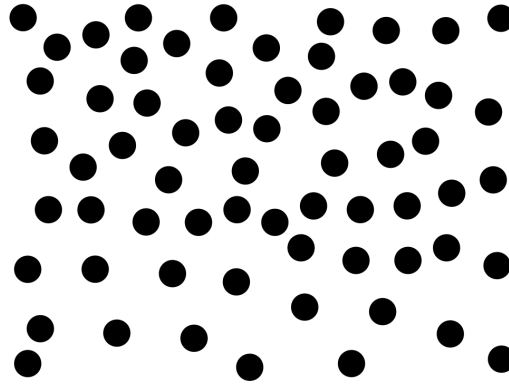


Figure 4.15: Calibration pattern: The planar dot pattern with 64 uniquely placed black dots [22].

4.5.3 Color images pre-processing

As already mentioned, the RGB images from the video-cameras need a distortion correction. Therefore, the MATLAB toolbox from Bouguet [23] is applied to each color camera with the corresponding radial distortion coefficients obtained from the camera calibration. Once that the undistortion is performed, the RGB images are down-sampled by a scale factor 8 (from 1920×1080 to 240×135) to have the same resolution of the depth after the mesh mapping.

4.5.4 ToF pre-processing

The ToF pre-processing is used to correct the raw depth map and it follows the framework shown in Figure 4.16. As for the RGB images, also the ToF depth needs an initial

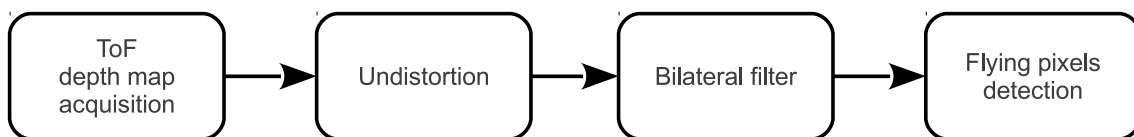


Figure 4.16: Overview of the ToF pre-processing.

undistortion correction. This is as well performed with the Bouguet [23] toolbox. Then a bilateral filter is applied. This allows to reduce the noise which may invalidate the subsequent mesh mapping. Now the factor Φ in Equation (4.12) really refers to all the possible noise sources and $\sigma_{ToF}(d)$ is the ToF noise standard deviation experimentally measured at distance d . Then a variable $\sigma_r = 3\sigma_{ToF}(d)$, is used in the filtering. In fact, as explained in Section 4.4.3, if the estimated depth standard deviation (4.13) is under $3\sigma_{ToF}$ it is not possible to differentiate between depth discontinuities or noisy flat areas.

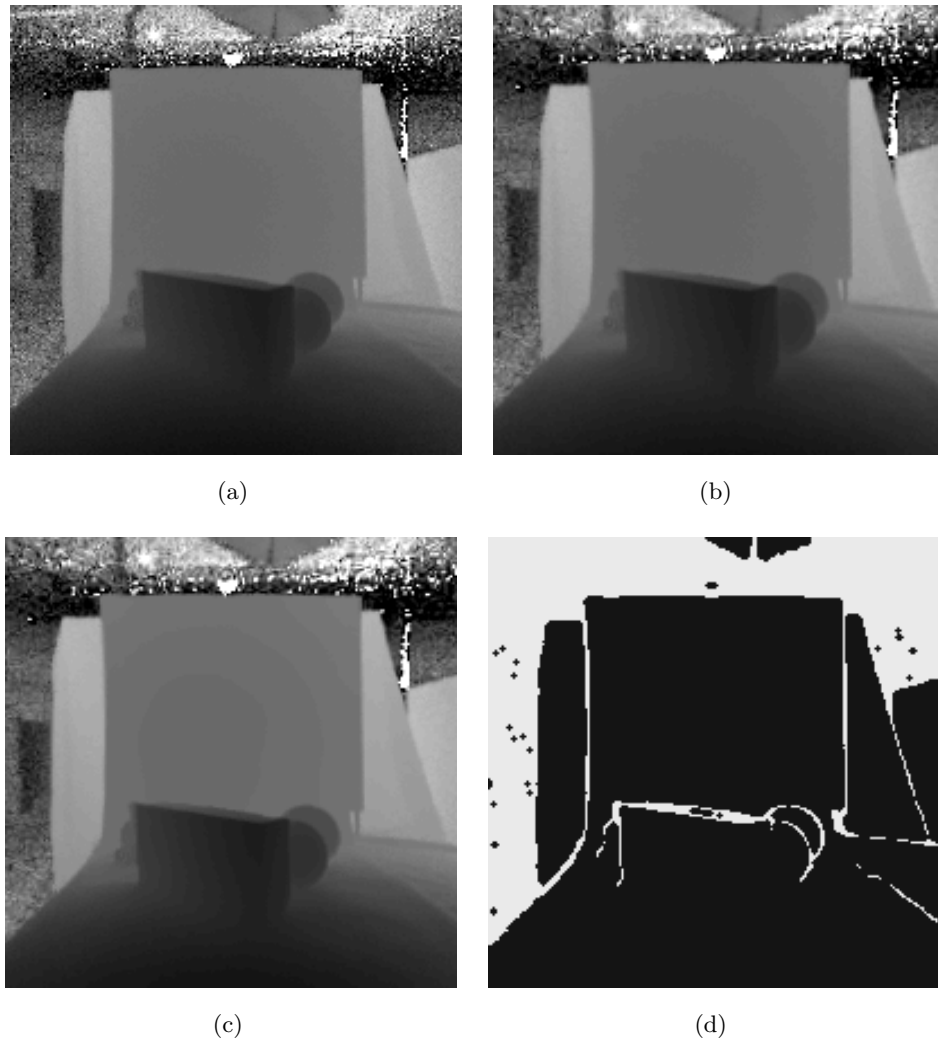


Figure 4.17: Results of the ToF pre-processing: (a) Raw ToF depth map, (b) Undistorted ToF depth map, (c) Depth map after the bilateral filter, (d) Flying pixels detection.

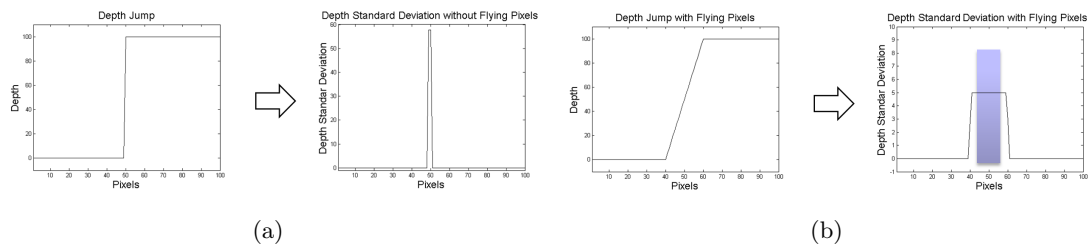


Figure 4.18: Flying pixel detection: (a) Ideal depth discontinuity and the associated standard deviation, (b) Real depth discontinuity and the associated standard deviation, the flying pixels area is also marked.

Figure 4.17 shows the effect of undistortion and bilateral filtering onto the raw ToF depth of the acquired scene, from now called *Pyramid*. Finally the flying pixels are detected

and marked in order to be removed from the subsequent mapping. An initial binary map which selects the pixels for which the estimated standard deviation is greater than $5\sigma_{ToF}(d)$ is obtained. This is then eroded of one pixel. In fact, as it is shown in Figure 4.18, an ideal depth discontinuity presents a very narrow standard deviation peak while a wider peak is obtained in presence of flying pixels. Then the difference between the two peaks width refers to the flying pixels region and this can be detected with the erosion process.

4.5.5 Mesh mapping

Thanks to the knowledge of the intrinsic and extrinsic parameters of each camera it is possible to project all the pixels from the pre-processed ToF depth onto each video-camera image plane. Let us denote with (u, v) the coordinates in the ToF image plane of the pixel \mathbf{p}_{ToF} and the correspondent depth value with z_{ToF} . This coordinates are firstly de-normalized (multiplication for z_{ToF}) and consequently multiplied with the inverse of the ToF camera calibration matrix \mathbf{K}_{ToF}^{-1} . Then, in order to project the points into the RGB camera image plane, the roto-translation matrix $[\mathbf{R}|\mathbf{t}]$ and the RGB camera calibration matrix \mathbf{K}_{RGB} are applied. Eventually the coordinates are normalized to obtain the final homogeneous coordinates (x_{RGB}, y_{RGB}) . The overall mapping process can be then summarized by the following Equation:

$$\begin{pmatrix} x_{RGB} \\ y_{RGB} \\ 1 \end{pmatrix} = \begin{pmatrix} x'_{RGB}/z_{RGB} \\ y'_{RGB}/z_{RGB} \\ 1 \end{pmatrix} = \mathbf{K}_{RGB} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{K}_{ToF}^{-1} \begin{pmatrix} u \cdot z_{ToF} \\ v \cdot z_{ToF} \\ z_{ToF} \end{pmatrix} \\ 1 \end{bmatrix}. \quad (4.14)$$

It is important to notice that the ToF camera ④ and the three available RGB cameras ①, ② and ③, see Figure 4.14, have different viewpoints. Hence the detection of the

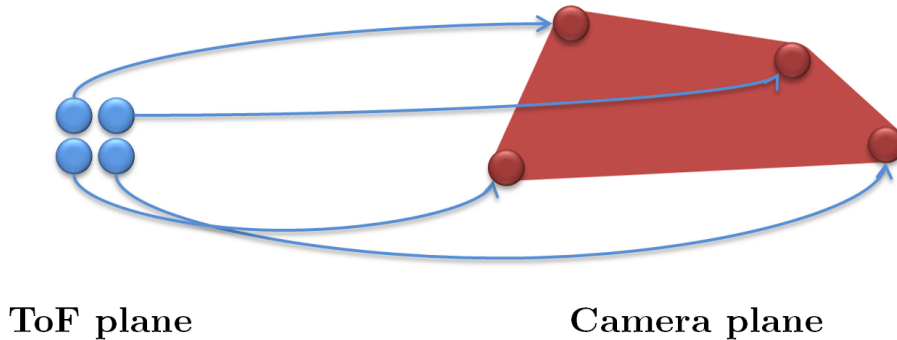


Figure 4.19: Graphical representation of the mesh mapping: four neighbors ToF pixels (blue pixels) are mapped onto the reference camera image plane as a mesh (red mesh).

areas which are occluded from another camera's view may fail due to the sparsity of a simple mapping. In order to solve this issue we introduced a *mesh mapping* procedure (see

Figure 4.19). For every pixel in the ToF image plane the three neighbors are selected (blue pixels). These are mapped into the RGB camera image plane forming a mesh where the central area is filled with the mean value of the four red vertices. The average value is used instead of a more sophisticated interpolation since we want to reduce the computational cost and the meshing process is only used to detect occlusion regions. We decided to use the average value in the quadratic section is used to When a closer mesh partially or completely overlaps a farther mesh only the foreground mesh is maintained. Moreover also the meshes which are composed by at least two flying pixels are discarded. The result is then shown in Figure 4.20(a).

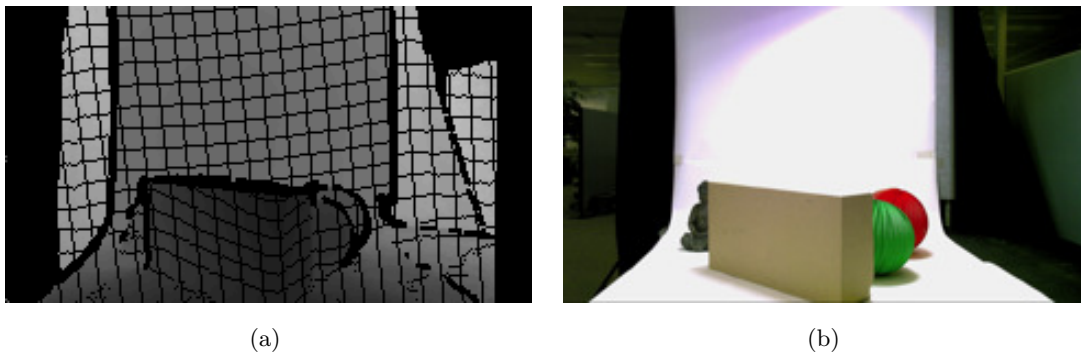


Figure 4.20: Mesh mapping results: (a) Mesh mapped, 240×135 , ToF depth map on the RGB image plane, the black areas are removed mesh due to occlusions or flying pixels detection, (b) scaled RGB image, 240×135 .

It is important to notice that not only the occlusion and flying pixels areas are removed (in black) but also the areas at the left and at the right borders of the image. In these regions the distance to the ToF camera is greater than the ToF range and are affected by the phase wrapping problem. Then, following the model in Equation (4.12), they have $\Phi > 0$ and they are consequently detected and removed.

4.5.6 Iterative depth super-resolution

The iterative super-resolution is the core of the proposed algorithm, Figure 4.21 shows a block diagram of the procedure. An iterative approach has been used to reduce the filter aperture in the bilateral super-resolution approach and also to decrease the sparsity in the mapping process with the related occlusions issues. The inputs for this process are the sparse 240×135 mapped depth map, see Figure 4.20(a), and its corresponding 240×135 color image, see Figure 4.20(b). From these images a refined *dense depth map* can be obtained using the WJBF approach presented in Section 4.4.3. Then an *up-sampling* with a simple nearest-neighbor interpolation of the dense depth map is performed. Hence a new *sparse depth map* with a spatial resolution of 480×270 pixel is obtained. After that a *depth jump detection* is applied by using the same method for the detection of the flying

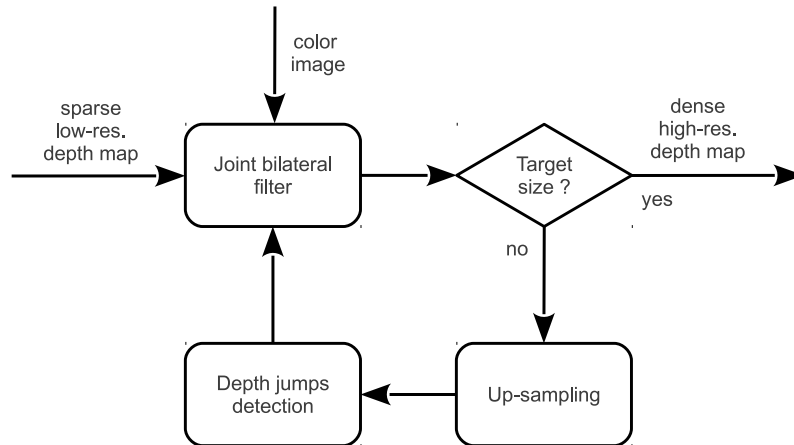


Figure 4.21: Overview of the iterative super-resolution process.

pixels during the ToF pre-processing. Then the edge pixels are removed and the sparse depth is filtered using again the WJBF with a new 480×270 guidance image, obtained from the down-sampling of the color image. This procedure is repeated until the target resolution is achieved, with a maximum spatial resolution of 1920×1080 pixel. Figure 4.22(a) shows the resulting super-resolved depth image obtained from the low-resolution ToF depth of Figure 4.20(a).

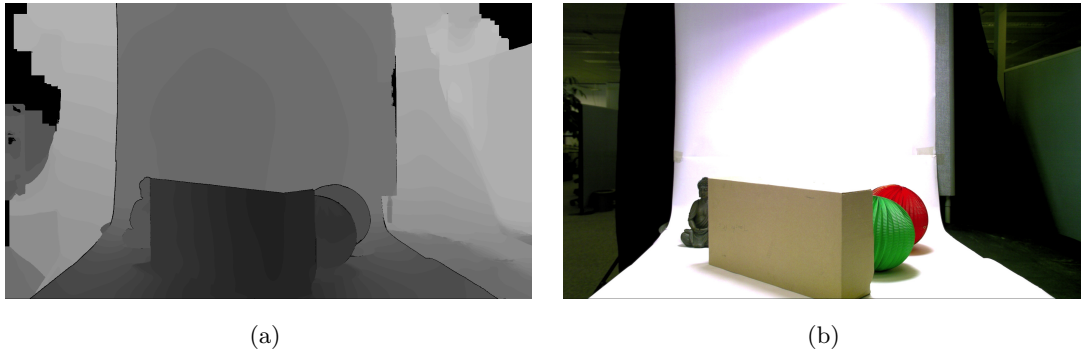


Figure 4.22: Iterative super-resolution results: (a) Iteratively up-sampled ToF depth map, 1920×1080 , (b) full resolution RGB image, 1920×1080 .

In particular is possible to notice the presence of black areas along the object borders. These pixels are still undetermined after the super-resolution. In fact, a problem may arise when all the pixels within the filter aperture which belongs to the same object do not have a valid depth value (either for the depth map sparsity or for the removal of the edge border). Then during the interpolation a threshold is applied to avoid that only depth values from another object will be used.

4.5.7 Stereo refinement

The last step in the super-resolution framework is the *stereo refinement*. This optional refinement is used to exploit the information from the multiple cameras setup and in particular to fill the pixels that are still undetermined after the iterative super-resolution process. A graphical explanation of this post-processing procedure is provided in Figure 4.23. For each undetermined pixel \mathbf{p}_R with coordinates (x_R, y_R) in the 1920×1080 refer-

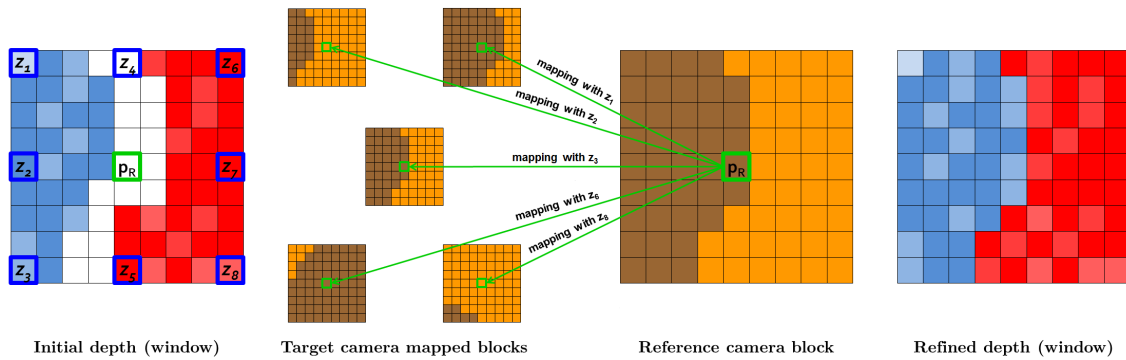


Figure 4.23: Overview of the stereo refinement process: (Left) The depth map window in an edge area with the undetermined pixels in white, (Center) the mapping of the reference camera block with the unique and available depth values, (Right) the refined depth map window.

ence color image is searched the conjugate point \mathbf{p}_T in the target color image. This search is carried out by assigning to \mathbf{p}_R all the possible depths z_1, z_2, \dots, z_8 from the neighbors pixels in a window centered in \mathbf{p}_R . The depth is used to map the correspondent reference color image window to the target color image by means of Equation (4.14). Then for every possible target color image window a similarity value, based on SSD, with the reference window is calculated. Eventually the depth which provide the lowest SSD value is selected as final depth for the undetermined pixel \mathbf{p}_R .



Figure 4.24: Stereo refinement results: (a) Stereo refined ToF depth map, 1920×1080 , (b) JBF refined ToF depth map, 1920×1080 .

Obviously this final post-processing refinement is used only in case that a second video-camera is available. An alternative refinement based on a simple JBF (4.8) is also possible. The difference between the two approaches is shown in Figure 4.24.

4.5.8 Experimental results

Different scenes have been acquired with the camera rig of Figure 4.14. The super-resolution procedure explained in the previous Sections was applied to them. For each scene are provided:

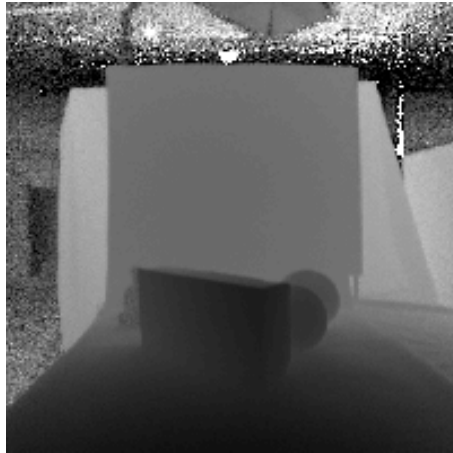
- The 200×200 raw ToF depth image.
- The 1920×1080 super-resolved depth image.
- The 1920×1080 reference video-camera color image.

Moreover the following parameters were used:

- **ToF pre-processing:** The chosen kernel size was composed by 15×15 pixels, while the range σ_r and spatial σ_s parameters were set to $3\sigma_{ToF}(d)$ and 15. For the flying pixels detection the standard deviation of the 200×200 depth image was calculated over a 3×3 mask.
- **Iterative depth super-resolution:** For the WJBF a filter size of 31×31 and 15×15 pixels were used respectively for the first and the subsequent iterations. In fact in the first iteration a larger filter size is needed to fill all the undetermined areas due to occlusions and flying pixels removal. In the following iterations the depth has to be only refined, hence a smaller filter is more suitable. The spatial σ_s and the two range parameters $\sigma_{r,FLAT}$ and $\sigma_{r,EDGE}$ were set to 15, 0.1 and 0.03.
- **Refinement:** For the *stereo refinement* the size of the window centered around the undetermined pixel is 9×9 pixel, while for the *joint bilateral refinement* a JBF with a kernel size of 31×31 pixels was applied only to the undefined pixels.

Pyramid

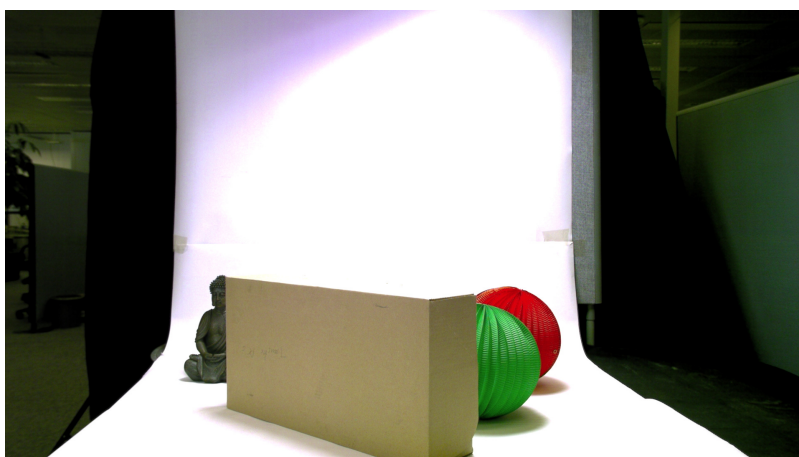
This scene presents a collection of objects, a box, two balls and a Buddha statue, arranged in a pyramidal fashion. The super-resolved depth shown in Figure 4.25(b) presents smooth regions inside the objects and sharp depth discontinuities but one issue is evident at the top of the brown box. In fact, due to the additional illumination, the corresponding color saturate to white. Then it is not anymore possible to distinguish the top of the box from the background and the filter assigns the background depth values to this area. Nevertheless, looking at the ToF raw data (Figure 4.25(a)) it is possible to notice that the top part of the box has a width of only 3 pixels, which are all flying pixels.



(a)

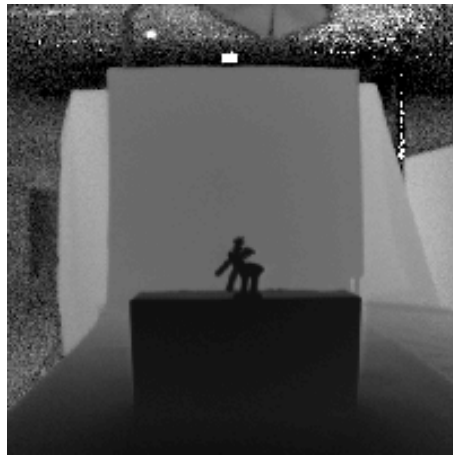


(b)

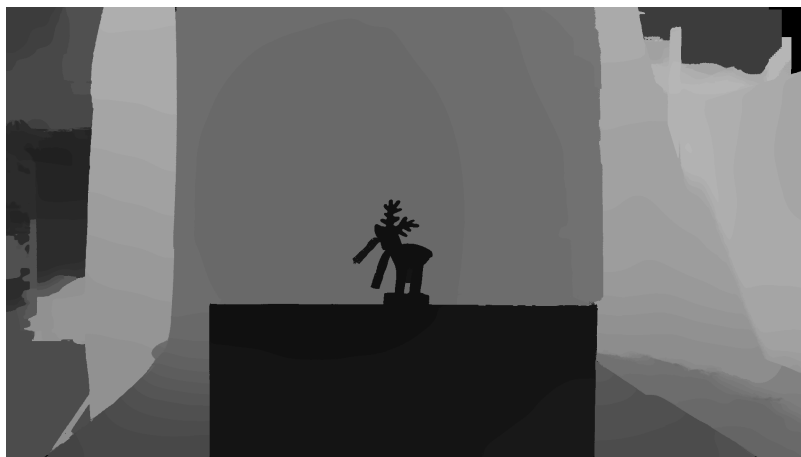


(c)

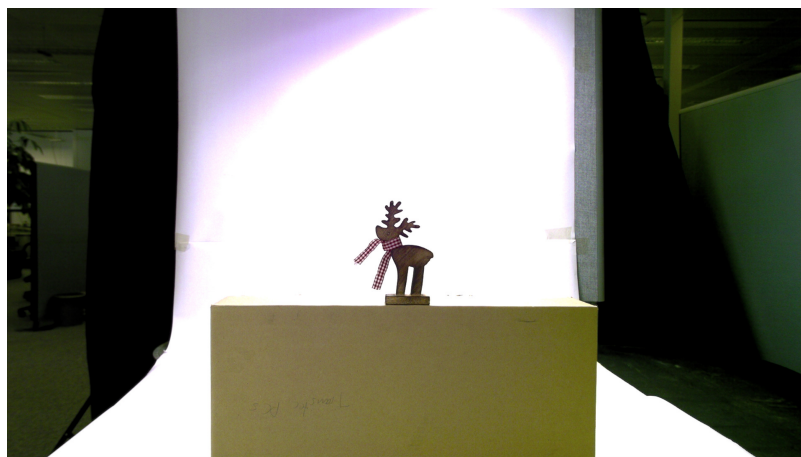
Figure 4.25: Results of super-resolution on *Pyramid*: (a) Raw ToF depth map, 200×200 , (b) super-resolved ToF depth map, 1920×1080 , (c) color camera image, 1920×1080 .



(a)

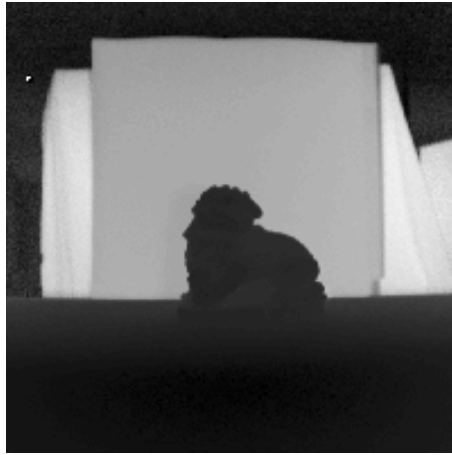


(b)



(c)

Figure 4.26: Results of super-resolution on *Elk*: (a) Raw ToF depth map, 200×200 , (b) super-resolved ToF depth map, 1920×1080 , (c) color camera image, 1920×1080 .



(a)



(b)



(c)

Figure 4.27: Results of super-resolution on *Lion*: (a) Raw ToF depth map, 200×200 , (b) super-resolved ToF depth map, 1920×1080 , (c) color camera image, 1920×1080 .

In fact these values do not correspond to the correct depth but to a linear combination between the background and the front of the box. Therefore a reconstruction in this area is not possible.

Elk

This scene presents a wooden elk located at the top of a brown box. Even in this case the top of the box can not be reconstructed due to the color saturation, see Figure 4.26(b). On the other side is possible to appreciate, especially on the elk's horns and the scarf, the accuracy of the super-resolution. Moreover a small error is visible between the elk's legs which is due to the stereo refinement. After the iterative super-resolution procedure the background area between the two legs is marked as undetermined. Then this is processed with the stereo refinement. In particular in the upper region only disparity values which belong to the elk are selected within the filter aperture and therefore the area is filled with these values.

Lion

This scene presents a Chinese lion statue placed on a white stand. The related super-resolution, shown in Figure 4.27(b), exhibits a quite precise reconstruction, in particular along the object border. As it will be more visible in Section 6.4 the reconstruction precision has an accuracy of about $3 - 4$ [cm]. Indeed the ToF noise standard deviation at the considered distance is around 1 [cm] [24] and the accuracy after the filtering is related to the *three sigma rule* described in Section 4.4.3.

4.6 Conclusions

The aim of this chapter was to combine the information from a ToF camera with a stereo vision system in order to obtain high-resolution depth images. A super-resolution approach based on joint bilateral filtering was developed. This was compared against other state of the art methods in an ideal scenario, where only an additive Gaussian noise was considered. Then this method was also tested in a real scenario. For this purpose a complete super-resolution framework was presented. An offline camera rig calibration was initially used to obtain the intrinsic and extrinsic parameters of the cameras. Then a ToF depth image pre-processing procedure based on experimental camera noise measurements was performed. This was suitable to reduce the noise of the acquired depth map and to detect the flying pixels. Eventually, an iterative approach based on the bilateral filtering developed in the ideal scenario was used to increase the ToF depth resolution up to the color camera resolution. The final results show a high-resolution depth image, with a strong reduction of the noise thanks to the filtering procedure, and with sharp edges due to the additional information coming from the color image.

Chapter 5

Combined motion and disparity estimation

5.1 Introduction

In the recent years there has been considerable interest in recovering 3D motion in stereo image sequences. In [63, 64] Min et al. estimate the dense disparity associated to the following time instance by using the constraint between motion and disparity in stereo image sequence. Indeed thanks to the vector projection they are able to obtain an initial disparity that can be consequently refined. In particular, they consider a sequence of rectified images and for both the motion and disparity estimation they solve the Euler-Lagrange equation. In [65, 66] the authors reverse the problem. They assume the actual disparity as known and they jointly estimate the motion and the disparity related to the next frame. In [67] Valgaert et al. extend the problem to an uncalibrated stereo pair. They assume that the stereo rig does not change over time and consequently there exist a constant fundamental matrix which describes the epipolar geometry. Eventually they process the four estimations at once with a joint energy functional. In this Chapter we propose a method to estimate both the disparities and the MVFs. Moreover we do not follow the method described in [67] due to its computational complexity. Specifically four hierarchical estimations are processed in which at every estimation level a consistency between the estimated vectors is calculated. This information is then used in the subsequent iterations to obtain a final result that is robust and precise at the same time. The single estimation system based on a hierarchical evolution of the method shown in Chapter 2, which tries to address the aperture problem, is described in Section 5.2. Then the concept of round trip check is explained in Section 5.3. This is then used as quality evaluation of the estimation and permits to perform a more selective decision within the iterative process. In addition the results are evaluated in two different datasets, the Middlebury for optical flow [18] and the KITTI Vision Benchmark Suite [68]. Eventually we draw some conclusions in Section 5.5.

5.2 Independent estimations

In contrast to the methods presented in [63–66] we do not suppose a rectified stereo sequence. Moreover we do not even assume a constant fundamental matrix over time as in [67]. Then, as already mentioned in Chapter 3, with these constraints the estimation of the disparity is similar to the motion estimation. Indeed when the two images are not rectified a two dimensional displacement field has to be found. In total, we consider four types of two dimensional correspondences estimations: two motion estimations between consecutive frames of the same camera and two bi-dimensional disparity estimations between the left and right cameras at time t and $t + 1$. For this purpose an evolution of the method proposed in Chapter 2 can be used. The block based RS estimation provides a flexible and low computational complexity estimation. It should also be noticed that this can be included into the semi-global algorithms. In fact the estimation is performed minimizing the combination of a similarity function with an implicit smoothness constraint. This is integrated in the algorithm by means of the predictors choice, the updates range and the associated penalties. Eventually, thanks to the recursion, the minimization is semi-globally processed. Unfortunately a simple block based RS estimation presents two main drawbacks. The first is related to the convergence of the algorithm. Indeed there is no a priori information on the disparity or motion vectors range. This problem was not noticeable in Chapter 3 since the vectors in the Middlebury database for optical flow [18] present a low magnitude. Vice versa when we consider a high resolution sequence the vectors range can be sufficiently large to do not permit the convergence with only the intra frame recursion. This also justifies the temporal recursion inside the estimation. The second problem is related to the well known aperture problem and is explained in Table 5.1. The size of the block is related to the algorithm accuracy and reliability. A

Table 5.1: Aperture problem: advantages and disadvantages in the block size selection.

Larger block size	Smaller block size
Stable estimation ✓	Precise assignment ✓
Object borders problem ✗	Wrong minima problem ✗

larger block size provides a more stable estimation since it has more chances to include intensity differences but it may not follow correctly the object borders. With a smaller block a more precise assignment is possible but a wrong minimum is more easily found. Both problems can be attenuated by means of a hierarchical approach [69]. In fact the multiple estimation increases the possibility to locate large displacement and different block sizes can be used. At the coarse level large blocks provide a rough but reliable estimation which can be consequently refined with smaller blocks. In particular, as it is shown in Figure 5.1, the search range should also be reduced along the iterations to avoid wrong minima. Of course the hierarchical estimation has the disadvantage of an increased

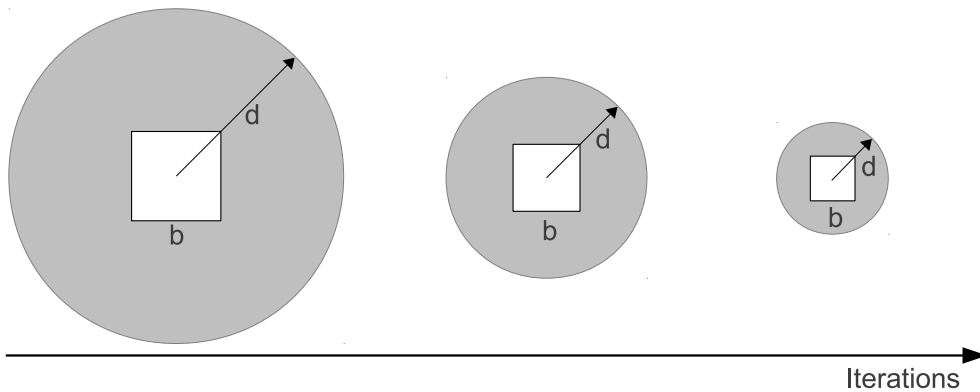


Figure 5.1: Block size in hierarchical estimation, from coarse to fine: the block size b decrease at every iteration as well as the search range d .

computational cost. Considering only independent estimations the hierarchical framework for the estimation of the 3D motion is depicted in Figure 5.2. We refer to it as combined motion and disparity estimation (CoMEDE). To improve the quality of the estimations it is important that the calculated motions and the disparities are consistent. This can be achieved by exploiting the spatial and temporal dependencies in the image sequence throughout the estimation process. In particular a method based on a local consistency between the estimations is described in the following Section.

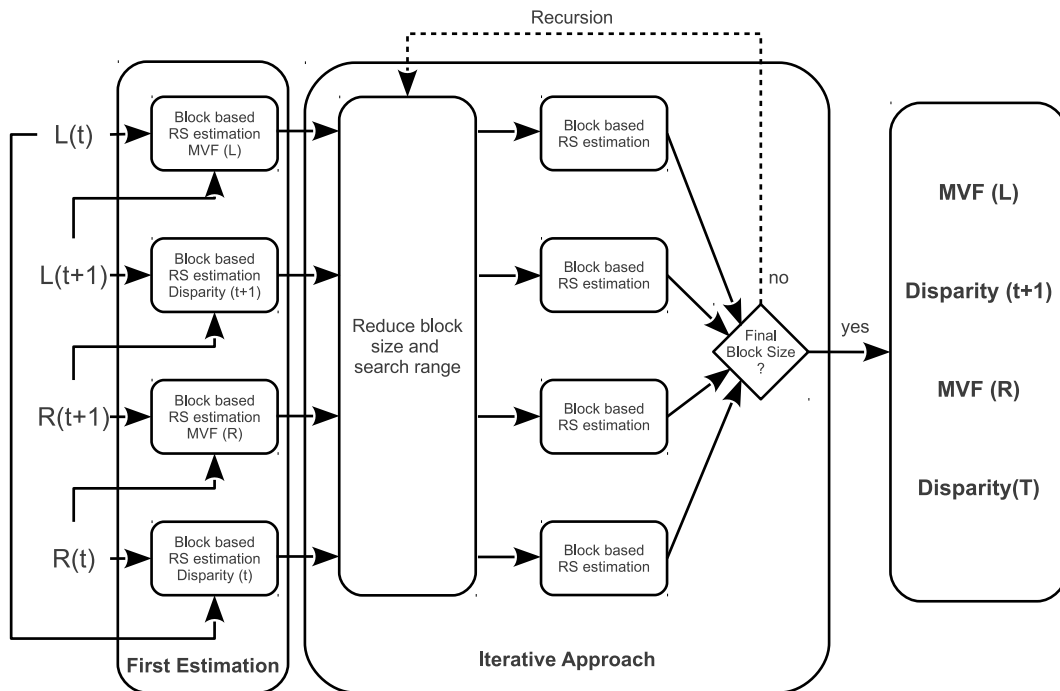


Figure 5.2: CoMEDE framework: Hierarchical estimation.

5.3 Dependent estimations: round trip check

In case of multiple estimations the spatial and temporal dependencies can be used to increase the quality of the results. In [67] Valgaert et al. closely couple all the estimations by means of a joint energy functional. This lead to the drawback of a high computational cost. In case of a simple block based RS the complexity of the problem increases exponentially. We can describe a single estimation as a tree where the root is the block for which the vector has to be estimated. From the root a first level of branches correspond to the different predictors, while a second level can be seen as the vector updates. Eventually the final candidates, that have to be compared with the root, coincide with the tree leaves. When another estimation is concatenated with the first a new tree is consequently associated to every leaf position. In case of a 3D motion in stereo image sequences the final number of leaves becomes a power of a factor four. This produces an undesirable growth of the computational cost that we want to avoid. Then a looser consistency between the estimations has to be considered. For this purpose we propose the framework shown in Figure 5.3. This extends the one of Figure 5.2 by taking advantage of the hierarchical process in

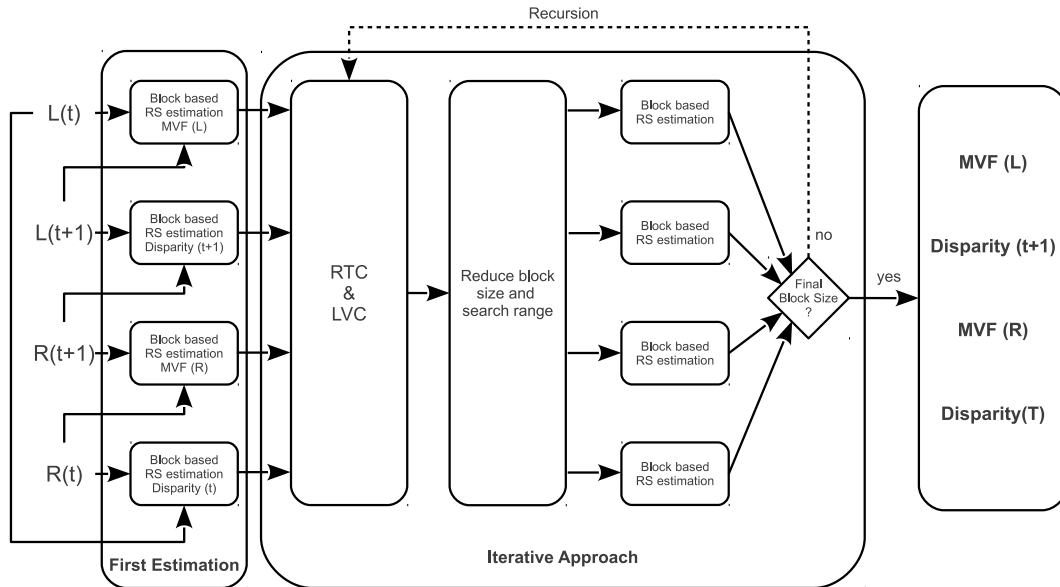


Figure 5.3: CoMEDE with RTC framework: Hierarchical estimation.

the single estimations. Specifically after every iteration a local vector consistency (LVC) inside all the displacement fields and a round trip check (RTC) between all the estimations are calculated. Then we will refer to the framework as CoMEDE RTC. The LVC is used to locate the object borders and consists in a bi-dimensional first derivative filtering of the estimated vector fields. The RTC correspond to a sequence of vector projections along a Hamiltonian path as the one shown in Figure 5.4. Representing as $(d_x(\cdot), d_y(\cdot))$ and $(v_x(\cdot), v_y(\cdot))$ the two dimensional displacements of a disparity and a motion estimation,

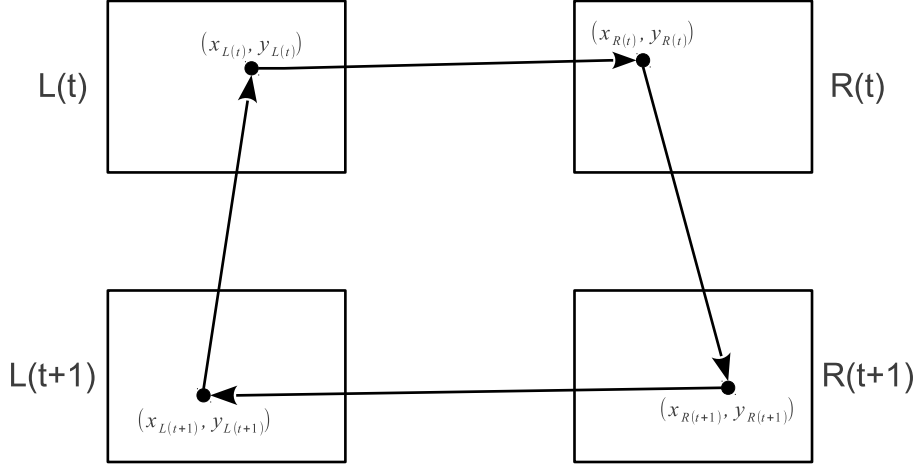


Figure 5.4: Round trip check: When the vector are correctly estimated the sum of all the vector projections should arrive to the original position.

the points $(x_{R(t)}, y_{R(t)})$, $(x_{R(t+1)}, y_{R(t+1)})$, $(x_{L(t+1)}, y_{L(t+1)})$ and $(x_{L(t)}, y_{L(t)})$ respect the following relations:

$$\left\{ \begin{array}{l} (x_{R(t)}, y_{R(t)}) = (x_{L(t)} + d_x(t), y_{L(t)} + d_y(t)) \\ (x_{R(t+1)}, y_{R(t+1)}) = (x_{R(t)} + v_x(R), y_{R(t)} + v_y(R)) \\ (x_{L(t+1)}, y_{L(t+1)}) = (x_{R(t+1)} + d_x(t+1), y_{R(t+1)} + d_y(t+1)) \\ (x_{L(t)}, y_{L(t)}) = (x_{L(t+1)} + v_x(L), y_{L(t+1)} + v_y(L)) . \end{array} \right. \quad (5.1)$$

Then the estimated displacements are likely correct when the position after a round trip is close to the starting point:

$$\left\{ \begin{array}{l} d_x(t) + v_x(R) + d_x(t+1) + v_x(L) = RTC_x < \epsilon \\ d_y(t) + v_y(R) + d_y(t+1) + v_y(L) = RTC_y < \epsilon . \end{array} \right. \quad (5.2)$$

Many circumstances can invalidate a similarity evaluation (e.g., different light conditions, smooth regions, non-Lambertian surfaces). Indeed due to the image differences it is not always true that the minimization of a matching criterion corresponds to a correct estimation. A comparison between the RTC error and the SAD error is shown in Figure 5.5. It is possible to see how the SAD error map has a high value in the textured areas of the leaves or in the rocks, while the error is not noticeable in case of smooth regions along vector changes. The RTC error map provides a low value in the textured areas, since in this regions the vector can be uniquely determined, and a high error in the occluded areas around the object borders. Of course the RTC can be directly minimized by using a close coupling constraint as in [67] or indirectly minimized by a RTC based selection of the predictors in the subsequent block based RS estimations. In particular we propose to combine the RTC and the LVC informations and to select the predictors of the subsequent estimation as it is shown in Figure 5.6. Since the RTC calculates a global consistency be-

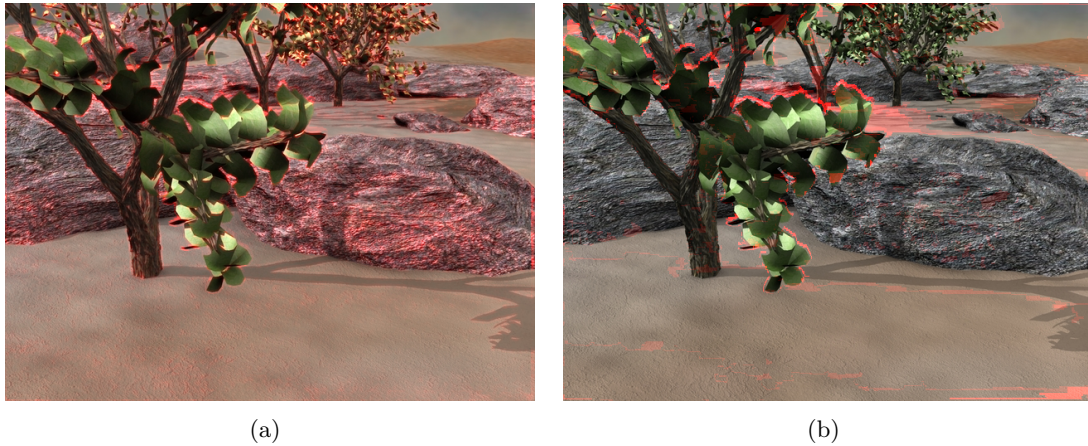


Figure 5.5: Comparison of SAD and RTC: (a) SAD error map superimposed to the image, (b) RTC error map superimposed to the image.

tween all the displacement fields, in case of a high RTC it is not possible to derive which estimation produced the error. Then we decided to follow a conservative approach and to apply the usual RS scheme (Figure 5.6(a)) in presence of a high RTC. In case of a low RTC the LVC information is used to detect the object borders. When the LVC is low we can assume that we are inside an object and only predictors from the previous estimation are used (Figure 5.6(b)). In particular also the search range is reduced since the previous iteration's vectors should provide a more reliable estimation value due to the bigger block size. This is not true in case of a high LVC. Indeed estimation along the objects borders has to be refined and this is carried out with the predictor scheme shown in Figure 5.6(c).

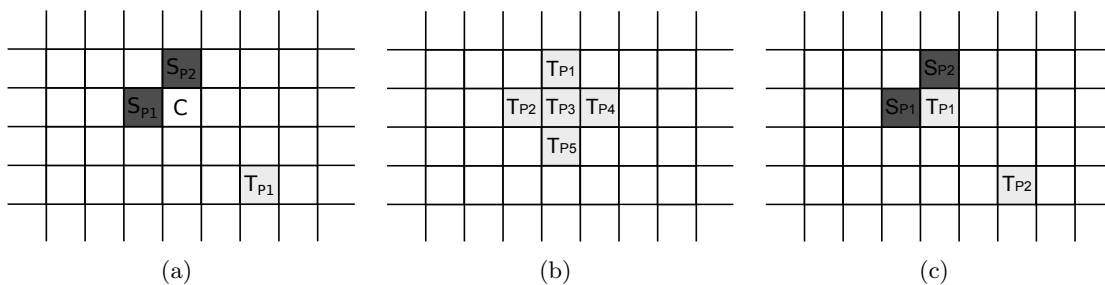


Figure 5.6: Selection of the predictors based on the RTC and LVC: based on a conservative approach we decided to use (a) the usual RS scheme in case of a high RTC, (b) a set of predictors from the previous iteration in case of a low RTC and a high LVC, and when both the RTC and the LVC are low (c) the usual RS scheme with in addition the temporal predictor at the current block position due to its related low RTC.

5.4 Experimental results

In order to prove the effectiveness of the RTC we tested the hierarchical estimation with and without this constraint. Moreover two matching criteria were also considered, the SAD and the BCC. The test was made on the Middlebury database [18] and in the KITTI Vision Benchmark Suite [68]. It should also be noticed that, for this evaluation, no temporal consistency, as in Section 2.6, was used.

5.4.1 Middlebury database for Optical Flow

As already mentioned in Section 2.6, the Middlebury database for optical flow [18] provides a set of different sequences with an available ground truth MVF related to the fourth frame. In particular, since the sequences are related to only one view, the scheme shown in Figure 5.4 is reduced to a two frames estimation. Then the only available Hamiltonian path is related to the trivial forward backward consistency check. Moreover for the BCC a search range of 7×3 pixels for the first iteration was considered. The results of the objective evaluation are shown in Figure 5.7 for the AE and the EE respectively. As in Chapter 2 both the metrics present a similar trend. The results based on the BCC

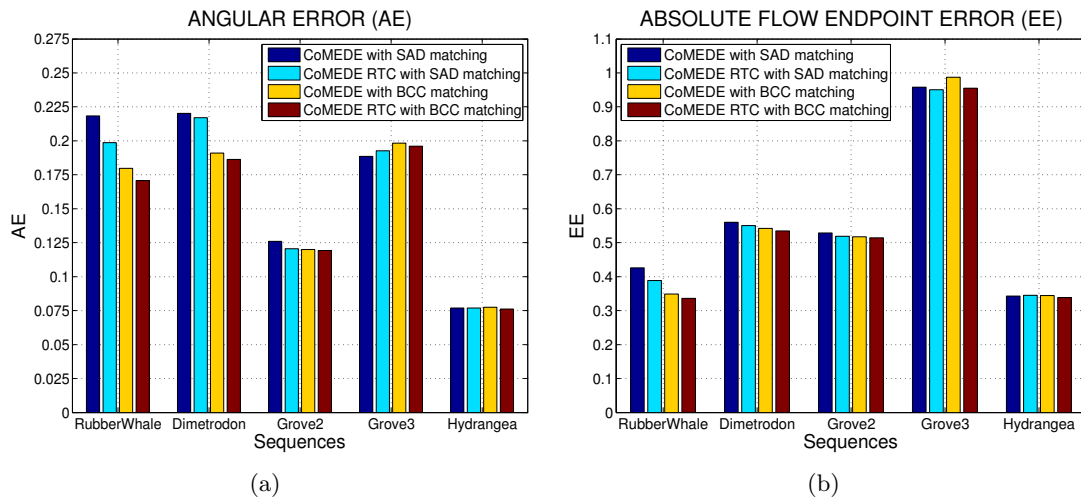


Figure 5.7: Results comparison CoMEDE system: (a) Angular error results, (b) Absolute flow endpoint error results.

provide, in general, a more precise estimation and the accuracy is increased with the RTC constraint. Moreover thanks to this constraint the BCC drawback of a slightly more noisy estimation in the homogeneous regions is reduced. This is particularly evident in the *Rubber Whale* sequence shown in Figures 5.8(e) and 5.8(f). Also for the SAD based estimation the RTC causes noticeable benefits in the *Rubber Whale* sequence. Figure 5.8(c) and 5.8(d) show how the RTC constraint avoids a wrong minimum in the upper smooth area of the rotated D .

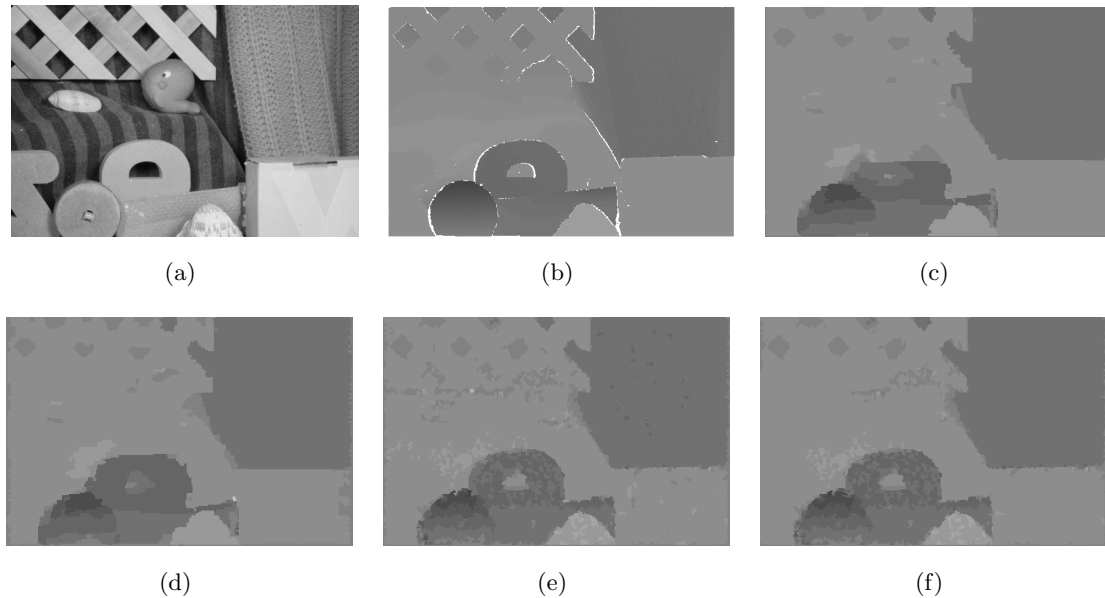


Figure 5.8: Results comparison in RubberWhale sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for CoMEDE with SAD matching, (d) MVF for CoMEDE with RTC constraint and SAD matching, (e) MVF for CoMEDE with BCC matching and (f) MVF for CoMEDE with RTC constraint and BCC matching.

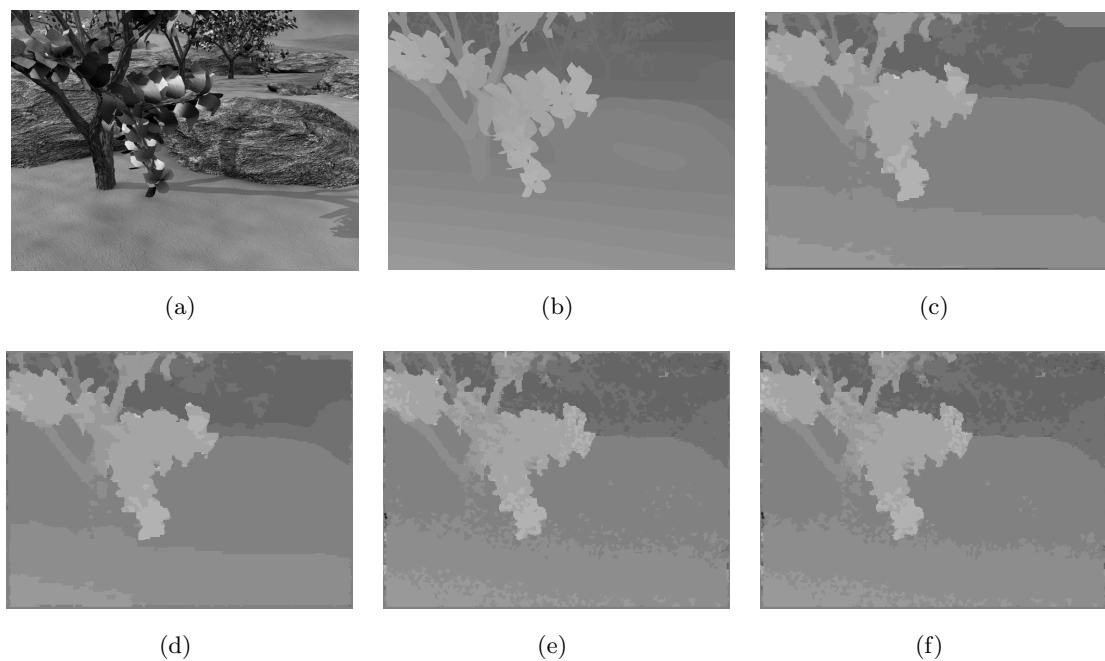


Figure 5.9: Results comparison in Grove2 sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for CoMEDE with SAD matching, (d) MVF for CoMEDE with RTC constraint and SAD matching, (e) MVF for CoMEDE with BCC matching and (f) MVF for CoMEDE with RTC constraint and BCC matching.

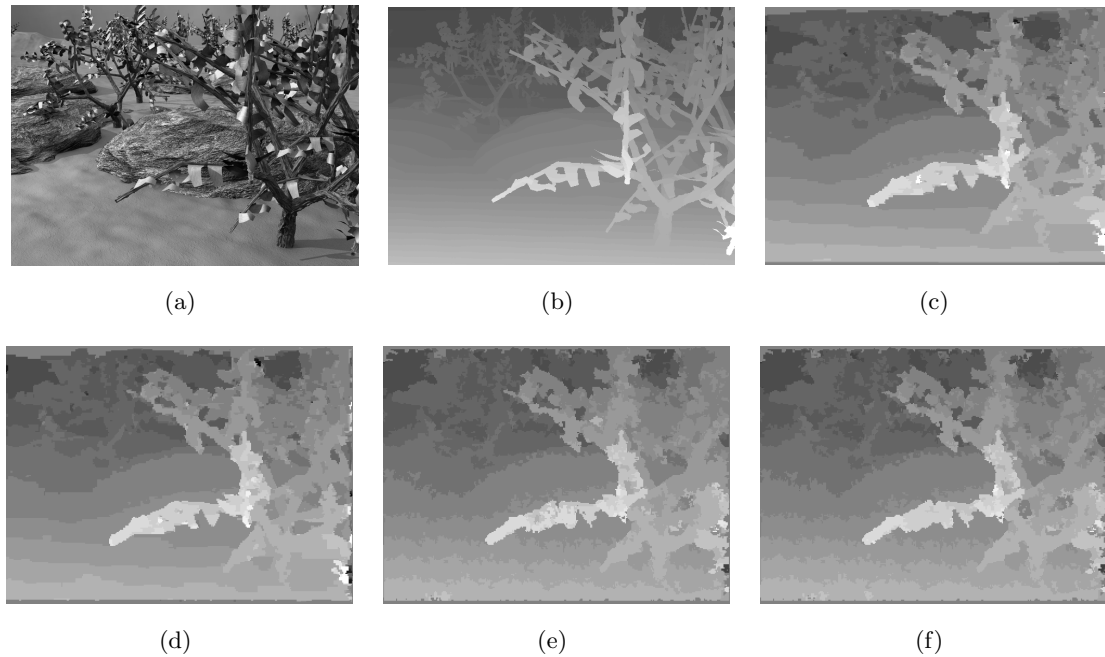


Figure 5.10: Results comparison in Grove3 sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for CoMEDE with SAD matching, (d) MVF for CoMEDE with RTC constraint and SAD matching, (e) MVF for CoMEDE with BCC matching and (f) MVF for CoMEDE with RTC constraint and BCC matching.

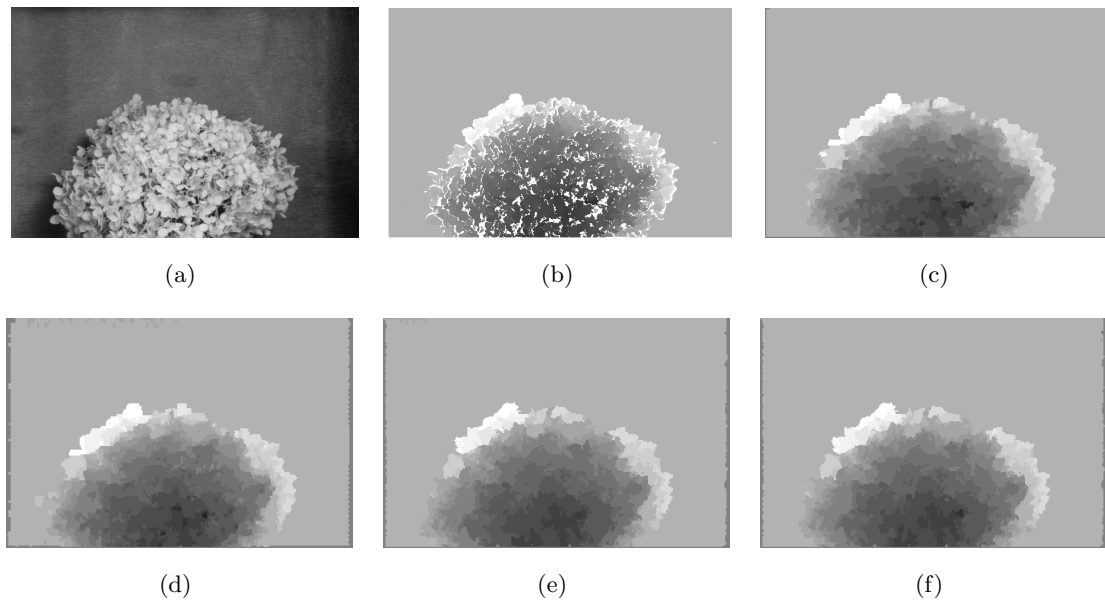
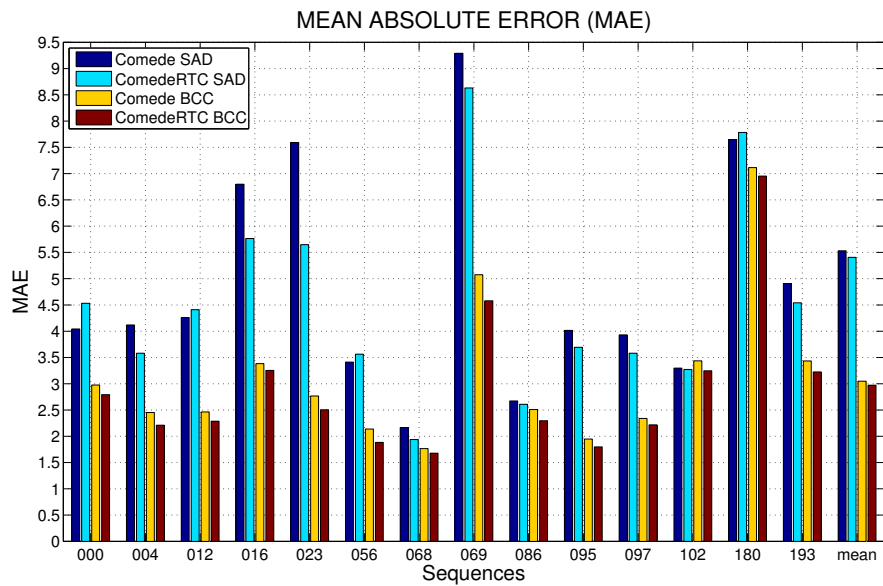


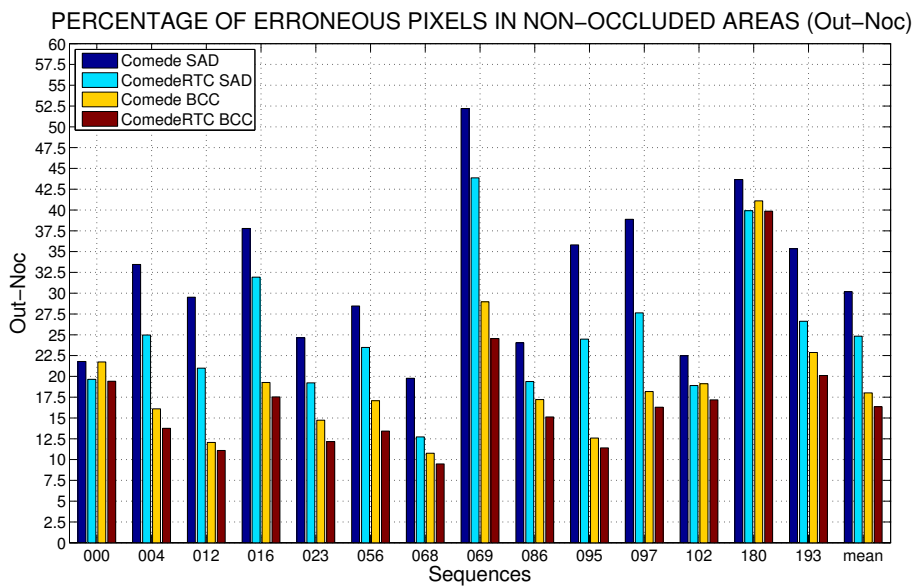
Figure 5.11: Results comparison in Hydrangea sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for CoMEDE with SAD matching, (d) MVF for CoMEDE with RTC constraint and SAD matching, (e) MVF for CoMEDE with BCC matching and (f) MVF for CoMEDE with RTC constraint and BCC matching.

5.4.2 KITTI Vision Benchmark Suite

The KITTI Vision Benchmark Suite [68] provides a set of different stereo sequences. These sequences were formed by two frames of rectified images and a sparse ground truth for both the disparities and MVFs is also available. Then the RTC follows the Hamiltonian path shown in Figure 5.4. The results are shown in Figure 5.12(a) and 5.12(b) for the



(a)



(b)

Figure 5.12: Results comparison CoMEDE system: (a) Mean absolute error results, (b) Percentage of correctly estimated pixels, with threshold $th = 3$.

MAE and Out-Noc (with a threshold of three pixels) metrics respectively. It is possible to see that again the RTC improves the resulting estimation, in general for the MAE and for all the sequences considering the Out-Noc metrics. In particular the application of the RTC constraint is quite evident in the BCC matching since also all the MAE results improve with the RTC constraint. Indeed with the 1 bit estimation and the reduction of the block size in the hierarchical approach, a wrong minimum is more easily found than with the SAD based matching criteria. It should also be noticed that the gain related to the RTC application is more significative than in the Middlebury database. In fact, in the KITTI database the range of the displacements is considerably higher and the estimations are more prone to errors. Moreover, as it is shown in Figures 5.13 and 5.15, with the BCC it is even possible to match the fine structures of the tarmac. It should also be noticed that no assumption were done on the disparity range or in the disparity direction. Moreover the estimated disparity was not restricted to a mono-dimensional case. This, of course, are limitations which can definitely be used to improve the final results.

5.5 Conclusions and future work

In this Chapter an algorithm for the joint motion and disparity estimation was described. Every single estimation was performed by using the present state of the art methods. The different iterations are carried out in coarse to fine fashion, in which the block size as well as the search range decrease at every step. Then the concept of RTC and LVC were introduced. In order to maintain an acceptable computational cost the RTC was minimized modifying the implicit smoothness constraint of a RS algorithm provided by the predictors choice. The resulting joint estimation was tested against the independent estimations to prove its validity. It should also be noticed that the proposed implementation follows a conservative approach. Indeed the RTC information is used only to prevent possible errors in the subsequent iterations. Further investigations should be carried out to verify if the combination of the informations provided by the LVC and RTC can be used to identify which estimation produced the error. Another possible improvement is related to a possible estimation of the fundamental matrix for the disparity estimation. After the first estimation level in the hierarchical process, the RTC can be used to provide to every displacement a reliability value. Then, from the most reliable values the fundamental matrix can be inferred and used in the subsequent estimations to increase the algorithm robustness.

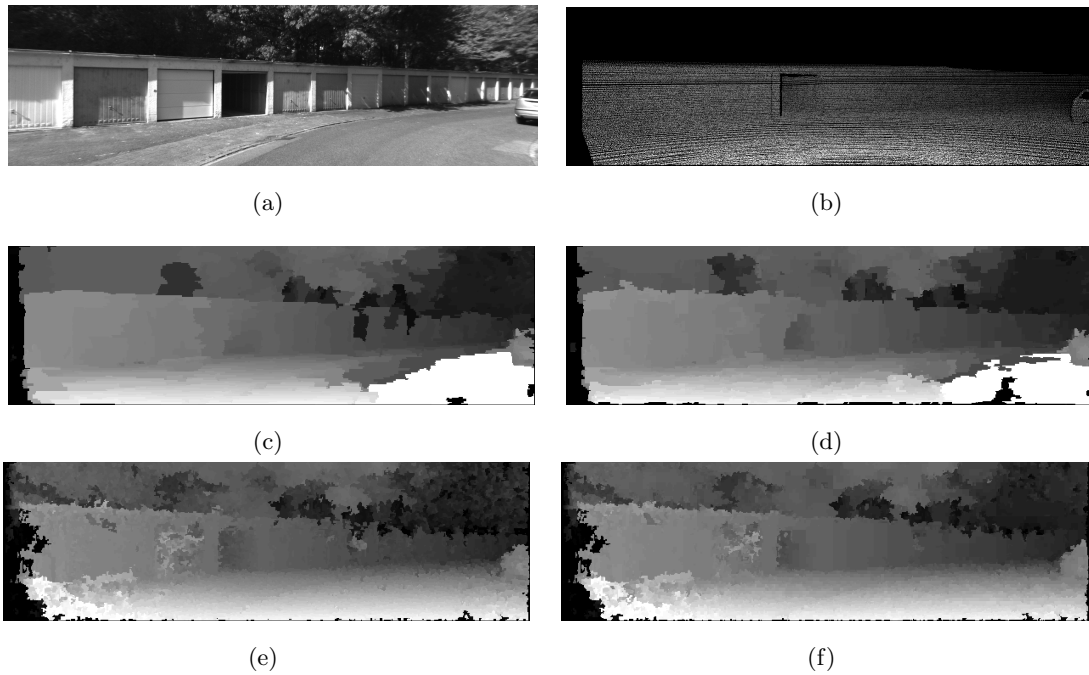


Figure 5.13: Results comparison in 023 sequence, (a) Original image, (b) Ground truth disparity, (c) Disparity for CoMEDE with SAD matching, (d) Disparity for CoMEDE with RTC constraint and SAD matching, (e) Disparity for CoMEDE with BCC matching and (f) Disparity for CoMEDE with RTC constraint and BCC matching.

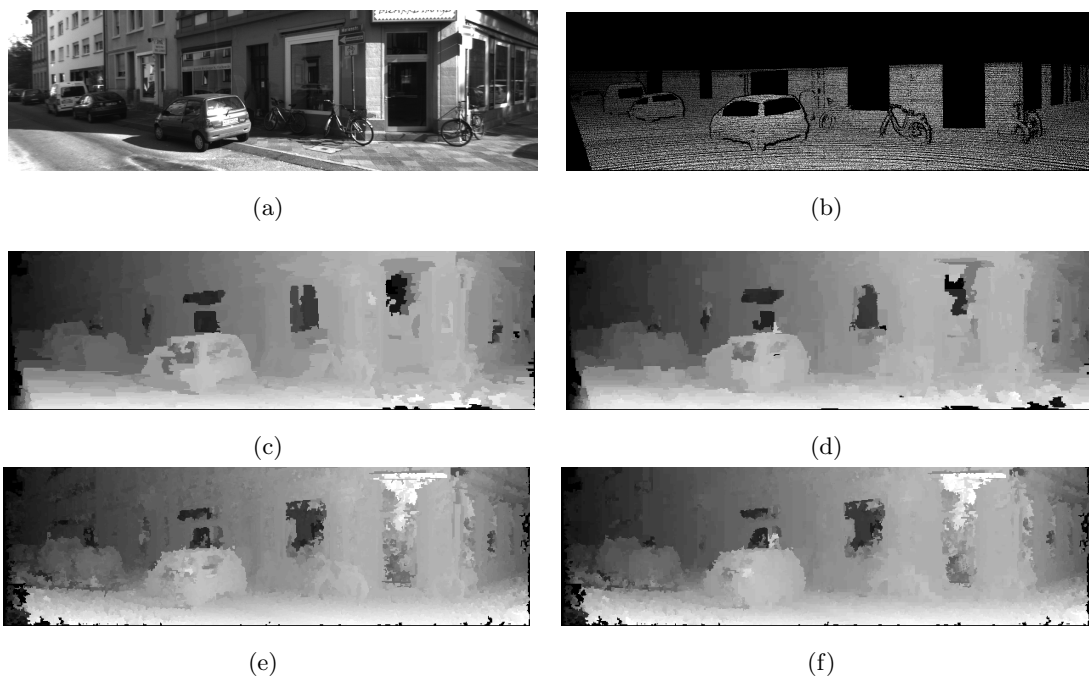


Figure 5.14: Results comparison in 068 sequence, (a) Original image, (b) Ground truth disparity, (c) Disparity for CoMEDE with SAD matching, (d) Disparity for CoMEDE with RTC constraint and SAD matching, (e) Disparity for CoMEDE with BCC matching and (f) Disparity for CoMEDE with RTC constraint and BCC matching.

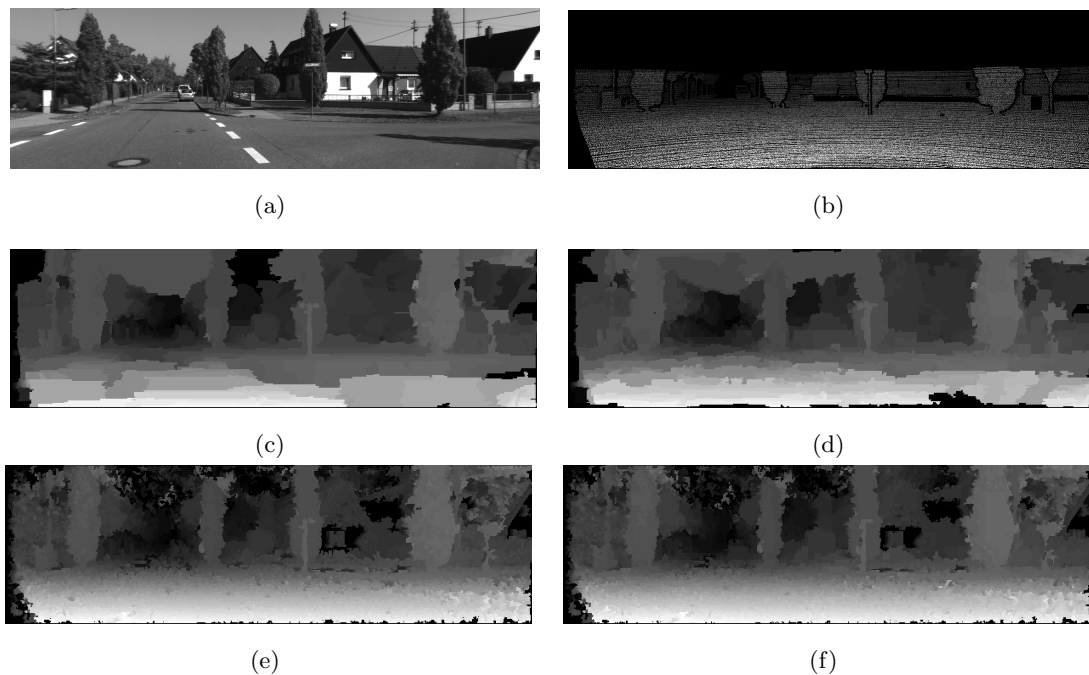


Figure 5.15: Results comparison in 095 sequence, (a) Original image, (b) Ground truth disparity, (c) Disparity for CoMEDE with SAD matching, (d) Disparity for CoMEDE with RTC constraint and SAD matching, (e) Disparity for CoMEDE with BCC matching and (f) Disparity for CoMEDE with RTC constraint and BCC matching.

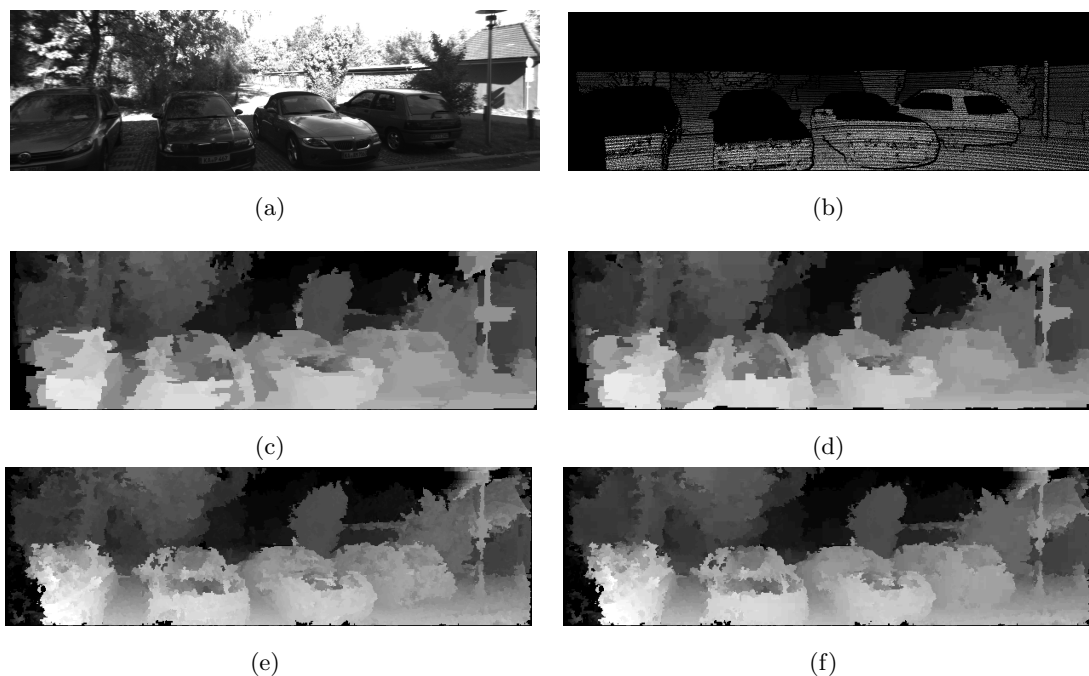


Figure 5.16: Results comparison in 193 sequence, (a) Original image, (b) Ground truth disparity, (c) Disparity for CoMEDE with SAD matching, (d) Disparity for CoMEDE with RTC constraint and SAD matching, (e) Disparity for CoMEDE with BCC matching and (f) Disparity for CoMEDE with RTC constraint and BCC matching.

Chapter 6

Application to 3D reconstruction

6.1 Introduction

The 3D reconstruction of a scene or object is one of the most important and challenging topic in computer vision. In this field many different methods have been proposed. These can be divided in active [52, 70, 71], passive [72–76] and hybrid procedures [77, 78]. Active methods refer to a large number of hardware devices based on laser scanners [70], structured light [71] or time-of-flight systems [52]. Since these systems provide already a geometry estimation, a relatively high effort was spent for the fusion of the different views into a single 3D geometry [79]. While some automatic registration schemes have been proposed [80] this is still an issue far from being solved and generally it requires a considerable human interaction. Passive approaches permit the 3D reconstruction of a scene or object geometry using only pictures. One of the first methods was based on shape-from-silhouette [72] but relies on the segmentation and camera calibration accuracy. Recently many new procedures have been developed. In *Bundler* [73] the authors use a modified version of the *Sparse Bundle Adjustment* package of Lourakis and Argyros [81] to estimate the camera positions and to obtain a sparse scene geometry. A similar approach is used in *Microsoft Photosynth* [74] which permits a 3D reconstruction based on a large collection of pictures available on the web. Other available methods are the *Autocad 123D* [75] and the *3DF Zephyr PRO* [76]. In particular with [75] is possible to upload pictures of an object in a cloud system that consequently calculates the associated geometry. Hybrid methods try to combine the informations given by an active and passive systems. In [77] the authors combines the color image and the depth data provided by a *Microsoft Kinect* [55]. In particular they use the color information to improve the different views registration. Another example is the kickstarter project *Lynx* [78]. This is a tablet shape stand-alone system which combine a 3D depth sensor with a standard color camera to obtain a 3D mesh model of the recorded space.

In this Chapter we provide an overview on how a dense estimation can support a 3D reconstruction method. Furthermore also the ToF super-resolution method presented in Chapter 4 are considered. In particular the work that will be described is the outcome of a collaboration with the Technical University of Dortmund (TUDo) and with the German Research Center for Artificial Intelligence (DFKI) of Kaiserslauten. In Section 6.2 an overview of the proposed framework for the 3D reconstruction is presented. Specifically, the system is firstly analyzed in a reduced environment where only two views are considered. This is used to evaluate the results of the dense estimation based on the combination of the CoMEDE system of Chapter 5 and the OF method of TUDo. Then, in Section 6.3 the system is extended to a multiview scenario to achieve the reconstruction of complete objects. A possible support of a ToF camera, specifically as depth initialization and segmentation, is analyzed in Section 6.4. Eventually, in Section 6.5 some conclusions and a possible future work are outlined.

The methods presented in this chapter were applied for two patents in [82, 83]. Furthermore, parts of the work described here has already been published by O. Wasenmueller in his MSc thesis [84].

6.2 System explanation

The overview of the 3D reconstruction framework is shown in Figure 6.1. The system permits the reconstruction of a 3D pointcloud from two not calibrated cameras. Firstly the

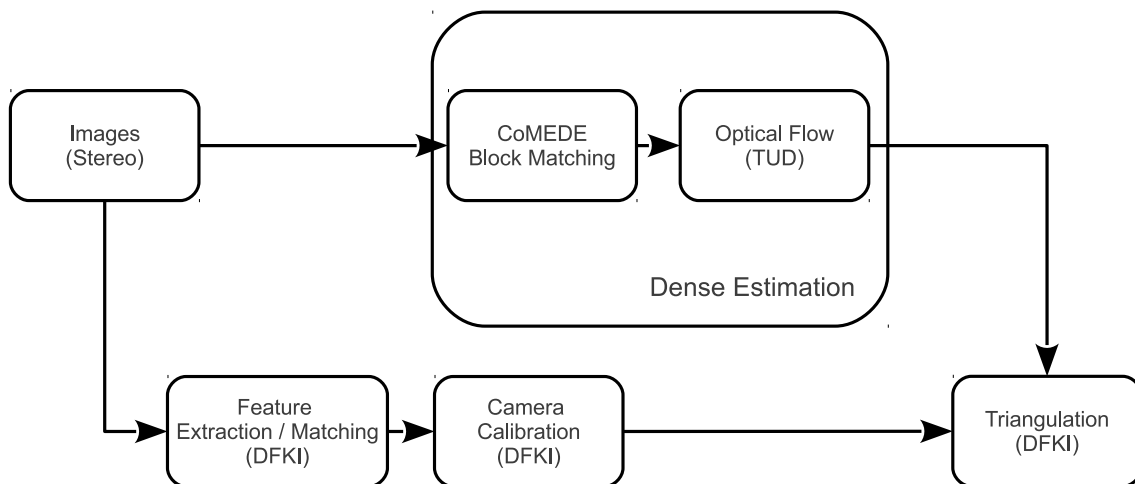


Figure 6.1: Framework for the 3D reconstruction with two cameras.

salient points of the images are extracted and matched to perform a subsequent camera calibration. Then the two images are passed to the dense estimator. This produces a similarity map in the form of disparity. It is important to notice that in the dense estimation the images are not rectified and consequently the disparity is provided both

for horizontal and vertical directions. Eventually the triangulation process is performed. This is an adaptation of the procedure presented in [85] and takes into account the camera lens distortion.

6.2.1 Camera calibration

The camera calibration is divided in two different steps due to the different way in which the intrinsic and extrinsic parameters are estimated.

Intrinsic parameter estimation

The target scenario of the application is the reconstruction of 3D object from a set of different images captured with the same camera. Then the intrinsic parameters can be estimated offline to obtain a more robust calculation. For this purpose the method proposed in [22] was considered. This uses the planar dot pattern with 64 uniquely placed dots that was already shown in Figure 4.15.

Extrinsic parameter estimation

Due to the applicative scenario the extrinsic parameters can not be estimated in advance. Then for every image a set of salient points are found. Specifically the SIFT [86] or the SURF [87] descriptors are considered. These are consequently processed with the Robust-Matcher algorithm described in [88] to find the corresponding feature points between the two images. This is performed in a symmetrical fashion to remove the possible outliers and form a set of n pairs. With this set it is possible to calculate the fundamental matrix as it is explained in [85].

6.2.2 CoMEDE and Optical Flow combination

Between the two cameras a dense correspondence estimation is carried out. This is obtained combining the CoMEDE procedure explained in Chapter 5 with the OF of the University of Dortmund. As it is summarized in Table 6.1 the two techniques present complementary features. With a RS based estimation it is possible to estimate large

Table 6.1: Advantages and disadvantages of RS based and OF based approaches: The two systems can be considered complementary.

	Recursive Search	Optical Flow
Assignment resolution	X (Block based)	✓ (Pixel based)
Large displacement estimation	✓	X
Vector precision	X (Integer resolution)	✓ (Float resolution)
Computational complexity	✓	X

correspondence vectors but only one vector per block is assigned to the output field. Furthermore, in case no interpolation is applied, the accuracy of the estimation is limited to integer accuracy. With an OF method it is possible to compute floating point accurate correspondence vectors for each pixel of the input image, but the estimation is limited to rather small correspondence vectors. This issue is generally addressed in literature with an initialization based on features to provide some anchor points for the subsequent OF estimation. Unfortunately a feature based initialization provides only a too sparse flow initialization and usually a trade-off between the number of features and their reliability has to be chosen. Eventually an OF system has a higher computational complexity than a RS based one. Then with the CoMEDE system we are able to provide a full dense and reliable flow initialization to a subsequent OF approach achieving at the same time a really precise estimation which is able to track large displacements. This is supported by the results that will be provided in Section 6.2.3. In particular the CoMEDE system is based on the BCC matching criteria and follow the RTC constraint. Specifically, since only two images are used, the only available Hamiltonian path is the trivial forward, backward consistency check. Then the OF used is an evolution of the Lucas & Kanade [4]. This is performed on the image color gradient in an iterative fashion.

6.2.3 Results

To prove the accuracy of the combination between the CoMEDE system with the OF we experimentally tested the MVF estimation on the Middlebury database [18]. Regarding the CoMEDE system the parameters are the same used in Section 5.4. The results, shown in Figure 6.2 in terms of objective evaluation and in Figures 6.3 and 6.4 for the MVF of sequences *RubberWhale* and *Grove2* respectively, illustrate how the combination of the two methods generally outperforms their single application.

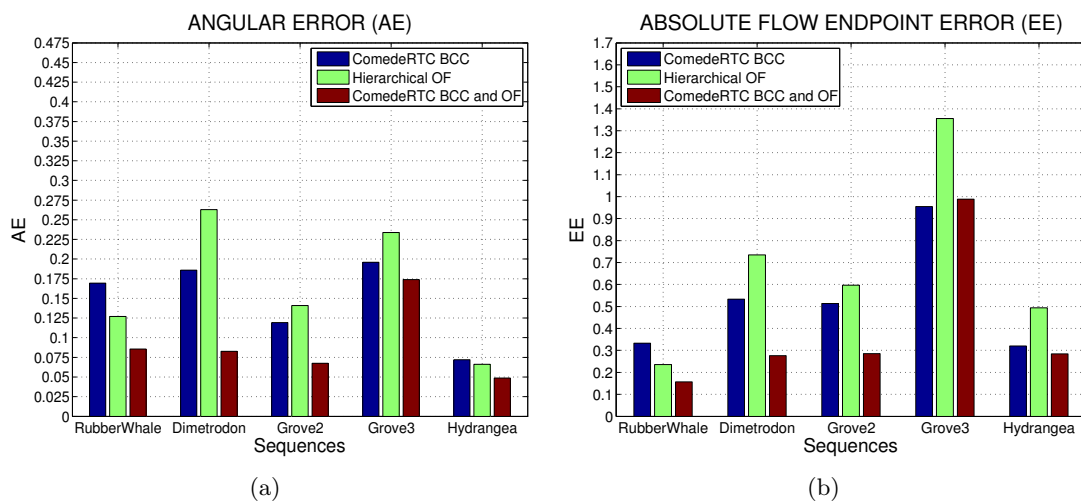


Figure 6.2: Results comparison of CoMEDE system and OF: (a) Angular error results, (b) Absolute flow endpoint error results.

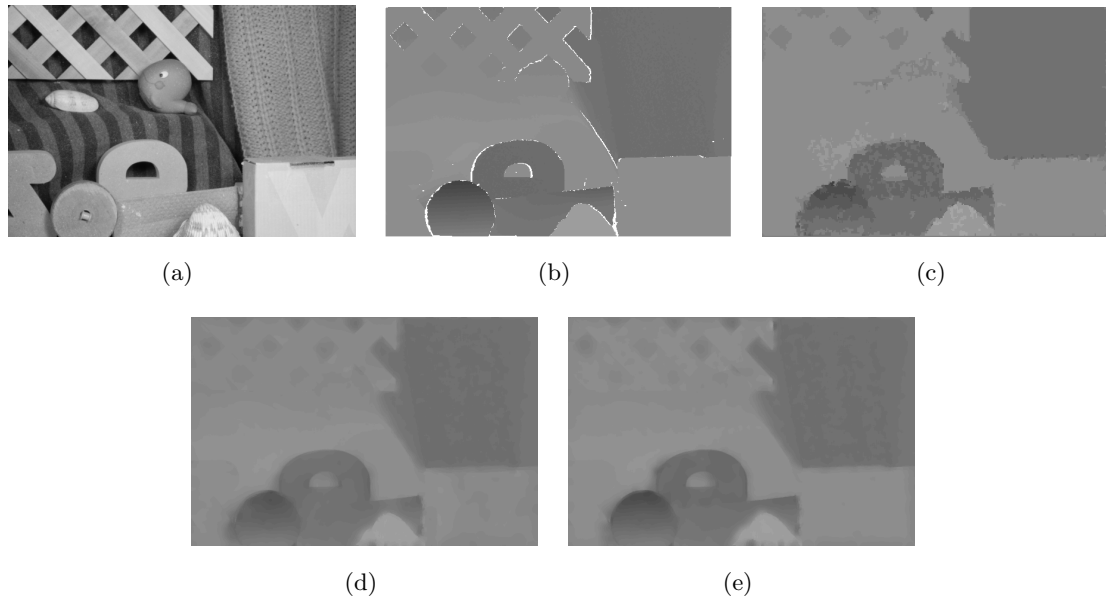


Figure 6.3: Results comparison in RubberWhale sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for CoMEDE with RTC constraint and BCC matching, (d) MVF for Hierarchical OF of TUDo, (e) MVF for CoMEDE with RTC constraint and BCC matching and OF.

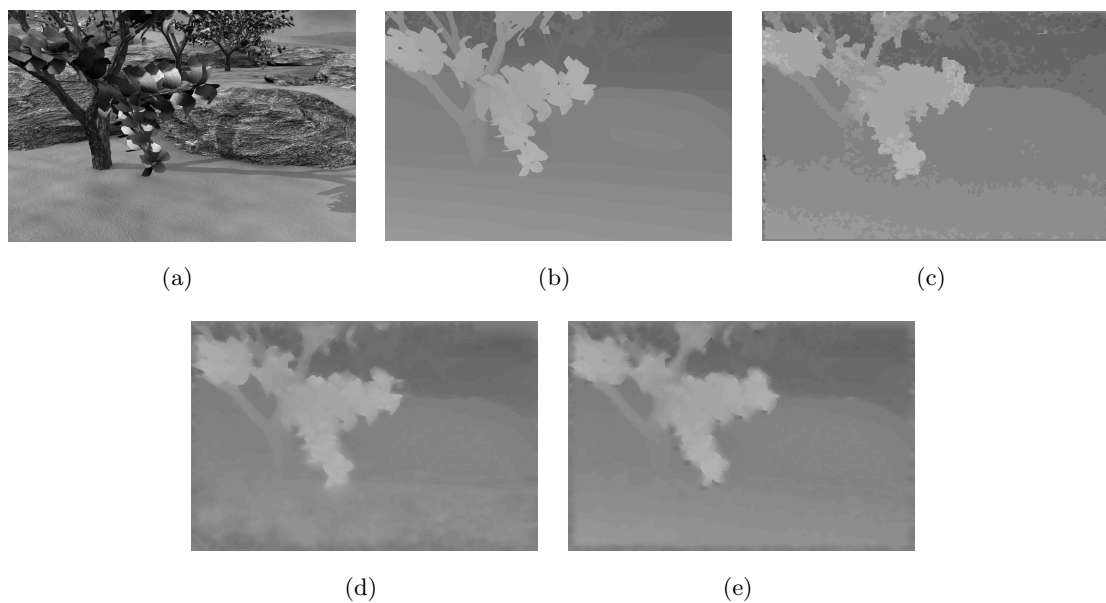


Figure 6.4: Results comparison in Grove2 sequence, (a) Original image, (b) Ground truth MVF, (c) MVF for CoMEDE with RTC constraint and BCC matching, (d) MVF for Hierarchical OF of Uni Dortmund, (e) MVF for CoMEDE with RTC constraint and BCC matching and OF.

In particular for the CoMEDE approach is not really possible to achieve an average absolute flow endpoint error in a sequence that is lower than 0.3 pixels. This is due to the integer pixel resolution of the estimation. On the other way by using the CoMEDE estimation as anchor point for a subsequent OF is definitely possible to improve the results with a final floating point resolution of the MVF. Another visual example of the benefit provided by the combination of the two techniques is shown in Figure 6.5. This Figure shows the results of the 3D reconstruction based on two views. As it was expected a



Figure 6.5: Results comparison in two view 3D reconstruction, (a) Features based 3D reconstruction, (b) University of Dortmund OF based 3D reconstruction, (c) CoMEDE based 3D reconstruction, (d) combination of CoMEDE and University of Dortmund OF based 3D reconstruction.

dense correspondence estimation (Figures 6.5(b), (c) and (d)) produces a much more complete reconstruction than a feature based one (Figure 6.5(a)). Moreover is possible to notice how a OF based 3D reconstruction (Figure 6.5(b)) permits a really smooth surface reconstruction but fails to estimate the correct depth in case of high displacements. Vice versa in the CoMEDE based reconstruction (Figure 6.5(c)) the calculated depth is correct but it is possible to notice that it is formed by a set of slices which are due to the integer pixel accuracy. Finally with the combination of the two techniques (Figure 6.5(d)) is

possible to achieve a reconstruction that is really precise and accurate. Obviously some outliers may occur, this is in particular true at the borders of the estimation where the object rotation is higher. In general these can be removed with a consistency check that will be explained when the extension to a multiview scenario will be presented in Section 6.3. In Figure 6.6 it is also shown another example of 3D reconstruction based on two views. In particular the Figure shows the differences between the usage of a SAD based



Figure 6.6: Results comparison in two view 3D reconstruction based on the combination of CoMEDE and University of Dortmund OF, (a) SAD matching criteria, (b) BCC matching criteria.

and a BCC based matching criteria. When multiple pictures are shot from different angles the results are prone to have illumination changes. Then with a SAD based estimation it is not possible to obtain a correct correspondence match. On the contrary thanks to the proposed BBC is possible to extract the fine texture that is quite unique in the human skin.

6.3 Multiple images

The extension to a multiple image framework is shown in Figure 6.7. Specifically the feature extraction and the camera calibration follow the same procedures already explained in the stereo scenario. The camera positions information is therefore used to feed pairwise the images into the dense estimator. Then an optional color based segmentation can be used to identify the object to be reconstructed. Eventually, before the final triangulation, a consistency check between all the dense estimation is applied [84].

6.3.1 Color based segmentation

When the target of the 3D reconstruction is an object which presents a color that differs from the background a segmentation can be applied to reduce the dense estimation results and consequently the final number of points that have to be triangulated. An initial over-segmentation is obtained by means of the *Mean-shift* algorithm [89] applied

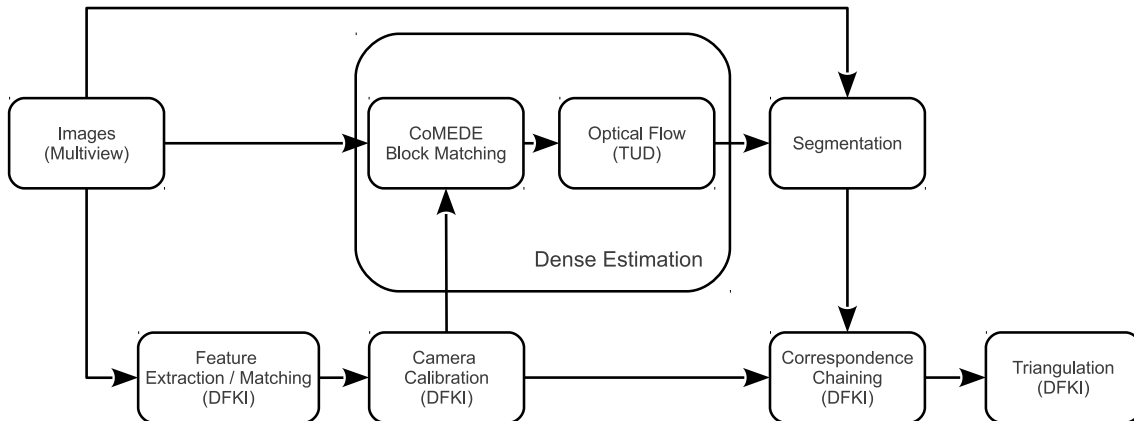


Figure 6.7: Framework for the 3D reconstruction with multiple cameras.

in the CIELab color space. Then the segments are merged together with a *K-means* [90] clustering where $k = 2$. Finally the dense estimation associated to the object cluster is extracted and used in the following Correspondence Chaining procedure. Moreover it should be noticed that the *Mean-shift* algorithm can be easily extended to the combination of non uniform spaces [19], e.g., color, reliability or the dense estimation.

6.3.2 Correspondence Chaining

In a two view reconstruction each 3D point is obtained from a couple of 2D points. Due to the extension into a multi-view reconstruction, multiple 2D points refer to a single 3D point. This allows a wider triangulation angle and a consequent more precise estimation of the 3D point position. Since the applied dense estimation provides correspondences between image pairs, different chains can be built by linking together consecutive estimations. These chains are formed consequently to a validity check based on the reprojection error or on an extended round trip check [84]. The Correspondence Chaining produces two main advantages for the subsequent triangulation. First, it merges together the linked points reducing the final pointcloud and consequently the related processing time. Second, it removes outliers for the subsequent triangulation. In fact the chains with an associated length equal to two or less are removed since they usually refer to a wrong dense estimation or occlusion.

6.3.3 Results

In our experiments the two different datasets shown in Figure 6.8 were considered. For both cases a white background was used and after every shot the object was manually rotated to obtain a random rotation angle between the consecutive images. The first dataset presents a set of 27 images of a Chinese *Lion* reproduction while the second is formed by 29 images representing the *Civetta* sculpture of Gino Cortelazzo [91]. For these

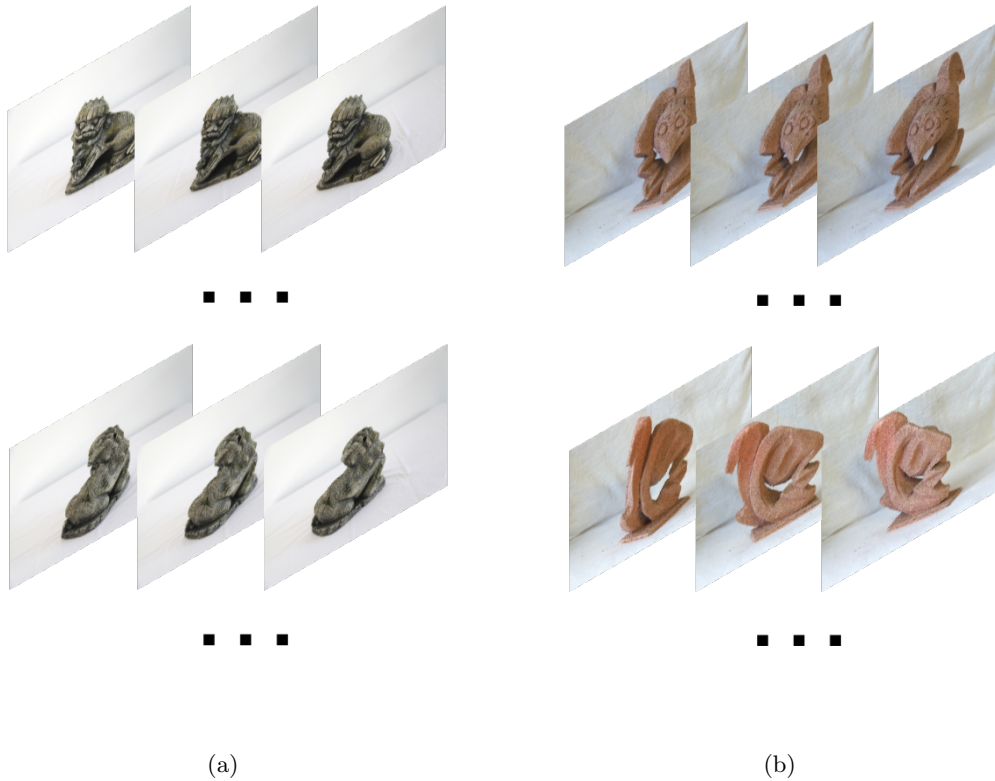


Figure 6.8: Datasets for the 3D reconstruction, (a) *Lion*, (b) *Civetta* sculpture of Gino Cortelazzo [91].

objects a ground truth is available. For the *Lion* this was obtained with the ORCAM system in DFKI [71] while for the *Civetta* a *NextEngine* 3D laser scanner [70] was used. Finally the graphical results of the proposed reconstruction are shown in Figures 6.9(a), 6.9(d), 6.10(a) and 6.10(d). Specifically, after the framework depicted in Figure 6.7 the system provides a dense pointcloud. Due to its density a mesh can be calculated. For this purpose the Poisson based meshing procedure available in [92] was used, see Figures 6.9(b), 6.9(e), 6.10(b) and 6.10(e). Finally, the textures can be mapped to the meshes obtaining the results shown in Figures 6.9(c), 6.9(f), 6.10(c) and 6.10(f). In order to obtain an objective evaluation the ground truths and the reconstructed objects have to be aligned. In particular, since the distance between the cameras is not known, the objects can be only reconstructed up to a multiplicative constant. Then between the two models there is not only a roto-translation transformation but also a scaling factor. To overcome this issue the point based glueing of *MeshLab* [92] was used. In this method a set of n conjugate points have to be marked manually. Unfortunately this method is prone to errors but it is suitable for the scaling factor estimation. A final alignment is obtained by iterative closest point (ICP) [93]. Then the shortest distances between the points of the reference and the reconstructed surfaces were calculated. A threshold of 1 [cm] as maximum distance was also set. This limits the error in case that the related ground truth is not complete, as in the *Civetta* dataset, or there are points in the object itself that can not be reconstructed,

e.g., the bottom of the object in both the datasets.

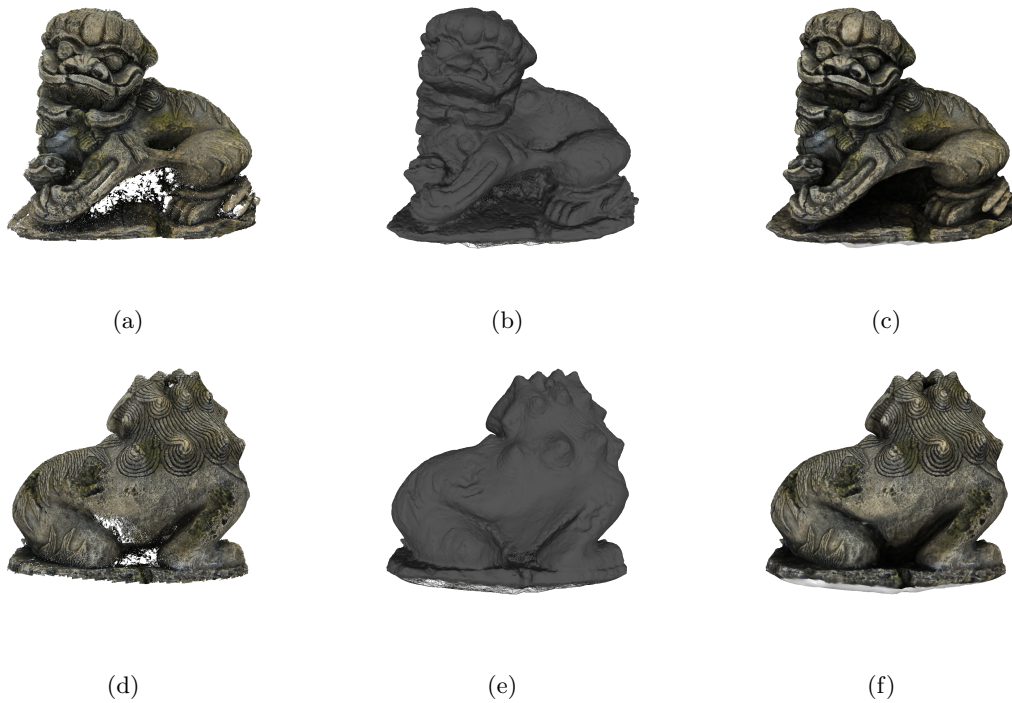


Figure 6.9: 3D reconstruction of the *Lion* dataset, (a) Pointcloud, front view, (b) Mesh, front view, (c) Textured mesh, front view, (d) Pointcloud, back view, (e) Mesh, back view, (f) Textured mesh, back view.



Figure 6.10: 3D reconstruction of the *Civetta* dataset, (a) Pointcloud, front view, (b) Mesh, front view, (c) Textured mesh, front view, (d) Pointcloud, back view, (e) Mesh, back view, (f) Textured mesh, back view.

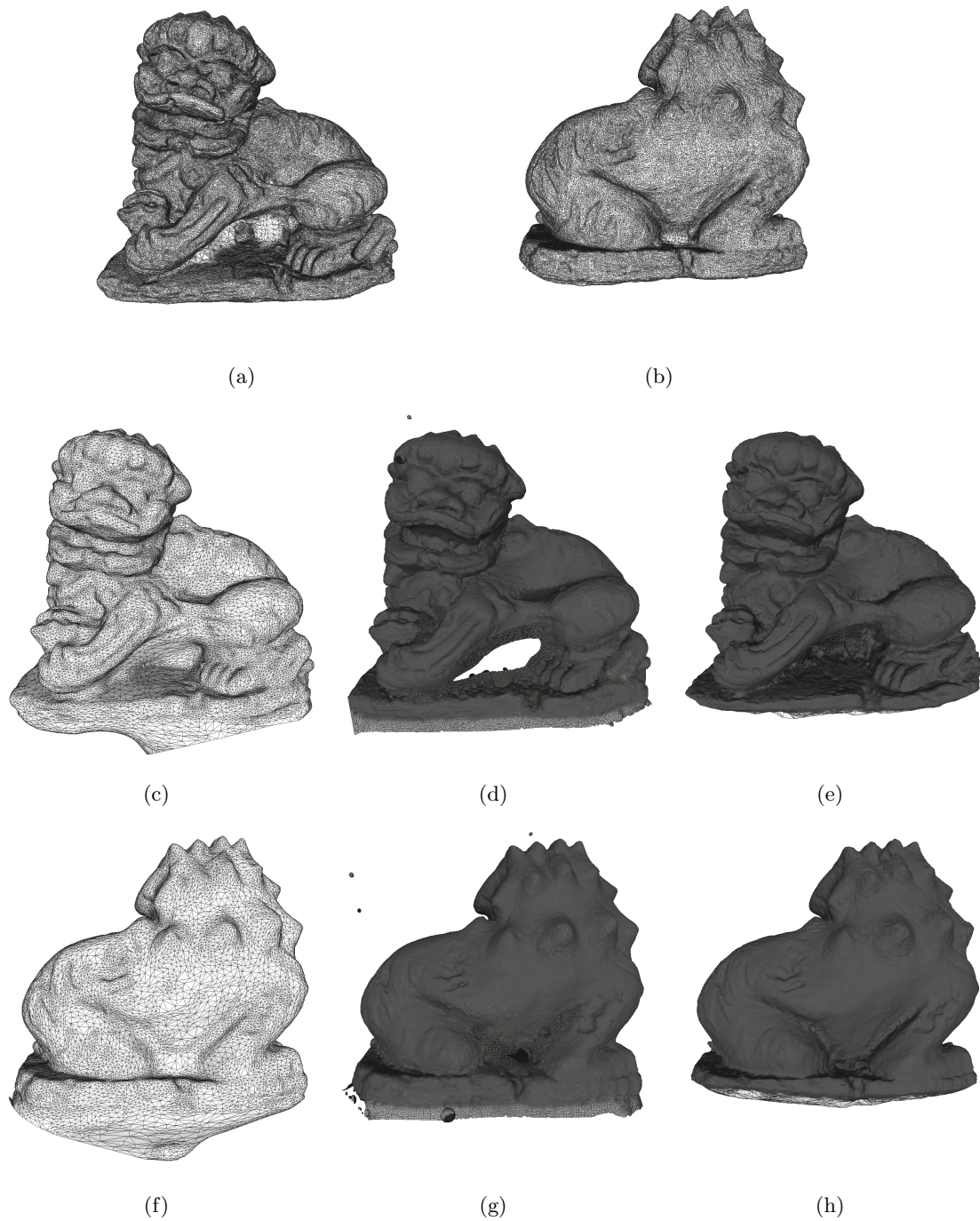


Figure 6.11: 3D reconstruction of the *Lion* dataset, front view: (a) Ground truth mesh, (c) Reconstructed mesh with Autocad123D, (d) Reconstructed mesh with 3DF Zephyr PRO, (e) Reconstructed mesh with the proposed method, back view: (b) Ground truth mesh, (f) Reconstructed mesh with Autocad123D, (g) Reconstructed mesh with 3DF Zephyr PRO, (h) Reconstructed mesh with the proposed method.

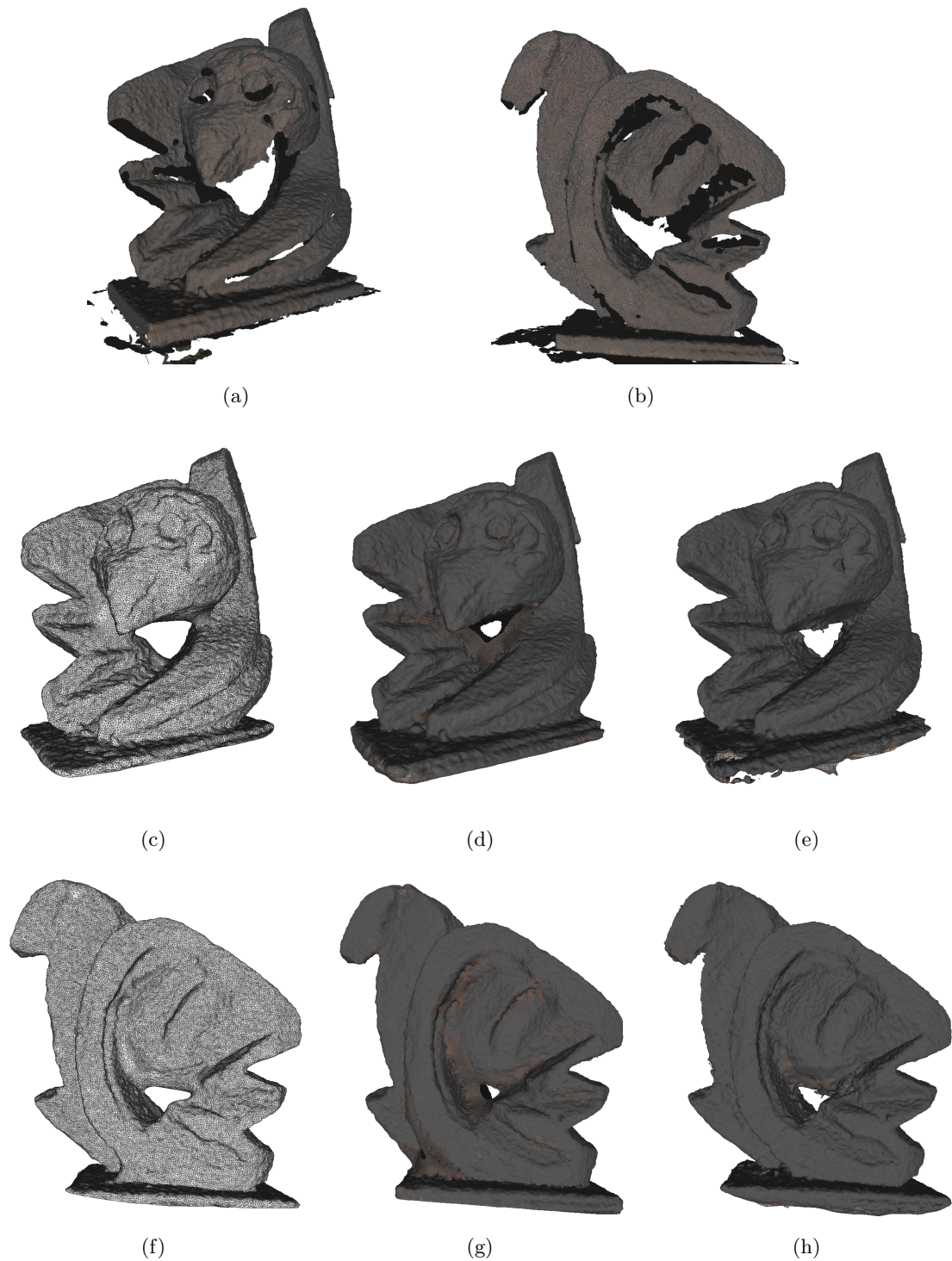


Figure 6.12: 3D reconstruction of the *Civetta* dataset, front view: (a) Ground truth mesh, (c) Reconstructed mesh with Autocad123D, (d) Reconstructed mesh with 3DF Zephyr PRO, (e) Reconstructed mesh with the proposed method, back view: (b) Ground truth mesh, (f) Reconstructed mesh with Autocad123D, (g) Reconstructed mesh with 3DF Zephyr PRO, (h) Reconstructed mesh with the proposed method.

Table 6.2: 3D reconstruction of the *Lion* dataset.

Method	RMS Error	RMS Error %
Autocad 123D [75]	1.340 [mm]	0.433%
3DF Zephyr PRO [76]	1.479 [mm]	0.478%
Proposed	1.285 [mm]	0.415%

Table 6.3: 3D reconstruction of the *Civetta* dataset.

Method	RMS Error	RMS Error %
Autocad 123D [75]	1.646 [mm]	0.268%
3DF Zephyr PRO [76]	1.481 [mm]	0.241%
Proposed	1.454 [mm]	0.236%

The results in term of RMS error and the related percentage calculated with respect to the bounding box diagonal are reported in Tables 6.2 and 6.3. With all the methods an accuracy between $1 - 2$ [mm] is achieved. Then due to the ambiguity in the object alignment we can only assume that all the method present comparable results. Moreover a visual presentation of the results is available in Figures 6.11 and 6.12. The first noticeable difference between the methods is the mesh density. This is particular evident in the *Lion* dataset. The meshes in the *Autocad123D* based 3D reconstruction are quite large, see Figure 6.11(c), while for the proposed method, see Figure 6.11(e), as well for the *3DF Zephyr PRO*, see Figure 6.11(d), the density is even higher than the related ground truth, see Figure 6.11(a). To this density is also associated a richness of details. In particular it is possible to see that we are even able to reconstruct the mane of the lion, see Figure 6.11(h). On the other side when the calibration and the dense estimation do not share the same precision there could be some problem in the triangulation. In particular this could lead to the formation of multiple overlapped surfaces in the pointclouds. Then the Poisson based meshing tries to follow all these point based surfaces resulting in a nubby set of meshes as in the top back of the *Lion*, see Figure 6.11(e). This problem is already reduced thanks to the correspondence chaining [84] but is still an open issue that should be addressed in future works.

6.4 ToF support

An initial investigation on a possible ToF support for the 3D reconstruction was also considered. The overview of the developed framework, which extends the stereo one shown in Figure 6.1, is depicted in Figure 6.13. The array composed by the ToF and color cameras is now considered as input, specifically only the camera ① and ③ are considered to take advantage of a wider baseline. Then the depth super-resolution proposed in Chapter

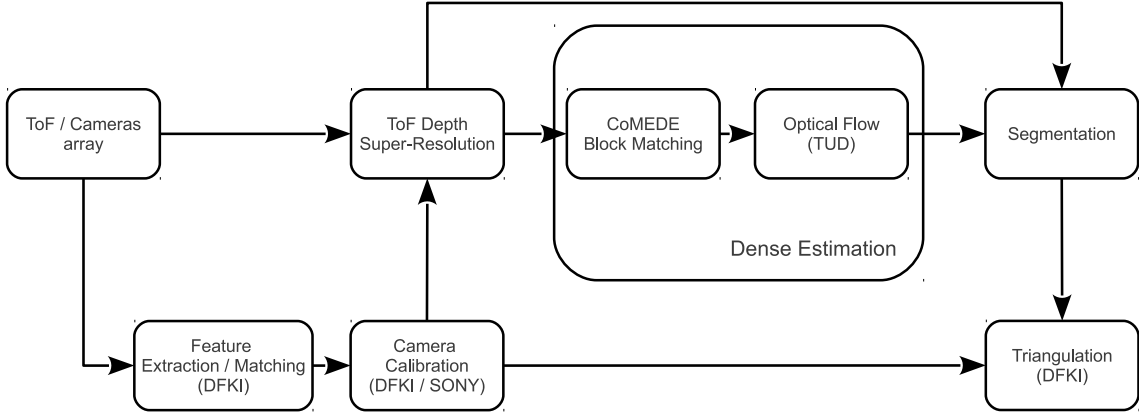


Figure 6.13: Framework for the 3D reconstruction based on the ToF and cameras array.

4 is carried out. This can be used as initialization of the subsequent dense estimation and to perform a foreground-background segmentation which extracts the object to be reconstructed. For the dense estimation initialization the depth has to be firstly mapped into disparity. This can be calculated as:

$$\begin{cases} d_x(x_{RGB_1}, y_{RGB_1}) = x_{RGB_1} - x_{RGB_3} \\ d_y(x_{RGB_1}, y_{RGB_1}) = y_{RGB_1} - y_{RGB_3} \end{cases} \quad (6.1)$$

where the following relation between every couple of associated points (x_{RGB_1}, y_{RGB_1}) and (x_{RGB_3}, y_{RGB_3}) holds,

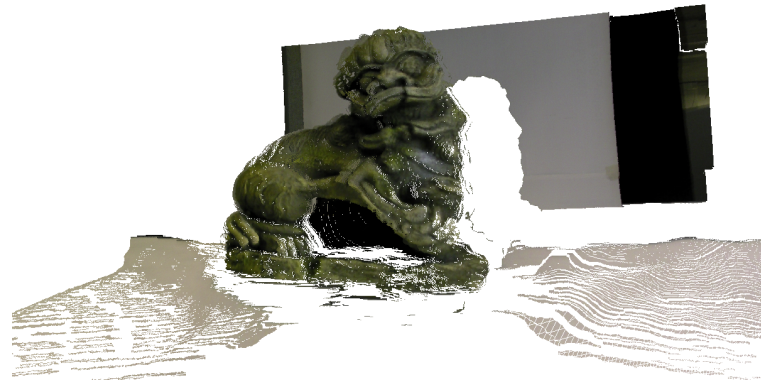
$$\begin{pmatrix} x_{RGB_3} \\ y_{RGB_3} \\ 1 \end{pmatrix} = \begin{pmatrix} x'_{RGB_3}/z_{RGB_3} \\ y'_{RGB_3}/z_{RGB_3} \\ 1 \end{pmatrix} = K_{RGB_3} [R|t] \begin{bmatrix} K_{RGB_1}^{-1} \begin{pmatrix} x_{RGB_1} \cdot z_{RGB_1} \\ y_{RGB_1} \cdot z_{RGB_1} \\ z_{RGB_1} \end{pmatrix} \\ 1 \end{bmatrix}. \quad (6.2)$$

In particular z_{RGB_1} is the output of the depth super-resolution processed into camera ① and the K_{RGB_1} , K_{RGB_3} and $[R|t]$ are the intrinsic and extrinsic cameras parameters. The ToF based segmentation is similar to the procedure already explained in Section 6.3.1. A *Mean-shift* on the super-resolved depth provides an initial over-segmentation which is subsequently densified in two clusters applying a *k-means* with $k = 2$. The results are shown in Figure 6.14. Specifically, Figure 6.14(a) shows the 3D reconstruction based on the ToF depth super-resolved only. It is possible to see that the procedure explained in Chapter 4 is able to provide a quite accurate super-resolved depth map. Unfortunately this is limited by the ToF resolution and is further reduced with the filtering in the super-resolution process. On the other hand the accuracy along the object border is quite high and therefore it is suitable to divide the foreground from the background and to obtain the 3D reconstruction depicted in Figure 6.14(b). Finally the dense estimation explained

in Section 6.2.2 is able to increase the depth resolution of the final 3D reconstruction, see Figure 6.14(c).

6.5 Conclusion and future work

In this Chapter a novel method for the 3D reconstruction from a set of not calibrated images was presented. This was obtained merging the 3D reconstruction framework of the DFKI with the CoMEDE system and the OF of the TUDo. It was demonstrated that the combination of the CoMEDE and the OF outperforms the single estimations. This was proved by an objective evaluation based on the Middlebury database for optical flow [18] and a subjective evaluation of the two view 3D reconstruction. Then the extension to a multiview framework was introduced. This is suitable for an accurate reconstruction of objects and achieves comparable results with the present state of the art. Finally, an initial integration of the ToF based super-resolution presented in Chapter 4 is proposed. While the results prove the possibility of a quite precise reconstruction, some possible improvements are possible. One of the main problem in the current 3D reconstruction implementation is that the calibration and the estimation precision do not match. This may produce nubby meshes as in the top back of the *Lion*. A possible solution could be to perform the camera calibration by means of the dense estimation only. Specifically, an initial feature based calibration [94] could detect the neighbor images. After that all the dense estimations between the couples are processed, the correspondence chaining can be applied. Eventually, only the longest and consequently reliable chains can be used for the camera calibration. The results would be a pointcloud where the 3D points and the camera calibration are jointly optimized [84]. Another possible improvement is in the dense estimation. At the current status the OF is used as refinement step of the CoMEDE estimation to obtain a pixel based estimation with float accuracy. Then a more close combination between the CoMEDE and the OF systems should be studied. Regarding the ToF estimation this should be more deeply integrated in the system. In particular this should be extended to a multi-position camera array for which the calibration within the array is fixed while the motion is calculated inside the 3D reconstruction. Finally the whole system should be more flexible. At the current status the multiple views 3D reconstruction impose that the sequence of pictures follow a circular order. This should be extended to a graph structure in which the images are unordered and they can have multiple predecessors and successors [84].



(a)



(b)



(c)

Figure 6.14: Results comparison in ToF based 3D reconstruction, (a) ToF depth super-resolved only, (b) ToF depth super-resolved and ToF based segmentation, (c) ToF depth super-resolved and CoMEDE with OF after ToF based segmentation.

Chapter 7

Conclusions

This thesis provides an overview of the research that was done along the three years of the Ph.D. studies in the field of motion and disparity estimations. We start our study in Chapter 2 by introducing the problem of motion estimation. Two contributions are described in this field. The first is a novel matching criterion based on a cross correlation between frames beforehand binarized. This significantly reduces the computational cost while maintaining comparable results with respect to a standard recursive search method. The second contribution refers to a parallel algorithm which is suitable for a GPU based implementation. In Chapter 3 we consider the disparity estimation problem. We propose an offline processing based on a possible user interaction to improve the final quality of the Simple Tree stereo algorithm [2]. This method shows the possibility to align the disparity edges with the object borders by using a feedback loop in the smoothness cost of the algorithm. Further studies should be done in order to adapt the framework to an automatic disparity refinement. In Chapter 4 we consider the combination of active and passive systems for the purpose of geometry estimation. We propose an algorithm for the super-resolution of a ToF depth map driven by a high resolution RGB camera. Specifically we develop the method and compare it with the state of the art approaches in a controlled scenario. Numerical results show that we are able to outperform the other methods, in particular with an increasing level of noise. The procedure is then adapted to a real set of data and it permits to achieve an accurate and high resolution depth map. In Chapter 5 we consider the problem of a joint motion and disparity estimation. We introduce a quality evaluation with the concept of RTC and we use this to loosely couple a set of independent estimations. With this solution we try to address the aperture problem and to obtain a more robust estimation. This is then used in the 3D reconstruction framework that we present in Chapter 6. Specifically, we show the the outcome of a collaboration with the TUDo and with the DFKI. We show how the combination of the CoMEDE approach with the TUDo OF outperforms the single estimations. This combination is therefore used in the 3D reconstruction pipeline of the DFKI to achieve a dense and accurate objects reconstruction. Numerical results show that comparable results with the state of the art methods can be obtained.

Bibliography

- [1] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, "True-motion estimation with 3-d recursive search block matching," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 3, no. 5, pp. 368–379, 388, 1993.
- [2] M. Bleyer and M. Gelautz, "Simple but effective tree structures for dynamic programming-based stereo matching," in *VISAPP (2)*, 2008, pp. 415–422.
- [3] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [4] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.
- [5] D. A. Marzat J., Dumortier Y., "Real-time dense and accurate parallel optical flow using cuda," in *Proceedings of the 17th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, ser. WSCG'09, 2009.
- [6] C. Kuglin and D. Hines, "The phase correlation image alignment method," in *In Proceedings of the IEEE 1975 International Conference on Cybernetics and Society, San Francisco*, September 1975.
- [7] F. Kelly, "Fast Probabilistic Inference and GPU Video Processing," Ph.D. dissertation, Trinity College Dublin, May 2006.
- [8] M. Peeters, "Implementation of the Phase Correlation Algorithm - Motion Estimation in the Frequency Domain," Ph.D. dissertation, Eindhoven University of Technology, 2003.
- [9] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit-plane matching," in *Image Processing, 1995. Proceedings., International Conference on*, vol. 3, 1995, pp. 496–499 vol.3.

-
- [10] G. de Haan and P. Biezen, "An efficient true-motion estimator using candidate vectors from a parametric motion model," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 1, pp. 85–91, 1998.
- [11] J. Wang, D. Wang, and W. Zhang, "Temporal compensated motion estimation with simple block-based prediction," *Broadcasting, IEEE Transactions on*, vol. 49, no. 3, pp. 241–248, 2003.
- [12] G.-G. Lee, M.-J. Wang, H.-Y. Lin, D. W.-C. Su, and B.-Y. Lin, "Algorithm/architecture co-design of 3-d spatio-temporal motion estimation for video coding," *Multimedia, IEEE Transactions on*, vol. 9, no. 3, pp. 455–465, 2007.
- [13] C. Bartels and G. de Haan, "Smoothness constraints in recursive search motion estimation for picture rate conversion," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 10, pp. 1310–1319, 2010.
- [14] F. Michielin, G. Calvagno, P. Sartor, and O. Erdler, "A true motion estimation method based on binarized cross correlation," in *Consumer Electronics - Berlin (ICCE-Berlin), 2013 IEEE International Conference on*, 2013.
- [15] O. C. L. Au and M. C. Kung., "Block parallel and fast motion estimation in video coding," Patent U.S. Patent 2009/0 268 821, Oct 29, 2009.
- [16] P. Sartor and F. Michielin, "Multi-mode motion estimation with binarized cross correlation," Patent.
- [17] P. Sartor, F. Michielin, T. Emmerich, and C. Unruh, "Parallel motion estimation," Patent.
- [18] S. Baker, S. Roth, D. Scharstein, M. Black, J. P. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1–8.
- [19] C. Dal Mutto, "Acquisition and Processing of ToF and Stereo Data," Ph.D. dissertation, University of Padua (Italy), 2013.
- [20] A. Fusiello, *Visione Computazionale. Tecniche di Ricostruzione Tridimensionale*. Milano: Franco Angeli, 2013.
- [21] R. Streubel, "Simultaneous time-of-flight and stereo camera calibration," Master's thesis, University of Stuttgart (Germany), 2012.
- [22] C. Hernández, G. Vogiatzis, and Y. Furukawa, "3d shape reconstruction from photographs: A multi-view stereo approach," San Francisco, June 2010. [Online]. Available: <http://cvl.umiacs.umd.edu/conferences/cvpr2010/tutorials/>

- [23] J.-Y. Bouguet. Camera calibration toolbox for matlab. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/
- [24] A. Vianello, “Depth super-resolution with hybrid camera system,” Master’s thesis, University of Padua (Italy), 2013.
- [25] P. Sartor and F. Michielin, “Automatic refinement of user assisted object segmentation,” Patent.
- [26] L. Di Stefano, M. Marchionni, and S. Mattoccia, “A fast area-based stereo matching algorithm,” *Image and Vision Computing*, vol. 22, no. 12, pp. 983–1005, Oct 2004.
- [27] C. Zach, K. Karner, and H. Bischof, “Hierarchical disparity estimation with programmable 3D Hardware,” in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2004, pp. 275–282.
- [28] A. S. Ogale and Y. Aloimonos, “Shape and the stereo correspondence problem,” *Int. J. Comput. Vision*, vol. 65, no. 3, pp. 147–162, Dec. 2005. [Online]. Available: <http://dx.doi.org/10.1007/s11263-005-3672-3>
- [29] E. Binaghi, I. Gallo, M. Raspanti, and G. Marino, “Neural adaptive stereo matching,” *PATTERN RECOGNITION LETTERS*, vol. 25, pp. 1743–1758, 2004.
- [30] Z. Lee, J. Juang, and T. Nguyen, “Local disparity estimation with three-moded cross census and advanced support weight,” *Multimedia, IEEE Transactions on*, vol. 15, no. 8, pp. 1855–1864, 2013.
- [31] L. Nalpantidis, A. Gasteratos, and G. Sirakoulis, “Review of stereo vision algorithms,” *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.
- [32] K. jin Yoon, S. Member, and I. S. Kweon, “Adaptive support-weight approach for correspondence search,” *IEEE Trans. PAMI*, vol. 28, pp. 650–656, 2006.
- [33] P. Mordohai and G. Medioni, “Stereo using monocular cues within the tensor voting framework,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 6, pp. 968–982, 2006.
- [34] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 3rd ed. Springer Publishing Company, Incorporated, 2009.
- [35] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, Jul. 2003. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2003.1206509>
- [36] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001. [Online]. Available: <http://dx.doi.org/10.1109/34.969114>

- [37] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, “A maximum likelihood stereo algorithm,” *Computer Vision and Image Understanding*, vol. 63, pp. 542–567, 1996.
- [38] H. Hirschmuller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 2005, pp. 807–814 vol. 2.
- [39] S. Birchfield and C. Tomasi, “Depth discontinuities by pixel-to-pixel stereo,” *International Journal of Computer Vision*, vol. 35, pp. 1073–1080, 1996.
- [40] J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer, “Fast anisotropic gauss filtering,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, 2003. [Online]. Available: <http://www.science.uva.nl/research/publications/2003/GeusebroekTIP2003>
- [41] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.” in *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002., 2002.
- [42] V. Garro, C. Dal Mutto, P. Zanuttigh, and G. Cortelazzo, “A novel interpolation scheme for range data with side information,” in *Visual Media Production, 2009. CVMP '09. Conference for*, 2009, pp. 52–60.
- [43] Q. Yang, R. Yang, J. Davis, and D. Nister, “Spatial-depth super resolution for range images,” in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.
- [44] J. Diebel and S. Thrun, “An application of markov random fields to range sensing,” in *In NIPS*. MIT Press, 2005, pp. 291–298.
- [45] T. Prasad, K. Hartmann, W. Weihs, S. Ghobadi, and A. Sluiter, “First steps in enhancing 3d vision technique using 2d/3d sensors,” in *In 11th Computer Vision Winter Workshop*. O. Franc Ed., 2006, p. 82?86.
- [46] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral upsampling,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, vol. 26, no. 3, p. to appear, 2007.
- [47] C. Kim, H. Yu, and G. Yang, “Depth super resolution using bilateral filter,” in *Image and Signal Processing (CISP), 2011 4th International Congress on*, vol. 2, 2011, pp. 1067–1071.
- [48] C. Dal Mutto, P. Zanuttigh, S. Mattoccia, and G. Cortelazzo, “Locally consistent tof and stereo data fusion,” in *Computer Vision ? ECCV 2012. Workshops and*

- Demonstrations*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7583, pp. 598–607.
- [49] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun, “Multi-view image and tof sensor fusion for dense 3d reconstruction,” in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 2009, pp. 1542–1549.
- [50] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–.
- [51] P. Sartor, A. Vianello, and F. Michielin, “Sensor fusion for depth super-resolution,” Patent.
- [52] Pmd technologies. [Online]. Available: <http://www.pmdtec.com/>
- [53] Mesa imaging. [Online]. Available: <http://www.mesa-imaging.ch/>
- [54] Softkinetic. [Online]. Available: <http://www.softkinetic.com/>
- [55] Microsoft®. [Online]. Available: <http://www.microsoft.com/>
- [56] M. Schmidt and B. Jähne, “A physical model of time-of-flight 3d imaging systems, including suppression of ambient,” in *Dynamic 3D Imaging*. Springer, 2009, pp. 1–15.
- [57] T. Oggier, B. Buttgen, F. Lustenberger, G. Becker, B. Ruegg, and A. Hodac, “Swissranger sr3000 and first experiences based on miniaturized 3d-tof cameras.” in *Proceedings of the First Range Imaging Research Day*, ETH Zurich, 2005.
- [58] F. Mufti and R. Mahony, “Statistical analysis of measurement processes for time-of-flight cameras,” in *Proceedings of SPIE the International Society for Optical Engineering*, 2009.
- [59] M. Schmidt, “Analysis, modeling and dynamic optimization of 3d time-of-flight imaging systems,” Ph.D. dissertation, University of Heidelberg (Germany), 2011.
- [60] Lux media plan. [Online]. Available: <http://luxmediaplan.de/>
- [61] H. Hirschmüller and D. Scharstein, “Evaluation of cost functions for stereo matching,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, vol. 0, pp. 1–8, 2007.
- [62] F. Pukelsheim, “The Three Sigma Rule,” *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

- [63] D. B. Min, H. Kim, and K. Sohn, “Edge-preserving joint motion-disparity estimation in stereo image sequences.” in *Proceedings of the 6th IASTED international conference on signal and image processing*, 2004.
- [64] ———, “Edge-preserving joint motion-disparity estimation in stereo image sequences.” *Sig. Proc.: Image Comm.*, vol. 21, pp. 252–271, 2006.
- [65] I. Patras, E. A. Hendriks, and G. Tziritas, “A joint motion/disparity estimation method for the construction of stereo interpolated images in stereoscopic image sequences.”
- [66] J. Cech and R. P. Horaud, “Joint Disparity and Optical Flow by Correspondence Growing,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Prague, Czech Republic: IEEE, 2011, pp. 893–896. [Online]. Available: <http://perception.inrialpes.fr/Publications/2011/CH11>
- [67] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt, “Joint estimation of motion, structure and geometry from stereo sequences,” in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ser. ECCV’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 568–581. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1888089.1888133>
- [68] J. Fritsch, T. Kuehnl, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [69] O. Schreer, P. Kauff, and T. Sikora, *3D Videocommunication: Algorithms, Concepts and Real-time Systems in Human Centred Communication*. John Wiley & Sons, 2005.
- [70] Nextengine. [Online]. Available: <http://www.nextengine.com/>
- [71] Orcam. [Online]. Available: http://av.dfki.de/publications_2013/faithful-compact-and-complete-digitization-of-cultural-heritage-using-a-full-spherical-scanner
- [72] A. Laurentini, “The visual hull concept for silhouette-based image understanding,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 2, pp. 150–162, 1994.
- [73] Bundler. [Online]. Available: <http://www.cs.cornell.edu/~snaveily/bundler/>
- [74] N. Snavely, S. Seitz, and R. Szeliski, “Modeling the world from internet photo collections,” *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s11263-007-0107-3>
- [75] Autoca123d. [Online]. Available: <http://www.123dapp.com/>

- [76] 3df zephyr pro. [Online]. Available: <http://www.3dflow.net/3df-zephyr-pro-3d-models-from-photos/>
- [77] E. Cappelletto, P. Zanuttigh, and G. Cortelazzo, "Handheld scanning with 3d cameras," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, 2013, pp. 367–372.
- [78] Lynx. [Online]. Available: <http://www.lynxlaboratories.com>
- [79] F. Bernardini and H. E. Rushmeier, "The 3d model acquisition pipeline." *Comput. Graph. Forum*, vol. 21, no. 2, pp. 149–172, 2002. [Online]. Available: <http://dblp.uni-trier.de/db/journals/cgf/cgf21.html#BernardiniR01>
- [80] M. Andreetto, N. Brusco, and G. Cortelazzo, "Automatic 3d modeling of textured cultural heritage objects," *Image Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 354–369, 2004.
- [81] M. I. A. Lourakis and A. A. Argyros, "Sba: a software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, pp. 1–30, 2009.
- [82] F. Michielin, M. Brueggemann, B. Krolla, O. Erdler, P. Sartor, P. Springer, E. Thimo, and Y. Incesu, "Combined recursive block matching and optical flow correspondence vector estimation," Patent.
- [83] O. Wasenmueller, B. Krolla, Y. Incesu, E. Thimo, and F. Michielin, "Correspondence chaining for precise dense 3d reconstruction," Patent.
- [84] O. Wasenmueller, "Enhancing dense 3d models by multi-view triangulation and subsequent illumination estimation," Master's thesis, DFKI (Germany), 2013.
- [85] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.
- [86] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>
- [87] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.cviu.2007.09.014>
- [88] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, May 2011.
- [89] D. Comaniciu, P. Meer, and S. Member, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

-
- [90] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.
- [91] Ginocortelazzo. [Online]. Available: <http://ginocortelazzo.it/>
- [92] P. Cignoni, M. Corsini, and G. Ranzuglia, “Meshlab: an open-source 3d mesh processing system,” *ERCIM News*, no. 73, pp. 45–46, April 2008.
- [93] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” *Proc. SPIE*, vol. 1611, pp. 586–606, 1992.
- [94] O. Wasenmueller and B. Krolla, “Optimized image matching for efficient 3d reconstruction,” Master’s thesis, DFKI (Germany), 2012.

List of Publications

The work presented in this thesis has in part been published in the references listed below.

Patents

- [P1] P. Sartor and F. Michielin, “Multi-mode motion estimation with binarized cross correlation,” *Patent*.
- [P2] P. Sartor, F. Michielin, T. Emmerich, and C. Unruh, “Parallel motion estimation,” *Patent*.
- [P3] P. Sartor and F. Michielin, “Automatic refinement of user assisted object segmentation,” *Patent*.
- [P4] P. Sartor, A. Vianello, and F. Michielin, “Sensor fusion for depth super-resolution,” *Patent*.
- [P5] F. Michielin, M. Brueggemann, B. Krolla, O. Erdler, P. Sartor, P. Springer, E. Thimo, and Y. Incesu, “Combined recursive block matching and optical flow correspondence vector estimation,” *Patent*.
- [P6] O. Wasenmueller, B. Krolla, Y. Incesu, E. Thimo, and F. Michielin, “Correspondence chaining for precise dense 3d reconstruction,” *Patent*.

Conference Papers

- [C1] F. Michielin, G. Calvagno, P. Sartor, and O. Erdler, “A Wavelets base deblocking technique for DCT based compressed materials,” in *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on*, 2012.
- [C2] F. Michielin, G. Calvagno, P. Sartor, and O. Erdler, “A true motion estimation method based on binarized cross correlation,” in *Consumer Electronics - Berlin (ICCE-Berlin), 2013 IEEE International Conference on*, 2013.

- [C3] F. Michielin, G. Calvagno, P. Sartor, and O. Erdler, “A wavelets based de-ringing technique for DCT based compressed visual data,” in *ICIP 2013 Melbourne*, Sept. 15-18 2013.