University of Padova

Department of Information Engineering

Ph.D. School of Information Engineering

Information Science and Technology

XXIX class

3D data fusion from multiple sensors and its applications

Giulio Marin

Supervisor:
Pietro Zanuttigh, Ph.D.

Ph.D. School director:
Prof. Matteo Bertocco

January 31, 2017

A Ilene, che mi è sempre vicina.

Abstract

The introduction of depth cameras in the mass market contributed to make computer vision applicable to many real world applications, such as human interaction in virtual environments, autonomous driving, robotics and 3D reconstruction. All these problems were originally tackled by means of standard cameras, but the intrinsic ambiguity in the bidimensional images led to the development of depth cameras technologies. Stereo vision was first introduced to provide an estimate of the 3D geometry of the scene. Structured light depth cameras were developed to use the same concepts of stereo vision but overcome some of the problems of passive technologies. Finally, Time-of-Flight (ToF) depth cameras solve the same depth estimation problem by using a different technology.

This thesis focuses on the acquisition of depth data from multiple sensors and presents techniques to efficiently combine the information of different acquisition systems. The three main technologies developed to provide depth estimation are first reviewed, presenting operating principles and practical issues of each family of sensors. The use of multiple sensors then is investigated, providing practical solutions to the problem of 3D reconstruction and gesture recognition. Data from stereo vision systems and ToF depth cameras are combined together to provide a higher quality depth map. A confidence measure of depth data from the two systems is used to guide the depth data fusion. The lack of datasets with data from multiple sensors is addressed by proposing a system for the collection of data and ground truth depth, and a tool to generate synthetic data from standard cameras and ToF depth cameras. For gesture recognition, a depth camera is paired with a Leap Motion device to boost the performance of the recognition task. A set of features from the two devices is used in a classification framework based on Support Vector Machines and Random Forests.

Sommario

L'introduzione di sensori di profondità nel mercato di massa ha contribuito a rendere la visione artificiale applicabile in molte applicazioni reali, come l'interazione dell'uomo in ambienti virtuali, la guida autonoma, la robotica e la ricostruzione 3D. Tutti questi problemi sono stati originariamente affrontati con l'utilizzo di normali telecamere ma l'ambiguità intrinseca delle immagini bidimensionali ha portato allo sviluppo di tecnologie per sensori di profondità. La visione stereoscopica è stata la prima tecnologia a permettere di stimare la geometria tridimensionale della scena. Sensori a luce strutturata sono stati sviluppati per sfruttare gli stessi principi della visione stereoscopica ma risolvere alcuni problemi dei dispositivi passivi. Infine i sensori a tempo di volo cercano di risolvere lo stesso problema di stima della distanza utilizzando una differente tecnologia.

Questa tesi si focalizza nell'acquisizione di dati di profondità da diversi sensori e presenta tecniche per combinare efficacemente le informazioni dei diversi sistemi di acquisizione. Per prima cosa le tre principali tecnologie sviluppate per fornire una stima di profondità sono esaminate in dettaglio, presentando i principi di funzionamento e i problemi dei diversi sistemi. Successivamente è stato studiato l'utilizzo congiunto di sensori, fornendo delle soluzioni pratiche al problema della ricostruzione 3D e del riconoscimento dei gesti. I dati di un sistema stereoscopico e di un sensore a tempo di volo sono stati combinati per fornire una mappa di profondità più precisa. Per ognuno dei due sensori sono state sviluppate delle mappe di confidenza utilizzate per controllare la fusione delle mappe di profondità. La mancanza di collezioni con dati di diversi sensori è stato affrontato proponendo un sistema per la collezione di dati da diversi sensori e la generazione di mappe di profondità molto precise, oltre ad un sistema per la generazioni di dati sintetici per sistemi stereoscopici e sensori a tempo di volo. Per il problema del riconoscimento dei gesti è stato sviluppato un sistema per l'utilizzo congiunto di un sensore di profondità e un sensore Leap Motion, per migliorare le prestazioni dell'attività riconoscimento. Un insieme di descrittori ricavato dai due sistemi è stato utilizzato per la classificazione dei gesti con un sistema basato su Support Vector Machines e Random Forests.

Acknowledgements

I want to start by apologizing to all my friends for the little free time I had for them during these three very busy years, I missed all of you. It has been an intense journey that made me learn many new things, but also travel and meet many people. I want to thank my supervisor Pietro Zanuttigh who made me a better researcher and who gave me the freedom to work on different interesting projects.

Thanks to my friends in Padova, in particular Davide, Mattia and Giacomo because they are valuable friends without whom my time at the University would have not be the same. Marco and Anna Valeria for the great time we spent together every day at lunch and for the reciprocal encouragement in tough periods. Marta and Marzia for the hard time they gave me while living together but also for all the funny moments and the company. Guido Maria Cortelazzo for inspiring me with the effort and passion he put in writing the book and leading the research group for many years.

Thanks to everyone at Aquifi for the stimulating projects and for everything I learned there. Thanks to "Little Italy" for making me feel part of a family, and to Carlo and Pietro for all the crazy adventures.

Words are not enough to express how thankful I am to my parents, Mauro and Rosanna and to my sister Sara, for their constant presence and care they always reserve for me. Finally, I wish to express my sincere gratitude to Ilene, especially for her constant support and for nicely pushing me to improve every day.

Contents

Chapter 1

Introduction

Nowadays the two dimensional view of the world provided by standard cameras has been extended to three dimensions thanks to the introduction of depth cameras. These devices have expanded the possible applications usually provided by standard cameras, accurately recognizing objects, inferring shape and size of the environment and interacting with a virtual reality through gesture recognition. Applications include, but are not limited to, virtual and augmented reality, autonomous driving, security systems and robotics.

The first depth camera technology introduced in the market is the stereo vision system. Stereo vision just requires two standard cameras to generate a depth map of the scene framed by the two cameras. Despite its simplicity, stereo vision has several well known drawbacks, such as the poor performance in uniform regions. Structured light depth cameras were introduced to solve the problems of passive technologies. However, even if recent research in this field has greatly improved the quality of the estimated geometry, results are still not completely reliable and strongly depend on scene characteristics. The last family of depth cameras includes devices based on the the Time-of-Flight (ToF) technology. ToF depth cameras are able to estimate in real time the 3D geometry of a scene but they are also limited by a low spatial resolution and noisy measurements, especially for low reflective surfaces. ToF depth cameras are also affected by the multipath effect for which no definitive solutions have been proposed yet. Active depth cameras in general are able to provide a higher quality depth maps compared with passive devices at the cost of relying on an additional illuminator and particular infrared (IR) filters in the optics, that make active devices less reliable in outdoor scenes.

Since the characteristics of different depth cameras are somehow complementary, the problem of combining data from multiple sensors has attracted considerable interest. This problem has numerous applications, for example in the field of

autonomous driving the vision task is of fundamental importance and to provide reliable information data fusion is frequently used. Applications of depth data are not limited to 3D reconstruction. For example, virtual reality and other human-machine interaction schemes require reliable gesture recognition approaches to make humans able to interact with the virtual environment.

My research activity focused on the analysis of 3D data, including the acquisition and processing of data from different sensors, and some related applications. This thesis describes the technology behind current depth cameras, the 3D data processing to best combine data produced by multiple sensors and finally some applications where depth data provide significant contributions.

Chapter 2 reviews the operating principles of different depth camera families, including stereo vision systems, structured light and ToF depth cameras. To study the working principles of depth cameras I interned at Aquifi Inc, a startup located in Palo Alto (CA), USA. During my period there I participated to the design and development of a structured light camera. After a deep analysis on the available technologies, I contributed to the design and optimization of the IR pattern used in the illuminator of the structured light camera. I also developed a system to simulate the acquisition of the pattern from a stereo camera, according to the projection laws of the diffractive optical element (DOE) used in the illuminator, and the standard pin-hole model for cameras. Then, I contributed to the development of the pipeline to generate 3D data in real time from a pair of calibrated images. I also developed algorithms of image processing to be used both as pre-processing and post-processing of the depth map. Due to a non disclosure agreement this thesis does not contain the detailed description of the algorithms developed during the internship.

The fusion of depth data acquired from multiple sensors is described in Chapter 3, where depth data from multiple sensors are combined together to provide a higher quality depth map. The approach that we developed [73] uses the depth maps from a stereo system and a Time-of-Flight (ToF) camera for which the calibration is known, and a set of confidence measures that we estimate from the acquired data. The proposed approach extends a framework for cost aggregation called Local Consistency [74], originally proposed for stereo systems, to use the depth maps and confidence maps estimated. For the ToF sensor we developed a confidence measure that models the received signal and the geometry of the scene. Another contribution is the introduction of a new confidence metric for the stereo data. Typical confidence measures already available in the literature do not consider the effects of the global optimization performed by most of the best performing stereo algorithms. First we analyzed the properties of the cost functions of the correspondence problem

before and after the optimization. After the characterization of the behavior of such functions in different conditions we proposed different models that combine information of both the functions. One model in particular has been used for the fusion of data with Local Consistency, with results that outperform the state of the art if compared with traditional confidence measures.

Another problem for the data fusion from multiple sensors is the lack of datasets in the literature that includes calibrated images of different sensors with the related ground truth map. For this purpose we developed a system for the simultaneous acquisition of data from different sensors, including stereo, ToF and structured light depth cameras. For the acquisition of the ground truth we developed a system based on line laser that allows one to obtain a detailed depth map of the scene. We also developed a simulator of ToF and stereo systems that allows one to generate synthetic views of a given 3D model as if they were acquired from real cameras. Such a simulator also includes realistic models of the devices, allowing one to generate a big amount of realistic data. Chapter 4 presents the framework developed for the acquisition of the dataset with real cameras and the synthetic dataset.

For applications of depth data from multiple sensors I focused on two aspects of gesture recognition presented in Chapter 5. The first one is classification of different parts of the hand, while the second one involves the study of depth based descriptors for the task of hand gesture classification. Some algorithms for gesture recognition rely on palm detection as the first step, and for this task we proposed a tridimensional based method, that analyzes the structure of the point cloud acquired from a depth camera to classify fingers and palm [78]. For this approach we based our analysis on the different geometry of fingers and palm, proposing a density based clustering algorithm. This approach allowed us to correctly segment the fingers from the palm also in challenging situations including occlusions.

For the task of gesture classification presented in Chapter 5, we extended a method based on SVM, considering different descriptors both in the 2D and 3D domain. In particular, we developed descriptors that analyze the shape and contour of the hand [30, 29]. The joint usage of data from multiple sensors has been considered in a project for gesture recognition from a depth camera and a Leap Motion, a portable device that provides the 3D position of the hand's skeleton. This topic was new in the literature, therefore we had to run many preliminary experiments to assess the quality of the data and outline advantages and disadvantages of both sensors. The goal of this project was to combine data from a depth camera and a Leap Motion to provide a more robust estimate of the gesture performed by the user [70, 72]. Data from Leap Motion are very precise but in some configurations, that include occlusions and particular views, the errors

can be very high and some measurements can be missing. Depth sensors instead provide a higher number of 3D points but less accurate. A calibration of the two system is first presented to jointly use the data from the two sensors, then an SVM based approach with some descriptors typical of the two systems is proposed. Experimental results show that such a system improves the performance of the two systems considered independently.

The main topics faced during my research activity have been collected in a book published by Springer [118], in collaboration with other students and professors, on the technologies and working principles of depth sensors like ToF and structured light. It also includes an overview on the calibration of such devices and applications of depth cameras like gesture recognition, segmentation, 3D reconstruction and pose estimation.

Some of the material in this thesis has been already published in conference proceedings, journals and books, but some include works still under development.

Chapter 2

Depth acquisition systems

The acquisition of the geometric description of static or dynamic scenes has traditionally been a challenging task. The synopsis of distance measurement methods in Figure 2.1, derived from [8], offers a good framework to introduce different solutions proposed for the acquisition of depth data. Among all the possible methods that have been developed, in this thesis we will focus on the three reflective optical methods highlighted in Figure 2.1, classified into *passive* and *active*.

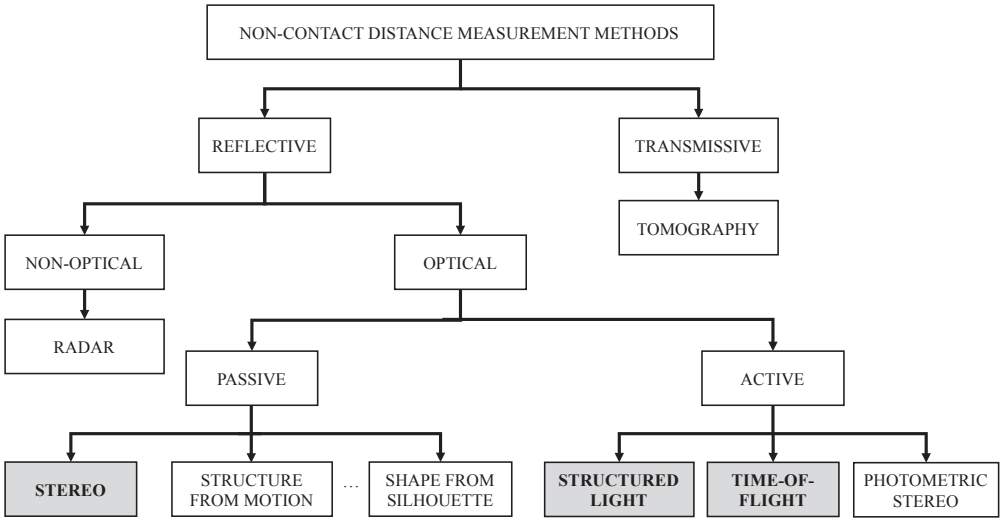


Figure 2.1: Taxonomy of distance measurement methods.

Passive range sensing refers to 3D distance measurement by way of radiation, typically in the visible spectrum already present in the scene. Stereo vision systems are a classical example of this family of methods. Active sensing refers instead to 3D distance measurement obtained by projecting some form of radiation in the scene. Two main families of devices belong to the active range sensing. The first family is based on the active triangulation working principle and the other is based on the

Time-of-Flight working principle. Cameras belonging to the active triangulation family are usually called structured light depth cameras, while cameras belonging to the second family are usually called matricial Time-of-Flight depth cameras, or simply ToF depth cameras. These three families of acquisition systems are generally referred to as depth cameras. The operation of stereo vision, structured light and ToF depth cameras involves a number of different concepts about imaging systems, ToF sensors and computer vision. These concepts are recalled in the next sections of this chapter.

2.1 Stereo vision systems

A stereo vision system, is a framework made by two regular cameras, that relies on the same principles of stereopsis adopted by humans, to provide an estimate of depth distribution of the scene acquired by the two cameras. Stereopsis, also known as binocular vision, is the process that allows our brain to extract information on the tridimensional structure from a pair of slightly different images of the same scene captured by the two eyes. The same concept can be applied to a pair of cameras framing the same scene, separated by a certain distance. It is common to call *reference camera* the left camera L, and *target camera* the right camera R. Each camera is assumed to be calibrated, with matrix of intrinsic parameters $\mathbf{K}_L$ and $\mathbf{K}_R$ for the L and R cameras respectively. Each camera has its own 3D reference system, also called camera coordinate system (CCS), and 2D reference systems, as shown in Figure 2.2. Namely, the L camera has CCS with coordinates (x_L, y_L, z_L), also called *L-3D reference system*, and a 2D reference system with coordinates (u_L, v_L). The R camera has CCS with coordinates (x_R, y_R, z_R), also called *R-3D reference system*, and a 2D reference system with coordinates (u_R, v_R). The two cameras may be different, but for the sake of clarity they are assumed to be identical, with $\mathbf{K} = \mathbf{K}_L = \mathbf{K}_R$, unless explicitly stated. A common convention is to consider the L-3D reference system as the reference system of the stereo vision system and to denote it as *S-3D reference system*

The 3D position of a point can be inferred by means of triangulation of correspondent points. We consider the case of a calibrated and rectified stereo vision system, i.e., a stereo vision system made by two identical standard cameras with coplanar and aligned imaging sensors and parallel optical axes as shown in Figure 2.3. Consider now a 3D point P with coordinates $\mathbf{P} = [x, y, z]$ and the projections $\mathbf{p}_L = [u_L, v_L]$ and $\mathbf{p}_R = [u_R, v_R]$ in the two camera image planes, left and right respectively. Triangulation is the process of determining the coordinates of P,

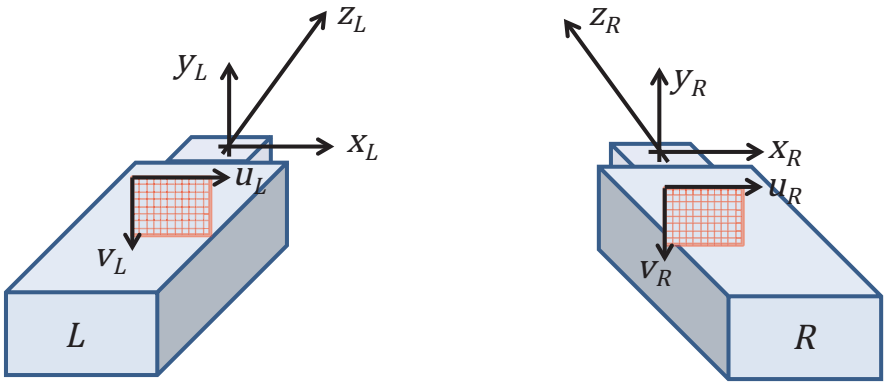


Figure 2.2: Stereo vision system coordinates and reference systems.

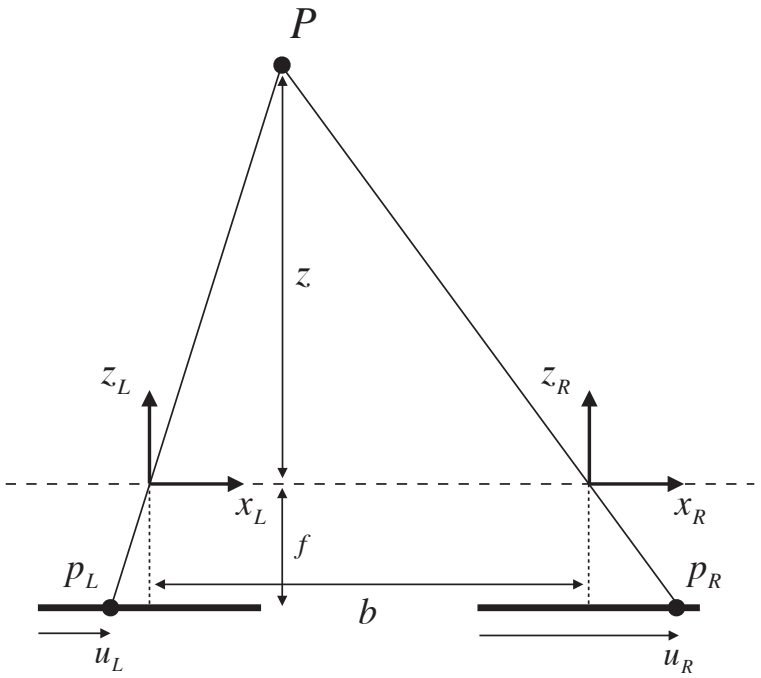


Figure 2.3: Triangulation with a rectified stereo system.

especially the depth coordinate z, given its projections $\mathbf{p}_L$ and $\mathbf{p}_R$.

In rectified stereo vision systems points $\mathbf{p}_L$ and $\mathbf{p}_R$ have the same vertical coordinates. Given the geometry depicted in Figure 2.3 and similar triangles properties, the following equations can be derived

$$\begin{cases} \dfrac{f}{z} = \dfrac{u_L - c_x}{x} \\ \dfrac{f}{z} = \dfrac{u_R - c_x}{x - b} \end{cases} \tag{2.1}$$

from which after some manipulation we obtain

$$z = \frac{b\ f}{u_L - u_R} = \frac{b\ f}{d} \tag{2.2}$$

In the previous equations, f is the focal length of the two cameras, b is the distance between the two optical centers, also known as *baseline* and $d = u_L - u_R$ is the so called *disparity* associated to point $\mathbf{p}_L$, i.e. the difference between x coordinate of the two corresponding points in left and right image planes. Equation (2.2) shows how it is possible to retrieve the third component z when disparity and geometry of the system are known.

From Equation (2.2), given the calibration parameters of the stereo vision system one can also compute the depth resolution Δz as reported in [110]

$$\Delta z = \frac{z^2}{bf}\Delta d \tag{2.3}$$

where Δd is the disparity resolution. Equation (2.3) shows that the depth resolution is quadratically dependent on the depth of the measured object (i.e., its z coordinate). Disparity resolution Δd can be 1 in the case of pixel resolution or less than 1 in the case of sub-pixel resolution. The relationship between depth and disparity of Equation (2.2) and the theoretic depth resolution computed with Equation (2.3) are important quantities to consider in the design process of a stereo rig.

While f and b can be estimated by the calibration of the system, the disparity d requires to find corresponding points, also known as conjugate points, in the two images. Given a point $\mathbf{p}_L$ in the left image, the correspondent point $\mathbf{p}_R$ in the right image has to be found. We know that the two images are not so different since they represent the same scene seen from slightly different point of views, however the correspondent point could be at any pixel. A search of that point in the entire image requires many operations, also because the most common similarity criterions require to do operations in a window for every pixel. Fortunately, the search domain can be limited to one dimension thanks to the epipolar constraint. A geometrical analysis shows that the conjugate point of $\mathbf{p}_L$ in the second image, must lie in a straight line called epipolar line of $\mathbf{p}_L$. In a more realistic scenario the two cameras are not perfectly aligned, however, after the image distortion due to the lens has been compensated, it is always possible to rectify the two acquired images with a linear transformation to simplify the task of correspondence selection.

2.1.1 The correspondence problem

The triangulation procedure assumes the availability of a pair of conjugate points p_L and p_R. This represents a delicate and tricky assumption for the triangulation procedure, first of all because such a pair may not exist due to occlusions. Even if it exists, it may not be straightforward to find it. Indeed, the correspondence problem, i.e. the detection of conjugate points between the stereo image pairs, is one of the major challenges of stereo vision algorithms. The methods proposed for this task can be classified according to various criteria.

A first distinction concerns dense and sparse stereo algorithms. The former, representing current trends [97], are methods aimed at finding a conjugate point for every pixel of the left image, of course within the limits imposed by occlusions. The latter are methods which do not attempt to find a conjugate for every pixels.

A second distinction concerns *local* and *global* approaches. Local methods consider only local similarity measures between the region surrounding p_L and regions of similar shape around all the candidate conjugate points p_R of the same row. The selected conjugate point is the one which maximizes the similarity measure, a method typically called winner takes all (WTA) strategy. Conversely, global methods do not consider each couple of points on their own, but instead estimate all of the disparity values at once, exploiting global optimization schemes. Global methods based on Bayesian formulations are currently receiving great attention in dense stereo. Such techniques generally model the scene as a Markov Random Field (MRF), and include within a unique framework clues coming from local comparisons between the two images and scene depth smoothness constraints. Global stereo vision algorithms typically estimate the disparity image by minimizing a cost function made by a *data term* representing the cost of local matches, similar to the computation of local algorithms (e.g., covariance) and a *smoothness term* defining the smoothness level of the disparity image by explicitly or implicitly accounting for discontinuities [106].

There is a third class of stereo matching algorithms that lies in between local and global approaches, the so called *semi-global* approaches. The Semi-Global Matching (SGM) approach proposed by Hirschmuller [46] is an example of algorithms belonging with this class. It explicitly models the 3D structure of the scene by means of a point-wise matching cost and a smoothness term. Several 1D energy functions computed along different paths are independently and efficiently minimized, and their costs are summed up. For each point, the disparity corresponding to the minimum aggregated cost is selected.

The algorithm is briefly described here since it will be used in the data fusion

of Chapter 3. The matching cost in the original implementation is computed using a mutual information based approach for compensating radiometric differences of input images. Other implementations instead, use faster cost calculation techniques, as the Birchfield and Tomasi [7] metric. Another valid alternative is the census cost function, that gives the best overall results for different datasets and is rather robust under adverse lighting conditions. The *local* cost $C_L(\mathbf{p}_L, d)$ for pixel $\mathbf{p}_L$ is defined for each disparity hypothesis d.

Cost aggregation is the real strength of this approach. Pixelwise cost $C_L(\mathbf{p}_L, d)$ is generally prone to wrong matches, therefore an additional constraint is added to the energy function to support smoothness and penalize changes of neighboring disparities. By assuming that the observed surfaces are smooth, disparity shifts can be penalized by setting an additional cost of assigning a depth to a pixel if it does not agree with its neighbors. This means that when the algorithm tries to estimate the disparity of a pixel having several possible matches, it will probably choose the match which agrees more with the depth estimates of the neighboring pixels. Instead of solving a 2D global optimization of the energy function, multiple 1D optimization can be performed efficiently in polynomial time along 8 or 16 paths. The final cost $C_G(\mathbf{p}_L, d)$ is defined as the summation of the energy function along all the paths and the final disparity for each pixel is computed as the argument that minimizes the *global* cost $C_G(\mathbf{p}_L, d)$.

2.1.2 Practical issues

The detection of pairs of conjugate pixels is the most complex part of the depth map estimation. Correspondence problem relies on the main assumption that left and right images are not too different from each other and they have to exhibit a certain level of disparity while framing the same scene. Many problems afflict correspondence detection, some are related to the geometry of the system and some to the scene itself. The major issues related to correspondence selection are described next.

Occlusions and discontinuities Due to discontinuities of the surfaces and particular displacement of the objects in the scene, some points in one image may not be visible in the other image. For those points that do not have the relative conjugate, disparity has no reason and meaning to be defined. This is maybe the most known problem in stereo vision and can be observed by looking at the edge of an object, the background close to the edge is visible only from one of the two cameras. There exists a common procedure to detect occlusions, called Left-Right consistency check, but no exact solutions exist

to retrieve the disparity of such areas. A similar problem can be experienced because of perspective projection. An object may assume different shapes in the two images and some detail may be visible from one view but not in the other.

Edge fattening Most of the stereo matching algorithms make use of appropriate support windows surrounding the considered point, to find its corresponding point on the other image. The use of a window instead of matching point to point, make the matching problem more robust to noise in the images. The main assumption when a window is considered is that all the points inside the window have the same disparity. This is necessary otherwise the matching window in the other image would contain points from a different portion of the scene. An optimal support window should be large enough to capture sufficient intensity variation for handling textureless regions. At the same time, the window should be small enough not to include pixels with different disparity. A small window leads to noisy disparity maps but larger windows produce fatter edges near disparity discontinuities. In these regions indeed, only the points in the foreground part of the scene match in the two images. Points in the background instead have different disparities and so pixels in the same relative location inside the window will have different intensity values, since they correspond to different points of the scene. The effect is that the same disparity value of the foreground points is associated to points in the background next to depth discontinuities.

Radiometric distortion and noise For materials not perfectly lambertian, the observed point can be different in the two images. Moreover due to the always present noise, color and intensity of the two acquired scenes can be different, increasing the complexity in the correspondence search.

Specular surfaces Similar to the previous issue, glossy materials may reflect external lights directly into the camera. Due to different viewpoint of the two cameras, a region in one image may be visible and the correspondent in the other one may be overexposed. If the illumination of the scene does not come from a direct spot light, the likelihood of having such overexposed regions decreases.

Perspective foreshortening Because each stereo camera has a slightly different view, the image of the surface is more compressed and occupies a smaller area in one view. The more an object is horizontally slanted, the more pronounced this effect is. Foreshortening causes problems especially to methods using

fixed-size windows to aggregate costs, because they tacitly assume that objects occupy the same extents in both images.

Transparent objects Objects with a certain transparency cause an intrinsic ambiguity. Background that is visible through these objects actually would be occluded by the object itself. This inevitably introduces uncertainty that influences the results of both local and global methods.

Uniform regions Poorly textured areas still continue to plague stereo matching systems. The ability to detect similar regions assumes that correlation or other methods are able to detect a peak of some functions. If a uniform region sufficiently large is considered, for example a white wall, neither local or global methods can overcome this issue with sufficient certainty. Although this is a common problem in all stereo matching methods, techniques that propagate disparity cues are likely to assign a valid disparity also to these regions.

Repetitive pattern Correspondence of regions without texture is difficult to find, and so is the case of highly textured regions with periodic patterns. Without a global knowledge of the scene, it is impossible to distinguish between the correct correspondence or an erroneous translated version. A classic example is provided by framing a checkerboard, in this case it is easily deductible that the shape of the cost function for the points inside the checkerboard presents a certain number of peaks. Also in this case, the ambiguity can be reduced with the aid of global methods.

All these physical issues account for increasing the probability of false correspondences. Some of them can be handled by means of image processing or other techniques, but others, like occlusions, are physically impossible to manage. Although specific stereo algorithms may have a considerable impact on the solution of the correspondence problem, the ultimate quality of 3D stereo reconstruction inevitably also depends on scene characteristics. This can be readily realized considering the case of a scene without geometric or color features, such as a straight wall of uniform color. The stereo images of such a scene will be uniform, and since no corresponding points can be detected from them, no depth information about the scene can be obtained by triangulation.

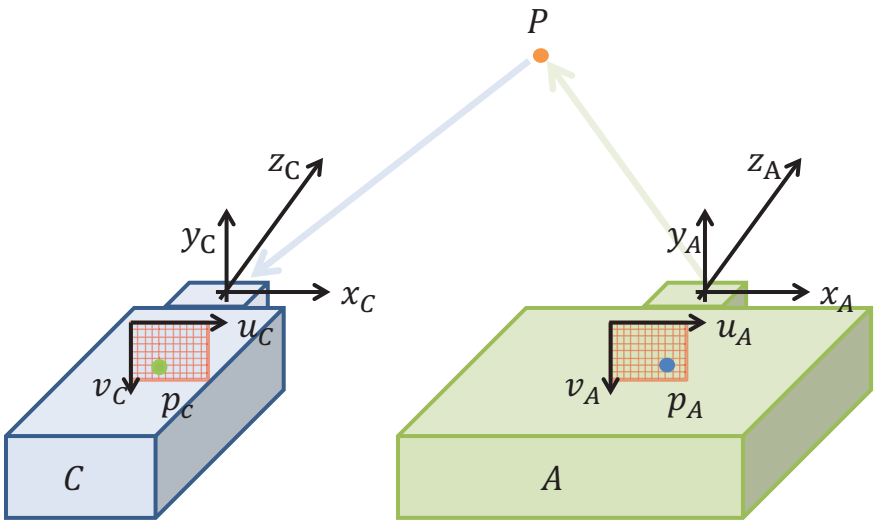


Figure 2.4: Active triangulation by a system made of a camera C and a light projector A.

2.2 Structured light depth cameras

As previously noted, the reliability of the correspondences remains a critical step of computational stereopsis. Structured light depth camera systems address this issue and provide effective solutions. In triangulation or computational stereopsis procedures, the main concept at the basis of triangulation is of geometric nature and is shown in the triangle arrangement between rays Pp_L, Pp_R and p_Lp_R in Figure 2.3. Since from a perspective geometry standpoint [43], image points are equivalent to rays exiting a center of projection, any device capable of projecting rays between its center of projection and the scene points is functionally equivalent to a standard camera. Therefore, light projectors or illuminator devices in which each pixel p_A illuminates a scene point P by its specific light value thus creating a spatial pattern, can be modeled as active pin-hole systems where light rays connecting the center of projection and the scene point P through pixel p_A (as shown in Figure 2.4) are emitted, rather than received as in standard cameras. Triangulation also remains applicable if one of the two cameras of the stereo system of Figure 2.2, is replaced by a projector as in Figure 2.4, granted by triangle arrangement Pp_C, Pp_A and p_Cp_A. The active, rather than passive, nature of ray Pp_A does not affect the reasoning behind the demonstration of triangulation. Such an arrangement made by a camera C and a projector A as shown in Figure 2.4, is called structured light system.

Structured light systems have the same structural geometry of standard passive stereo systems, thus calibration and rectification procedures [108] can also be applied to them to simplify the depth estimation process. In the case of a rectified system,

pixel p_A with coordinates $\mathbf{p}_A = [u_A, v_A]^T$ of the projected pattern casts a ray that intersects the acquired scene at a certain 3D location $\mathbf{P}_C = [x_C, y_C, z_C]^T$. If both the projective distortion of A and C are compensated, p_C has coordinates

$$\mathbf{p}_C = \begin{bmatrix} u_C = u_A + d \\ v_C = v_A \end{bmatrix} \tag{2.4}$$

with disparity value $d = u_C - u_A$, defined exactly as in the standard passive stereo system, apart from the different notation adopted for the coordinate system.

Since we have established that all triangulation expressions derived for a 2-camera stereo system also apply to structured light systems made by an illuminator and a single camera, let us now consider the advantages of the latter with respect to the former. As previously noted, in passive stereo systems made by a pair of cameras, the possibility of identifying conjugate points depends completely on the visual characteristics of the scene. In particular, in the case of a feature-less scene, like a flat wall of uniform color, a stereo system could not establish any point correspondence between the image pair and could not give any depth information about the scene. On the contrary, in the case of a structured light system the light pattern pixel p_A of the projector "colors" the scene point P to which it projects with its radiant power. Assuming a straight wall without occlusions, the pixel p_C of the camera C where P is projected, receives from P the "color" of p_A and becomes recognizable among its neighboring pixels. This enables the possibility of establishing a correspondence between conjugate points p_A and p_C. Structured light systems can therefore also provide depth information in scenes without geometry and color features where standard stereo systems fail to give any depth data.

It is also clear that a system with two cameras $C1$ and $C2$ and a projector A, as shown in Figure 2.5, is a variation of a structured light system by which the coordinates of point P in principle can be obtained by any of the types of triangulation seen so far, or by a combination of them. Indeed, P can be computed by triangulation upon knowledge of either conjugate points p_{C1} and p_{C2}, points p_{C1} and p_A, or points p_{C2} and p_A.

It is possible to demonstrate the complete functional equivalence between the various structured light systems configurations, namely the single camera, the two cameras and the so called space-time stereo systems [24, 59]. The generalization of this idea leads to the so called *camera virtualization*, i.e., a procedure hinted in [24], by which a structured light depth camera made by a single camera and an illuminator operates equivalently to a structured light depth camera made by two rectified cameras and an illuminator. In the case of a single camera, the system is equivalent

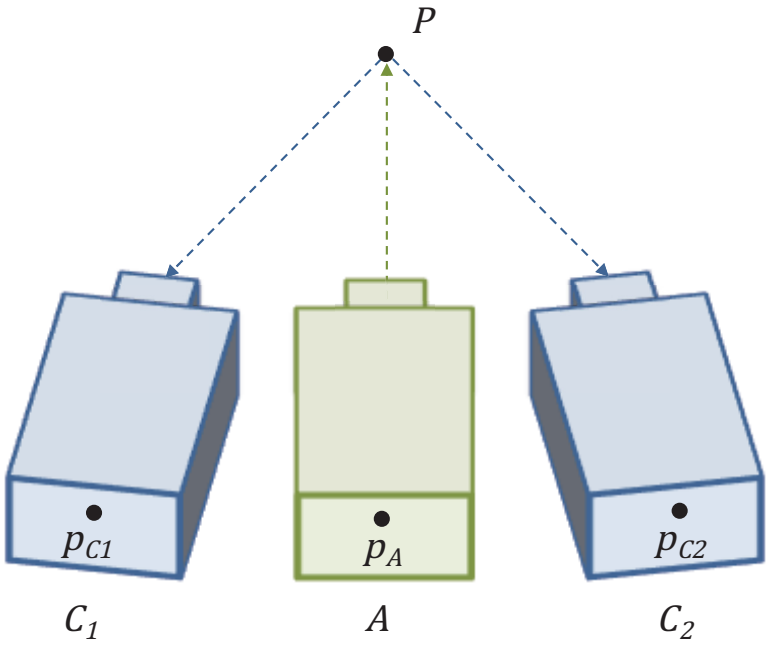


Figure 2.5: Structured light system with two cameras and a projector

to a stereo vision system with a real camera and a "virtual" camera co-positioned with the projector. Camera virtualization plays a fundamental conceptual role since it decouples the structured light system geometry from the algorithms used on them: in other words, standard stereo algorithms can be applied to structured light systems whether they have one or two cameras, unifying algorithmic methods for passive and active methods independently from the geometric characteristics of the latter.

2.2.1 Illuminator design approaches

The objective of structured light systems is to simplify the correspondence problem through projecting effective patterns by the illuminator A. This section reviews current pattern design methodologies. The characteristics of the projected patterns are fundamental for the solution of the correspondence problem and for the overall system performance. In addition, the specific design of the illuminator as well as its implementation are at the core of all structured light depth cameras. The illuminators mainly belong to two families, namely, *static* illuminators, which project a static pattern/texture into the scene, and *dynamic* illuminators, which project a pattern/texture that varies in time. In general, active techniques are slower and more expensive than passive methods but much more accurate and robust. The structure of the projected pattern can be either in the form of a pattern characterized by dots (e.g., Primesense cameras [90]), in the form of a continuous

texture (e.g., Intel RealSense R200 [51]) or in the form of a striped pattern (e.g., Intel RealSense F200 [50]).

A code word alphabet can be implemented by a light projector considering that it can produce n_P different illumination values called *pattern primitives* (e.g., $n_P = 2$ for a binary black-and-white projector, $n_P = 2^8$ for a 8-bit gray-scale projector, and $n_P = 2^{24}$ for a RGB projector with 8-bit color channels). The local distribution of a pattern for a pixel p_A is given by the illumination values of the pixels in a window around p_A. If the window has n_W pixels, there are $n_P^{n_W}$ possible pattern configurations on it. From the set of all possible configurations, N configurations need to be chosen as code words. What is projected to the scene and acquired by C is the pattern resulting from the code words relative to all the pixels of the projected pattern. Let us assume that the projected pattern has $N_R^A \times N_C^A$ pixels p_A^i, $i = 1, \ldots, N_R^A \times N_C^A$ where N_R^A and N_C^A are the number of rows and columns of the projected pattern, respectively.

The concept of pattern uniqueness is an appropriate starting point to introduce the various approaches for designing illuminator patterns. Consider an ideal system in which images I_C and $I_{C'}$ are acquired by a pair of rectified cameras C and C' (whether C' is real or virtual is immaterial for the subsequent discussion) and assume the scene to be a fronto-parallel plane corresponding to disparity 0 at infinity and infinite reflectivity. Since the cameras are rectified, points of I_C and $I_{C'}$ corresponding to the same 3D point P, are characterized by coordinates with the same v-component and u-components differing by disparity d: $\mathbf{p} = [u, v]^T$, $\mathbf{p}' = [u', v']^T = [u - d, v]^T$. The correspondences matching process searches the conjugate of each pixel p in I_C, by allowing d to vary in the range $[d_{min}, d_{max}]$ and by selecting the value $\hat{d}$ for which the local configuration of I_C around $\mathbf{p}$ is most similar to the local configuration of $I_{C'}$ around $\mathbf{p} - [d, 0]^T$ according to a suitable metric.

Images I_C and $I_{C'}$ can carry multiple information channels, for instance encoding data at different color wavelengths (e.g., R, G, B channels) or at multiple timestamps $t = 1, \ldots, N$ with N being the timestamp of the most recent frame acquired by cameras C and C'. The local configuration in which the images are compared is a cuboidal window $W(\mathbf{p})$ made by juxtaposing windows centered at $\mathbf{p}$ in the different channels. If there is only one channel (with respect to time), the system is characterized by an instantaneous behavior and is called a *spatial stereo system*, according to [24]. On the contrary, if the matching window is characterized by a single-pixel configuration in the image (e.g., the window is only made by the pixel with coordinate $\mathbf{p}$) and by multiple timestamps, the system is called a *temporal stereo* system. If the matching window has both a spatial and temporal component,

the system is called *spacetime stereo*. A standard metric to compute the local similarity between I_C in the window $W(\mathbf{p})$ and $I_{C'}$ in the window $W(\mathbf{p}')$ is the Sum of Absolute Differences (SAD) of the respective elements in the two windows, defined as

$$SAD[I_C(W(\mathbf{p})), I_{C'}(W(\mathbf{p}'))] \triangleq \sum_{\mathbf{q}\in W(\mathbf{p}),\mathbf{q}'\in W(\mathbf{p}')} |I_C(\mathbf{q}) - I_{C'}(\mathbf{q}')|. \tag{2.5}$$

rewritten for simplicity just as $SAD(\mathbf{p}, d)$. For each pixel p one selects the disparity that minimizes the local similarity as $\hat{d}(\mathbf{p}) = \operatorname{argmin} SAD(\mathbf{p}, d)$. A pattern is said to be unique if in an ideal system, i.e., a system without any deviation from theoretical behavior, for each pixel p in the lattice of I_C, the value of the SAD metric of the actual estimated disparity d^* coincides with minimum $\hat{d}(\mathbf{p}) = \operatorname{argmin} SAD(\mathbf{p}, d)$, which is unique. The uniqueness U of a pattern is defined as

$$U \triangleq \min_{p\in\Lambda_C} U(\mathbf{p}) \tag{2.6}$$

where $U(\mathbf{p})$ is computed as the second argmin of the SAD metric, excluding the first argmin $\hat{d}(\mathbf{p})$ and the values within one disparity value from it, i.e.,

$$d \in \{d_{min}, \ldots, d_{max}\} \setminus \{\hat{d}(\mathbf{p}) - 1, \hat{d}(\mathbf{p}), \hat{d}(\mathbf{p}) + 1\}. \tag{2.7}$$

For each pixel in the image I_C the uniqueness map $U(\mathbf{p})$ is computed as the cost of the non-correct match that gives the minimum matching error. The higher such cost is, the more robust the pattern is against noise and and other practical issues. The minimum uniqueness value across the entire pattern is selected to obtain a single uniqueness value for the entire pattern.

This concept of uniqueness is a function of the number of color channels, the range of values in the image representation, and the shape of the matching window, which may have both a spatial and temporal component. Following the framework of [96], different choices of these quantities lead to different ways to encode the information used for correspondences estimation, typically within the following four signal multiplexing families:

- wavelength multiplexing;
- range multiplexing;
- temporal multiplexing;
- spatial multiplexing.

Each multiplexing technique performs some kind of sampling in the information dimension typical of the technique, limiting the reconstruction capability in the specific dimension [118].

2.2.2 One and two cameras setups

Although the presence of a second physical camera may seem redundant, given the complete operational equivalence between single camera and double camera systems, in practice it leads to several system design advantages. The usage of two cameras leads to better performance because it simplifies the handling of many manufacturing imperfection and practical issues, such as the distortion of the acquired pattern with respect to the projected one due to camera and projector imperfections and to their relative alignment. Furthermore, to benefit from the virtual camera methodology, the projected pattern should maintain the same geometric configuration at all times. This requirement can be demanding for camera systems with an illuminator based on laser technology, because the projected pattern tends to vary with the temperature of the projector. For this reason, an active cooling system is used in the Primesense single camera system design, while it is unnecessary in the two cameras Intel RealSense R200.

Another fundamental weakness of single camera systems is that any ambient illumination at acquisition time leads to a difference between the appearance of the acquired representation and that of the reference representation. This effect is most evident in outdoor scenarios where the sunlight interferes with the pattern. To cope with the mentioned illumination issues, single camera structured light systems adopt a notch optical filter on the camera lenses with a band-pass bandwidth tightly matched to that of the projected pattern. Moreover, in the case of extremely high external illumination in the projector's range of wavelengths, a double camera structured light depth camera can be used as a standard stereo system, either by neglecting or switching off the contribution of the active illuminator A.

The difference between one and two cameras can be exemplified by the following simulation with a test scene made by a flat wall textured by an image, e.g., the standard "Cameraman" of Figure 2.6. This scene offers a straightforward depth ground truth which is a constant value everywhere if the structured light system is positioned in a fronto-parallel situation with respect to the wall (i.e., if the optical axis of the rectified system cameras and projector are assumed orthogonal to the wall). With respect to the above scene, let us computationally simulate a structured light system projecting the Primesense pattern with a single acquisition camera, like in commercial products, and a structured light system projecting the Primesense

pattern but carrying two acquisition cameras instead of just one. For simplicity we will call S1 the former and S2 the latter.

As a first approximation, the scene brightness can be considered proportional to the reflectance and illumination made by a uniform component (background illumination) and by a component due to the Primesense pattern. In the case of S1, to mimic camera virtualization we consider only one acquisition of a shifted version of "Cameraman", and compare it with respect to the actually projected pattern. In S2 to simulate the acquisition from two cameras we consider two acquisitions of a shifted version of "Cameraman". The acquisitions with S1 and S2 are repeated using versions of the "Cameraman" images corrupted by independent additive Gaussian noise with different standard deviations.

Determining which of the two systems performs a better disparity estimation can be easily ascertained from the percentage of non constant, i.e., wrong depth values (in this case produced by a block-matching stereo algorithm with window size 9×9) as a function of the independent additive Gaussian camera noise, as shown in Figure 2.6. The performance of the depth estimation procedure of S1 (red) is worse than the one of S2 (blue), especially for typical camera noise values (black line).

Performance of system S1 in Figure 2.6 has an interesting behavior as the image noise increases. Let us recall that with S1 the disparity map is estimated by comparing the image of the noisy scene acquired by the camera, with the image of the pattern stored in the camera. The intensity of the acquired image can be divided into two components: the projected pattern and the texture already present in the scene. When the level of noise is low, the component due to the texture in the scene has more impact in the process of matching windows. Indeed, a window in the image storing the reference pattern contains only the component related to the pattern itself, while a window in the acquired image contains also the texture of the scene. When the noise increases, the component due to the texture in the scene becomes less strong, as the noise corrupts uniformly the image, and so the number of wrong disparities decreases. Although counterintuitive, the noise makes the underneath texture look more uniform, not corrupting much the projected pattern. When the noise increases more, the uniqueness of the pattern decreases and so the number of wrong disparities increases again. For system S2 instead the behavior is the same as the one of passive stereo, the percentage of wrong disparities increases with image noise. For system S2 the texture of the scene helps the selection of matching points in stereo algorithms, since the matching windows are sought in images of the same scene.

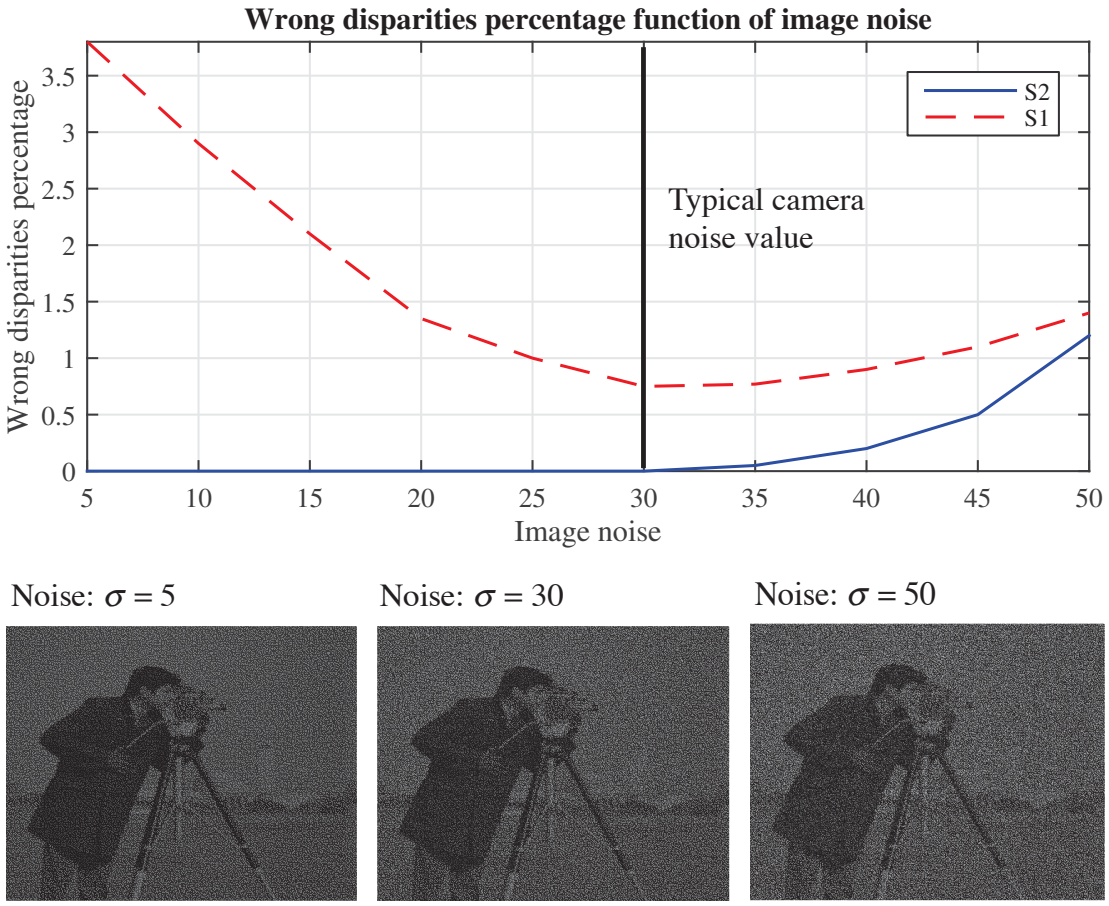


Figure 2.6: Simulation of the performance of a single camera structured light system projecting the Primesense pattern (S1) and of a double-camera structured light system projecting the Primesense pattern (S2) for a flat scene textured by the "Cameraman" image at various noise levels.

2.2.3 Structured light systems non-idealities

Structured light depth cameras are affected by a number of imperfections, independent from the actual implementation. Some of these issues are related to fundamental properties of optical and imaging systems, e.g., camera and projector thermal noise. A list of the most important issues is presented next.

1. *Perspective distortion.* Since the scene points may have different depth values z, neighboring pixels of the projected pattern may not be mapped to neighboring pixels of I_C. In this case the local distribution of the acquired pattern becomes a distorted version of the relative local distribution of the projected pattern (see the first row of Figure 2.7).

2. *Color or gray-level distortion due to scene color distribution and reflectivity properties of the acquired objects.* The projected pattern undergoes reflection and absorption by scene surfaces. The ratio between incident and reflected radiant power is given by the scene reflectance, generally related to the scene color distribution. In the common case of IR projectors, the appearance of the pixel p_C on the camera C depends on the reflectance of the scene surface at the IR frequency used by the projector. For instance, a high intensity pixel of the projected pattern at p_A may undergo strong absorption because of the low reflectance value of the scene point to which it is projected, and the values of its conjugate pixel p_C on I_C may consequently appear much darker. This is an extremely important issue, since it might completely distort the projected code words. The second row of Figure 2.7 shows how the radiometric power of the projected pattern may be reflected by surfaces of different color.

3. *External illumination.* The color acquired by the camera C depends on the light falling on the scene's surfaces, which is the sum of the projected pattern and of scene illumination, i.e., sunlight, artificial light sources, etc. This second contribution with respect to code word detection acts as a noise source added to the information signal of the projected light (see third row of Figure 2.7).

4. *Occlusions.* Because of occlusions, not all the pattern pixels are projected to 3D points seen by camera C. Depending on the 3D scene geometry, there may not be a one-to-one association between the pattern pixels p_A and the pixels of the acquired image I_C. Therefore, it is important to correctly identify the pixels of I_C that do not have a conjugate point in the pattern, to discard erroneous correspondences (see fourth row of Figure 2.7).

5. *Projector and camera non-idealities.* Both projector and camera are not ideal imaging systems. In particular, they generally do not behave linearly with respect to the projected and the acquired colors or gray-levels.

6. *Projector and camera noise.* The presence of random noise in the projection and acquisition processes is typically modeled as Gaussian additive noise in the acquired image or images.

From the list of imperfections just presented, one can notice that some of the problems corresponds to the practical issues of passive stereo systems presented in Section 2.1.2. Occlusions and perspective distortion, typical of stereo systems remain a problem also for structured light depth cameras.

2.2.4 Comparison of structured light depth cameras

After this introduction of theoretical and practical facts on structured light depth cameras, we now review the actual implementations of the presented design concepts by the most diffused structured light depth cameras in the market, namely, the Primesense camera, used in the Kinect$^{\text{TM}}$ v1, the Intel RealSense F200, and the Intel RealSense R200.

The Primesense camera (Kinect$^{\text{TM}}$ v1)

The Primesense camera, known to be used in the Kinect$^{\text{TM}}$ v1, is a less compact and more powerful system not suited for integration into mobile devices or computers when compared to the Intel RealSense F200 and R200[1]. As shown in Figure 2.8, the Primesense system generally comes with a color camera and a structured light depth camera made by an IR camera C and an IR projector A. While the IR camera of the Primesense system is a high-resolution sensor with 1280×1024 pixels, the depth-map produced by the structured light depth camera is 640×480. In spite of the nominal working depth range being $800 - 3500$ [mm], the camera produces reliable data up to 5000 [mm] and in some cases even at greater distances. The temporal resolution is up to 60 [Hz]. The resolution downscaling not only reduces the sensor acquisition noise by aggregating more pixels, but also improves the effective spatial resolution of the estimated disparity map. The horizontal Field-of-View (FoV) of the Primesense structured light depth camera is approximately 58° and the vertical FoV is 44°, with a focal length in pixels of approximately 600 [pxl] The presence of

[1]For completeness, one should recall that the design of the Primesense Capri targeted integration into mobile devices and computers, but it never reached production. This section focuses on the Primesense Carmine, the only product which was commercialized.

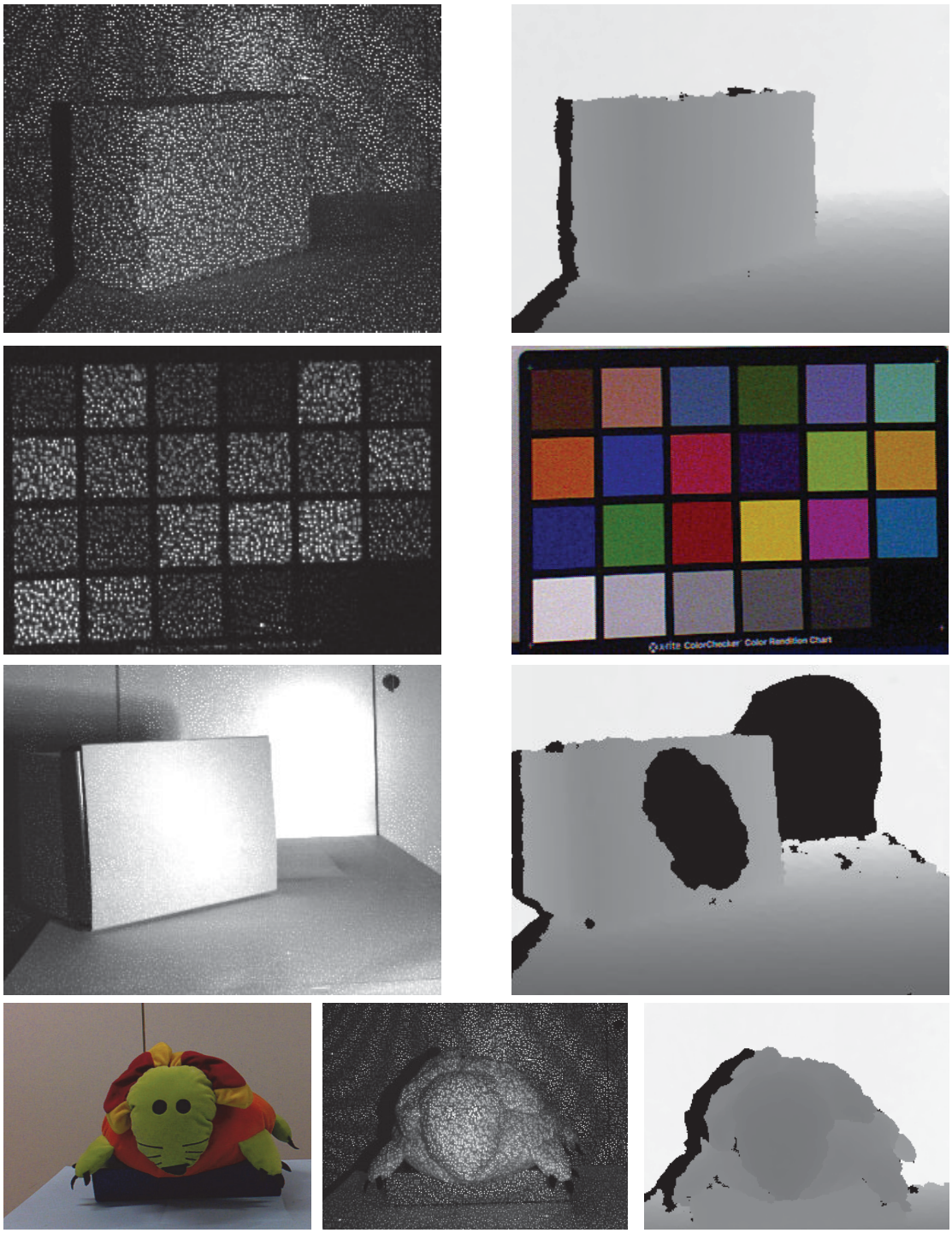


Figure 2.7: Examples of different artifacts affecting the projected patter. In the depth maps, black pixels correspond to locations without a valid depth measurement. *First row*: projection of the IR pattern on a slanted surface and corresponding depth map. *Second row*: Primesense pattern projected on a color checker and corresponding color image. *Third row*: a strong external illumination affects the acquired scene. *Fourth row*: the occluded area behind the stuffed toy is visible from the camera but not from the projector's viewpoint, consequently, the depth of this region cannot be computed.

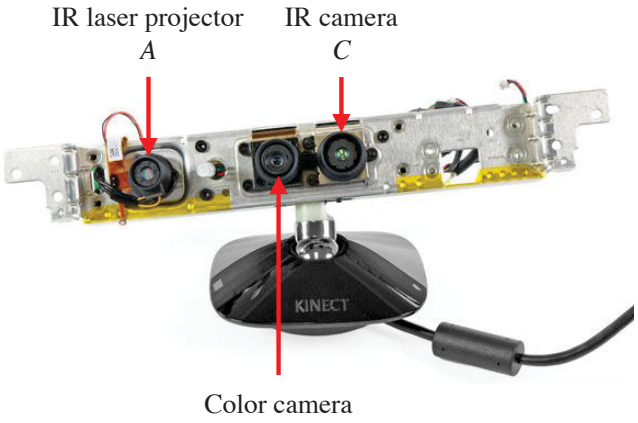


Figure 2.8: Primesense system components: color camera and depth camera made by an IR camera C and an IR projector A.

a high resolution IR camera in the Primesense structured light depth camera gives better performance with respect to the Intel RealSense F200 and R200 in terms of range, spatial resolution, noise, and robustness against external illumination.

The baseline between the IR camera C and the IR projector A is approximately 75 [mm]. Figure 2.9 shows the depth resolution of the Primesense depth camera, without sub-pixel interpolation and also with an estimated sub-pixel interpolation of 1/8, according to [58], as a function of the measured depth, according to (2.3) given the baseline and the focal length in pixels[2].

The projector is the most interesting component: it is a static projector that produces a pattern made by collimated dots, as shown in Figure 2.10. The collimated dots pattern appears to be subdivided into 3×3 tiles characterized by the same projected pattern up to holographic distortion. Collimated dots favor long-distance performance. Each tile of the pattern is characterized by a very bright dot at its center, usually called 0-th order, which is an artifact of the collimated laser going through a diffractive optical element.

The pattern of the Primesense depth camera has been thoroughly reverse engineered [58]. A summary of the major findings is reported next. A binary representation of the projected pattern is shown by Figure 2.11. Each one of the 3×3 tiles is made by 211×165 holographic orders (equivalent in diffractive optics to the concept of pixels in standard DLP projectors), hence the overall tiled pattern is made by $633 \times 495 = 313335$ holographic orders. For each tile only 3861 of these orders are lit (bright spots), for a total of 34749 lit orders in the tiled pattern.

[2]Even though depth resolution with practical sub-pixel interpolation is reported only for the Primesense structured light depth camera, it is expected to be also present in the Intel RealSense F200 and R200 structured light depth cameras. The practical sub-pixel interpolation value is theoretically better for the Primesense structured light depth camera than for the Intel RealSense F200 and R200 because of the higher resolution of its IR camera.

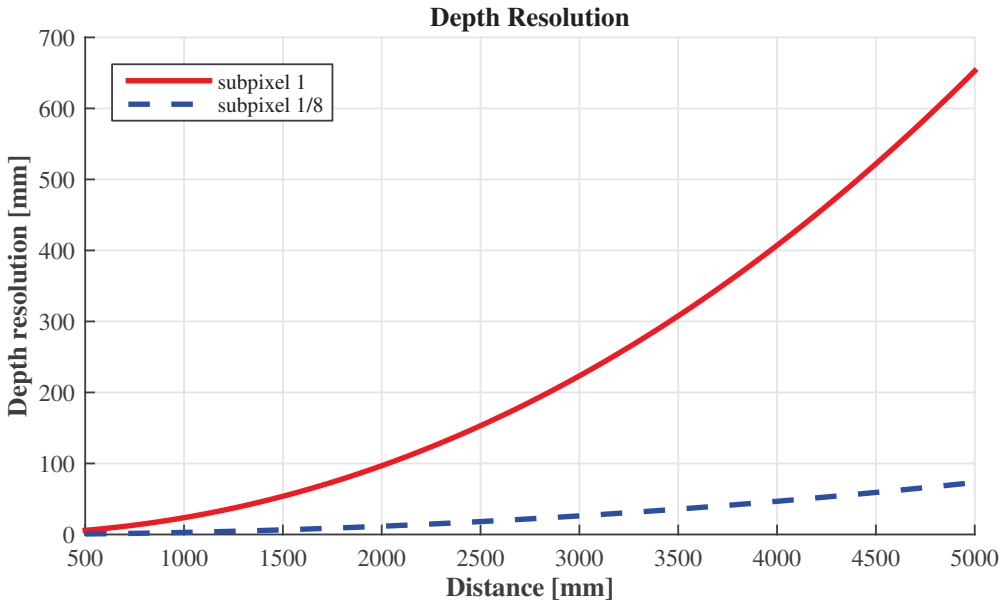


Figure 2.9: Primesense depth resolution without sub-pixel interpolation and with 1/8 sub-pixel interpolation.

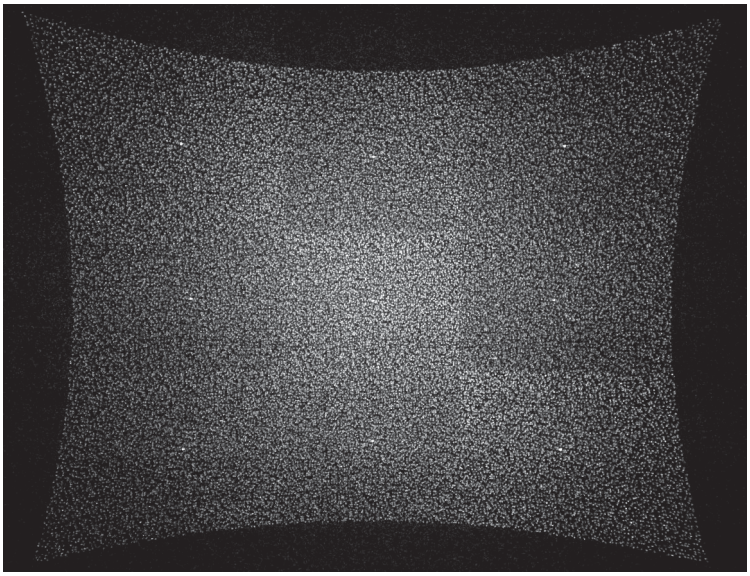


Figure 2.10: Pattern projected by the Primesense illuminator and acquired by a high-resolution camera.

Figure 2.11: Binary pattern projected by the Primesense camera reverse engineered by [58]. In this representation, there is a single white pixel for each dot of the projected pattern.

Therefore, on average, there is approximately one lit order for each 3×3 window and approximately 9 of them in a 9×9 window.

The uniqueness of the Primesense pattern can be computed according to (2.6). We recall that it is possible to compute a uniqueness value for each pixel and that the overall uniqueness is the minimum of such uniqueness values. The plot of the minimum uniqueness in the pattern, i.e., what has been defined as pattern uniqueness in (2.6), and of the average uniqueness are shown in Figure 2.12, together with the uniqueness map that can be computed pixel-by-pixel for a squared matching window of size 9×9. This figure shows how the Primesense pattern is a "unique pattern" if one uses a window of at least of 9×9 pixels.

The Primesense pattern only exploits spatial multiplexing without any temporal or range multiplexing. The fact that there is no temporal multiplexing ensures that each frame provides an independent depth estimate. The lack of range multiplexing, as well as the presence of collimated dots, enhances the system's ability to estimate depth at far distances. The adopted spatial multiplexing technique leads to a reduced spatial resolution, i.e., the localization of depth edges is reduced.

The Intel RealSense F200

The Intel RealSense F200 has a very compact depth camera that can either be integrated in computers and mobile devices or used as a self-standing device. The Intel RealSense F200 generally comes with an array of microphones, a color camera,

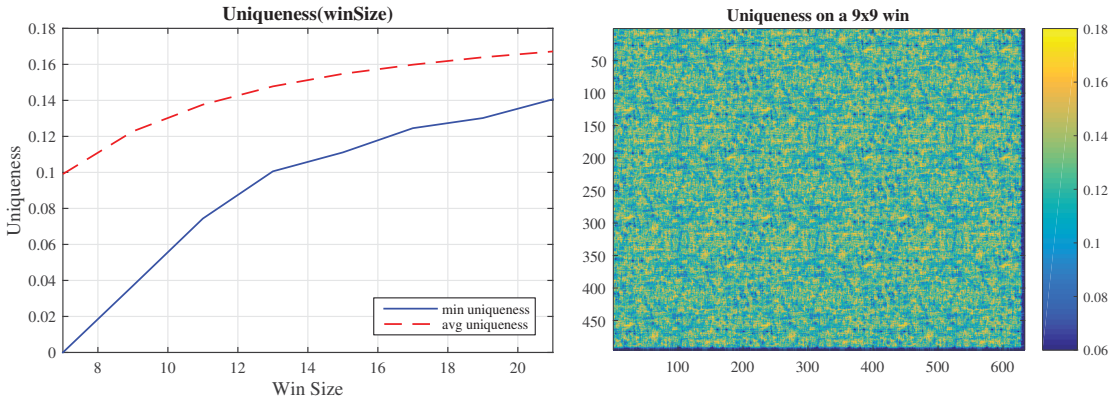


Figure 2.12: Plot of the minimum and average uniqueness of the Primesense pattern as a function of the window size (left) and uniqueness map for a 9×9 window (right).

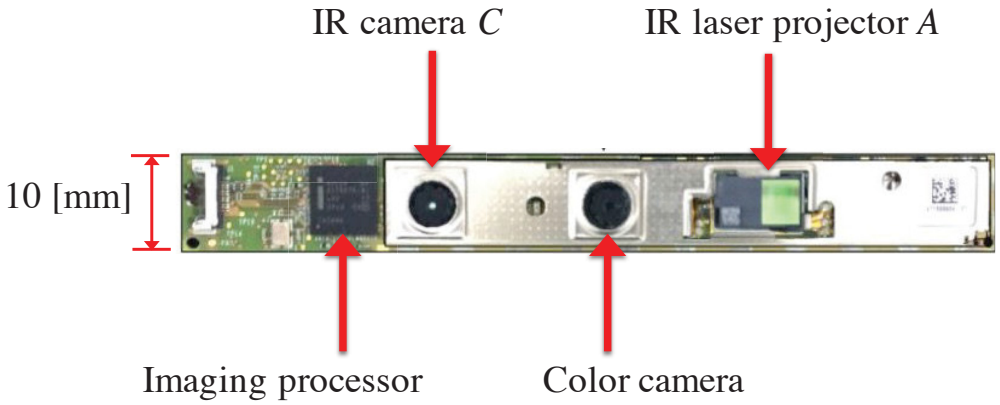


Figure 2.13: Intel RealSense F200 under the hood.

and a depth camera system, made by an IR camera and an IR projector.

The spatial resolution of the depth camera of the Intel RealSense F200 is VGA (640×480), the working depth range is $200 - 1200$ [mm], and the temporal resolution is up to 120 [Hz]. The horizontal Field-of-View (FoV) of the Intel RealSense F200 depth camera is 73° and the vertical FoV is 59°, with a focal length in pixels of approximately 430 [pxl]. Such characteristics are well suited to applications such as face detection or face tracking, gesture recognition, and to applications that frame a user facing the screen of the device. The letter "F" in the name hints at the intended "Frontal" usage of this device.

Figure 2.13 shows the positions of the three most important components of the structured light depth camera, i.e., the IR camera, the IR projector plus a color camera. The presence of a single IR camera indicates that the Intel RealSense F200 exploits the concept of a virtual camera.

Note that the baseline between the IR camera C and the IR projector A is approximately 47 [mm]. Figure 2.14 shows the depth resolution of the Intel

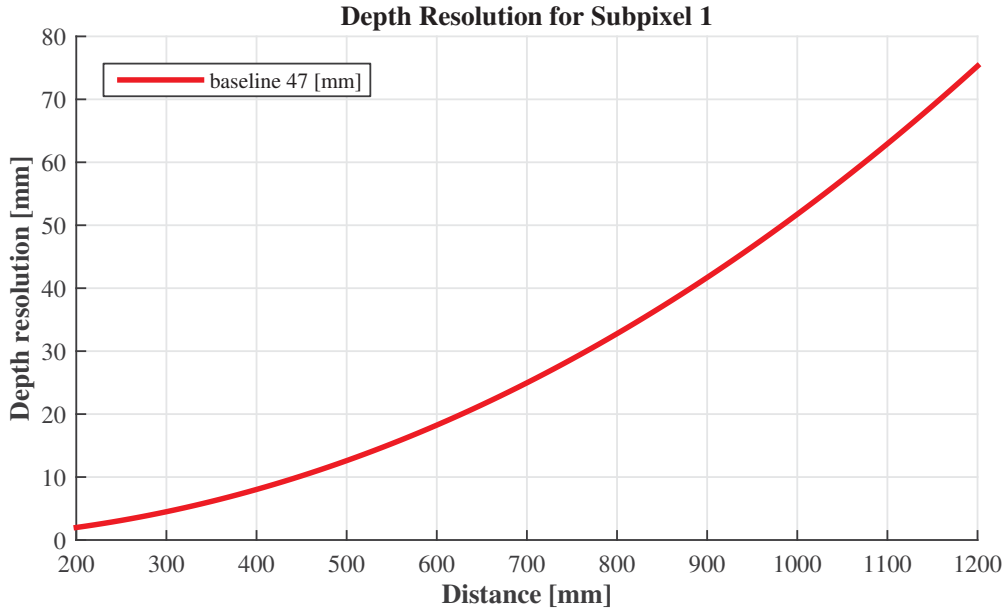


Figure 2.14: Depth resolution without sub-pixel interpolation vs. measured depth distance of Intel RealSense F200.

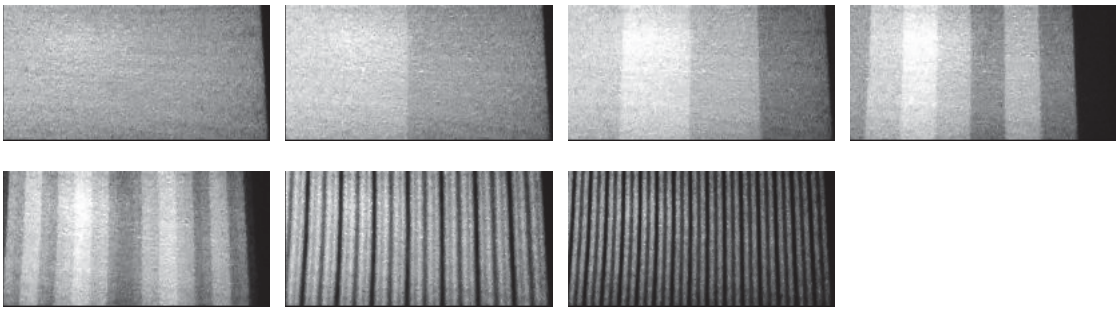


Figure 2.15: Patterns projected by the projector of the Intel RealSense F200 camera.

RealSense F200 depth camera, without sub-pixel interpolation, as a function of the measured depth, according to (2.3) given the baseline and the focal length in pixels.

The projector of the Intel RealSense F200 is the most interesting component of the depth camera itself. It is a dynamic projector, which projects vertical light stripes of variable width at three different brightness or range levels, an approach similar to Gray code patterns. According to the adopted terminology, the Intel RealSense F200 depth camera uses both temporal and range multiplexing.
The impressively high pattern projection frequency in the order of 100 [Hz] makes reverse engineering complex. Figure 2.15 shows the pattern projected by the Intel RealSense F200 obtained by a very fast camera operating at frame rate 1200 [Hz]. Figure 2.15 clearly shows that there are at least six layers of independent projected patterns at three range levels, leading to $3^6 = 729$ possible pattern configurations for a set of six frames. Since the number of different configurations is an upper

Figure 2.16: Example of pixel-wise independent depth measurements obtained by the Intel RealSense F200 depth camera. The edges of the framed hand are pixel-precise and do not present edge jaggedness typical of spatial multiplexing techniques.

bound for the maximum measurable disparity (corresponding to the closest measurable distance), this characteristic is functional to avoid limitations on the closest measurable depth and to reliably operate in close ranges. Since the Intel RealSense F200 projector does not use spatial multiplexing, there is no spatial sampling and the depth camera operates at full VGA spatial resolution. Figure 2.16 shows that the edge jaggedness typical of spatial multiplexing is not exhibited by the image captured by the Intel RealSense F200 due to its pixel-precise spatial resolution.

Conversely, the data produced by Intel RealSense F200 exhibit artifacts typical of temporal multiplexing when the scene content moves during the projection of the set of patterns needed for depth estimation. An example of these artifacts is the *ghosting effect* shown by Figure 2.17. Moreover, the combination of the characteristics of the illuminator design, of the fact that the illuminator produces stripes and not dots, and of the virtual camera approach makes the Intel RealSense F200 depth camera highly sensitive to the presence of external illumination. In fact, as indicated by the official specifications, this structured light system is meant to work indoors, as the presence of external illumination leads to a considerable reduction of its working depth range.

The above analysis suggests that the design of the Intel RealSense F200 depth camera is inherently targeted to a limited depth range allowing for pixel-precise, fast, and accurate depth measurements, particularly well suited for frontal facing applications with maximum depth range of 1200 [mm].

Figure 2.17: Artifacts in the depth estimate of a moving hand acquired by the Intel RealSense F200 depth camera. The depth of the moving hand should only be the brightest silhouette, however a shadowed hand appears in the estimated depth map.

The Intel RealSense R200

Like the Intel RealSense F200, the Intel RealSense R200 has a very compact depth camera that can either be integrated in computers and mobile devices or used as a self-standing device. The Intel RealSense R200 generally comes with a color camera and a depth camera system, made by two IR cameras and not only one like the Intel RealSense F200, and by an IR projector.

The spatial resolution of the structured light depth camera of the Intel RealSense R200 is VGA (640×480), the working depth range is $510 - 4000$ [mm], and the temporal resolution is up to 60 [Hz]. The horizontal Field-of-View (FoV) of the Intel RealSense R200 depth camera is approximately 56° and the vertical FoV is 43°, with a focal length in pixels of approximately 600 [pxl]. Such characteristics are very well suited for applications such as people tracking and 3D reconstruction, and in general for applications that frame the portion of the world behind the rear part of the device. The letter "R" in the name hints at the intended "Rear" usage of this device.

Figure 2.18 shows the Intel RealSense R200's most important components, namely, the two IR cameras and the IR projector plus the color camera. Since the Intel RealSense R200 carries a pair of IR cameras, there is no need for a virtual camera. The baseline between the left IR camera and the IR projector is 20 [mm] and the baseline between the two IR cameras is 70 [mm]. Figure 2.19 shows the depth resolution of the Intel RealSense R200 depth camera (without sub-pixel interpolation) as a function of the measured depth, according to (2.3) given the baseline and the focal length in pixels.

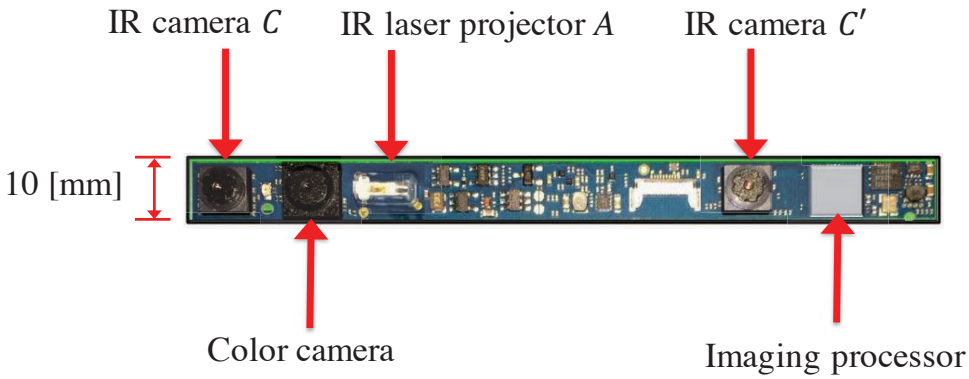


Figure 2.18: Intel RealSense R200 under the hood.

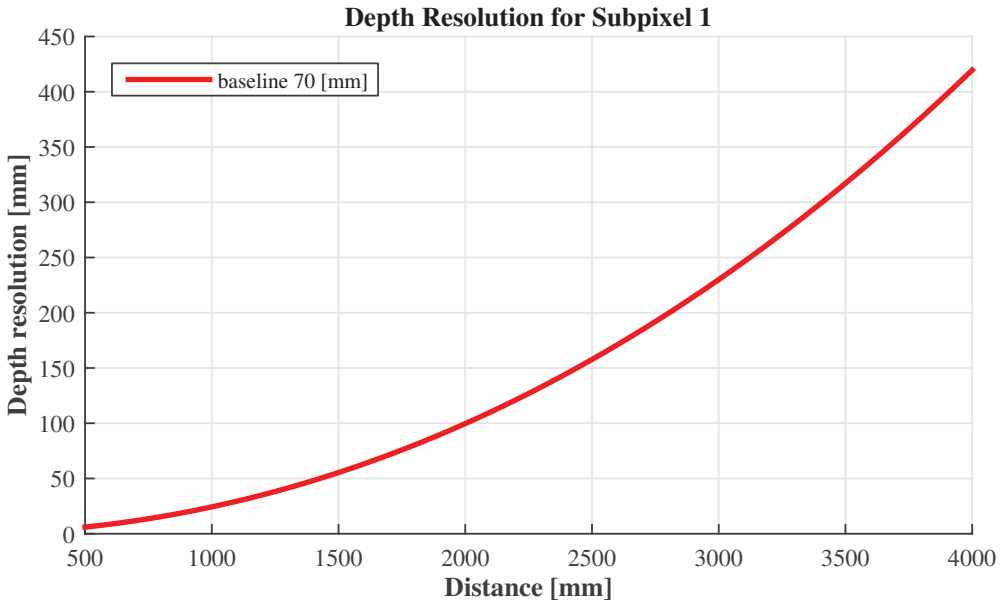


Figure 2.19: Depth resolution without sub-pixel interpolation vs. measured depth distance of Intel RealSense R200.

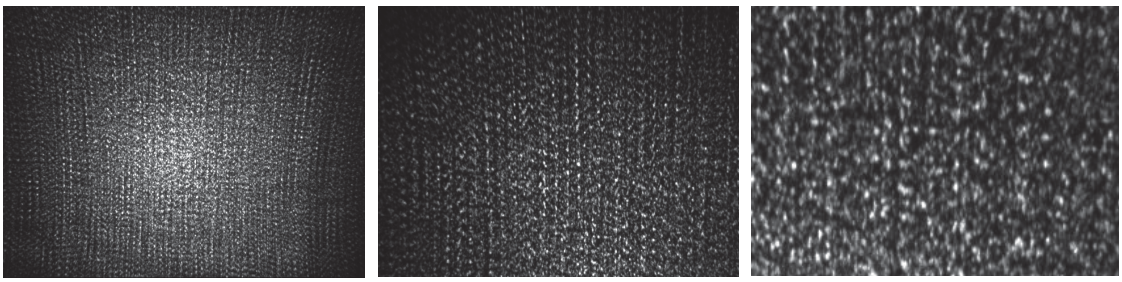


Figure 2.20: Texture projected by the illuminator of the Intel RealSense R200 camera, framed at different zoom levels: (left) the full projected pattern; (center) a pattern zoom; (right) a macro acquisition.

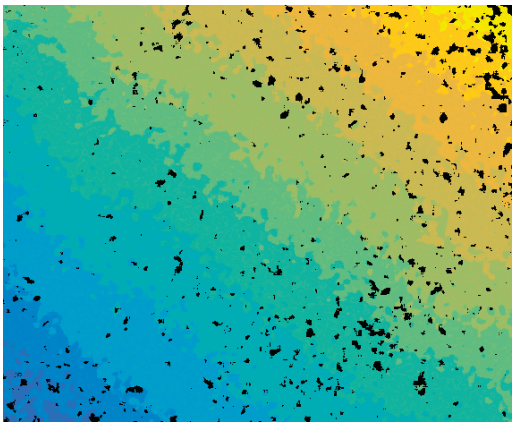


Figure 2.21: Missing depth estimates, "black holes", in the data produced by the Intel RealSense R200 camera in the aquisition of a planar surface.

Also in this case, the projector of the Intel RealSense R200 is the most interesting component of the depth camera itself. Here, it is a static projector providing texture to the scene. Differently from the Primesense camera, the pattern of the Intel RealSense R200's projector is not made by collimated dots. Compared to other cameras, the projector dimensions are remarkably small. In particular, the box length along the depth axis, usually called Z-height, is about 3.5 [mm], a characteristic useful for integration in mobile platforms.

Figure 2.20 shows the pattern projected by the IntelRealSense R200 camera. These images show how the texture is uncollimated and made by elements of different intensity and without a clear structure. The purpose of this texture is to add features to the component of the different reflectance elements of the scene to improve uniqueness. Since the projected texture is not collimated, it does not completely dominate the scene uniqueness, with the consequence of possibly missing depth estimates, i.e., of undefined depth values called "black holes" in some areas of the framed scene, as exemplified by Figure 2.21

The Intel RealSense R200 projects constant illumination that does not vary in

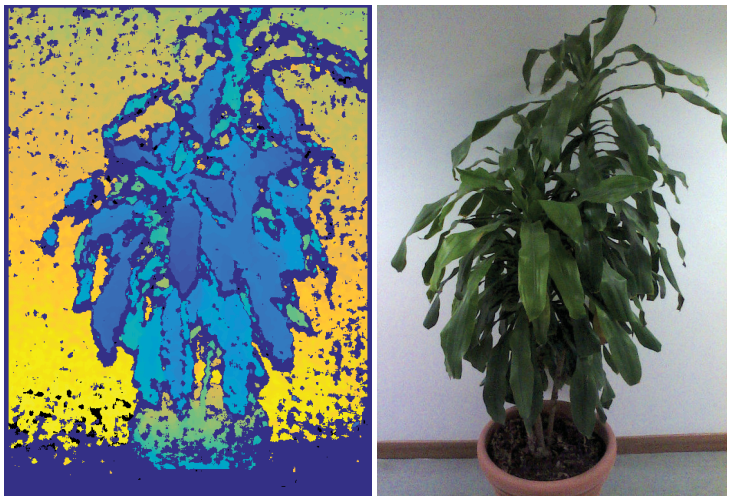


Figure 2.22: The Intel RealSense R200 camera depth estimation process is based on spatial multiplexing, leading to coarse edges, as clearly shown from the depth map of the leaves of the framed plant.

time, hence the system is characterized only by range and spatial multiplexing. There is no temporal multiplexing. The estimated depth-maps are therefore characterized by full temporal resolution with an independent depth estimate provided for each acquired frame, and by a subsampled spatial resolution, i.e., the localization of edges in presence of depth discontinuities is bounded by the size of the correlation window used in the depth estimation process. This subsampled spatial resolution leads to coarse estimation of the depth edges, as shown in Figure 2.22.

The above analysis suggests that the Intel RealSense R200 structured light depth camera is designed to target rear-facing applications, such as objects or environment 3D modeling. The Intel RealSense R200 has an illuminator which projects a texture meant to aid scene reflectance, making this depth camera suitable for acquisitions both indoors and outdoors under reasonable illumination, within nominal range $500 - 4000$ [mm]. Since the projected texture is not made by collimated dots, the depth estimates may exhibit missing measurements, especially outdoors when the external illumination affects the contribution of the projected texture, and indoors when the scene texture is inadequate to provide uniqueness.

2.3 Time-of-Flight depth cameras

Time-of-Flight depth cameras (or simply ToF cameras) are active sensors capable of acquiring 3D geometry of a framed scene at video rate. ToF and Light Detection And Ranging (LIDAR) devices operate on the basis of the Radio Detection And Ranging (RADAR) principle, which rests on the fact that the electro-magnetic

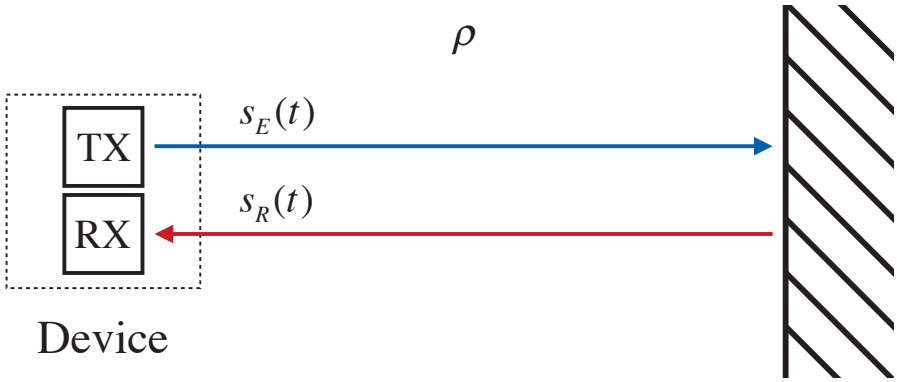


Figure 2.23: Scheme of principle of ToF measurement.

radiation travels in air at light speed $c \approx 3 \times 10^8 [m/s]$. Hence, the distance ρ [m] covered at time τ [s] by an optical radiation is $\rho = c\tau$. Figure 2.23 shows the typical ToF measurement scheme: the radiation $s_E(t)$ emitted at time 0 by the ToF transmitter (or illuminator) TX on the left travels straight towards the scene for a distance ρ. It is then echoed or back-reflected by a point on the scene surface and travels a distance ρ. At time τ it reaches the ToF receiver (or sensor) RX, ideally co-positioned with the transmitter, as signal $s_R(t)$. Since at time τ the path length covered by the radiation is 2ρ, the relationship between ρ and τ is

$$\rho = \frac{c\tau}{2} \tag{2.8}$$

which is the basic expression of a ToF camera's distance measurement.

ToF systems made by a single transmitter and receiver, as schematically shown in Figure 2.23, are typically used in range-finders for point-wise or 0D measurements. ToF cameras estimate the scene geometry in a single shot by a matrix of $N_R \times N_C$ in-pixel ToF sensors where all the pixels independently but simultaneously measure the distance of the scene point in front of them.

In stereo or structured light systems, occlusions are inevitable due to the presence of two cameras, or a camera and a projector, in different positions. Additionally, the distance between the camera positions (i.e. the baseline) improves the distance measurement accuracy. This is an intrinsic difference with respect to ToF, in which measurements are essentially occlusion-free, because the ToF measurement scheme assumes the transmitter and receiver are collinear and ideally co-positioned. In common practice such a requirement is enforced by placing them as close together as possible. Another important characteristic of ToF systems which differs from stereo and structured light systems is that the measurement accuracy is distance independent, only depending on the accuracy of the time or phase measurement devices.

ToF depth cameras lend themselves to a countless variety of different solutions, however, all the current implementations share the same structure shown in Figure 2.24 made by the following basic components:

- a transmitter made by an array of LEDs which generates a sinusoidal or square wave modulating signal in the high HF or low VHF bands, in tens of MHz, embedded in an optical NIR signal, in hundreds of THz;
- a suitable optics diffusing the optical signal generated by the transmitter to the scene;
- a suitable optics collecting the NIR optical radiation echoed by the scene and imaging it onto the receiver matricial ToF sensor. This component includes an optical band-pass filter with center-band tuned to the NIR carrier frequency of the transmitter to improve the SNR;
- a matricial ToF sensor of $N_R \times N_C$ pixels estimating simultaneously and independently the distance between each ToF sensor pixel p_T and the imaged scene point P;
- suitable circuitry providing the needed power supply and control signals to transmitter and receiver.

The choice of modulation determines the basic transmitter and receiver functions and structure. Although in principle many modulation types suit ToF depth cameras, in practice, all current commercial ToF depth camera products [88, 48, 76, 77] adopt only one type of CW modulation, namely homodyne amplitude modulation with either a sinusoidal or square wave modulating signal $m_E(t)$. This is because current microelectronic technology solutions for homodyne AM are more mature than others for commercial applications. The advantages of AM modulation, besides its effective implementability by current CMOS solutions, are that it uses a single modulation frequency f_m and does not require a large bandwidth. A major disadvantage is that it offers little defense against multipath and other propagation artifacts (see Chapter 5 of [91]).

Other modulation types than AM could be usefully employed in ToF depth cameras and their implementation is being actively investigated [91]. Other candidate modulation types include pulse modulation and pseudo-noise modulation. The former, as already mentioned, is the preferred choice for single transmitter and receiver ToF systems. Although in principle it would be equally suitable for matrix ToF sensors, in practice its application is limited by the difficulties associated

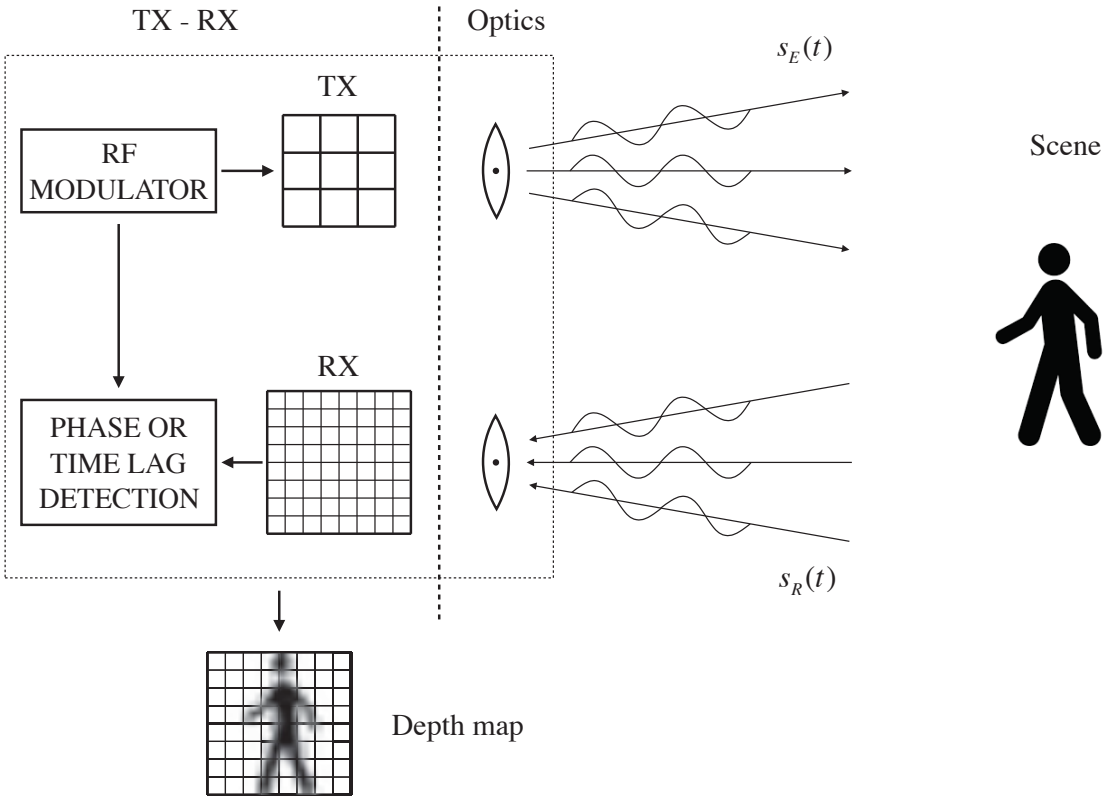


Figure 2.24: Basic ToF depth camera structure

with implementing effective stop-watch at pixel level within matrix arrangements. Current research approaches this issue in various ways (see Chapter 2 and 3 of [91]). Pseudonoise modulation would be very effective against multipath, as other applications such as indoor radio localization [23] indicate.
CW modulation itself offers alternatives to homodyne AM, such as heterodyne AM or frequency modulation (FM) with chirp signals. Such properties, although reported in ToF measurement literature, are still problematic for matrix ToF sensor electronics. The remainder of this section considers the basic characteristics of ToF depth camera transmitters and receivers assuming the underlying modulation is Continuous Wave Amplitude Modulation (CWAM).

ToF depth camera transmitter basics

Lasers and LEDs are the typical choice for the light sources at the transmitter since they are inexpensive and can be easily modulated by signals within the high HF or low VHF bands up to some hundreds of MHz. The LED emissions typically used are in the near infrared (NIR) range, with wavelength around $\lambda_c = 850$ [nm],

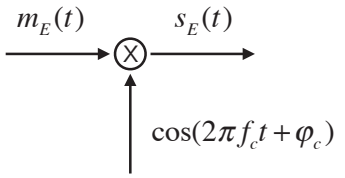


Figure 2.25: Transmitter model of a ToF camera.

corresponding to

$$f_c = \frac{c}{\lambda_c} = 3 \times 10^8 \ \frac{[\text{m}]}{[\text{s}]} \ \frac{1}{850 \times 10^{-9} \ [\text{m}]} \cong 352 \ [\text{THz}]. \tag{2.9}$$

The transmitter illuminates the scene by an optical 2D wavefront signal which, for simplicity, can be modeled as in Figure 2.25, where

$$s_E(t) = m_E(t) \ \cos(2\pi f_c t + \varphi_c) \tag{2.10}$$

denotes the emitter NIR signal structured as the product of a carrier with NIR frequency f_c,of some hundreds of THz, and phase φ_c and a modulating signal $m_E(t)$. Signal $m_E(t)$, in turn, incorporates AM modulation of either sinusoidal or square wave type in current products with frequency f_m, of some tens of MHz, and φ_m. In current products there are two levels of AM modulation. The first is AM modulation at NIR frequencies concerning the optical signal $s_E(t)$ used to deliver the modulating signal $m_E(t)$ at the receiver. The second is AM modulation in the high HF or low VHF bands embedded in $m_E(t)$, which delivers information related to round-trip time τ to the receiver, either in terms of phase or time lag.

The current ToF camera NIR emitters are either lasers or LEDs. Since they cannot be integrated, they are typically positioned in configurations mimicking the presence of a single emitter co-positioned with the optical center of the ToF camera. The geometry of the emitters' position is motivated by making the sum of all the emitted NIR signals equivalent to a spherical wave emitted by a single emitter, called *simulated emitter*, placed at the center of the emitters constellation. The LED configuration of the Mesa Imaging SR4000, shown in Figure 2.26, is an effective example of this concept.

The arrangement of the actual emitters, such as the one of Figure 2.26, is only an approximation of the non-feasible juxtaposition of single ToF sensor devices with emitter and receiver perfectly co-positioned and it introduces a number of artifacts, including a systematic distance measurement offset that is larger for close scene points than far scene points.

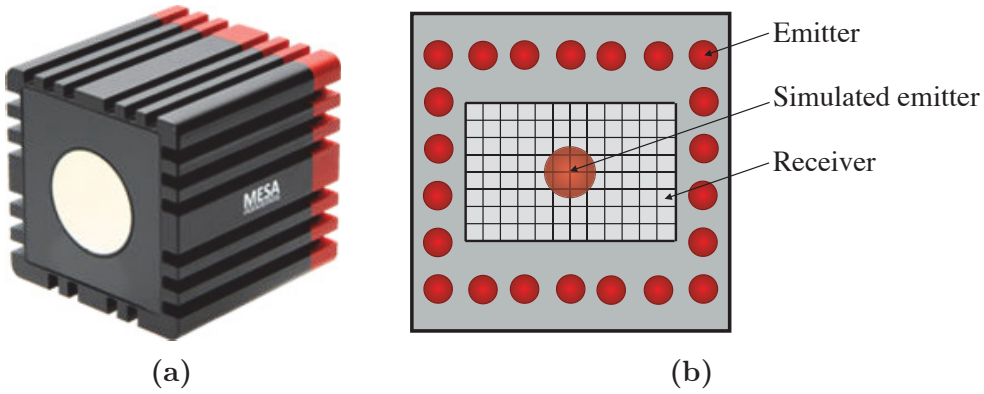


Figure 2.26: The NIR emitters of the MESA Imaging SR4000: (a) the actual depth camera; (b) the emitters are distributed around the lock-in pixels matrix and mimic a simulated emitter co-positioned with the center of the lock-in pixel matrix.

ToF depth camera receiver basics

The heart of ToF camera receivers is a matricial sensor with individual elements, called pixels because of their imaging role, individually and simultaneously capable of independent ToF measurements. Each pixel independently computes the delay between the departure of the sent signal $s_E(t)$ and the arrival of the signal $s_R(t)$ back-projected by the scene point P imaged by the pixel. Currently there are three main technological solutions (Chapter 1 of [91]) considered best suited for the realization of such matricial ToF sensors, namely Single-Photon Avalanche Diodes (SPADs) assisted by appropriate processing circuits, standard photo diodes coupled to dedicated circuits and the In-Pixel Photo-Mixing devices. The latter technology includes the lock-in CCD sensor of [65], the photonic mixer device [32, 115], and other variations [3, 5]. Section 2.3 will only recall the main characteristics of the In-Pixel Photo-Mixing devices, since so far it is the only one adopted in commercial products [88, 48, 76, 77]. An in-depth treatment of such a technology can be found in [65] and [91].

Figure 2.27 offers a systems interpretation of the basic functions performed by each pixel of a sensor based on photo-mixing device technology, which are

a) photoelectric conversion

b) correlation or fast shutter

c) signal integration by charge storage on selectable time intervals

For analysis purposes it is useful to recognize and subdivide the various operations as much as possible. On the contrary, multifunctional components are the

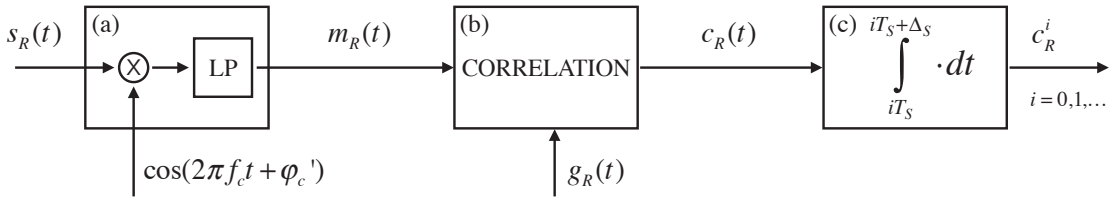


Figure 2.27: System interpretation of the operation of a single pixel of a sensor based on In-Pixel Photo-Mixing devices technology.

typical choice for circuit effectiveness. This section presents ToF depth cameras from a system perspective that it does not always coincide with the circuit block description.

Each sensor pixel receives as input the optical NIR signal back-projected by the scene point P imaged by the pixel itself, which can be modeled as

$$s_R(t) = m_R(t)\ \cos(2\pi f_c t + \varphi_c') + n_R(t) \tag{2.11}$$

where $m_R(t)$ denotes the transformations of the modulating signal $m_E(t)$ actually reaching the receiver, since direct and reflected propagation typically affect some parameters of the transmitted signal $m_E(t)$ (for instance amplitude attenuation is inevitable) and $n_R(t)$ is the background wide-band light noise at the receiver input[3].

The photoelectric conversion taking place at the pixel in the scheme of Figure 2.27 is modeled as a standard front-end demodulation stage (a) with a carrier $\cos(2\pi f_c t + \varphi_c')$ at NIR frequency f_c followed by a low pass filter (LP). The input of stage (a) is the optical signal $s_R(t)$ and the output is the baseband electrical signal $m_R(t)$. Stage (b) represents the correlation between baseband signal $m_R(t) + n(t)$ and reference signal $g_R(t)$. Stage (c) models the charge accumulation process as an integrator operating on time intervals of selectable lengths Δ_S starting at uniformly spaced clock times iT_s, $i = 1, \ldots$ where T_s is the sampling period.

2.3.1 ToF measurement methods

In spite of the conceptual simplicity of relationship (2.8), its implementation presents tremendous technological challenges because it involves the speed of light.

[3]The phase φ_c of the carrier at the transmitter side is generally different from the phase φ_c' at the receiver. Both φ_c and φ_c' are usually unknown, especially in the case of a non-coherent process. However, the system does not need to be aware of the values of φ_c and φ_c' and it is inherently robust to the lack of their knowledge.

For example, since

$$c = 3 \times 10^8 \ \frac{[m]}{[s]} = 2 \times 150 \ \frac{[m]}{[ps]} = 2 \times 0.15 \ \frac{[\text{mm}]}{[\text{ps}]} \tag{2.12}$$

it takes 6.67 [ns] to cover a 1 [m] path and distance measurements of nominal resolution of 1 [mm] need time measurement mechanisms with accuracy superior to $6.67 \div 7$ [ps], while a nominal resolution of 10 [mm] needs accuracy superior to 70 [ps].

The accurate measurement of round-trip time τ is the fundamental challenge in ToF systems and can be solved by two approaches: direct methods, addressing either the measurement of time τ by pulsed light or of phase φ with continuous wave operation, and indirect methods deriving τ (or φ as an intermediate step) from time-gated measurements of signal $s_R(t)$ at the receiver.

As anticipated in the previous section, all the current commercial depth cameras adopt homodyne AM modulation with circuitry based on various solutions related to In-Pixel Photo-Mixing devices [38, 65], simply called in-pixel devices. Figure 2.25 and 2.27 show a conceptual model of the operation of an homodyne AM transmitter and receiver, which are co-sited in a ToF camera, unlike in typical communication systems. Telecommunication systems convert the signal sent by the transmitter into useful information. In contrast, ToF systems only estimate the round-trip delay of the signal rather than the information encoded inside the signal.

Both $s_E(t)$ and $s_R(t)$ are optical signals and that the modulation schemes of Figure 2.27 are an appropriate description for the operation of the transmitter but not for the photoelectric conversion of the receiver. Indeed, the actual light detection mechanism of the in-pixel devices is such that a baseband voltage signal $m_R(t)$ is generated from the optical input $s_R(t)$, without direct demodulation as in the transmitter side.

The electric modulating signal $m_E(t)$ can be either a sine wave of period T_m

$$m_E(t) = A_E[1 + \sin(2\pi f_m t + \varphi_m)] \tag{2.13}$$

with $f_m = 1/T_m$, or a square wave of support $\Delta_m < T_m$ spaced by the modulation period T_m

$$m_E(t) = A_E \sum_{k=0}^{\infty} p(t - kT_m + \varphi_m;\ \Delta_m) \tag{2.14}$$

where

$$p(t;\Delta) = \text{rect}\left(\frac{t-\frac{\Delta}{2}}{\Delta}\right) = \begin{cases} 1 & 0 \le t \le \Delta \\ 0 & \text{otherwise.} \end{cases} \tag{2.15}$$

The pulse $p(t;\Delta)$ in (2.15) is modeled by a rectangle for simplicity, however, this is only a nominal reference signal given the practical difficulty of obtaining sharp rise and fall signals.

At the receiver, after the demodulation of the optical signal $s_R(t)$, the baseband electrical signal $m_R(t)$, of shape similar to that of $m_E(t)$, is correlated with the reference signal $g_R(t)$ with period T_m, obtaining

$$c_R(t) = \int_0^{T_m} m_R(t) g_R(t+t')\ \mathrm{d}t'. \tag{2.16}$$

The signal $c_R(t)$ is sampled according to the "natural sampling" paradigm by the charge accumulator circuit at the back-end of the receiver and can be modeled as a system which at each sampling time iT_s, $i = 0, 1, \ldots$, returns the integration of $c_R(t)$ in the support Δ_S

$$c_R^i = \int_{iT_S}^{iT_S+\Delta_S} c_R(t)\ \mathrm{d}t. \tag{2.17}$$

Clearly for designing ToF camera sensors there is a countless number of combinations of $m_E(t)$, $g_R(t)$ and Δ_S value choices. The two basic situations of sinusoidal and square modulating signal $m_E(t)$ and related choices of $g_R(t)$ and Δ_S will be discussed next.

Sinusoidal modulation

In the case of sinusoidal modulation, the ToF camera transmitter modulates the NIR optical carrier by a modulation signal $m_E(t)$ made by a sinusoidal signal of amplitude A_E and frequency f_m, namely

$$m_E(t) = A_E[1 + \sin(2\pi f_m t + \varphi_m)]. \tag{2.18}$$

Signal $m_E(t)$ is reflected back by the scene surface within $s_E(t)$ and travels back towards the receiver ideally co-positioned with the emitter.

The HF/VHF modulating signal reaching the receiver, due to factors such as the energy absorption associated with the reflection, the free-path propagation attenuation (proportional to the square of the distance), and the non-instantaneous

propagation of IR optical signals leading to a phase delay $\Delta\varphi$, can be written as

$$\begin{aligned} m_R(t) &= A_R[1 + \sin(2\pi f_m t + \varphi_m + \Delta\varphi)] + B_R \\ &= A_R \sin(2\pi f_m t + \varphi_m + \Delta\varphi) + (A_R + B_R) \end{aligned} \tag{2.19}$$

where A_R is the attenuated amplitude of the received modulating signal and B_R is due to the background light interfering with λ_c and to other artifacts. Figure 2.28 shows an example of emitted and received modulating signal. For simplicity we will call A_R simply A and $A_R + B_R$ simply $B/2$, obtaining

$$m_R(t) = A \sin(2\pi f_m t + \varphi_m + \Delta\varphi) + \frac{B}{2}. \tag{2.20}$$

Quantity A is called *amplitude*, since it is the amplitude of the useful signal. Quantity B is called *intensity* or *offset*, and it is the sum of the received modulating signal, with a component A_R due to the sinusoidal modulation component at f_m, and an interference component B_R, mostly due to background illumination. It is common to call A and B amplitude and intensity respectively, even though both A and B are signal amplitudes (measured in $[V]$).

If the correlation signal at the receiver is

$$g_R(t) = \frac{2}{T_m} [1 + \cos(2\pi f_m t + \varphi_m)] \tag{2.21}$$

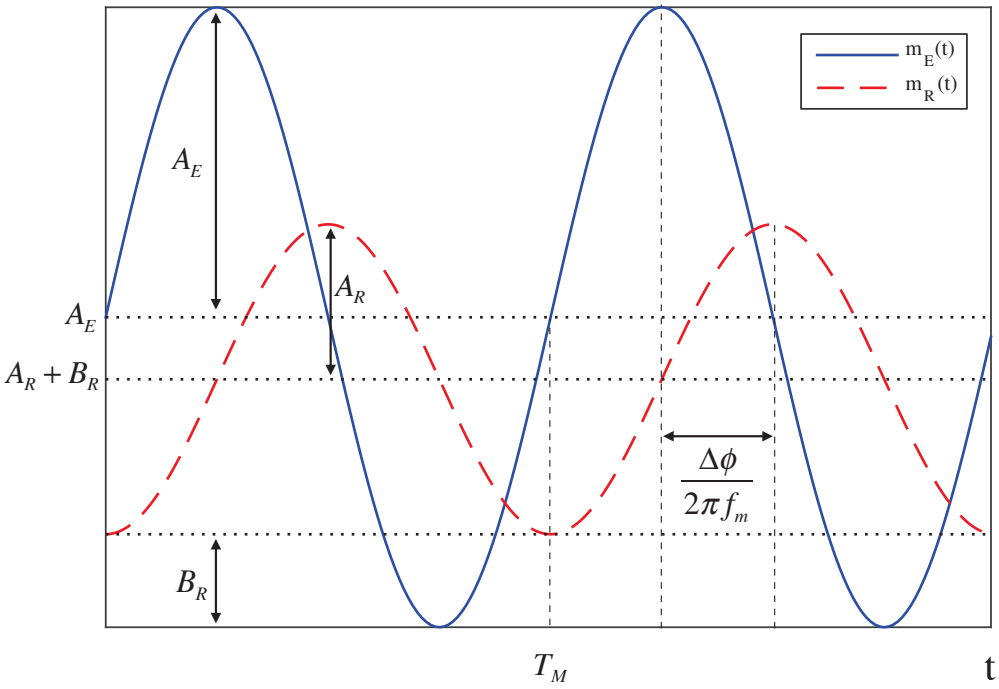


Figure 2.28: Example of an emitted modulating signal $m_E(t)$ and a received modulating signal $m_R(t)$.

and the output of the correlation circuit is

$$
\begin{aligned}
c_R(t) &= \int_0^{T_m} m_R(t')g_R(t'+t)\ \mathrm{d}t' \\
&= \frac{2}{T_m}\int_0^{T_m}\left[A\sin(2\pi f_m t' + \varphi_m + \Delta\varphi) + \frac{B}{2}\right]\left[1+\cos(2\pi f_m(t'+t)+\varphi_m)\right]\ \mathrm{d}t' \\
&= \frac{2}{T_m}\int_0^{T_m} A\sin(2\pi f_m t' + \varphi_m + \Delta\varphi)\ \mathrm{d}t' + \frac{2}{T_m}\int_0^{T_m}\frac{B}{2}\ \mathrm{d}t' + \\
&+ \frac{2}{T_m}\int_0^{T_m} A\sin(2\pi f_m t' + \varphi_m + \Delta\varphi)\cos(2\pi f_m(t'+t)+\varphi_m)\ \mathrm{d}t' + \\
&+ \frac{2}{T_m}\int_0^{T_m}\frac{B}{2}\cos(2\pi f_m(t'+t)+\varphi_m)\ \mathrm{d}t' \\
&= A\sin(\Delta\varphi - 2\pi f_m t) + B.
\end{aligned}
\tag{2.22}
$$

Note that since transmitter and receiver are co-sited, the modulation sinusoidal signal (therefore including its phase φ_m) is directly available at the receiver side.

The unknowns of (2.22) are A, B and $\Delta\varphi$, where A and B are measured in Volts $[V]$ and $\Delta\varphi$ as phase value is a pure number. The most important unknown is $\Delta\varphi$, since it can deliver distance ρ. Unknowns A and B will be shown later to be important for SNR considerations.

To estimate the unknowns A, B and $\Delta\varphi$, $c_R(t)$ must be sampled by an ideal sampler, i.e. with $\Delta_S \to 0$ in (2.17), at least 4 times per modulation period T_m [65], i.e., $T_s = T_m/4$. For instance, if the modulation frequency is 30 [MHz], signal $c_R(t)$ must be sampled at least at 120 [MHz]. Assuming a sampling frequency $F_S = 4f_m$, given the 4 samples per period $c_R^0 = c_R(t=0)$, $c_R^1 = c_R(t=1/F_S)$, $c_R^2 = c_R(t=2/F_S)$ and $c_R^3 = c_R(t=3/F_S)$, the receiver estimates values $\hat{A}$, $\hat{B}$ and $\widehat{\Delta\varphi}$ as

$$(\hat{A}, \hat{B}, \widehat{\Delta\varphi}) = \underset{A,B,\Delta\varphi}{\operatorname{argmin}} \sum_{n=0}^{3} \left\{ c_R^n - \left[A \sin\left(\Delta\varphi - \frac{\pi}{2}n\right) + B \right] \right\}^2. \tag{2.23}$$

After some algebraic manipulations of (2.23) one obtains

$$\begin{aligned} \hat{A} &= \frac{\sqrt{(c_R^0 - c_R^2)^2 + (c_R^3 - c_R^1)^2}}{2} \\ \hat{B} &= \frac{c_R^0 + c_R^1 + c_R^2 + c_R^3}{4} \\ \widehat{\Delta\varphi} &= \operatorname{atan2}\left(c_R^0 - c_R^2, c_R^3 - c_R^1\right). \end{aligned} \tag{2.24}$$

The final distance estimate $\hat{\rho}$ can be obtained as

$$\hat{\rho} = \frac{c}{4\pi f_m} \widehat{\Delta\varphi}. \tag{2.25}$$

If one takes into account that the sampling is not ideal but actually made by a sequence of rectangular pulses of width Δ_S within the standard natural sampling model, the estimates of A and B in this case become [81]

$$\begin{aligned} \hat{A}' &= \frac{\pi}{T_S \sin\left(\frac{\pi\Delta_S}{T_S}\right)} \hat{A} \\ \hat{B}' &= \frac{\hat{B}}{\Delta_S} \\ \widehat{\Delta\varphi} &= \widehat{\Delta\varphi}' \end{aligned} \tag{2.26}$$

showing that the phase shift $\Delta\varphi$ is independent from the size of the sampling duration Δ_S, that instead affects both the estimate of A and B. A typical value of

Δ_S is $\Delta_S = T_m/4 = 1/(4f_m)$.

Square wave modulation

In the case of square wave modulation the ToF camera transmitter modulates the NIR optical carrier by a square wave $m_E(t)$ of amplitude A_E and frequency $f_m = 1/T_m$ in the HF/VHF band

$$m_E(t) = A_E \sum_{k=0}^{\infty} p(t - kT_m;\ \Delta_m) \tag{2.27}$$

where $\Delta_m \leq T_m$. The phase φ_m of $m_E(t)$ is not explicitly written for notational simplicity. Because of the co-siting of transmitter and receiver, $m_E(t)$ is available also at the receiver and the specific value of φ_m for practical demodulation purposes is irrelevant.

The back-reflected HF/VHF modulating signal within $s_R(t)$ reaching the receiver can be written as

$$m_R(t) = A \sum_{k=0}^{\infty} p(t - \tau - kT_m;\ \Delta_m) + B \tag{2.28}$$

where A is the attenuated amplitude of the received modulating signal, B is due to the background light interfering with λ_c and τ is the round-trip time. Clearly, A_E is known and A, τ and B are unknown since the first two depend on target distance and material NIR reflectivity and the latter on the background noise.

In the square wave modulation case there are many ways to estimate A, B and τ, which will be introduced next with a few examples.
Let us first consider the situation exemplified by Figure 2.29 where $m_E(t)$ is defined in (2.27), $m_R(t)$ is defined in (2.28) and

$$g_R(t) = \sum_{k=0}^{\infty} (-1)^k p(t - 2kT_S;\ 2T_S). \tag{2.29}$$

The following reasoning assumes $\tau < T_S$ and $T_S = T_m/4$ and it can be generalized to $m_E(t)$ and $m_R(t)$ having pulses $p(t; \Delta)$ with a different support Δ_m.

For notational convenience, in Figure 2.29 the areas of the portions of the useful signal of $m_E(t)$ falling respectively in the first, second and third sampling period

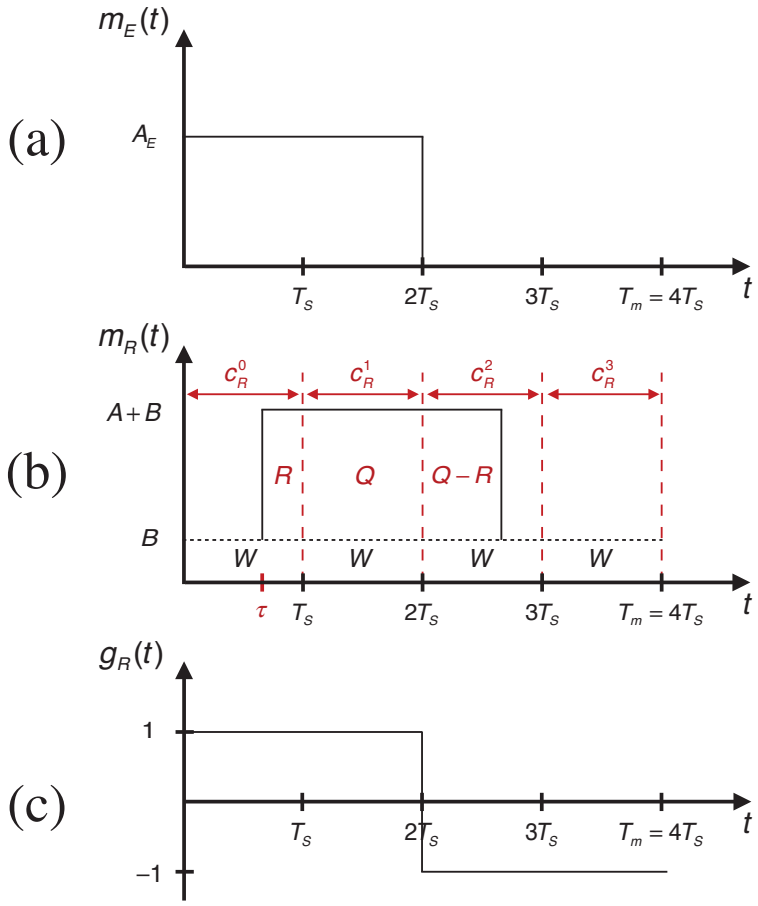


Figure 2.29: Example of one period of square wave signaling: (a) $m_E(t)$, (b) $m_R(t)$ and (c) $g_R(t)$.

are denoted as

$$\begin{aligned} R =& (T_S - \tau)A \\ Q =& T_S A \\ Q - R =& \tau A \end{aligned} \tag{2.30}$$

while the area of the optical noise signal, modeled for simplicity as a constant deterministic signal, in each sampling period is denoted as

$$W = BT_S. \tag{2.31}$$

In this case, from (2.16) and (2.17), again without considering the noise $n_R(t)$, the outputs of the back-end integrator stage are

$$c_R^i = \int_{iT_S}^{iT_S+\Delta_S} m_R(t) g_R(t)\, \mathrm{d}t \tag{2.32}$$

where $\Delta_S = T_S$. As Figure 2.29 schematically indicates, they correspond to the sum of the area of the two components of $m_R(t)$ in each sampling period T_S equal to

$$\begin{aligned} c_R^0 &= R + W = Q\left(1 - \frac{\tau}{T_S}\right) + W \\ c_R^1 &= Q + W \\ c_R^2 &= -[Q - R + W] = -\left[Q\frac{\tau}{T_S} + W\right] \\ c_R^3 &= -W. \end{aligned} \tag{2.33}$$

From (2.33) it is straightforward to see that

$$\begin{aligned} \hat{\tau} =& \frac{T_S}{2}\left(1 - \frac{c_R^2 + c_R^0}{c_R^1 + c_R^3}\right) \\ \hat{A} =& \frac{1}{T_S}\left(c_R^1 + c_R^3\right) \\ \hat{B} =& -\frac{c_R^3}{T_S}. \end{aligned} \tag{2.34}$$

Figure 2.30 shows an alternative scheme for the in-pixel receiver, typically called differential, differing from the scheme of Figure 2.29 for the presence of two

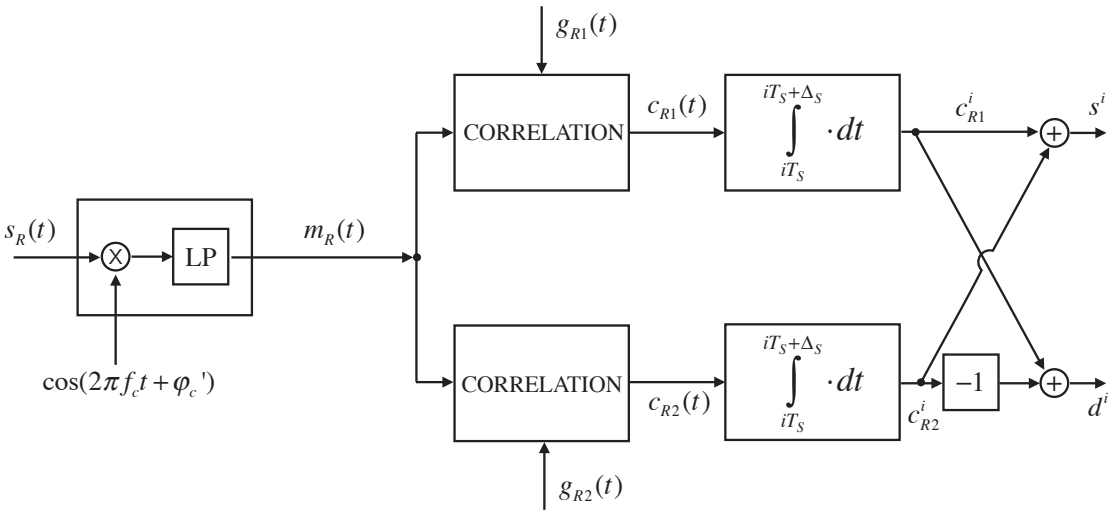


Figure 2.30: Conceptual model of the differential scheme for the in-pixel receiver.

correlators with reference signal $g_{R1}(t)$ and $g_{R2}(t)$ respectively defined as

$$\begin{aligned} g_{R1}(t) &= \sum_{k=0}^{\infty} p(t-(2k)2T_S;\ T_S) \\ g_{R2}(t) &= \sum_{k=0}^{\infty} p(t-(2k+1)2T_S;\ T_S) \end{aligned} \tag{2.35}$$

which operate in parallel. The correlation stage is followed by a subsequent stage where samples c^i_{R1} and c^i_{R2} are added and subtracted obtaining

$$\begin{aligned} s^i &= c^i_{R1} + c^i_{R2} \\ d^i &= c^i_{R1} - c^i_{R2}. \end{aligned} \tag{2.36}$$

At a circuit level, the double correlation and integration stage of Figure 2.30 is amenable to simple and effective solutions, such as a clock signal of sampling period T_S controlling that the incoming photons contribute to charge c^i_{R1} when the clock signal is high, and to charge c^i_{R2} when the clock signal is low [3]. From area relationships (2.36), which apply also in this case, and from Figure 2.31 it is readily seen that

$$\begin{array}{llll} c^0_{R1} = R+W & c^0_{R2} = 0 & s^0 = R+W & d^0 = R+W \\ c^1_{R1} = 0 & c^1_{R2} = Q+W & s^1 = Q+W & d^1 = -[Q+W] \\ c^2_{R1} = Q-R+W & c^2_{R2} = 0 & s^2 = Q-R+W & d^2 = Q-R+W \\ c^3_{R1} = 0 & c^3_{R2} = W & s^3 = W & d^3 = -W \end{array} \tag{2.37}$$

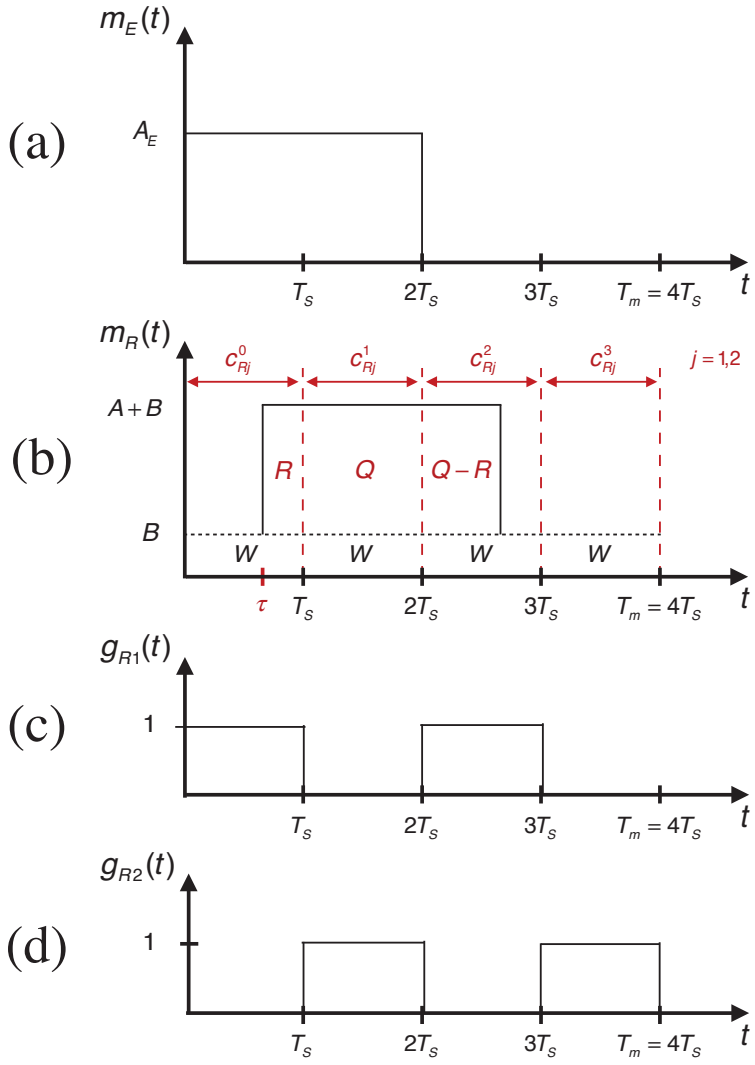


Figure 2.31: Example of one period of square wave signaling for the differential scheme of the in-pixel receiver: (a) $m_E(t)$, (b) $m_R(t)$ and correlation reference signals (c) $g_{R1}(t)$ and (d) $g_{R2}(t)$ defined in (2.35).

from which the unknown parameters can be estimated as

$$\begin{aligned}\hat{\tau} &= T_S \frac{d^1 + d^0}{d^1 - d^3} = T_S \frac{d^3 + d^2}{d^3 - d^1} \\ \hat{A} &= \frac{1}{T_S}(d^3 - d^1) \\ \hat{B} &= -\frac{1}{T_S} d^3.\end{aligned} \tag{2.38}$$

In this case both c^i_{R1} and c^i_{R2} have an intrinsic sampling period of $2T_S$, with c^i_{R2} lagged by T_S with respect to c^i_{R1}. Consequently, samples s^i and d^i of (2.36) carry the same information, therefore unknown parameters $\hat{\tau}$, $\hat{A}$ and $\hat{B}$ could also be obtained from samples s^i instead of d^i with some sign changes.

As a final consideration let us note that if one was only interested in estimating the two parameters τ and A, relying on theoretical or statistical considerations for the noise estimate, the number of measurements per modulation period could be halved. Within our model such a situation corresponds to $B = 0$ and would require the assumptions $\tau < T_S$, $\Delta_S < T_S$ and $T_S = T_m/2$, equivalent to two measurements per modulation period T_m instead of four as in the previous cases. Our simple model requires the assumption $B = 0$ to be extended to the case of two samples per period, even though the noise component B cannot be zero and it can be estimated not from the values of the sample c^i_{R1} and c^i_{R2} but from circuital considerations and measurements. Practically, this can be the case in which B is considered constant within the temporal scale of the receiver sampling period, hence B can be estimated only once and then removed for a subsequent set of measurements.

2.3.2 Imaging characteristics

ToF depth cameras, in spite of their complexity due to the components listed above, can be modeled as pin-hole imaging systems since their receiver has the optics (c) and the sensor (d) made by a $N_R \times N_C$ matrix of lock-in pixels. All the pin-hole imaging system concepts apply to ToF depth cameras. The notation will be used with subscript T to recall that it refers to a ToF depth camera. The CCS of the ToF camera will be called the $3D - T$ *reference system*. The position of a scene point P with respect to the $3D - T$ reference system will be denoted as P_T and its coordinates as $\mathbf{P}_T = [x_T, y_T, z_T]^T$. Coordinate z_T of P_T is called the *depth* of point P_T and the z_T-axis is called the *depth axis*.
The coordinates of a generic sensor pixel p_T of lattice Λ_T with respect to the 2D-T reference system are represented by vector $\mathbf{p}_T = [u_T, v_T]^T$, with $u_T \in [0, \ldots, N_C]$ and $v_T \in [0, \ldots, N_R]$. Therefore the relationship between the $3D$ coordinates

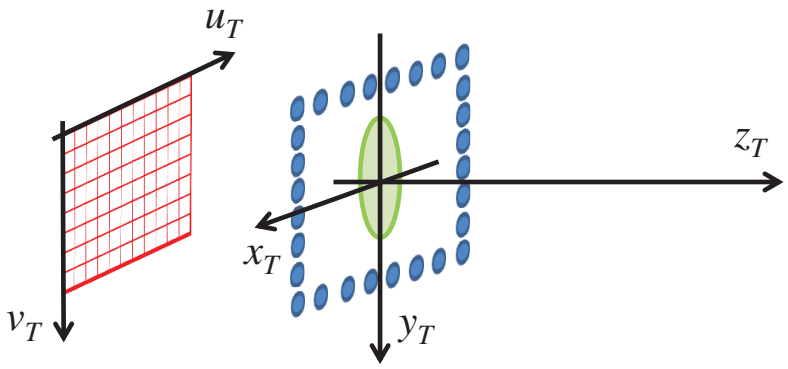


Figure 2.32: 2D T-reference system with axes $u_T - v_T$ and 3D T-reference system with axes $x_T - y_T - z_T$.

$\mathbf{P}_T = [x_T, y_T, z_T]^T$ of a scene point P_T and the $2D$ coordinates $\mathbf{p}_T = [u_T, v_T]^T$ of the pixel p_T receiving the NIR radiation echoed by P_T is given by the perspective projection equation, rewritten for clarity's sake as

$$z_T \begin{bmatrix} u_T \\ v_T \\ 1 \end{bmatrix} = \mathbf{K}_T \begin{bmatrix} x_T \\ y_T \\ z_T \end{bmatrix} \tag{2.39}$$

where $\mathbf{K}_T$ is the ToF camera intrinsic parameters matrix.
Because of lens distortion, coordinates $\mathbf{p}_T = [u_T, v_T]^T$ of (2.39) are related to the coordinates $\hat{\mathbf{p}}_T = [\hat{u}_T, \hat{v}_T]^T$ actually measured by the ToF camera by a relationship of type

$$\mathbf{p}_T = \Psi^{-1}(\hat{\mathbf{p}}_T) \tag{2.40}$$

where $\Psi(\cdot)$ denotes the distortion transformation.
Anti-distortion model (2.41), also called the *Heikkila model* [44], has become popular since it adequately corrects the distortions of most imaging systems and effective methods exist for computing its parameters:

$$\begin{bmatrix} u_T \\ v_T \end{bmatrix} = \Psi^{-1}(\hat{\mathbf{p}}_T) = \begin{bmatrix} \hat{u}_T(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2d_1 \hat{v}_T + d_2(r^2 + 2\hat{u}_T^2) \\ \hat{v}_T(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + d_1(r^2 + 2\hat{v}_T^2) + 2d_2 \hat{u}_T) \end{bmatrix} \tag{2.41}$$

where $r = \sqrt{(\hat{u}_T - c_x)^2 + (\hat{v}_T - c_y)^2}$, parameters k_i with $i = 1, 2, 3$ are constants accounting for radial distortion and d_i with $i = 1, 2$ accounts for tangential distortion. A number of other more complex models, e.g. [15], are also available.

Each sensor pixel p_T directly estimates the radial distance $\hat{r}_T$ from its corre-

sponding scene point P_T as

$$\hat{r}_T = \sqrt{\hat{x}_T^2 + \hat{y}_T^2 + \hat{z}_T^2} = \left|\left| \left[\hat{x}_T^2, \hat{y}_T^2, \hat{z}_T^2\right]^T \right|\right|_2 . \tag{2.42}$$

From radial distance $\hat{r}_T$ measured at pixel p_T with distorted coordinates $\hat{\mathbf{p}}_T = [\hat{u}_T, \hat{v}_T]^T$ the $3D$ coordinates of $\mathbf{P}_T$ can be computed according to the following steps:

1. Given the lens distortion parameters, estimate the non-distorted $2D$ coordinates $\mathbf{p}_T = [u_T, v_T]^T = \Psi^{-1}(\hat{\mathbf{p}}_T)$, where $\Psi^{-1}(\cdot)$ is the inverse of $\Psi(\cdot)$.

2. The estimated depth value $\hat{z}_T$ can be computed from (2.39) and (2.42) as

$$\hat{z}_T = \frac{\hat{r}_T}{\left|\left| \mathbf{K}_T^{-1} \left[u_T, v_T, 1\right]^T \right|\right|_2} \tag{2.43}$$

   where $\mathbf{K}_T^{-1}$ is the inverse of $\mathbf{K}_T$.

3. The estimated coordinates values $\hat{x}_T$ and $\hat{y}_T$ can be computed by inverting (2.39), i.e., as

$$\begin{bmatrix} \hat{x}_T \\ \hat{y}_T \\ \hat{z}_T \end{bmatrix} = \mathbf{K}_T^{-1} \begin{bmatrix} u_T \\ v_T \\ 1 \end{bmatrix} \hat{z}_T. \tag{2.44}$$

Since amplitude $\hat{A}$, intensity $\hat{B}$ and depth $\hat{z}_T$ are estimated at each sensor pixel, ToF depth cameras handle them in matricial structures, and return them as 2D maps or depth maps. Therefore a ToF depth camera can in principle provide as output the following types of data:

- an *amplitude map* $\hat{A}_T$, i.e., a matrix obtained by juxtaposing the amplitudes estimated at all the ToF sensor pixels. It is defined on lattice Λ_T and its values, expressed in volts [V], belong to the pixel non-saturation interval. Map $\hat{A}_T$ can be modeled as realization of a random field $\mathcal{A}_T$ defined on Λ_T, with values expressed in volts [V] in the pixel non-saturation interval;

- an *intensity map* $\hat{B}_T$, i.e., a matrix obtained by juxtaposing the intensity values estimated at all the ToF sensor pixels. It is defined on lattice Λ_T and its values, expressed in volts [V], belong to the pixel non-saturation interval. Map $\hat{B}_T$ can be modeled as the realization of a random field $\mathcal{B}_T$ defined on Λ_T, with values (expressed in volts [V]) in the pixel non-saturation interval;

- a *depth map* $\hat{Z}_T$, i.e, a matrix obtained by juxtaposing the depth values estimated at all the ToF sensor pixels. It is defined on lattice Λ_T and its values, expressed in [mm], belong to interval $[0, r_{MAX} = c/(2f_m))$. Map $\hat{Z}_T$ can be considered as the realization of a random field $\mathcal{Z}_T$ defined on Λ_T, with values (expressed in [mm]) in $[0, r_{MAX})$.

Some of the commercial ToF depth cameras do not expose all the intermediate data, however, in addition to the depth map $\hat{Z}_T$, all the ToF depth cameras provide at least either the intensity map $\hat{A}_T$ or the amplitude map $\hat{B}_T$.

2.3.3 Practical implementation issues

The previous sections highlight the conceptual steps needed to measure the distances of a scene surface by a ToF depth camera, but they do not consider a number of issues which must be taken into account in practice. The major contributions to imperfection are described next.

Phase wrapping

The first fundamental limitation of ToF sensors comes from the fact that the estimate of $\widehat{\Delta\varphi}$ is obtained from an arctangent function, which has co-domain $[-\pi/2, \pi/2]$. Therefore, the estimates of $\widehat{\Delta\varphi}$ can only assume values in this interval. Since the physical delays entering the phase shift $\Delta\varphi$ can only be positive, it is possible to shift the arctan$(\cdot)$ co-domain to $[0, \pi]$ to have a larger interval available for $\widehat{\Delta\varphi}$. Moreover, the usage of atan2$(\cdot, \cdot)$ allows one to extend the co-domain to $[0, 2\pi]$. From (2.25) it is immediate to see that the estimated distances are within range $[0, c/(2f_m)]$. If for instance $f_m = 30$ [MHz], the interval of measurable distances is $[0 - 5]$ [m].

Since $\widehat{\Delta\varphi}$ is estimated modulo 2π from (2.25) and the distances greater than $c/(2f_m)$ correspond to $\widehat{\Delta\varphi}$ greater than 2π, they are incorrectly estimated. In practice the distance returned by (2.25) corresponds to the remainder of the division between the actual $\Delta\varphi$ and 2π, multiplied by $c/(2f_m)$, a well-known phenomenon called *phase wrapping* since it refers to a periodic wrapping around 2π of phase values $\widehat{\Delta\varphi}$. Clearly, if f_m increases, the interval of measurable distances becomes smaller, and vice-versa. Possible solutions to overcome phase wrapping include the use of of multiple modulation frequencies or of non-sinusoidal wave-forms (e.g., chirp wave-forms), e.g. Kinect$^{\text{TM}}$ v2. Other works such as [17] use only one frequency and the amplitude image.

Harmonic distortion

The generation of perfect sinusoids of the needed frequency is not straightforward. In practice, actual sinusoids are obtained as low-pass filtered versions of square waveforms emitted by LEDs [10]. Moreover, the sampling of the received signal is not ideal, but it takes finite time intervals Δ_S. The combination of these two factors introduces an harmonic distortion in the estimated phase-shift $\widehat{\Delta\varphi}$ and consequently in the estimated distance $\hat{\rho}$. Such harmonic distortion leads to a systematic offset component dependent on the measured distance. A metrological characterization of this harmonic distortion effect is reported in [53] and [109]. Harmonic distortion offset exhibits an oscillatory behavior which can be up to some tens of centimeters, clearly reducing the accuracy of distance measurements. This systematic offset can usually be fixed using a look-up-table to compensate for this offset, estimated with a calibration procedure.

Material reflectivity

The amount of reflected light strongly depends on the reflectivity of the target object, which leads to erroneous distance calculation. Materials can be divided into two categories according to their reflection coefficient in the IR band of the emitters.

For diffusely reflecting materials such as dull surfaces, the reflectivity coefficient has values in the range $[0, 1]$, where 0 means that all incoming light is absorbed or transmitted, and 1 that all the incident rays are reflected. The reference value of 1 is given by the case of a perfect Lambertian reflector, where all the light is back-scattered with an intensity distribution that is independent of the observation angle.

For directed reflecting materials such as glossy surfaces, the reflection coefficient might be even ≥ 1 for specific angles at which the light is directly reflected into the sensor. Camera measurements for such directed reflections might saturate, causing errors in distance estimation. The same problem may be encountered in the opposite condition, that is when the reflected ray points away from the camera, preventing the sensor from capturing enough signal intensity to deliver valid measurements.

Authors of [109] proposed a method to correct the distance non linearities as well as the integration time offsets for different reflectivity. They found that a difference in amplitude as well as measured distance between the black and white targets are attributed to the differences in reflectivity. In [41] it is shown that the systematic error in depth measurement can be reduced using the object's intensity. Depth and inverse amplitude $1/A$ are compared, discovering that these two measures are

correlated.

Angle of emission and incidence

Quality of the received signal also depends on the angle at which the light is emitted, reflected and received. In [109], the model to correct distance nonlinearity also considers a term related to the angle of emitted and received rays. The lenses in front of the emitter do not distribute the light uniformly, resulting in a strong vignetting effect. The lenses at the receiver also cause a light fall-off more or less severe depending on the optical design and construction of the lens. Moreover, materials with different reflection coefficient impact the measurement characteristics of the camera in different ways. The best measure is given by the case of Lambertian reflection of a 90° incident and received ray. Since a prior knowledge about objects material composition and orientation in the scene is not available, modeling this inaccuracy is a quite difficult task. The only information that is always known is the angle associated to the emitted light rays. A general characterization of this phenomenon is available in the datasheet of the actual camera. MESA SR4000, for example, defines two measurement regions (Figure 2.33): the first region involves central pixels while the second one involves pixels far away from the center point. A larger error is associated to the outer region, and this is due to the larger angle of the emitted light rays. This indication of the measurement accuracy is also known as repeatability and is characterized by the spread σ of the measurement around the mean value.

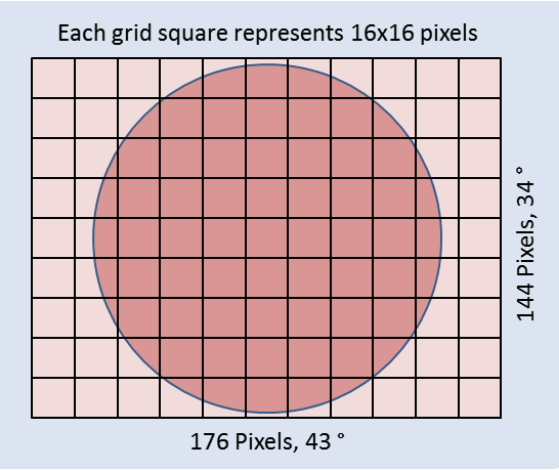


Figure 2.33: Measurement regions with different repeatability for the MESA SR4000. Darker region in the center has higher repeatability.

Photon-shot noise

Because of the light-collecting nature of the receiver, the acquired samples c_R^0, c_R^1, c_R^2 and c_R^3 are affected by photon-shot noise, due to dark electron current and photon-generated electron current as reported in [11]. Dark electron current can be reduced by lowering the sensor temperature or by technological improvements. Photon-generated electron current, due to light-collection, cannot be completely eliminated. Photon-shot noise is statistically characterized by a Poisson distribution. Since $\hat{A}$, $\hat{B}$, $\widehat{\Delta\varphi}$ and $\hat{\rho}$ are computed directly from the corrupted samples c_R^0, c_R^1, c_R^2 and c_R^3, their noise distribution can be computed by propagating the Poisson distribution through (2.24-2.25). A detailed analysis of error and noise propagations can be found in [81].

Quite remarkably, the probability density function of the noise affecting estimate $\hat{\rho}$, in the case of sinusoidal modulation, according to [11] and [81] can be approximated by a Gaussian[4] with standard deviation

$$\sigma_\rho = \frac{c}{4\pi f_{mod}} \frac{1}{SNR} = \frac{c}{4\pi f_{mod}} \frac{\sqrt{B/2}}{A} \tag{2.45}$$

in which the SNR of the signal is measured as

$$SNR = \frac{A}{\sqrt{B/2}}. \tag{2.46}$$

Standard deviation (2.45) determines the precision, or repeatability, of the distance measurement and is directly related to f_m, A and B. In particular, if the received signal amplitude A increases, the precision improves. This suggests that the precision improves as the measured distance decreases and the reflectivity of the measured scene point increases.

Equation (2.45) also indicates that as the interference intensity B of the received signal increases, precision worsens. This means that precision improves as the scene background IR illumination decreases. Note that B may increase because of two factors: an increment of the received signal amplitude A or an increment of the background illumination. While in the second case the precision gets worse, in the first case there is an overall precision improvement, given the squared root dependence of B in (2.45). Finally, observe that B cannot be 0 as it depends on carrier intensity A.

[4] An explicit expression of the Gaussian probability density function mean is not given in [11, 81]. However, the model of [81] provides implicit information about the mean which is a function of both A and B, and contributes to the distance measurement offset. For calibration purposes the non-zero mean effect can be included in the harmonic distortion.

If modulation frequency f_m increases, precision improves. The modulation frequency is an important parameter for ToF sensors, since f_m is also related to phase wrapping and the maximum measurable distance. In fact, if f_m increases, the measurement precision improves, while the maximum measurable distance decreases (and vice-versa). Therefore, there is a trade-off between distance precision and range. Since f_m is generally a tunable parameter, it can be adapted to the distance precision and range requirements of the specific application.

Saturation and motion blur

Averaging over multiple periods is effective against noise but it introduces dangerous side effects, such as *saturation* and *motion blur*. Saturation occurs when the received quantity of photons exceeds the maximum quantity that the receiver can collect. This phenomenon is particularly notable in presence of external IR illumination (e.g., direct solar illumination) or in the case of highly reflective objects (e.g., specular surfaces). The longer the integration time, the higher the quantity of collected photons and the more likely the possibility of saturation. Specific solutions have been developed to avoid saturation, i.e., in-pixel background light suppression and automatic integration time setting [10, 11].

Motion blur is another important phenomenon accompanying time averaging. It is caused by imaged objects moving during integration time, as in the case of standard cameras. Time intervals of the order of $1 - 100$ $[ms]$ make object movement likely unless the scene is perfectly still. In the case of moving objects, the samples entering (2.49) do not concern a specific fixed scene point at subsequent instants as is expected in theory, but different scene points at subsequent instants, causing distance measurement artifacts. The longer the integration time, the higher the likelihood of motion blur (but better the distance measurement precision). Integration time is another parameter to set according to the characteristics of the specific application.

Multipath error

In the model presented in Section 2.3.1, we assumed that the signal $s_E(t)$ transmitted from the source is reflected back by the scene in a single ray. In a more realistic scenario, the signal transmitted from the source will encounter multiple objects in the environment that produce reflected, diffracted, or scattered copies of the transmitted signal, as shown in Figure 2.34. These additional copies of the transmitted signal, called *multipath* signal components, are summed together at the receiver, leading to a combination of the incoming light paths and thus to a wrong

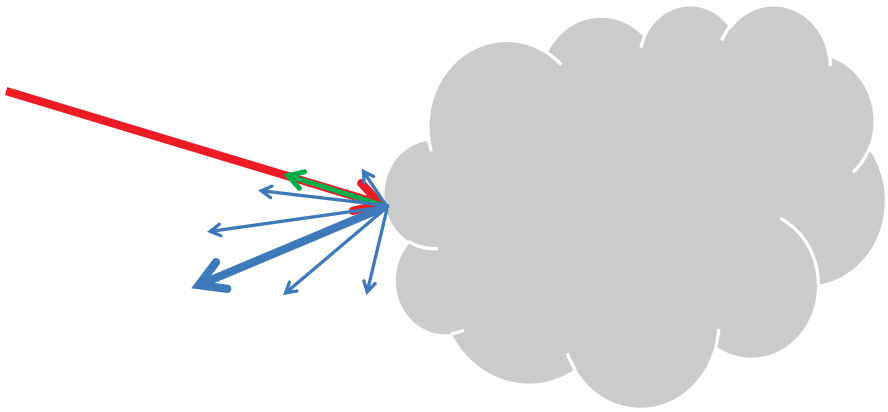


Figure 2.34: Scattering effect.

distance estimation. Since the radial distance of a scene point P from the ToF camera is computed from the time-length of the shortest path between P and the camera, the multipath effect leads to over-estimation of the scene points' distances.

Figure 2.34 shows an optical ray (red) incident to a non-specular surface reflected in multiple directions (green and blue). The ideal propagation scenario with co-positioned emitters and receivers, considers only the presence of the green ray of Figure 2.34, i.e., the ray back reflected in the direction of the incident ray and disregards the presence of the other (blue) rays. In practical situations, however, the presence of the other rays may not always be negligible. In particular, the ray specular to the incident ray direction with respect to the surface normal at the incident point (thick blue ray) generally is the reflected ray with greatest radiometric power.

All the reflected (blue) rays may first hit other scene points and then travel back to the ToF sensor, therefore affecting distance measurements of other scene points. For instance, as shown in Figure 2.35, an emitted ray (red) may be first reflected by a point surface A with a scattering effect. One of the scattered rays (orange) may then be reflected by another scene point B and travel back to the ToF sensor. The distance measured by the sensor pixel relative to B is therefore a combination of two paths, namely path to ToF camera-B-ToF camera and path ToF camera-A-B-ToF camera. The coefficients of such a combination depend on the optical amplitude of the respective rays.

There are multiple sources of the multipath effect, and most of them are related to the properties of the scene. In general, all materials reflect incoming light in all directions, so a normal scene will produce indirect reflections everywhere and each camera pixel will measure the superposition of infinite waves. Fortunately, most of the time the indirect reflections are order of magnitudes weaker than direct

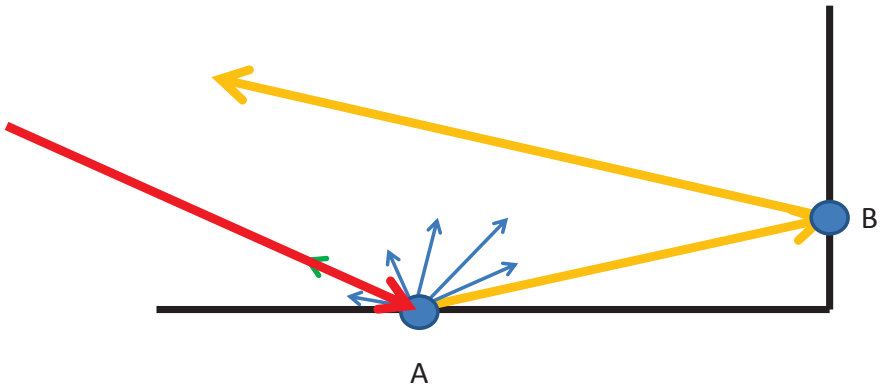


Figure 2.35: Multipath phenomenon: the incident ray (red) is reflected in multiple directions (blue and orange rays) by the surface at point A. The orange ray reaches then B and travels back to the ToF sensor.

reflections, and the camera can easily resolve the reflected signal. When the object is highly reflective or transparent, however, the camera pixel will receive multiple signals with different phase and attenuation, leading to incorrect measurements. One of the most visible effects of multipath interference is relative to concave corners, which often appear rounded in ToF depth maps. This happens because each point belonging to one side of the corner will receive light reflected by any point of the other side and reflects parts of it towards the camera, resulting in an over-estimation of the distances of the points on the corner surface. The interference of different waves is not necessarily related only to the scene; the optics and other internal components of the ToF camera may scatter and reflect small amounts of the received signal as well.

To model the multipath error, in the case of sinusoidal modulation we can rewrite (2.22), by considering N incoming waves

$$s_R(t) = \sum_{i=1}^{N} (a)_i sin(2\pi f_m t + \Delta\varphi_i) + B_i. \tag{2.47}$$

Since the sum of sinusoidal functions is still a sinusoid, and it is difficult to estimate the contribution of each independent ray, in practice only two components are considered: a first direct signal, and a second indirect signal that takes into account all the additional reflections. With these assumptions, (2.47) can be rewritten as

$$s_R(t) = [Asin(2\pi f_m t + \Delta\varphi) + B] + [A_{MP} sin(2\pi f_m t + \Delta\varphi_m) + B_m] \tag{2.48}$$

where the second component takes into account the multipath signal.

In the literature, there are several works that propose solutions to multipath

interference, e.g., [114] reviews current state of the art techniques used to correct for this error. When a single frequency is used in the presence of multipath, the relationship among modulation frequency, measured phase, and amplitude is nonlinear. By exploiting modulation frequency diversity, it is possible to iteratively reconstruct the original signal using two or more modulation frequencies [31, 57, 6], and find a closed-form solution by using four modulation frequencies [37]. Another solution to address multipath is to use coded waves [10, 52] where the signal in (2.18) is replaced by a binary sequence or more particular custom codes, and the received signal is estimated by means of sparse deconvolution. The general idea is that the combination of pure sinusoidal signals is still a sinusoid and this creates a unicity problem at the receiver. The use of different signals instead allows one to recognize when the received signal has been corrupted by the scene.

Let us finally observe that ToF cameras can be considered as special Multiple-Input and Multiple-Output (MIMO) communication systems, where the emitters array is the input array and the lock-in matrix of the ToF receiver sensor the output array. In principle, this framework would allow one to approach multipath as customarily done in communication systems. However, the number of input and output channels of a ToF camera vastly exceed the complexity of the MIMO systems used in telecommunications, in which the number of inputs and outputs rarely exceed the 10s of units. Even though the current multipath analysis methods used for MIMO systems cannot be applied to ToF depth cameras, the application of communications systems techniques for characterizing ToF depth cameras operations and improving their performance appears an attractive possibility.

Flying pixels

Another problem similar to multipath is the *flying pixel* effect. Since the pixels of any imaging sensor don't have infinitesimal size but some physical size, as shown in Figure 2.36, each pixel receives the radiation reflected from all the points of the corresponding small scene patch and the relative distance information. If the scene patch is a flat region with constant reflectivity, the approximation that there is a single scene point associated with the specific pixel does not introduce any artifacts. However, if the scene patch corresponds to a discontinuity of the scene reflectivity, the values of $\hat{A}_T(p_T)$ and $\hat{B}_T(p_T)$ estimated by the correspondent pixel p_T average its different reflectivity values.

A worse effect occurs if the scene patch associated with p_T corresponds to a depth discontinuity. In this case, assume that a portion of the scene patch is at a closer distance, called z_{near}, and another portion at further distance, called z_{far}. The

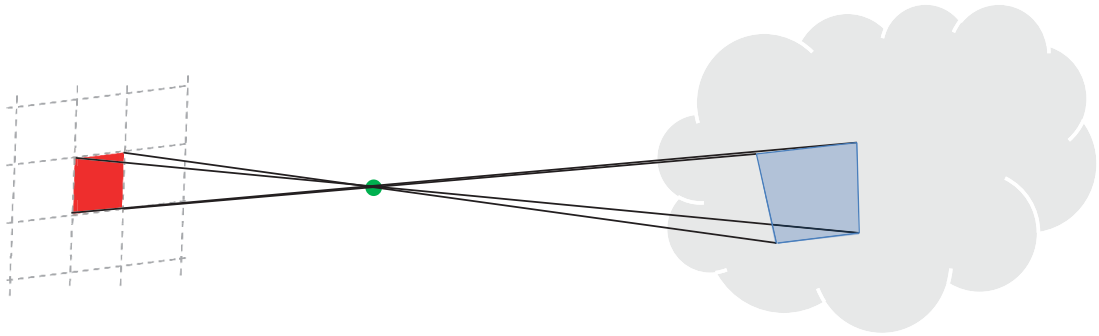


Figure 2.36: Finite size scene patch (right) associated with a pixel (left) of any imaging sensor.

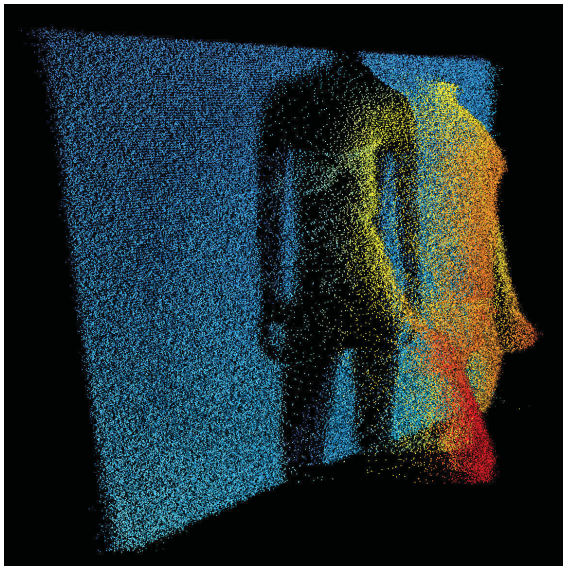


Figure 2.37: An example of flying pixels at the depth edge between a person and the wall.

resulting depth estimate $\hat{Z}_T(p_T)$ is a convex combination of z_{near} and z_{far}, where the combination coefficients depend on the percentage of area at z_{near} and at z_{far} respectively reflected on p_T. The presence of flying pixels leads to severe depth estimation artifacts, as shown by the example of Figure 2.37, where foreground and background are blended together.

The most effective solutions to this problem tackle the detection and eventual correction of these points as shown in [95]. More recent works aim at providing a confidence value for each pixel, based on analysis of intensity and amplitude of the received signal [94].

Other noise sources

There are several other noise sources affecting the distance measurements of ToF sensors, notably *flicker* and *kTC noise*. The receiver amplifier introduces a Gaussian-distributed thermal noise component. Since the amplified signal is quantized to be digitally treated, this introduces another error source, customarily modeled as random noise. Quantization noise can be controlled by the number of used bits and it is typically neglectable with respect to the other noise sources. All noise sources, except photon-shot noise, may be reduced by adopting high quality components. A comprehensive description of various ToF noise sources can be found in [65, 11, 81, 10].

Averaging distance measurements over several modulation periods T_m is a classical provision to mitigate the noise effects. If N is the number of periods, in the case of sinusoidal modulation the estimated values $\hat{A}$, $\hat{B}$ and $\widehat{\Delta\varphi}$ become

$$\begin{aligned}
\hat{A} &= \frac{\sqrt{\left(\frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n} - \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+2}\right)^2 + \left(\frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+1} - \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+3}\right)^2}}{2} \\
\hat{B} &= \frac{\sum_{n=0}^{N-1} c_R^{4n} + \sum_{n=0}^{N-1} c_R^{4n+1} + \sum_{n=0}^{N-1} c_R^{4n+2} + \sum_{n=0}^{N-1} c_R^{4n+3}}{4N} \\
\widehat{\Delta\varphi} &= \operatorname{atan2}\left(\frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n} - \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+2}, \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+1} - \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+3}\right)
\end{aligned} \tag{2.49}$$

where

$$\begin{aligned}
c_R^{4n} &= c_R\left(\frac{4n}{F_S}\right) \\
c_R^{4n+1} &= c_R\left(\frac{4n+1}{F_S}\right) \\
c_R^{4n+2} &= c_R\left(\frac{4n+2}{F_S}\right) \\
c_R^{4n+3} &= c_R\left(\frac{4n+3}{F_S}\right).
\end{aligned} \tag{2.50}$$

This provision reduces but does not completely eliminate noise effects. The averaging intervals used in practice are typically between 1 $[ms]$ and 100 $[ms]$. For instance, when $f_m = 30\ MHz$, where the modulating sinusoid period is $33.3 \times 10^{-9}[s]$, the averaging intervals concern a number of modulating sinusoid periods from 3×10^4 to 3×10^6. The averaging interval length is generally called *integration time*, and its proper tuning is extremely important in ToF measurements. Long integration

times lead to repeatable, reliable ToF distance measurements at the expense of motion blur effects.

2.3.4 Comparison of ToF depth cameras

Kinect™ v2

With the introduction of the Xbox One gaming device in November 2013, Microsoft also presented a second version of the Kinect™, called Kinect™ v2 for simplicity. Kinect™ v2 with respect to Kinect™ v1 is a completely different product, since it employs a ToF depth camera while Kinect™ v1 employed a structured light depth camera. As with the Kinect™ v1, the Kinect™ v2 includes the depth sensing element, a video camera and an array of microphones.

A high level description of the operating principles of Kinect™ v2 can be found in [77], while more details are given in Microsoft patents. The ToF depth camera was developed from former products by Canesta, a ToF depth camera producer acquired by Microsoft in 2010. Some innovative details introduced to overcome some of the issues of Section 2.3.3 are worth noting. Kinect™ v2 is able to acquire a 512×424 [pxl] depth map (the largest resolution achieved by a ToF depth camera at the time of writing this thesis) at 50 [fps] with a depth estimation error typically smaller than 1% of the measured distances. The emitted light is modulated by a square wave (see Section 2.3.1) instead of a sinusoid as in most previous ToF depth cameras. The receiver ToF sensor is a differential pixels array, i.e., each pixel has two outputs and the incoming photons contribute to one or the other according to the current state of a clock signal. The clock signal is the same square wave used for the modulation of the emitter. Let us denote with U the signal corresponding to the photons arriving when the clock is high and L the signal corresponding to the low state of the clock. The difference $(U - L)$ depends on both the amount of returning light and on the time it takes to come back, and allows one to estimate the time lag used to compute the distance. Square wave modulation helps against harmonic distortion issues.

Another well-known critical trade-off is between precision and the maximum measurable range given by (2.25) and (2.45), i.e., by increasing f_m the measurement precision increases but the measurable range gets smaller. Kinect™ v2 deals with this issue by using multiple modulation frequencies which are 17, 80 and 120 [MHz]. Multiple modulation frequencies allow one to extend the acquisition range, overcoming limits due to phase wrapping. Indeed, the correct measurement can be disambiguated by identifying the measurement values consistent with respect to all three modulation frequencies, as visually exemplified by Figure 2.38.

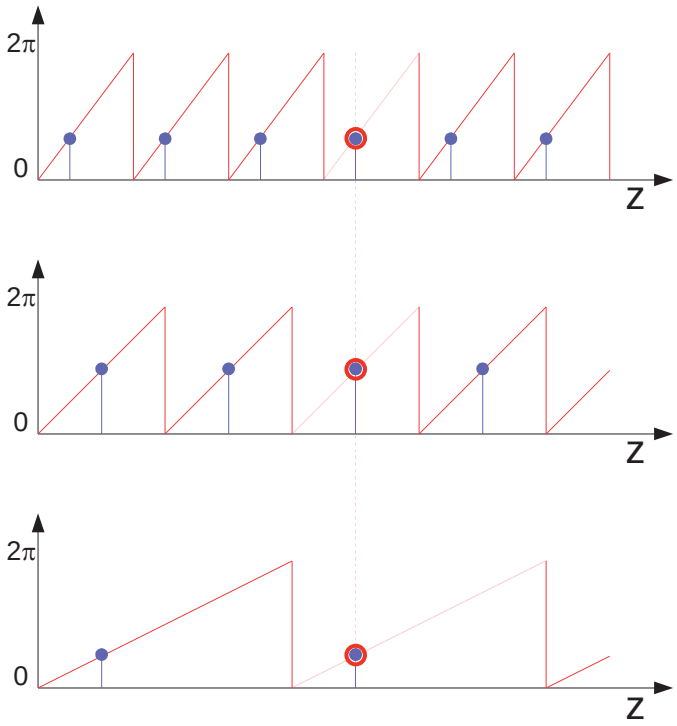


Figure 2.38: Disambiguation of phase wrapping errors by multiple modulation frequencies. Notice how by looking at each single plot there are ambiguities on the actual distance value (represented by the multiple dots), but by comparing all the plots it is possible to disambiguate the various measurements.

Another improvement introduced by the Kinect^{TM} v2 is the capability of simultaneously acquiring two images with different shutter times, namely 100 $[\mu s]$ and 1000 $[\mu s]$ and selecting whichever one leads to the best pixel by pixel result; this is made possible by the non-destructive pixel reading feature of its sensor.

MESA ToF depth cameras

MESA Imaging, which was founded in 2006, is a spin-off of the Swiss Center for Electronics and Microtechnology (CSEM). It was one of the first companies to commercialize ToF depth cameras and its main product, the SwissRanger, is now in its 4th generation. Differently from the Kinect^{TM} v2, the SwissRanger is an industrial grade product developed for measurement applications rather than for interfaces or gaming. The SwissRanger uses CWAM with sinusoidal modulation according to the principles presented in this chapter. For a detailed description see [49]. The modulation frequency can be chosen among 14.5, 15, 15.5, 29, 30 and 31 [MHz]. Typical SwissRanger operation is in the $[0, 5]$ [m] range with nominal accuracy of 10 [mm] at 2 [m] and in the longer range of $[0, 10]$ [m] with lower accuracy of 15 [mm] at 2 [m]. Notably, one can use up to three SwissRanger

(a) (b) (c)

Figure 2.39: Mesa Imaging ToF depth cameras: (a) MESA Imaging SR3000^{TM}; (b) MESA Imaging SR4000^{TM}; MESA Imaging SR4500^{TM}.

cameras together without interference issues.

PMD devices

PMD technologies is another early ToF depth camera producer. This spin-off of the Center for Sensor Systems (ZESS) of the University of Siegen (Germany) was founded in 2002. In 2005, it launched the Efector camera, its first commercial product. The company then introduced the Efector 3D in 2008, a 64×48 pixels ToF depth camera developed for industrial use. In 2009 the company launched the CamCube, a 204×204 pixels ToF depth camera characterized by the highest resolution ToF sensor until the introduction of Kinect^{TM} v2. The initial focus of the company was on industrial applications but recently it entered other fields, including automotive, gesture recognition and consumer electronics (it is taking part in Google's Project Tango). Recent products include the CamBoard, a 200×200 single board 3D ToF depth camera, and the PhotonICs 19k-S3 chip for camera developers and system integrators. PMD depth cameras operate according to the CWAM modulation principles introduced earlier in this section.

ToF depth cameras based on SoftKinetic technology

SoftKinetic is a Belgian company, founded in 2007 and acquired by Sony in 2015, which has produced two generations of ToF depth cameras, the DS311 and the newer DS325. These ToF depth cameras are based on the company's patented CMOS pixel technology, called Current Assisted Photonic Demodulation (CAPD). This technique uses a driving current to move electrons towards two different detecting

(a) (b) (c)

Figure 2.40: PMD ToF depth cameras: (a) PMD PhotonICs; (b) PMD CamCube; (c) PMD CamBoard pico.

(a) (b) (c)

Figure 2.41: SoftKinetic ToF depth cameras: (a) SoftKinetic DS311; (b) SoftKinetic DS325; (c) Creative Senz3D^{TM}.

junctions as a result of an alternating current, with a result similar to the differential pixels of the Kinect^{TM} v2. SoftKinetic adopts CWAM modulation with a square wave modulating signal similar to the Kinect^{TM} v2. The transmitter uses a laser illuminator and the receiver has a resolution of 320×240 pixels. The DS325 can acquire data at up to 60 [fps] within a nominal range 150 [mm] - 1000 [mm], thus being particularly suited for hand gesture recognition applications and computer interfaces. It is possible to acquire data up to 4 [m], but the range increase decreases the resolution or the frame rate. The accuracy is about 14 [mm] at 1 [m] and the built-in calibration is not very accurate, making the DS325 more suited to gesture recognition applications than to 3D reconstruction purposes. The camera is also sold with different form factors like in the newer DS525, and from other vendors like Creative under the Senz3D^{TM} name. The DS325 and the Senz3D^{TM} essentially share same hardware with a different case and exterior appearance.

Chapter 3

Depth data fusion with confidence measures

As discussed in the previous chapters, data provided by depth cameras have several limitations. In particular, data from structured light or ToF depth cameras are usually noisier and at a lower resolution than data from standard cameras, because depth camera technology is still far from the maturity of standard camera technology. This fact suggests that combining active depth cameras with standard cameras may lead to more accurate 3D representations than those provided by depth cameras alone, and that the higher resolution of standard cameras may be exploited to obtain higher resolution depth maps. Furthermore, depth cameras can only provide scene geometry information, while many applications, e.g., 3D reconstruction, scene segmentation, and matting, also need the color information of the scene.

ToF depth cameras are generally characterized by low spatial resolution, and structured light depth cameras by poor edge localization, as seen in Chapter 2. Therefore, a depth camera alone is not well suited for high-resolution and precise 3D geometry estimation, especially near depth discontinuities. If such information is desired, as is usually the case, it is worth coupling a depth camera with a standard camera. In addition, it is possible to consider an acquisition system made by a depth camera and a stereo system where both sub-systems are able to provide depth information and take advantage of the depth measurements' redundancy. This solution can also reduce occlusion artifacts between color and 3D geometry information and it may also be beneficial, for example, in 3D video production and 3D reconstruction. In synthesis, the quality of acquired depth data can be improved by combining high resolution color data, particularly in critical situations typical of each family of depth cameras.

There are several ways of combining standard cameras and depth data and this chapter will focus on a framework for the fusion of depth data produced by a ToF camera and stereo vision system. In the proposed approach, depth data acquired by the ToF camera are upsampled by an ad-hoc algorithm based on image segmentation and bilateral filtering. In parallel a dense disparity map is obtained using the Semi-Global Matching stereo algorithm. Reliable confidence measures are extracted for both the ToF and stereo depth data. In particular, ToF confidence also accounts for the mixed-pixel effect and the stereo confidence accounts for the relationship between the pointwise matching costs and the cost obtained by the semi-global optimization. Finally, the two depth maps are synergically fused by enforcing the local consistency of depth data accounting for the confidence of the two data sources at each location. Experimental results show that the proposed method produces accurate high resolution depth maps and outperforms the compared fusion algorithms [73]. In this chapter all the building blocks of the proposed approach will be further analyzed.

To motivate the benefit of combining depth data from multiple sensors, we show in Figure 3.1 an example of point clouds produced with a stereo system, a ToF depth camera and the proposed fusion approach. In the three pictures, annotations with the same shape match portions of the scene, while colors indicate whether the portion is correct in a given point cloud. Starting from the rectangular shape framing a portion of the scene with a depth discontinuity, it is clear that while stereo data do not present particular errors, ToF data have a substantial error originating from the flying pixels problem typical of ToF depth cameras. The circular region instead, framing a portion of the scene with a planar region, highlights the issues of stereo system in regions with periodic pattern, being the planar surface the cover of a book with a repetitive texture. ToF data instead seem to be correct in that region, as it does not represent a problematic case for ToF depth cameras. In both the highlighted regions, the proposed fusion approach instead is able to provide the correct depth value.

3.1 Related Works

Matricial ToF range cameras have been the subject of several recent studies, e.g., [42, 91, 118, 87, 53, 39]. In particular, [53] focuses on the various error sources that influence range measurements while [39] presents a qualitative analysis of the influence of scene reflectance on the acquired data.

Stereo vision systems have also been the subject of a significant amount of

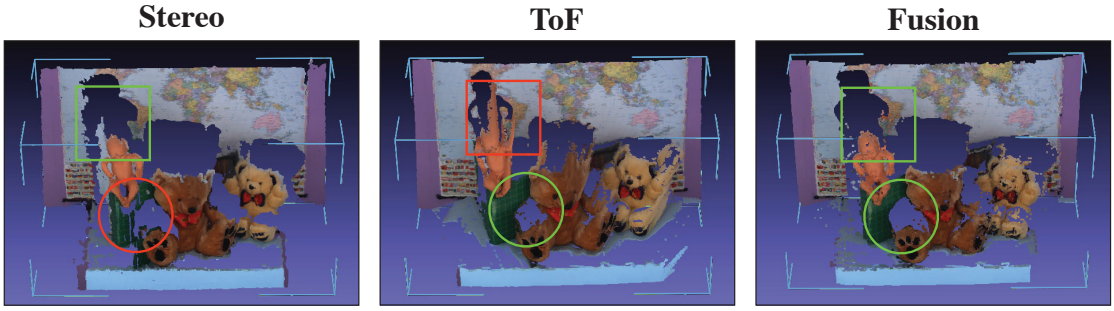


Figure 3.1: Examples of point cloud generated by a depth map from a stereo camera, a ToF depth camera, and from the proposed approach to fuse data.

research, and a recent review on this topic can be found in [107]. The accuracy of stereo vision depth estimation strongly depends on the framed scene's characteristics and the algorithm used to compute the depth map, and a critical issue is the estimation of the confidence associated with the data. Various metrics have been proposed for this task and a complete review can be found in [47].

The idea of combining ToF sensors with standard cameras has been used in several recent works, and recent surveys of this field can be found in [83, 118]. Some work focused on the combination of a ToF camera with a single color camera [25, 117, 116, 101, 35, 27]. An approach based on bilateral filtering is proposed in [117] and extended in [116]. The approach of [35] instead exploits an edge-preserving scheme to interpolate the depth data produced by the ToF sensor. The recent approach of [101] also accounts for the confidence measure of ToF data. The combination of a ToF camera and a stereo camera is more interesting, because in this case both subsystems can produce depth data [61, 39, 34, 56]. A method based on a probabilistic formulation is presented in [21], where the final depth-map is recovered by performing a ML local optimization to increase the accuracy of the depth measurements from the ToF and stereo vision system. This approach has been extended in [20] with a more refined measurement model which also accounts for the mixed pixel effect and a global optimization scheme based on a MAP-MRF framework. The method proposed in [120, 122] is also based on a MAP-MRF Bayesian formulation, and a belief propagation based algorithm is used to optimize a global energy function. An automatic way to set the weights of the ToF and stereo measurements is presented in [121]. Another recent method [82] uses a variational approach to combine the two devices. The approach of [22], instead, uses a locally consistent framework [74] to combine the measurements of the ToF sensor with the data acquired by the color cameras, but the two contributions are equally weighted in the fusion process. This critical issue has been solved in the proposed approach by extending the LC framework. Finally the approach of [33] computes

the depth information by hierarchically solving a set of local energy minimization problems. The setup with multiple cameras is not limited to stereo cameras only, [80] presented a framework that uses multiple confidence measures to select among multiple disparity hypotheses generated by a trinocular stereo system.

3.2 Proposed Method

We consider an acquisition system made of a ToF camera and a stereo vision system. The goal of the proposed method is to provide a dense confidence map for each depth map computed by the two sensors, then use this information to fuse the two depth maps into a more accurate description of the 3D scene. The approach assumes that the two acquisition systems have been jointly calibrated and we consider the left camera of the stereo vision system to be the reference system. The proposed algorithm is divided into three different steps:

1. The low resolution depth measurements of the ToF camera are reprojected into the lattice associated with the left camera and a high resolution depth-map is computed by interpolating the ToF data. The confidence map of ToF depth data is estimated using the method described in Section 3.3.

2. A high resolution depth map is computed by applying a stereo vision algorithm on the images acquired by the stereo pair. The confidence map for stereo depth data is estimated as described in Section 3.4.

3. The depth measurements obtained by the upsampled ToF data and the stereo vision algorithm are fused together by means of an extended version of the LC technique [74] using the confidence measures from the previous steps.

3.3 ToF confidence estimation

Before describing how to compute the confidence map for ToF data we briefly describe how we obtain a high resolution depth map from ToF data, from the point of view of the left camera of the stereo vision system. Since stereo data typically have higher resolutions than those of ToF cameras, the projection of ToF data on the lattice associated with the left color camera produces a set of sparse depth measurements that need to be interpolated. To obtain an accurate high resolution map, especially in proximity of edges, we exploit the method of [22], combining cross bilateral filtering with the help of segmentation. First, all the 3D points acquired by the ToF camera are projected onto the left camera lattice Λ_l, obtaining

a set of samples $p_i, i = 1, ..., N$ that does not include samples that are occluded from the left camera point of view. The color image acquired by the left camera is then segmented using mean-shift clustering [16], obtaining a segmentation map used to guide an extended bilateral filter developed for the interpolation of the p_i samples. The output of the interpolation method is a disparity map defined on the left camera lattice Λ_l. Since the fusion algorithm works in the disparity space, the interpolated depth map is converted into a disparity map with the well known relationship $d = bf/z$, where d and z are disparity and depth values, b is the baseline of the stereo system and f is the focal length of the rectified stereo camera.

As reported in Chapter 2, the reliability of the ToF measurements is affected by several issues, e.g., the reflectivity of the acquired surface, the measured distance, multi-path issues or mixed pixels in proximity of edges, and thus is very different for each different sample. A proper fusion algorithm requires a reliable confidence measure for each pixel. We propose a novel model for the confidence estimation of ToF measurements, using both radiometric and geometric properties of the scene. As described in the rest of this section, our model is based on two main clues that can be separately captured by two metrics. The first one, P_{AI}, considers the relationship between amplitude and intensity of the ToF signal, while the second one, P_{LV}, accounts for the local depth variance. The two confidence maps P_{AI} and P_{LV} consider independent geometric and photometric properties of the scene, therefore, the overall ToF confidence map P_T is obtained by multiplying the two confidence maps together

$$P_T = P_{AI} P_{LV}. \tag{3.1}$$

Equation (3.1) implicitly assumes the independence of P_{AI} and P_{LV}. Given the different nature of the two confidence maps, their independence, although it is not proved here, is a reasonable assumption.

3.3.1 Confidence from amplitude and intensity values

ToF cameras provide both the amplitude and the intensity of the received signal for each pixel. The amplitude of the received signal depends on various aspects, but the two most relevant are the reflectivity characteristics of the acquired surfaces and the distance of the scene samples from the camera. Intensity also depends on these two aspects, but is additionally affected by the ambient illumination in the wavelength range of the camera. A confidence measure directly using the distance of objects in the scene could be considered, but distance strongly affects the amplitude, and thus the proposed measure already implicitly takes the distance

into account. The received amplitude strongly affects the accuracy of the measures and a higher amplitude leads to a better signal-to-noise ratio and thus to more accurate measurements [91]. Equation (2.45), reported here for convenience with the notation of this chapter, shows that the distribution of the ToF pixel noise can be approximated by a Gaussian with standard deviation

$$\sigma_z = \frac{c}{4\pi f_{mod}} \frac{1}{SNR} = \frac{c}{4\pi f_{mod}} \frac{\sqrt{B/2}}{A} \tag{3.2}$$

where f_{mod} is the IR frequency of the signal sent by the ToF emitters, A is the amplitude value at the considered pixel, B is the intensity value at the same location and c is the speed of light. Note that since the data fusion is performed on the upsampled disparity map, the confidence maps must be of the same resolution, but amplitude and intensity images are at the same low resolution of the ToF depth map. To solve this issue, each pixel $\mathbf{p}_L$ in the left color image is first back-projected to the 3D world and then projected to the corresponding pixel coordinates in the ToF lattice $\mathbf{p}_L^{TOF}$.

From (3.2) it can be observed that when amplitude A increases, precision improves, since the standard deviation decreases, while when intensity I increases, the precision decreases. Intensity I depends on two factors: the received signal amplitude A and the background illumination. An increase in the amplitude leads to an overall precision improvement given the squared root dependence with respect to I in (3.2), while in the second case precision decreases since A is not affected.

Before mapping σ_z to the confidence values, it is important to notice that the proposed fusion scheme works on the disparity domain, while the measurement standard deviation (3.2) refers to depth measurements. For a given distance z, if a certain depth error Δ_z around z is considered, the corresponding disparity error Δ_d also depends on the distance z, due to the inverse proportionality between depth and disparity. If σ_z is the standard deviation of the depth error, the corresponding standard deviation σ_d of the disparity measurement can be computed as:

$$2\sigma_d = |d_1 - d_2| = \frac{bf}{z - \sigma_z} - \frac{bf}{z + \sigma_z} = bf \frac{2\sigma_z}{z^2 - \sigma_z^2} \quad \Rightarrow \quad \sigma_d = bf \ \frac{\sigma_z}{z^2 - \sigma_z^2} \tag{3.3}$$

where b is the baseline of the stereo system and f is the focal length of the camera. Equation (3.3) provides the corresponding standard deviation of the noise in the disparity space for a given depth value. The standard deviation of the measurements in the disparity space is also affected by the mean value of the measurement itself, unlike the standard deviation of the depth measurement.

To map the standard deviation of the disparity measurements to the confidence

values, we define two thresholds computed experimentally over multiple measurements. The first is $\sigma_{min} = 0.5$, corresponding to the standard deviation of a bright object at the minimum measurable distance of 0.5 [m], while the second is $\sigma_{max} = 3$, corresponding to the case of a dark object at the maximum measurable distance of 5 [m] with the SR4000 sensor used in the experimental results dataset. If a different sensor is employed, the two thresholds can be updated by considering these two boundary conditions. Then, we assume that values smaller than σ_{min} correspond to the maximum confidence value, i.e., $P_{AI} = 1$, values bigger than σ_{max} have $P_{AI} = 0$ while values in the interval $[\sigma_{min}, \sigma_{max}]$ are linearly mapped to the confidence range $[0, 1]$, i.e.:

$$P_{AI} = \begin{cases} 1 & if\ \sigma_d \leq \sigma_{min} \\ \frac{\sigma_{max} - \sigma_d}{\sigma_{max} - \sigma_{min}} & if\ \sigma_{min} < \sigma_d < \sigma_{max} \\ 0 & if\ \sigma_d \geq \sigma_{max} \end{cases} \tag{3.4}$$

3.3.2 Confidence from local variance

One of the main limitations of (3.2) is that it does not take into account the effect of the finite size of ToF sensor pixels, i.e., the mixed pixel effect [20]. To account for this issue we introduce another term in the proposed confidence model. When the scene area associated with a pixel includes two regions at different depths, e.g. close to discontinuities, the resulting estimated depth measure is a convex combination of the two depth values. For this reason, it is reasonable to associate a low confidence to these regions. The mixed pixel effect leads to convex combinations of depth values but this is not true for the multipath effect. These considerations do not affect the design of the ToF confidence since the LV metric just assumes that pixels in depth discontinuities are less reliable. If pixel p_i^{TOF} in the low resolution lattice of the ToF camera is associated with a scene area crossed by a discontinuity, some of the pixels p_j^{TOF} in the 8-neighborhood $\mathcal{N}(p_i^{TOF})$ of p_i^{TOF} belong to points at a closer distance, and some other pixels to points at a farther distance. Following this intuition the mean absolute difference of the points in $\mathcal{N}(p_i^{TOF})$ has been used to compute the second confidence term, i.e.:

$$D_l^{TOF} = \frac{1}{|\mathcal{N}(p_i^{TOF})|} \sum_{j \in \mathcal{N}(p_i^{TOF})} |z_i - z_j| \tag{3.5}$$

where $\left|\mathcal{N}(p_i^{TOF})\right|$ is the cardinality of the considered neighborhood, in this case equal to 8, and z_i and z_j are the depth values associated with pixels p_i^{TOF} and p_j^{TOF}, respectively. We use the mean absolute difference instead of the variance to avoid assigning very high values to edge regions due to the quadratic dependence of

the variance with respect to the local differences. For this term we used the depth values and not the disparity ones because the same depth difference would lead to different effects on the confidence depending if close or far points are considered. This computation is performed for every pixel with a valid depth value. Notice that some p_j^{TOF} considered in an 8-connected patch may not have a valid value. To obtain a reliable map, a constant value $K_d = T_h$ has been used in the summation (3.5) in place of $|z_i - z_j|$ for the pixels p_j^{TOF} without a valid depth value. To obtain the confidence information D_l on the left camera lattice, samples p_i on this lattice are projected on the ToF camera lattice and the corresponding confidence value is selected after a bilinear interpolation.

Points with high local variance are associated with discontinuities, therefore, low confidence should be assigned to them. Where the local variance is close to zero, the confidence should be higher. To compute the confidence term we normalize D_l to the $[0, 1]$ interval by defining a maximum valid absolute difference $T_h = 0.3$ corresponding to 30 cm and assigning higher likelihood values to the regions with lower local variability:

$$P_{LV} = \begin{cases} 1 - \frac{D_l}{T_h} & if \quad D_l < T_h \\ 0 & if \quad D_l \geq T_h \end{cases} \tag{3.6}$$

3.4 Stereo confidence estimation

The considered setup includes two calibrated color cameras, therefore an additional high resolution disparity map D_s can be inferred by stereo vision. The data fusion algorithm presented in the next section is independent of the choice of the stereo vision algorithm, however, for our experiments we used the Semi-Global Matching (SGM) algorithm [46] reviewed in Chapter 2. The goal of this algorithm is to perform a 1D disparity optimization on multiple paths. Such an optimization minimizes on each path an energy term made of point-wise or aggregated matching costs C^l and a regularization term. We used the pointwise Birchfield-Tomasi metric over color data and 8 paths for the optimization, with window size of 7×7, $P_1 = 20$ and $P_2 = 100$. The energy terms are summed up obtaining a global cost function C^g that usually presents a very sharp peak at the minimum cost's location. In the rest of the section we analyze how the relationship between local cost C^l and global cost C^g can provide an effective confidence measure. The combination of local and global cost functions was not used by any other confidence measure proposed in the literature.

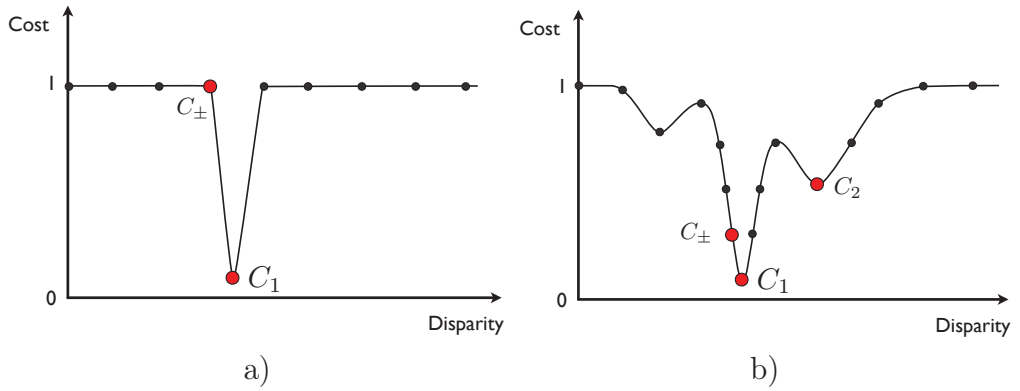


Figure 3.2: Examples of cost curves: a) ideal cost function; b) ambiguous cost function.

3.4.1 Analysis of cost function

The cost value assigned to a disparity hypothesis d for a pixel (u, v) will be denoted as $C(d)$, without the explicit pixel coordinates label. Moreover, without loss of generality, the cost range is considered normalized to the unit interval, i.e.

$$0 \leq C(d) \leq 1 \tag{3.7}$$

The ideal cost curve for a pixel, as a function of disparity, is shown in Figure 3.2 a). The ideal cost is 0 for the correct disparity and 1 for all the others. It is reasonable to believe that if for a pixel the cost curve exhibits a behavior like the one of Figure 3.2 b), the disparity estimation will be more ambiguous. This is due to the presence of multiple local minima or multiple adjacent disparities with similar costs, making exact localization of the global minimum hard and often uncertain.

Figure 3.2 b) also shows the terminology used to denote some point of interest. The minimum cost for a pixel is denoted by C_1 and the corresponding disparity value by d_1, i.e.

$$C_1 = C(d_1) = \min C(d) \tag{3.8}$$

where disparity d has sub-pixel resolution. The second smallest cost value which occurs at disparity d_2 is C_2. For the selection of C_2, disparity values that are too close to d_1 (i.e., $|d_2 - d_1| \leq 1$) are excluded to avoid suboptimal local minima too close to d_1.

The reliability of the disparity map is affected by the content of the acquired images, in particular by the texture of the scene. Uniform regions are usually the most challenging since it is difficult to estimate corresponding image points reliably. Global (or semi-global) methods tackle this problem by propagating neighbor values enforcing a smoothness constraint at the cost of a higher uncertainty in the disparity

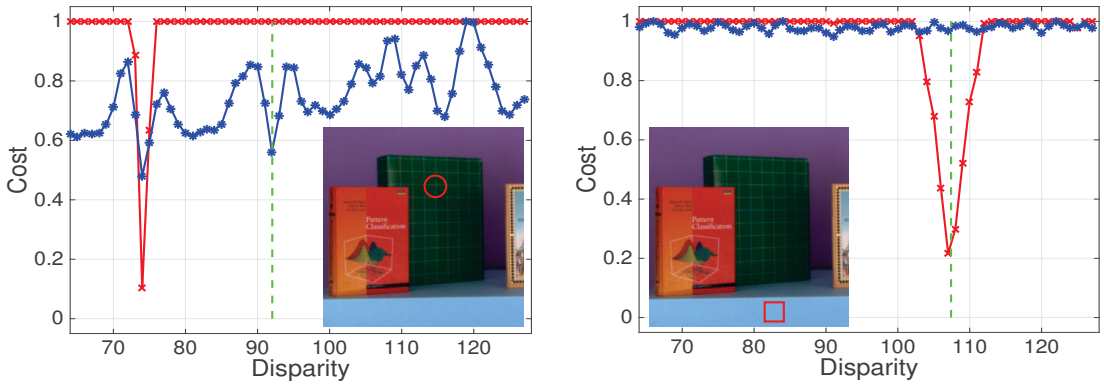


Figure 3.3: Comparison of local (*blue*) and global (*red*) costs: a) Cost functions of a repetitive pattern; b) Cost functions of a uniform region. The green line represent the ground truth disparity value.

assignments. The globally optimized cost function typically has a very sharp peak, often resulting from the enforced smoothness constraint, corresponding to the propagated value even in areas where the data are not reliable. Current stereo vision confidence estimation approaches analyzing the cost function [47] do not account for the impact of global optimizations performed by most recent stereo vision methods. We believe that an optimal confidence metric can only be obtained by analyzing both cost functions. In the proposed approach this issue is handled by introducing a novel confidence measure considering both the local cost function C^l and the globally optimized one C^g.

In our analysis, at each pixel location for each disparity hypothesis d, we consider the point-wise local cost $C^l(d)$ and the global cost from the SGM algorithm $C^g(d)$, both scaled to the interval $[0, 1]$. Ideally the cost function should have a very well-defined minimum corresponding to the correct depth value but, as expected, in many practical situation this is not the case. Figure 3.3 shows two points in the scene where the confidence should be low. In Figure 3.3a the region surrounding the selected point has a periodic pattern and in Figure 3.3b the region surrounding the selected point has a uniform color. However, the global cost function has a sharp peak and conventional confidence measures based only on global cost analysis would assign a high confidence to these pixels.

The proposed stereo confidence metric P_S is the combination of multiple clues, depending both on the properties of the local cost function and on the relationship between local and global costs. In particular it is defined as the product of three factors:

$$P_S = \frac{\Delta C^l}{C_1^l}\left(1 - \frac{\min\{\Delta d^l, \gamma\}}{\gamma}\right)\left(1 - \frac{\min\{\Delta d^{lg}, \gamma\}}{\gamma}\right) \tag{3.9}$$

where $\Delta C^l = C_2^l - C_1^l$ is the difference between the second and first minimum local cost, $\Delta d^l = |d_2^l - d_1^l|$ is the corresponding absolute difference between the second and first minimum local cost locations, $\Delta d^{lg} = |d_1^l - d_1^g|$ is the absolute difference between the local and global minimum cost locations and γ is a normalization factor.

The first term accounts for the robustness of the match, both the cost difference and the value of the minimum cost are important, as the presence of a single strong minimum with an associated small cost are usually sufficient conditions for a good match. However, in the case of multiple strong matches, the first term still provides a high score, e.g., in regions of the scene with a periodic pattern (see Figure 3.3b).

The second term is a truncated measure of the distance between the first two cost peaks. It discriminates potentially bad matches due to the presence of multiple local minima. If the two minimum values are close enough, the associated confidence measure should provide a high value since the global optimization is likely to propagate the correct value and to provide a good disparity estimation.

So far only the local cost has been considered so the last term accounts for the relationship between the local and global cost functions, scaling the overall confidence measure depending on the level of agreement between the local and global minimum locations. If the two minimum locations coincide, there is a very high likelihood that the estimated disparity value is correct, while on the other hand, if they are too far apart the global optimization may have produced incorrect disparity estimations, e.g. due to the propagation of disparity values in textureless regions.

The constant γ controls the weight of the two terms and sets the maximum distance of the two minimum locations, after which the estimated value is considered unreliable. In our experiments we set $\gamma = 10$. Finally, if a local algorithm is used to estimate the disparity map, the same confidence measure can be used by considering only the first two terms.

3.5 Extended local consistency framework

Given the disparity maps and the confidence information for the ToF camera and the stereo vision system, the final step combines the multiple depth hypotheses available for each point by means of a technique that guarantees a locally consistent disparity map. Our method extends the LC technique [74], originally proposed for

stereo matching, to deal with the two disparity hypotheses provided by our setup and modifies the original formulation to take advantage of the confidence measures to weight the contributions of the two sensors.

In the original LC method, given a disparity map provided by a stereo algorithm, the overall accuracy is improved by propagating, within an active support centered on each point f of the initial disparity map, the plausibility $\mathcal{P}_{f,g}(d)$ of the same disparity assignment made for the central point by other points g within the active support. Specifically, the clues deployed by LC to propagate the plausibility of disparity hypothesis d are the color and spatial consistency of the considered pixels:

$$\mathcal{P}_{f,g}(d) = e^{-\frac{\Delta_{f,g}}{\gamma_s}} \cdot e^{-\frac{\Delta^{\psi}_{f,g}}{\gamma_c}} \cdot e^{-\frac{\Delta^{\psi}_{f',g'}}{\gamma_c}} \cdot e^{-\frac{\Delta^{\omega}_{g,g'}}{\gamma_t}} \tag{3.10}$$

where f, g and f', g' refer to points in the left and right image respectively, $\Delta_{f,g}$ is the Euclidean distance between f and g and accounts for spacial proximity, $\Delta^{\psi}_{f,g}$ (and similarly $\Delta^{\psi}_{f',g'}$) and $\Delta^{\omega}_{g,g'}$ encode color similarity:

$$\Delta^{\psi}_{f,g} = \sqrt{\sum_{c \in R,G,B} \left(I_c(f) - I_c(g)\right)^2}\,, \qquad \Delta^{\omega}_{g,g'} = \sqrt{\sum_{c \in R,G,B} \left(I_c(g) - I_c(g')\right)^2} \tag{3.11}$$

where $I_c(p)$ encodes the color intensity of point p. Parameters γ_s, γ_c and γ_t control the behavior of the distribution (see [74] for a detailed description). For the experimental results these parameters have been set to $\gamma_s = 8$, $\gamma_c = \gamma_t = 4$. The overall plausibility $\Omega_f(d)$ of each disparity hypothesis is given by the aggregated plausibility for the same disparity hypothesis d propagated from neighboring points within the active support $\mathcal{A}$ according to

$$\Omega_f(d) = \sum_{g \in \mathcal{A}} \mathcal{P}_{f,g}(d). \tag{3.12}$$

This aggregation is computed both on the left and the right image and then the results are normalized over the plausibility at all disparity levels, obtaining $\Omega_f(d)^L$ and $\Omega_f(d)^R$ respectively. To obtain a robust disparity estimation, after these calculations, the cross-validation of the accumulated plausibility is computed

$$\Omega_f(d)^{LR} = \Omega_f(d)^L \;\cdot\; \Omega_f(-d)^R \tag{3.13}$$

and then the final disparity D_f is obtained as

$$D_f = \underset{d}{\operatorname{argmin}} \ \Omega_f(d)^{LR} \tag{3.14}$$

Disparity propagation allows to overcome many of the problems typical of local approaches, however the presence of wrong disparity hypothesis, e.g. due to occlusions, may perturb the aggregated plausibility. A left-right consistency check is useful to limit such undesired effects. The effectiveness of this algorithm is also visible if a sparse disparity map is used as input. Disparity propagation acts like an interpolating function, assigning valid disparity also to regions without original values. It is worth to notice that the plausibility function defined on color and range information ensures robustness to this approach at the cost of having multiple parameters that require an empirical estimation.

The LC approach has been extended in [22] to allow the fusion of two different disparity maps by adding a term to (3.12) to account for the two sensors

$$\Omega'_f(d) = \sum_{g \in \mathcal{A}} \Big(\delta_T(g) \mathcal{P}_{f,g,T}(d) \ + \ \delta_S(g) \mathcal{P}_{f,g,S}(d) \Big) \tag{3.15}$$

where $\mathcal{P}_{f,g,T}(d)$ is the plausibility for ToF data and $\mathcal{P}_{f,g,S}(d)$ for stereo data. According to (3.15), for each point of the input image there can be 0, 1 or 2 disparity hypotheses, depending on which sensor provides a valid measurement. The functions $\delta_T(g)$ and $\delta_S(g)$ return 1 if the measurement of the relative sensor if available at location g. Although [22] produces reasonable results, it has the fundamental limitation that gives exactly the same relevance to the information from the two sources without taking into account their reliability.

We propose an extension to (3.15) to account for the reliability of the measurements of ToF and stereo described in Sections 3.3 and 3.4. To exploit these additional clues we multiply the plausibility for an additional factor that depends on the reliability of the considered depth acquisition system, computed for each sensor in the considered point, as follows:

$$\Omega''_f(d) = \sum_{g \in \mathcal{A}} \Big(P_T(g) \mathcal{P}_{f,g,T}(d) \ + \ P_S(g) \mathcal{P}_{f,g,S}(d) \Big) \tag{3.16}$$

where $P_T(g)$ and $P_S(g)$ are the confidence maps for ToF and stereo data respectively.

The proposed fusion approach implicitly addresses the complementary nature of the two sensors. In fact, in uniformly textured regions, where the stereo range sensing is quite inaccurate, the algorithm should propagate mostly the plausibility originated by the ToF camera. Conversely, in regions where the ToF camera is

less reliable (e.g. dark objects), the propagation of plausibility concerned with the stereo disparity hypothesis should be more influential. Without the two confidence terms of (3.16), all the clues are propagated with the same weight, as in (3.15). In this case an erroneous disparity hypothesis from a sensor could negatively impact the overall result. Therefore, the introduction of reliability measures allows us to automatically discriminate between the two disparity hypotheses provided by the two sensors and thus improve the fusion results.

The adoption of the proposed model for the new plausibility is also supported by the nature of the confidence maps, that can be interpreted as the probability that the corresponding disparity measure is correct. A confidence of 0 means that the disparity value is not reliable and in this case such hypothesis should not be propagated. The opposite case is when the confidence is 1, meaning a high likelihood that the associated disparity is correct. All the intermediate values will contribute as weighting factors. This definition is also coherent when a disparity value is not available, for example due to occlusions: the associated confidence is 0 and propagation does not occur at all. An interesting observation on the effectiveness of this framework is that Equation (3.16) can be extended to deal with more than two input disparity maps, simply adding other plausibility terms for the new disparity clues and an associated confidence measures. Other families of sensors can be included as well, by simply devising proper confidence measures.

Both ToF and stereo disparity maps are computed at sub-pixel resolution, but the original LC algorithm [74] only produces integer disparities, therefore we propose an additional extension to handle sub-pixel precision. We consider a number of disparity bins equals to the number of disparities to be evaluated multiplied by the inverse of the desired sub-pixel resolution (i.e., we multiply by 2 if the resolution is 0.5). Then, at every step the algorithm propagates the plausibility of a certain disparity by contributing to the closest bin. With this strategy, the computation time remains the same as in the original approach [74, 75] and only the final winner-takes-all step is slightly affected.

3.6 Experimental Results

To evaluate the performance of the proposed algorithm, we used the dataset provided in [20], that at the time of the writing is the largest available collection of real world ToF and stereo data with ground truth. This dataset contains 5 different scenes acquired by a trinocular setup made of a Mesa SR4000 ToF range camera and two Basler video cameras. The ToF sensor has a resolution of 176×144 pixels while

the color cameras one is 1032×778 pixels, which is also the output resolution of the proposed method. Calibration and ground truth information are also provided with the dataset. The different scenes contain objects with different characteristics that allow one to evaluate the proposed algorithm in challenging situations, including depth discontinuities, materials with different reflectivity and objects with both textured and un-textured surfaces. Scene 1 and 2 present piecewise smooth surfaces, ideal for the implicit assumption of stereo matching, but also reflective materials and textureless regions. Scene 3, 4 and 5 are more complex and also include curved and fuzzy surfaces.

3.6.1 Evaluation of confidence metrics

Figure 3.4 shows the confidence maps that are used in the fusion process. For each scene of the dataset we show the confidence maps associated to ToF and stereo data. The first column shows the reference color images, the second, third and fourth columns show the confidence maps associated to the ToF and the last column shows the confidence maps of stereo data. For ToF data, we show the confidence from amplitude and intensity values P_{AI}, the confidence from local variance P_{LV} and their product $P_T = P_{AI}P_{LV}$. The last column shows the confidence of the stereo system, i.e., P_S. As shown in the color map below, dark values correspond to low confidence and bright values correspond to higher confidence values.

Starting from the ToF confidence, the amplitude and intensity related term tends to assign lower confidence to the upper part of the table that is almost parallel to the emitted rays. Therefore the amplitude of the received signal is low, thus reducing the precision. This term also assigns a smaller confidence to farther regions, reflecting another well known issue of ToF data. ToF confidence is low for dark objects but measurement accuracy depends on the reflectivity of the surface at ToF IR wavelengths and the reflectivity can be different for objects looking similar to the human eye (i.e., the black plastic finger in scene 5 reflects more IR light than the bear's feet). In addition, the four corners of the image also have lower confidence, in agreement with the lower quality of the signal in those regions, affected by higher distortion and attenuation. Local variance instead, as expected, contributes by assigning a lower confidence value to points near depth discontinuities.

Stereo confidence has on average a lower value, consistently with the fact that stereo data is less accurate (see Table 3.1) but locally reflects the texture of the scene, providing high values in correspondence of high frequency content, and low values in regions with uniform texture (the blue table) or periodic pattern (e.g., the

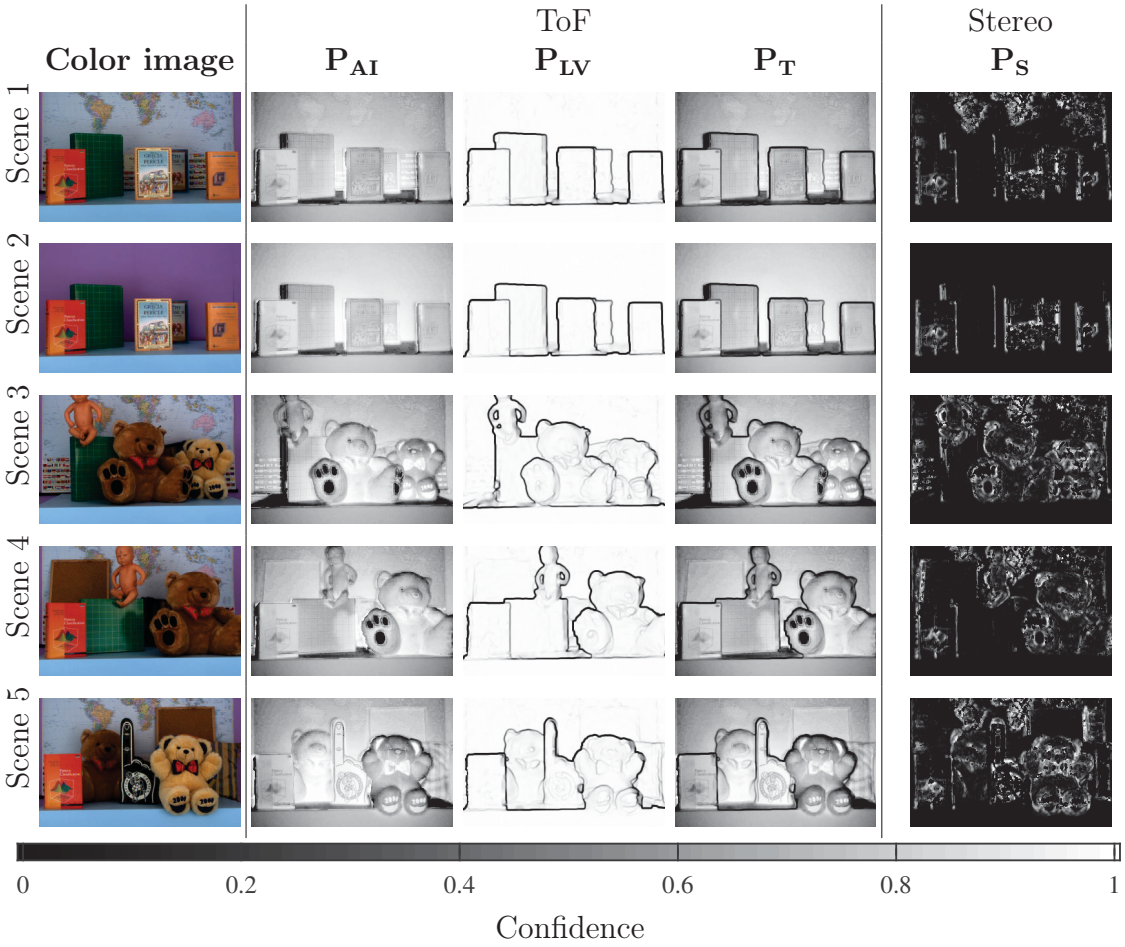


Figure 3.4: Confidence maps for ToF and stereo disparity. Brighter areas correspond to higher confidence values, while darker pixels are less confident.

green book). Scene 2 compared to scene 1 clearly shows the effect that textured and untextured regions have in the confidence map. The map in the first scene is able to provide enough texture to consider reliable the depth measurements in that region. In the orange book on the left side, stereo confidence assigns high values only to the edges and to the logo in the cover, correctly penalizing regions with uniform texture. The teddy bear in scene 3, 4 and 5 has more texture than the table or the books and the relative confidence value is higher overall.

To evaluate the effectiveness of the proposed confidence metrics we show in Figure 3.5 the scattergram for ToF and stereo confidence metrics. The meaning of this plot is to show the correlation between errors and confidence. Pixels with low confidence should be associated to larger errors. Both in ToF and stereo metrics the trend of the number of wrong pixels is decreasing as the confidence increase. The stereo scattergram has a large number of pixels with low confidence corresponding to large uniform regions on the table and the background. The proposed metrics have been developed targeting the fusion of data from the two sensors, with particular

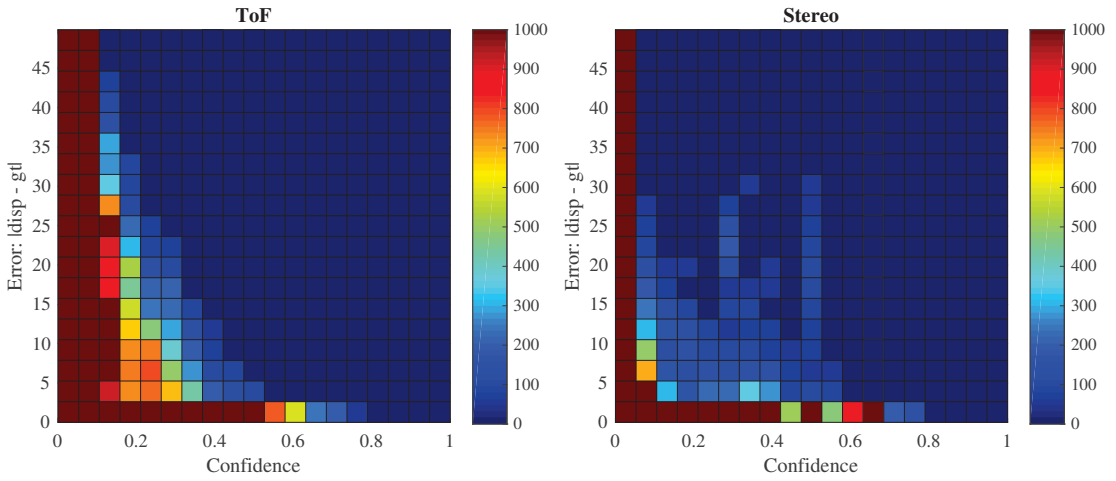


Figure 3.5: Scattergram relating errors and confidence. Colors represent the number of pixels at each location.

attention to the nature of the depth data. Confidence values close to 0 are associated to stereo data in textureless regions, even if the estimated depth is correct. This justifies the high number of pixels with low error and low confidence. Although the proposed confidence metric for stereo systems is not as good as top performing stereo metrics evaluated in [47] in terms of AUC (e.g., PKRN), it performs better when used in our fusion framework. Indeed our goal is to propose a good confidence metric for the stereo system in the context of data fusion, where low confidence should be assigned to pixels belonging to textureless surfaces propagated by the global optimization, since ToF data are more reliable there. This feature is well captured by the proposed metric, but not by conventional stereo confidence metrics.

3.6.2 Evaluation of disparity maps

The disparity maps of the proposed framework are compared with the estimates of ToF and stereo system alone and with state of the art methods of [22], [117], [120] and [20]. The method of [20] has been computed from the ToF viewpoint at a different resolution, therefore we reprojected the data on the left camera viewpoint to compare it with other methods. We re-implemented the methods of [117] and [120] following the description in the papers. Figure 3.6 shows the estimated disparity maps and results of other methods as well.

Figure 3.7 shows the absolute difference between the output disparity maps and the ground truth, i.e., $|D_i - D_{GT}|$, where D_i is the considered disparity map i for the evaluation, and D_{GT} is the ground truth. Figure 3.8 shows the squared error map between the output disparity maps and the ground truth, i.e., $(D_i - D_{GT})^2$.

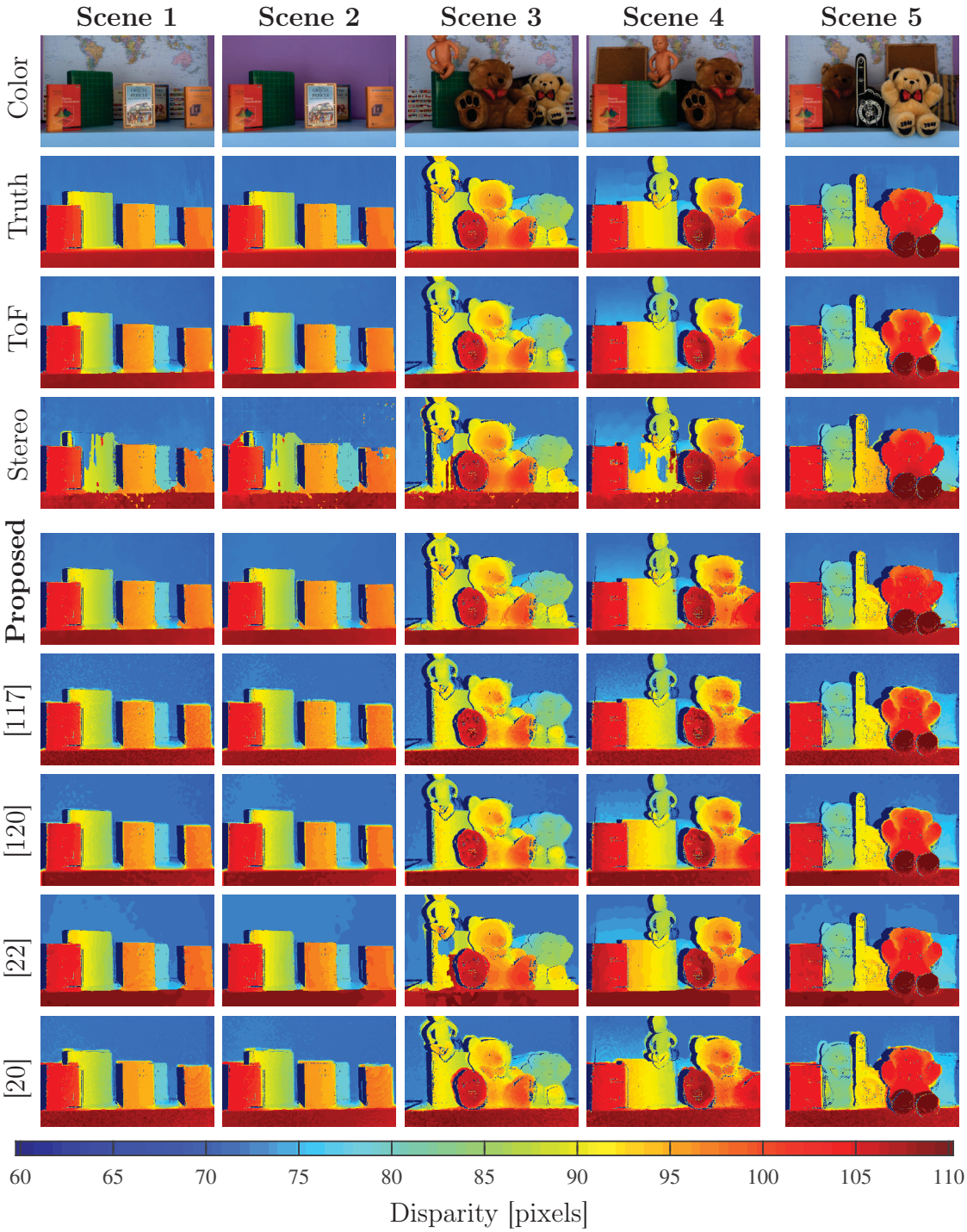


Figure 3.6: Disparity maps of the proposed fusion framework and comparison with other methods.

With respect to the absolute difference, the MSE penalizes more larger errors and less errors smaller than 1 pixel.

The average mean squared error (MSE) has been computed considering all the five scenes, and the results are reported in Table 3.1. Since the output of the fusion process is a disparity map, we computed the error in the disparity space. For

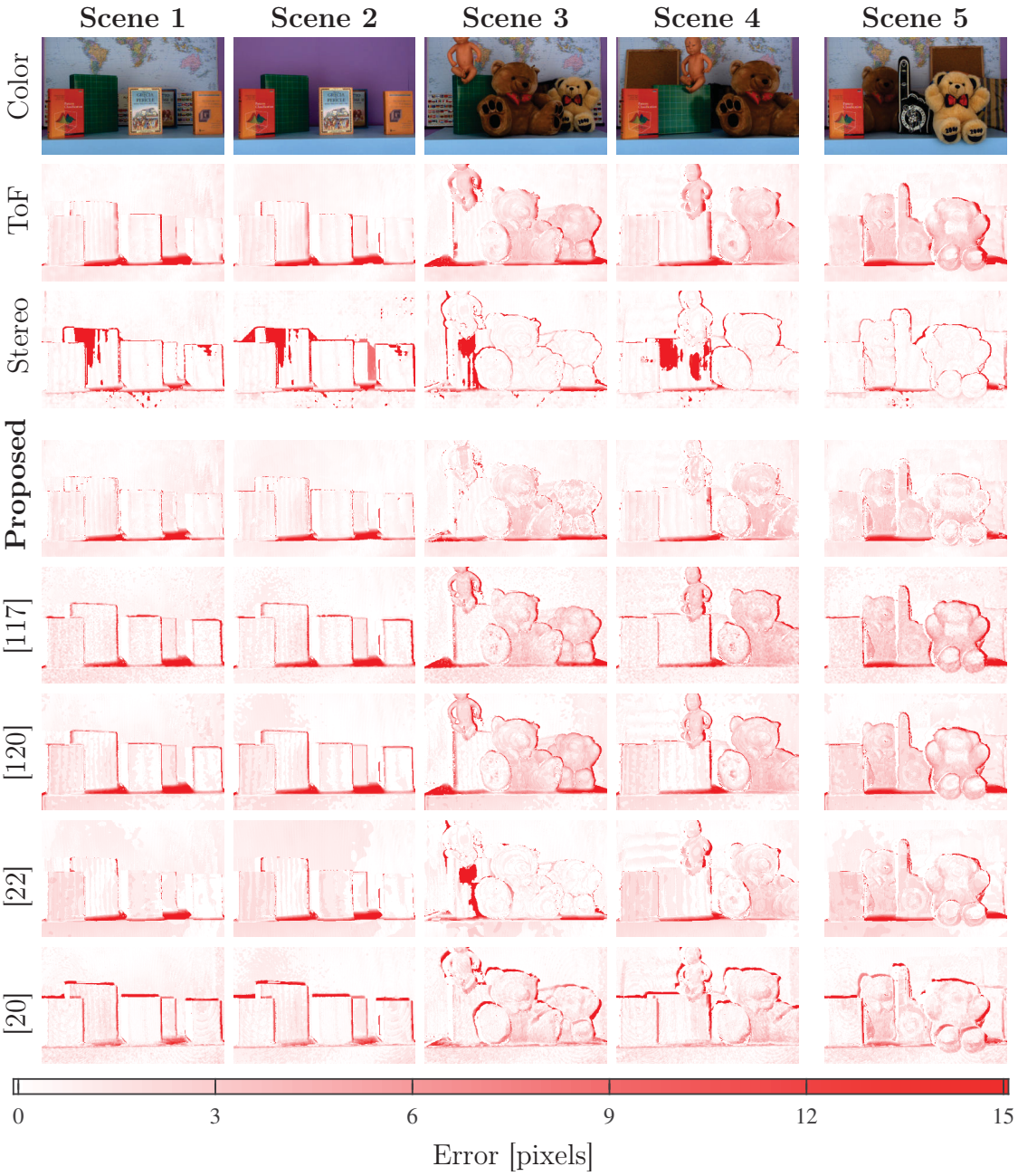


Figure 3.7: Squared error map of the proposed approach and other methods.

a fair comparison, we computed the error on the same set of valid pixels for all the methods, where a pixel is considered valid if it has a valid disparity value in all the compared maps and in the ground truth data. We also consider the ideal case obtained by selecting for each pixel the ToF or stereo disparity closer to the ground truth. From the MSE values on the five different scenes, it is noticeable how the proposed framework provides more accurate results than the interpolated ToF data and the stereo measurements alone. Even if stereo data have typically lower accuracy the proposed method is still able to improve the results of the ToF

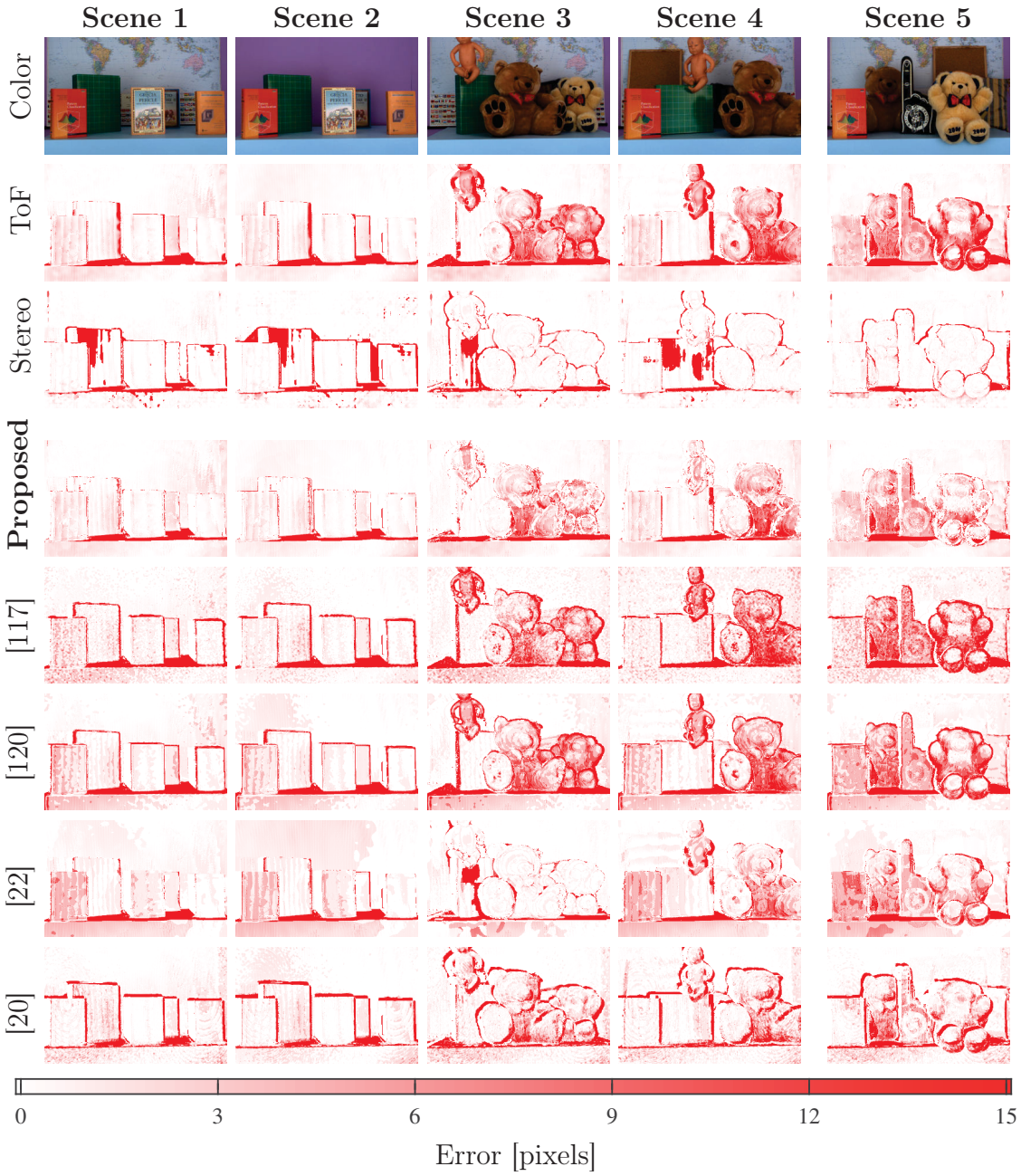


Figure 3.8: Squared error map of the proposed approach and other methods.

interpolation, especially by leveraging on the more accurate edge localization of stereo data. The proposed approach also obtains a lower average MSE than all the compared methods. The average error is about 24% lower than [22], which is the best among the compared schemes. Conventional stereo confidence metrics of [47] produce an higher MSE if compared with our stereo metric, e.g., by using PKRN as confidence in the fusion framework the average MSE is 7.9. Our method has better performance than that of the compared schemes for all scenes except the very simple scene 2, in particular notice how it has a larger margin on the most

complex scenes. This implies that our approach captures small details and complex structures while many of the compared approaches rely on low pass filtering and smoothing techniques which work well on simple planar surfaces but cannot handle more complex situations.

Scene	1	2	3	4	5	Avg.
ToF Int.	9.83	10.33	14.43	8.68	15.12	11.67
Stereo	19.17	27.83	18.06	25.52	11.49	20.42
Fusion	**7.40**	9.33	**6.92**	**6.30**	**8.39**	**7.67**
[22]	7.43	**9.27**	12.60	7.99	13.01	10.06
[117]	8.49	9.92	11.44	9.88	15.19	10.98
[120]	9.04	10.04	13.04	9.52	14.03	11.13
[20]	10.98	13.19	9.83	13.93	13.10	12.21
Ideal	2.50	2.60	3.22	2.42	3.16	2.78

Table 3.1: MSE in disparity units with respect to the ground truth, computed only on non-occluded pixels for which a disparity value is available in all the methods.

Chapter 4

Data collection from multiple sensors

Any computer vision algorithm requires to be validated and compared with other methods on a dataset. In the literature many datasets have been proposed for different applications, including detection, classification, recognition, tracking, segmentation, and multiview stereo. An exhaustive and updated list of datasets can be found in [18, 1]. For depth estimation most of the publicly available datasets include only stereo systems, such as the well known works of [36, 97, 99]. To verify the correctness of the results it is required to have both the data acquired from the sensors and the ground truth depth map.

Despite the large amount of datasets publicly available, none of them provides calibrated data from multiple depth cameras and for those with multiple cameras the ground truth depth map is missing. The only datasets publicly available for stereo vision and ToF depth cameras are the ones from Dal Mutto et al. [22, 20] that contain 3 and 5 scenes respectively acquired with two standard cameras and a MESA SR4000 ToF depth camera. These datasets have been criticized for not having enough variability in the different scenes. Recent works made available additional datasets [33] but the ground truth is missing. Also the availability of synthetic datasets is limited. An example of synthetic dataset with a stereo vision system and a ToF depth camera is the HCIBOX depth evaluation dataset [82] that only include data for one scene. A common solution to the lack of data from stereo vision systems and ToF depth cameras is to use the datasets created for stereo systems such as [36, 97, 99] and to subsample the ground truth depth map, add noise and apply a 3D rotation to the point cloud to simulate the different camera pose. Although this solution is widely used, it is only an approximation and does not include many issues of real ToF depth cameras. In addition, this approximation

may introduce artifacts in the generated data due to occlusions and reprojection.

In this chapter we propose two datasets developed for providing data from multiple calibrated sensors, as well as the ground truth depth map. The first dataset is made of real data acquired by consumer depth cameras, while the second dataset contains data synthetically generated that include realistic camera models.

4.1 Real dataset

This section describes the system developed for the simultaneous acquisition of data from different sensors, including stereo, ToF and structured light depth cameras. Figure 4.1 shows a rendering of the acquisition system with the actual displacement of the cameras. We decided to use consumer depth cameras as opposed to expensive professional equipments. In particular the depth cameras used in the collection are:

Stereo vision system We used the ZED camera from Stereolabs [104]. This depth camera based on a passive stereo technology is equipped with two 4MP cameras that provide images up to 2208×1242 [pxl] at 15 fps. The sensor is able to provide images up to 100 fps at a lower resolution. With a baseline of 120 [mm] and a diagonal field of view of 110° this stereo system is able to work in the range $0.7 - 20$ [m] providing 32-bits depth images. In our dataset we only provide left and right images, and not the depth map provided by the software that comes with the sensor.

ToF depth camera The best ToF consumer depth camera is the Kinect™ v2. Compared to other ToF cameras it provides a cleaner and denser depth map, in addition to have the largest resolution. Kinect™ v2 is able to acquire a 512×424 [pxl] depth map at 30 [fps] with a depth estimation error typically smaller than 1% of the measured distances and a diagonal field of view of 92°. The depth camera is able to provide depth images up to 4 [m]. In addition to the depth, Kinect™ v2 also has an additional color camera with the resolution of 1920×1080 [pxl].

Structured light depth camera Given the range of the other two cameras we decided to use the Intel RealSense R200 depth camera, an active stereo system. The spatial resolution of the depth camera is 640×480 [pxl], the working depth range is $510 - 4000$ [mm] and the temporal resolution is up to 60 [fps]. The diagonal field of view of the depth camera is approximately 70°. Also

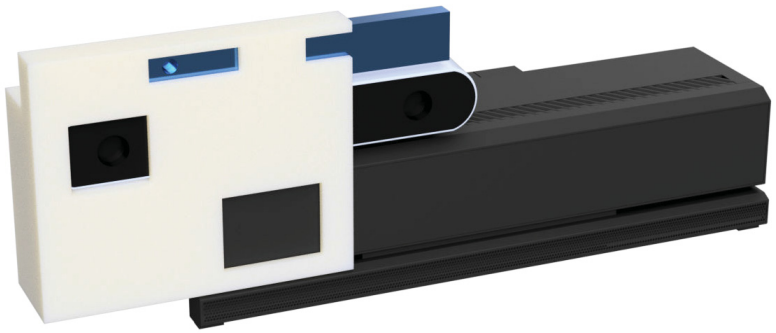


Figure 4.1: Rendering of the multicamera acquisition system.

this sensor comes with an additional color camera providing color images at 1920×1080 [pxl].

Figure 4.1 also shows the holder that has been designed and 3D printed to guarantee that the position of the cameras remains fixed during the calibration procedure and the subsequent data collection. The goal was to arrange the cameras such as they were as close as possible, to reduce the artifacts during the reprojection between different views and to limit the occlusions. In the rest of the section we describe the calibration of the three systems and the procedure to generate a dense ground truth depth map, and finally we show some example of acquired data.

4.1.1 Calibration

Color imaging instruments, such as photo and video cameras, and depth imaging instruments, such as ToF and structured light depth cameras, require preliminary calibration to be used for measurement purposes. Calibration must account both for geometric and photometric properties, and should be accurate and precise for reliable measurements. Geometric calibration accounts for the internal characteristics, called *intrinsic parameters*, and the spatial positions of the considered instruments, called *extrinsic parameters*. Photometric calibration accounts for the relationship between the light emitted from a scene point and the light information acquired by the sensor. We are not interested in the calibration of each internal single component of ToF and structured light depth cameras, since we consider a depth camera as a device providing depth information from a certain reference system.

The calibration process consists of estimating the following quantities for each camera n:

- intrinsic parameters matrix $\mathbf{K}_n$

- distortion coefficients $\mathbf{d}_n$
- rotation matrix $\mathbf{R}_n$ and translation vector $\mathbf{t}_n$ describing the roto-translation between the camera and a reference system

The purpose of this calibration is not to compensate for photometric artifacts of such depth cameras, rather to provide the necessary information to map data of one camera to the others. Most of the time, structured light and ToF depth camera calibration can be performed by the methods for standard camera calibration. This is because if the depth cameras to be calibrated provide an image from the depth camera viewpoint, then the problem of multiple depth camera calibration corresponds to the N-view system calibration. The required images can be, for example, the IR reference camera for structured light depth cameras, or signal amplitude or intensity for ToF depth cameras, replacing the color image for standard cameras.

We followed the approach of Zhang [119] for camera calibration with a regular black and white checkerboard. This method requires to acquire images of the planar checkerboard from different positions and orientations. Differently than the single camera calibration, a setup including 7 different physical cameras is more complicated to deal with. In this case each checkerboard is acquired from 7 cameras with different point of views without perfectly overlapping fields of view, it is important to collect numerous images with the checkerboard visible on all the cameras in this step. There must be also good checkerboard coverage separately on all the cameras to estimate a good undistortion map (the undistortion of the images is performed independently on the cameras). We decided to keep fixed the acquisition system and to move the checkerboard at every acquisition. To avoid misalignment and motion blurry we used a tripod for the checkerboard and waited the checkerboard to be completely stable before acquiring. We also collected 20 images of the same scene for each acquisition, and averaged the images to reduce the noise.

Depth cameras are usually pre-calibrated by proprietary algorithms, and the calibration parameters are stored in the device during manufacturing and made accessible to the user only by official drivers. Usually, the manufacturer's calibration does not completely correct depth distortion, and accuracy can be improved by software procedures correcting camera output data. The correction, however, is based on a specific calibration model whose parameters are identified during the calibration process.

While passive devices such as passive stereo systems usually do not need additional precautions, studies show that ToF depth cameras need a time delay, usually

referred to as *pre-heating time*, before providing reliable depth measurements [64, 66]to reduce the systematic errors in terms of accuracy. For Kinect^{TM} v2, for example, the accuracy is reduced from 5 to 1 [mm] after 30 minutes from the first acquisition. For longer acquisition times the temperature of the camera may increase and affect the measurements, however, passive or active cooling systems usually compensate for such temperature variation.

Another practical difference with respect to standard color cameras is that to calibrate an IR camera with the procedure described in [119] it is necessary to illuminate the scene by sources emitting light in the IR spectrum, as in the case of sunlight or common incandescent light bulbs. Common fluorescent lamps usually do not emit in the IR bands, therefore are not suited to structured light depth cameras' calibration. This practicality requires particular attention since an accurate calibration requires proper illumination. A non uniform illumination results in darker regions with consequently higher noise making the checkers corners localization less precise.

For ToF depth cameras, amplitude images can be collected in two different ways, either in the so called *standard mode*, i.e., with the ToF depth camera illuminators active during the acquisition, or in the so called *common mode*, i.e., with ToF camera illuminators off during the acquisition, namely, using the ToF camera as a standard IR camera. The first solution is more direct as it does not require external tools and generally produces better results, but it requires proper integration time setting to avoid saturation and reduce noise. The second solution requires an external auxiliary IR illumination system as for structured light depth cameras.

Figure 4.2 shows the images acquired from the three systems during the calibration process. For the stereo system only the two color images are available. ToF depth camera provides the intensity of the received signal, the depth map from the same camera and the color image. The structured light depth camera provides the two IR images and the color image. In addition to the two IR images, the structured light camera provides also the depth image, however, during the calibration process we had to turn off the illuminator, and so the depth provided is meaningless.

Once all the images have been collected we run a checkerboard detector on all the images but the depth image from the ToF depth camera, obtaining for each camera n and for each pose k a set of J points $\mathbf{p}_{nk}^{j}$. The calibration parameters are estimated by minimizing the Euclidean distance between the planar positions of

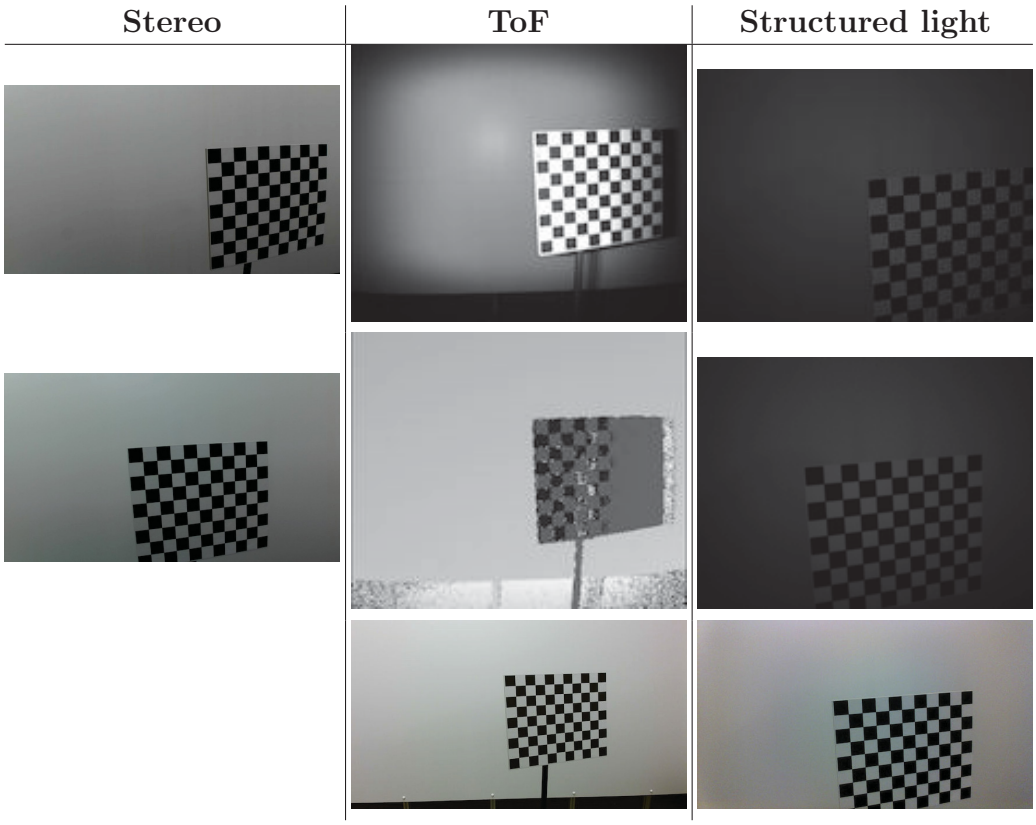


Figure 4.2: Images of the same checkerboard acquired from the three depth cameras during calibration process.

the measured and the projected 3D points after anti-distortion, given by

$$\min_{\mathbf{K}_n,\mathbf{d}_n,\mathbf{R}_{nk},\mathbf{t}_{nk}} \sum_{n=1}^{N}\sum_{k=1}^{M}\sum_{j=1}^{J} \delta_{nk}^{j} \|\mathbf{p}_{nk}^{j} - f(\mathbf{K}_n, \mathbf{d}_n, \mathbf{R}_{nk}, \mathbf{t}_{nk}, \mathbf{P}^{j})\|_2^2 \tag{4.1}$$

where $\mathbf{p}_{nk}^{j}$ is the projection of the 3D feature P^j with coordinates $\mathbf{P}^j$ on the n-th camera at the k-th pose of the checkerboard, δ_{nk}^{j} is 1 if P^j is visible by the n-th camera at the k-th pose and 0 otherwise. The function $f(\mathbf{K}_n, \mathbf{d}_n, \mathbf{R}_{nk}, \mathbf{t}_{nk}, \mathbf{P}^j)$ accounts for projection and distortion. The minimization of (4.1) is solved by nonlinear optimization techniques such as the Levenberg-Marquardt method. Matrices $\mathbf{R}_{nk}$ and $\mathbf{t}_{nk}$ describs the k-th checkerboard pose with respect to the n-th camera. Given that $\mathbf{R}_{nm}$ and $\mathbf{t}_{nm}$ are the rotation and translation matrices relating cameras n and m, the following relationships hold

$$\begin{aligned} \mathbf{R}_{mk} &= \mathbf{R}_{nk}\mathbf{R}_{nm} \\ \mathbf{t}_{mk} &= \mathbf{R}_{nk}\mathbf{t}_{nm} + \mathbf{t}_{nk}. \end{aligned} \tag{4.2}$$

from which one can retrieve the pose of a given camera with respect to the reference

camera.

Usually, single camera calibration is first performed on each camera in the system to reduce the number of unknowns in (4.1) or at least provide a good estimate of those parameters. It is important to constrain the minimization problem when possible, as the number of unknowns grows with the number of cameras in the system. If only data from two depth cameras are needed, it is convenient to calibrate only the two systems, so the number of unknowns is reduced.

4.1.2 Ground truth generation

Different approaches have been developed to acquire a precise depth map of a scene. Range scanners are usually very precise and do not require additional hardware to use them, but the depth map obtained from an external scanner would have a different reference system, and so the scanner needs to be calibrated as well. For stereo vision it is common to use the system of [98], where a regular projector is used to project a texture in the scene, in this case a Gray code pattern, and the stereo pair is used to estimate a very accurate depth map. With this setup it is not necessary to calibrate also the external projector. Following this rationale we developed a system based on a line laser that allows one to obtain a detailed depth map of the scene from the same point of view of one of the cameras used in the acquisition. Also in this case we do not require the projector to be calibrated, since we use the laser line only to facilitate the matching of correspondent points in the two cameras. With our acquisition system where IR cameras are used, the system of [98] cannot work with regular projectors, since they don't emit enough light in the portion of the spectrum where the IR filters of the cameras are set.

The algorithm developed to compute a dense depth map uses a stereo camera to match corresponding pixels and estimate the disparity between them. Since in our acquisition system we have two stereo cameras, one from the stereo vision system and one from the structured light depth camera, we provide the ground truth from both the cameras. However, since the camera of the structured light system have IR filters that the standard cameras of the passive stereo system do not have, we had to use two different line lasers, one with IR illuminator acquired by the structured light depth caera, and one with a regular red illuminator visible to humans, acquired by the passive stereo vision system.

The goal is to "paint" the scene with the line laser and for each acquisition match corresponding lit points in the two images. Ideally we want to match only 1 point for each row of the image for each acquisition. Due to noise in the images we update the estimated disparity for a given pixel, every time there is a new

measurement, by accumulating all the values and keeping the median value. Figure 4.3 shows an example of images acquired by the structured light depth camera (first row) and the stereo vision system (second row). The third column shows a zoomed version of the acquired laser line in the two systems. From the images in the third column it is visible that the line spans multiple columns in the image, therefore we estimate the center of the line by computing the maximum value and refining it using the parabola fit to obtain sub-pixel precision in the localization. We want to keep the width of the laser line as small as possible to reduce errors. The width of the laser line can be adjusted by operating on the lens system of the laser itself but the quality of the images acquired by the cameras also depend on the camera's properties such as gain and exposure. For both the systems we collect images of the line laser without external illumination to reduce the noise of the acquired images and to increase the contrast of the line laser with respect to the background illumination.

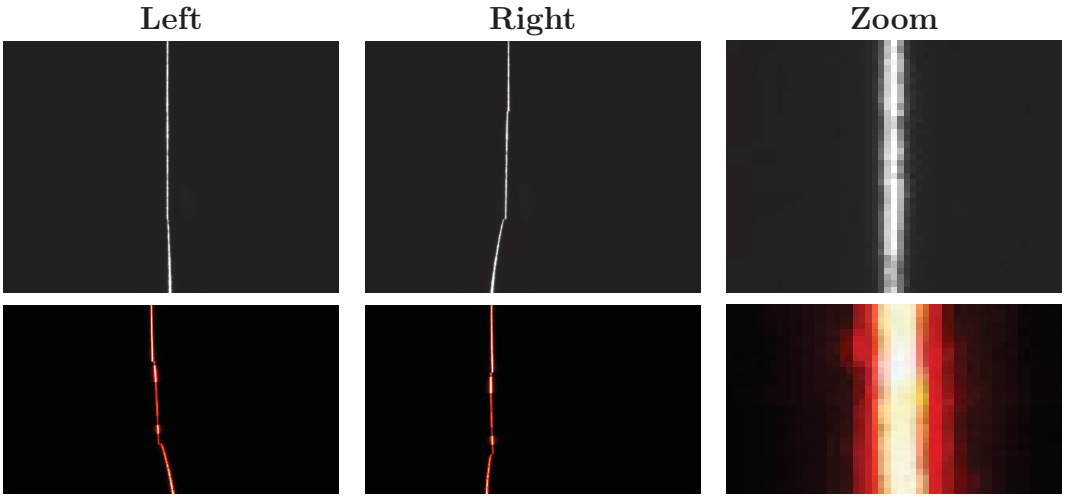


Figure 4.3: Line laser acquired from left and right camera: of the structured light depth camera (*first row*); of the stereo vision system (*second row*). The third column shows a closeup of the line laser.

From Figure 4.3 it is visible the difference between the two systems, in the structured light depth camera it is possible to control the exposure of the cameras and so the saturation of the acquired line. With the stereo camera used in the acquisition instead in was not possible to control the exposure of the cameras, resulting in wider lines. In the structured light depth camera we set the gain to the minimum value and adjusted the exposure such that the line laser was visible also in dark regions of the scene. To avoid casting unwanted shadows in the scene, the line laser should be kept as close as possible to the acquiring cameras. To control the laser movement we used a servomotor controlled by an Arduino that makes the

system fully automatic.

4.1.3 Acquired scenes

To validate the system we acquired 10 scenes of different nature, all including static scenes in an indoor environment. The scenes have different complexity, ranging from flat surfaces to more complex shapes like the leaves of a plant. We acquired objects with different texture as well so it is possible to check the behavior of the algorithms with different texture. Different scenes have materials with different specularity, including reflective and glossy surfaces as well as rough material that usually cause problems to active cameras. Figure 4.4 shows a reference color image

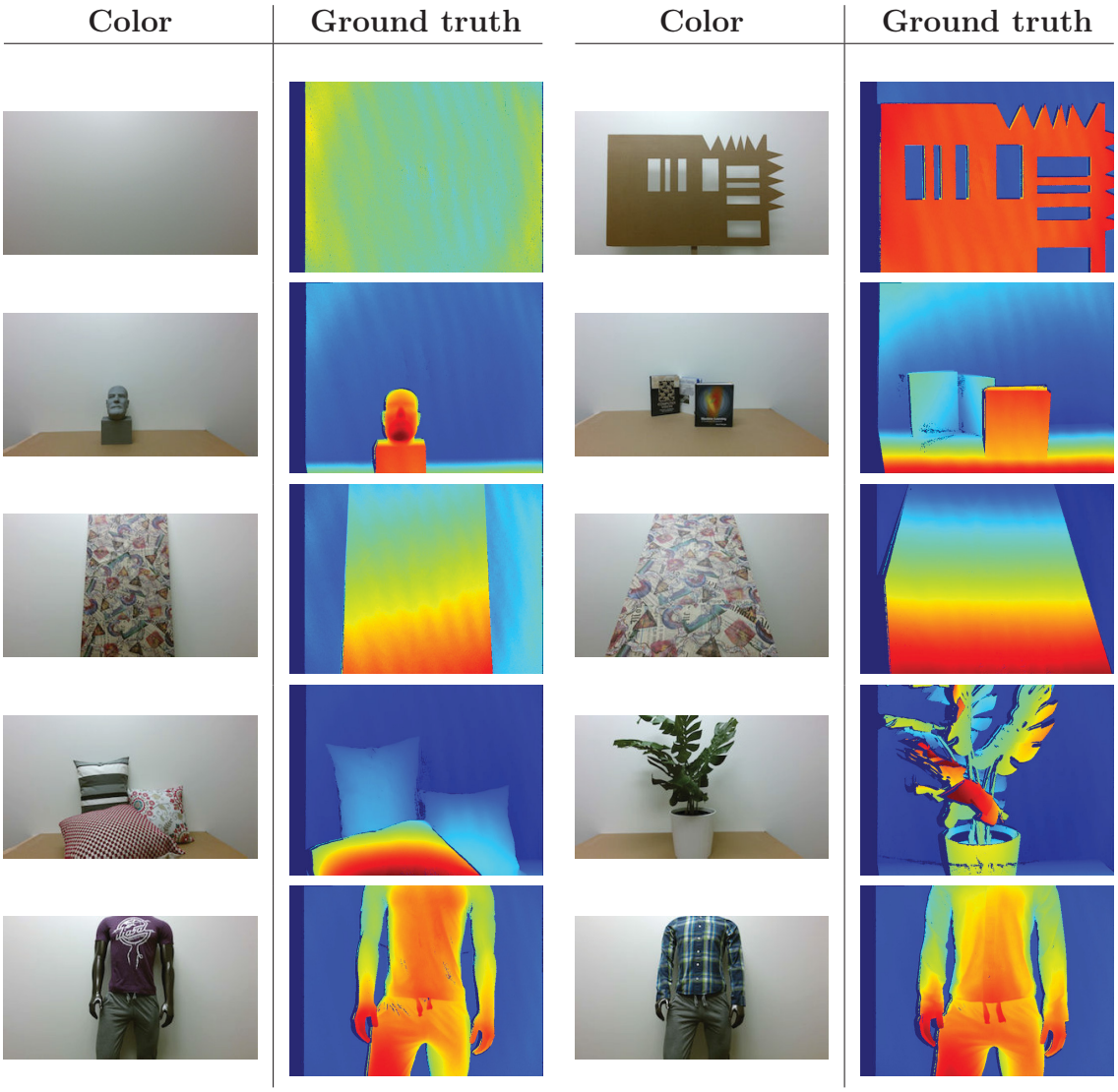


Figure 4.4: Color images of the acquired scenes and relative depth map.

and the associated ground truth image for the 10 sequences. The color image comes

from the color camera of the ToF depth camera while the ground truth is estimated from the point of view of the structured light depth camera.

We acquired each scene with 4 different external illumination, to test the robustness of the algorithms with different lightning conditions. Different levels of illumination include an acquisition with no external light, one with regular lighting, another one with stronger light and the last one with an additional incandescent light to stress the active cameras. We added the last mode because the standard illumination that we used for the first modes does not have an IR component, while the spectrum of incandescent lights also include frequencies in the working range of active depth cameras.

Figure 4.5 shows an example of acquisition with the structured light depth camera in the three different lighting conditions. Each row represent a different intensity of the illumination, starting from no illumination in the first row, followed by regular indoor illumination in the second row and additional incandescent illumination in the third row. Comparing the first two rows we can confirm that the presence of fluorescent light in the scene does not affect the performance of the structured light depth camera, this is because the spectrum of fluorescent lights does not include emissions in the spectrum of the structured light depth camera. The presence of additional light from an incandescent source causes a degradation of the depth quality. This is a well known problem for active devices. From the IR image in the third row we can notice that the structure projected by the illuminator is attenuated, causing a reduction in the uniqueness and so a degradation of the overall depth quality. This effect is stronger in slanted surfaces like the top of the table, where the intensity of the projected pattern received by the camera is lower due to the orientation of the table with respect to the camera.

4.2 Synthetic dataset

The acquisition of the dataset described in the previous section has several limitations. First of all it provides only one acquisition for each scene, acquiring the same scene from a different point of view would require to repeat the acquisition from the sensors and the generation of the ground truth from the different point of view. Although the process is automatic, it still requires some manual adjustment to tune the line laser and most important it requires a substantial amount of time. Furthermore, recent results obtained by machine learning suggest that deep learning based approaches may be also used for the task of 3D data fusion. For example, Convolutional Neural Networks (CNN) require a large labelled dataset to train the

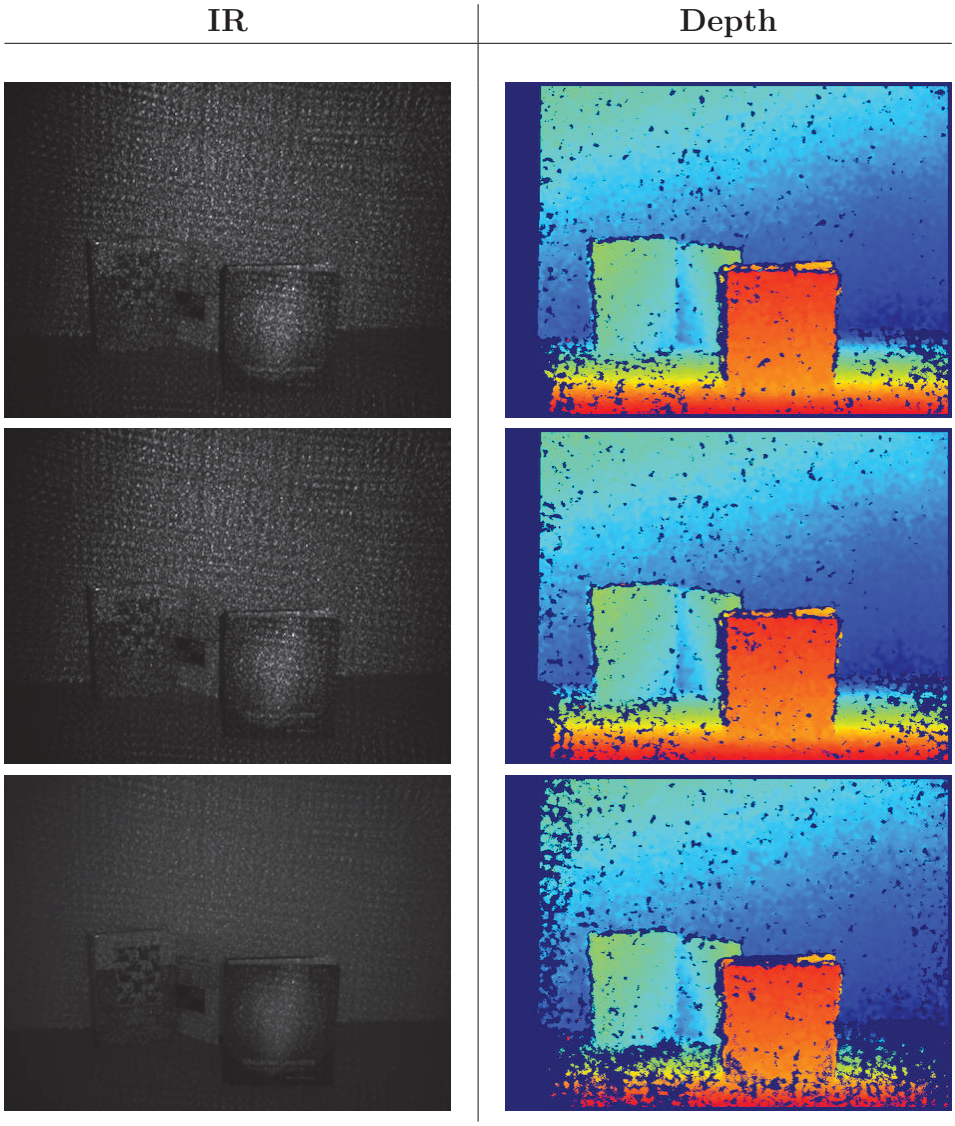


Figure 4.5: Effects of different external illumination on a structured light depth camera.

network, and for this task a simulator of ToF and stereo systems that allows one to generate realistic synthetic views of a given 3D model is fundamental.

The simulation of the acquisition from standard cameras is a well studied problem and many physical models to generate realistic acquisitions have been proposed. In the context of stereo vision systems an example of realistic synthetic dataset is "Tsukuba" proposed in [86]. This dataset provides 1800 stereo images of an indoor scene with four different illuminations, as well as the true depth map for each view. For ToF depth cameras, [54] proposes a framework to simulate a ToF sensor but it is missing many fundamental components. In contrast to other methods [100] focuses on the simulation of sensor hardware and do not handle illumination or other fundamental aspects.

In this section we introduce a synthetic dataset for stereo vision and ToF depth

cameras that can be easily extended also to structured light depth cameras. The image acquisition process in synthetic datasets only consists of rendering a 3D model on a computer and apply some post processing to take into account physical properties of the acquisition system. In addition, a synthetic data generator allows one to easily change the scene and lighting conditions as well as camera parameters. The disadvantage of synthetic datasets is usually the lack of realism in the acquired images. The goal of the proposed framework is to provide support to real datasets and not to replace them. In addition, it provides an easy way of testing algorithms under different aspects, from camera parameters, to scene geometry and lighting.

4.2.1 Scene rendering

The software to generate synthetic data is written in C++ and OpenGL, that allows the system to automatically handle occlusions and interpolation between vertexes of the 3D model. While CPUs apply single instructions sequentially to each element, GPUs are highly optimized to efficiently process input data in parallel, making the simulator able to generate data in real time.

The proposed framework requires in input:

- a 3D model with associated texture, such as a Wavefront obj file. Figure 4.6 shows some of the models available in the dataset. This framework is not limited to work with the models currently available, but it just requires a 3D model with associated texture to work;

- a calibration file with all the calibration information of each camera in the system. Those information are for example intrinsic and extrinsic parameters of each camera. The relative position of each camera in the acquisition system is fixed;

- a list of positions of the camera from which the acquisition has to be performed. Each entry specifies the camera position, where the camera is looking at and the up position. Figure 4.7 shows an example of the trajectory generated for an acquisition. In this example the camera is moving around the origin of the reference system. Each entry specifies the position and orientation of the reference camera, all the other cameras moves accordingly;

- a parameters file with all the settings required to generate specific data, such as noise settings and illumination.

Given all these inputs, the simulator loads the 3D model, creates the virtual cameras according to the parameters in the calibration file and generates the images

Figure 4.6: Example of 3D models from [13, 103] used to generate synthetic data.

acquired by the virtual cameras. Figure 4.8 shows the data generated by the simulator for the stereo vision system that include:

- left and right images acquired from the two cameras;
- ground truth depth or disparity map for the left and the right view. Since for each pixel the depth is known, there are no pixels with invalid depth as in the case of the real dataset.

For the ToF depth camera the generated data are shown in Figure 4.9 and include:

- the intensity map including the realistic models described in the next section;
- the depth map including the realistic models described in the next section;
- the ground truth depth map corresponding to the real depth before applying any processing.

This framework can be extended to generate data for structured light depth cameras by replacing the illumination function with the actual pattern of the structured light illuminator, and use the same pipeline developed to generate stereo data.

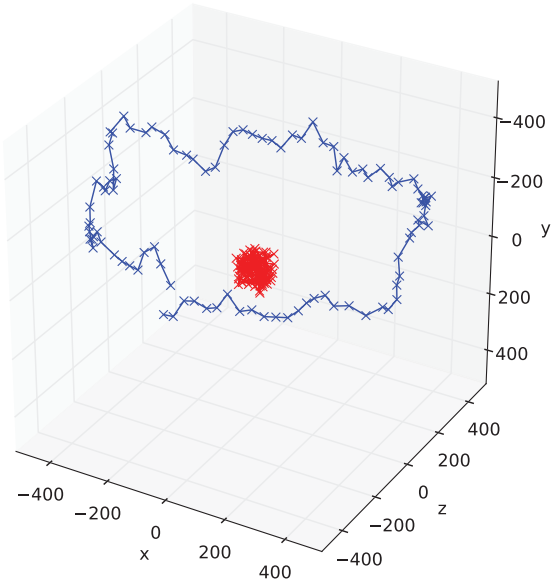


Figure 4.7: Trajectory of the virtual cameras. *Blue* lines represent the position of the cameras at different time, while *red* lines represent where the cameras are looking at.

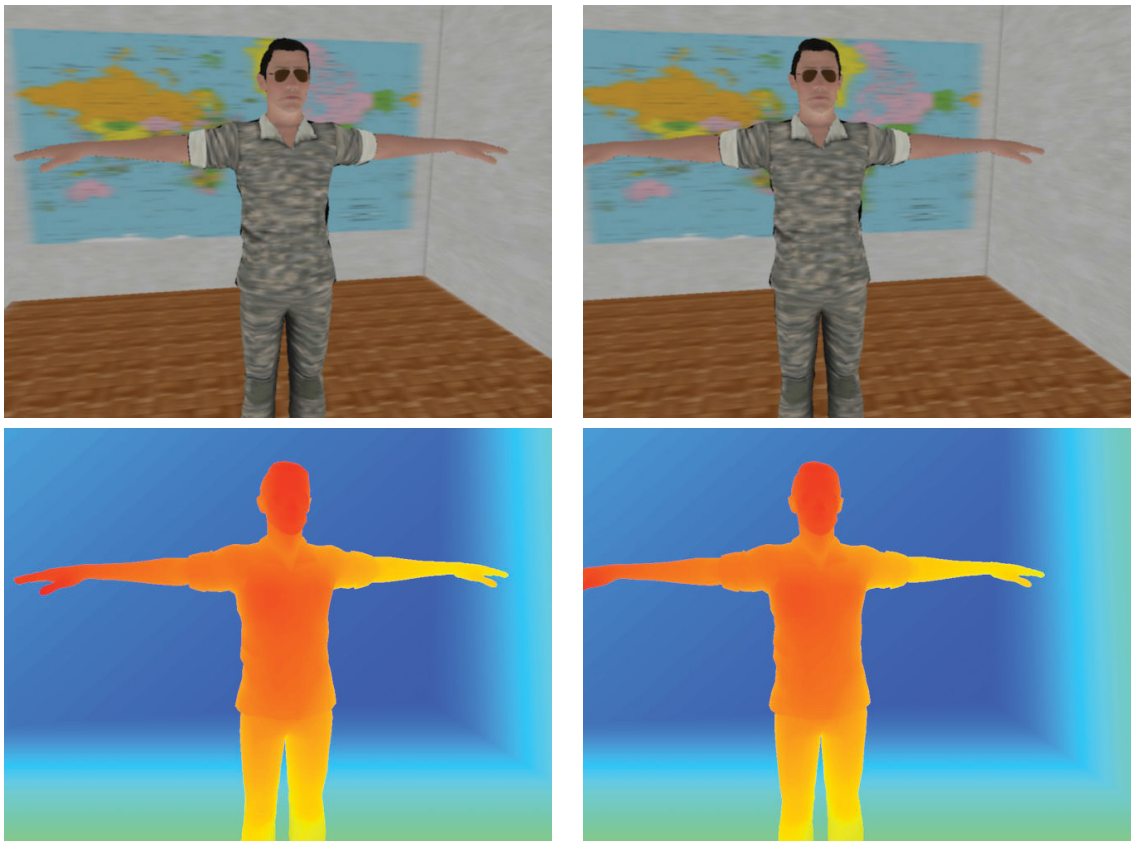


Figure 4.8: Data generated by the simulator for the stereo vision system. *First row* show the color images for left and right cameras. *Second row* shows the ground truth depth for left and right cameras.

Figure 4.9: Data generated by the simulator for the ToF depth camera. The data are the intensity map, the depth and the ground truth depth.

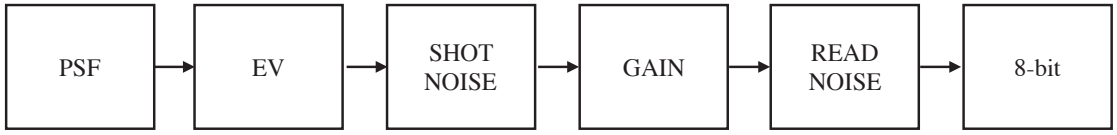


Figure 4.10: Overview of the processing applied to color images.

4.2.2 Camera models

Generated data shown in Figure 4.8 and Figure 4.9 are the result of post processing applied to the raw images of color and depth that OpenGL provides. In addition to the parameters of the models, it is possible to specify additional illumination in the scene that will affect the acquisition of the two systems. The simulator replicates the major artifacts of stereo vision and ToF depth cameras as explained in the next two sections.

Stereo vision system

Stereo vision system is made of two independent cameras, therefore the same pipeline is run on both the color images independently. To simulate the acquisition of a standard camera we decomposed the acquisition process according to the steps in Figure 4.10.

The input to this pipeline is the color image from the OpenGL pipeline, that is the rasterization of the input model according to the camera pose and camera parameters. Since the pipeline just described works in a spatial domain that includes neighboring pixels it runs on CPU, since the GPU architecture does not allow easily to use information from neighboring pixels.

The first processing is performed to simulate the presence of the lens, resulting in a defocusing of blurring of the input image. This is usually described with the point spread function (PSF) describing the response of the imaging system to a point source. We approximate the PSF with a Gaussian function with parametrized

standard deviation and kernel size.

The next step includes the simulation of the shutter speed, encoded in the exposure value (EV), that we implemented as a multiplication of the input image by a parametrized factor representing the integration time. The idea is that acquisitions with longer EV result in brighter images.

Images from real cameras are corrupted with noise at different stages of the acquisition and until this point no noise has been introduced. The next step is the simulation of shot noise, that we model as a random noise with Poisson distribution with mean proportional to the intensity of the pixel (that is proportional to the number of received photons). The intensity of the noise is parametrized and we apply the noise independently in each of the three channels.

The next step is the simulation of the digital gain that consists in a multiplication by a scalar applied to the images already corrupted by the shot noise. The effect is that both the useful signal and the noise get amplified.

An additional source of noise in real cameras is the read noise, corresponding to the amount of noise generated by electronics as the charge present in the pixels is converted to voltage and amplified prior to digitization in the Analogue to Digital Converter (ADC) of the camera. It is modeled as additive white Gaussian noise (AWGN) with parametrized average and standard deviation.

The final step corresponds to the conversion of the image to 8-bit. This process provides as output images in the range $0 - 255$ with integer pixel values.

ToF depth camera

ToF depth camera provides two different outputs that are an intensity map of the signal and a depth map. Figure 4.11 shows the steps required to obtain the two output images.

To obtain the output depth map we start from the ground truth depth generated with OpenGL. First we model the flying pixel effect by applying a Gaussian blur with parametrized standard deviation and kernel size to the input depth map. To affect only depth edges we generate a mask of the edges, where a pixel is considered an edge if the gradient computed in the depth map is above a certain threshold. To simulate the multipath effect we apply the same blurring using another mask computed using the normal vectors. For each point of the scene, in addition to the depth value we also have the coordinates of the normal vector to the surface in that position. The mask is valid where the orientation of the normal vectors in a window surrounding the considered pixel has high variability. This approximation does not take into account the difference between concave and convex angles and

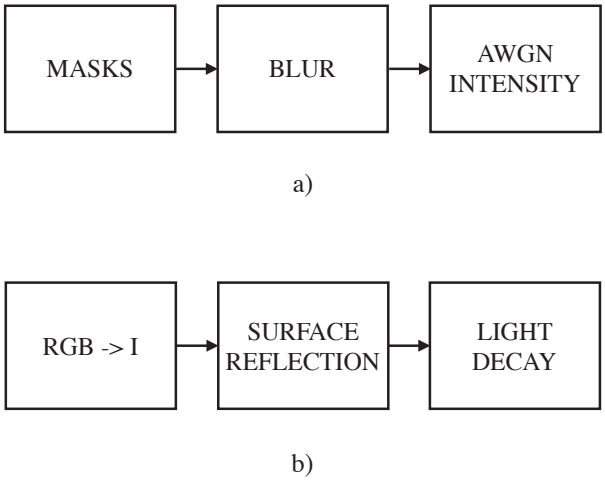


Figure 4.11: Overview of the processing applied to generate data of the ToF depth camera: a) pipeline for the depth map; b) pipeline for the amplitude image.

need to be improved. Most complex models directly handle the multipath effect by means of ray tracing techniques.

Then we add white Gaussian noise with parametrized average and standard deviation to model the random noise affecting the circuitry that processes the received signal. To make the noise dependent on the intensity of the signal we scale the AWGN by the intensity map. In this way the distance measurement is influenced by the total amount of received light. However, this is not a generic ToF problem but only applies to some ToF based depth camera.

While the processing applied to the depth image is perform on CPU because of the interaction with neighboring values, the generation of the amplitude signal is performed entirely on the GPU. According to the OpenGL specification, the fragment shader is responsible to assign a color value to the fragment generated by the rasterization. The information available in the fragment shader for each point is: the color of the associated point in the model, the normal vector to the surface of the model in the considered point and the position of the point with respect to the camera.

The first step involves the conversion of the color input value to intensity. Instead of simply converting the color input to grayscale, we estimated the intensity I to be

$$I = R * 0.677 + G * 0.115 + B * 0.208 \tag{4.3}$$

where R, G and B are the three input channels red, green and blue. Those numbers have been set by acquiring with a ToF camera three patches of different colors, red, green and blue, and computing for each channel the transformation between intensity provided by the IR camera and the associated RGB value.

To simulate the physical interaction of the light with the model and the camera, first we attenuate the input intensity by a factor proportional to the scalar product between the normal of the point and the direction of the light. In the model implemented the light is supposed to come from the optical center of the camera but with a small modification it can be made more general and simulate the position of the illuminator in a different position. For example for the MESA SR ToF depth cameras the assumption that the illuminator is positioned in the center of the camera is correct, but it is not true for the Kinect$^{\text{TM}}$ v2, where the illuminator is positioned with a certain horizontal disparity from the camera.

Another effect of ToF intensity signal is the illumination decay from the center of the image as seen in Chapter 2. The quality of the received signal depends on the angle at which the light is received and to model this effect we attenuate the intensity from the previous step with a factor that depends on the scalar product of the direction of the incoming ray and the direction of the camera.

Chapter 5

Gesture recognition with depth camera and Leap Motion

Gesture recognition, either static or dynamic, can be framed as a family of pattern recognition tasks including the extraction from the object of interest of one or more feature sets describing relevant pattern properties, and the comparison of features' values with a classification model previously trained. The goal is detecting the most likely entry from a given "gesture dictionary" that generated the actual gesture. In this chapter we propose a framework for static gesture recognition using a depth camera and a Leap Motion device. Chapter 2 already introduced the technology of current depth cameras. The Leap Motion instead is a device targeted to recognition and tracking of hands and fingers. The device provides the discrete position of hands and fingers with high precision and tracking frame rate. The Leap Motion controller uses two IR cameras and an IR diffuse illuminator. The cameras have a field of view of about 150°. The effective range of the Leap Motion controller is between 25 and 600 [mm] above the device.

Depth cameras allow one to obtain a complete 3D description of the framed scene while the Leap Motion sensor is a device explicitly targeted for hand gesture recognition and provides only a limited set of relevant points. Since depth cameras and the Leap Motion have quite complementary characteristics (e.g., a few accurate and relevant keypoints against a large number of less accurate 3D points), it seems reasonable to use them together for gesture recognition purposes. This chapter presents a novel approach for the combined use of the two devices for hand gesture recognition. An ad-hoc solution for the joint calibration of the two devices is firstly presented. Then a set of novel feature descriptors is introduced both for the Leap Motion and for depth data. Various schemes based on the distances of the hand samples from the centroid, on the curvature of the hand contour and on the

convex hull of the hand shape are employed and the use of Leap Motion data to aid feature extraction is also considered. The proposed feature sets are fed to two different classifiers, one based on multi-class SVMs and one exploiting Random Forests. Different feature selection algorithms have also been tested to reduce the complexity of the approach. Experimental results show that a very high accuracy can be obtained from the proposed method. The current implementation is also able to run in real-time.

5.1 Related Works

Hand gesture recognition from data acquired by consumer depth cameras is a well studied problem. Gestures can be classified according to their dynamism into static and dynamic. Static gestures are often characterized by the shape or the pose assumed by the hand at a given instant, e.g., a gesture from the American Sign Language alphabet. Dynamic gestures instead represent continuous and atomic movements, e.g., raising an arm. Gestures are often characterized by the trajectory followed by the hand's center throughout the whole input sequence [12, 85], or by its speed [67]. Most gesture recognition methods share a common pipeline. First, the hand is identified in the framed scene and segmented from the background. Then, relevant features are extracted from the segmented data and eventually the performed gesture is identified from a set of predefined gestures, possibly exploiting suitable machine learning techniques. In the case of non-static gestures, the general pipeline also includes tracking features among multiple frames. In this chapter we focus on static gesture recognition.

Many approaches have been presented for static gesture recognition, mostly based on the standard scheme of extracting relevant features from the depth data and then applying machine-learning techniques to the extracted features. In the approach of [63], silhouette and cell occupancy features are extracted from the depth data and used to build a shape descriptor. The descriptor is then used inside a classifier based on action graphs. Other approaches, e.g., [105] and [112] are based on volumetric shape descriptors. The two approaches both exploit a classifier based on Support Vector Machines (SVM). The histograms of the distance of hand edge points from the hand center are instead used in the approaches of [93] and [92]. Another approach based on an SVM classifier is [28], that employs 4 different types of features extracted from the depth data.

Other approaches instead estimate the complete 3D hand pose from depth data. Keskin et Al. [55] try to estimate the pose by segmenting the hand depth map into

its different parts, with a variation of the machine learning approach used for full body tracking in [102]. Multi-view setups have also been used for this task [4], since approaches based on a single camera are affected by the large amount of occluded parts, making the pose estimation rather challenging.

The use of Leap Motion data for gesture recognition systems has recently attracted more interest [60, 26, 69]. A preliminary study on the usage of this device for sign language recognition has been presented in [89]. The device has been used for sign language recognition in [79]: in this work the data extracted from the sensor is fed directly to two different machine learning classification algorithms, one based on a Naive Bayes Classifier and one exploiting Multilayer Perceptron Neural Networks. Another recent work [111] analyzes the trajectory of a finger returned by the Leap Motion to recognize handwriting. The approach exploits Dynamic Time Warping and a nearest neighbor search. The sensor has also been used for signature recognition using features based on the optical flow and on the trajectories in a recent work [84]. A gesture interface based on the Leap Motion has been presented in [40], where the authors use the device to control a robot arm.

5.2 Problem Formulation

The general architecture of the approach presented in this chapter is shown in Figure 5.1: there are two different feature extraction pipelines, one for the Leap Motion data and one for depth data and finally a classification stage that takes in input all the features and recognizes the performed gesture.

The the depth camera and the Leap Motion require a joint calibration before combining their data. An ad-hoc approach for this critical step based on the fingertips positions in the two reference systems is presented in Section 5.3. The Leap Motion feature extraction pipeline, described in Section 5.4 exploits only the data from this sensor and extracts 4 different types of features, i.e., fingertip distances from the centroid of the hand, fingertip elevations from the palm plane, the angles between the vectors connecting the fingertips with the palm center and the 3D positions of the fingertips in the hand reference system. Before describing the features extracted from the depth camera data, Section 5.5 and 5.6 describe how to segment the hand and classify fingers and the palm region. This step is required by the feature extraction pipeline, described in Section 5.7, mainly based on the information extracted from the depth sensor. It extracts four different sets of features based on the distances of the finger samples from the hand center, on the local curvature of the hand contour, on the similarity between distance feature

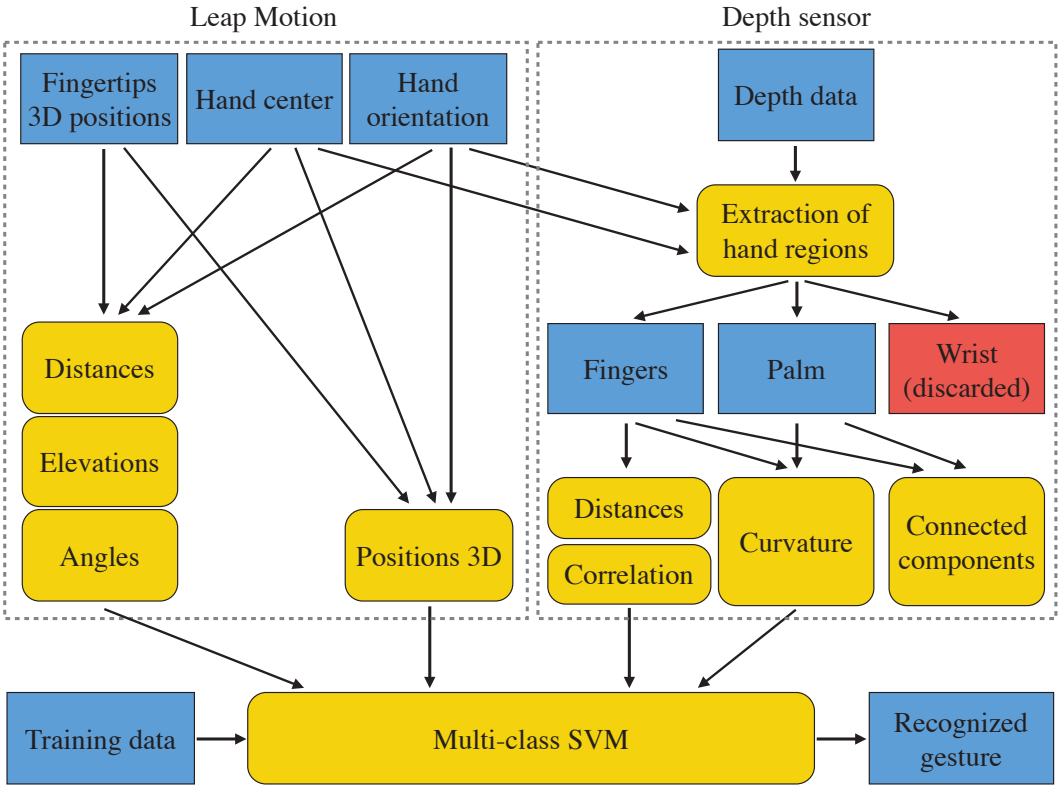


Figure 5.1: Pipeline of the proposed approach.

histograms and on the connected components in the convex hull of the hand shape. Finally, the features are processed with the classification method based on Support Vector Machines (SVM) presented in Section 5.8.

5.3 Calibration

Since the employed acquisition setup jointly exploits the 3D measures from two different sensors, i.e, the Leap Motion device and the depth sensor (with optionally a color camera rigidly attached to the depth one), it is necessary to jointly calibrate the two devices to bring the measures of one sensor in the reference system of the other. The proposed approach is independent from the relative position of the two sensors, however notice that a set of practical limitations of the sensors limits the choices in the setup construction:

- The Leap Motion must be placed under the hand, typically on the desk looking up. Furthermore its operating range is limited.
- The depth sensor has typically a minimum working distance, below which it does not provide depth estimates. This distance depends on the employed

sensor, e.g., the Kinect™ v1 we used for the results section has the limitation that it cannot acquire objects closer than 500 [mm] to the sensor. The maximum distance is instead typically bigger than the Leap Motion one.

- If the palm plane is roughly perpendicular to the optical axis of the depth camera more depth samples are acquired for the hand leading to better performances
- Inside the working range, also having the sensor closer to the hand leads to more accurate data

Considering all the previous observations, we found that the setup that allows to obtain the best performance is the one shown in Figure 5.2. As it is possible to see from the figure, in the proposed setup the Leap Motion has to be put under the performed gesture, while the depth sensor has been placed a little more forward, facing the user, as in most gesture acquisition systems using this sensor.

Figure 5.2: Acquisition setup.

The aim of the calibration procedure is to estimate the extrinsic parameters of the two devices, i.e., the coordinate system transformation between the reference systems of the two devices, or equivalently the position of one sensor with respect to the other one. Notice that our implementation for testing the algorithm uses the Kinect™ v1 sensor but the proposed calibration algorithm remains valid also for other depth cameras. In particular, our approach does not require an additional color stream. Furthermore, the two devices need also to be independently calibrated to correctly locate points in the 3D space. The Leap Motion software already

provides a calibration tool, while the Kinect™ v1 requires an external calibration, e.g., it is possible to use the approach of [45], in which both the color and the depth map from the sensor are used to extract intrinsic and extrinsic parameters. Our gesture recognition scheme requires to associate to each point in the scene a depth value, therefore only the projection matrix of the depth camera will be used. Given the two sensors independently calibrated, for every acquisition we get two sets of data describing the scene. The Leap Motion provides a point cloud with up to 6 points, including one for the palm center and up to 5 for the fingertips. Data retrieved from the Kinect™ v1 consist instead in a full frame depth map with an associated color image (the latter is not used in the proposed approach).

The standard procedure to find the roto-translation between the two sensors requires to have the 3D coordinates of a set of points in the two coordinate systems. From the description of Leap Motion data (Section 5.4), it naturally follows that the only calibration clue that can be used is the hand itself. We decided to use the open hand gesture as the calibration tool (i.e., gesture G9 of the results database, see Figure 5.13). This is because the Leap Motion software is not able to provide a one-to-one map between fingertips and real fingers, it just gives the positions in a random fashion: when 5 fingers are detected, though, we are quite sure that all the fingertips have been detected and with a few pre-processing they can be ordered and then associated to the correct fingers. The same points then need to be detected also from the depth camera. The two sets of points will then be used inside the calibration algorithm. The proposed calibration of a Leap Motion and a depth sensor allows to easily make the two devices working together, without the need of external tools like checkerboards or other classic calibration devices. This is a key requirement for a human-computer interaction system. Moreover, the proposed approach allows to easily set up a gesture recognition system exploiting the two devices, without the need of having them rigidly attached to a fixed structure. Whenever one of the two devices is moved, the system re-calibration only requires the acquisition of a couple of frames of the user's open hand. Notice that a new calibration is mandatory only if the devices are moved.

5.3.1 Extraction of fingertips position from Leap Motion data

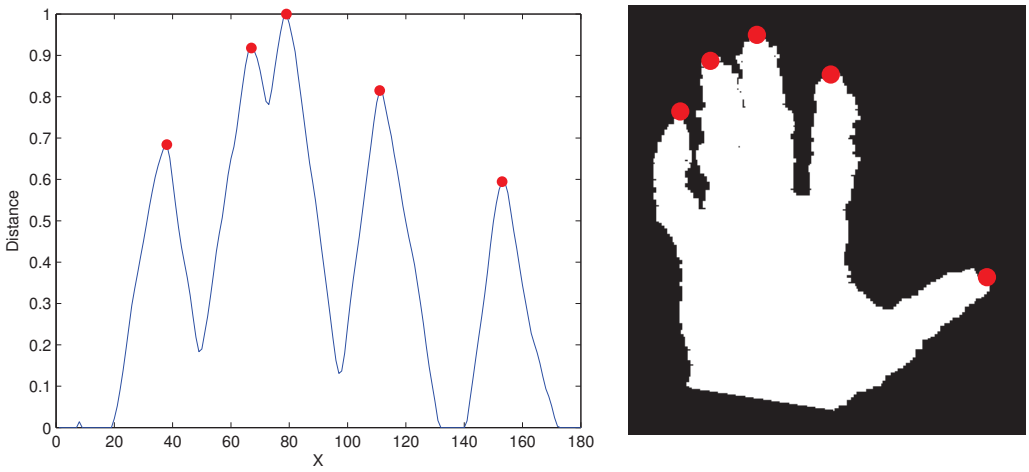
Starting from the hand orientation and the palm center estimated from the Leap Motion, the palm plane can be extracted and the fingertips projected on it. We decided to use the hand direction as a reference and then to associate to the thumb the fingertip with the most negative angle between the principal axis and the

projected fingertip, and to the other fingers the remaining fingertips by increasing angular values, up to the fingertip with the greatest angular value associated to the pinky. Section 5.4 presents a description of the data acquired from the sensor and in particular provides more details on the angle computation. After this operation we obtain a set of 5 points $\mathcal{X}_L = \mathbf{X}_L^1, ..., \mathbf{X}_L^5$ describing the fingertips in the Leap Motion coordinate system.

5.3.2 Extraction of fingertip positions from depth data

For the depth sensor, instead, a more complex approach is required to extract fingertip positions from the acquired depth image. In order for the calibration process to be completely automatic, we decided to avoid the need to manually selecting points, relying instead on an automatic fingertips extraction algorithm. The idea is to extract the hand region from the acquired depth stream and then to process the hand contour to detect fingertips. Notice that the hand extraction scheme of Section 5.5 exploits also the Leap Motion data so it can not be directly applied in this case. The extraction of the hand has instead been performed using the approach of [28] where the hand center is initially estimated by using a Gaussian filter on the samples density and then refined by fitting a circle on the palm. Finally PCA is used for the computation of the hand orientation.

Then the hand contour is analyzed using the same approach used for the distance features in Section 5.7. The distance d of each point X of the hand contour from the palm center is computed, thus obtaining the function $d(\mathbf{X})$. The fingertips are assumed to be the points of the fingers at the maximum distance from the center. Given the function $d(\mathbf{X})$, its local maxima are the points $\bar{\mathbf{X}}$ where $f'(\bar{\mathbf{X}}) = 0$ and $f''(\bar{\mathbf{X}}) < 0$. Due to the inaccuracy in the depth image, the hand contour is usually irregular and needs to be smoothed before searching for the local maxima. In addition, only the 5 highest maxima are used and to avoid multiple detections on the same finger a minimum distance between two candidates is guaranteed. Figure 5.3 shows an example of function $d(\mathbf{X})$ in blue, red circles show the detected local maxima and the relative fingertips in the depth image. Once these points have been detected, the correspondent values in the depth image are selected and through the projection matrix of the depth camera they are back-projected in the 3D space obtaining the 3D coordinates of the fingertips in the depth camera coordinate system $\mathcal{X}_D = \{\mathbf{X}_D^1, ..., \mathbf{X}_D^5\}$. It is worth noticing that the Leap Motion API does not specify which actual point of the finger shape is returned as the fingertip, therefore we decided to consider as fingertip the farthest point of the finger.



a) b)

Figure 5.3: Hand contour and detected fingertips: a) distance of each point of the hand contour, the red circles are the detected local maxima; b) projected local maxima on the hand mask of the depth image.

5.3.3 Roto-translation estimation

The final step is the computation of the roto-translation that links the two reference systems. To be more robust against noise we acquire several frames, even if a single frame is theoretically sufficient. Let us denote with $\mathcal{X}_{L,f}$ and $\mathcal{X}_{D,f}$ the sets of points acquired by the Leap Motion and the depth camera respectively, each relative to each frame $f = 1, ..., F$. With the acquired fingertip 3D positions, the goal is to find the roto-translation parameters R and $\mathbf{t}$ using a mean squared error cost function that will best align all the considered fingertip points in the two reference systems in all the acquired frames:

$$(R, \mathbf{t}) = \arg\min_{R,\mathbf{t}} \sum_{f=1}^{F} \sum_{i=1}^{5} ||R\mathbf{X}^i_{L,f} + \mathbf{t} - \mathbf{X}^i_{D,f}||_2^2 \tag{5.1}$$

i.e., to find the best roto-translation that brings the point cloud $\mathcal{X}_L$ to the point cloud $\mathcal{X}_D$ (the point clouds $\mathcal{X}_L$ and $\mathcal{X}_D$ are the union of all the points clouds of the considered frames). Since the corresponding set of equations corresponds to an over-determined system and the measures are affected by noise, we used a RANSAC robust estimation approach to solve it. From our tests we found out that the assumption of considering as fingertip the extreme point of the finger is quite a valid assumption and that the mean error obtained from the square root of (5.1) for all the tested people is about 9 [mm].

5.4 Feature extraction from the Leap Motion data

As already stated, the Leap Motion device provides only a limited set of relevant points and not a complete description of the hand shape. The amount of information is more limited if compared to the one provided by depth cameras, but on the other side the device provides directly some of the most relevant points for gesture recognition and allows to avoid complex computations needed for their extraction from depth and color data. The Leap Motion sensor mainly provides the following data (Figure 5.4):

- **Number of detected fingers** $N \in [0, 5]$ that the device is currently seeing.
- **Position of the fingertips** $\mathbf{F}_i$, $i = 1, ..., N$. Vectors $\mathbf{F}_i$ containing the 3D positions of each of the detected fingertips. The sensor however does not provide a mapping between the vectors $\mathbf{F}_i$ and the fingers.
- **Palm center C** that represents the 3D location roughly corresponding to the center of the palm region in the 3D space.
- **Hand orientation** consists in two unit vectors representing the hand orientation computed in the palm center **C**. The first vector, denoted with **h**, points from the palm center to the direction of the fingers, while the second, denoted with **n**, is the normal to the plane that corresponds to the palm region pointing downward from the palm center.
- **Hand radius** r is a scalar value corresponding to the radius of a sphere that roughly fits the curvature of the hand (it is not too reliable and it is not used in the proposed approach).

Note that the accuracy is not the same for all the reported data vectors. The 3D positions of the fingertips are quite accurate: according to a recent research [113] the error is about 200 μm. This is a very good accuracy, specially if compared to the one of depth data acquired by the Kinect™ v1 and from other similar devices. While the localization of the detected fingers is accurate, their recognition is not too reliable. There are some situations in which the sensor is not able to recognize all the fingers. Fingers folded over the hand or hidden from the sensor viewpoint are not captured, furthermore fingers touching each other are sometimes detected as a single finger. Even in situations where the fingers are visible and separated from the hand and the other fingers it can happen that some fingers are lost, specially if the hand is not perpendicular to the camera. Another typical issue of this sensor is that protruding objects near the hand, like bracelets or sleeve edges, can be confused

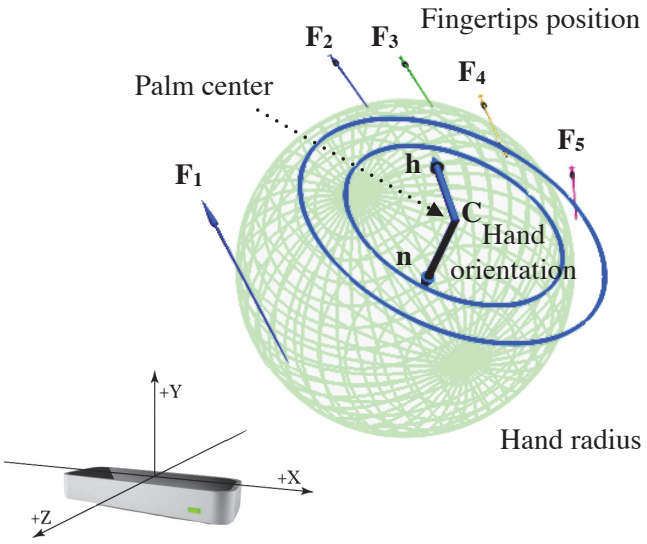


Figure 5.4: Data acquired by the Leap Motion device.

with fingers. These issues are quite critical and must be taken into account in developing a reliable gesture recognition approach since in different executions of the same gesture the number of captured fingers could vary. For this reason simple schemes based on the number of detected fingers have poor performance.

As previously stated, the Leap Motion does not provide a one-to-one map between fingers and fingertips detected. In the proposed approach we deal with this issue by sorting the features on the basis of the fingertip angles respect to the hand direction **h**. To this purpose, we consider the projection of the hand region into the palm plane described by **n** and passing through **C**, as depicted in Figure 5.6. The plane is then divided into five angular regions S_i, $i = 1, ..., 5$ as in Figure 5.5, and each captured finger is assigned to a specific region according to the angle between the projection of the finger in the plane and the hand direction **h**. Note that a unique matching between the sectors and the fingers is not guaranteed, i.e., some of the sectors S_i could be associated to more than one finger and other sectors could be empty. When two fingers lie in the same angular region, one of the two is assigned to the nearest adjacent sector if not already occupied, otherwise the maximum between the two feature values is selected.

In this work we analyze 4 different types of features computed from the Leap Motion data and these will be described in the rest of this section:

- **Fingertip angles**: angles corresponding to the orientation of each fingertip projected on the palm plane with respect to the hand orientation **h**.

- **Fingertip distances**: 3D distances of the fingertips from the hand center.

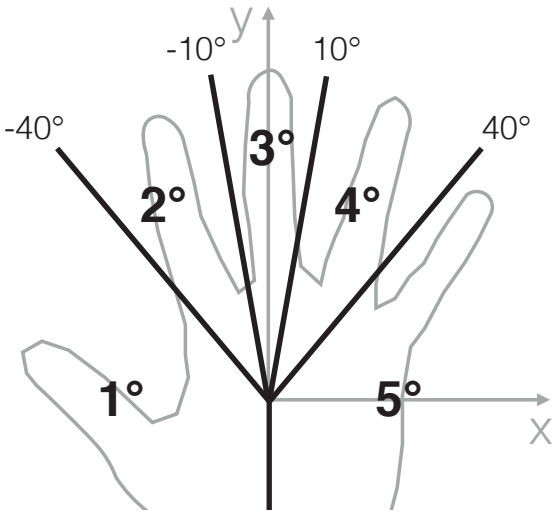


Figure 5.5: Angular regions in the palm plane.

- **Fingertip elevations**: distances of the fingertips from the palm region plane.
- **Fingertip positions**: x, y and z coordinates of the fingertips in the 3D space.

To make the approach robust to people with hands of different size all the feature values (except for the angles) are normalized in the interval $[0, 1]$ by dividing the values for the distance between the hand center and the middle fingertip length $S = ||\mathbf{F}_{middle} - \mathbf{C}||_2$, where $\mathbf{F}_{middle}$ is the position of the middle fingertip. The scale factor S can be computed during the calibration of the system. Figure 5.6 depicts a sample gesture acquisition and the related feature set.

5.4.1 Fingertip angles

The computation of this feature plays a key role also for the other features since the angle is used as a metric to order the fingertips. The fingertip angle is defined as:

$$A_i = \angle(\mathbf{F}_i^{\pi} - \mathbf{C}, \mathbf{h}), i = 1, ..., N \tag{5.2}$$

where $\mathbf{F}_i^{\pi}$ is the projection of $\mathbf{F}_i$ on the plane identified by $\mathbf{n}$, and corresponds to the orientation of the projected fingertip with respect to the hand orientation. The estimated hand orientation $\mathbf{h}$ and consequently the fingertips angles are strongly affected by the number of detected fingers. The obtained values A_i have been scaled and the interval has been set to $[0.5, 1]$ to better discriminate, in the classification step, the valid values from the missing ones, that have been set to 0. These values have also been used to assign each finger to the corresponding sector as described before. Fingertip angles features are then collected into vector $\mathbf{F}^a$.

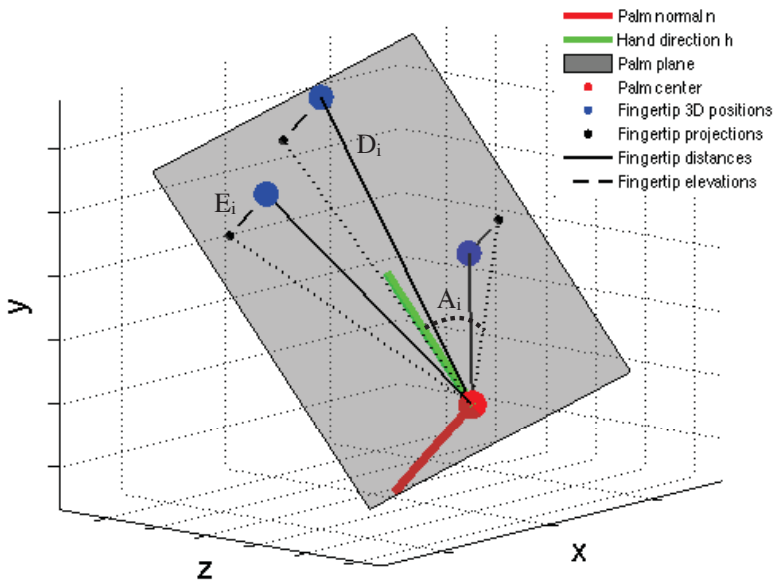


Figure 5.6: Considered Leap Motion features on a gesture example (sample of gesture G8 from our dataset).

5.4.2 Fingertip distances

This feature represents the distance of each fingertip from the palm center. Distances are defined as:

$$D_i = ||\mathbf{F}_i - \mathbf{C}||_2/S, i = 1, ..., N \tag{5.3}$$

and they are ordered according to increasing angles. At most one feature value is associated to each sector and the missing values have been set to 0. Fingertip distances are collected into vector $\mathbf{F}^d$.

5.4.3 Fingertip elevations

Another descriptor for a fingertip is its elevation from the palm plane. Elevations are defined as:

$$E_i = \text{sgn}((\mathbf{F}_i - \mathbf{F}_i^\pi) \cdot \mathbf{n})||\mathbf{F}_i - \mathbf{F}_i^\pi||_2/S, i = 1, ..., N \tag{5.4}$$

and thanks to the sign operator it describes also to which of the two semi-spaces, defined by the palm plane, the fingertip belongs. As for the previous features, there is at most one feature value for each sector and the missing values have been set to 0. Note that as for the fingertip angles, the values range has been scaled to the interval $[0.5, 1]$ and then collected into vector $\mathbf{F}^e$.

5.4.4 Fingertip 3D positions

This feature set represents the positions of the fingertips in the 3D space. As for the previous features, firstly the fingertips have been ordered according to increasing angles, then, since a reliable hand gesture recognition system must be independent from the hand position and orientation inside the frame, it is necessary to normalize the coordinates with respect to the hand position and orientation:

$$\begin{aligned} P_i^x &= (\mathbf{F}_i - \mathbf{C}) \cdot (\mathbf{n} \times \mathbf{h}) \\ P_i^y &= (\mathbf{F}_i - \mathbf{C}) \cdot \mathbf{h} \\ P_i^z &= (\mathbf{F}_i - \mathbf{C}) \cdot \mathbf{n} \end{aligned} \tag{5.5}$$

It is worth noticing that the fingertip 3D positions can be seen as the compact representation of the combination of angles, distances and elevations, i.e., of the first three features. Fingertip 3D positions have been collected into vector $\mathbf{F}^p$.

5.5 Hand segmentation using depth and Leap Motion data

In previous approaches [28] the extraction of the hand from color and depth data was performed with a time-consuming procedure based on several steps. Firstly the closest point was localized on the depth data. Then a multiple thresholding on the depth values, on the distance from the closest point and on the color values with respect to the skin color was used to obtain a first estimate of the hand samples. The hand centroid was estimated in the subsequent step by finding the maximum of the output of a Gaussian filter with a large standard deviation applied to the estimated hand mask (this corresponds to assume that the densest region belongs to the hand palm). A circle is then fitted on the hand palm to precisely locate its center and to divide the hand into palm, wrist and fingers regions. Finally PCA is exploited to compute the hand orientation. The details of this approach can be found in [28], however it is clear that it is a quite complex operation as most of the computation time of the entire pipeline of [28] was spent on this step. Moreover, there is a couple of critical assumptions, i.e., that the closest point matching the skin color correspond to the hand and that the palm is the densest region, that can lead to wrong detections in particular situations. This typically happens in simple settings with a user is in front of the computer, but limits the applicability of the approach in more complex scenarios.

Since in the proposed approach the Leap Motion data are also available, this

information can be exploited to make the identification of the hand position and of its orientation faster and more reliable. Firstly the hand centroid computed by the Leap Motion $\mathbf{C}$ can be expressed according to the depth camera coordinate system using the calibration information. In this way, if the Leap Motion correctly recognizes the hand, we can ensure that the hand is properly identified even if there are objects of similar shape and color in the depth sensor acquisition. Moreover, we can also avoid the use of color information thus making the approach faster and allowing the use of depth sensors that do not have an associated color camera (e.g., industrial ToF depth cameras like MESA or PMD devices). In this section we will assume that the two devices have been jointly calibrated obtaining a rotation matrix R and a translation vector $\mathbf{t}$ between the two reference systems. How to perform the calibration will be the subject of Section 5.3. The location of the Leap Motion hand centroid in the depth camera reference system will be denoted with $\mathbf{C}_D = R\mathbf{C} + \mathbf{t}$ and used as a starting point for the hand detection. A sphere of radius r_h is then centered on $\mathbf{C}_D$ and the samples inside the sphere are selected, i.e:

$$\mathcal{H} = \{X : \|\mathbf{X} - \mathbf{C}_D\|^2 \leq r_h\} \tag{5.6}$$

where X is a generic 3D point acquired by the depth camera and r_h is set on the basis of the physical hand size (in the tests, $r_h = 100$ [mm] has been used). The points in the set $\mathcal{H}$ inside the sphere represent the initial hand estimate. This allows to remove the assumption that the hand is the closest point to the sensor. Furthermore, the thresholding in the color space can be avoided, as well as the acquisition and processing of color data, making this step faster and simpler. The centroid located by the Leap Motion is very reliably located in the hand region but its localization is not too accurate, due to the uncertainty in the position estimated from the Leap Motion. For this reason, its position is optimized with the circle fitting scheme of [28]. A more refined scheme employing an ellipse in place of the circle can also be used [71]. Let us denote with $\mathbf{C}_{palm}$ the final circle and with r its radius computed by the algorithm.

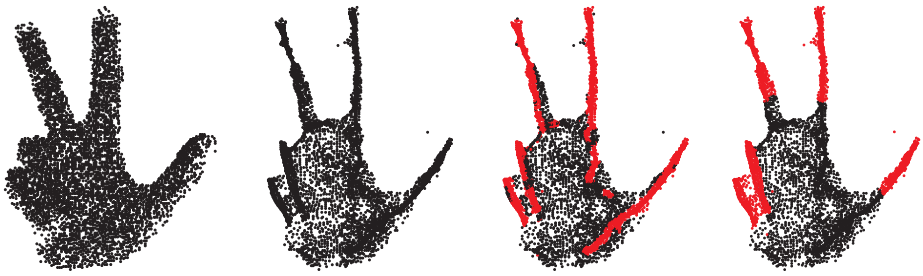
The hand orientation can also be extracted from the Leap Motion data (it is given by the vectors $\mathbf{h}$ and $\mathbf{n}$ as discussed in Section 5.4), therefore also the computation of the PCA can be avoided. Another critical aspect in the approach of [28] is that with PCA the orientation was quite well estimated, but the direction was supposed always pointing upward. With the proposed approach, instead, this assumption can be removed, relying on the direction estimated by the Leap Motion.

Finally, the hand samples are subdivided into fingers, palm and wrist regions. Palm samples ($\mathcal{P}$) are the ones inside the circle of radius r centered on $\mathbf{C}_{palm}$; the

finger samples set $\mathcal{F}$ contains the samples X outside $\mathbf{C}_{palm}$ that satisfy $(\mathbf{X}-\mathbf{C}_D)\cdot\mathbf{h} > r$, i.e., the ones outside the circle in the direction of $\mathbf{h}$; the remaining samples are associated to the wrist region ($\mathcal{W}$).

5.6 Hand segmentation using density based clustering

Another possible approach to palm and finger segmentation is proposed in [78], where we propose a density based clustering approach to divide the hand into palm and fingers using a single depth map. The hand is firstly segmented from the rest of the scene, then it is divided into palm and fingers regions. For this task we employed a novel scheme that exploits the idea that fingers have a tubular shape while the palm is more planar. Following this rationale we applied a contraction guided by the normals to reduce the fingers into thinner structures that can be identified by analyzing the changes in the point density. Density-based clustering is then applied to classify the points into palm and fingers. Figure 5.7 shows all the steps performed by the algorithm.



a) b) c) d)

Figure 5.7: Hand segmentation with normal guided contraction: a) Original point cloud $\mathcal{H}$; b) Contracted version of the point cloud $\mathcal{H}^c$; c) Contracted cloud with the labels after the first assignment; d) Final assignment after the refinement. *red* samples are associated to fingers, *black* to the palm.

The input of this algorithm shown in Figure 5.7a is a point cloud $\mathcal{H} = \{p_1, ..., p_n\}$ containing the hand samples. Figure 5.7b shows the result of the normal contraction, in which a new point cloud $\mathcal{H}^c = \{p_1^c, ..., p_n^c\}$ is built by moving each point p_i in the direction opposite to the surface normal $\mathbf{n}_i$ at that location, i.e.:

$$p_i^c = p_i - t\mathbf{n}_i \tag{5.7}$$

The offset t is set to a fixed value, corresponding approximately to the average radius of a finger, to maximize the contraction of the fingers regions (for the experimental

results we used $t = 9$ [mm]). In this way the tubular surfaces are contracted into thinner structures, while planar surfaces are just shifted of a small amount in the direction perpendicular to the plane, keeping the same point density. The idea is that after the contraction step, the high density regions are more likely to be fingers while low density regions are associated to the palm.

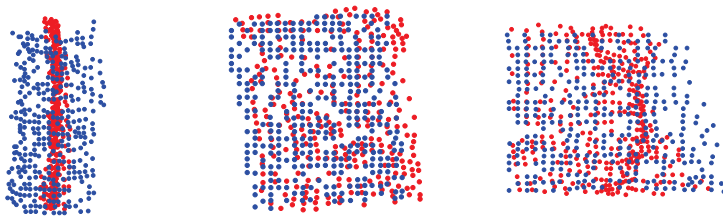
The next step is the segmentation of the hand into the palm and fingers region. This operation is simple in the case of raised fingers but becomes very challenging when the fingers are bent over the palm. In our approach we intuitively associate the samples of $\mathcal{H}^c$ within the high density regions to the fingers, the remaining points belonging to the palm. A naive approach to divide the two clusters is to consider a threshold on the number of points inside a spherical neighborhood of a given point in the contracted cloud. Some regions of the palm showing an initial density greater than the one of finger samples may however maintain a final high density even when subject to a slight contraction. Instead, the number of misclassified points is greatly reduced if, given a point, we consider its neighborhood and compare the original spacing between samples in the point cloud $\mathcal{H}$ with the spacing in the contracted point cloud $\mathcal{H}^c$. To label the ith point in the cloud as finger $\mathcal{F}$ or palm $\mathcal{P}$, we first consider the set of its k closest points in the contracted cloud $\mathcal{N}^c_{i,k} = \{p^c_{j_1}, ..., p^c_{j_k}\}$ and compute their average distance from p^c_i. We then consider the same neighbors as they appears in the original cloud, that is $\mathcal{N}_{i,k} = \{p_{j_1}, ..., p_{j_k}\}$, and compute their average distance from p_i. The ratio between the average distances before and after the contraction is then compared to the average of the same ratio computed in the overall hand point cloud. Points with a ratio greater than the average are assigned to the finger set $\mathcal{F}$ while the others are assigned to the palm set $\mathcal{P}$, i.e,:

$$\begin{aligned} \bar{d}_i &= \frac{\sum_{s=1}^{k} \|p_{j_s} - p_i\|}{k} \\ \bar{d}^c_i &= \frac{\sum_{s=1}^{k} \|p^c_{j_s} - p^c_i\|}{k} \\ r_i &= \bar{d}_i / \bar{d}^c_i \\ \bar{r} &= \frac{\sum_{i=1}^{n} r_i}{n} \end{aligned} \tag{5.8}$$

$$\begin{aligned} r_i < \bar{r} \quad &\Rightarrow \quad p^c_i \in \mathcal{P} \\ r_i \geq \bar{r} \quad &\Rightarrow \quad p^c_i \in \mathcal{F} \end{aligned} \tag{5.9}$$

Figure 5.8 helps to better understand this step. Let us first consider a region associated to fingers (shown in Figure 5.8a), the average spacing between a point

and its neighbors in $\mathcal{N}^c_{i,k}$ is much smaller than the one computed with respect to $\mathcal{N}_{i,k}$. In Figure 5.8b, an internal region of the palm is shown, where the spacing does not decrease after the contraction, as the normals in this region are almost parallel. Figure 5.8c shows instead a boundary region of the palm, where the spacing decreases but not as significantly as in the fingers region. Here in fact, differently from the fingers, there are more parallel normals or in general the curvature is less pronounced. We decided to use the mean of all the ratios as threshold value, but of course a different thresholding criteria can be used. Figure 5.7c shows the output of this first raw assignment; notice how, by working with point clouds and using densities in the 3D space, the proposed approach is invariant to rotations and to the orientation of the hand.



a) b) c)

Figure 5.8: Difference of the density before and after the contraction for three particular regions (*best viewed in colors*, *blue* points belong to the original point cloud $\mathcal{H}$, *red* points belong to the contracted point cloud $\mathcal{H}^c$): a) Fingers region; b) Palm region; c) Palm edge region.

After this operation, there could still be some isolated spots of erroneously classified points, especially along the palm edges. A refinement process is therefore needed. In particular, small spots labeled as fingers surrounded by larger areas labeled as palm are very likely to be artifacts. For this reason we iteratively check for each point the ratio between palm points and finger points in a neighborhood of the point itself and update its label according to this ratio. To be more robust, we define two thresholds δ_f and δ_p:

$$\begin{aligned} \frac{|\mathcal{N}^c_{i,k} \cap \mathcal{F}|}{|\mathcal{N}^c_{i,k} \cap \mathcal{P}|} > \delta_f \quad &\Rightarrow \quad p^c_i \in \mathcal{F} \\ \frac{|\mathcal{N}^c_{i,k} \cap \mathcal{P}|}{|\mathcal{N}^c_{i,k} \cap \mathcal{F}|} > \delta_p \quad &\Rightarrow \quad p^c_i \in \mathcal{P} \end{aligned} \tag{5.10}$$

The two thresholds should be both larger than 1 (e.g. $\delta_f = 1.2$ and $\delta_p = 1.5$ in the experimental results), to ensure that the assignment is changed only if the sample is surrounded by a large set of samples in the other region. Different values however

do not affect too much the results. From Figure 5.7c we can notice how the small spots classified as fingers in the region of the palm by the first raw estimation are then correctly classified, as shown in Figure 5.7d.

Figure 5.9 shows some results of the proposed method compared with the method based on fitting a circle in the palm presented earlier [28].

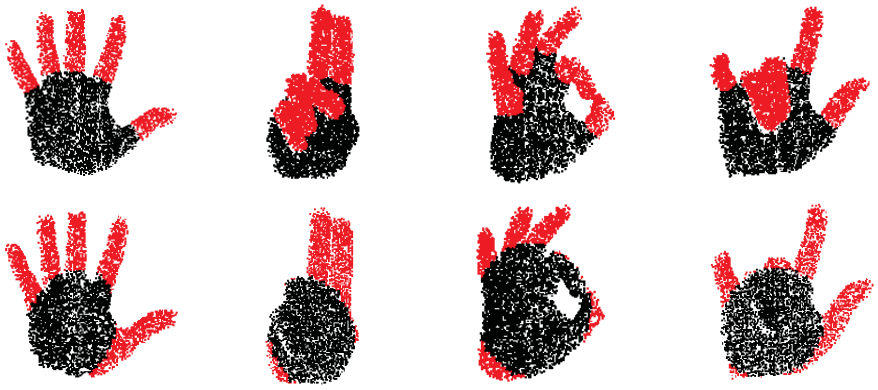


Figure 5.9: Palm and finger segmentation: (*first row*) density based approach; (*second row*) circle based approach [28].

5.7 Feature extraction from depth camera data

In the proposed approach, gestures are acquired with both a Leap Motion and a depth camera. We used a Kinect™ v1 for testing the algorithm but any other depth camera can be used for this purpose. Feature extraction from depth data requires two main steps: firstly the hand is extracted from the rest of the scene using the acquired depth information, then, a set of features is computed from the segmented region.

The first step is quite time-consuming if solved by using only the depth and color data as done in previous works [30, 28]. In the proposed approach, the Leap Motion information is used both to improve the accuracy and to reduce the computation time of the hand detection and segmentation. Using this information, the assumption that the hand is the closest object can be safely removed.

In the second step four different kinds of features are computed from the depth data:

- **Curvature features**: analyze the hand contour shape to extract the particular shape description.
- **Distance features**: consider the distance of each point of the hand contour from the palm center to describe the hand shape.

- **Correlation features**: these are a measure of similarity between distance features.
- **Connected components features**: exploiting the convex hull, compute the size and the number of connected components in the hand silhouette.

5.7.1 Distance features

This feature set aims at capturing the profile of the hand contour to extract informative description of the performed gesture. We start by considering each point X in the hand contour, extracted from the hand mask in the depth image, the distance $d(\mathbf{X})$ with respect to the hand center $\mathbf{C}_{palm}$:

$$d(\mathbf{X}) = ||\mathbf{X} - \mathbf{C}_{palm}||_2 \tag{5.11}$$

Given the hand orientation, then, we are able to provide a coherent function $d(\mathbf{X})$ among different gestures and repetitions. For example we can set as starting point $\mathbf{X}_1$ the intersection between the hand contour and the hand direction $\mathbf{h}$, and then proceed clockwise with the other points until the last one $\mathbf{X}_n$. For each acquisition, though, the number of points in the hand contour n is not fixed, as it depends on the actual distance of the hand from the camera. Therefore, the function $d(\mathbf{X})$ is sampled to get 180 values that makes the descriptor independent from the hand to camera distance. This value can be chosen even smaller without excessively impacting the overall accuracy, but reducing the computation time. An example of this function is shown in Figure 5.3a.

The distance function $d(\mathbf{X})$ is then normalized by the length L_{max} of the middle finger to scale the values within the range $[0, 1]$ and to account for different hand sizes among people. The distance samples are collected into feature vector $\mathbf{F}^l$. Notice that this descriptor is different from the distance descriptors used in [28]: the approach proposed in this work turned out to be simpler, faster and more accurate.

5.7.2 Correlation features

This feature set is based on the similarity between distance functions of subsection 5.7.1. For each considered gesture, a reference acquisition is selected and the corresponding distance function is computed with the approach of Equation (5.11), thus obtaining a set of reference functions $d^r_g(\mathbf{X})$, where g is the considered gesture. The distance function of the acquired gesture $d(\mathbf{X})$ is also computed and the maximum of the correlation between the current histogram $d(\mathbf{X})$ and a shifted version of the reference histogram $d^r_g(\mathbf{X})$ is selected:

$$R_g = \max_{\Delta} \left[\rho \left(d(\mathbf{X}), d_g^r(\mathbf{X} + \Delta) \right), \rho \left(d(-\mathbf{X}), d_g^r(\mathbf{X} + \Delta) \right) \right] \tag{5.12}$$

where $g = 1, \ldots, G$ and $d(-\mathbf{X})$ is the flipped version of the distance function to account for the possibility for the hand to have either the palm or the dorsum facing the camera. The computation is performed for each of the candidate gesture, thus obtaining a set $\mathbf{F}^\rho$ containing a different feature value f_g^ρ for each of them. Note how, ideally, the correlation with the correct gesture should have a larger value than the others.

5.7.3 Curvature features

This feature set describes the curvature of the hand edges on the depth map. A scheme based on on integral invariants [68, 62] has been used. The approach for the computation of this feature is basically the same of [28]. The main steps of the approach are here briefly recalled. The curvature feature extractor algorithm takes as input the edge points of the palm and fingers regions and the binary mask B_{hand} corresponding to the hand samples on the depth map. A set of circular masks with increasing radius is then built on each edge sample (for the results $S = 25$ masks with radius varying from $0.5cm$ to $5cm$ have been used, the radius correspond to the scale level at which the computation is performed).

The ratio between the number of samples falling in B_{hand} for each circular mask and the size of the mask is computed. The values of the ratios (i.e., curvatures) range from 0 (extremely convex shape) to 1 (extremely concave shape), with 0.5 corresponding to a straight edge. The $[0, 1]$ interval is quantized into N bins. Feature values $f_{b,s}^c$ collects how many edge samples have a curvature of a value inside bin b at scale level s. The values are finally normalized by the number of edge samples and the feature vector $\mathbf{F}^c$ with $B \times S$ entries is built. For faster processing, the circular masks can be replaced with simpler square masks and then integral images can be used for the computation. This approximation, even if not perfectly rotation invariant, is significantly faster and the performance loss is very small.

5.7.4 Connected components features

Another useful clue used for gesture recognition schemes [85] is the convex hull of the hand shape in the depth map. The idea is to look for regions within the convex hull of the hand shape but not belonging to the hand. These typically correspond to the empty regions between the fingers and those are a good clue to recognize the fingers arrangement. Let $\mathcal{S} = C_{hull}(\mathcal{B}) \setminus \mathcal{B}$ be the difference between

the convex hull and the hand shape (see Figure 5.10 a and b). Region $\mathcal{S}$ is made of a few connected components S_i. The size of each region S_i is compared with a threshold T_{cc} and the ones that are smaller than the threshold are discarded (this allows to avoid considering in the processing small components due to noise, as the one shown on the right of the hand in Figure 5.10 c). The output of this procedure is the set $\mathcal{S}_{cc} = \{S_i : S_i > T_{cc}\}$ (Figure 5.10 c and d).

The feature set is given by the ratios between the area of each connected components and the convex hull area, i.e.:

$$f_i^{cc} = \frac{area(S_i | S_i \in \mathcal{S}_{cc})}{area(C_{hull}(\mathcal{B}))} \tag{5.13}$$

where the areas have been sorted according to the angle of their centroid with respect to the hand direction (i.e., from the thumb to the pinky). These numbers are then collected into vector $\mathbf{F}^{cc}$.

a) b) c) d)

Figure 5.10: Areas of the connected components: a) and b): difference between the convex hull and the hand shape; b) connected components in set $\mathcal{S}_{cc}$ highlighted in green.

5.8 Gesture classification

The approaches of Sections 5.4 and 5.7 produce eight different feature vectors, four for the Leap Motion data and four for the depth data. Each vector describes some relevant clues regarding the performed gesture and two different classification schemes have been used to perform the recognition, one based on a multi-class Support Vector Machine classifier and one based on Random Forests. There are 8 feature vectors grouped into the two sets $\mathbf{V}_{leap} = [\mathbf{F}^a, \mathbf{F}^d, \mathbf{F}^e, \mathbf{F}^p]$ that contains all the features extracted from Leap Motion data and $\mathbf{V}_{depth} = [\mathbf{F}^l, \mathbf{F}^\rho, \mathbf{F}^c, \mathbf{F}^{cc}]$ that collects the features computed from depth information. Feature vectors extracted from the two devices are visually summarized in Figure 5.11. Each vector can be used alone or together with any of the other descriptors. The combination of

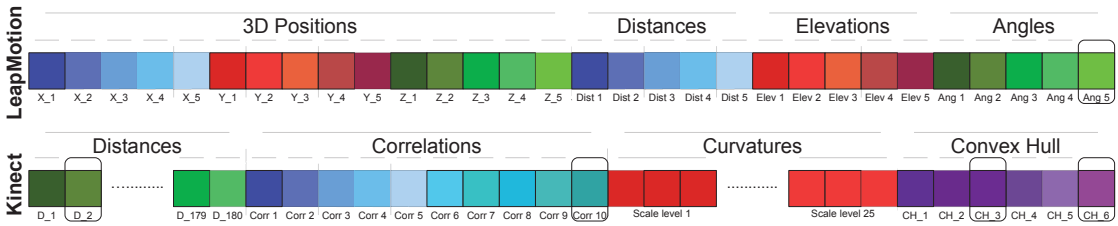


Figure 5.11: Feature vectors extracted from the two devices.

multiple feature descriptors can be obtained by simply concatenating the vectors corresponding to the selected features. The target of the approach is to classify the performed gestures into G classes, one for each gesture in the considered database.

The first classification scheme exploits a multi-class SVM classifier [14] based on the *one-against-one* approach. In the employed scheme a set of $G(G-1)/2$ binary SVM classifiers are used to test each class against each other. The output of each of them is chosen as a *vote* for a certain gesture. For each sample in the test set, the gesture with the maximum number of votes is selected as the output of the classification. In particular a non-linear Gaussian Radial Basis Function (RBF) kernel has been selected and the classifier parameters have been tuned exploiting grid search and cross-validation on the training set. Let us consider a training set containing data from M users. The space of parameters (C, γ) of the RBF kernel is divided by a regular grid. For each couple of parameters the training set is divided into two parts, one containing $M-1$ users for training and the other with the remaining user for validation and performance evaluation. The procedure is repeated M times changing the user in the validation set. The couple of parameters that gives the best accuracy on average is selected as the output of the grid search. Finally the SVM has been trained on all the M users of the training set with the optimal parameters.

Alternatively we also tested a second classification scheme exploiting Random Forests (RF) [9]. Each tree has been trained on a random sampling of the training set leaving out one third of the sampled vectors for the estimation of the out-of-bag error. The only model parameter to optimize, differently from the pair for the RBF kernel of SVM, is the size m of the feature subset in each node. The parameter controls a trade-off between the tree correlation and the predictive "strength" of each tree, and may be easily found by analyzing the out-of-bag error. The size of the forest, is not a critical parameter since the classification error remains relatively stable if a sufficient number of trees is used. In our case we trained a Random Forest of 100 decision trees with a default value of $m = \sqrt{|\mathbf{F}|}$ with $|\mathbf{F}|$ the length of the feature vectors in the dataset ($|\mathbf{F}| = 435$ when all the considered features are

used). The implementation of the Random Forest classifier provided by Matlab has been used.

Finally, since the considered vectors contain a large number of elements we also considered the use of feature selection schemes to reduce the number of features and avoid the usage of useless or redundant descriptors. Three different feature selection schemes have been tested. The first uses the *F-score* approach [19], i.e., the F-score is computed for each feature and the most discriminative features according to this measure are selected (i.e., the features with an F-score bigger than a pre-defined threshold). Two different thresholds have been used to produce two subsets with 16 and 128 features.

The second scheme is based on the Forward Sequential Selection (FSS) algorithm [2]. In this case, starting from the empty set, at each step a new feature is added to the selected ones by choosing the one that allows to obtain the larger improvement in the classification accuracy with respect to the previous step (the SVM classifier previously described has been used to evaluate the classification accuracy).

Finally a third feature selection scheme exploiting Random Forests has been tested. In this case a classification is performed with the approach of [9] and the out-of-bag error is estimated. Then, to measure the importance of the various features, the values of one of the features are permuted and the out-of-bag error is estimated again. The procedure is repeated for each feature and the importance of each feature is given by the normalized average increase of the out-of-bag error after the permutation. This approach is detailed in [19]. The number of selected features is the same of the previous cases to allow a fair comparison.

5.9 Experimental results

The results have been obtained using the setup depicted in Figure 5.2. A Leap Motion device and a Kinect^{TM} v1 have been used to jointly acquire the data relative to the performed gestures. Any other depth camera can be used in the proposed approach. The two devices have been jointly calibrated using the approach of Section 5.3 and synchronized in time. A software synchronization scheme has been used: its precision is sufficient for the recognition of gestures based on static poses like the ones considered in this chapter. The considered dataset of gestures contains the 10 different gestures shown in Figure 5.13 executed by 14 different people. Each user has repeated each gesture 10 times for a total of 1400 different data samples. Up to our knowledge this is the first database containing both depth data and Leap Motion data and it is available on our website at the

url `http://lttm.dei.unipd.it/downloads/gesture`. To compute the results we split the dataset in a train and a test set by using the *leave-one-person-out* approach of Section 5.8, i.e., we placed in the training set the data from all the users except one and in the test set the data from the remaining user. Since the amount of data associated to a single user (100 samples) is not sufficient for a reliable assessment of the performances we executed 14 completely independent tests changing each time the person in the test set, i.e., as shown in Figure 5.12, in each test we used a train set with 13 people and a test set with a single person that is the remaining one. The results of the 14 tests have been averaged to obtain the final accuracy. Note that this is a more challenging test than the standard *leave-one-out* approach, since not only it guarantees that the data in the train set is different from the ones in the test set as in the standard case, but also that the train set does not contain any sample from the user in the test set. This means that the system should be able to classify the data from the user in the test set from what it has *learned* from users different from the one that is using it, a typical situation in real setups. This approach has been used to train both classifiers, i.e., the Support Vector Machines (SVM) one and the one exploiting Random Forests (RF) as explained in Section 5.8. In this section we will firstly report the performance that can be obtained by using the SVM classifier (that is the better performing one) with the various feature types of each of the two sensors alone. Then the results that can be obtained by combining the two sensors will be presented. Finally we will show the accuracy that can be obtained with various combinations of classifiers (SVM or RF) and of feature selection strategies.

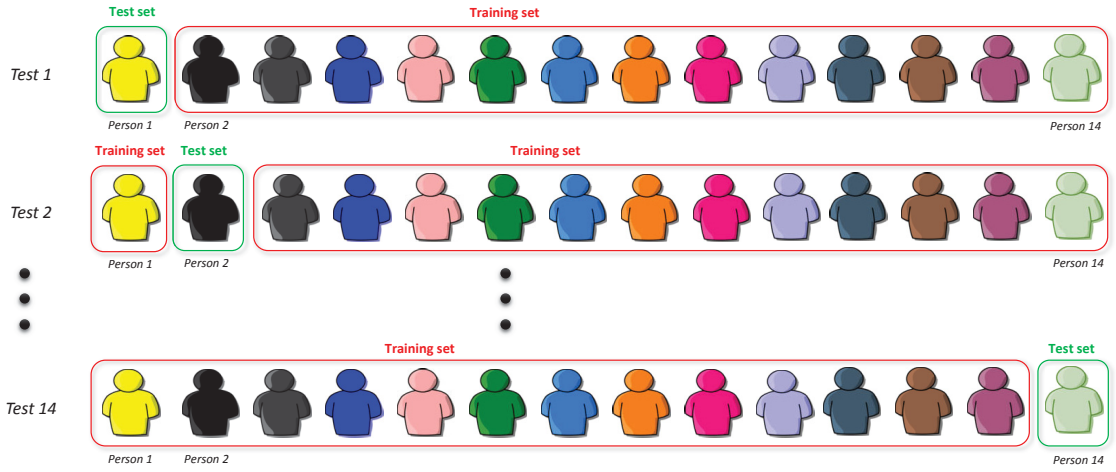


Figure 5.12: The results are the average of 14 independent tests each one performed by placing a person in the test set and the remaining 13 in the train set.

Let us start from the Leap Motion device. Table 5.1 shows the accuracy obtained

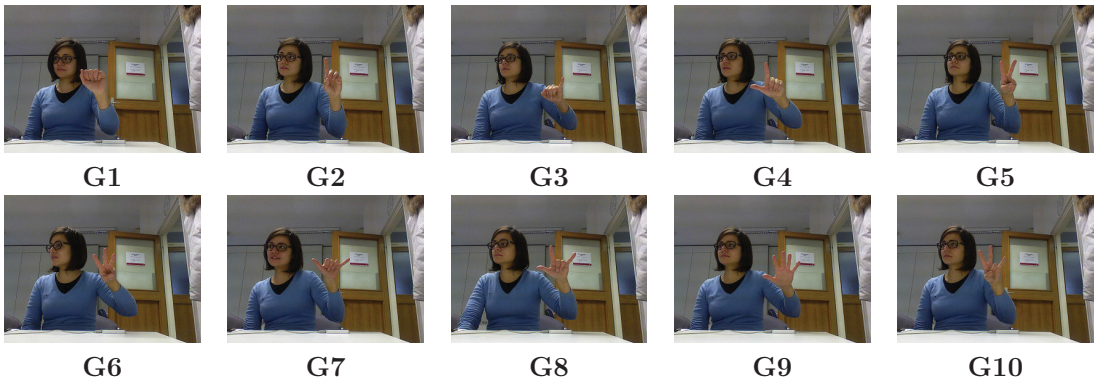


Figure 5.13: Gestures from the American Sign Language (ASL) contained in the database that has been acquired for experimental results.

using the classification algorithm of Section 5.8 on the data from this sensor. The 3D positions of the fingertips give a very good representation of the arrangement of the fingers and allow to obtain an accuracy of 81.5%. They allow to recognize the majority of the gestures even if the recognition of some gestures is not always optimal, as it is possible to see from the confusion matrix in Table 5.2. For example, gestures G2 and G3 are sometimes confused with gesture G1. This is due mostly to the false positives returned by the Leap Motion sensor that sometimes detects a raised finger in gesture G1.

Feature set	**Accuracy**
Fingertips 3D positions ($\mathbf{F}^p$)	81.5%
Fingertips distances ($\mathbf{F}^d$)	76.1%
Fingertips angles ($\mathbf{F}^a$)	74.2%
Fingertips elevations ($\mathbf{F}^e$)	73.1%
$\mathbf{F}^d + \mathbf{F}^a + \mathbf{F}^e$	80.9%

Table 5.1: Performance with the Leap Motion data.

Fingertip distance features allow to obtain an accuracy of about 76%: they are able to recognize most gestures but there are some critical issues, e.g. G2 and G3 are easily confused. A relevant issue for this descriptor is the limited accuracy of the hand direction estimation from the Leap Motion that does not allow a precise match between the fingertips and the corresponding angular regions (i.e., it is not easy to recognize which finger has been raised if a single finger is detected). The other two features have slightly lower performance. The angles allow to obtain an accuracy of 74.2% and a similar result (73%) can be obtained from the elevations alone. The last three features can be combined together since they capture different properties of the fingers arrangement. Their combination leads to an accuracy of

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	**0.893**		0.021	0.064	0.021					
G2	0.300	**0.564**	0.136							
G3	0.143	0.093	**0.700**	0.043			0.021			
G4	0.029			**0.900**			0.050		0.007	0.014
G5	0.050	0.050	0.029	0.021	**0.757**	0.014	0.021	0.021		0.036
G6	0.007		0.029		0.029	**0.836**	0.014	0.014		0.071
G7	0.014		0.036	0.079			**0.814**	0.029	0.007	0.021
G8				0.036	0.029		0.029	**0.829**		0.079
G9		0.007		0.007				0.014	**0.971**	
G10			0.014		0.036	0.007	0.050	0.007		**0.886**

Table 5.2: Confusion matrix for the 3D positions from the Leap Motion data. *Yellow* cells represent true positive, while *gray* cells show false positive with failure rate greater than 5%.

almost 81%, better than any of the three features alone. This result is quite similar to the performance of the 3D positions, consistently with the fact that the two distances from the center and the plane, together with the angle can be viewed as a different representation of the position of a point in 3D space.

Results from the Leap Motion data are good but not completely satisfactory. Better results can be obtained from the depth data, that offers a more informative description of the arrangement of the hand in 3D space. Depth data contain the complete 3D structure of the hand but they also represent a lower-level scene description and a larger amount of processing is needed to extract the features from it.

Feature set	**Accuracy**
Distance features ($\mathbf{F}^l$)	94.4%
Correlations features ($\mathbf{F}^\rho$)	68.7%
Curvature features ($\mathbf{F}^c$)	86.2%
Convex Hull features($\mathbf{F}^{cc}$)	70.5%
$\mathbf{F}^l + \mathbf{F}^c$	96.35%

Table 5.3: Performance with the depth data.

Table 5.3 shows the results obtained from the depth information acquired with a Kinect. Distance features are the best performing descriptor and allow to obtain an accuracy of 94.4%, much higher than the one that can be obtained from the Leap Motion sensor. This descriptor alone allows to recognize all the gestures with an high accuracy.

Correlation features have lower performance (68.7%). This descriptor is also based on the distances of the hand samples from the hand centroid, but compared

to the distances they contain a less informative description (the feature vector size is also much smaller) that is not sufficient for an accurate recognition. However thanks to the small descriptor size and very fast computation time they still can be considered for applications where the running time and the memory footprint of the descriptors are critical.

Another very good descriptor is the curvature of the hand contour. It allows a correct recognition of 86.2% of the considered gestures. Only distance features outperforms this descriptor. It has also the advantage that it does not rely on the computation of the hand center and orientation, making it very useful in situations where an estimation of these parameters is difficult. Finally, the convex hull features have an accuracy of 70.5%, slightly better than the correlations even if not too impressive. Again its small size and simple computation makes this descriptor interesting when a trade-off between performance and accuracy is needed.

The combination of multiple descriptors allows to improve the performance, e.g., by combining the two best performing descriptors, distances and curvatures a quite impressive accuracy of 96.35% can be obtained as it is possible to see also from the corresponding confusion matrix (Table 5.4). This is an indication that the different descriptors capture different properties of the hand arrangement and contain complementary information.

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	**0.971**	0.021	0.007							
G2	0.007	**0.971**	0.021							
G3	0.007		**0.986**		0.007					
G4		0.036		**0.964**						
G5		0.007			**0.986**	0.007				
G6		0.036			0.036	**0.893**		0.014		0.021
G7			0.014				**0.986**			
G8				0.007		0.014		**0.964**	0.007	0.007
G9				0.007		0.007			**0.986**	
G10					0.007	0.043		0.021		**0.929**

Table 5.4: Confusion matrix for the combined use of distance and curvature descriptors from depth data. *Yellow* cells represent true positive.

Feature set	Accuracy
$\mathbf{F}^l + \mathbf{F}^c + \mathbf{F}^p$	96.5%

Table 5.5: Performance from the combined use of the two sensors.

Descriptors based on the Leap Motion data and on the depth data can also be combined together. In the last test we combined the 3D positions from the

Leap Motion with the two best descriptors from depth data, i.e., the distances and the curvatures. The obtained accuracy is 96.5% as shown in Table 5.5. The corresponding confusion matrix (Table 5.7) shows also how the recognition rate is very high for all the considered gestures. The improvement with respect to depth data alone is limited, as expected since the accuracy from the 3D positions of the Leap Motion is much lower. However consider that Leap Motion data are used also for the computation of the depth-based features (i.e., for the initial centroid and hand orientation) and allow to reduce the computational time as it will be shown at the end of this section. Furthermore Leap Motion data allow a more reliable extraction of the hand in some complex settings, a feature that is not possible to appreciate on the employed dataset. Finally the Leap Motion provides a few but very relevant features and allows to obtain a good accuracy with a smaller number of features with respect to the depth-based approach.

A comparison with [70], that presents an earlier version of this approach, shows how the proposed algorithm clearly outperform the previous method (see Table 5.6). By exploiting both sensors, the accuracy is 96.5% against 91.3% of the previous scheme, a quite relevant improvement. This result is mostly due to the improvement in the feature extraction scheme from depth data, that has an accuracy of 96.3% instead of 89.7% of the previous scheme. This proves the reliability of the new depth features extraction algorithm exploiting the Leap Motion data and a more refined distance features extraction scheme.

Feature set	Accuracy	
	Marin et Al. [70]	Proposed method
Leap Motion features	80.9%	**81.5**%
Kinect^{TM} v1 features	89.7%	**96.3**%
Leap Motion + Kinect^{TM} v1 features	91.3%	**96.5**%

Table 5.6: Comparison between the performances of the proposed approach and of [70].

In Section 5.8 a second classification scheme based on Random Forests has been presented. This approach is simple and fast and does not require the complex grid search procedure for the optimization of the parameters. On the other side this classifier has slightly lower performances than the SVM approach and with the complete feature set is able to achieve an accuracy of 94.7%, a very good result but about 2% lower than the one of the SVM classifier.

The proposed approach makes use of a large number of features, with the complete feature set each vector has 435 elements. Furthermore there is also a much larger number of feature values extracted from the Kinect^{TM} v1 data with respect to the ones from the Leap Motion. For these reasons it is reasonable to

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	**0.979**	0.021								
G2	0.014	**0.964**	0.021							
G3	0.007	0.007	**0.986**							
G4		0.029		**0.971**						
G5	0.007				**0.986**	0.007				
G6		0.029			0.043	**0.886**		0.007		0.036
G7			0.014				**0.986**			
G8				0.014		0.014		**0.957**	0.007	0.007
G9				0.007		0.007			**0.986**	
G10					0.007	0.029		0.014		**0.950**

Table 5.7: Confusion matrix for the combined use of Leap Motion and depth data. *Yellow* cells represent true positive.

employ a feature selection scheme to reduce the number of features and to better balance the information coming from the two sensors. As already described three different feature selection strategies have been tested, i.e., F-Score, Sequential Feature Selection and Random Forests. All the three methods have been tested both with the SVM and the RF classifier. For each combination of feature selection strategy and classifier we selected the 16 and 128 best features. The results are presented in Table 5.8. The table shows how by properly selecting the best features it is possible to greatly reduce the number of employed features with only a limited impact on the performances.

The F-Score feature selection method is the simplest and fastest but also the one leading to the worst results. In particular if the number of features is reduced to 128 (about one third of the original number of features), this approach is still able to achieve acceptable performances with a loss of about 2% on the accuracy of the SVM classifier. If the number of features is further reduced to 16 this approach is instead not able to properly select a good combination of features, mostly due to the fact that it does not properly account for the correlation between the different features. In this case there is a huge performance drop with an accuracy of 60%, more than 36% less than the one obtained with all the features. If the F-Score approach is used together with the Random Forests classifier the results are very similar with losses on the accuracy of 2.1% (128 features) and of 37.2% (16 features).

The sequential feature selection algorithm is instead the best performing one when the SVM classifier is used. The accuracy is very close to the original value with both 128 and 16 features. Even by using only 16 features the accuracy is only 0.7% less than the optimal value obtained by using all the features. This is a quite impressive result and opens the way to several optimization and simplification

strategies for the proposed approach. Results are very good also for the Random Forest classifier, the loss in this case is 0.6% with 128 features and 4% with 16 features. Notice how in this case the reduction to 16 features has a more noticeable impact.

Finally Random Forests can be used also for the feature selection. If they are used together with the SVM classifier the performances are very good but slightly worse than the ones of the sequential feature selection scheme, specially if 16 features are used. In this case there is a loss of about 3%, much better than the F-score but not so good as the sequential feature selection result. When, instead, Random Forests are used for both the feature selection and the classification, results are very similar to the sequential feature selection strategy (in fact even better although with a very small difference), according to the idea that having the same approach used for both steps also allows to simplify and speed-up the training procedure.

Concluding, the best solution for optimal performances is to use the Sequential Feature Selection scheme together with the SVM classifier. The Random Forests for both training and classification can be used when a simpler and faster training phase is needed.

	SVM			RF		
Feature selection strategy	435	128	16	435	128	16
F-Score	96.5	94.5%	60.1%	94.7	92.6%	57.5%
Sequential		95.9%	95.8%		94.1%	90.7%
Random Forests		95.8%	93.7%		94.2%	90.8%

Table 5.8: Performances with different combinations of classification algorithms and feature selection strategies.

Finally, notice how the proposed approach is particularly suitable for real time gesture recognition schemes. The current implementation in C++ (that has not been fully optimized) has been tested on a not too performing desktop PC with an Intel Q6600 processor and 4Gb of RAM and real-time performances have been obtained. The initial hand detection phase, that took $46ms$ in the implementation of the approach of [28] and that we used to start the development of this work can now be completed in a few milliseconds thanks to the exploitation of the Leap Motion centroid. Notice also that the processing of color data for the check on skin color compatibility has also been removed in this work since it was used only in the initial phase. The extraction of palm and fingers regions with the circle fitting requires about $25ms$. The orientation of the hand is also directly computed from the Leap Motion data (this step took about $4ms$ in the old approach). Feature extraction is quite fast, the most demanding ones are curvature descriptors that

take about 28 ms to be computed while the other features are way faster to be computed. Finally SVM classification is performed in just $1ms$. This allows to obtain a frame rate of about $15fps$ if depth data are used with respect to the $10fps$ achieved by the previous approach on the same computer. Gesture recognition with the Leap Motion data alone is very fast (just a few milliseconds) but performances are also lower.

Chapter 6

Conclusions

This thesis provides an overview of the research carried out during the three years of the Ph.D. program. The problem of fusing 3D data from multiple sensors has been studied under different aspects, from the acquisition of the data to the applications that are possible combining multiple sensors.

Chapter 2 describes the most common systems capable of producing depth data, in particular stereo cameras, structured light cameras, and ToF cameras. The operating principles and practical issues of these acquisition systems are described to provide solid foundations to the methods for fusing their data. For ToF cameras a unified framework is proposed by considering the internal components like a telecommunication system, where the transmitter converts an electrical signal to a NIR signal and the receiver correlates the demodulated signal to estimate the distance of the framed scene.

Fusion of 3D data from multiple sensors is described in Chapter 3. In the proposed approach data from a stereo vision system and a ToF depth camera are combined to provide a more accurate depth map. A set of confidence measures is computed for both stereo camera and ToF camera, the input depth maps are then fuse together enforcing the local consistency of depth data accounting for the confidence of the two systems at pixel level. Experimental results show the effectiveness of the proposed approach comparing the performance with state of the art methods. Another approach based on deep learning is currently under investigation, where a CNN is trained to combine depth maps and raw images from multiple sensors to produce a more accurate depth map.

The need of data from multiple sensors is of fundamental importance to the development of algorithms that fuse their data. Chapter 4 describes the setup used to collect data from a set of three different commercial acquisition systems: a stereo camera, a ToF camera and a structured light camera. The three acquisition systems

have been calibrated to provide the possibility of combining data from different cameras. In addition to the raw images from the sensors, also the ground truth depth map is estimated by using an ah hoc framework that computes the ground truth depth map from the same point of view of one of the cameras, using a line laser and the same cameras of the acquisition systems. In addition to real data collection, Chapter 4 describes a synthetic data generator that includes realistic models of color cameras and ToF cameras. When deep learning based approaches are used, the need of a big amount of data is crucial, and the proposed simulator fulfills this requirement by generating data that are comparable to those acquired with a real camera.

The use of multiple sensors is not limited to 3D fusion, Chapter 5 describes how to combine a depth camera with a Leap Motion device to boost the performance of gesture recognition. A set of novel descriptors is introduced for both the devices and a multi-class SVM classifier is trained to predict the performed gesture. A novel scheme for extraction and identification of palm and fingers from a single depth map is also presented. The density based clustering framework has been tested in a challenging dataset showing the effectiveness of the proposed method also in complex situations and in presence of occlusions.

Bibliography

[1] URL: http://riemenschneider.hayko.at/vision/dataset (cit. on p. 89).

[2] D. Aha and R. Bankert. "A Comparative Evaluation of Sequential Feature Selection Algorithms". In: *Learning from Data*. Ed. by D. Fisher and H.-J. Lenz. Vol. 112. Lecture Notes in Statistics. Springer New York, 1996, pp. 199–206 (cit. on p. 129).

[3] J. Andrews and N. Baker. "Xbox 360 System Architecture". In: *Micro, IEEE* 26.2 (2006), pp. 25–37 (cit. on pp. 38, 48).

[4] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys. "Motion Capture of Hands in Action using Discriminative Salient Points". In: *Proceedings of European Conference on Computer Vision (ECCV)*. Firenze, 2012 (cit. on p. 109).

[5] C. Bamji, P. O'Connor, T. Elkhatib, S. Mehta, B. Thompson, L. Prather, D. Snow, O. Akkaya, A. Daniel, A. Payne, T. Perry, M. Fenton, and V.-H. Chan. "A 0.13 um CMOS System-on-Chip for a 512x424 Time-of-Flight Image Sensor With Multi-Frequency Photo-Demodulation up to 130 MHz and 2 GS/s ADC". In: *Solid-State Circuits, IEEE Journal of* 50.1 (2015), pp. 303–319 (cit. on p. 38).

[6] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar. "Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization". In: *Opt. Lett.* 39.6 (2014), pp. 1705–1708. URL: http://ol.osa.org/abstract.cfm?URI=ol-39-6-1705 (cit. on p. 60).

[7] S. Birchfield and C. Tomasi. "Depth Discontinuities by Pixel-to-Pixel Stereo". In: *Int. Journal of Computer Vision* 35.3 (Dec. 1999), pp. 269–293 (cit. on p. 10).

[8] J. Bouguet, B. Curless, P. Debevec, M. Levoy, S. Nayar, and S. Seitz. *Overview of active vision techniques. SIGGRAPH 2000 Course on 3D Photography*. Workshop. 2000 (cit. on p. 5).

[9] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32 (cit. on pp. 128, 129).

[10] B. Buttgen and P. Seitz. "Robust Optical Time-of-Flight Range Imaging Based on Smart Pixel Structures". In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 55.6 (2008), pp. 1512–1525 (cit. on pp. 54, 57, 60, 62).

[11] B. Buttgen, T. Oggier, M. Lehmann, R. Kaufmann, and F. Lustenberger. "CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art". In: *1st range imaging research day*. 2005 (cit. on pp. 56, 57, 62).

[12] T. I. Cerlinca and S. G. Pentiuc. "Robust 3D Hand Detection for Gestures Recognition". English. In: *Intelligent Distributed Computing V*. Ed. by F. Brazier, K. Nieuwenhuis, G. Pavlin, M. Warnier, and C. Badica. Vol. 382. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2012, pp. 259–264. URL: http://dx.doi.org/10.1007/978-3-642-24013-3_27 (cit. on p. 108).

[13] *cgtrader*. https://www.cgtrader.com (cit. on p. 101).

[14] C.-C. Chang and C.-J. Lin. "LIBSVM: A library for support vector machines". In: *ACM Trans. on Intelligent Systems and Technology* 2 (3 2011), 27:1–27:27 (cit. on p. 128).

[15] D. Claus and A. Fitzgibbon. "A Rational Function Lens Distortion Model for General Cameras". In: *CVPR*. 2005 (cit. on p. 51).

[16] D. Comaniciu and P. Meer. "Mean shift: a robust approach toward feature space analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002) (cit. on p. 71).

[17] R. Crabb and R. Manduchi. "Fast single-frequency time-of-flight range imaging". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015, pp. 58–65 (cit. on p. 53).

[18] *CV papers*. http://www.cvpapers.com/datasets.html. URL: http://www.cvpapers.com/datasets.html (cit. on p. 89).

[19] M. Dahan, N. Chen, A. Shamir, and D. Cohen-Or. "Combining color and depth for enhanced image segmentation and retargeting". In: *The Visual Computer* 28.12 (2012), pp. 1181–1193 (cit. on p. 129).

[20] C. Dal Mutto, P. Zanuttigh, and G. Cortelazzo. "Probabilistic ToF and Stereo Data Fusion Based on Mixed Pixels Measurement Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.11 (2015), pp. 2260–2272 (cit. on pp. 69, 73, 80, 83–87, 89).

[21] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo. "A probabilistic approach to tof and stereo data fusion". In: *3DPVT, Paris, France* 2 (2010) (cit. on p. 69).

[22] C. Dal Mutto, P. Zanuttigh, S. Mattoccia, and G. Cortelazzo. "Locally consistent tof and stereo data fusion". In: *ECCV Workshop on Consumer Depth Cameras for Computer Vision*. Springer, 2012, pp. 598–607 (cit. on pp. 69, 70, 79, 83–87, 89).

[23] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win. "Ranging With Ultrawide Bandwidth Signals in Multipath Environments". In: *Proceedings of the IEEE* 97.2 (2009), pp. 404–426 (cit. on p. 36).

[24] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. "Spacetime stereo: A unifying framework for depth from triangulation". In: *CVPR*. 2003 (cit. on pp. 14, 16).

[25] J. Diebel and S. Thrun. "An Application of Markov Random Fields to Range Sensing". In: *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2005 (cit. on p. 69).

[26] Z. Ding, Z. Zhang, Y. Chen, Y. L. Chen, and X. Wu. "A real-time dynamic gesture recognition based on 3D trajectories in distinguishing similar gestures". In: *Information and Automation, 2015 IEEE International Conference on*. 2015, pp. 250–255 (cit. on p. 109).

[27] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. "Upsampling range data in dynamic environments". In: *Proceedings of CVPR*. 2010, pp. 1141–1148 (cit. on p. 69).

[28] F. Dominio, M. Donadeo, and P. Zanuttigh. "Combining multiple depth-based descriptors for hand gesture recognition". In: *Pattern Recognition Letters* 50 (2014), pp. 101–111 (cit. on pp. 108, 113, 119, 120, 124–126, 136).

[29] F. Dominio, G. Marin, M. Piazza, and P. Zanuttigh. "Feature Descriptors for Depth-Based Hand Gesture Recognition". English. In: *Computer Vision and Machine Learning with RGB-D Sensors*. Ed. by L. Shao, J. Han, P. Kohli, and Z. Zhang. Advances in Computer Vision and Pattern Recognition. Springer International Publishing, 2014, pp. 215–237. URL: `http://dx.doi.org/10.1007/978-3-319-08651-4_11` (cit. on p. 3).

[30] F. Dominio, M. Donadeo, G. Marin, P. Zanuttigh, and G. M. Cortelazzo. "Hand gesture recognition with depth data". In: *Proceedings of the 4th ACM/IEEE international workshop on Analysis and retrieval of tracked events and motion in imagery stream*. 2013, pp. 9–16 (cit. on pp. 3, 124).

[31] A. A. Dorrington, J. P. Godbaz, M. J. Cree, A. D. Payne, and L. V. Streeter. "Separating true range measurements from multi-path and scattering interference in commercial range cameras". In: vol. 7864. 2011, pp. 786404–786404–10. URL: http://dx.doi.org/10.1117/12.876586 (cit. on p. 60).

[32] S. R. et al. "Pseudo-noise (PN) laser radar without scanner for extremely fast 3D-imaging and navigation". In: *MIOP '97* (1997) (cit. on p. 38).

[33] G. Evangelidis, M. Hansard, and R. Horaud. "Fusion of Range and Stereo Data for High-Resolution Scene-Modeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.11 (2015), pp. 2178 –2192 (cit. on pp. 69, 89).

[34] A. Frick, F. Kellner, B. Bartczak, and R. Koch. "Generation of 3D-TV LDV-content with Time-Of-Flight Camera". In: *Proc. of 3DTV Conf.* 2009 (cit. on p. 69).

[35] V. Garro, P. Zanuttigh, and G. M. Cortelazzo. "A new super resolution technique for range data". In: *Proceedings of GTTI Meeting. Citeseer* (2009) (cit. on p. 69).

[36] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013) (cit. on p. 89).

[37] J. P. Godbaz, M. J. Cree, and A. A. Dorrington. "Closed-form inverses for the mixed pixel/multipath interference problem in AMCW lidar". In: vol. 8296. 2012, pp. 829618–829618–15. URL: http://dx.doi.org/10.1117/12.909778 (cit. on p. 60).

[38] J. Godbaz, A. Dorrington, and M. Cree. "Understanding and Ameliorating Mixed Pixels and Multipath Interference in AMCW Lidar". English. In: *TOF Range-Imaging Cameras*. Ed. by F. Remondino and D. Stoppa. Springer Berlin Heidelberg, 2013, pp. 91–116. URL: http://dx.doi.org/10.1007/978-3-642-27523-4_5 (cit. on p. 40).

[39] S. A. Gudmundsson, H. Aanaes, and R. Larsen. "Fusion of stereo vision and Time Of Flight imaging for improved 3D estimation". In: *Int. J. Intell. Syst. Technol. Appl.* 5 (2008), pp. 425–433 (cit. on pp. 68, 69).

[40] C. Guerrero-Rincon, A. Uribe-Quevedo, H. Leon-Rodriguez, and J.-O. Park. "Hand-based tracking animatronics interaction". In: *Robotics (ISR), 2013 44th International Symposium on*. 2013, pp. 1–3 (cit. on p. 109).

[41] S. Guomundsson, H. Aanaes, and R. Larsen. "Environmental Effects on Measurement Uncertainties of Time-of-Flight Cameras". In: *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on*. Vol. 1. 2007, pp. 1–4 (cit. on p. 54).

[42] M. Hansard, S. Lee, O. Choi, and R. Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. SpringerBriefs in Computer Science. Springer, 2013, p. 96 (cit. on p. 68).

[43] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004 (cit. on p. 13).

[44] J. Heikkila and O. Silven. "A Four-step Camera Calibration Procedure with Implicit Image Correction". In: *CVPR*. 1997 (cit. on p. 51).

[45] D. Herrera, J. Kannala, and J. Heikkilä. "Joint Depth and Color Camera Calibration with Distortion Correction". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34.10 (2012), pp. 2058–2064 (cit. on p. 112).

[46] h. Hirschmuller. "Stereo Processing by Semi-Global Matching and Mutual Information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2 2008), pp. 328–341 (cit. on pp. 9, 74).

[47] X. Hu and P. Mordohai. "A Quantitative Evaluation of Confidence Measures for Stereo Vision". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012), pp. 2121–2133 (cit. on pp. 69, 76, 83, 86).

[48] *IEE*. http://www.iee.lu (cit. on pp. 35, 38).

[49] M. Imaging. *SR4000 user manual*. http://www.mesa-imaging.ch (cit. on p. 64).

[50] *Intel RealSense F200*. https://software.intel.com/en-us/realsense/f200camera (cit. on p. 16).

[51] *Intel RealSense R200*. https://software.intel.com/en-us/realsense/r200camera (cit. on p. 16).

[52] A. Kadambi, R. Whyte, A. Bhandari, L. Streeter, C. Barsi, A. Dorrington, and R. Raskar. "Coded Time of Flight Cameras: Sparse Deconvolution to Address Multipath Interference and Recover Time Profiles". In: *ACM Trans. Graph.* 32.6 (Nov. 2013), 167:1–167:10. URL: `http://doi.acm.org/10.1145/2508363.2508428` (cit. on p. 60).

[53] T. Kahlmann and H. Ingensand. "Calibration and development for increased accuracy of 3D range imaging cameras". In: *Journal of Applied Geodesy* 2 (2008), pp. 1–11 (cit. on pp. 54, 68).

[54] M. Keller, J. Orthmann, A. Kolb, and V. Peters. "A Simulation Framework for Time-Of-Flight Sensors". In: *2007 International Symposium on Signals, Circuits and Systems*. Vol. 1. 2007, pp. 1–4 (cit. on p. 99).

[55] C. Keskin, F. Kirac, Y. Kara, and L. Akarun. "Real time hand pose estimation using depth sensors". In: *ICCV Workshops*. 2011, pp. 1228 –1234 (cit. on p. 108).

[56] Y. Kim, C. Theobald, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun. "Multi-view Image and ToF Sensor Fusion for Dense 3D Reconstruction". In: *Proc. of 3DIM Conf.* 2009 (cit. on p. 69).

[57] A. Kirmani, A. Benedetti, and P. Chou. "SPUMIC: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods". In: *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. 2013, pp. 1–6 (cit. on p. 60).

[58] K. Konoldige and P. Mihelich. *Technical description of Kinect calibration*. Tech. rep. http://www.ros.org/wiki/kinect_calibration/technical: Willow Garage, 2011 (cit. on pp. 24, 26).

[59] K. Konolige. "Projected texture stereo". In: *ICRA* (2010) (cit. on p. 14).

[60] J. F. Kooij. "SenseCap: Synchronized Data Collection with Microsoft Kinect2 and LeapMotion". In: *Proceedings of the 2016 ACM on Multimedia Conference*. MM '16. Amsterdam, The Netherlands: ACM, 2016, pp. 1218–1221. URL: http://doi.acm.org/10.1145/2964284.2973805 (cit. on p. 109).

[61] K.-D. Kuhnert and M. Stommel. "Fusion of Stereo-Camera and PMD-Camera Data for Real-Time Suited Precise 3D Environment Reconstruction". In: *Proc. of Int. Conf. on Intelligent Robots and Systems*. 2006, pp. 4780 –4785 (cit. on p. 69).

[62] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares. "Leafsnap: A Computer Vision System for Automatic Plant Species Identification". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2012 (cit. on p. 126).

[63] A. Kurakin, Z. Zhang, and Z. Liu. "A Real-Time System for Dynamic Hand Gesture Recognition with a Depth Sensor". In: *Proc. of EUSIPCO*. 2012 (cit. on p. 108).

[64] E. Lachat, H. Macher, M.-A. Mittet, T. Landes, and P. Grussenmeyer. "First Experiences with Kinect v2 Sensor for Close Range 3d Modelling". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Feb. 2015), pp. 93–100 (cit. on p. 93).

[65] R. Lange. "3D Time-Of-Flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology". PhD thesis. University of Siegen, 2000 (cit. on pp. 38, 40, 44, 62).

[66] D. Lefloch, R. Nair, F. Lenzen, H. Schäfer, L. Streeter, M. Cree, R. Koch, and A. Kolb. "Technical Foundation and Calibration Methods for Time-of-Flight Cameras". English. In: *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*. Ed. by M. Grzegorzek, C. Theobalt, R. Koch, and A. Kolb. Vol. 8200. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 3–24. URL: http://dx.doi.org/10.1007/978-3-642-44964-2_1 (cit. on p. 93).

[67] X. Liu and K. Fujimura. "Hand gesture recognition using depth data". In: *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. 2004, pp. 529–534 (cit. on p. 108).

[68] S. Manay, D. Cremers, B.-W. Hong, A. Yezzi, and S. Soatto. "Integral Invariants for Shape Matching". In: *IEEE Trans. on PAMI* 28.10 (2006), pp. 1602 –1618 (cit. on p. 126).

[69] T. Mantecon, C. R. del Blanco, F. Jaureguizar, and N. Garcia. "Hand Gesture Recognition Using Infrared Imagery Provided by Leap Motion Controller". In: *Advanced Concepts for Intelligent Vision Systems: 17th International Conference, ACIVS 2016, Lecce, Italy, October 24-27, 2016, Proceedings*. Ed. by J. Blanc-Talon, C. Distante, W. Philips, D. Popescu, and P. Scheunders. Cham: Springer International Publishing, 2016, pp. 47–57. URL: http://dx.doi.org/10.1007/978-3-319-48680-2_5 (cit. on p. 109).

[70] G. Marin, F. Dominio, and P. Zanuttigh. "Hand gesture recognition with Leap Motion and Kinect devices". In: *Proceedings of IEEE International Conference on Image Processing (ICIP)*. Paris, France, 2014 (cit. on pp. 3, 134).

[71] G. Marin, M. Fraccaro, M. Donadeo, F. Dominio, and P. Zanuttigh. "Palm area detection for reliable hand gesture recognition". In: *Proceedings of MMSP 2013*. 2013 (cit. on p. 120).

[72] G. Marin, F. Dominio, and P. Zanuttigh. "Hand gesture recognition with jointly calibrated Leap Motion and depth sensor". English. In: *Multimedia Tools and Applications* (2015), pp. 1–25. URL: http://dx.doi.org/10.1007/s11042-015-2451-6 (cit. on p. 3).

[73] G. Marin, P. Zanuttigh, and S. Mattoccia. "Reliable Fusion of ToF and Stereo Depth Driven by Confidence Measures". In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on pp. 2, 68).

[74] S. Mattoccia. "A locally global approach to stereo correspondence". In: *Proc. of 3DIM*. 2009 (cit. on pp. 2, 69, 70, 77, 78, 80).

[75] S. Mattoccia. "Fast locally consistent dense stereo on multicore". In: *6th IEEE Embedded Computer Vision Workshop (CVPR Workshop)*. San Francisco, USA, 2010 (cit. on p. 80).

[76] *Mesa Imaging*. http://www.mesa-imaging.ch (cit. on pp. 35, 38).

[77] Microsoft. *Kinect*. http://www.xbox.com/en-US/kinect. 2012 (cit. on pp. 35, 38, 63).

[78] L. Minto, G. Marin, and P. Zanuttigh. "3D hand shape analysis for palm and fingers identification". In: *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. Vol. 07. 2015, pp. 1–6 (cit. on pp. 3, 121).

[79] M. Mohandes, S. Aliyu, and M. Deriche. "Arabic sign language recognition using the leap motion controller". In: *Industrial Electronics (ISIE), IEEE 23rd International Symposium on*. 2014, pp. 960–965 (cit. on p. 109).

[80] A. Motten, L. Claesen, and Y. Pan. "Trinocular disparity processor using a hierarchic classification structure". In: *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*. 2012, pp. 247–250 (cit. on p. 70).

[81] F. Mufti and R. Mahony. "Statistical analysis of signal measurement in time-of-flight cameras". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2011) (cit. on pp. 44, 56, 62).

[82] R. Nair, F. Lenzen, S. Meister, H. Schaefer, C. Garbe, and D. Kondermann. "High Accuracy ToF and Stereo Sensor Fusion At Interactive Rates". In: *Proceedings of 2nd Workshop on Consumer Depth Cameras for Computer Vision*. 2012 (cit. on pp. 69, 89).

[83] R. Nair, K. Ruhl, F. Lenzen, S. Meister, H. Schäfer, C. Garbe, M. Eisemann, M. Magnor, and D. Kondermann. "A Survey on Time-of-Flight Stereo Fusion". In: *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*. Ed. by M. Grzegorzek, C. Theobalt, R. Koch, and A. Kolb. Vol. 8200. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 105–127 (cit. on p. 69).

[84] I. Nigam, M. Vatsa, and R. Singh. "Leap Signature Recognition using HOOF and HOT features". In: *Proceedings of IEEE International Conference on Image Processing (ICIP)*. Paris, France, 2014 (cit. on p. 109).

[85] F. Pedersoli, N. Adami, S. Benini, and R. Leonardi. "XKin - eXtendable hand pose and gesture recognition library for Kinect". In: *In: Proceedings of ACM Conference on Multimedia 2012 - Open Source Competition*. Nara, Japan, 2012 (cit. on pp. 108, 126).

[86] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui. "Towards a simulation driven stereo vision system". In: *Pattern Recognition (ICPR), 2012 21st International Conference on*. 2012, pp. 1038–1042 (cit. on p. 99).

[87] D. Piatti and F. Rinaudo. "SR-4000 and CamCube3.0 Time of Flight (ToF) Cameras: Tests and Comparison". In: *Remote Sensing* 4.4 (2012), pp. 1069–1089 (cit. on p. 68).

[88] *PMD Technologies*. http://www.pmdtec.com/ (cit. on pp. 35, 38).

[89] L. E. Potter, J. Araullo, and L. Carter. "The Leap Motion Controller: A View on Sign Language". In: *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. OzCHI '13. Adelaide, Australia: ACM, 2013, pp. 175–178 (cit. on p. 109).

[90] *Primesense*. http://www.primesense.com/ (cit. on p. 15).

[91] F. Remondino and D. Stoppa, eds. *TOF Range-Imaging Cameras*. Springer, 2013, p. 240 (cit. on pp. 35, 36, 38, 68, 72).

[92] Z. Ren, J. Meng, and J. Yuan. "Depth camera based hand gesture recognition and its applications in Human-Computer-Interaction". In: *Proc. of ICICS*. 2011, pp. 1 –5 (cit. on p. 108).

[93] Z. Ren, J. Yuan, and Z. Zhang. "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera". In: *Proc. of ACM Conference on Multimedia*. ACM, 2011, pp. 1093–1096 (cit. on p. 108).

[94] M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G. J. Brostow. "Capturing Time-of-Flight Data with Confidence". In: *CVPR*. 2011 (cit. on p. 61).

[95] A. Sabov and J. Krüger. "Identification and Correction of Flying Pixels in Range Camera Data". In: *Proceedings of the 24th Spring Conference on Computer Graphics*. SCCG '08. Budmerice, Slovakia: ACM, 2010, pp. 135–142. URL: http://doi.acm.org/10.1145/1921264.1921293 (cit. on p. 61).

[96] J. Salvi, J. Pagès, and J. Batlle. "Pattern Codification Strategies in Structured Light Systems". In: *Pattern Recognition* 37 (2004), pp. 827–849 (cit. on p. 17).

[97] D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". In: *International Journal of Computer Vision* (2001) (cit. on pp. 9, 89).

[98] D. Scharstein and R. Szeliski. "High-accuracy Stereo Depth Maps Using Structured Light". In: *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR'03. Madison, Wisconsin: IEEE Computer Society, 2003, pp. 195–202. URL: http://dl.acm.org/citation.cfm?id=1965841.1965865 (cit. on p. 95).

[99] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth". In: *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings*. Ed. by X. Jiang, J. Hornegger, and R. Koch. Cham: Springer International Publishing, 2014, pp. 31–42. URL: http://dx.doi.org/10.1007/978-3-319-11752-2_3 (cit. on p. 89).

[100] M. Schmidt and B. Jähne. "A Physical Model of Time-of-Flight 3D Imaging Systems, Including Suppression of Ambient Light". In: *Proceedings of the DAGM 2009 Workshop on Dynamic 3D Imaging*. Dyn3D '09. Jena, Germany: Springer-Verlag, 2009, pp. 1–15. URL: http://dx.doi.org/10.1007/978-3-642-03778-8_1 (cit. on p. 99).

[101] S. Schwarz, M. Sjostrom, and R. Olsson. "Time-of-flight sensor fusion with depth measurement reliability weighting". In: *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*. 2014, pp. 1–4 (cit. on p. 69).

[102] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *CVPR*. 2011 (cit. on p. 109).

[103] *Sketchfab*. https://sketchfab.com (cit. on p. 101).

[104] *Stereolabs*. https://www.stereolabs.com (cit. on p. 90).

[105] P. Suryanarayan, A. Subramanian, and D. Mandalapu. "Dynamic Hand Pose Recognition Using Depth Data". In: *Proc. of ICPR*. 2010, pp. 3105 –3108 (cit. on p. 108).

[106] R. Szeliski. *Computer Vision: Algorithms and Applications*. New York: Springer, 2010 (cit. on p. 9).

[107] B. Tippetts, D. Lee, K. Lillywhite, and J. Archibald. "Review of stereo vision algorithms and their suitability for resource-limited systems". In: *Journal of Real-Time Image Processing* (2013), pp. 1–21 (cit. on p. 69).

[108] M. Trobina. *Error Model of a Coded-Light Range Sensor*. Tech. rep. Communication Technology Laboratory Image Science Group, ETH-Zentrum, Zurich, 1995 (cit. on p. 13).

[109] C. Uriarte, B. Scholz-Reiter, S. Ramanandan, and D. Kraus. "Modeling Distance Nonlinearity in ToF Cameras and Correction Based on Integration Time Offsets". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer Berlin / Heidelberg, 2011 (cit. on pp. 54, 55).

[110] A. Verri and V. Torre. "Absolute depth estimate in stereopsis". In: *Journal of the Optical Society of America A* 3.3 (1986), pp. 297–299. URL: `http://josaa.osa.org/abstract.cfm?URI=josaa-3-3-297` (cit. on p. 8).

[111] S. Vikram, L. Li, and S. Russell. "Handwriting and Gestures in the Air, Recognizing on the Fly". In: *ACM Conference on Human Factors in Computing Systems (CHI) Extended Abstracts*. 2013 (cit. on p. 109).

[112] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. "Robust 3D Action Recognition with Random Occupancy Patterns". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2012 (cit. on p. 108).

[113] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. "Analysis of the Accuracy and Robustness of the Leap Motion Controller". In: *Sensors* 13.5 (2013), pp. 6380–6393 (cit. on p. 115).

[114] R. Whyte, L. Streeter, M. Cree, and A. Dorrington. "Review of methods for resolving multi-path interference in Time-of-Flight range cameras". In: *SENSORS, 2014 IEEE*. 2014, pp. 629–632 (cit. on p. 60).

[115] Z. Xu. *Investigation of 3D-imaging Systems Based on Modulated Light and Optical RF-interferometry (ORFI)*. Shaker Verlag GmbH, Germany, 1999 (cit. on p. 38).

[116] Q. Yang, N. Ahuja, R. Yang, K. Tan, J. Davis, B. Culbertson, J. Apostolopoulos, and G. Wang. "Fusion of Median and Bilateral Filtering for Range Image Upsampling". In: *Image Processing, IEEE Transactions on* (2013) (cit. on p. 69).

[117] Q. Yang, R. Yang, J. Davis, and D. Nistér. "Spatial-Depth Super Resolution for Range Images". In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. 2007 (cit. on pp. 69, 83–87).

[118] P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo. *Time-of-Flight and Structured Light Depth Cameras: Technology and Applications*. 1st ed. Springer International Publishing, 2016. URL: `http://www.springer.com/book/9783319309712` (cit. on pp. 4, 18, 68, 69).

[119] Z. Zhang. "A Flexible New Technique for Camera Calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (1998), pp. 1330–1334 (cit. on pp. 92, 93).

[120] J. Zhu, L. Wang, R. Yang, and J. Davis. "Fusion of time-of-flight depth and stereo for high accuracy depth maps". In: *CVPR*. 2008 (cit. on pp. 69, 83–87).

[121] J. Zhu, L. Wang, R. Yang, J. Davis, and Z. Pan. "Reliability Fusion of Time-of-Flight Depth and Stereo Geometry for High Quality Depth Maps". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (2011), pp. 1400–1414 (cit. on p. 69).

[122] J. Zhu, L. Wang, J. Gao, and R. Yang. "Spatial-Temporal Fusion for High Accuracy Depth Maps Using Dynamic MRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010), pp. 899–909 (cit. on p. 69).