# Towards a Complete Pipeline for 3DS Conversion of Monocular Uncalibrated Images

Ph.D. Thesis

Candidate:
Francesco Malapelle

Advisors:
Prof. Andrea Fusiello
Dr. Pasqualina Fragneto
Dr. Beatrice Rossi

March 8, 2016

# Contents

# Chapter 1

# Introduction

The first chapter of the thesis is devoted to the definition of where our work lies. We will begin by providing a brief description of the creation and development of 3D content through the past years, starting from the first examples of stereoscopic images, in Section 1.1, up to recent popularity of 3D video within the movie industry in Section 1.2. We then introduce, in Section 1.3, the topic of 3D stereo (3DS) conversion, i.e. the process of converting 2D images to 3D stereoscopic images, and motivate the need for more studies on this topic. Then in Section 1.4 we describe different family of approaches from a high-level point of view and narrow it down to our field of interest. At last in Section 1.5 we explain the structure of the thesis and highlight its contributions in Section 1.6.

## 1.1 Stereoscopy

Stereoscopy is a technique for creating or enhancing the illusion of depth in an image by means of stereopsis for binocular vision. The word stereoscopy derives from Greek *stereos*, meaning "firm, solid", and *skopeo*, meaning "to look, to see". Any stereoscopic image is called a stereogram. Originally, stereogram referred to a pair of stereo images which could be viewed using a stereoscope.

Most stereoscopic methods present two offset images separately to the left and right eye of the viewer. These two-dimensional images are then combined in the brain to give the perception of 3D depth.

Human vision, including the perception of depth, is a complex process which only begins with the acquisition of visual information taken in through the eyes. Much processing ensues within the brain, as it processes the raw information provided.

One of the very important visual functions that occurs within the brain as it interprets what the eyes see is that of assessing the relative distances of various objects from the viewer, and the depth dimension of those same perceived objects. More specifically the brain makes use of a number of cues to determine relative distances and depth in a perceived scene, including: stereopsis, accommodation of the eye (focus), overlapping of one object by another, subtended visual angle of an object of known size, linear perspective (convergence of parallel edges), vertical position (objects higher in the scene generally tend to be perceived as further away), haze, desaturation and change in size of textured pattern detail. Most of this cues are already present in 2D images except for the first two.

Traditional stereoscopic photography consists of creating a 3D illusion starting from a pair of 2D images, a stereogram. The easiest way to enhance depth perception in the brain is to provide the eyes of the viewer with two different images, representing two perspectives of the same object, with a minor deviation equal or nearly equal to the perspectives that both eyes naturally receive in binocular vision.

## 1.2 3D films: some history

In the context of movie production industry, a 3D stereoscopic film is a motion picture that enhances the illusion of depth perception, hence adding a third dimension. The most common approach to the production of 3D films is derived from stereoscopic photography. In it, a regular motion picture camera system is used to record the images as seen from two perspectives or computer-generated imagery generates the two perspectives in post-production. Special projection or display hardware and/or special eyewear are used to provide the illusion of depth when viewing the film. Nowadays 3D films are not limited to feature film theatrical releases; television broadcasts and direct-to-video films have also incorporated similar methods, especially since the advent of 3D televisions. 3D films have existed in some form since 1915, but had been largely relegated to a niche in the motion picture industry because of the costly hardware and processes required to production and display, and the lack of a standardized format for all segments of the entertainment business. Nonetheless, 3D films were prominently featured in the 1950s in American cinema, and later experienced a worldwide resurgence in the 1980s and 1990s driven by animated movie themed-venues and IMAX high-end theaters. In particular, in the mid-1980s, IMAX began producing non-fiction films for its nascent 3D business and a key point was that its productions emphasized mathematical correctness of the 3D rendition

Figure 1.1: audience wearing special glasses watch a 3D stereoscopic film at the Telekinema on the South Bank in London during the Festival of Britain 1951 – OGL (`http://goo.gl/WQOkoF`).

and thus largely eliminated the eye fatigue and pain that resulted from the approximate geometries of previous 3D incarnations. 3D films went on to become more and more successful throughout the 2000s, culminating in the unprecedented success of 3D presentations of movies like "Avatar" in 2009.

## 1.3 3DS conversion

3DS conversion is the process of transforming 2D images to a 3D Stereo form, i.e. creating imagery for each eye from one 2D image. With the expansion of 3D market, 3DS conversion has become more common, also thanks to the fact that even in the case of native 3D movies there is the need to convert a large quantity of video material. A big portion of 3D blockbusters still are converted fully or at least partially from 2D footage. The reasons for shooting in 2D and convert afterwards are financial, technical and sometimes artistic: stereoscopic rigs are much more expensive and bulky than customary monocular cameras, thus some shots, can be only shot with relatively small 2D cameras. Moreover stereo cameras can introduce various mismatches in stereo images (such as vertical parallax, tilt, color shift, reflections and glares in different positions) that should be fixed in post-production anyway because they ruin the 3D effect, this correction sometimes may have

complexity comparable to stereo conversion. Also, stereo cameras can betray practical effects used during filming, such as using forced perspective to allow two actors to appear to be different physical sizes or shooting from a distance using zoom lenses. These are all factors that make stereo native capture extremely difficult, according to many filmmakers. Thus, even in the case of stereo shooting there may well be a need to convert some footage and that high quality conversion is an important tool in the box of any effects house.

Moreover, with the lack of stereo content, 3DS conversion is the only way to meet demands of the market, since it is needed for converting older popular films.

## 1.4 Approaches

3DS conversion approaches go from simple tricks like the exploitation of the *Pulfrich effect* (when lateral motion of an object is perceived as having a depth component, due to a relative difference in signal timings between the two eyes), to *homemade* 3DS conversion methods that can be found on *Youtube* tutorials and up to high quality semiautomatic conversion for cinema. The price of high quality stereo conversion is estimated at tens of thousands of dollars per minute, mainly because a lot of the work has to be done manually and frame-by-frame and a lot of the actual conversion relies on image processing and editing more than geometry.

From the computer vision community's point of view, the problem of 3DS conversion falls within the *View-Synthesis* (VS) or *Image Based Rendering* (IBR) family, i.e. the generation of novel images as if they were captured from virtual viewpoints, starting from a set of actual images or frames.

The rendering of virtual images is based on geometric relationships that are found between the positions of pixels representing the same point in the scene observed from different viewpoints. The procedure requires some geometry information, either explicit (depth) or implicit (depth-proxies), and suitable warping functions. In Figure 1.2, we report a basic representation of this family of techniques.

Both the computation of the depth-proxy and the warping function, are key ingredients for the quality of the rendered view. Moreover the goal-application, i.e. 3DS conversion of generic footage, implies the the necessity to work in a uncalibrated environment, i.e. without the knowledge of neither the camera's internal parameters and the camera position. The key advantage of uncalibrated view-synthesis (UVS), is the possibility to perform view-synthesis without any knowledge on the imaging device nor the

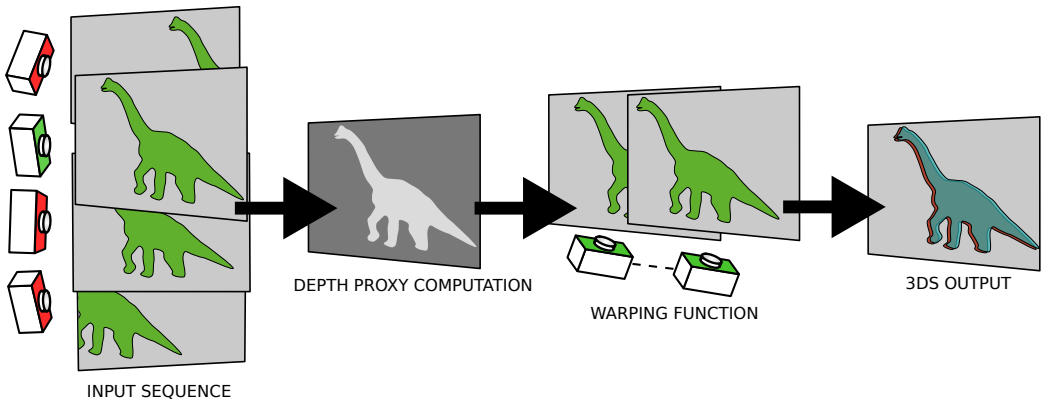


Figure 1.2: a pictorial representation of a 3DS conversion pipeline.

need of any kind of user interaction including manual camera calibration, but, as argued in the next sections, is challenging for several reasons.

## 1.5 Thesis structure

The rest of this thesis is structured as follows. Chapter 2 presents an overview of the background knowledge that is needed to understand the following parts but also reports the first theoretical contributions that we propose.

We first focused on the computation of the depth-proxy, this part of our work is presented in Chapter 3. We developed a data-fusion framework that locates itself among *Motion-stereo* or *Dynamic-stereo* techniques. The reason for this choice is that this kind of framework allowed us to exploit the information coming from the abundance of images depicting the same portion of the scene from different perspectives. More precisely the two main advantages of integrating information coming from several point of views are, first that the number of pixels that result occluded decreases with respect to simpler image-pair processing methods, and second that many estimations of the same measures are available and allow us to merge them into a more robust estimation.

The following phase of our work is presented in Chapter 4 and it is focused on the study of uncalibrated view-synthesis. We developed a fully automatic image rendering method that exploits the presence of several input images and addresses several subproblems of its class of methods, both from a geometric and a more practical point of view.

Chapter 5 presents the results that we obtained: it begins with presenting specific experiments designed to evaluate the two methods separately and

in the last section exhibits the benefits of concatenating motion-stereo and view-synthesis.

In Chapter 6 we recap the main contributions of our work, presenting the final considerations on the obtained results and discussing possible future works. This last chapter concludes the main part of the thesis. The Appendix and the References follow, ending the thesis.

## 1.6 Contributions

First, we dealt with the computation of a depth-proxy, developing a data-fusion framework that can take disparity maps as input and process them integrating the information coming from many images. We present a framework that takes into account both inter-frame (temporal) information and intra-frame (spatial) information. Moreover, it is very versatile as it can deal with both calibrated and uncalibrated quantities. During the description of the method we also present a useful analysis of suitable depth-proxies, as well as an extensive comparison of several confidence measures in the context of data-fusion. A preliminary *temporal only integration* version of this work has been presented in [1] and the complete method has presented in [2].

Regarding the view-synthesis part, we presented a fully automatic method that addresses most of the critical problems arising in uncalibrated scenarios. The method has a solid geometrical foundation and is able to take into account several images for a higher quality rendering of the virtual view. The method has been presented in [3], along with its experimental evaluation.

We also test the method in a case study that has been conducted during a collaboration with professor Anders Hast of the University of Uppsala, Sweden. In this work we apply our view-synthesis method to historical aerial photographs taken during World War II. The section does not only presents the results but also gives some insights on the specific application. This work has been presented in [4].

We then concatenate the two main parts of this thesis, i.e. the motion-stereo framework and the view-synthesis, and corroborate the idea that using a depth–proxy map refined with the motion-stereo pipeline for view-synthesis purposes produces sensible benefits to the final result.

Overall, the work described in this thesis constitutes a step towards building a complete pipeline for automatic 3DS conversion of uncalibrated images. All the building blocks are designed to be able to exploit the abundance of images and use them to achieve better quality. With respect to Figure 1.2 the proposed methods represent improvements both on the depth-proxy computation and on the warping function.

### Other works

During the past three years we had the opportunity to work on different topics other than the ones that are reported in this thesis. Since there was no prominent connection it seemed inappropriate to try to fit such topics into the chapters of this document, hence, we refer the reader, if interested, to the two papers that have been published as a result of these works.

The first paper is titled *Robust Global Motion Estimation with Matrix Completion* [5]. This work has been carried on during the Master thesis of Federica Arrigoni at the University of Milan. Federica is currently a Ph.D. Student at the University of Udine.

The second article is titled *Procrustean point-line registration and the NPnP problem* [6] and it is the result of a collaboration between our group and Professor Fabio Crosilla of the University of Udine.

# Chapter 2

# Geometric background

## Contents



In this chapter we present the basic theory behind our problem and behind the proposed methods. During the description of these topics, the main idea is to maintain a duality between the calibrated framework, where we

can count on euclidean geometry and the uncalibrated framework, where everything is known up to a projective transformation.

The first part of the chapter includes classical topics such as the pinhole camera model (Section 2.1), followed by epipolar and planar induced geometries (Sections 2.2 and 2.3) and image rectification Section 2.4). We then proceed, in Section 2.6, to describe the procedure for projective reconstruction and how to realign other projective frames to a reference one. In Section 2.7 we will describe the geometric quantities that are denoted as *depth–proxies* and that are computed in our algorithm. Maintaining the same line of thought we will analyze the relationship between depth and its uncalibrated version: planar-parallax, which is also known as projective depth. At last, Section 2.8, shows how to specify a motion trajectory in an uncalibrated framework, pointing out the correspondences to the equivalent calibrated operation.

For more details on the topics introduced in this chapter, we refer the interest reader to [31] wichh contains more detailed and in–depth explanations. Some of the figures and of the content are derived from the ones in [22]. More punctual references can be found as each notion is introduced.

### Notation

From now on, we will refer to a pair of images using subscript $_r$ for quantities that are related to the *reference image* $I_r$, and the subscript $_i$ for the *auxiliary image* $I_i$, i.e. the second element of the pair. This will be useful when we will start considering image sequences instead of pairs: we will work on a reference image, which will be kept the same while the auxiliary one will be one of the other elements of the sequence.

## 2.1 Model of the camera

We start defining some basilar elements that will be useful in the following sections. The camera model that we adopt is a 3–*by*–4 matrix $P$, called *perspective projection matrix* (PPM for simplicity). The intrinsic parameters, which map the 3D space into the image plane of the camera are modeled into matrix $K$. The extrinsic parameters, which define the position of the camera in world reference system, are defined by two matrices: $R$ and $\mathbf{t}$, which are respectively a rotation and a translation, that together form a rigid transformation that maps the camera reference system in the world reference system. Matrix $P$ is defined as follows

$$P = K[R|\mathbf{t}] \tag{2.1}$$

The camera is centered in the optical center $\mathbf{C}$. A generic 3D point $\mathbf{M}$ is projected on an *image point* $\mathbf{m}$ in the image plane. This is expressed by the equation

$$\mathbf{m} \simeq P\mathbf{M} \tag{2.2}$$

and represented in Figure 2.1. In the following it will be useful to write the

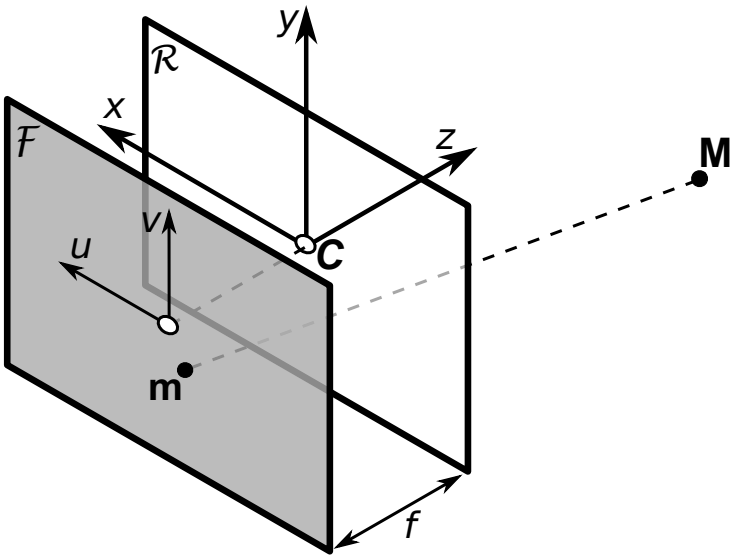


Figure 2.1: geometric model of our camera.

camera matrix as

$$P = [Q|\mathbf{q}]. \tag{2.3}$$

Starting from Equation (2.2) we can derive that the coordinates of the projected point $\mathbf{m}$ are:

$$\begin{cases} u = \frac{\mathbf{p}_1\mathbf{M}}{\mathbf{p}_3\mathbf{M}} \\ v = \frac{\mathbf{p}_2\mathbf{M}}{\mathbf{p}_3\mathbf{M}} \end{cases} \tag{2.4}$$

Where $p_1$, $p_2$ and $p_3$ are the rows of the matrix $P$. From Figure 2.1 we can see that the optical center $\mathbf{C}$ is the intersection of three planes: $\mathbf{p_1M} = 0$ $\mathbf{p_2M} = 0$ and $\mathbf{p_3M} = 0$. Thus $\mathbf{C}$ is the solution of the system

$$\begin{cases} \mathbf{p}_1\mathbf{C} = 0 \\ \mathbf{p}_2\mathbf{C} = 0 \\ \mathbf{p}_3\mathbf{C} = 0 \end{cases} \tag{2.5}$$

which is the same of saying $P\mathbf{C} = \mathbf{0}$. Remembering Equation (2.3), and that

$$\mathbf{C} = \begin{bmatrix} \tilde{\mathbf{C}} \\ 1 \end{bmatrix} \tag{2.6}$$

we can write

$$Q\tilde{\mathbf{C}} + \mathbf{q} = 0 \tag{2.7}$$

and so

$$\tilde{\mathbf{C}} = -Q^{-1}\mathbf{q} \tag{2.8}$$

Now if we want to define the optical axis of $\mathbf{m}$ we have to specify two points lying on it. The first one is $\mathbf{C}$ and the other one is an ideal point:

$$\begin{bmatrix} -Q^{-1}\mathbf{m} \\ 0 \end{bmatrix} \tag{2.9}$$

It is correct to assume that the optical axis contains the point defined in Equation (2.9) and that by substituting it to $\mathbf{M}$ in Equation (2.2) we can express the parametric equation for the optical axis

$$\mathbf{M} = \mathbf{C} + \lambda \begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix}, \quad \lambda \in \mathbb{R}. \tag{2.10}$$

## 2.2 Epipolar geometry

Epipolar geometry describes the relation between two images of the same scene taken from two different cameras or from the same camera at different times. In a stereo acquisition system, a 3D point $\mathbf{M}$ is projected on an image point $\mathbf{m_r}$ in the first image $P_r$, centered in $\mathbf{C_r}$, and on an image point $\mathbf{m_i}$ in the second image through a camera $P_i$ centered in $\mathbf{C_i}$, as represented in Figure 2.2.

The equations that represent this situation are

$$\begin{cases} \mathbf{m}_r \simeq P_r\mathbf{M} \\ \mathbf{m}_i \simeq P_i\mathbf{M} \end{cases} \tag{2.11}$$

Points $\mathbf{m_r}$ and $\mathbf{m_i}$ are called corresponding points and $\mathbf{m_i}$ is constrained to lie on a line, called the *epipolar line* of $\mathbf{m_r}$. This can be easily seen in Figure 2.2, $\mathbf{m_i}$ lies on the interception of two planes: the image plane and the *epipolar plane*, which is the plane determined by $\mathbf{M}$, $\mathbf{C_r}$ and $\mathbf{C_i}$ (gray colored in Figure 2.2). The explanation of this constraint is that $\mathbf{m_i}$ could be the projection of any point lying on the optical axis of $\mathbf{m_r}$. We can also observe that all the epipolar lines intersect at a point which is called the *epipole*. The epipole is defined as the projection of $\mathbf{C_r}$, the center of the first camera, through the second camera $P_i$

$$\mathbf{e_i} = P_i\mathbf{C_r} \tag{2.12}$$

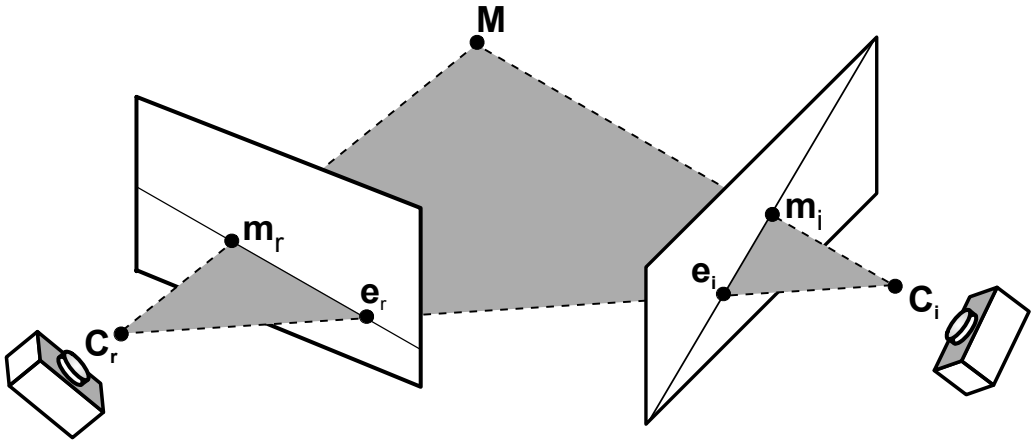


Figure 2.2: projection of a point $\mathbf{M}$ by two cameras centered in $\mathbf{C}_r$ and $\mathbf{C}_i$ and main elements of epipolar geometry.

All the epipolar planes contain the line defined by $\mathbf{C}_r$ and $\mathbf{C}_i$, i.e. the *baseline*.

Like we said, the epipolar line corresponding to $\mathbf{m}_r$ is the projection of Equation (2.10) through $P_i$ and thus has the equation

$$\mathbf{m_i} \simeq \lambda Q_i Q_r^{-1} \mathbf{m_r} + \mathbf{e}_i \tag{2.13}$$

We can elaborate (2.13) to show that a bilinear relation exists between $\mathbf{m}_r$ and $\mathbf{m}_i$. If we multiply both sides of equation (2.13) by $[\mathbf{e}_i]_\times$, where $[\mathbf{e}_i]_\times$ is the skew-symmetric matrix associated with the cross-product and defined in Appendix C.4, we can write

$$[\mathbf{e}_i]_\times \mathbf{m}_i \simeq \lambda [\mathbf{e}_i]_\times Q_i Q_r^{-1} \mathbf{m} \tag{2.14}$$

The left element is a vector orthogonal to $\mathbf{m}_i$, so if we multiply both sides by $\mathbf{m}_i^\top$ we obtain

$$0 = \mathbf{m}_i^\top [\mathbf{e}_i]_\times Q_i Q_r^{-1} \mathbf{m}_r \tag{2.15}$$

Let us rewrite it this way

$$\mathbf{m}_i^\top [\mathbf{e}_i]_\times Q_i Q_r^{-1} \mathbf{m}_r = 0 \tag{2.16}$$

This equation is also called *Longuet–Higgins equation*. We observe that matrix $F = [\mathbf{e}_i]_\times Q_i Q_r^{-1}$, which is called the *fundamental matrix*, contains all the information about the epipolar geometry and allows us to trace the epipolar line of any point $\mathbf{m}$ as $F\mathbf{m}$. Now, we can rewrite Equation (2.16) as:

$$\mathbf{m}_i^\top F \mathbf{m}_r = 0 \tag{2.17}$$

Another important property is given by the observation that the epipole is the kernel of $F$, i.e.

$$F\mathbf{e}_r = \mathbf{0} \quad \text{and} \quad \mathbf{e}_i^\top F = \mathbf{0} \tag{2.18}$$

so it can be easily extracted from it via singular values decomposition (the solution is the eigenvector associated with the minimum eigenvalue of matrix $F^\top F$).

### 2.2.1 Computing the fundamental matrix

In this section we will describe how to compute the fundamental matrix $F$ from a set of correspondences $(m_r^j, m_i^j)$ with $j = 1 \dots n$. When the context is clear we will omit the pixel's index $^j$ for a cleaner notation.

Using Equation (C.2) (In Appendix C.2) of the Kronecker product, we can elaborate Equation 2.17

$$\mathbf{m}_i^\top F \mathbf{m}_r = 0 \iff \text{vec}(\mathbf{m}_i^\top F \mathbf{m_r}) = 0 \iff (\mathbf{m}_i^\top \otimes \mathbf{m_r}^\top)\text{vec}(F) = 0 \tag{2.19}$$

This means that every pair of corresponding points generates a homogeneous equation linear in the nine elements of $F$ (vectorized). From $n$ points we obtain a linear system with $n$ equations:

$$\underbrace{\begin{bmatrix} \mathbf{m}_r^{1\top} \otimes \mathbf{m}_i^{1\top} \\ \mathbf{m}_r^{2\top} \otimes \mathbf{m}_i^{2\top} \\ \vdots \\ \mathbf{m}_r^{n\top} \otimes \mathbf{m}_i^{n\top} \end{bmatrix}}_{U_n} \text{vec}(F) = 0. \tag{2.20}$$

The solution we are looking for is the nucleus of $U_n$. With $n = 8$ the nucleus of the matrix has dimension one, thus the solution is determined up to a scale. Therefore this method is called the 8-point algorithm (note that the eight points must be in general position, see [18] for degenerate configurations) but it is indeed a variant of the DLT method that will be described in Section 2.3.1.

In practice there usually are more than eight point correspondences and we can obtain the elements of $F$ by solving a linear least squares problem. The solution is the eigenvector associated with the minimum eigenvalue of $U_n^\top U_n$ that can be computed with the singular value decomposition of $U_n$.

Please note that the matrix $F$ that is found solving this system of equations will, in general, not be compliant to the requisite of being singular. This can be forced at posterior by substituting $F$ con $\hat{F}$, the closest matrix in Frobenius norm that is singular.

Let $F$ be a matrix $3\times3$ and $F = UDV^\top$ be its SVD with $D = \text{diag}(r, s, t)$ and $r \geq s \geq t$. It can be demonstrated that $\hat{E} = U\hat{D}V^\top$ where $\hat{D} = \text{diag}(r, s, 0)$.

Although the linear algorithm that we just described needs at least eight points to compute $F$, since the matrix only depends on seven parameters, it is possible to compute it from seven correspondences with a non-linear procedure ([31]).

At last, observe that the least squares minimization solution is obtained by a minimization of an algebraic error. To refine results we can minimize, e.g. using the Levenberg-Marquardt method, a geometric error

$$\min_F \sum_j d(F\mathbf{m}_r^j, \mathbf{m}_i^j)^2 + d(F^\top \mathbf{m}_i^j, \mathbf{m}_r^j)^2 \tag{2.21}$$

Where $d(\cdot)$ is the point–line distance in the Cartesian plane. Note that equation (2.21) is non–linear and that this minimization is slow and does not assure convergence to the absolute minimum so it is convenient to use it as a refinement of the least square solution allowing us to trust to start the minimization near the minimum error. Moreover, $F$ must be properly parametrized to reflect its 7 degrees of freedom.

## 2.3 Planar induced homography

In a less general situation, corresponding points are linked not only by the fundamental matrix, but also by a *projectivity* or *homography*. This happens when observed points are lying on the same plane in the 3D space. Primarily we can observe that the transformation between the plane $\Pi$ and its projection on the image plan is an homography. In a situation like the one represented in Figure 2.3, where we have two different images, composing the two transformations, we obtain an homography from the left image to the right image. We can say that the plane $\Pi$ induces an homography $H^\Pi$ between the two images that gives us the relation

$$\mathbf{m}_i \simeq H^\Pi \mathbf{m}_r \quad \text{if } \mathbf{M} \in \Pi \tag{2.22}$$

To see in which cases images of the same scene are linked by an homography we have to restart from the epipolar geometry. The two cameras can be expressed as

$$P_r = K_r[I|\mathbf{0}] = [K_r|\mathbf{0}] \quad \text{and} \quad P_i = K_i[R|\mathbf{t}] \tag{2.23}$$

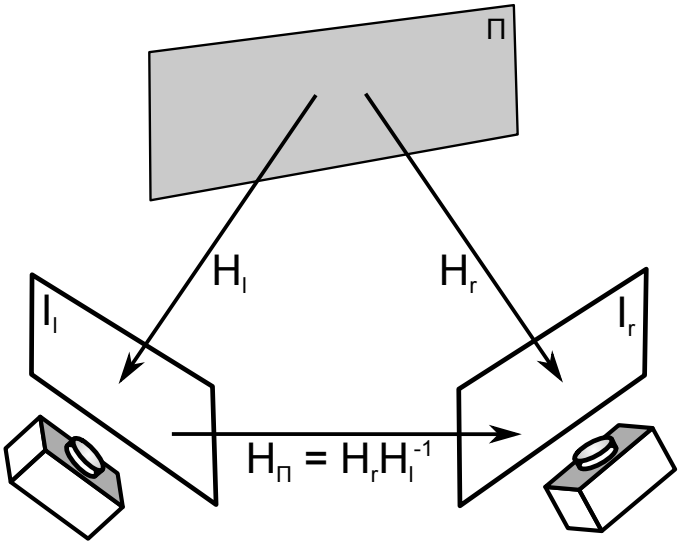


Figure 2.3: plane Π induces an homography between the two image planes.

Where $K$ contains the intrinsic parameters of the camera and $R$ and $t$ represent the extrinsic parameters. Substituting the two cameras into the Equation (2.13) we obtain

$$\mathbf{m}_i \simeq \lambda K_i R K_r^{-1} \mathbf{m}_r + K_i \mathbf{t}. \tag{2.24}$$

Adding explicit depth values, Equation (2.24) becomes

$$\zeta_i \mathbf{m}_i = \zeta K_i R K_r^{-1} \mathbf{m}_r + K_i \mathbf{t} \tag{2.25}$$

We can observe two situations where the two images are connected by an homography.

**Rotational motion of the camera:** if the movement of the camera is purely rotational, then $\mathbf{t} = \mathbf{0}$ and

$$\frac{\zeta_i}{\zeta_r} \mathbf{m_i} = K_i R K_r^{-1} \mathbf{m}_r \tag{2.26}$$

Where $K_i R K_r^{-1} = H^{\infty}$ is an homography that does not depend on the 3D structure.

**Planar scene:** For points $\mathbf{M}$ lying on a plane $\Pi$ with equation $\mathbf{n}^{\top} \tilde{\mathbf{M}} = d$, where $n$ is the normal vector to the plane and $d$ is the distance of $\Pi$ from the origin, then we can derive

$$\frac{\zeta_i}{\zeta_r} \mathbf{m}_i = K_i \left( R + \frac{\mathbf{t}\mathbf{n}^{\top}}{d} \right) K_r^{-1} \mathbf{m}_r \tag{2.27}$$

This means that the two images are linked by an homography $H^{\Pi} = K'\left(R + \frac{\mathbf{t}\mathbf{n}^{\top}}{d}\right)K^{-1}$.

It is also worth observing that if $d \to \infty$ we have that $H^{\infty} = H^{\Pi}$. This means that $H^{\infty}$ links all the points lying on the infinity plane (for example vanishing points) or every point in the image if the camera moves without translating.

### 2.3.1 Computing the homography

The classical way to compute a certain homography is by knowing at least $n$ points (by the end of this subsection we will determine how much is $n$) lying on the same plane $\Pi$ (which can also be the infinity plane), with the *Direct Linear Transform* method, which can be found in [31]. Given $n$ correspondences $(\mathbf{m}_r, \mathbf{m}_i)$, where $\mathbf{m}_r$ and $\mathbf{m}_i$ are projections of the 3D a point $\mathbf{M}$ lying on a certain plane $\Pi$, we want to determine $H^{\Pi}$ such that

$$\mathbf{m}_i \simeq H^{\Pi}\mathbf{m}_r \tag{2.28}$$

Which is equivalent to

$$\mathbf{m}_i \times H^{\Pi}\mathbf{m}_r = \mathbf{0} \tag{2.29}$$

As we know from Appendix C.4, we can substitute the cross-product with its associated matrix $[\mathbf{m}_i]_{\times}$ and write

$$[\mathbf{m}_i]_{\times}H^{\Pi}\mathbf{m}_r = \mathbf{0} \tag{2.30}$$

And at last using the vectorization and exploiting the Kronecker product and its property described in Equation (C.2) (in Appendix C.2), we come to

$$\operatorname{vec}([\mathbf{m}_i]_{\times}H^{\Pi}\mathbf{m}_r) = \mathbf{0} \Longleftrightarrow (\mathbf{m}_r^{\top} \otimes [\mathbf{m}_i]_{\times})\operatorname{vec}(H^{\Pi}) = \mathbf{0} \tag{2.31}$$

Where $\operatorname{vec}(H^{\Pi})$ contains 9 unknown values and $(\mathbf{m}_r^{\top} \otimes [\mathbf{m}_i]_{\times})$ is a $3 \times 9$ and rank deficient (rank 2) matrix. We have two equations for every correspondence $(\mathbf{m}_r, \mathbf{m}_i)$ of points that are projections of 3D points that lie on $\Pi$. Thus for $n$ points we have $2n$ equations. By stacking them we obtain a $2 \times 9$ matrix, let's call it $A$, for which

$$A\operatorname{vec}(H^{\Pi}) = \mathbf{0} \tag{2.32}$$

Using $n = 4$, $A$ is a 8–*by*–9, with rank 8 and its unidimensional kernel (as we know from the *rank–nullity* theorem) is the solutions vector, containing the nine elements of $H^{\Pi}$. Note that the $n$ points must be coplanar on the plane

$\Pi$ but they must be in general position, i.e. there cannot be three collinear points. For $n > 4$, the least square minimization solution can be found via the singular values decomposition (the solution is the eigenvector associated with the minimum eigenvalue of matrix $A^\top A$).

The least square minimization solution described above, is obtained by minimizing an algebraic error and to refine results we can minimize a geometric error, e.g. using the distance between points in the Cartesian plane $d(\cdot)$

$$\min_H \sum_i d(H\mathbf{m}_r, \mathbf{m}_i)^2 + d(H^{-1}\mathbf{m}_i, \mathbf{m}_r)^2 \tag{2.33}$$

Note that equation (2.33) is non-linear and that this minimization is slow and does not assure convergence to the absolute minimum. It can be carried out with a Levenberg-Marquardt procedure, but it is convenient to use it as a refinement of the least square solution allowing to start the minimization near actual minimum error. Also, unlike $F$, $H$ does not need any parametrization, having 8 degrees of freedom.

### Homography from the PPMs

There is another way to compute the infinity plane homography $H^\infty$ when the PPMs are known. Consider two PPMs and their factorization $P_r = [Q_r|\mathbf{q}_r]$ and $P_i = [Q_i|\mathbf{q}_i]$ (as in Equation (2.3)) and consider a generic 3D point lying on the infinity plane, thus with the last element of its homogeneous coordinates equal to zero $M = (X, Y, Z, 0)^\top$. We can rewrite Equation (2.11) as

$$\begin{cases} m_r \simeq [Q_r|\mathbf{q}_r](X, Y, Z, 0)^\top \\ m_i \simeq [Q_i|\mathbf{q_i}](X, Y, Z, 0)^\top \end{cases} \tag{2.34}$$

And, from Equation (2.22), since $M$ lies in the infinity plane, we know that

$$\mathbf{m}_i \simeq H^\infty \mathbf{m}_r \tag{2.35}$$

It is easy to obtain the following equation

$$H^\infty \simeq Q_i {Q_r}^{-1} \tag{2.36}$$

which relies on the computation of the PPMs instead of a set of corresponding image points. This alternative method can be useful if the correspondences are not trustworthy.

### 2.3.2 Epipole mapping through homographies

A very interesting property, is that for every $\Pi$ it is true that

$$\mathbf{e_i} \simeq H^{\Pi}\mathbf{e_r} \tag{2.37}$$

We now will give an intuitive demonstration of this property.

Given the two PPMs in Equation (2.23) and the definition of the epipole in Equation (2.12), we observe that

$$\mathbf{e_r} = K_r R^\top \mathbf{t} \quad \text{and} \quad \mathbf{e_i} = K_i \mathbf{t} \tag{2.38}$$

Using the most general definition of the plane induced homography from Equation (2.27) we can write:

$$H^{\Pi}\mathbf{e_r} = K_i \left( R + \frac{\mathbf{t}\mathbf{n}^\top}{d} \right) K_r^{-1} K_r R^\top \mathbf{t} = K_i \mathbf{t}(1 + \frac{\mathbf{n}^\top}{d} R^\top \mathbf{t}) \tag{2.39}$$

Where the last term is a scalar value, thus for any $\Pi$ we can write:

$$H^{\Pi}\mathbf{e_r} \simeq \mathbf{e_i} \tag{2.40}$$

Observe that if we choose three random 3D points they are coplanar by definition and thus their projections on a pair of images always satisfy Equation (2.22). This tells us that three random points and the epipole as fourth form a suitable input point for the DLT algorithm. The output would be the homography induced by the plane containing the three points.

## 2.4 Stereo rectification

Given a pair of stereo images $(I_r, I_i)$, a rectification procedure determines two transformations (homographies, actually) $T_r$ and $T_i$, one for each image plane. When $T_r$ and $T_i$ are applied, pairs of conjugate epipolar lines become collinear and parallel to one of the image axes and the epipoles become points at the infinity. The rectified images can be thought of as acquired by a new pair of cameras, obtained by rotating the original ones about their optical centers until their focal planes become coplanar and contain the baseline. The result sought by the rectification is depicted in Figure 2.4. This configuration is also called *normal* case (for stereo).

### 2.4.1 Calibrated rectification

The method that is here described is based on [25], please refer to the paper for a full description.

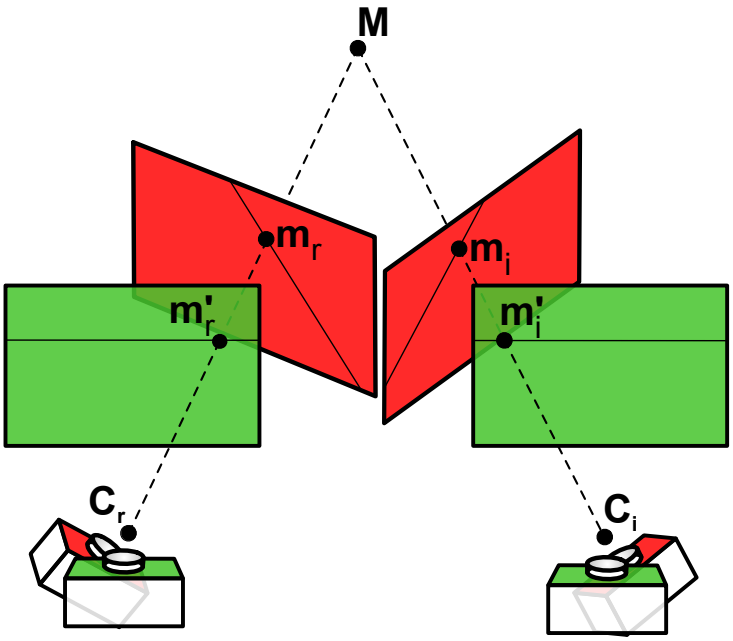


Figure 2.4: pictorial representation of the rectification. The two original image planes (red, in this example) are warped to become coplanar. The epipolar lines become collinear and the epipoles are located at infinity.

### Rectifying thee PPMs

Let $P_{o,r}$ and $P_{o,i}$ be the two PPMs for the image pair $(I_r, I_i)$, and let $P_{n,r}$ and $P_{n,i}$ b e the two matrices obtained after the rectification procedure. Their factorization is

$$P_{n,r} = K[R \mid -R\,\tilde{\mathbf{C}}_r], \qquad P_{n,i} = K[R \mid -R\,\tilde{\mathbf{C}}_i]. \tag{2.41}$$

Indeed, the intrinsic parameters matrix $K$ is the same for both PPMs (and can be set arbitrarily). The optical centers $\tilde{\mathbf{C}}_\mathbf{r}$ and $\tilde{\mathbf{C}}_\mathbf{i}$ are the same one of the original cameras and matrix $R$, that determines the camera orientation is the same for both PPMs. If we write the rotation as follows:

$$R = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix} \tag{2.42}$$

we obtain that $\mathbf{r}_1^\top, \mathbf{r}_2^\top, \mathbf{r}_3^\top$ are, respectively, the axis $X, Y$ and $Z$ of the cameras reference frame, expressed in world coordinates. We are able to determine $R$ by setting:

1. the new $X$ axis parallel to the baseline: $\mathbf{r}_1 = (\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_r)/\|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_r\|$;

2. the new $Y$ axis orthogonal to the $X$ and to an arbitrary versor $\mathbf{k}$: $\mathbf{r}_2 = \mathbf{k} \times \mathbf{r}_1$. The unit vector $\mathbf{k}$ fixes the position of the new $Y$ axis in the plane orthogonal to $X$ (vertical direction). We choose it equal to the old camera's $Z$ versor, forcing the new $Y$ axis to be orthogonal to both the new $X$ and the old $Z$ axis.

3. the new $Z$ axis orthogonal to $X$ and $Y$: $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$.

**Rectifying the image planes**

To rectify the plane of a camera, e.g. $P_{o,r}$, one needs to compute the transformation that brings the image plane of $P_{o,r} = [Q_{o,r}|\mathbf{q}_{o,r}]$ to the one of $P_{n,r} = [Q_{n,r}|\mathbf{q}_{n,r}]$. The desired transformation is the homography (non singular linear transformation) defined by the $3 \times 3$ matrix $H = Q_{n,r}Q_{o,r}^{-1}$. For each 3D point $\mathbf{M}$, we can write

$$\begin{aligned} \mathbf{m}_{o,r} &\simeq P_{o,r}\mathbf{M} \\ \mathbf{m}_{n,r} &\simeq P_{n,r}\mathbf{M}. \end{aligned} \tag{2.43}$$

According to Equation (2.10), the optical rays equations are the following (since the rectification does not move the camera centers):

$$\begin{aligned} \tilde{\mathbf{M}} &= \tilde{\mathbf{C}} + \lambda_{o,r}Q_{o,r}^{-1}\mathbf{m}_{o,r} \qquad \lambda_{o,r} \in \mathbb{R} \\ \tilde{\mathbf{M}} &= \tilde{\mathbf{C}} + \lambda_{n}Q_{n,r}^{-1}\mathbf{m}_{n,r} \qquad \lambda_{n,r} \in \mathbb{R} \end{aligned} \tag{2.44}$$

Thus:

$$\mathbf{m}_{n,r} \simeq Q_{n,r}Q_{o,r}^{-1}\mathbf{m}_{o,r}. \tag{2.45}$$

The transformation $H$ is then applied to the original image to produce the rectified image, as shown in Figure 2.4.

### 2.4.2 Uncalibrated rectification

If calibration data is not available and our only knowledge are point correspondences, a suitable the method is the one proposed in [24] and that we will herein summarize. Whereas in the case of calibrated cameras the epipolar rectification is unique up to trivial transformations, in the case of uncalibrated cameras there are more degrees of freedom in choosing the rectifying transformation. Calibrated rectification is done with respect to the plane at infinity, while uncalibrated rectification can be seen as referred to a plane that approximates the plane at infinity. The method seeks the collineations that make the correspondent points satisfy the epipolar geometry of a rectified image pair, refer to the original work [24] for all the details of the method.

First, let us observe that the fundamental matrix of a rectified pair has a very specific form, namely it is the skew-symmetric matrix $[\mathbf{u_1}]_\times$ associated with the cross-product by the vector $u_1 = (1, 0, 0)$.

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.46}$$

Let $T_r$ and $T_i$ be the unknown rectifying homographies. The transformed corresponding points must satisfy the epipolar geometry of a rectified pair, hence

$$(T_i \mathbf{m}_i^j)^\top [\mathbf{u_1}]_\times (T_r \mathbf{m}_r^j) = 0. \tag{2.47}$$

As this equation must hold for any correspondence, we obtain a system of non–linear equations in the unknown $T_r$ and $T_i$. The left-hand side of Equation 2.47 is an algebraic error, i.e. it has no geometrical meaning, so as in [24], we use instead the Sampson distance (see Appendix C.3) which is a first order approximation of the geometric reprojection error. The matrix $T_i^\top [\mathbf{u_1}]_\times T_r$ can be considered as the fundamental matrix $F$ between the original images, therefore, in our case, the Sampson error for the j-th correspondence is defined as

$$S_i = \frac{(\mathbf{m}_i^{j\top} F \mathbf{m}_r^j)^2}{\left\|[\mathbf{u}]_\times F m_r^j\right\|^2 + \left\|m_i^{j\top} F [\mathbf{u}]_\times\right\|^2}. \tag{2.48}$$

A least-squares solution to the system of equations is sought. The way in which $T_r$ and $T_i$ are parameterized is crucial: the rectifying homographies are forced to have the same structure as in the calibrated case, i.e. to be homographies induced by the plane at infinity, namely

$$T_r = K_{n,r} R_r K_{o,r}^{-1} \quad \text{and} \quad T_i = K_{n,i} R_i K_{o,i}^{-1} \tag{2.49}$$

The old intrinsic parameters $(K_{o,r}, K_{o,i})$ and the rotation matrices $(R_r, R_i)$ are unknown, whereas the new intrinsic parameters $(K_{n,r}, K_{n,i})$ can be set arbitrarily, provided that vertical focal length and vertical coordinate of the principal point are the same. Each homography depends in principle on five (intrinsic parameters) plus three (rotation angles) unknown parameters. The rotation of one camera along its X-axis, however, can be eliminated, as this is tantamount to rotating a rectified pair around the baseline. The number of parameters is further reduced by making an approximation on the old intrinsic parameters, no skew, principal point in the center of the image, aspect ratio equal to one. The only remaining unknowns in $(K_{o,r}, K_{o,i})$ are

the focal lengths. Assuming that they are identical and equal to $f$, we get

$$K_{o,r} = K_{o,i} = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.50}$$

where $w$ and $h$ are width and height (measured in pixels) of the image. The minimization can be carried out using Levenberg-Marquardt, starting with all the unknown variables set to zero. At last, the rectifying homographies are computed with Equation 2.49.

## 2.5 Observations and properties

As a by-product of the rectification procedure, it can be shown that we obtain the homography induced by the plane at infinity (or its approximation in the uncalibrated case) between the two original cameras, which is given by

$$H_{ri}^{\infty} = T_i^{-1} T_r \tag{2.51}$$

This also tells us that the infinity plane homography of a rectified pair is the identity matrix

$$H_{ri}^{\infty} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.52}$$

Moreover, we can observe that since the fundamental matrix of a rectified pair is the skew-symmetric matrix in Equation (2.46), the epipole is $u_1 = (1, 0, 0)$ vector itself (Equation (2.18)).

## 2.6 Projective reconstruction

When camera parameters are unavailable we are still able to estimate information about the scene, but up to an unknown projectivity. This procedure is known as projective reconstruction.

Consider a set of 3D points seen by $m$ cameras $\{P_i\}_{i=1\ldots m}$. Let $\mathbf{m}_i^j$ be the (homogeneous) coordinates of the projection of the $j$-th point onto the $i$-th camera.

The projective reconstruction problem can be seen as the one of finding the set of cameras' PPMs $\{P_i\}$ and the scene structure $\{\mathbf{M}^j\}$, given the set of pixel coordinates $\{\mathbf{m}_i^j\}$, such that

$$\mathbf{m}_i^j \simeq P_i \mathbf{M}^j. \tag{2.53}$$

Without further constraints this reconstruction is defined up to an arbitrary projectivity. As a matter of fact if $\{P_i\}$ e $\{\mathbf{M}^j\}$ satisfy Equation (2.53), then $\{P_i T\}$ and $\{T^{-1}\mathbf{M}^j\}$ satisfy (2.53) for any non-singular $4 \times 4$ matrix $T$. Matrix $T$ specifies a linear transformation in the 3D projective framework, i.e. a projectivity.

Several methods for projective reconstruction exist in the literature. We use the classical method proposed in [68] which is briefly presented in Section 2.6.1. This method is based on the factorization procedure presented in [69]. It is an iterative method but one can easily find out empirically that it is fast and it does not require an informed initialization, even if convergence is not guaranteed as in other methods (e.g. [50]). Moreover the method is not minimal since it solves an overdetermined problem providing a solution in a least square sense. A minimal method is e.g. the one described in [31] which provides the projective reconstruction from three views using six input points.

### 2.6.1 The iterative factorization algorithm

Consider $m$ cameras looking at $n$ points in 3D space, $\mathbf{M}^1 \dots \mathbf{M}^n$. Let us rewrite Equation (2.2) with an explicit scale factor

$$\zeta_i^j \mathbf{m}_i^j = P_i \mathbf{M}^j \quad i = 1 \dots m, \quad j = 1 \dots n \tag{2.54}$$

which in matrix form becomes

$$\underbrace{\begin{bmatrix} \zeta_1^1\mathbf{m}_1^1 & \zeta_2^1\mathbf{m}_2^1 & \dots & \zeta_m^1\mathbf{m}_m^1 \\ \zeta_1^2\mathbf{m}_1^2 & \zeta_2^2\mathbf{m}_2^2 & \dots & \zeta_m^2\mathbf{m}_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_1^n\mathbf{m}_1^n & \zeta_2^n\mathbf{m}_2^n & \dots & \zeta_m^n\mathbf{m}_m^n \end{bmatrix}}_{\text{misure } W} = \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \mathbf{M}^1, & \mathbf{M}^2, & \dots & \mathbf{M}^n \end{bmatrix}}_{\text{struttura } M}. \tag{2.55}$$

In this equation, everything but the $\mathbf{m}_i^j$ is unknown, even the values of $\zeta_i^j$. What we learn from it is that $W$ can be factorized in the product of a $3m \times 4$ matrix $P$ and a $4 \times n$ matrix $M$. Thus $W$ has rank four.

If, for a moment, we assume $\zeta_i^j$ as known, matrix $W$ becomes known and we can compute the singular values decomposition

$$W = UDV^\top. \tag{2.56}$$

Theoretically, if point correspondences are not affected by noise, rank of $W$ is four, thus $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \dots 0)$. This means that only the first four columns of $U$ ($V$) contribute to the matrix product. Let $U_{3m\times 4}$

($V_{n\times 4}$) be the matrix formed by the first 4 columns of $U$ ($V$). We can rewrite compact SVD of $W$:

$$W = U_{3m\times 4}\,\mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,\sigma_4)\,V_{n\times 4}^{\top}. \tag{2.57}$$

If we compare this with Equation (2.55) we are able to identify:

$$P = U_{3m\times 4}\,\mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,\sigma_4) \quad \text{e} \quad M = V_{n\times 4}^{\top} \tag{2.58}$$

obtaining the desired reconstruction. Please note that the choice of including $\mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$ in $P$ is arbitrary. It could have been included in $M$ or be split across. This is coherent with the fact that the obtained reconstruction is up to a projectivity that assimilates $\mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$ as well.

In real cases, data (point correspondences) is noise affected thus the rank of $W$ is not four. If we force $D = \mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,\sigma_4,0,\ldots 0)$ we obtain the approximate solution that minimizes the error in Frobenius norm:

$$\|W - PM\|_F^2 = \sum_{i,j} \|\zeta_i^j \mathbf{m}_i^j - P_i \mathbf{M}^j\|^2. \tag{2.59}$$

This leaves us with the problem of estimating the unknown $\zeta_i^j$ values. As we have seen above if they were known we would be able to calculate $P$ and $M$. On the other hand i would be able to calculate $\zeta_i^j$ values by knowing $P$ and $M$, indeed, given a point $j$, the projection equation can be rewritten as:

$$\begin{bmatrix} \zeta_1^j \mathbf{m}_1^j \\ \zeta_2^j \mathbf{m}_2^j \\ \vdots \\ \zeta_m^j \mathbf{m}_m^j \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{m}_1^j & 0 & \ldots & 0 \\ 0 & \mathbf{m}_2^j & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{m}_m^j \end{bmatrix}}_{Q^j} \underbrace{\begin{bmatrix} \zeta_1^j \\ \zeta_2^j \\ \vdots \\ \zeta_m^j \end{bmatrix}}_{\boldsymbol{\zeta}^j} = PM^j. \tag{2.60}$$

An iterative solution known as block relaxation is suitable in these kind of problems and consists in solving alternatively the two problems: estimate $\zeta_i^j$ given $P$ and $M$, and in the subsequent step estimate $P$ and $M$ given $\zeta_i^j$, and iterate until convergence. The procedure is summarized in Algorithm 1. Step 1 is necessary to avoid convergence to the trivial solution $\zeta_i^j = 0$.

### 2.6.2 Alignment of projective frames

If we want to perform projective reconstruction on an image sequence, it is necessary to consider an input set of corresponding points across the entire

**Algorithm 1** PROJECTIVE RECONSTRUCTION

**Input:** Image point correspondences $W$, con $\zeta_i^j = 1$
**Output:** Reconstructed 3D points and cameras' PPMs $M$, $P$

1. Normalize $W$ as $\|W\|_F = 1$;
2. Obtain $P$ and $M$ from the SVD of $W$;
3. If $\|W - PM\|_F^2$ is small enough, stop;
4. extract $\boldsymbol{\zeta}^j$ from $Q^j\boldsymbol{\zeta}^j = PM^j$, $\forall\ j = 1 \dots n$;
5. Update $W$;
6. Repeat from (i).

sequence, otherwise each projective reconstruction performed on a portion of the original image sequence will lie in its own projective reference frame.

When correspondences are not available across the entire image sequence, a solution is to split it into subsequences and realign, afterwards, each projective frame to a certain reference frame. Each one of these reconstructed subsequences is connected to each other through an unknown projectivity. The realignment consists in computing and applying such projectivities in order to obtain a coherent projective reconstruction (i.e. within the same projective reference frame) across the entire image sequence.

In this section we present a procedure to realign projective reconstructions that lay in different projective reference frames.

Keeping Equation (2.36) in mind, let $P_i := [Q_i|q_i]$ be the PPMs, associated with the $i$-th image and let:

$$H_{ri}^{\Pi} := Q_i Q_r^{-1} \quad i = 2, \cdots, N \tag{2.61}$$

be the infinity plane homography between views $I_r$ and $I_i$. Observe that in a Euclidean frame $H_{ri}^{\Pi}$ is the homography induced by the *true* infinity plane, whereas in a projective frame, the infinity plane corresponds to a generic plane $\Pi$ in the Euclidean frame.

We run the projective reconstruction procedure that, given a set of sparse matches, yields an estimation of PPMs and 3D points.

The sparse visual features across the video sequence can be carried out using standard procedures. We follow the approach of [17]. The output of this stage is a set of tracks, i.e. keypoints matching in more than three images, and a set of fundamental matrices and homographies linking pairs

of views, each endowed with GRIC scores ([70]), which reveals which of the two models is more likely.

The projective reconstruction is obtained using the procedure described above and using the longest possible portions of the sequence. The length of the image subsequence is determined by the presence of a minimal number of complete keypoint tracks across it (we use at least 50 points). An alternative procedure is to use a minimal approach e.g. the projective reconstruction from three views using a 6–points procedure – described in [31] – inside a RANSAC/MSAC iteration. We prefer to use longer subsequences in order to minimize the number of alignments that can cause misalignment drifts.

Lying in a projective stratum, each subsequence of reconstructed PPMs is related to the correct (Euclidean) one by a collineation of the 3D space. Once a reference projective frame is fixed, e.g. the one associated with the first subsequence, following subsequences of PPMs with an overlap of (at least) two can be brought to the same frame by computing the proper collineation $T$ as explained in the following.

The first subsequence is brought to a *quasi–euclidean* stratus using the procedure described in Appendix B. The remaining subsequences of PPMs can then be brought, by computing the proper collineation following the scheme described in Figure 2.5 and explained in the following.

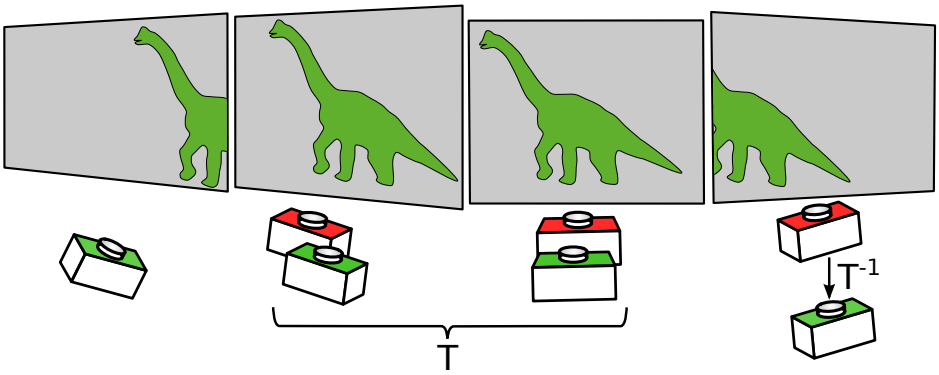


Figure 2.5: threading of projective cameras. Red/green represent different projective frames. Each subsequence (length 3 in this example) has an overlap of two. $T$ is computed from the overlapping elements and then $T^{-1}$ is used to bring the last camera to the reference projective stratum (green in this example).

Let $P_i$ and $P_i'$ be the same camera in two different projective frames, i.e., $P_i$ and $P_i'$ represents the same camera in two different subsequences. They

are related by an unknown collineation $T$:

$$P_i T \simeq P'_i. \tag{2.62}$$

Using the vec operator, defined in Appendix C.1, we obtain:

$$\mathrm{vec}(P_i T) \simeq \mathrm{vec}(P'_i). \tag{2.63}$$

In Appendix A, we show that the equality of two vectors $\mathbf{a}$ and $\mathbf{b}$ of $\mathbb{R}^n$ up to a scale can be written as $[\mathbf{a}]_\times \mathbf{b} = 0$ where $[\mathbf{a}]_\times$ is a suitable $n(n-1)/2 \times n$ matrix that generalizes the external product matrix of $\mathbb{R}^3$. Hence Equation (2.63) can be rewritten as:

$$[\mathrm{vec}(P'_i)]_\times \,\mathrm{vec}(P_i T) = 0. \tag{2.64}$$

Using the properties of the Kronecker product, Equation (2.64) is equivalent to the following linear system of equations in the unknown $\mathrm{vec}(T)$:

$$[\mathrm{vec}(P'_i)]_\times (I_{4\times 4} \otimes P_i)\,\mathrm{vec}(T) = 0. \tag{2.65}$$

Since the coefficient matrix has rank at most 11, at least two camera matrices are needed to stack-up the 15 equations required to compute the $4\times4$ matrix $T$ up to scale. This is the reason why our projective reconstruction processes sequences of cameras with an overlap of two.

Bringing all the PPMs into a common projective frame ensures that, when computing homographies using Equation 2.36, the space plane associated with homographies $H^{\Pi}_{ri}$ is the same. In this way we obtain an estimate for a fixed reference plane that does not depend on a particular choice of the corresponding points which generate the projective reconstruction. This has clear advantages over other strategies such as tracking 3D points belonging to a plane along the video sequence, or by considering the dominant collineation. A projective bundle adjustment is run eventually over cameras and sparse triangulated 3D points in order to improve the reconstruction precision.

## 2.7 Depth Proxies

If we do not make any hypothesis on whether the camera is calibrated or not, or if motion is constrained/known or not, the scenario we put ourselves in changes and influences which kind of information we are able to extract and process from the images. In particular we are interested in finding a suitable *depth–proxy*.

A depth-proxy is a quantity that is connected to the depth values of each pixel and is computable from knowing correspondences of a stereo-pair

of images. Moreover, since we are working with image sequences, we are interested in a depth–proxy that depends only on a certain reference image and not on the other element of the stereo pair being considered. In this way, if we process the elements of a sequence in a pairwise fashion, but keeping the first element of the pair fixed, then each iteration provides a new estimate commensurate with the others. Several depth–proxies can be computed depending on factors such as the constraints on the motion of the camera and/or the availability of the camera parameters. In this section we present three suitable candidates.

### 2.7.1 Planar Parallax

*Planar parallax*, also known as projective depth, represents the displacement in the apparent position of objects imaged from different points of view with respect to a reference plane [63]. In the case where camera calibration is unavailable and the camera undergoes a general motion, *planar parallax* can be profitably employed instead of depth and it can be computed from stereo correspondences.

Equation (2.25) relates two corresponding points using the homography of a generic plane $\Pi$ and can be rewritten as:

$$\zeta_i \mathbf{m}_i = \zeta_r H^{\Pi} \mathbf{m}_r + \mathbf{e}_i. \tag{2.66}$$

If we look at it as a formulation of the two-views geometry, Equation (2.66) leads to plane+parallax [38] formulation also known as relative affine structure [66] (please refer to the original works for a complete discussion and formulation of the planar parallax theory).

Given a plane $\Pi$, with equation $\mathbf{n}^\top \mathbf{M} = d$, two corresponding points $\mathbf{m}$ and $\mathbf{m}'$ are related by

$$\frac{\zeta_i}{\zeta_r} \mathbf{m}_i = H_\Pi \mathbf{m}_r + \mathbf{e}_i \left( \frac{a}{d\, \zeta} \right) \tag{2.67}$$

where $a := d - \mathbf{n}^\top \zeta K^{-1} \mathbf{m}$ is the orthogonal distance of the 3D point $\mathbf{M}$ (of which $\mathbf{m}_r$ and $\mathbf{m}_i$ are projections) to the plane $\Pi$, and $\zeta_r$ and $\zeta_i$ are the distance of $\mathbf{M}$ from the focal plane of the first and second camera respectively.

If $\mathbf{M} \in \Pi$, then Equation (2.67) reduces to Equation (2.22). Otherwise, there is a residual displacement, called *parallax*, proportional to the *relative affine structure* $\gamma := \dfrac{a}{d\, \zeta}$ of $\mathbf{M}$, with respect to the plane $\Pi$.

Which can be rewritten as

$$\mathbf{m}_i \simeq H_\Pi \mathbf{m}_r + \mathbf{e}_i \gamma \tag{2.68}$$

where $H_\Pi$ is the homography induced by plane $\Pi$ , $\mathbf{e}_i$ is the epipole in $I_i$ and $\gamma$ is the planar parallax (or, simply, *parallax* if the context is clear), which can be interpreted as the displacement between the point $H^\Pi \mathbf{m}_r$, (i.e. $\mathbf{m}_r$ mapped via the homography $H^\Pi$), and its actual corresponding point $\mathbf{m}_i$. Figure 2.6 depicts a geometric representation of the above quantities.

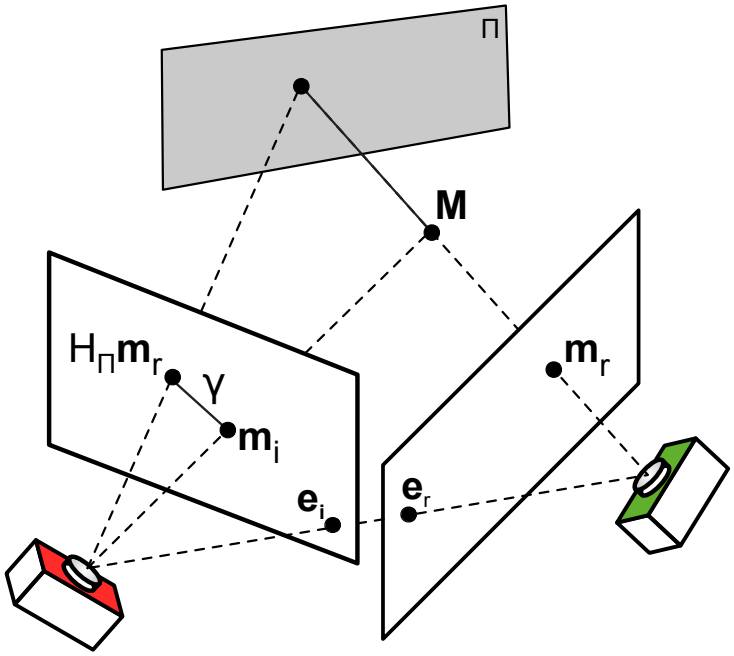


Figure 2.6: parallax $\gamma$ associated with the image point $\mathbf{m}_r$ is the length of the segment joining $\mathbf{m}_i$ and $H_\Pi \mathbf{m}_r$.

Given a certain number of corresponding pairs $(\mathbf{m}_1^k\,;\,\mathbf{m}_2^k) \quad \forall k = 1, \ldots, m$ their parallax is obtained by solving for $\gamma_r^k$ in Equation (2.68):

$$\gamma_r^k = \frac{(\mathbf{m}_i^k \times \mathbf{e}_i)^\top (H^\Pi \mathbf{m}_r^k \times \mathbf{m}_i^k)}{||\mathbf{m}_i^k \times \mathbf{e}_i||^2}. \tag{2.69}$$

Furthermore in the stereo normal case then $H^\infty$ is the identity and the epipole is $\mathbf{e}_i \simeq \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top$, thus parallax in Equation 2.68 results to be proportional to binocular disparity.

To summarize: when $H^\Pi = H^\infty$ the parallax $\gamma$ reduces to the reciprocal of the depth (while in general it is proportional to it), and in the normal case it is proportional to disparity. Moreover, it can be demonstrated that $\gamma$ depends only on the reference image and the plane $\Pi$, and not on the parameters of the second image. This is why the parallax can be seen as a useful generalization of depth and disparity.

By setting the reference image, together with a fixed reference plane $\Pi$, one can thus obtain a projective proxy for the depth of a point that is consistent across several images of a same scene, modulo a global scale factor. In fact, independent estimates of parallax derived from different image pairs $(I_r, I_i)$, differ from each other by an unknown scale factor, which must be estimated independently.

In practice, parallax values are computed using Equation (2.69) for each pixel: the dense set of correspondences $(\mathbf{m}_r^k; \mathbf{m}_i^k)$ on the pair of images $(I_r, I_i)$ is known from a regular stereo matching step; the homography $H_{ri}^{\Pi}$ can be obtained as described in Section 2.3.1 and epipole $\mathbf{e}_i$ can be estimated from epipolar geometry.

At last, we saw that Equation (2.68) describes the relationship between two views through a reference plane. Since $\gamma$ does not depend on the position of the second camera, we can replace the second image with a *new one*, thus we can *transfer* or *warp*, pixel $\mathbf{m}_r$ onto $\mathbf{m}_v$ with:

$$\mathbf{m}_v \simeq H^{\Pi}\mathbf{m}_r + \mathbf{e}_v\gamma \tag{2.70}$$

where $H^{\Pi}$ and $\mathbf{e}_v$ define the position of the new camera. This can be used to transfer a parallax map from one reference frame to another, although this operation brings in several issues related to non-injectivity and non-surjectivity of the transfer map, that are well known in the context of view-synthesis [53]. These last issues will be better explored in Section 4.

### 2.7.2 Depth

The depth of a point is its distance from the focal plane of the camera, as represented in Figure 2.7. If the interior camera parameters are available, stereo correspondences can be converted directly into depth values. The depth values for a given pixel obtained from subsequent frames are directly comparable.

Let $\mathbf{M}$ be a 3D point and let $(\mathbf{m}_r, \mathbf{m}_i)$ be its projections onto the image planes $I_r$ and $I_i$ respectively. Let $P_r = K_r[R_r|\mathbf{t}_r]$ and $P_i = K_i[R_i|\mathbf{t}_i]$ be the perspective projection matrices of the two cameras (that must be known). The equation of the epipolar line of $\mathbf{m}_r$ in $I_i$ is

$$\zeta_i\mathbf{m}_i = \mathbf{e}_i + \zeta_r K_i R_i R_r^{\top} K_r^{-1}\mathbf{m}_r \tag{2.71}$$

where $\mathbf{e}_i := K_i(t_i - R_i R_r^{\top} t_r)$ is the epipole and $\zeta_r$ and $\zeta_i$ are the unknown depths of $\mathbf{M}$ (with reference to $P_r$ and $P_i$, respectively). Thus we can write

$$\mathbf{e}_i = \zeta_i\mathbf{m}_i - \zeta_r\mathbf{m'}_r \tag{2.72}$$

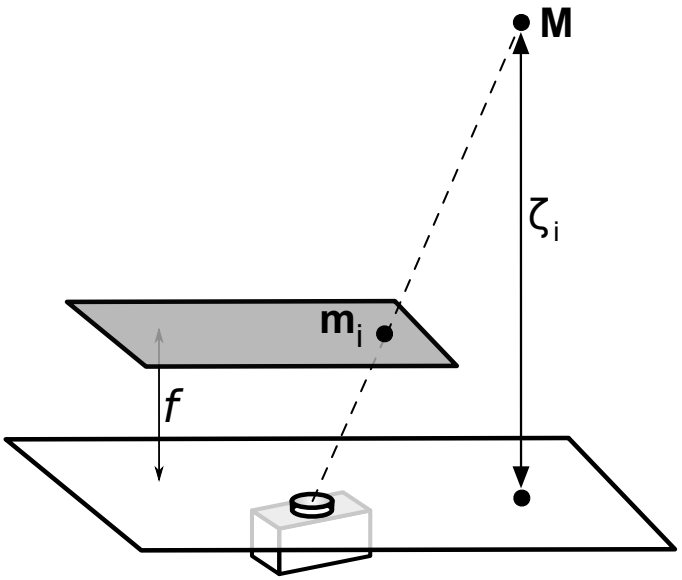


Figure 2.7: the depth $\zeta_i$ is the distance of the 3D point from the focal plane of the camera.

where $\mathbf{m}'_r := K_i R_i R_r^\top K_r^{-1} \mathbf{m}_r$. Since the three points $\mathbf{e}_i$, $\mathbf{m}'_r$ and $\mathbf{m}_i$ are collinear, one can solve for $\zeta_r$ using the following closed form expression [42]

$$\zeta_r = \frac{(\mathbf{e}_i \times \mathbf{m}_i)(\mathbf{m}_i \times \mathbf{m}'_r)}{\|\mathbf{m}_i \times \mathbf{m}'_r\|^2}. \tag{2.73}$$

Since in real situations camera parameters and image locations are known only approximately, the back–projected rays do not actually intersect in space. It can be shown, however, that Formula (2.73) solves Equation (2.72) in a least squares sense, see [42] for more details.

The actual computation of depth values is performed by applying Equation (2.73): $\mathbf{e}_i$ is obtained as the projection of the optical center of the reference camera $C_r$, through the second camera $P_i$; the set of dense correspondences $(\mathbf{m}_r^k; \mathbf{m}_i^k)$ with $k = 1, \ldots, K$, where $K$ is the number of correspondences for the current image pair, is known from the stereo matching step; image points $\mathbf{m}_i^{'}$ are computed according to Equation (2.72).

Please observe how this formulation elegantly avoids the explicit triangulation of $\mathbf{M}$, which would be required in a naive approach.

### Comparison with planar parallax

One can compare Equation (2.73) with Equation (2.69) and observe the similarity of the two formulations. In particular if we solve Equation (2.68)

for $1/\gamma$ instead of $\gamma$ we obtain

$$\frac{1}{\gamma} = \frac{(\mathbf{e}_i \times \mathbf{m}_i)^\top (\mathbf{m}_i \times H_\Pi \mathbf{m}_r)}{\|\mathbf{m}_i \times H_\Pi \mathbf{m}_r\|^2}. \tag{2.74}$$

which coincides with Equation (2.73) if $\mathbf{m}'_r = H_\Pi \mathbf{m}_r$, in which case $\frac{1}{\gamma} = \zeta_r$. In particular, it can be seen that this condition is equivalent to the special choice $H_\Pi = H_\infty$, where $H_\infty$ is the infinite plane homography, i.e. the homography induced by the infinite plane between the pair of images ($I_r$,$I_i$).

### 2.7.3 Disparity

If interior camera parameters are unavailable, but camera motion is constrained, binocular disparity is the first depth–proxy that is readily available from stereo correspondences. The disparity values of a pixel computed from subsequent frames are commensurate only if motion is constrained such that all cameras share a common focal plane (the focal plane is parallel to the image plane and contains the camera center).

When two focal planes are coplanar (i.e. up to a coordinate change, motion is along X axis) then $\zeta_i = \zeta_r := \zeta$ and the epipole is $\mathbf{e}_i = \begin{bmatrix} bf & 0 & 0 \end{bmatrix}^\top$, where $f$ is the focal length $b$ is the magnitude of the translation. Moreover, if $K_i = K_r$ then $\mathbf{m}'_r = \mathbf{m}_r$, hence Equation (2.72) simplifies to:

$$\mathbf{m}_i - \mathbf{m}_r = \begin{bmatrix} bf/\zeta & 0 & 0 \end{bmatrix} \tag{2.75}$$

The disparity, defined only in the normal case, is the non-zero (horizontal) component of the pixel coordinates differences. Two cameras can be always brought to the normal case by rectification [25, 24].

In the case of multiple cameras, since disparity is proportional to the reciprocal of the depth and the depth is defined with respect to the focal plane, there must be a common focal plane in order for disparities to be commensurate. This can always be achieved for $N \leq 3$ cameras by rectification (rotating the focal planes around the optical centers until they coincide with the plane defined by the three centers), but cannot be guaranteed for more cameras, unless camera centers are coplanar.

## 2.8 Uncalibrated motion description

In this section we will first derive a description of a rigid motion that can be achieved when cameras are not calibrated (*uncalibrated motion*), resting on the knowledge of the epipole and the homography of the plane at infinity.

Then we will draw its relationship with the Euclidean description of rigid motions, represented by the special Euclidean group $\mathrm{SE}(3,\mathbb{R})$.

We aim to obtain an equation relating views 1-3 in terms of 1-2 and 2-3. To this end, let us consider Equation (2.67), which expresses the epipolar geometry with reference to a plane, in the case of view pair 1-2:

$$\frac{\zeta_2}{\zeta_1}\mathbf{m}_2 = H_{12}\mathbf{m}_1 + \mathbf{e}_{21}\gamma_1 \tag{2.76}$$

and view pair 2-3:

$$\frac{\zeta_3}{\zeta_2}\mathbf{m}_3 = H_{23}\mathbf{m}_2 + \mathbf{e}_{32}\gamma_2. \tag{2.77}$$

By substituting the first into the second, we obtain:

$$\frac{\zeta_3}{\zeta_1}\mathbf{m}_3 = H_{23}H_{12}\mathbf{m}_1 + (H_{23}\mathbf{e}_{21} + \mathbf{e}_{32}\frac{d_1}{d_2})\gamma_1 \tag{2.78}$$

where $d_1$ and $d_2$ are the distances of the plane $\Pi$ from the first and the second camera respectively.

Comparison with Equation (2.67) yields:

$$H_{13} = H_{23}H_{\Pi 12} \quad \text{and} \quad \mathbf{e}_{31} = H_{23}\mathbf{e}_{21} + \mathbf{e}_{32}\frac{d_1}{d_2} \tag{2.79}$$

The ratio $\dfrac{d_1}{d_2}$ in general is unknown, but if $\Pi$ is the plane at infinity then $\dfrac{d_1}{d_2} = 1$ (please note that this is approximately true for planes distant from the camera). Therefore, taking the plane at infinity as $\Pi$, Equation (2.78) writes:

$$H_{\infty 13} = H_{\infty 23}H_{\infty 12} \quad \text{and} \quad \mathbf{e}_{31} = H_{\infty 23}\mathbf{e}_{21} + \mathbf{e}_{32} \tag{2.80}$$

Albeit, in general, homographies can be computed only up to a scale factor, in the case of the infinity plane homography, if internal parameters are assumed constant (as we do henceforth), the scale is fixed by the requirement that $\det(H_\infty) = 1$.

Let $\mathbf{e}_{ji}$ and $H_{\infty ij}$ be the epipole and the plane at infinity, respectively, linking two cameras $i$ and $j$. The matrix

$$D_{ij} := \begin{bmatrix} H_{\infty ij} & \mathbf{e}_{ji} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.81}$$

is called the *uncalibrated rigid motion matrix*.

As opposed to a Euclidean rigid motion matrix, $D_{ij}$ contains the homography of the plane at infinity in place of the rotation, and the epipole in lieu of the translation.

In matrix form Equation (2.80) writes:

$$D_{13} = D_{23}D_{12} \tag{2.82}$$

Interestingly enough, uncalibrated rigid motion matrices $D_{ij}$ follow the same multiplicative composition rule as the homogeneous rigid motion matrices $G_{ij}$ of $\mathrm{SE}(3,\mathbb{R})$. In a sense, $D_{ij}$ is a homogeneous representation of the rigid motion at the uncalibrated stratum. This observation leads to the realization that the uncalibrated rigid motions form a group that is isomorphic to $\mathrm{SE}(3,\mathbb{R})$, under the assumption of constant internal parameters $K$. Let

$$G_{ij} := \begin{bmatrix} R_{ij} & \mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3,\mathbb{R}) \tag{2.83}$$

be a matrix that represent a rigid motion, where $R$ is a rotation matrix and $\mathbf{t}$ is a vector representing a translation.

First, let us observe that the operator $\varphi_K : \varphi_K(G_{ij}) = D_{ij}$ that maps calibrated operations into the uncalibrated stratum, where the infinity plane homography substitutes the rotation and the epipole substitutes the translation, is a conjugacy map:

$$\varphi_K(G_{ij}) = D_{ij} = \begin{bmatrix} KR_{ij}K^{-1} & K\mathbf{t}_{ij} \\ \mathbf{0} & 1 \end{bmatrix} = \tilde{K}G_{ij}\tilde{K}^{-1} \tag{2.84}$$

with

$$\tilde{K} = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2.85}$$

Then, it is easy to shown that $\varphi_K$ is an homomorphism:

$$\varphi_K(G_{23})\varphi_K(G_{12}) = \tilde{K}G_{23}\tilde{K}^{-1}\tilde{K}G_{12}\tilde{K}^{-1} = \tilde{K}G_{23}G_{12}\tilde{K}^{-1} = \varphi_K(G_{23}G_{12}) \tag{2.86}$$

and, being $\varphi_K$ invertible, it is an isomorphism.

Thanks to the fact that uncalibrated motions are isomorphic to $\mathrm{SE}(3,\mathbb{R})$, every operation carried out in the uncalibrated stratum reflects itself consistently in the Euclidean stratum, even if the map $\varphi_K$ is unknown. Thanks to this isomorphism, and since $SE(3,\mathbb{R})$ is a Lie group, it is possible to continuously parametrize the uncalibrated motion of the virtual camera as:

$$D_{rv} := D_{ri}^t := \exp\left(t\log(D_{ri})\right) \quad t \in \mathbb{R}. \tag{2.87}$$

Varying the value of $t$, we obtain a 1-parameter family orientations which naturally interpolates/extrapolates the orientations of the of reference and the auxiliary cameras along a geodesic path. The infinite plane homography $H_{\Pi}^{rv}$, along with the epipole $\mathbf{e}_v$ can be extracted from $D_{rv}$ according to

$$D_{rv} = \begin{bmatrix} H_{\Pi}^{rv} & \mathbf{e}_v \\ 0 & 1 \end{bmatrix} \tag{2.88}$$

when the reference plane $\Pi$ is the plane at infinity (i.e. $H_{\Pi} = H_{\infty}$) and $D_{rv}$ is the orientation of the virtual camera.

# Chapter 3

# Motion-stereo

## Contents



## 3.1 Introduction

This part of the thesis deals with the problem of *motion-stereo*, i.e. the estimation of depth (or a depth-proxy) in a monocular sequence of images taken by a moving camera [75]. Whereas in binocular stereo two cameras separated by a fixed baseline are employed, in motion-stereo a single camera moves through a static scene. As a result, over a period of time, the camera traverses a "baseline" of undetermined length. The grounds for addressing such problem lie in the attempt to solve the *accuracy-precision* trade-off in stereo matching, which can be summarized as follows: due to quantization errors, the estimated disparity is more precise with a larger baseline, but the matching is less accurate, because of the exacerbation of perspective and radiometric nuisances that cause false and missing matches. There is

manifestly a conflict between accuracy and precision, which motion-stereo approaches attempt to resolve. Early works in motion-stereo [71, 52, 62], integrate depth maps from different frames into a single map. They require motion and camera parameters to be known, and most of them restricts to lateral motion. A common drawback is that they warp the disparity map from frame to frame, thereby introducing errors and approximations that disrupt the prediction, and make the integration pointless. More recent motion-stereo approaches aggregate measures in a discretized 3D volume [73, 55, 80], but they need calibrated cameras as well.

The multiple-baseline approaches [48, 56, 43] generalize the binocular one by computing an aggregated matching cost which considers all the images simultaneously, and then proceed as in the binocular case. These methods require camera centers to be collinear (equivalent to lateral motion). Generalizations of these approaches can be found in the multi-view stereo literature, where the aggregated cost is computed along the optical ray in a discretized volume [34, 28].

From the geometrical point of view, the problem raised by motion-stereo is how to set a common reference frame where measures from different images can be integrated. The discretized volume seems the natural choice, however computation in 3D space can be avoided by considering image-based quantities such as depth, binocular disparity or *planar parallax*. As shown in Section 2.7 when camera parameters and its motion are unknown, planar parallax is a suitable depth–proxy that generalizes disparity and depth. This approach based on pixel-based measures – also called "iconic" – is motivated by applications like view-synthesis, video interpolation and enhancement (frame rate up-conversion) and free viewpoint 3D TV.

Apart from the accuracy-precision trade off, it is important to notice that motion-stereo approaches, have another intrinsic advantage: the fact that information coming from multiple images is integrated into one reference frame allows to unveil many areas that would be otherwise occluded when processing images in pairs. Each new point of view brings new information to the depth–proxy estimation. This is very important for any application that relies on the depth map as input data.

## 3.2 Motivation

Once a common reference is set, the problem posed by motion-stereo can be seen as the one of integrating measures from different images, hence the focus of this chapter will be on this *data-fusion* problem. Our solution aims at being agnostic with respect to: i) the depth–proxy that is being used

ii) the binocular stereo matching algorithm which is considered as part of the input of our method. Indeed both the depth–proxy measures and the disparity maps produced by the binocular stereo matching can be considered as inputs of the data-fusion algorithm. In Section 3.3 we set parallax as our depth-proxy for coherence with the method proposed in Chapter4 although other depth-proxies could be adopted without impacts on the method.

As in [52, 71], we use a *dynamic* approach, as we apply Kalman filtering for recursive estimation of depth maps by combining measurements along the time line and within a spatial neighborhood. Pixel–wise depth measures are relaxed by considering the information coming from the neighbors within the same *superpixels*, using a spatial Kalman filter. An analogous result has been obtained in [52] by smoothing disparity maps with piecewise continuous splines, where a regularization-based smoothing is used to reduce measurement noise and to fill in areas of unknown disparity. Other methods perform adaptive smoothing in a *edge–aware fashion*, e.g. [45] where temporal consistency is enforced among different depth maps using an edge-aware Gaussian filtering extended to the temporal dimension in video volumes, or [65] where the depth map is filled by solving a least square error problem using edge and temporal information as weights. With respect to our approach, the key difference is that these works are post–processing approaches that aim at improving the quality of depth maps whereas our method uses edge information (in the form of superpixels) to be aware of which neighbors are relevant *while* updating depth values on the current reference map.

In both temporal and spatial dimensions, the depth measures are trusted using confidence metrics attached to the measures.

As for experimental analysis, it is left to Chapter 5), further in the thesis.

## 3.3 Our method

The input of the method is a monocular video sequence of $N$ frames, of which one is set as the reference, denoted by $I_r$. For every pair of images $(I_r, I_i)$ (where, for example, if $I_r = I_1$ and $i = 2, \ldots, N$), estimates of the depth–proxy map relative to the reference frame are computed independently by binocular stereo matching. The iterative pairwise processing mechanism is shown in Figure 3.1. We designate parallax as the depth–proxy, because the application we intend to focus on in the next Chapters deals with uncalibrated data. Moreover, in general, parallax is the more general depth–proxy and subsumes all the others. However disparity or depth can be used instead when certain conditions are fulfilled. Regarding camera motion, the only assumption made herein is that a relevant portion of the reference frame is kept

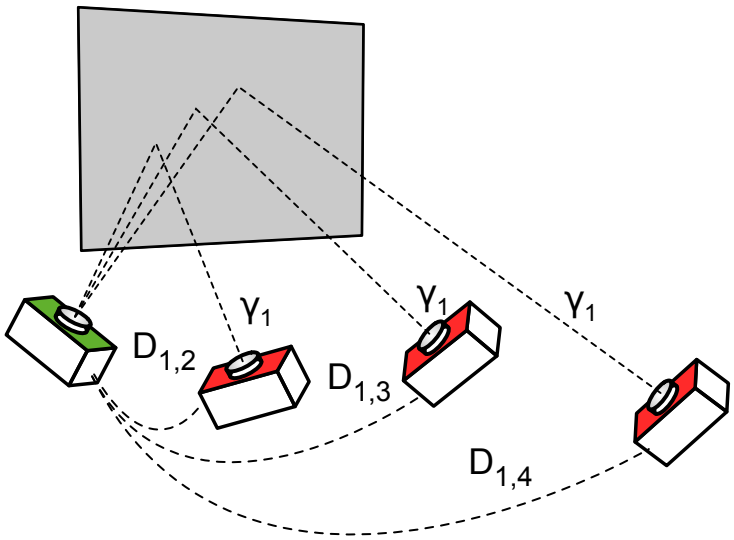


Figure 3.1: pairwise processing. The reference camera (green in this example) is kept fixed whereas the second element (red) of the pair changes at each iteration.

visible at all the subsequent frames of the video segment. When this assumption fails, a new reference frame is set and the filter is restarted. Note that information about the temporal trajectory is not used, i.e. the pairs could occur in any order. This property has two main advantages: i) pairwise processing can be performed independently, making the algorithm highly parallelizable ii) sets of still images as input, instead of video sequences, can be processed.

Each of the $N-1$ independent estimates of the parallax map contains errors and valuable information, the goal of data fusion is two-sided: on one hand it is to enhance the valuable information while smoothing out the errors, indeed the classic rationale behind motion-stereo is to break the accuracy vs precision trade-off by using multiple baseline lengths (a small baseline implies few occlusions, easier stereo matching but raw quantization of the parallax, whereas a large baseline implies better quantization of the parallax but more occlusions and harder matching). On the other hand, the aspect that we believe to be of more useful is that the information coming from different images, through the integration framework, is *gathered* on a reference frame. Such information contains the contribution of several different point of views. This drastically decreases the number of occluded pixels. In our framework all these parallax maps are combined together using spatial and temporal coupled Kalman filters, achieving more stable and accurate values. Superpixels provide the spatial support for the relaxation

of parallax values among the image neighbors.

Figure 3.2 shows a schematic overview of our method. Each step will be described in the following sections.

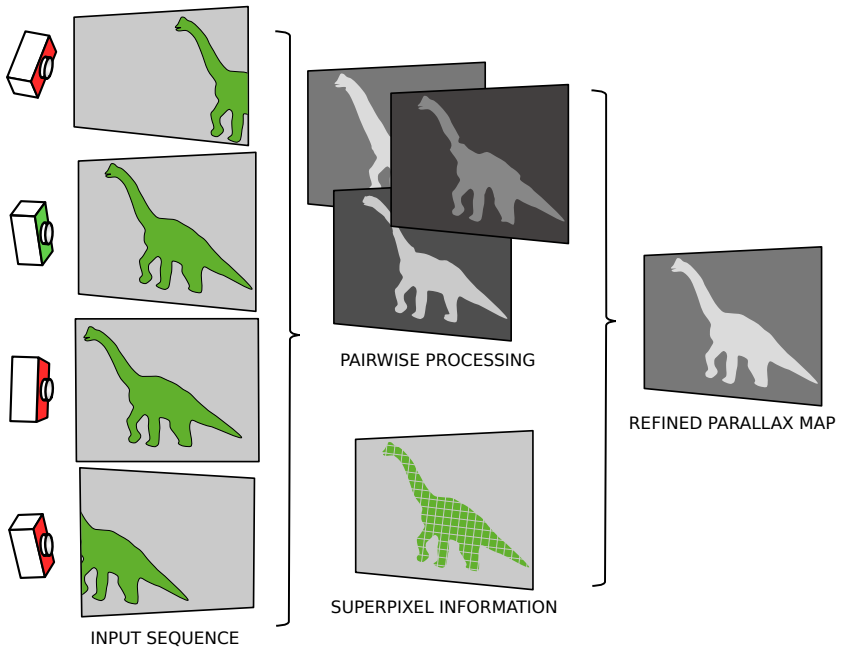


Figure 3.2: overview of the motion-stereo method. The reference image $I_r$ (the image with the green camera in this example) is selected from the input sequence of $N$ elements ($N = 4$ in this example) and it is oversegmented into superpixels (bottom image of the second column). The pairwise processing computes $N-1$ maps, one for each pair $(I_r, I_i)$. The Kalman filter merges the pairwise maps using superpixel information and confidence measures, and it yields a refined parallax map.

## 3.4 Stereo Matching

The image pairs ($I_r$, $I_i$) need to be rectified for the subsequent stereo matching step to work. In particular, each pair must be rectified independently, unless the camera centers are coplanar. In the calibrated scenario, one can use [25], where the algorithm takes the perspective projection matrices of the original cameras and computes a pair of rectifying projection matrices. When internal parameters are unknown, a suitable approach is [24], which assumes that a number of corresponding points are available and we seeks

the rectifying homographies that make the original points satisfy the epipolar geometry of a rectified image pair. Both approaches were summarized in Section 2.4.

Dense correspondences between $I_r$ and $I_i$ can be obtained using any stereo matching algorithm. In our experiments, since we focus on the *integration framework* and not on the performance of the stereo itself, we used a simple block–matching with Normalized Cross Correlation (NCC) as a matching score

$$\frac{\sum_{n\in W}(I_r(x_n,y_n)-\mu_r)(I_i(x_n,y_n)-\mu_i)}{\sqrt{\sum_{n\in W}(I_r(x_n,y_n))^2}\sqrt{\sum_{n\in W}(I_i(x_n,y_n))^2}} \tag{3.1}$$

where $\mu_r$ and $\mu_i$ are the means of window $W$ respectively in images $I_r$ and $I_i$.

After the block–matching step, we perform a left–right consistency (LRC) check, which is a standard procedure based on the uniqueness principle [51]. The consistency is verified if $p$ is matched with $p'$ when searching on the pair ($I_r$, $I_i$) and $p'$ is matched with $p$ when searching on the pair ($I_i$, $I_r$), where $p$ is a point in $I_r$ and $p'$ is a point in $I_i$. All non–consistent matches are discarded. This procedure skims the results from occluded pixels and bad matches. Dense correspondences are then transferred back to the original reference images by applying the inverse of the rectifying homographies (de–rectification).

## 3.5 Confidence Measures

During the stereo matching step, a confidence map, associated with the parallax map, is also computed. For each pixel we integrate the LRC check with a confidence indicator based on the matching score profile.

Thus, the confidence associated with the parallax computed at pixel $i$ is,

$$\varphi(i) := \begin{cases} 0 & \text{if pixel } i \text{ fails the LRC check} \\ \phi_*(i) & \text{otherwise} \end{cases} \tag{3.2}$$

where $\phi_*(i)$ is one of the metrics discussed below. The confidence $\varphi(i)$ varies in $[0,1]$, where 0 means that pixel $i$ is *totally unreliable* and 1 means *maximally confident*.

We tested and compared different confidence measures, briefly reported here. The reader is referred to [37] for a more detailed description.

In the following $c(d)$ denotes the matching cost – normalized in $[0,1]$ – associated with disparity hypothesis $d$. Since NCC is a similarity measure

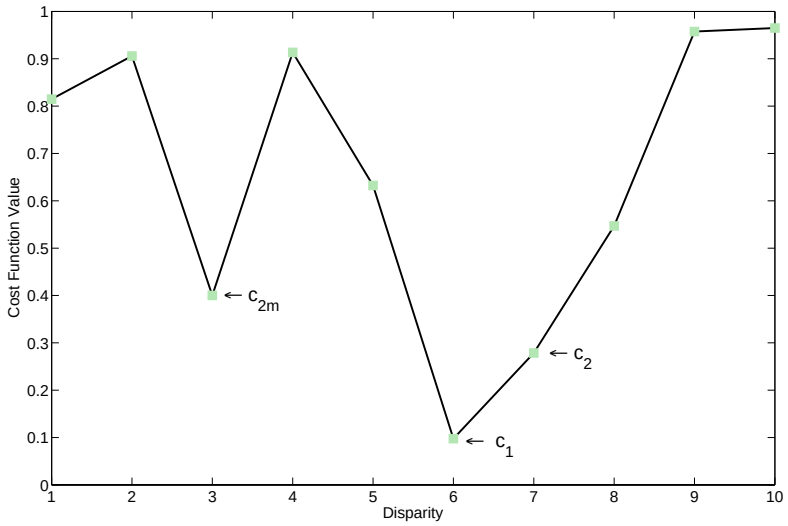


Figure 3.3: example of a cost function with a 10 pixel disparity range. The minimum cost is denoted by $c_1$, the second smallest cost value is denoted by $c_2$ and the second smallest value that is also a local minimum is represented by $c_{2\mathrm{m}}$.

and all the confidence measures are defined using a cost function, $1 - \mathrm{NCC}$ will be used instead.

The minimum cost for a pixel and its correspondent disparity value are respectively denoted by $c_1$ and $d_1$ (i.e. $c(d_1) = c_1 = \min(c(d))$). The second smallest cost value is denoted by $c_2$, while the second smallest value that is also a local minimum is represented by $c_{2\mathrm{m}}$ (see Figure 3.3 for an example).

A very simple confidence metric is the matching score:

**Matching Score (MSM):**

$$\phi_{\mathrm{MSM}} = 1 - c_1. \tag{3.3}$$

The first group of measures assess the cost function around its minimum by comparing it to the following smaller cost values ($c_2$ or $c_{2\mathrm{m}}$) or to the disparity neighbors.

**Curvature of the cost function (CUR):**

$$\phi_{\mathrm{CUR}} = \frac{2 + (-2c_1 + c(d_1 - 1) + c(d_1 + 1))}{4} \tag{3.4}$$

**Peak Ratio (PKR):**

$$\phi_{\mathrm{PKR}} = 1 - \frac{c_1}{c_{2\mathrm{m}}} \tag{3.5}$$

**Maximum Margin (MMN):**

$$\phi_{\mathrm{MMN}} = \frac{c_2 - c_1}{c_2} \tag{3.6}$$

**Winner Margin (WMN):**

$$\phi_{\mathrm{WMN}} = \frac{c_{2\mathrm{m}} - c_1}{c_{2\mathrm{m}}} \tag{3.7}$$

The following metrics take into account the entire cost curve by assuming that it follows a normal distribution.

**Maximum Likelihood Measure (MLM):**

$$\phi_{\mathrm{MLM}} = \frac{e^{-\frac{c_1}{2\sigma^2_{\mathrm{MLM}}}}}{\sum_d e^{-\frac{c(d)}{2\sigma^2_{\mathrm{MLM}}}}} \tag{3.8}$$

**Attainable Maximum Likelihood (AML):**

$$\phi_{\mathrm{AML}} = \frac{1}{\sum_d e^{-\frac{(c(d)-c_1)^2}{2\sigma^2_{\mathrm{AMLM}}}}} \tag{3.9}$$

We also considered two special measures to use as a touchstone. GT assigns confidence 1 if the corresponding pixel's disparity is correctly computed and 0 otherwise, according to the ground truth. UNI is an uninformed metric that assigns the same confidence to all the pixels.

**Ground truth (GT):**

$$\phi_{\mathrm{GT}} = \begin{cases} 1 & \text{if disparity is correct} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

**Uniform (UNI):**

$$\phi_{\mathrm{UNI}} = \mathrm{cost} \tag{3.11}$$

## 3.6 Temporal integration

Temporal integration of parallax data is performed through a simple 1-d Kalman filter with constant (up to a scale) state and direct measurement model. Let $x_t(m)^+$ be the best parallax estimate (the state) available at time $t$ for pixel $m$, let $p_t(m)^+$ be its variance; let $z_t(m)$ be the parallax measured at pixel $m$ of frame $t$ (via stereo matching), and let $r_t(m)$ be its variance. The Kalman filter equations write:

$$\textbf{Process:}\quad x_t = s \cdot x_{t-1} + w_t, \quad \mathrm{Var}(w_t) = q_t \tag{3.12}$$

$$\textbf{Measure:}\quad z_t = x_t + v_t, \quad \mathrm{Var}(v_t) = r_t \tag{3.13}$$

$$\textbf{Prediction:}\quad x_t^- = s \cdot x_{t-1}^+, \quad p_t^- = s^2 \cdot p_{t-1}^+ + q_t \tag{3.14}$$

$$\textbf{Update:}\quad x_t^+ = \frac{x_t^- r_t + z_t\, p_t^-}{p_t^- + r_t}, \quad p_t^+ = \frac{p_t^- r_t}{p_t^- + r_t} \tag{3.15}$$

Where $x_t^-$ and $p_t^-$ represent the *a priori* estimations of the state and its variance respectively, whereas $x_t^+$ and $p_t^+$ are their updates using measurement $z_t$ and its variance $r_t$. The variable $m$ has been omitted as the treatment is uniform over the pixels.

It turns out to be more convenient to formulate the update equations in terms of the inverse variance, which will be henceforth called *information* (the *Fisher information* of a random multivariate distribution is the inverse covariance [9]). Let ${}^ip = 1/p$ and ${}^ir = 1/r$, then Equation (3.15) becomes:

$$x_t^+ = \frac{z_t\, {}^ir_t + x_t^-\, {}^ip_t^-}{{}^ir_t + {}^ip_t^-}, \quad {}^ip_t^+ = {}^ir_t + {}^ip_t^-. \tag{3.16}$$

The process model contains a multiplicative factor $s$ which takes into account the fact that independent measures of the parallax are scaled by an unknown factor: in fact, the current state is always scaled to match the measure. The scale $s$ is estimated by comparing $x_{t-1}^+$ with $z_t$ in a robust (outliers resilient) way. First the ratio between the two maps is computed pixelwise, considering only the pixels that, given their information value, are the most reliable (i.e. upper quartile of the ${}^ir_t$ map); then the ratios which are greater than 5.2 median absolute deviations from the median are discarded as outliers (a.k.a. x84 rejection rule [61]); finally the scale is computed as the mean of the inlier ratios.

The process noise $w_t$ accounts for the errors introduced in predicting the state. Since the state we are estimating is constant (up to a scale), and no approximation are made in the prediction, our temporal model has $q_t = 0$.

The measurements noise $v_t$ models errors that affect the parallax estimation, hence its information ${}^ir_t$ is directly related to the confidence $\varphi$ defined in

Equation (3.2). We use ${}^{i}r_t = 12\varphi$, which sets the maximum information for a correct parallax value to the reciprocal of the variance of the quantization noise (which is 1/12).

The update of the filter state takes place through a *validation gate* to ensure that outliers do not skew the estimate. In particular, we consider the Mahalanobis distance as a gating criterion [67]. The update is accepted only if:

$$\frac{(x_t^- - z_t)^2}{p_t^- + r_t} \leq \chi_1^2(\alpha) \tag{3.17}$$

where $\chi_1^2(\alpha)$ is the upper $100\alpha^{th}$ percentile of a chi-square distribution with 1 d.o.f. (we used $\alpha = 0.98$).

The update equation fails when ${}^{i}p_t^- = {}^{i}r_t = 0$, because a 0/0 form is obtained. This happens at $t = 1$ if a reliable measure (${}^{i}r_1 \neq 0$) is not available, and at any subsequent $t$ until a reliable measure is found. This special case is handled within the validation gate by simply skipping the update whenever ${}^{i}r_t = 0$. Please note that ${}^{i}r_t = 0$ means that the pixel is unmatched (not visible in the conjugated image).

In the most general case, the filter starts with ${}^{i}p_0^- = 0$ and $x_0^-$ undefined, however, if a parallax map is available for the reference frame of the previous video segment, it can be warped to the current reference frame with Equation (2.70) and provides a partial initialization for the state. The information of the warped parallax is downweighted by a factor 10 to account for errors introduced by the warping.

Finally, it is worth noting here that this simple Kalman filter – ignoring the scale $s$ – reduces to a weighted average of the measures $z_t$ with the information values ${}^{i}r_t$ as weights, as can be observed by solving the recursive update equations, thus obtaining:

$${}^{i}p_t^+ = \sum_{k=0}^{t} {}^{i}r_k \tag{3.18}$$

$$x_t^+ = \frac{z_t\, {}^{i}r_t + x_{t-1}^+ \sum_{k=0}^{t-1} {}^{i}r_k}{\sum_{k=0}^{t} {}^{i}r_k} = \frac{\sum_{k=0}^{t} z_k\, {}^{i}r_k}{\sum_{k=0}^{t} {}^{i}r_k}. \tag{3.19}$$

Indeed, the middle term of Equation (3.19) is the well known formula for the recursive computation of the average. A matrix equivalent to Equations (3.18) and (3.19) can be also derived as the least squares solution to the problem of optimally (in terms of Mahalanobis distance) combining an ensemble of independent (multivariate) random variables which estimate the same true parameter [59]. The advantage of the Kalman filter is in its recursive formulation, which leads to a *causal* filter that produces at each time

instant (dynamically) the best estimate based on the past measures, whereas the weighted average considers all the measures in a batch.

## 3.7 Spatial support

### 3.7.1 Superpixel extraction

The spatial relaxation requires to identify a neighborhood of each pixel in the reference image where the depth is ideally constant. This is achieved by computing *superpixels*, i.e., compact and almost uniform regions of the image, using the Simple Linear Iterative Clustering algorithm (SLIC) [8], which starts with a regular grid of centers and then locally clusters pixels in the combined five–dimensional color (CIELab) and image coordinates space. The density of the initial grid plus a regularization coefficient are the only two parameters that need to be set. The (approximated) desired size of the superpixels is specified so that

$$\text{number of initial cells} = \frac{\text{reference frame resolution}}{\text{desired size of the superpixel}}.$$

Some segmentation examples are shown in Figure 3.4.

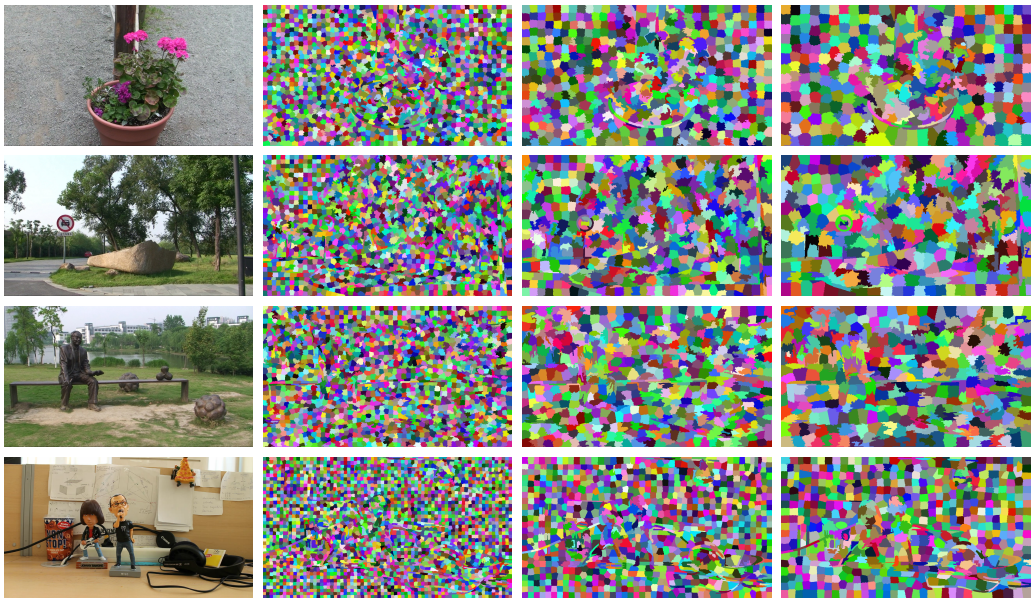

Figure 3.4: examples of superpixel extraction using different values for the grid density, which controls the size of the superpixel.

### 3.7.2 Extension to the spatial domain

Once the superpixels are extracted, in principle, the integration with the spatial neighborhood should take place by introducing spatial correlations between neighboring pixels, which entails a state vector of the size of the image ($M$) and a non-diagonal ($M \times M$) covariance matrix. However, this would become too computationally demanding, so we approximate its effect by modifying the prediction step of the temporal 1-d Kalman filter, without changing its structure. In particular, in the prediction formula (3.14), we substitute the state $x^+_{t-1}$ with a smoothed state $\hat{x}^+_{t-1}$ that depends on the neighboring pixels *within the same superpixel* (and the information ${}^ip^+_{t-1}$ accordingly).

To be consistent with the temporal dimension, we derive $\hat{x}^+_t$ within the Kalman filter framework. As mentioned in [52], an alternative approach to the prediction of state variance is the so-called "exponential age-weighting" of measurements, where the current variance is inflated by a small multiplicative factor [9]:

$$p^-_t = (1+\epsilon)p^+_{t-1}. \tag{3.20}$$

Equations (3.18) and (3.19) can be generalized to:

$${}^ip^+_t = \sum_{k=0}^{t} {}^ir_k\delta^{t-k} \tag{3.21}$$

$$x^+_t = \frac{\sum_{k=0}^{t} z_k\ {}^ir_k\delta^{t-k}}{{}^ip^+_t} \tag{3.22}$$

where we introduced $\delta = 1/(1+\epsilon)$ which is the inverse of the exponential age-weighting, since we are dealing with information instead of variance.

These formulae can be translated into the spatial domain by substituting the exponential age-weighting term, which gives smaller weights to older measures, with an *exponential distance-weighing* term (with a parameter $\rho < 1$) which serves the purpose of weighting the measure according to the distance to the current pixel. Let $x(m)^+$ be the parallax (state) at pixel $m$ and let ${}^ip(m)^+$ be its information value:

$${}^i\hat{p}^+(m) = \sum_{q\in\Omega(m)} {}^ip^+(q)\rho^{\|m-q\|} \tag{3.23}$$

$$\hat{x}^+(m) = \frac{\sum_{q\in\Omega(m)} x^+(q)\ {}^ip^+(q)\rho^{\|m-q\|}}{{}^i\hat{p}^+(m)} \tag{3.24}$$

where $\Omega(m)$ is the superpixel to which pixel $m$ belongs. In this paragraph we will omit the constant temporal index, as we are dealing with the spatial dimension only.

The information of the combined measure is the *sum* of the information values of the original measures (with exponential distance-weighing), so it is much greater than the original point-wise information. This would be correct only if the combined measures are *not correlated*, but this is not the case here, for neighboring parallax measures are indeed correlated.

The problem of combining correlated measures of the same variable has been addressed in the data fusion literature, and one solution that provides consistent estimates is the Covariance Intersection approach [72], where "consistent" means that the estimated covariance is an upper bound of the true covariance. When considering scalar variables, Covariance Intersection boils down to selecting the measure with the highest information value:

$$ {}^{i}\hat{p}^{+}(m) = \max_{q \in \Omega(m)} \{ {}^{i}p^{+}(q) \rho^{\|m-q\|} \} \tag{3.25} $$

$$ \bar{q} = \arg \max_{q \in \Omega(m)} \{ {}^{i}p^{+}(q) \rho^{\|m-q\|} \} $$

$$ \hat{x}^{+}(m) = x^{+}(\bar{q}) \tag{3.26} $$

Please note that Equation (3.26) would yield the same value of $\hat{x}^{+}(m)$ for each pixel $m \in \Omega(m)$ if $\rho = 1$, whereas with $\rho < 1$ it produces different values within the same superpixel. The value of $\rho$ can be computed as a function of the cut-off radius $\theta$ (in pixels) at which the function $\rho^{\|m-q\|}$ falls below a given threshold, $10^{-2}$ in our implementation. The value of $\theta$ should be of the order of the stereo matching window size. Please note that, as the smoothing is limited within the superpixel, there is no point in choosing $\theta$ larger than the superpixel radius. The Matlab pseudo-code reported in Algorithm 2 illustrates one iteration of the filter: the function takes in input the current state estimate `(x,ip)` and the measure `(z,ir)` and updates the state estimate accordingly. It also shows how temporal and spatial integration are iterated. The `compute_scale` function implements the robust method described in the text (after Equation (3.16)). In the `for` cycle we have been sloppy about the difference between linear indexing and subscripts (row, column), for the sake of readability. Also the subtraction in `pix-pix(j)` is not syntactically correct, as `pix(j)` should have been replicated. The actual working code is available on-line [77].

**Algorithm 2** KALMAN FILTER WITH SPATIAL SUPPORT

```
function [x,ip]  = STKalmanStep(x,ip,z,ir)
% update state (x,ip)
% in the face of measure (z,ir)
% prediction
s=compute_scale(x,z);
x=s*x;
ip=1/s^2 * ip;
% validation gate
res=((x-z).^2)./(1./ip + 1./ir);
v=(res <= Chi) & ir>0;
% temporal update
x(v)=(z(v).*ir(v)+x(v).*ip(v))./(ir(v)+ip(v));
ip(v)=ip(v) + ir(v);
% spatial relaxation
for k=1:numel(superpixels)
pix=superpixels(k).PixelList;
for j=1:length(pix)
w=rho.^sum(sqrt((pix-pix(j)).^2),2);
[val,pos]=max(ip(pix).*w);
ip(pix(j))=val;
x(pix(j))=x(pix(pos));
end
end
```

# Chapter 4

# View-synthesis

## Contents



## 4.1 Introduction

As we already stated view-synthesis is the problem of rendering virtual images starting from actual images. Applications include the generation of a 3DS video from a monocular one [78, 41, 14, 81], upsampling of video sequences in order to achieve slow-motion effects (e.g., [15, 47, 29]), video-conferencing ([40, 44, 54]).

When cameras are *calibrated*, i.e. when both internal and external parameters are available, given the depth of an image point, it is straightforward to compute the position of the point in virtual image from any viewpoint. Techniques based on this paradigm, known as *Depth Image Based Rendering*

(DIBR), have been extensively studied and several solutions are available in the literature ([82] and references therein).

But when dealing with the 3DS conversion problem, calibration data is hardly available. Despite this, many works addressing this application ([78, 41, 14, 81]) assume some knowledge on the camera parameters, and fall within the DIBR family described above. The *uncalibrated* view-synthesis (UVS) is less explored and more challenging for several reasons.

First of all, depth cannot be used in uncalibrated situations, and suitable depth-proxies must be defined, together with proper warping functions based on fundamental matrices [46], trilinear tensors [10], or plane-parallax representation [39, 66].

Second, specifying the external *orientation* (position and attitude) of virtual views is unnatural, since they are embedded in a projective frame, linked to the Euclidean one by an *unknown* projective transformation. Only few works address this problem. In [21] an automatic method based on the planar parallax as a geometry proxy is presented: given two or more reference images, the possible uncalibrated orientations describe a 1-parametric trajectory obtained interpolating or extrapolating the relative motion among reference images. This approach is expanded in [27] by extending to 3-parametric trajectories, thus allowing additional positions along and orthogonally the line of sight, and in [19] by defining a 1-parametric rectified trajectory that, when derectified, is compatible with the one in [21], and is more resilient to errors induced by poor epipolar geometry estimation. In the upsampling of video sequences the virtual views are always very close to the reference view, hence simple interpolation along motion vectors is widely used.

Finally – but this issue is shared with DIBR techniques – several subproblems have to be addressed when applying warping functions: *folding*, which occurs when two or more pixels in the reference image are warped to the same pixel in the virtual image, *holes*, which may be caused either by occlusions or by missing geometric information and result in points that are supposed to be visible in the virtual image but do not have a correspondent point in the source image, *magnification*, when a projected area of a surface is much bigger in the virtual view than the source, and at last *resampling*, due to the discrete nature of digital images, because mapping to the virtual view will in general yield non-integer coordinates.

## 4.2 Motivation and contributions

In this chapter we present our method for uncalibrated view-synthesis (UVS) in the context of 3DS conversion from a set of monocular and uncalibrated

images. We developed a fully automatic UVS pipeline that addresses most of the critical problems arising in uncalibrated scenarios. The method takes inspiration from the pipeline presented in [23], but instead of transferring points directly from the reference image to the novel one (forward), we use a *backward mapping* strategy which yields finer results and allows to combine information coming from *multiple reference images*, blending several parallax maps into one. At last, we propose a simple and suitable method to fill holes in the final virtual image and cope with resampling artifacts.

As for the motion-stereo method the experimental evaluation is reported in the next Chapter 5.

## 4.3 Method

In this section we describe the steps of our method. The input is a set of reference images $I_r$ and a set (not necessarily disjoint) of auxiliary images $I_i$. One or more parallax maps are computed for the reference image $I_r$, with the support of auxiliary images $I_i$ (the ones with the highest overlap with $I_r$). These parallax maps are transferred (forward) to the virtual image $I_v$ and merged together. The resulting map is then used to synthesize $I_v$ by (backward) mapping to the *right* pixel in the *right* source image. Please note that the use of parallax instead of disparity is crucial to allow the fusion of multiple parallax maps, as discussed in Section 2.7.

### 4.3.1 Stereo processing

The purpose of this part of the algorithm is to compute the parallax value $\gamma$ for each pixel of the reference image $I_r$, with the support of an auxiliary image to constitute a stereo pair.

First the image pair is rectified. Since the internal parameters are unknown, we use the uncalibrated procedure described in [24] and summarized in Section 2.4, which relies on sparse correspondences. To this end, first SIFT features are extracted in both images and descriptors are matched, and then a RANSAC estimation of the fundamental matrix is performed in order to discard outlier matches.

The subsequent step is the stereo matching. We use the Matlab/OpenCV implementation of the Semi-Global Matching (SGM) algorithm [35]. In SGM, matching cost is based on mutual information. Mutual information (MI), is a measure, for model alignment based on the entropy $H$, i.e. the amount of uncertainty in a probability density function. $MI$ has three terms: entropy

of the first image, entropy of the second image and joint entropy

$$MI_{I_r,I_i} = H_{I_r} + H_{I_i} - H_{I_r,I_i} \tag{4.1}$$

When two images are well aligned using a disparity map, then MI is maximized. The first two terms of mutual information measure the entropies in the un-occluded pixels of each individual image. The histograms can be used as an estimate of the probability density function of the image. If all bars in the histogram have the same length, then there is maximal uncertainty in the picture and we have a high entropy (this would be the case in a picture of white noise where we can hardly predict the intensity of the next pixel). If the bars are unevenly distributed then there is less uncertainty and we have a lower entropy. The third term is about the joint entropy of the two images. It is obtained overlaying the images based on the disparity map, and then for each intensity in the first image, creating a histogram of the matching pixels in the second image. This yields a two-dimensional histogram.

The matching cost for a given pixel $p$ and its disparity $D_p$, is the negative of the mutual information

$$-C(p, D_p) = h_{I_r}(p) + h_{I_i}(D_p) - h_{I_r,I_i}(p, D_p). \tag{4.2}$$

The first term $h_{I_r}$ is the entropy of the un-occluded pixel in the first image, the second term is the entropy of the corresponding un-occluded pixels in the second image. We want those terms to go up, because we want to identify as many un-occluded pixels as possible. The third term is the joint entropy. We want the joint entropy to go down, because we want a good alignment between the un-occluded pixels.

The next step is an aggregation of the cost. Pixel-wise cost calculation is inherently ambiguous. This is usually done using an energy function. The energy function is evaluated globally or over some kind of window. This method uses a global energy function, but only performs semi-global matching.

The energy function adds smoothness terms to the cost function, trying to resolve ambiguities by requiring smoothness in the disparity map

$$E(D) = \sum_p \left( C(p, D_p) + \sum_{q \in N_p} P_1 T(|D_p - D_q| = 1) + \sum_{q \in N_p} P_2 T(|D_p - D_q| > 1) \right) \tag{4.3}$$

The first term of the energy function is the pixel's cost function (4.2). The second term adds a penalty $P_1$ for all pixels $q$ in the neighborhood $N_p$, for which the disparity changes a little bit (that is, for one pixel). The third term

adds a larger adaptive penalty $P_2$, for all bigger disparity changes. Using a low penalty $P_1$ for small changes permits an adaptation to slanted or curved surfaces. The penalty $P_2$ for larger changes preserves discontinuities. The $T$ stands for an indicator function, which is one when the condition is true, and zero if the condition is wrong.

The energy function is global and has to be evaluated for all possible disparities of each pixel. The asymptotic time for evaluating the global energy function is the square of the width times the square of the height times the possible disparities. This is an NP-complete problem and thus takes a very long time to compute, this is why semi-global matching is used. The method uses multiple dynamic programming passes, which are 1-dimensional paths and then combines the results of the passes. A common approach is to combine the results of a horizontal path with a vertical path. But this leads to streaking effects. In SGM the results of sixteen directions are combined. This gives a good coverage of the area around the pixel. The results are almost as good as with global matching, but much faster.

The stereo matching algorithm is applied to both images in order to obtain two disparity maps, one referred to $I_r$ and the other to $I_i$.

At this point we perform some post-processing on the disparity maps. First we use a simple hole-filling technique as the one suggested in [58]. Afterwards, we use anisotropic diffusion [57] to smooth out the maps without compromising the edges. At last we run a left-right consistency check to gather precious occlusion (i.e., visibility) information. The disparity maps are used to obtain a set of dense correspondences that are then derectified using the inverse rectifying homographies.

Ultimately, parallax values are computed using Eq. (2.69) for each pixel: the dense set of correspondences $\mathbf{m}_r^k \leftrightarrow \mathbf{m}_i^k$ on the pair of images $(I_r, I_i)$ is known from the stereo matching step; the collineation is $H_\Pi^{ri} = T_2^{-1} T_1$ and epipole $\mathbf{e}_i$ is estimated from epipolar geometry. As a by-product of the rectification method [24], $H_\Pi$ approximates the homography of the plane at infinity.

### 4.3.2 Virtual camera orientation

In the upsampling application the orientation of the virtual camera interpolates between two actual ones, hence it can be specified by computing $D_{rv}$ using Equation (2.87) and selecting $t \in [0, 1]$. In the 3DS conversion application, on the contrary, the virtual camera position is alongside the actual one, thus, in general, outside its actual trajectory. A sketch of the two configurations is represented in Figure 4.1.

Assuming that the video has been shot with zero roll angle, i.e. the

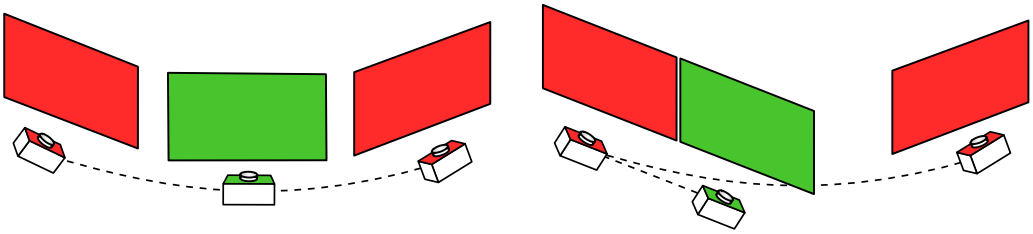

Figure 4.1: virtual camera (represented in green) positioning when interpolating between two views (on the left) and in the context of the 3DS conversion application (on the right).

image rows are parallel to the horizon, $H_{\Pi}^{rv}$ and $\mathbf{e}_v$ (that specify the position of the second camera) can be specified as follows. Since there is no rotation between the reference image and the virtual one (images are coplanar), from Section 2.4 we know that the infinite plane homography $H_{\Pi}^{rv}$ is the identity matrix and $\mathbf{e}_v = \begin{bmatrix} t & 0 & 0 \end{bmatrix}^T$ with $t \in \mathbb{R}^+$, since the virtual viewpoint is displaced horizontally. Thus the orientation of the virtual camera can be computed as

$$D_{rv} = \begin{bmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad t \in \mathbb{R}^+. \tag{4.4}$$

### 4.3.3 Forward mapping of parallax maps

Starting from a parallax map for the reference image $I_r$, we want to obtain a map referred to the virtual $I_v$ instead.

We begin with using the reference image $I_r$ as source image. The first step is to generate a set of corresponding points between $I_r$ and $I_v$ by instantiating Equation (2.68) as

$$\mathbf{m}_v \simeq H_{\Pi}^{rv}\mathbf{m}_r + \mathbf{e}_v\gamma \tag{4.5}$$

where $H_{\Pi}^{rv}$ is the infinite plane homography between the reference image $I_r$ and the virtual one $I_v$ and $\mathbf{e}_v$ is the epipole of $I_v$. Quantities $H_{\Pi}^{rv}$ and $e_v$ are specified through the parameter $t$ or in Equation (4.4), or its original formulation Equation (2.87), depending on the application, which encode the inter-ocular separation.

This process is a *forward mapping*: points in the original image are mapped forward, to the virtual image. Once we obtain the set of corresponding points among the reference image and the novel image we can compute parallax values with respect to our novel image using Equation (2.69).

As it is well known, forward mapping raises some intrinsic problems: i) small holes in the destination image due to the non-surjectivity of the map ii) the *folding* effect caused by the non-injectivity of the map. In the next two subsections we describe how we dealt with these problems.

#### Probabilistic splatting

In order to deal with the non-surjectivity issue, we developed a randomized technique that accounts for the quantization inherent to the forward mapping.

First, we generate noise in the form of random values drawn from the standard uniform distribution on the open interval $(-0.5, 0.5)$ (the amplitude of the interval is chosen to be equal to the maximum error introduced by the coordinates rounding). The noise is added to the non-integer coordinates $\mathbf{m}_v$ in Equation (4.5) which are then rounded to their closest integer value. The procedure is repeated for $n$ times (we choose $n = 100$) and all the perturbed parallax maps are merged into the final one by averaging them. This approach has two main advantages: i) as $n$ increases, the process will tend to approximate a proper linear interpolation between the neighbouring pixels, based on the distance from the integer values (i.e. the decimal parts of the coordinates) ii) this procedure fills holes in the map, since pixels with undefined value are likely to be filled with the value of the neighbouring valid ones.

#### Folding

Folding occurs when different source pixels are mapped to the same destination pixel. This phenomenon is due to the modification of the viewpoint, when two points that were visible in the original image fall along the same line of sight in the new image. As most approaches in literature, we deal with this problem by selecting the pixel with the greater disparity which, by definition will be occluding the one with a smaller disparity value.

### 4.3.4 Using multiple sources

If more source images are available the forward mapping procedure requires an additional step. For the forward mapping to produce a set of aligned parallax maps we must combine two uncalibrated motion matrices as described in the following. Let $I_r$ be the reference image and let $I_i$ be the current auxiliary image that we want to employ as additional source, first we must map the auxiliary view onto the reference view. This is done by computing

$D_{ir}$ from Equation (2.87) using $I_i$ as reference and using $t = 1$. The uncalibrated motion matrix that maps the auxiliary view to the virtual view is then obtained as

$$D_{iv} = D_{rv} D_{ir} \tag{4.6}$$

where $D_{rv}$ is the one originally computed for the reference image $I_r$. The uncalibrated motion composition scheme is shown in Figure 4.2

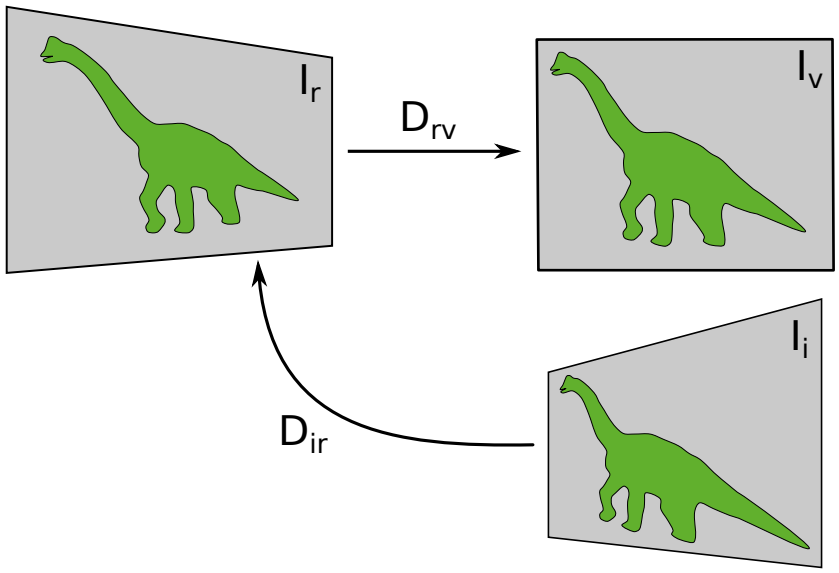


Figure 4.2: uncalibrated motion composition. The uncalibrated motion matrix from an auxiliary view to the virtual view can be obtained as $D_{iv} = D_{rv} D_{ir}$ .

### 4.3.5 Merging of parallax maps

At this point we have a collection of independent parallax maps for $I_v$ and the purpose of this step is to merge these maps into one, in order to reduce noise and fill holes.

Observe that even though these maps are commensurable, they differ by a global scale factor $s$. However, due to noise and outliers, the factor will not be unique for the entire image, thus we estimate it from the distribution of the pixel-wise ratios in a robust way using the Median Absolute Deviation and the x84 rejection rule [30].

Once the maps are brought to the same scale we merge them into a final one by keeping the highest parallax value in each pixel, where holes have conventionally assigned $-\infty$.

This fused parallax map is accompanied by a *source map* that records for each element of the final parallax map, which of the reference images it originates from. Together they define a mapping from $I_v$ to the reference images that will be exploited in the actual rendering of $I_v$, described in the next section.

### 4.3.6 Backward mapping of color

In the final stage of the method, the pixel grid of the virtual image is used as a reference to determine corresponding points in the reference images.

This process is a *backward mapping*, since points in the virtual image are mapped (backward), to points in the reference images to get a color assigned. Again, we rely on Eq. (2.68) and we rewrite it as

$$\mathbf{m}_r \simeq H_{\Pi}^{vr}\mathbf{m}_v + \mathbf{e}_r\gamma \tag{4.7}$$

Where $H_{\Pi}^{vr}$ and $\mathbf{e}_r$ are obtained according to Equation (2.87), but this time from matrix $D_{rv}^{-1}$.

The formula is applied pixels-wise using as the reference image the one specified in the source map. Bilinear interpolation is used to assign values to non-integer coordinates.

#### Occlusion filling

There can be points that are visible in the novel image, but for which a parallax value could not be computed, because they are not visible in the reference images or because of failure of the stereo matching. Such holes can only be filled heuristically. In our method, we build a binary map that estimates local foreground/background segmentation: on the disparity map, for each unassigned pixel we compute the variance of its neighborhood. A high variance indicates the presence of multiple depth layers, thus it is likely that an object in the foreground is occluding the background and the pixel is marked as background. Otherwise it is marked as foreground. This procedure is based on the idea presented in greater details in [58]. Once the binary map is built, we use it to fill holes on the virtual view, using the average colour of the neighbours that fall within the same class.

## 4.4 View-synthesis with motion-stereo

In this last section of the chapter we propose to combine the usage of our motion-stereo method, presented in Chapter 3, with our view-synthesis method described in the current chapter.

Parallax is a key ingredient for the quality of the rendered view, in fact if we look at the view-synthesis algorithm as a black-box, we can consider the input to be an image enclosed with its parallax map and the output to be an image depicted from a novel (virtual) view point. Unlinking the computation of the parallax map from the view-synthesis procedure allows us to choose the best way to derive it. Hence the idea of employing our motion-stereo algorithm for this task.

As we already pointed out, the benefits of adopting the data-fusion scheme fall within two main criteria: i) more pixels are *visible* when the information is coming from several stereo-pairs instead of a single one ii) the computation of parallax values is more accurate when temporal and spatial information are taken into account.

In the next chapter (Chapter 5), we will demonstrate how using the motion-stereo method for the computation of a parallax map in the context of the view-synthesis application actually improves the resulting virtual image.

# Chapter 5

# Experiments

## Contents



In this section we present the main results of our work. The chapter is organized as follows.

Section 5.1 presents the evaluation of our motion-stereo method alone, providing both quantitative and qualitative evaluation of the method.

In Section 5.2 we present the results for our view-synthesis method, evaluating the method as a stand-alone, and compare quantitatively the backward and the forward mapping and then provide some qualitative examples of 3DS conversion.

We then concatenate the two main parts of this thesis, i.e. the motion-stereo framework and the view-synthesis, and corroborate the idea that using a depth–proxy map refined with the motion-stereo pipeline for view-synthesis purposes produces sensible benefits to the final result.

At last, in Section 5.3, we present a case study that has been conducted during a collaboration with professor Anders Hast of the University of Uppsala, Sweden. In this work we apply our view-synthesis method to historical

aerial photographs taken during World War II. The section does not only present the results but also gives some insights on the specific application.

## 5.1 Motion-stereo

We run two set of experiments. In the first one we consider images from the Middlebury 2005 (6 sequences) and 2006 (21 sequences) datasets [64] with a ground truth in order to validate our method and quantify the benefit of the integration. In the second set we use more general sequences without ground truth.

It is important to stress that the method presented here focuses on the fusion of depth measurements, so the results reported should not be evaluated in absolute terms, but relatively to the *input* data, in this case disparity maps produced by NCC block–matching. More sophisticated stereo algorithms coupled with a global optimization would yield better depth maps, as those reported, e.g., in [80]. For these reasons a comparison with other stereo methods is pointless, since any of them could be plugged in our framework.

### 5.1.1 Middlebury datasets

In the Middlebury datasets the camera motion is constrained along the $X$ axis, so the integration takes place at the disparity level. The *error rate* is defined as the percentage of computed disparities values whose difference with the ground truth is $> 1$, as in [64]. Pixels marked as occluded in the ground truth have not been counted.

In all the experiments in this section, we used a square $3 \times 3$ window for the NCC stereo matching, and the size of the superpixels is set at 800 pixels.

First we performed a systematic evaluation of the confidence measures described in Section 3.5 with the Middlebury datasets. Results are reported in Table 5.1, where each entry contains the error rate of the disparity map produced by our method with a given confidence measure. Table 5.2 reports the results of a similar experiment in which the confidence measures do not include the LRC, i.e., $\varphi = \phi_*$. These figures compel us to make some observations:

- all confidence measures are equally suited to represent pixel's reliability, for in Table 5.1 all the entries are very close; however WNM obtains the lowest error rate, probably thanks to the fact that it considers distinctiveness of the match by looking at the second best match, the same recipe that proved so effective in SIFT matching (in fact, PKR, that uses a similar strategy, performs closely to WNM).

Table 5.1: error rates [%] of disparity maps obtained with different confidence measures (see Section 3.5 for explanation). Best and worst results are highlighted in boldface/green and lightface/red respectively. In the last two rows we report the mean and the trimmed mean (the best and the worst scores are not considered for the computation of the trimmed mean).

| Data | GT | MSM | CUR | PKR | MMN | MLM | AML | WMN | UNI |
|---|---|---|---|---|---|---|---|---|---|
| Art | 9.58 | 19.68 | 20.91 | **19.60** | 20.11 | 19.70 | 19.73 | 19.67 | 19.64 |
| Books | 8.78 | 23.14 | 24.17 | 22.90 | 23.51 | 22.97 | 23.01 | **22.83** | 23.03 |
| Dolls | 7.95 | **16.05** | 16.93 | 16.12 | 16.69 | 16.09 | 16.16 | 16.16 | 16.14 |
| Laundry | 13.37 | 28.71 | 31.73 | 28.84 | 30.23 | 28.96 | 29.08 | **28.37** | 29.02 |
| Moebius | 8.60 | 20.34 | 21.43 | 20.35 | 20.64 | 20.54 | 20.49 | **20.20** | 20.36 |
| Reindeer | 7.33 | 14.24 | 15.60 | 14.28 | 15.01 | 14.23 | **14.15** | 14.30 | 14.27 |
| Aloe | 5.86 | 8.51 | 9.32 | 8.57 | **8.41** | 8.56 | 8.56 | 8.56 | 8.51 |
| Baby1 | 3.83 | 12.01 | 13.38 | 11.34 | 11.92 | 11.82 | 11.80 | **11.31** | 11.57 |
| Baby2 | 3.83 | 17.13 | 19.13 | 16.82 | 17.41 | 16.89 | 16.91 | **16.65** | 16.90 |
| Baby3 | 6.88 | 13.30 | 14.78 | 13.24 | 13.75 | 13.33 | 13.35 | **13.21** | 13.31 |
| Bowling1 | 31.97 | 73.29 | 32.73 | 30.49 | 34.21 | 61.15 | 50.49 | **30.40** | 30.44 |
| Bowling2 | 6.75 | 16.91 | 18.18 | **16.84** | 19.41 | 17.15 | 16.91 | 16.91 | 16.80 |
| Cloth1 | 2.14 | 2.67 | 2.72 | 2.68 | **2.51** | 2.69 | 2.69 | 2.68 | 2.68 |
| Cloth2 | 5.04 | 8.06 | 8.58 | 8.06 | 8.22 | 8.10 | 8.11 | **8.04** | 8.10 |
| Cloth3 | 2.75 | 4.85 | 5.06 | 4.85 | **4.54** | 4.90 | 4.89 | 4.85 | 4.94 |
| Cloth4 | 5.20 | 8.19 | 8.77 | 8.11 | **7.83** | 8.12 | 8.16 | 8.10 | 8.12 |
| Flowerpots | 6.84 | 24.74 | 24.53 | **23.02** | 25.69 | 23.54 | 23.38 | 23.07 | 23.09 |
| Lampshade1 | 48.87 | 63.67 | 35.38 | **32.72** | 34.88 | 71.65 | 54.97 | 33.32 | 33.02 |
| Lampshade2 | 74.10 | 55.84 | 34.69 | 34.03 | 35.02 | 70.98 | 80.52 | **32.51** | 34.87 |
| Midd1 | 24.25 | 96.34 | 50.65 | **50.20** | 51.07 | 97.19 | 96.13 | 50.52 | 50.22 |
| Midd2 | 20.27 | 88.27 | 49.60 | **48.80** | 49.46 | 84.41 | 87.62 | 48.98 | 48.48 |
| Monopoly | 20.27 | 47.97 | 42.01 | **39.89** | 42.12 | 42.57 | 40.77 | 39.99 | 39.78 |
| Plastic | 91.03 | 78.18 | **67.13** | 86.55 | 73.56 | 79.49 | 84.42 | 80.93 | 90.23 |
| Rocks1 | 3.35 | 5.92 | 6.18 | 5.90 | **5.50** | 5.95 | 5.96 | 5.90 | 5.95 |
| Rocks2 | 3.48 | 5.19 | 5.41 | 5.27 | **5.13** | 5.23 | 5.25 | 5.27 | 5.28 |
| Woods1 | 6.44 | 10.47 | 11.11 | 10.39 | 11.24 | 10.40 | 10.40 | **10.38** | 10.43 |
| Woods2 | 4.33 | 17.20 | 19.64 | 17.11 | 19.35 | 17.40 | 17.30 | **17.10** | 17.28 |
| Mean | 16.04 | 28.92 | 22.58 | 22.11 | 22.50 | 29.04 | 28.56 | **21.86** | 22.31 |
| T-Mean | 13.60 | 27.27 | 21.60 | 20.31 | 21.25 | 27.37 | 26.90 | **20.26** | 20.38 |

Table 5.2: error rates [%] of disparity maps obtained with different confidence measures (see Section 3.5 for explanation), **without the LRC check**. Best and worst results are highlighted in boldface/green and lightface/red respectively. In the last two rows we report the mean and the trimmed mean (the best and the worst scores are not considered for the computation of the trimmed mean).

| Data | GT | MSM | CUR | PKR | MMN | MLM | AML | WMN | UNI |
|---|---|---|---|---|---|---|---|---|---|
| Art | 10.61 | 23.00 | 25.69 | 23.19 | **22.74** | 23.08 | 23.10 | 23.27 | 23.14 |
| Books | 11.58 | 42.90 | 28.25 | 25.29 | **24.89** | 27.42 | 26.34 | 25.28 | 25.53 |
| Dolls | 6.29 | 18.48 | 20.17 | 18.62 | **18.19** | 18.51 | 18.59 | 18.61 | 18.61 |
| Laundry | 57.15 | 44.91 | 46.94 | 40.46 | **32.39** | 44.28 | 41.33 | 39.04 | 41.35 |
| Moebius | 6.06 | 28.42 | 24.32 | 22.26 | **21.90** | 22.70 | 22.28 | 22.11 | 22.20 |
| Reindeer | 6.70 | 18.58 | 18.95 | **16.86** | 17.24 | 22.00 | 17.91 | 16.87 | 16.85 |
| Aloe | 5.36 | 10.69 | 12.21 | 10.75 | **10.28** | 10.81 | 10.82 | 10.74 | 10.70 |
| Baby1 | 3.17 | 35.59 | 17.27 | 14.73 | **13.74** | 21.58 | 17.66 | 14.59 | 14.82 |
| Baby2 | 3.21 | 38.04 | 23.59 | 21.36 | **19.60** | 29.16 | 27.75 | 21.09 | 21.92 |
| Baby3 | 3.56 | 18.91 | 18.63 | 15.78 | **15.20** | 16.43 | 16.08 | 15.74 | 15.86 |
| Bowling1 | 75.81 | 64.51 | **34.50** | 73.31 | 35.22 | 68.45 | 68.84 | 74.08 | 73.70 |
| Bowling2 | 7.97 | 25.04 | 23.17 | **20.20** | 21.58 | 76.65 | 50.94 | 20.33 | 20.27 |
| Cloth1 | 1.23 | 5.45 | 6.29 | 5.45 | **5.04** | 5.53 | 5.56 | 5.45 | 5.45 |
| Cloth2 | 3.41 | 10.81 | 12.49 | 10.84 | **10.22** | 10.86 | 10.89 | 10.84 | 10.88 |
| Cloth3 | 2.05 | 7.43 | 8.12 | 7.55 | **6.79** | 7.52 | 7.52 | 7.55 | 7.55 |
| Cloth4 | 3.14 | 9.78 | 11.24 | 10.00 | **9.26** | 9.88 | 9.82 | 9.99 | 9.93 |
| Flowerpots | 8.84 | 78.11 | 27.45 | 27.30 | **26.88** | 65.64 | 57.40 | 27.18 | 28.38 |
| Lampshade1 | 55.69 | 88.83 | 38.70 | 90.15 | **37.25** | 92.40 | 91.48 | 88.54 | 90.35 |
| Lampshade2 | 48.85 | 86.98 | **37.34** | 88.87 | 41.91 | 90.11 | 91.14 | 85.79 | 89.77 |
| Midd1 | 73.70 | 97.99 | 52.50 | 98.61 | **52.01** | 97.90 | 97.94 | 77.20 | 98.74 |
| Midd2 | 40.83 | 97.84 | **51.84** | 98.66 | 52.62 | 97.97 | 98.05 | 98.86 | 98.76 |
| Monopoly | 88.44 | 79.88 | 44.61 | 59.33 | **42.78** | 70.68 | 74.37 | 50.87 | 88.57 |
| Plastic | 65.28 | 96.20 | 92.10 | 93.47 | **87.72** | 94.31 | 93.76 | 92.89 | 94.46 |
| Rocks1 | 2.43 | 7.96 | 9.26 | 8.07 | **7.28** | 8.00 | 8.04 | 8.07 | 8.08 |
| Rocks2 | 2.07 | 7.38 | 8.61 | 7.60 | **6.94** | 7.55 | 7.58 | 7.60 | 7.60 |
| Woods1 | 4.29 | **12.52** | 14.15 | 12.67 | 12.69 | 17.81 | 14.23 | 12.64 | 12.71 |
| Woods2 | 2.79 | 62.23 | 22.74 | 19.48 | 21.43 | 77.68 | 71.80 | **19.45** | 19.71 |
| Mean | 22.24 | 41.42 | 27.08 | 34.85 | **24.96** | 42.03 | 40.05 | 33.51 | 36.14 |
| T-Mean | 20.43 | 40.60 | 25.31 | 33.47 | **23.24** | 41.26 | 39.10 | 32.01 | 34.87 |

- the UNI metric has surprisingly good performances, confirming the robustness of the integration framework; in other words, the data fusion works so well that the confidence becomes nearly irrelevant;

- if LRC is switched off, MMN is the best performer, although by a narrow margin; this suggests that MMN could be a proxy for occlusions detection if LRC cannot be performed;

- the comparison of the two tables indicates that the most important contribution to confidence is the the binary response of the LRC check.

Since WMN obtains the lowest error rate, we chose it as the default confidence measure for the rest of our experiments, although other choices would likely produce similar results.

Then, we assess the benefits of the spatio–temporal integration. Following [37], we consider two touchstones against which to compare the error rate obtained with our method (**Kalman ST**): the error rate of the "optimal selection" map (**Optimal**) obtained as if an oracle could somehow select the disparity value closest to the ground-truth among all the input estimates for each pixel, and the minimum error rate (**Best Map**) obtained selecting the best map among all the input disparity maps. Observe that the former represents the theoretical optimum that one can achieve with the given input disparity maps using the temporal dimension, while the latter is an indicator of whether the data-fusion is beneficial with respect to a simple two-views stereo. We also considered other integration strategies: the maximum confidence selection (**Max conf**), which consists in selecting, for each pixel, the disparity that achieves the maximum confidence $\varphi$, the temporal fusion (**Kalman T**), that consists in applying only the temporal Kalman filter, without spatial relaxation, as in [1].

Results with the Middlebury datasets are reported in Table 5.3, where it can be appreciated that **Kalman ST** (in boldface) achieves the lower scores, when compared to the other two strategies; in particular spatio–temporal integration always improves the pure temporal Kalman filter. Moreover our method always exceeds the best map and, in some cases, it also exceeds the optimal one, due to the spatial relaxation.

Figure 5.3 reports, for the same experiments, how the error rate decreases as more measures are integrated. We also included our implementation of [52] (henceforth **MKS**), for comparison with another method from the literature. Observe that **MKS**, despite the integration and spatial relaxation steps, only slightly improves the results obtained by the regular stereo matching algorithm. This confirms the idea that the warping of the disparity map


Figure 5.1: From top to bottom: the reference frame, the **Best Map**, the result of **MKS**, the result of **Kalman T** (temporal–only) and the result of **Kalman ST**. Images are automatically scaled in the range [0,255], hence the gray levels change from row to row. Full resolution images can be seen on line [77].

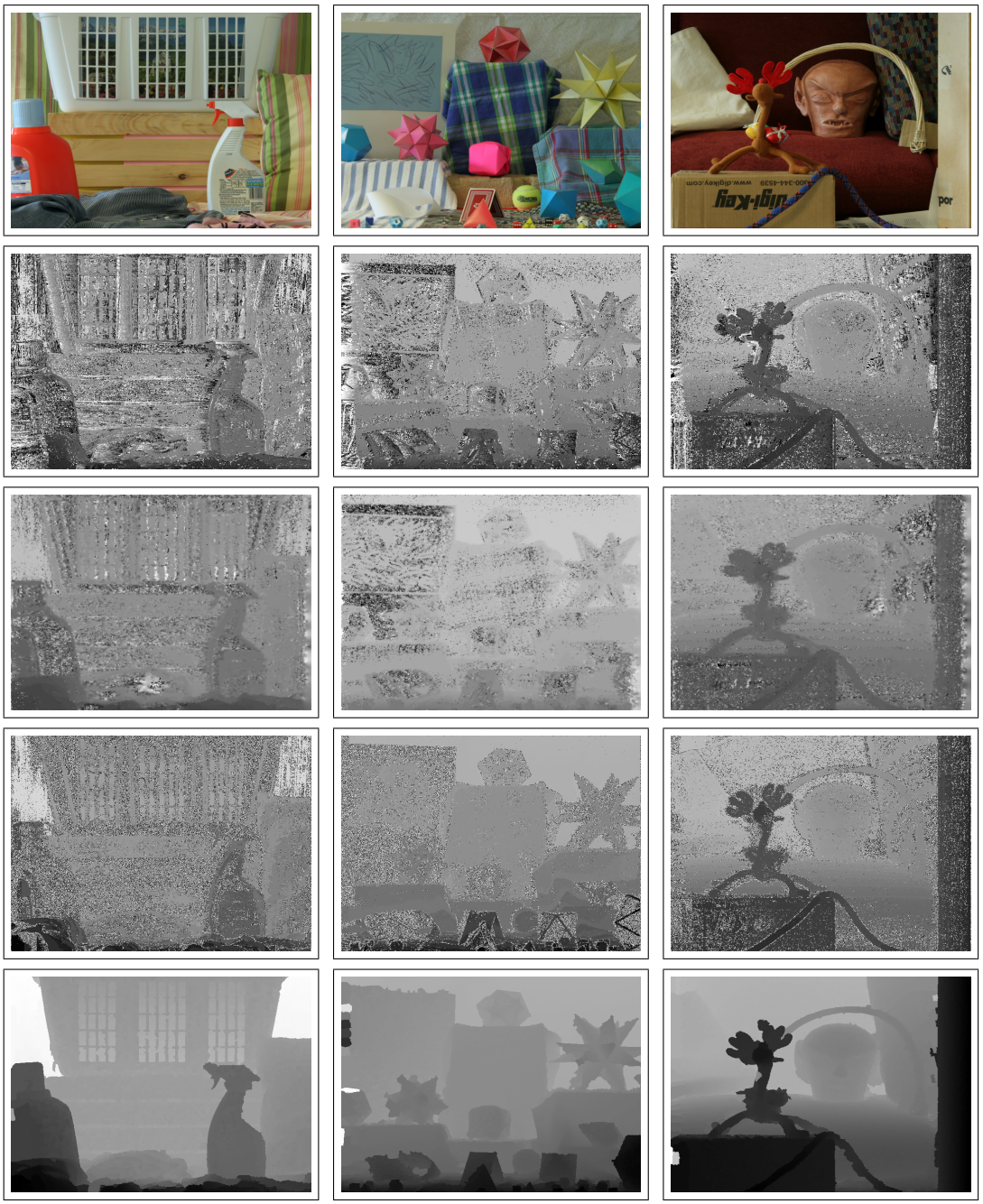

Figure 5.2: From top to bottom: the reference frame, the **Best Map**, the result of **MKS**, the result of **Kalman T** (temporal–only) and the result of **Kalman ST**. Images are automatically scaled in the range [0,255], hence the gray levels change from row to row. Full resolution images can be seen on line [77].

Table 5.3: error rates [%] of disparity maps obtained with different fusion strategies (see text for explanation) and WMN as the confidence measure.

| Data | Best map | Max conf | Average | Kalman T | Kalman ST | Optimal |
|---|---|---|---|---|---|---|
| Art | 49.76 | 53.12 | 48.15 | 35.13 | **19.67** | 21.54 |
| Books | 55.89 | 68.37 | 59.04 | 48.57 | **22.83** | 29.32 |
| Dolls | 42.01 | 54.71 | 38.52 | 29.01 | **16.16** | 15.95 |
| Laundry | 75.67 | 69.22 | 69.27 | 58.16 | **28.37** | 44.23 |
| Moebius | 45.73 | 63.53 | 45.98 | 35.70 | **20.20** | 22.21 |
| Reindeer | 45.11 | 57.24 | 49.95 | 32.49 | **14.30** | 17.13 |
| Aloe | 24.55 | 46.12 | 20.00 | 13.95 | **8.56** | 7.17 |
| Baby1 | 49.07 | 55.80 | 48.89 | 41.59 | **11.31** | 20.62 |
| Baby2 | 53.17 | 61.64 | 53.63 | 46.16 | **16.65** | 25.62 |
| Baby3 | 51.65 | 68.29 | 53.83 | 44.64 | **13.21** | 23.56 |
| Bowling1 | 91.97 | 84.63 | 83.59 | 86.52 | **30.40** | 82.53 |
| Bowling2 | 54.64 | 63.81 | 51.03 | 42.29 | **16.91** | 27.18 |
| Cloth1 | 14.61 | 44.27 | 10.24 | 5.41 | **2.68** | 2.46 |
| Cloth2 | 31.35 | 52.14 | 26.82 | 18.53 | **8.04** | 10.04 |
| Cloth3 | 20.27 | 48.22 | 16.27 | 9.11 | **4.85** | 4.83 |
| Cloth4 | 26.03 | 45.68 | 21.57 | 13.54 | **8.10** | 7.46 |
| Flowerpots | 65.25 | 71.41 | 63.69 | 55.56 | **23.07** | 34.20 |
| Lampshade1 | 98.18 | 96.58 | 90.19 | 98.63 | **33.32** | 91.83 |
| Lampshade2 | 98.13 | 98.36 | 96.05 | 98.01 | **32.51** | 91.72 |
| Midd1 | 98.33 | 98.49 | 76.29 | 99.04 | **50.52** | 93.18 |
| Midd2 | 98.24 | 99.09 | 98.08 | 98.72 | **48.98** | 93.03 |
| Monopoly | 98.11 | 97.30 | 92.42 | 98.15 | **39.99** | 90.96 |
| Plastic | 96.48 | 94.30 | 90.30 | 98.21 | **80.93** | 83.81 |
| Rocks1 | 22.36 | 48.31 | 19.33 | 10.83 | **5.90** | 5.72 |
| Rocks2 | 19.71 | 47.52 | 15.43 | 8.87 | **5.27** | 5.27 |
| Woods1 | 43.97 | 53.48 | 44.05 | 29.62 | **10.38** | 13.27 |
| Woods2 | 52.99 | 66.27 | 54.22 | 47.02 | **17.10** | 27.69 |
| Mean | 56.42 | 66.96 | 53.22 | 48.28 | **21.86** | 36.76 |
| T-Mean | 56.41 | 66.58 | 53.14 | 47.96 | **20.26** | 35.88 |

from frame to frame severely limits the benefits of the integration mechanism. Figures 5.1 and 5.2 show qualitative results for the above sequences.

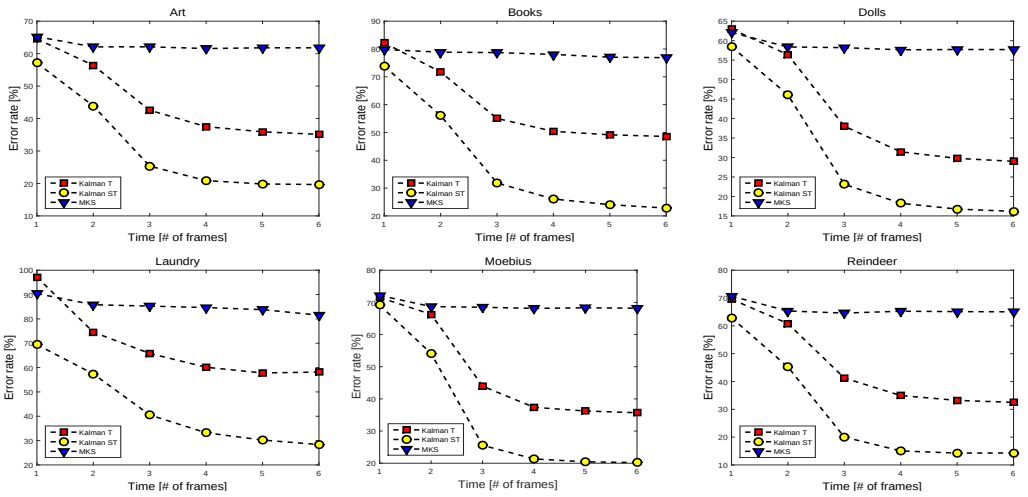


Figure 5.3: Each graph shows the error rate decreasing as more measures are integrated in the estimation for the three approaches, the **Kalman T** (temporal–only), the **Kalman ST** (spatial support) and the **MKS** (our implementation of [52]).

Finally, please note that what we refer to as **Kalman T** is the same implementation of the **Kalman ST** with the spatial step switched-off, which is slightly different from the original one described in [1] because of the validation gate and other tweakings and also because we are using a NCC based stereo algorithm instead of the Census transform. Consequently, figures reported in Table 5.3 are different (and better) from those reported in [1].

### 5.1.2 Casual video sequences

In the second set of experiments we test the method on the "Flower", "Road", "Lawn", from [80]. These are casual, uncalibrated sequences, hence we used parallax as a depth-proxy and went trough all the stages required to compute it. Since ground truth is not available, the evaluation will be only qualitative. Results, in Figure 5.4, show a significant improvement on the strategy without spatial support, and are more consistent with the scene content, especially on occluded or badly measured pixels. **MKS** could not be evaluated on these images, as it is restricted to pure lateral motion.

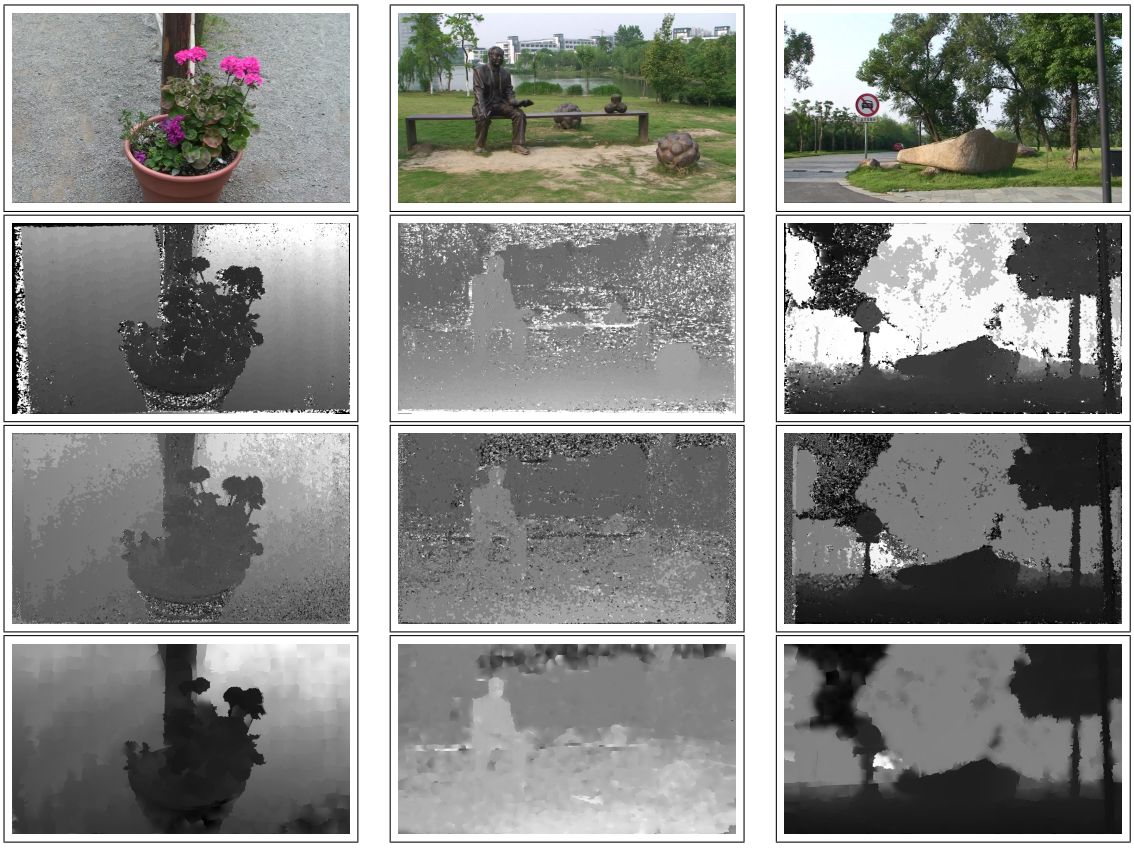

Figure 5.4: from top to bottom: the reference frame, the qualitative **Best Map** (manually selected), the result of **Kalman-T** (temporal–only) and the result of **Kalman-ST**. Images are automatically scaled in the range [0,255], hence the gray levels change from row to row. Full resolution images can be seen at [77].

## 5.2 View-synthesis

We report two sets of experiments that validate our approach. The first set focuses on the forward vs backward mapping issue, and shows quantitative results by comparing our rendered images against ground truth images. The second set of experiments shows the visual results of the rendering of a virtual frames in a 3DS conversion scenario, where the position of the virtual camera is set alongside the actual one.

First we validate the choice of the backward mapping (BWM) approach – described in Section 4.3.3 and previous ones – against the forward mapping (FWM). FWM, used e.g. in [23], is the fusion of two virtual images obtained from the reference images. Both approaches are evaluated before the hole filling post-processing step. In order to factor out the inaccuracies of the stereo matching we used the Middlebury 2006 dataset ([36]) which provides ground truth disparity maps. Each sequence of the dataset includes seven frames, we used the second and the sixth frame as the reference pair to synthesize the middle frame, which corresponds to the fourth frame. We then compared the ground truth image with the virtual one and obtained the results reported in Table 5.4. As figures of merit we considered the structure similarity index (SSIM) [74], the signal-to-noise ration (SNR) and the absolute error rate (ABS) where pixels that differ from the true value for more than 1 pixel and unassigned ones are counted as errors. The result is that BWM consistently outperforms FWM showing its better ability to recreate the virtual image.

At last, we report some qualitative results of the 3DS conversion in Figure 5.5. The images are taken from [79] and from [36]. Despite a few artifacts – mainly due to failures of the stereo matching – the results are convincing and visually plausible.

### 5.2.1 View-synthesis with motion-stereo

In this section, as mentioned in Section 4.4, we want to investigate the benefits of using a parallax map that is refined with the motion-stereo pipeline previously presented within the view-synthesis application.

The following experiments are aimed to evaluate this idea. We substitute the parallax map computed from a simple pair of images with a refined map computed with our motion-stereo method, varying the number of frames that are used to refine the map. Such maps are then provided to the view-synthesis method.

The experiment is structured as follows. We used the second frame as the reference view to next frame, which corresponds to the third frame. We

Table 5.4: comparison of performances of our BWM approach against the FWM employed in [23]. SSIM: and SNR the higher the better. ABS: the lower the better. See text for further explanations.

| | BWM | | | FWM | | |
|---|---|---|---|---|---|---|
| Sequence | SSIM | SNR | ABS | SSIM | SNR | ABS |
| Aloe | 0.77 | -33.59 | 33.74 | 0.72 | -34.15 | 37.25 |
| Baby1 | 0.89 | -30.48 | 33.83 | 0.87 | -30.80 | 34.79 |
| Baby2 | 0.90 | -26.91 | 31.67 | 0.88 | -27.04 | 33.75 |
| Baby3 | 0.88 | -25.80 | 21.73 | 0.86 | -26.45 | 23.35 |
| Bowling1 | 0.88 | -31.80 | 46.56 | 0.87 | -31.90 | 46.99 |
| Bowling2 | 0.87 | -30.81 | 37.62 | 0.85 | -31.23 | 39.04 |
| Cloth1 | 0.96 | -20.94 | 39.59 | 0.93 | -19.77 | 40.74 |
| Cloth2 | 0.91 | -27.83 | 42.95 | 0.86 | -28.75 | 44.66 |
| Cloth3 | 0.91 | -26.37 | 37.21 | 0.87 | -27.20 | 39.08 |
| Cloth4 | 0.88 | -31.44 | 35.91 | 0.82 | -32.17 | 39.33 |
| Flowerpots | 0.89 | -28.56 | 28.45 | 0.87 | -28.73 | 30.23 |
| Lampshade1 | 0.83 | -33.99 | 24.36 | 0.82 | -34.41 | 25.04 |
| Lampshade2 | 0.84 | -33.92 | 22.90 | 0.83 | -34.36 | 23.33 |
| Midd1 | 0.91 | -29.85 | 26.57 | 0.89 | -30.10 | 28.56 |
| Midd2 | 0.90 | -30.01 | 44.92 | 0.89 | -30.24 | 46.03 |
| Monopoly | 0.86 | -34.04 | 90.35 | 0.83 | -34.33 | 89.11 |
| Plastic | 0.94 | -25.85 | 29.51 | 0.92 | -26.54 | 29.49 |
| Rocks1 | 0.92 | -21.67 | 27.82 | 0.88 | -22.76 | 31.11 |
| Rocks2 | 0.93 | -21.22 | 33.51 | 0.89 | -22.32 | 35.77 |
| Woods1 | 0.92 | -27.89 | 40.22 | 0.90 | -27.76 | 42.10 |
| Woods2 | 0.94 | -25.05 | 28.45 | 0.92 | -25.05 | 29.70 |

Figure 5.5: examples of the 3DS output. From left to right: reference image (left eye), virtual image (right eye), red-cyan anaglyph.

then compared the ground truth image with the virtual one and obtained the results reported in Figure 5.6. As a figure of merit we considered the absolute error rate (ABS) where pixels that differ from the true value for more than 1 pixel and unassigned ones are counted as errors.

As we can see from the graphs, the number of wrong pixels decreases as more images are integrated into the motion-stereo step. One frame corresponds to the case where we are using map computed from a single pair of images. We have the largest improvement when the first *new frame* after the *simple stereo pair* is integrated and the error rate keeps decreasing as more frames are added to the parallax map estimation, although in some cases the error stabilizes to a constant values, so it seems like there is a saturation of the quality of the synthetic view, with respect to the parallax map. Please note that we had a similar behavior during the evaluation of the parallax map itself in Figure 5.3.

## 5.3 Case study: historical aerial photography

This last Section reports on a very interesting study that we had the opportunity to do in collaboration with Professor Anders Hast of the University of Uppsala, Sweden. Starting from a set of 2D aerial photographs the task is the 3DS visualization. Apart from being a really fascinating project, it served as a valuable bench test for our view-synthesis method.

### 5.3.1 Motivation

Since the birth of modern aviation, aerial photos have been a rich source for understanding the historical development and geospatial changes. They offer a unique way to go back in time, exploring things as they were and therefore they have been used for aerial archaeology [11, 12, 60, 76]. An archive with several millions of such historical aerial photos is maintained by the Aerofototeca Nazionale (AFN) of the Italian Ministry of Cultural Heritage, in Rome. This archive portrays the Italian territory since the end of the nineteenth century, before its transformation due to the post-war reconstruction, the economic boom and changes by natural disasters such as earthquakes and floods. During World War II stereoscopic images played an important role in the success of missions. Pilots from the photographic reconnaissance units took several consecutive photos over Europe that covered a long line of each flight route. A sample sequence of exposures is shown in Figure 5.7. By meticulously photographing the ground it was possible for the photographic interpreters to visualize the area covered in 3D stereo (3DS), which can only

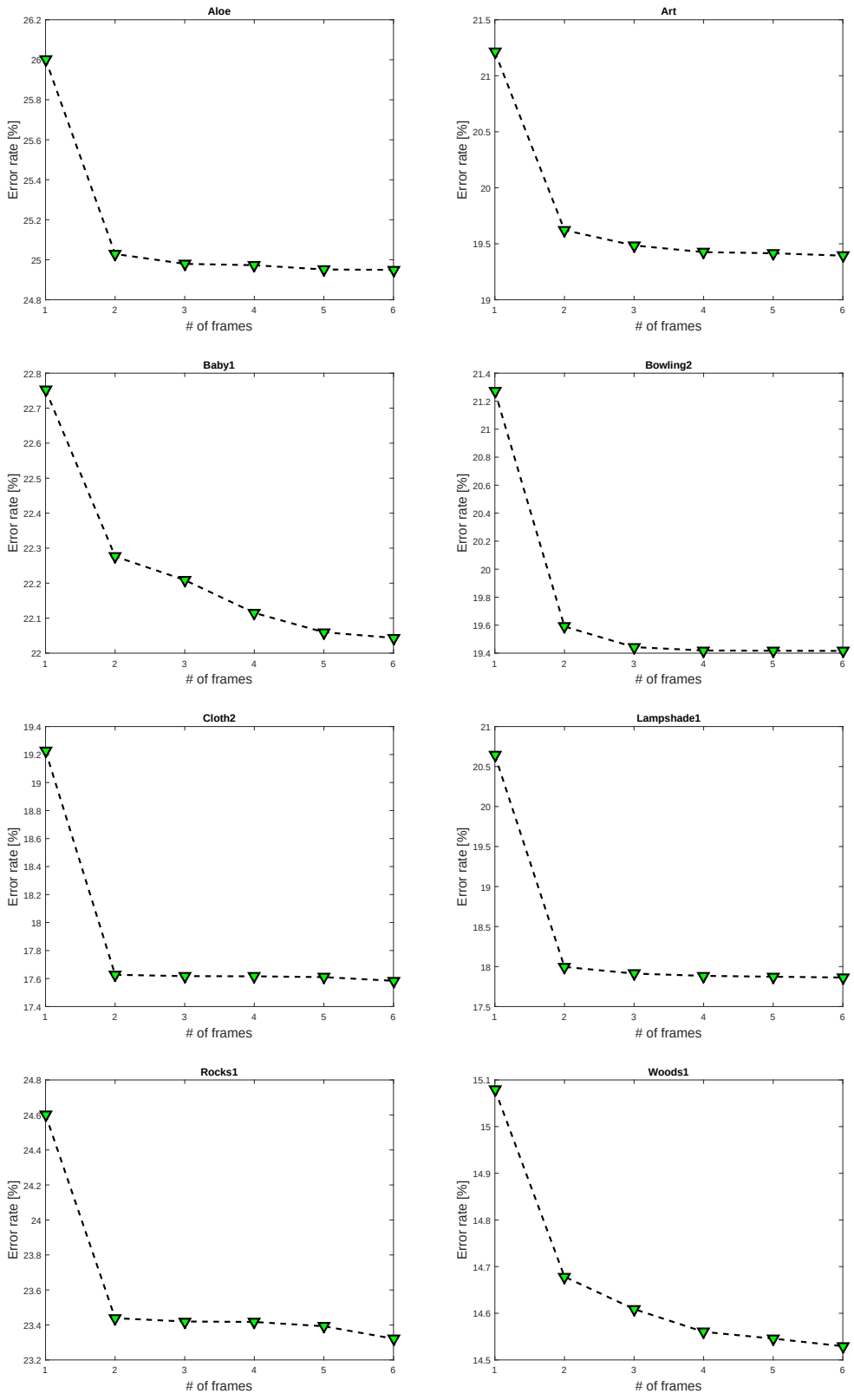


Figure 5.6: error rate of the view-synthesis with respect to the number of frames integrated into motion-stereo step.

Figure 5.7: an example of a sequence of exposures taken by the Royal Air Force (RAF) during WWII (from the AFN archive). The photos are placed in their relative positions by panoramic stitching, using [13]. MiBAC-ICCD, Aerofototeca Nazionale, fondo RAF ©.

be obtained if there is a substantial overlap between pairs of images.

Viewing aerial photos in 3DS is important, for the depth cue gives a much better understanding of the scene since the perceived depth information adds clues that are not available in a single exposure. This was important for the photographic interpreters, as it made possible to distinguish among objects such as houses, trees and the ground and especially estimating their height. Today, such stereo images can be a valuable tool for digital heritage research (e.g. extensions to projects such as [7] to also include stereo images).

In [32, 33] 3DS visualization is applied to historical aerial photographs with compelling results. These works concentrated on aspects such as pairs selection, geometric and illumination corrections in order to produce a 3DS display *with the available exposures*. This implies that the images must be viewed in such a way that one eye sees one exposure and the other eye sees the next one in line. In other words, the stereoscopic baseline (i.e., the line joining the two eyes) is parallel to the line of flight. This solution provides valid results for a static view, but falls short when trying to visualize the whole flight as a 3DS video. As a matter of fact, when moving along the line of flight it will be necessary, at a certain point, to switch from one stereo pair to the next in the sequence. However, there exists no such natural continuation between the stereo pairs, leading to a sudden switch between pairs that is perceptually disrupting.

The solution that we propose is to generate a 3DS video from the monocular sequence using our view-synthesis technique. This entails that the baseline is *orthogonal* to the line of flight: one eye sees the existing stream of exposures, and the other one sees a synthetic stream of images, corresponding to a virtual eye displaced from the other one orthogonally with respect to the flight trajectory.

The view-synthesis approach, as opposed to using actual images for stereoscopic display as in [32], is more flexible, for the virtual viewpoint can be placed anywhere and illumination (or color) is consistent by construction. The obvious drawback of this method is that the visual quality of the actual exposure is unmatched by any synthetic image, so, disregarding the temporal jittering, the stereoscopic display of [32] is more compelling. However our method provides a viable solution when the desired output is a *3DS video*.

### 5.3.2 Method

The aim of this procedure is the 3DS conversion of monocular aerial images, i.e. the generation of corresponding stereo images for a set of input exposures. Figure 5.8 shows a schematic representation of the algorithm: a triple of images is used to compute a disparity map referred to the central image which in turn allows to synthesize a stereoscopic pair. The procedure is repeated for every overlapping triplet of images to create a 3DS pair from every frame.

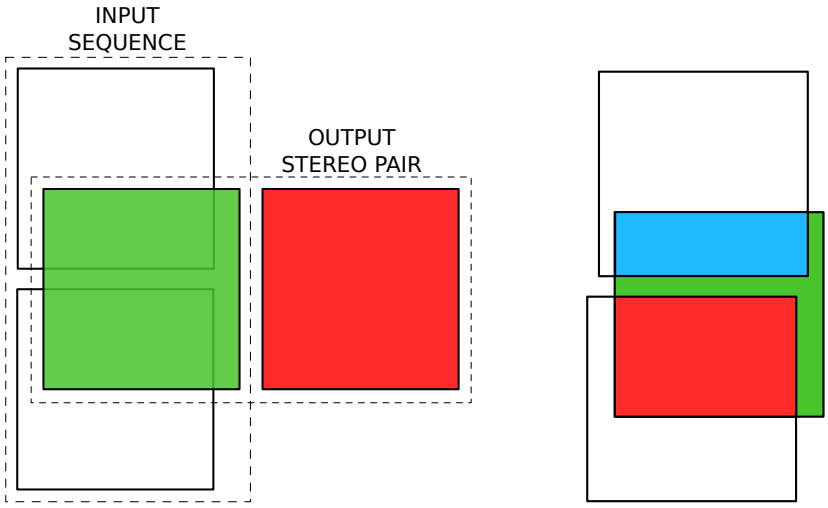


Figure 5.8: on the left, a pictorial representation of the 3DS conversion for one image of the input sequence (green). The two neighbouring images are used to compute a disparity map referred to the central image which in turn allows to synthesize a stereoscopic "right" image (red). On the right, a representation of the parallax integration. $I_r$ is the image in the middle. The blue portion of the corresponding parallax map is computed when $I_i$ is the image at the top, the red portion is computed when $I_i$ is the image at the bottom, the green portion is computed with the hole-filling heuristic.

For each input image two main steps are performed:

- computation of a parallax map using stereo matching

- rendering the novel stereo image through uncalibrated view-synthesis

It goes without saying that parallax can be computed only where correspondences can be established, hence in the area where the two images overlaps. Let $I_r$ be the *reference image*, i.e., the one for which we want to compute the parallax map. Let us consider a set of $n$ auxiliary images $I_i$, where $i = 1 \dots n$, that depict a relevant portion of the same scene as $I_r$. Then we compute a parallax map for $I_r$ by executing the steps described above for each pair $(I_r, I_i)$. Each $I_i$ will yield parallax values for a different portion of $I_r$.

In the sequences considered in this paper one image typically overlaps only with other two images, and these two does not overlap with each other, as represented in Figure 5.8. Therefore, there is usually a central stripe in $I_r$ where parallax cannot be computed. This is why we introduced a hole-filling heuristic similar to the one presented in [58]. First we perform a local foreground/background segmentation: in the parallax map, for each unassigned pixel we compute the variance of its neighborhood. A high variance indicates the presence of multiple depth layers, thus it is likely that the unassigned pixel is occluded by a foreground one in the conjugate image. Therefore, it is filled using the average of other background pixels in its neighborhood. Otherwise, a low variance indicate a single depth layer and the pixel is filled accordingly using the average value of its neighborhood. In principle and as we already argued in Section 4.4, when sequences present a higher degree of overlap than those considered in this paper, it could be necessary to employ an integration procedure that allows to merge parallax values coming from different pair of images, as in our motion-stereo pipeline described in Chapter 3. This possibility is explored using more suitable datasets in Section 5.2.1.

At last, once a parallax map is computed, we generate the second element of the 3D stereo pair using the uncalibrated view-synthesis approach described in Chapter 4.

### 5.3.3 Results

We report some experimental results obtained with our method applied to images taken from the AFN dataset that depict an aerial view of Pisa, Italy, during World War II (February 1944).

In Figure 5.9 we show the output obtained with the scheme described above and applied to an image triplet. We can observe that the rendered

image is geometrically correct, the illumination is the same as the reference one and it is visually plausible thanks to a very limited presence of artifacts. In Figure 5.10 we report some details of the same output, in order to better appreciate the good quality of the synthetic image. Concluding, in this section we evaluated our view-synthesis method in the context of 3DS conversion of historical aerial photographs. By rendering virtual images in an unconstrained fashion with respect to the flight trajectory, the proposed solution overcomes some potential limitations of previous works on the same dataset that used actual exposures for 3DS. The results are promising, however some issues are still to be addressed, such as stabilizing in time the virtual camera position, and upsampling the sequence in order to be able to play the 3DS video at a reasonable speed.

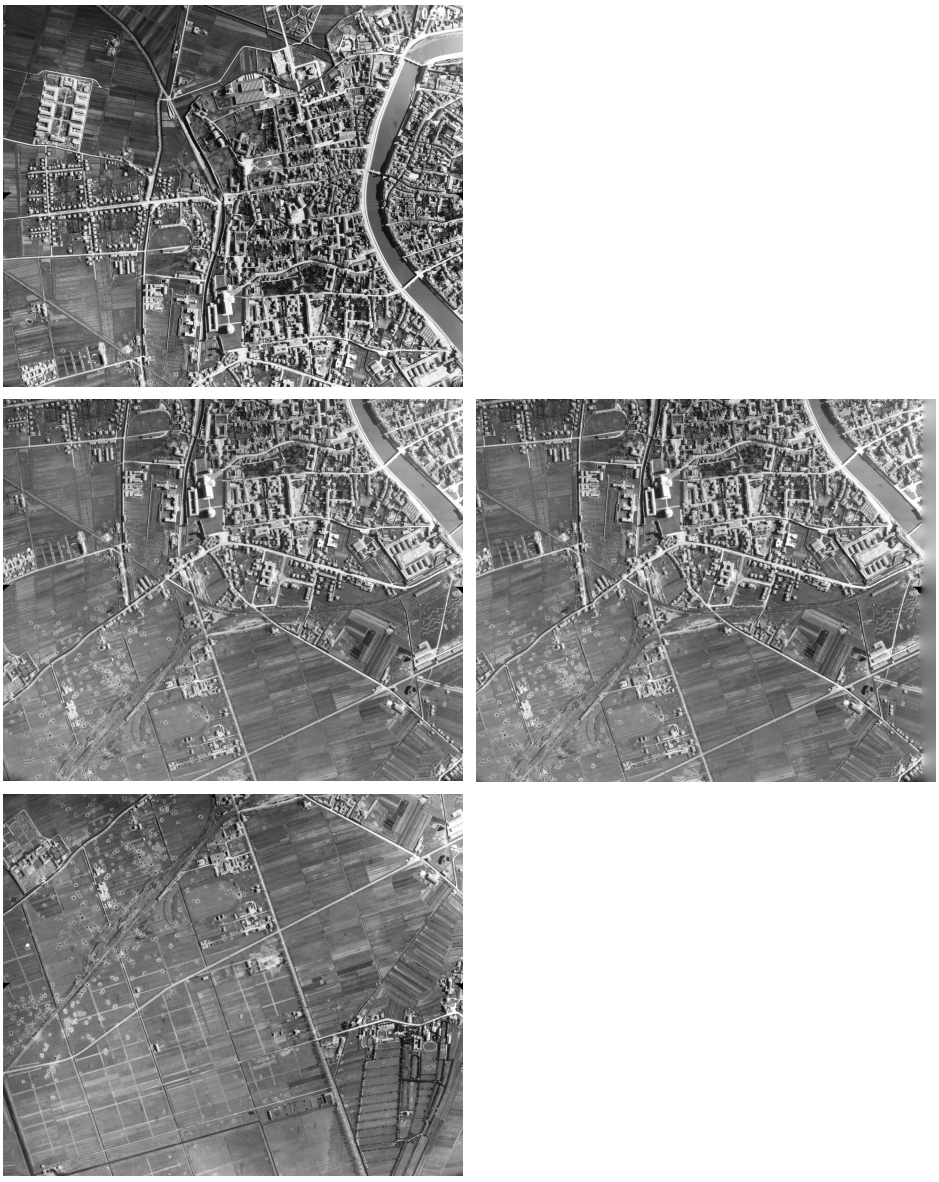

Figure 5.9: on the left the three input images ($4200 \times 4900$) , reference image is the one in the middle; on the right the rendered (synthetic) stereo image. MiBAC-ICCD, Aerofototeca Nazionale, fondo RAF ©

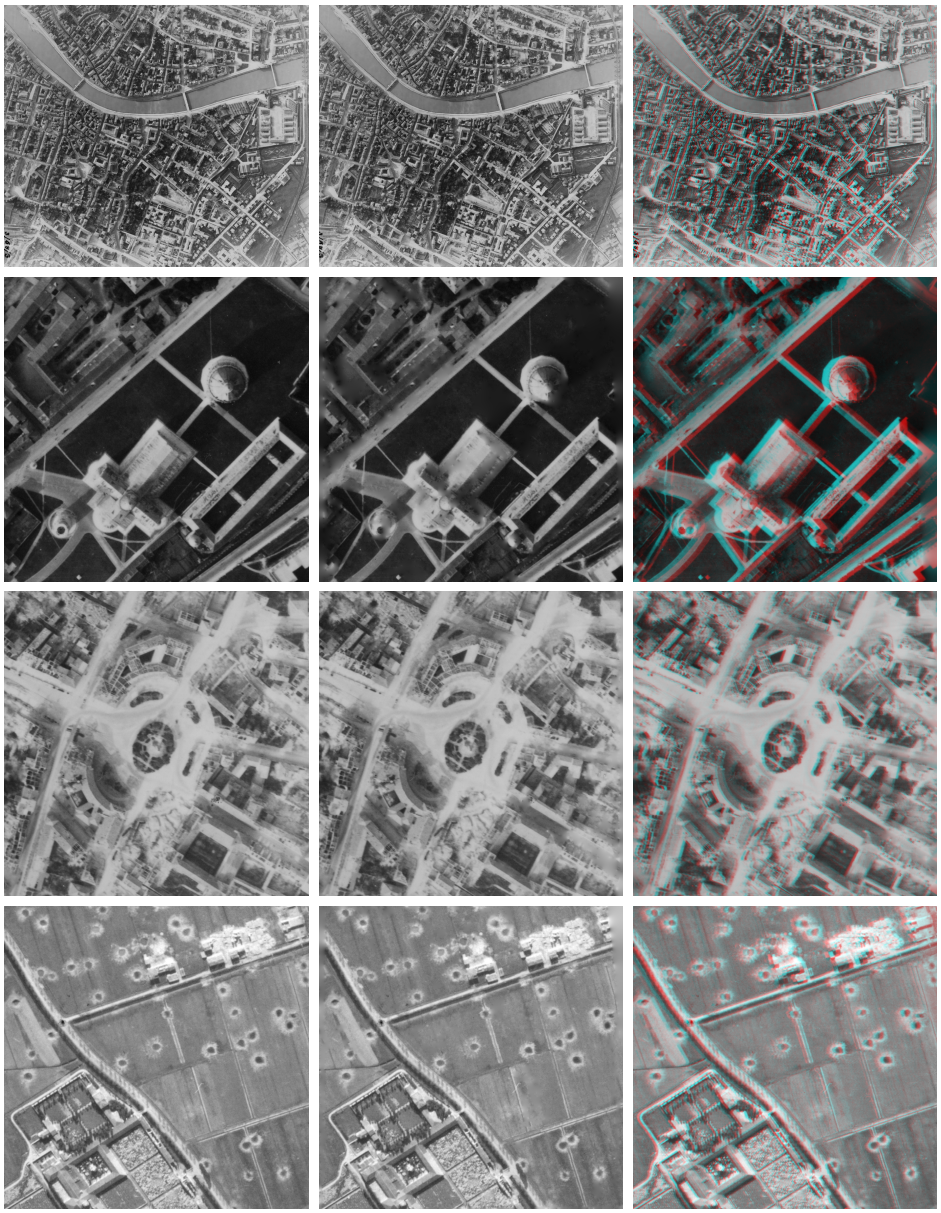

Figure 5.10: a more detailed view of the results. From left to right: reference image, synthetic image, red-cyan anaglyph (to be viewed in colour with suitable glasses). The second row depicts *Piazza dei Miracoli* with the famous leaning tower. MiBAC-ICCD, Aerofototeca Nazionale, fondo RAF ©.

# Chapter 6

# Conclusions

We began the work that lead to this thesis by looking at the big picture of the 3DS conversion problem and decided, in order to be able to make a significant contribution, to narrow our attention down to two key aspects of the problem: i) the computation of a depth-proxy and ii) the rendering of a synthetic image starting from a reference one. Given the nature of the problem, i.e. receiving video sequences as input, both parts share the idea that the proposed methods should be able to gather and exploit information coming from several images.

Regarding the computation of the depth-proxy, we presented a framework that allows to combine measurements obtained by processing the frames of a monocular video sequence. The integration takes place at two levels: i) temporal, where different estimates of depth-proxy values are merged along a timeline, and ii) spatial, where estimates are relaxed over pixel neighborhood. A segmentation into superpixels provides a spatial support that – in principle – does not cross objects boundaries.

Both spatial and temporal integration are derived as simple Kalman filters and are consistent with a data-fusion framework based on the Mahalanobis distance [59]. They exploit confidence values provided by the stereo matching step. In our experiments all the confidence measures provided comparable results, so there is no clear indication that one measure is superior to the others. Instead, it turns out that singling out occlusions (with LRC) makes a real difference. The spatio-temporal integration has shown to be effective, and the benefits of the spatial step have been demonstrated with respect to the temporal-only version with consistently better results. Also, a comprehensive review of the available depth-proxies has been presented in a unified framework and it has been shown how planar parallax can be applied with general motion and unknown camera parameters.

The method can be seen as an unconstrained, uncalibrated extension of

classical work [52], which constrained motion to be lateral and required camera internal parameters. Moreover, [52] warped the disparity map from frame to frame, thereby introducing errors and approximations that disrupted the prediction, whereas we fix this by keeping the reference frame constant. We demonstrate these facts with a comparison of our method against our implementation of [52].

We also report an extensive comparison of several confidence measures in the context of our approach. A preliminary version of this work appeared in [1] without the spatial relaxation and its current version, described in Section 3, has been presented in [2].

As a second main contribution, we presented a pipeline for uncalibrated view-synthesis of novel images that addresses most of the critical problems arising in uncalibrated scenarios. The method takes inspiration from the pipeline presented in [23], but instead of transferring points directly from the reference image to the novel one (forward), we use a *backward mapping* strategy which yields finer results and allows to combine information coming from *multiple reference images*, blending several parallax maps into one. We also propose a simple and suitable method to fill holes in the final virtual image and cope with resampling artifacts. We evaluated the method in the context of 3DS conversion of monocular images obtaining positive results that show a correct positioning of the virtual camera. We provided anaglyph images as qualitative output, processing indoor, outdoor and also aerial images. This work has been presented in [3] and in [4].

**Future works**

There are plenty interesting directions where our future works could go. First of all, the view-synthesis method still has limitations. Hole filling in the virtual image needs to be improved; where no information is available in the source images inpainting techniques (e.g. [16]) should be adopted. The probabilistic splatting step could also be improved by the working on superpixels (e.g. [8]). Heuristics to mitigate the effects of matching failures should also be investigated.

Regarding the motion-stereo part (and the computation of the depth-proxy map in general), moving objects are not dealt with. More specifically, the motion-stereo pipeline requires to work with a sequence of static images in order to produce a reliable depth-proxy map for a certain frame.

At last, the concatenation of motion-stereo and view-synthesis could be more than a simple concatenation. We feel that a better integration of the two methods could lead to better results as well.

# Appendix A

# Equality of two vectors up to a scale

Let $\mathbf{a}$ and $\mathbf{b}$ two vectors of $\mathbb{R}^n$. Their equality up to a scale can be written as: $\text{rank}[\mathbf{a}, \mathbf{b}] = 1$. This is tantamount to say that all minors of $[\mathbf{a}, \mathbf{b}]$ are zero. There are $n(n-1)/2$ of such order-two minors, and they can be obtained by multiplication of $\mathbf{b}$ by a suitable $n(n-1)/2 \times n$ matrix that contains the entries of $\mathbf{a}$. Let us call this matrix $[\mathbf{a}]_\times$ in analogy to the $\mathbb{R}^3$ case (see Appendix C.4), where equality up to a scale reduces to $\mathbf{a} \times \mathbf{b} = 0$. Since, by construction, $\mathbf{a}$ belongs to the null-space of $[\mathbf{a}]_\times$, its rank is at most $n-1$. Hence $\mathbf{a} \simeq \mathbf{b}$ gives rise to the linear system of $n(n-1)/2$ equations $[\mathbf{a}]_\times \mathbf{b} = 0$ where only $n-1$ of them are independent. The matrix $[\mathbf{a}]_\times$ is composed by $n-1$ blocks arranged by rows. The the $i^{th}$ block has $(n-i)$ rows and $n$ columns $(i = 1 \dots n-1)$:

$$B_i = \begin{bmatrix} \mathbf{0}_{1\times(i-1)} & -a_{i+1} & a_i & 0 & 0 & \dots & 0 \\ \mathbf{0}_{1\times(i-1)} & -a_{i+2} & 0 & a_i & 0 & \dots & 0 \\ \mathbf{0}_{1\times(i-1)} & -a_{i+3} & 0 & 0 & a_i & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1\times(i-1)} & -a_n & 0 & 0 & 0 & \dots & a_i \end{bmatrix} \tag{A.1}$$

and

$$[\mathbf{a}]_\times = \begin{bmatrix} B_1 \\ \vdots \\ B_{n-1} \end{bmatrix}. \tag{A.2}$$

# Appendix B

# Quasi-euclidean upgrade

When dealing with projective reconstructions a quasi–euclidean upgrade of the PPMs is of primary importance for a correct placement of the infinite plane at the actual infinity. This upgrade is based on [26].

Given a set of projective reconstructed PPMs $\{P_i\}$ $i = i \dots n$, which differ from the true ones by a collineation of space $H$, the set of camera matrices can always be transformed to the following canonical form by post-multiplying each $P_i$ by the matrix $[P_1;\ 0\ 0\ 0\ 1]^{-1}$:

$$P_1 = [I \mid \mathbf{0}] \qquad P_i = [Q_i \mid \mathbf{q}_i]\,. \tag{B.1}$$

In this situation, the collineation of space $H$ performing the Euclidean upgrade has the following structure:

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{v}^\top & \lambda \end{bmatrix} \tag{B.2}$$

where $K_1$ is the calibration matrix of the first camera, $\mathbf{v}$ a vector which determines the location of the plane at infinity and $\lambda$ a scalar fixating the overall scale of the reconstruction. A guess on the internal parameters is provided by the rectification step [24], as explained in Section 2.4. Given two PPMs and the guess of their intrinsic parameters, we compute the plane at infinity.

$$P_1 = [I \mid \mathbf{0}] \qquad P_2 = [Q_2 \mid \mathbf{q}_2] \tag{B.3}$$

and their intrinsic parameters matrices $K_1$ and $K_2$ respectively, the upgraded, Euclidean versions of the perspective projection matrices are equal to:

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{B.4}$$

$$P_2^E = K_2\,[R_2|\mathbf{t}_2] \simeq P_2 H = \left[Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top | \lambda \mathbf{q}_2\right] \tag{B.5}$$

The rotation $R_2$ can therefore be equated to the following:

$$R_2 \simeq K_2^{-1}\left(Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top\right) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{B.6}$$

in which it is expressed as the sum of a 3 by 3 matrix and a rank 1 term. Using the constraints on orthogonality between rows or columns of a rotation matrix, one can solve for $\mathbf{v}$ finding the value that makes the righthand side of Equation (B.6) equal up to a scale to a rotation. The solution can be obtained in closed form by noting that there always exists a rotation matrix $R^*$ such as: $R^*\mathbf{t}_2 = [\|\mathbf{t}_2\| \; 0 \; 0]^\top$. Left multiplying it to Equation (B.6) yields:

$$R^* R_2 \simeq \overbrace{R^* \; K_2^{-1} Q_2 K_1}^{W} + [\|\mathbf{t}_2\| \; 0 \; 0]^\top \mathbf{v}^\top \tag{B.7}$$

Calling the right hand side first term $W$ and its rows $\mathbf{w}_i^\top$, we arrive at the following:

$$R^* \; R_2 = \begin{bmatrix} \mathbf{w}_1{}^\top + \|\mathbf{t}_2\| \mathbf{v}^\top \\ \mathbf{w}_2{}^\top \\ \mathbf{w}_3{}^\top \end{bmatrix} / \|\mathbf{w}_3\| \tag{B.8}$$

in which the last two rows are independent from the value of $\mathbf{v}$ and the correct scale has been recovered normalizing to norm each side of the equation. Since the rows of the right hand side form a orthonormal basis, we can recover the first one taking the cross product of the other two. Vector $\mathbf{v}$ is therefore equal to:

$$\mathbf{v} = \left(\mathbf{w}_2 \times \mathbf{w}_3 / \|\mathbf{w}_3\| - \mathbf{w}_1\right) / \|\mathbf{t}_2\| \tag{B.9}$$

The upgrading collineation $H$ can be computed using Equation (B.2); the term $\lambda$ can be arbitrarily chosen, as it will just influence the overall scale of the reconstruction. Its sign however will affect the cheirality of the reconstruction, so it must be chosen positive if cheirality was previously adjusted.

# Appendix C

# Useful notions

In this appendix we recall some notions and definitions used in the thesis.

## C.1 Vectorization operator

The *vectorization* of a matrix is a linear transformation that converts a matrix in a (column) vector: vectorization of $A$ $m \times n$, denoted as $\mathrm{vec}(A)$, is the vector $mn \times 1$ obtained stacking all the columns of $A$.

## C.2 Kronecker product

Let $A$ be a matrix $m \times n$ and $B$ be a matrix $p \times q$. The Kronecker product of $A$ and $B$ is the $mp \times nq$ matrix defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}. \tag{C.1}$$

Note that the Kronecker product is defined for every pair of matrices.

The Kronecker product is connected to the vectorization operation defined in Appendix C.1 as given by the following equation

$$\mathrm{vec}(AXB) = (B^\top \otimes A)\,\mathrm{vec}(X) \tag{C.2}$$

for matrices $A, B, X$ having compatible dimensions. This equation is very useful to extract the unknown $X$ from a matrix equation.

For further reading on the Kronecker product and its uses in computer vision refer to [20].

## C.3 Sampson error

Geometrical errors, such as point-point distances, can be very computational demanding. A useful alternative is the Sampson approximation:

$$\sum_i \frac{(\mathbf{m}_i'^{\top} F \mathbf{m}_i)^2}{[F\mathbf{m}_i]_1^2 + [F\mathbf{m}_i]_2^2 + [F^{\top}\mathbf{m}_i']_1^2 + [F^{\top}\mathbf{m}_i']_2^2}. \tag{C.3}$$

As argued in [49], this residue yields similar results to the ones of (2.21).

## C.4 Cross-product matrix

Given two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{a} \times \mathbf{b}$ is equivalent to $[\mathbf{a}]_\times \mathbf{b}$ where $[\mathbf{a}]_\times$ is the skew-symmetric matrix defined as

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{C.4}$$

# List of Publications


[1] Francesco Malapelle, Andrea Fusiello, Beatrice Rossi, Emiliano Piccinelli, and Pasqualina Fragneto. Uncalibrated dynamic stereo using parallax. In *International Symposium on Image and Signal Processing and Analysis (ISPA)*, Trieste, Italy, 2013. IEEE.

[2] Francesco Malapelle, Andrea Fusiello, Beatrice Rossi, and Pasqualina Fragneto. A data-fusion approach to motion-stereo. In *Signal Processing: Image comunication*. El Sevier, 2016, IN PRESS.

[3] Francesco Malapelle, Andrea Fusiello, Beatrice Rossi, and Pasqualina Fragneto. Novel view-synthesis from multiple sources for conversion to 3DS. In *International Conference on Image Analysis and Processing (ICIAP)*, Genova, Italy, 2015. Springer.

[4] Francesco Malapelle, Anders Hast, Andrea Fusiello, Beatrice Rossi, Pasqualina Fragneto, and Andrea Marchetti. Automatic 3DS conversion of historical aerial photographs. In *International Conference on 3D Imaging (IC3D)*, Liège, Belgium, 2015. IEEE.

[5] Federica Arrigoni, Beatrice Rossi, Francesco Malapelle, Pasqualina Fragneto, and Andrea Fusiello. Robust global motion estimation with matrix completion. In *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Riva del Garda (TN), Italy, 2014.

[6] Andrea Fusiello, Fabio Crosilla, and Francesco Malapelle. Procrustean point-line registration and the NPnP problem. In *International Conference on Vision (3DV)*, Lyon, France, 2015. IEEE.

# Bibliography


[7] M. Abrate, C. Bacciu, A. Hast, A. Marchetti, S. Minutoli, and M. Tesconi. Geomemories: A platform for visualizing historical, environmental and geospatial changes in the italian landscape. *ISPRS International Journal of Geo-Information*, 2(2):432, 2013.

[8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Trans. on Patt. Analysis and Machine Intell.*, 34(11):2274 – 2282, 2012.

[9] B. D. Anderson and J. B. Moore. *Optimal filtering*. Prentice-Hall information and system sciences series. Englewood Cliffs, N.J. Prentice-Hall, 1979.

[10] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Trans Vis. and Comp. Graph.*, 4(4):293–306, 1998.

[11] J. Bourgeois and M. Meganck. *Aerial Photography and Archaeology 2003: A Century of Information*. Archaeological Reports. Academia Press, 2005.

[12] K. Brophy and D. Cowley. From the air: Understanding aerial archaeology. *Scottish Archaeological Journal*, 28(2):159–160, 2006.

[13] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[14] C.-C. Cheng, C.-T. Li, P.-S. Huang, T.-K. Lin, Y.-M. Tsai, and L.-G. Chen. A block-based 2D-to-3D conversion system with bilateral filter. In *Int. Conf. on Consumer Electronics*, pages 1–2, 2009.

[15] B.-T. Choi, S.-H. Lee, and S.-J. Ko. New frame rate up-conversion using bi-directional motion estimation. *IEEE Trans. on Consumer Electronics*, 46(3):603–609, 2000.

[16] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. on Image Proc.*, 13(9):1200–1212, 2004.

[17] M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. In *IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2009.

[18] O. D. Faugeras and S. Maybank. Motion from point matches: multiplicity of solutions. *International Journal of Computer Vision*, 4(3):225–246, 1990.

[19] P. Fragneto, A. Fusiello, L. Magri, B. Rossi, and M. Ruffini. Uncalibrated view synthesis with homography interpolation. In *2nd Joint 3DIM/3DPVT Conf.*, pages 270–277, 2012.

[20] A. Fusiello. A matter of notation: Several uses of the kronecker product in 3d computer vision. *Pattern Recognition Letters*, 28(15):2127–2132, 2007.

[21] A. Fusiello. Specifying virtual cameras in uncalibrated view synthesis. *IEEE Trans. on Circuits and Systems for Video Technology*, 17(5):604–611, 2007.

[22] A. Fusiello. *Visione Computazionale. Tecniche di ricostruzione tridimensionale*. Franco Angeli, Milano, 2013.

[23] A. Fusiello and L. Irsara. An uncalibrated view-synthesis pipeline. In *Proc. Int. Conf. on Image Analysis and Proc.*, pages 609–614, 2007.

[24] A. Fusiello and L. Irsara. Quasi-euclidean epipolar rectification of uncalibrated images. *Machine Vis. and Appl.*, 22(4):663 – 670, 2011.

[25] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[26] R. Gherardi, M. Farenzena, and A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1594 – 1600, San Francisco, CA, 2010.

[27] F. Gigengack and X. Jiang. Improved uncalibrated view synthesis by extended positioning of virtual cameras and image quality optimization. In *Asian Conf. Comp. Vis.*, pages 438–447, 2010.

[28] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2402–2409. IEEE, 2006.

[29] T. Gurdan, M. R. Oswald, D. Gurdan, and D. Cremers. Spatial and temporal interpolation of multi-view image sequences. In *Pattern Recognition*, pages 305–316. Springer International Publishing, 2014.

[30] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust statistics: the approach based on influence functions*, volume 114. John Wiley & Sons, 2011.

[31] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[32] A. Hast and A. Marchetti. Towards automatic stereo pair extraction for 3D visualisation of historical aerial photographs. In *3D Imaging (IC3D), 2014 International Conference on*, pages 1–8. IEEE, 2014.

[33] A. Hast and A. Marchetti. Stereo visualisation of historical aerial photos - a valuable digital heritage research tool. In *Digital Heritage*, pages 1–4, 2015. Short Paper.

[34] C. Hernández and G. Vogiatzis. Shape from photographs: A multi-view stereo pipeline. In *Computer Vision: Detection, Recognition and Reconstruction*, volume 285 of *Studies in Computational Intelligence*, pages 281–311. Springer, Berlin, 2010.

[35] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008.

[36] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conf. on Comp. Vis. and Patt. Rec.*, pages 1–8, 2007.

[37] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2121–2133, 2012.

[38] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *Proceedings of the European Conference on Computer Vision*, pages 17–30, 1996.

[39] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *Europ. Conf. Comp. Vis.*, pages 17–30, 1996.

[40] F. Isgro, E. Trucco, P. Kauff, and O. Schreer. Three-dimensional image processing in the future of immersive media. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(3):288–303, 2004.

[41] Y. J. Jung, A. Baik, J. Kim, and D. Park. A novel 2D-to-3D conversion technique based on relative height-depth cue. In *IS&T/SPIE Electronic Imaging*, pages 72371U–72371U, 2009.

[42] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, Inc., New York, NY, USA, 1993.

[43] S. B. Kang, J. A. Webb, C. L. Zitnick, and T. Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 88–, 1995.

[44] P. Kauff and O. Schreer. An immersive 3D video-conferencing system using shared virtual team user environments. In *Proceedings of the 4th international conference on Collaborative virtual environments*, pages 105–112. ACM, 2002.

[45] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.*, 31(4):34:1–34:8, July 2012.

[46] S. Laveau and O. Faugeras. 3D scene representation as a collection of images and foundamental matrices. In *Proc. Int. Conf. Patt. Rec.*, volume 1, pages 689–691, 1994.

[47] S.-H. Lee, O. Kwon, and R.-H. Park. Weighted-adaptive motion-compensated frame rate up-conversion. *IEEE Trans. on Consumer Electronics*, 49(3):485–492, 2003.

[48] H. Li and R. Hartley. Rectification-free multibaseline stereo for non-ideal configurations. In *Proceedings of the 13th international conference on Image Analysis and Processing*, pages 810–817, 2005.

[49] Q.-T. Luong and O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International journal of computer vision*, 17(1):43–75, 1996.

[50] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1018. IEEE, 2001.

[51] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[52] L. Matthies, T. Kanade, and R. Szelisky. Kalman filter based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[53] L. McMillan and G. Bishop. Head-tracked stereo display using image warping. In *Stereoscopic Displays and Virtual Reality Systems II*, number 2409 in SPIE Proceedings, pages 21–30, San Jose, CA, 1995.

[54] K. Mueller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. View synthesis for advanced 3D video systems. *EURASIP Journal on Image and Video Processing*, 2008(1):1–11, 2008.

[55] R. A. Newcombe and J. Andrew. Live dense reconstruction with a single moving camera davison. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[56] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.

[57] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Patt. Analysis and Machine Intell.*, 12(7):629–639, 1990.

[58] G. Ramachandran and M. Rupp. Multiview synthesis from stereo views. In *Int. Workshop. on Systems, Signals and Image Proc.*, pages 341–345, 2012.

[59] R. Ramparany. An integrated support for fusion perceptual information. In Groen, Hirose, and Thorpe, editors, *Intelligent Autonomuos Systems*, pages 500–508. IOS Press, 1982.

[60] D. N. Riley. *Air Photography and archaelogy.* University of Pennsylvania Press, 1987.

[61] P. J. Rousseeuw and A. M. Leroy. *Robust regression & outlier detection.* John Wiley & sons, 1987.

[62] J. Santos-Victor and J. Sentiero. Generation of 3D dense depth maps by dynamic vision. In *British Machine Vision Conference*, pages 129–138, 1992.

[63] H. Sawhney. 3D geometry from planar parallax. In *Computer Vision and Pattern Recognition (CVPR)*, pages 929–934, Jun 1994.

[64] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[65] S. Schwarz, M. Sjostrom, and R. Olsson. A weighted optimization approach to time-of-flight sensor fusion. *Image Processing, IEEE Transactions on*, 23(1):214–225, 2014.

[66] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, 1996.

[67] S. L. Spehn. Noise adaptation and correlated maneuver gating of an extended kalman filter. Naval Postgraduate School Monterey, CA, 1990.

[68] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Computer Vision ECCV'96*, pages 709–720. Springer, 1996.

[69] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[70] P. H. S. Torr. An assessment of information criteria for motion model selection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, pages 47 – 52, San Juan, Puerto Rico, 1997. IEEE.

[71] E. Trucco, V. Roberto, S. Tinonin, and M. Corbatto. SSD disparity estimation for dynamic stereo. In *Proceedings of the British Machine Vision Conference*, pages 342–352, 1996.

[72] J. K. Uhlmann. Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion*, 4(3):201 – 215, 2003.

[73] G. Vogiatzis and C. Hernàndez. Video-based, real-time multi-view stereo. *Image and Vision Computing*, pages 434–441, 2011.

[74] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Proc.*, 13(4):600–612, 2004.

[75] A. M. Waxman and S. S. Sinha. Dynamic stereo: Passive ranging to moving objects from relative image flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):406–412, 1986.

[76] D. R. Wilson. *Air Photo Interpretation for Archaeologists.* Tempus, 2000.

[77] www.diegm.uniud.it/fusiello/demo/dsp.

[78] G. Zhang, W. Hua, X. Qin, T. Wong, and H. Bao. Stereoscopic video synthesis from a monocular video. *IEEE Trans. on Vis. and Comp. Graph.*, 13(4):686–696, 2007.

[79] G. Zhang, J. Jia, T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Trans. on Patt. Analysis and Machine Intell.*, 31(6):974–988, 2009.

[80] G. Zhang, J. Jia, T.-T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):974–988, 2009.

[81] L. Zhang, C. Vázquez, and S. Knorr. 3D-TV content creation: automatic 2D-to-3D video conversion. *IEEE Trans. on Broadcasting*, 57(2):372–383, 2011.

[82] C. Zhu, Y. Zhao, L. Yu, and M. Tanimoto. *3D-TV System with Depth-image-based Rendering.* Springer, 2014.