UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École Polytechnique de Louvain
ICTEAM Institute

3D estimation and view synthesis in wide-baseline stereo

Cédric VERLEYSEN

Thesis submitted in partial fulfillment of the requirements for the degree of
Doctor in Applied Sciences

Ph.D. Jury

Prof. Christophe DE VLEESCHOUWER	UCL, ICTEAM, Belgium
Prof. Pascal FROSSARD	EPFL, LTS4, Switzerland
Prof. Laurent JACQUES	UCL, ICTEAM, Belgium
Prof. Gauthier LAFRUIT	ULB, VCL, Belgium
Prof. Luc VAN GOOL	ETHZ, CVL, Switzerland
Prof. Jean-Pierre RASKIN (chairman)	UCL, ICTEAM, Belgium

November 2015

Acknowledgments

First and foremost, I would like to thank my advisor, Christophe De Vleeschouwer. From the scientific point of view, Christophe has played a role model in my developement as a researcher. His wide knowledge of computer vision has pushed me to open my mind to not only the topic of my research (*i.e.*, view interpolation) but also other subjects, which had a profond impact on my ability to propose novel solutions. From the management aspect, he, with his motto "PhD is not a sprint but a marathon", has been the best coach that a PhD student could ever dream. His endless enthousiasm and optimism have made my thesis fascinating and thrilling, to the point where I used to loose all sense of time (email threads at past midnight or numerous extension of "5 minutes meetings" to hours of brainstorming).

I would also like to thank Laurent Jacques, who has been an astonishing mentor since my master thesis. I had great pleasure to share a tremendous amount of discussions about both technical and non-technical subjects (from compressed sensing to the best way to spike a volleyball). His contagious passion for mathematics has enabled me to keep a feet in this more theoretical but no less important world that I like.

Both Christophe and Laurent have been the instigators of my research internship at the École Polytechnique Fédérale de Lausanne (EPFL, Switzerland). Without their support, I would probably never have been invited by Pascal Frossard to work in his LTS4 laboratory, the most stimulating research atmosphere that I have known. In the LTS4 lab, the team spirit is so strong that the working-hours colleagues (especially Thomas Maugey, Alhussein Fawzi, Dorina Thanou, Xiaowen Dong and Elif Vural) used to transform into friends for the weekends, with whom I went visiting Switzerland, went to the pubs, sailed on the Leman Lake, etc. Thanks a lot for these unforgettable moments.

My gratitude also goes to the other members of my PhD committee for their time, effort and brillant suggestions. I believe that the quality of this thesis has greatly improved due to their constructive comments. Due to his prominent position in the MPEG committee, Gauthier Lafruit is among the very restricted group of experts who defines the future video standards, and I have been really honoured by its presence in my PhD committee. His industrial expertise has been perfectly completed by Luc Van Gool, who is among the top-5 best academic researchers in the field of computer vision[1]) and a central pillar of the most prolific labs (KULeuven and ETHZ) in 3D estimation and view synthesis.

I would also like to express my special thanks to my two closest collaborators and friends in the ICTEAM: Amit K.C. and Pascaline Parisot. They took so much care of me during my PhD that I own them my extra 5kg got during the PhD (thanks for the cakes!). Amit has been the best officemate one could have during his/her PhD: not only his expertise in computer vision has led to great discovers in my research, but also his english skills would turn pale any

[1] `http://academic.research.microsoft.com/RankList?entitytype=2&topDomainID=2&subDomainID=11&last=0&start=1&end=100` (consulted on 29 September 2015).

reviewer. I also had great pleasure to propose/supervise some crazy projects of the "Image processing and computer vision" lecture with him. Pascaline has been an extraordinary colleague over the years, and the most altruistic person that I know. Extra thanks goes to her insightful comments about my PhD manuscript, and the rollicking good times that we had when travelling (I mean attending to conferences).

Furthermore, my time at UCL has been made very pleasant in large part due to the many groups that became a part of my life. Among the researchers in my lab, I would like to thanks Damien Delannay, Quentin de Neyer, Arnaud Browet, Arnaud Delmotte, Damien Jacobs, Mireia Montanyola, Maxime Taquet, Ivan Alen Fernandez, Adriana Gonzalez, Augustin Cosse, Adrià Gusi Amigó, Lionel Lawson, Damien Leroy, Sébastien Lugan, Kaori Hagihara, Mohieddine El Soussi, Kévin Degraux, François Rottenberg, Pierre-Yves Gousenbourger, Stéphanie Guérit, Valerio Cambareri, Amir Moshtaghpour, Thomas Feuillen, Arnab Bhattacharya, Jeevan Shrestha, Benoît Macq, Christian Van Brussel, Anthony Legrand, Benoît Pairet, Cosmin Ancuti, etc.

I also would like to thanks my colleagues who have helped me to validate the maxim "a healthy mind in a healthy body". A special thanks goes to my teammates of the ICTEAM volleyball team, principally Charles Pecheur, Olivier Bonaventure, Pierre Reinbold, Fabien Duchêne and Romain Hollanders (who has helped me several times after the volleyball, with Nicolas Boumal, on some optimization problems).

These acknowledgements would not be complete if I did not mention the members of the IEEE student chapter and of the Ascii association. People do not realize how important and time-benefit it is to have researcher representatives, and I have been glad to share this role for four years with outstanding colleagues, such as Sébastien Combefis, Pierre-Antoine Haddad, Numa Couniot, François Botman and Thomas Walewyns.

Finally, I would like to thank all who supported me outside of the lab, including my friends (especially Arnaud, Adrien, Alexis, Nicolas, François, Gautier and Nathalie) for their encouragement and interest in my PhD. Words cannot express how grateful I am to my beloved parents, especially for having initiated my passion for engineering since my childhood, by showing me how fun and rewarding it can be to repair/build something. More than 20 years later, they still show the same enthusiasm to warmly welcome me at 4-5AM, during the rush periods spend on other "as fun and rewarding" projects (*i.e.*, presentations and papers). It is also impossible not to thank my sister, Magali, who has continuously incented me to strive towards my goal, and my brother, Quentin, who has always been able to make me smile during our multiple discussions about my "professional student" status. Last but not least, this thesis would not have been possible without the constant support of my life spouse, Emilie Willems. Her optimism, cheerfulness and love have been motivational driving forces behind the pursuit of my PhD, especially during the stressful times. Despite the fact that my mistresse (*aka* PhD thesis) has been quite intrusive in our private sphere, she made me the happiest man in the world for the last 9 years, and I will be eternally grateful to her.

Contents

Symbols and Notations

$\mathbf{\Omega}_{\mathcal{I}}$	The 2D domain of the image $\mathcal{I}$.
γ_X	The yaw Euler's rotation angle (about the x axis).
γ_Y	The pitch Euler's rotation angle (about the y axis).
γ_Z	The roll Euler's rotation angle (about the z axis).
$\boldsymbol{\pi}$	The vector the 3D plane $ax + by + cz + d = 0$.
$\mathbb{C}$	The set of complex numbers.
$\mathbb{R}^*$	The set of real numbers, excepted 0.
$\mathbf{e}$ ($\mathbf{e}'$)	The epipole of the first (respectively second) view.
$\mathbf{H}$	A (non-singular) 3×3 homography operator.
$\mathbf{H}_{\boldsymbol{\pi}}$	The homography operator induced by the 3D plane $\boldsymbol{\pi}$.
$\mathbf{I}$	The identity matrix (size defined by the context).
$\mathbf{l}'$ ($\mathbf{l}$)	The epipolar line associated to a 2D point $\mathbf{x}$ ($\mathbf{x}'$) in the second (respectively first) view.
$\mathbf{P}^+$	The Moore-Penrose pseudoinverse of $N \times M$ matrix $\mathbf{P}$.
$\mathbf{x} \leftrightarrow \mathbf{x}'$	The 2D coordinates $\mathbf{x}$ and $\mathbf{x}'$ in two different views represent the projection of the same 3D point.
$\mathcal{I}(\widetilde{\mathbf{x}})$	The brightness of the pixel with homogeneous coordinates $\widetilde{\mathbf{x}}$.
$\mathcal{SO}(3)$	The special orthogonal group of dimension 3.
$\|.\|_2$	The ℓ_2 norm.
$[\mathbf{v}]_\times$	The skew-symetric matrix form of the vector $\mathbf{v}$.
r	The radial ℓ_2 distance with respect to the lens center's point.
$vec(.)$	The matrix vectorization operator.
EPI	Epipolar Plane Image.
EPIV	Epipolar Plane Image Volume.
NW	The Needleman-Wunsch algorithm [162].
SNR	The signal-to-noise ratio.

CHAPTER 1
Introduction

This chapter introduces the concept of view interpolation, defines our associated research statement, and summarizes our key contributions.

1.1 View interpolation in a nutshell

View interpolation refers to the process of synthesizing images that would be seen from a different viewpoint than the ones captured by real cameras. As illustrated in Figure 1.1, it aims at generating a virtual view of a scene based on a set of images representing it and having a common field of view.

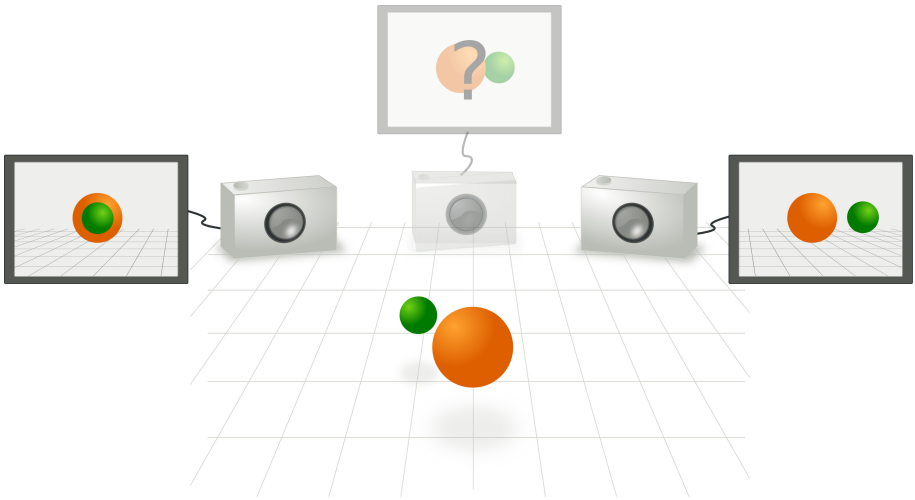
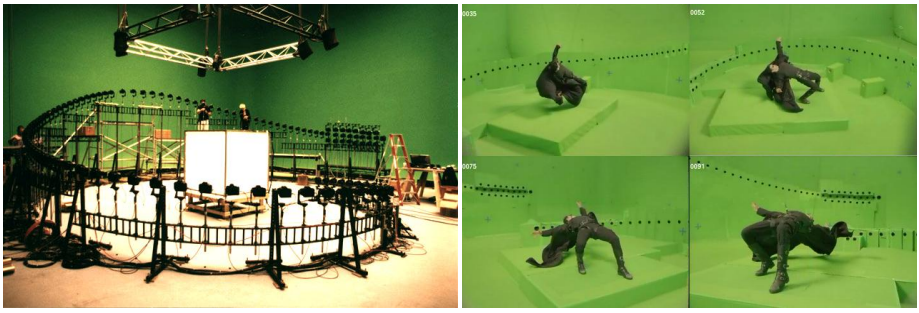


Figure 1.1: View interpolation aims at synthesizing novel views of a scene from a set of images representing it.

By reconstructing a continuous and smooth sequence of virtual views, the most general form of view interpolation, called Free-Viewpoint Rendering (FVR) [212][196], allows to freely navigate within a real world scene.

Although some basic notions such as perspective distortion [6] and binocal vision [236] have been discovered several centuries ago, the first virtual images have been numerically generated about 25 years ago on simple controlled-environment cases [54] [184] [52] [185].

The industry has directly shown a great interest in this technology, especially in the sport entertainment and film making markets. The earliest commercial product attempting to give to the viewer the feeling to navigate around a scene was the TimeTrack system [214], which rapidly jumped between a few hundreds of consecutive cameras placed along a 360° arc surrounding the scene, as illustrated in Figure 1.2(a). This system, specially designed to give the astonishing illusion of stopping the time and turning around the characters in the movie "The Matrix" (see Figure 1.2(b)), had to be organized much in advance. Indeed, not only the viewpoint trajectory had to be planned weeks in advance, to install the 120 precisely-mounted cameras, but hundreds man-hours were also required for the intensive manual post-smoothing of the transitions [84].



(a) The "bullet-time" setup (b) The "bullet-time" effect

Figure 1.2: The "bullet-time" effect, as seen in the movie "The Matrix", was the first one to give the feeling of navigating around a scene.

This principle was then transferred from the small and controlled studio environment to the large, illumination variant sport arenas based on the work of Kanade *et al.* [106] [107]. The new associated product was called the Eye-Vision System and has been used at the Super Bowl XXXV (2001). It was composed of more than 30 motorised pan-tilt cameras, which were manually controlled to ensure looking towards the action. However, such **hardware transitions** were still producing noticeable jumps when switching between the cameras [86].

Nowadays, thanks to the theorical and technological advances detailed in Chapter 2, companies such as Vizrt[1] (previously called LiberoVision[2]) and freeD™[3] offer to the consumers the ability to generate a smooth transition between some fixed cameras, by **numerically interpolating** in-between these views. Those applications only focus on stadium sport events (*e.g.*, soccer games [80] [73] [86] [101] [147]), giving to the viewer the feeling of being "inside the scene".

[1] `http://www.vizrt.com/products/viz_libero/`

[2] Their technology is principally based on the results obtained during the FP7 European project named "FINE".

[3] `http://replay-technologies.com/`

As main drawbacks that slow down their entry on the market, these digital technologies still require either to:

- observe the scene with a large amount of fixed cameras, called the *reference cameras* [73]. For example, the freeD™ technology uses from 8 to 32 full-HD reference views.

- observe and interpolate the scene from very distant viewpoints, with respect to the maximum thickness of the object/scene to synthesize [113].

To lower these requirements, virtual view synthesis has become, since a few years, an extensive research domain at the intersection of computer vision, image processing and computer graphics.

While reducing the amount of required cameras decreases the production cost, reducing their distances to the scene allows to consider new markets, such as the intermediate view synthesis in confined environments. One might directly think of the immersive rendering of cultural or indoor sport events, such as the generation of intermediate views in small theatres, around a dancer, around a martial art fighter, etc. But there also exists a plenitude of more hidden applications. Here, we only describe two easy examples, namely videoconferencing and virtual navigation through cities. In video conferences, high-end remote collaborations are made possible by generally capturing the faces and voices of the participants and transmitting them to all the members, located on different sites [85]. One of the most disturbing artefact of the most current practical systems is the loss of eye contact. This is due to the fact that the participants generally look towards their screens, instead of looking towards the camera which records their face. It implies that, while retransmitting their images to the other members, the presenter's gaze is not directed towards these members, leading to an uncomfortable feeling. By placing a few reference cameras around the presenter's screen, the interpolation of frontal views of its face [169] can create instantaneous views of remote speakers which are consistent with the local member's viewpoint [103].

The second example follows the actual trend of companies like Apple, Google, Accute3D, Blom, etc. to permit to the users to virtually navigate in cities, for example to visually discover a place before really visiting it [199]. Nowadays, these companies generally offer to navigate either through large-scale city models typically observed based on aerial images, which do not capture street level details, or through simple panoramic pictures of street-levels pictures captured by a car. This last feature, the most-demanded one[4], however exhibits strong artefacts when passing in-between the reference images of the panoramic composition. These artefacts, caused by the large depth changes of the buildings, with respect to the small depth separating them with the car, illustrate the necessity of improving the current view interpolation techniques, briefly introduced in the next section.

[4] http://www.theguardian.com/technology/appsblog/2012/jun/06/google-maps-3d-street-view1

1.2 How to interpolate a virtual view?

When it comes to synthesize numerically a virtual view, two fundamental questions arise: which algorithm should be used and what are the associated hardware requirements in terms of camera deployment? This section aims at giving a succinct answer to those two questions and concludes with the observation that the distance between the camera and the scene to render fundamentally affects the method adopted to interpolate images in-between the cameras.

1.2.1 Image-based rendering versus model-based rendering

In the literature, the methods that interpolate intermediate views of a 3D scene are often classified as a continum in-between two groups, namely *model-based rendering* and *image-based rendering* methods. The reader is refered to Chapter 2 for a detailed technical presentation of the different algorithms and methods composing the main software trends adapted to generate intermediate views.

In model-based rendering, the 3D geometry of the observed scene is explicitly estimated from the reference images. In most cases, this 3D geometry is represented on the basis of 3D point clouds, 3D squared area (called *voxels*) representing the occupancy of the scene in a predefined 3D grid, or based on smoother 3D meshes [213]. Adequate textures are then mapped on this 3D model, consistently with the observation in the reference views. Finally, this textured 3D model can be projected onto any arbitrary viewpoint, similarly to what is done when generating synthetic images in classical computer graphics. Model-based rendering offers thus a full freedom on the virtual viewpoint selection, at the price of requiring an accurate and robust 3D reconstruction of objects and scenes, which can often not be guaranteed [197] [147].

The other extreme group is called image-based rendering (IBR) and does not use any 3D model at all. It directly interpolates the virtual view in the image color space without explicit reconstruction of a 3D surface [108] [190]. It aims at finding a relation among the reference images, such as a (mutliple) predefined geometric transformation(s) (*e.g.*, rigid, affine, projective, etc.). The parameters of such relation(s) are extracted based on the determination of pixels or regions representing the same part of the scene, which are called *correspondences*. Image stitching, also well-known as panoramic image synthesis, is one of the most easy image-based rendering method that allows to generate an intermediate viewpoint by cropping in the reconstructed panoramic image. It generally refers to the process of gathering different overlapping images captured by a set of cameras (or a unique camera) that differ in their relative viewing poses and positions, which is generally modeled as a homography and requires four (or more) pixel correspondences to be determined. The panoramic image is then generated by transfering all the reference images into a common coordinate frame, based on the estimated homographies. However, as in most image-based rendering methods, image stitching requires a tremedeous amount of references images captured with relatively close viewpoints [197] to interpolate a large transition in-between two extreme reference images. Moreover, the trajectory of this transition is strongly constrained, as detailed in Sections 2.5.2 and 2.5.3.

In-between model-based rendering and image-based rendering, there exists a wide range of methods that attempt to combine the advantages of both groups. They represent the 3D of the scene based on grayscale images, called *depth* or *disparity* maps. Each pixel value of such maps is inversely proportional to the depth value, with respect to the camera view, of the 3D point which is imaged at this pixel location. Brighter areas are closer from the reference cameras, while darker areas are further in the background. From these maps, virtual views can be generated in a limited operating range around the real cameras, based on *depth image-based rendering* (DIBR) methods [197]. Those methods achieve realistic results already from a relatively small number of images, as long as those images are captured from similar viewpoints [77].

All those methods share a common challenge: the *search of correspondences* among the reference images. The difficulty of this task, which is still a very active and important research area, directly depends upon the spatial organization of the reference cameras, the discriminant nature of the scene content, and the type of sensors used, as introduced in the next section.

1.2.2 Acquisition setups

Intermediate view synthesis approaches rely on specific acquisition systems, composed of mutliple camera sensors whose images are captured synchronously. Generally, these cameras only observe the scene without interfering with it, and are thus called *passive cameras*. These are sometimes completed by depth sensors [13], which emit light onto the 3D scene to measure its depth, and are thus called *active cameras*. These active cameras generally use infrared wavelengths and estimate the depth of the scene either by measuring the time-of-flight of a light ray[5] between the camera and the scene for each point of the image, or by observing how a structured light pattern is distorted when projected on the 3D scene. Typical time-of-flight depth sensors have either a relatively low resolution, or a low depth range, or a very high cost (tens to hundreds of thousands dollars). Moreover, to exploit their captured 3D informations, their measured depth maps have to be registered with the color images acquired by the reference passive cameras, which is not an easy task since both types of cameras are inherently located at different positions [35]. Finally, low-price depth sensors are very sensitive to noise (*e.g.*, other infrared sources, such as the sun) and temperature. The impact of the noise is emphasized by the intrinsic non-linear distortions of those sensors [197]. Because of these limitations, **this thesis focuses on the synthesis of intermediate views captured only by passive cameras**.

Four spatial layouts are generally considered for passive camera networks, as illustrated in Figure 1.3: the cameras can be placed along a 3D line (Figure 1.3(a)), along an arc (Figure 1.3(b)), on a plane (Figure 1.3(c)) or spread (forward or outward) on a dome (Figure 1.3(d)).

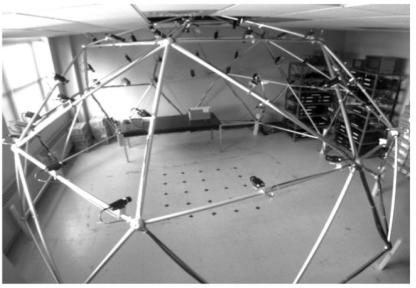
The spatial organization is chosen according to the area of the 3D scene and the degree of freedom that are desired to be covered by the virtual view.

[5] The propagation speed of the light ray is assumed to be known.

(a) Linear configuration

(b) Circular configuration

(c) Planar configuration



(d) Dome configuration

Figure 1.3: There exists multiple spatial configuration for networks of passive cameras (images from the courtesy of M. Tanimoto [212]).

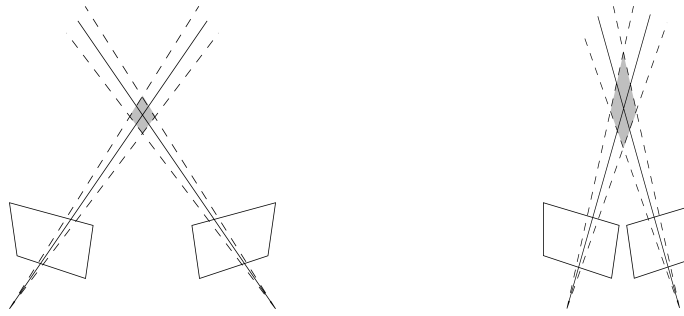
Because of the difficulty to install large linear, planar or dome configurations, all of them are generally restricted to confined spaces, such as theatres [202], TV studios, or laboratories [212]. At the opposite, circular (arc) settings have mainly been used to cover large area, such as sport stadiums, in which only a few widely spaced cameras are used as reference views.

The distance separating two reference cameras is measured along the line joigning their (optical) centers, which is called the *baseline* of the cameras. The length of the baseline is one of the most important parameters of an acquisition setup, because it imposes practical limitations on both the navigation range and the quality of the synthesized virtual views [218]. This measure is sometimes compared relatively to the minimum depth of the 3D scene, and leads to 3 groups of acquisition setups: *narrow-baseline*, *small-baseline* and *wide-baseline*. Practically, the term narrow-baseline setups generally refers to configurations in which the reference cameras are separated from a few microns (microlens arrays) to a few centimeters while the 3D scene is situated away from centimeters to meters. The second group refers to reference cameras separated of a several centimeters, while the 3D scene is situated away from centimeters to meters. The last group is typical to stereo setups in which the reference cameras are separated by meters, while the 3D scene is situated away from centimeters to meters. Each of these configurations have their own advantages and disadvantages, which are detailed in the next section.

1.2.3 Narrow/small-baseline versus wide-baseline camera networks

One of the main advantages of narrow and small baseline setups is that their high camera density ensures to observe the scene with large overlaps and low perspective distortions, simplifying the search of correspondences, at the core of all intermediate view synthesis techniques. This also makes narrow and small-baseline stereo well understood since decades [29] [181] [127] and algorithms are generally designed for small-baseline stereo pairs and their almost fronto-parallel scenes [110]. In contrast, its wide-baseline counterpart is much more challenging due to large perspective distortions and increased occluded parts [219]. Wide-baseline setups are nevertheless worth investigating because:

- they require fewer images to reconstruct the complete scene, facilitating the camera system deployment. Indeed, not only the cameras have to be mounted, synchronized, geometrically and colorimetrically calibrated (see Sections 2.1 and 2.3), but also the CPU processing the images, the network rooter, hard disks and cables have to be adapted to sustain high image flow rates (*e.g.*, tens of gigabytes per second) when working with a dense camera network, as it is the case in narrow and small-baseline setups.
- their 3D estimation is less impacted by unprecise correspondences[6], determined for example at the pixel level instead of the subpixel level. This principle is illustrated in Figure 1.4, in which the plain lines represent the 3D projection of accurately located 2D corresponding points, the dashed lines represent an uncertainty interval around the accurately located 2D points and the shaded regions is the 3D uncertainty region. The comparison of the area of the shaded regions shows that wide-baseline setups yield to more accurate 3D models than narrow/small-baseline ones.



(a) Wide-baseline setup (b) Narrow/small-baseline setup

Figure 1.4: Wide-baseline setups are less sensitive to inaccurate 2D correspondences, yielding to more accurate 3D estimations (smaller uncertainty shaded regions) than narrow/small-baseline setups [92].

[6]Wide-baseline setups are also less impacted by calibration errors, detailed in Section 2.1.

To benefit both from the easier search of correspondences of small-baseline setups and the accurate 3D associated to these correspondences in wide-baseline setups, the two types are more and more often used simultaneously. In this case, the small-baseline disparity/depth map is generally used to bias and complete the search of correspondences in the wide-baseline stereo pair [249].

1.3 Problem statement

Even when combining small and wide-baseline setups, depth estimation is still a challenging problem, which is still a very active and important research area. One of the main reason is that this problem can not be solvable uniquely, as explained in the next section.

1.3.1 Ill-posedness of virtual view interpolation in wide-baseline setups

One of the challenges of intermediate view synthesis lies in the fact that multiple different 3D models can generate the same reference images. It implies that recovering the 3D scene from multiple reference views does not admit a unique solution, and is thus an *ill-posed problem*. This phenomenon is illustrated in Figure 1.5, where the two presented 3D models are imaged indistinguishably by two reference views located on the left and on the right of the scene.

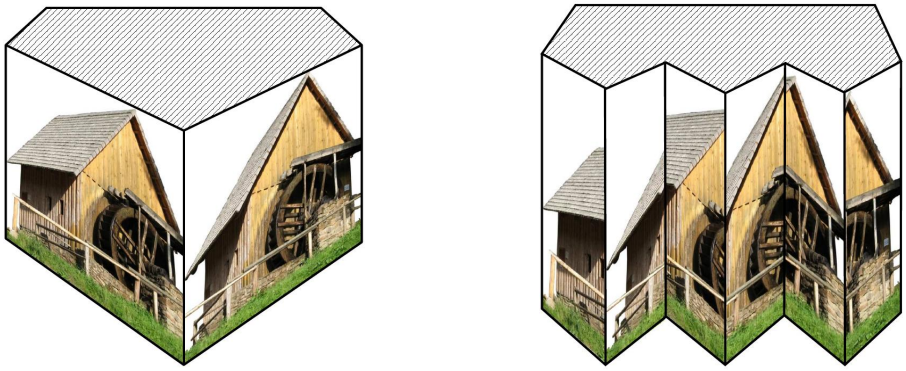


Figure 1.5: These two 3D models are consistent with the reference images captured by two reference cameras, facing respectively the left wall of the building and the water mill. Recovering a 3D model from 2D reference images is thus an ill-posed problem.

This ill-posed nature obviously results from the irreversible loss of informations when imaging a 3D scene onto a 2D image. This ambiguity does not only concern the 3D shape of the scene, but also its color [186]. The level of the ambiguity depends thus on the geometrical structure of the scene, on its textures and on the camera configuration, as deeply explained in [10].

1.3.2 Research questions

The ambiguity of the 3D model can be decreased by using some additional prior knowledge and constraints, when available, or by using more reference views in the reconstruction process [11]. The central and federative research question of this thesis consists in investigating how priors can help to disambiguate the view interpolation ill-posedness. Three different priors are considered. As introduced in the next section, the first one is related to the spatial organization of the scene, while the two others consider its 3D geometry, either through planar approximation of a far-away/man-made scene or by learning the plausible 2D projected silhouettes of a dynamic object of interest.

1.4 Contributions and thesis outline

As we have seen in the previous sections, the more reference cameras are used, the less ambiguous is the interpolation of intermediate views, but the more costly the setup is. Also, the wider is the distance separating these reference cameras, the larger can be the scene to reconstruct, and the more precise the reconstruction will be, but at the price of a more challenging search of correspondences. This thesis tackles the combination of these two most challenging cases: **we aim at synthesizing intermediate views of a scene captured by only two widely spaced cameras.** Chapter 2 first reviews the background material, including the modeling of a camera as a "pin-hole" sensor, the different ways of representing the images and how to find correspondences among them, the different priors used to disambiguate the correspondence problem, and finally, how to generate a virtual view when these correspondences are known. Then, the next three chapters detail the main contributions of this thesis, which are:

1. the relaxation of one of the most well-known small-baseline stereo prior to the wide-baseline stereo case. This prior, called the *ordering constraint*, enforces the strict preservation of the "left-right relations"[7] between the elements composing the reference images. It is valid in small-baseline setups, but appears to be violated in a wide-baseline stereo pair [202], as illustrated in Figure 1.6.

   In Chapter 3, we propose to disambiguate the search of correspondences in wide-baseline stereo setups based on a relaxed version of the ordering constraint, which only favors the preservation of the order of the elements without necessary strictly forcing it. To the author's best knowledge, this *soft ordering constraint* has never been investigated in the matching litterature, which led us to introduce an original energy-based optimization framework to address it.

[7] Quotes are used to emphase on the fact that these relations are only valid along corresponding epipolar lines, as detailed in Chapter 3.

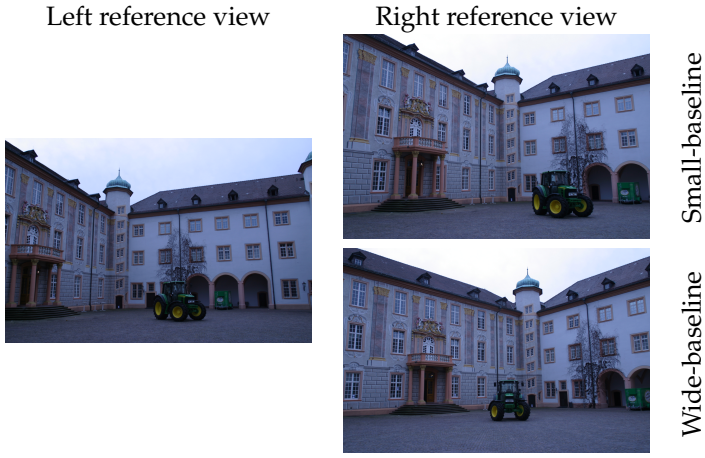


Figure 1.6: The "left-right relations", *e.g.*, the tractor is on the right of the tower, are strictly preserved among the reference views captured by a small-baseline stereo pair. However, the wider is the distance separating the reference cameras, the more likely those "left-right relations" are violated, although they still enable to disambiguate the search of correspondences.

2. the piecewise planar approximation of the background of a 3D scene captured by only two wide-baseline cameras. This light-weighted, automatically generated 3D model is composed of the minimum set of 3D planes that jointly best describe the 3D scene, and is specially appropriate to interpolate intermediate views of man-made scenes. The associate energy minimization problem is described in Chapter 4.

3. the intermediate view interpolation of dynamic (foreground) objects captured by only two widely spaced cameras. The interpolation is disambiguated based on a prior knowledge about the plausible shapes of the 2D projected object silhouettes. This prior is learnt before the interpolation, only from the images observed by the reference views. As explained in Chapter 5, it guides both the definition of pair-wise correspondences between the reference views, and the synthesis of intermediate views in regions that get occluded while going from one reference view to the other. To the best of our knowledge, our work is the first one to propose a solution to synthesize intermediate views of occluded parts, without a specific 3D model.

Although being presented in independent chapters for the sake of clarity, these three complementary contributions permit, altogether, to synthesize intermediate views of an arbitrary scene. Precisely, the wide applicability of the relaxed ordering prior (Chapter 3) permits to reconstruct the 3D of an arbitrary scene, at the price of inaccuracies in the depth estimation. For improved accuracy, Chapters 4 and 5 propose more specific priors (respectively the piecewise-planarity and object shape priors) to reconstruct separately the background and the foreground of a scene, when such separation is possible.

CHAPTER 2

Background and state-of-the-art

This chapter reviews the background material needed for the developments of this thesis, as well as the mathematical frameworks of the seminal and state-of-the-art methods in view interpolation.

In the literature, state-of-the-art methods that interpolate intermediate views of a 3D scene are often classified as a continuum in-between two groups, namely *model-based rendering* and *image-based rendering* methods. In model-based rendering, the 3D geometry of the observed 3D scene is explicitly estimated from the reference images. Adequate textures are then mapped on this 3D model and projected onto any arbitrary viewpoint. Image-based rendering (IBR) methods interpolate directly the virtual view in the image color space without explicit reconstruction of a 3D surface [108] [190]. Although being based on very different concepts, all of these methods require to:

1. Estimate the (relative) position, orientation, etc. of the reference cameras, *i.e.*, their *projective geometry*. This allows, for example, to define the parameters of a virtual view with respect to the reference ones (see Section 2.2.3) or to avoid topologically incoherent deformations during the transition from one view to another (see Section 2.5.2), as illustrated in Figure 2.1.

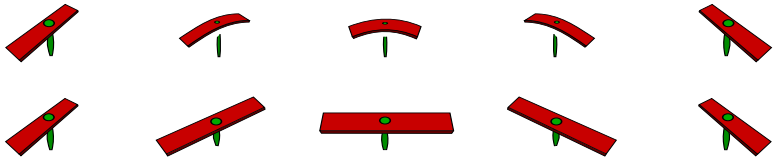


Figure 2.1: Any (unconstrained) interpolation in-between two views can cause topologically inconsistent geometrical deformations of the observed 3D scene (first row, generated by linearly interpolating in-between corresponding points in the two extreme views), which can be avoided by considering the (projective) geometry linking the two views (second row, see Section 2.5.2, image courtesy from S. Seitz [182]).

2. Estimate correspondences between the reference images, either to triangulate a 3D model (see Section 2.2.2) or to use them as anchor points for the 2D interpolation (see Section 2.3.3).

3. Generate the intermediate views, either based on the estimated 3D model (see Section 2.5.1) or on these anchor points (see Sections 2.5.2 and 2.5.3).

The rest of the chapter surveys the earlier key contributions related to those three points. Precisely, while the estimation of the (projective) geometry of a single camera, detailed in [92], is summarized in Section 2.1, Section 2.2 investigates how to exploit this knowledge to constrain the set of possible correspondences in-between the reference views. Given such a set of possible correspondences, the colorimetric information is generally used to determine the best matches of points between two views. However, as explained in Section 2.3, this information is not invariant with respect to the camera's viewpoint, meaning that the same (infinitively small) 3D surface can be represented with a different color in the two reference images. To mitigate this problem, image descriptors that tend to be robust to this change of viewpoint have been designed. They characterize the local appearance of 3D points on the reference images, and Section 2.3 details their state-of-the-art for wide-baseline configurations. As explained previously in Section 1.3.1, even if the correspondence is determined on (a geometrically constrained set of) projective-invariant descriptors, the 3D reconstruction remains an ill-posed problem. To regularize this ill-posed problem, Section 2.4 surveys the state-of-the-art priors generally used to disambiguate the solution. Finally, Section 2.5 shows how to synthesize a virtual view based on a set of optimal correspondences.

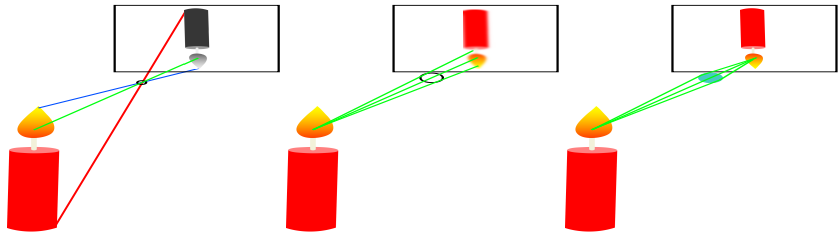
2.1 The pinhole camera model

To capture an image, a camera projects the 3D world onto a 2D plane, called the *image plane*, on which photosensitive sensors (generally CCD or CMOS) are located. By measuring the energy of the light that is either generated or reflected by the observed 3D scene, these sensors amass the photometric information composing the image. Assuming that the propagation of the light is rectilinear, it can be represented as a set of light rays, each one starting at a 3D world coordinate and passing through the 3D coordinate of one of these photosensitive sensors. This thesis focuses on *pinhole* cameras, meaning that it considers that all of these light rays converge through a common single 3D point, called the *optical center* of the camera. As mentionned by its name, such camera can be obtained by drilling a small hole on one side of a light-proof box. As illustrated in Figure 2.2, the light rays coming from a 3D scene pass through this single point and project an inversed image on the opposite of the box. However, the reversed images observed by such a simple device, imagined approximatively 300 years before Christ[1], are either dark or unsharp. This trade-off is relative to the size of the hole, known as the *aperture size*.

For a fixed distance between the image plane and the hole, *i.e.*, a fixed *focal length*, reducing the *aperture size* allows less light rays to enter in the light-proof box, resulting in a darker image, as illustrated in Figure 2.2(a). Nowadays, the choice of aperture size is still one of the crucial parameters when taking pictures, because the less entering light, the smaller is the SNR of the image[2].

[1] At this time, the device was used to project the 3D world onto a plane, and the projected contours were manually highlighted to draw the image.

[2] For a fixed focal length, a fixed sensibility of the photometric sensors, expressed in *ISO* units, and a fixed shutter speed.



(a) Small aperture (b) Large aperture (c) With lens

Figure 2.2: *Pinhole* cameras (see Section 2.1.1) use a lens to capture more collimated light rays, allowing brighter and sharper images, at the price of chromatic and geometric abberations (see Section 2.1.2).

At the opposite, increasing the aperture size lets enter more uncollimated light (unparallel rays), which increases the blurryness of the image, as illustrated in Figure 2.2(b). To reduce the effect of this compromise, lenses have been added in front of the aperture. As illustrated in Figure 2.2(c), convex lenses have the advantage to concentrate the light rays into a unique point, making them more collimated, while capturing the same amount of light. However, they practically generate some colorimetric and geometric distortions of the observed scene, which have to be estimated, to be compensated. The rest of the section introduces the basic notions of multi-view camera geometry, such as presented in [92], by defining independently the mathematical model of a pinhole camera, and the lens distortion compensation.

2.1.1 Projective camera parameters and calibration

This section details how an arbitrary 3D point projects onto a 2D camera image. As done all along this thesis, it refers to 3D coordinates by capital letters, while 2D coordinates are defined by lower case letters. The 3D coordinates are defined with respect to a reference orthonormal basis, centered on the *optical center* $\mathbf{C} \in \mathbb{R}^3$ of the camera. As illustrated in Figure 2.3, the Z axis of this basis is chosen aligned with the camera's viewing direction, defined by the *camera optical axis*. The 2D image coordinates are first defined with respect to a 2D basis, called the *image's referencial*. As illustrated on the left side of Figure 2.3, the origin of this basis is centered on the image *principal point* $\mathbf{p}$, defined by the intersection of the camera optical axis with the image's plane, considered located at a focal length $Z = f$. The image's abscissa u is defined to be parallel to the X axis of the 3D basis, while its ordinate v is defined to be parallel to its Y axis. In this configuration, the central projection from the 3D point $\mathbf{X}_{\text{CAM}} = (X_{\text{CAM}}, Y_{\text{CAM}}, Z_{\text{CAM}})^\top \in \mathbb{R}^3$, onto the image plane, considered located at a focal length $Z = f$, is described by the mapping $(X_{\text{CAM}}, Y_{\text{CAM}}, Z_{\text{CAM}})^\top \mapsto (fX_{\text{CAM}}/Z_{\text{CAM}},\ fY_{\text{CAM}}/Z_{\text{CAM}},\ f)^\top$.

The non-linear mapping from the 3D coordinates, expressed in the camera's coordinate system, to the image's coordinates, *i.e.*, $(X_{\text{CAM}}, Y_{\text{CAM}}, Z_{\text{CAM}})^\top \mapsto (fX_{\text{CAM}}/Z_{\text{CAM}},\ fY_{\text{CAM}}/Z_{\text{CAM}})^\top$, can be linearly expressed by adding an extra dimension to the coordinates.

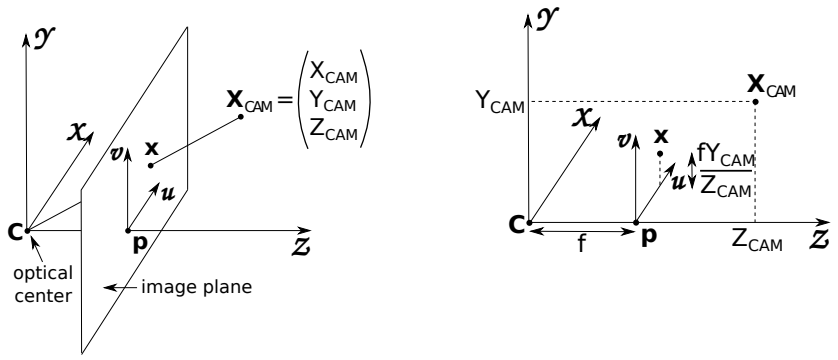


Figure 2.3: Central projection of a 3D point, expressed with respect to the camera's optical center, onto the image plane.

This procedure, called the *projective completion*, defines an equivalence between the point $\mathbf{X} = (x_1, x_2, \cdots, x_N)^\top \in \mathbb{R}^N$ and the set of points $\widetilde{\mathbf{X}} = (x_{N+1} \cdot \mathbf{X},\ x_{N+1})^\top \in \mathbb{R}^{N+1}$ for any $x_{N+1} \in \mathbb{R}^*$, called the *homogeneous coordinates* of $\mathbf{X}$. Given an arbitrary *homogeneous* vector $\widetilde{\mathbf{X}} = (x_1, x_2, \cdots, x_N, x_{N+1})^\top$, its equivalent point of dimension N can be recovered by dividing $\widetilde{\mathbf{X}}$ by its last coordinate, in such a way to transform it into its *inhomogeneous coordinates* $\mathbf{X} = (x_1/x_{N+1},\ x_2/x_{N+1},\ \cdots,\ x_N/x_{N+1})^\top$. For example, the Cartesian point $(1,2)^\top$ can be equivalently represented in homogeneous coordinates as $(1,2,1)$ or $(2,4,2)$. Adding $x_{N+1} = 0$ in the admissible set of values for x_{N+1} creates a singularity in the equivalence class, in the form of additional points called *points at infinity*. Such points complete the N dimensional space in such a way that any pair of lines always crosses at one point, which is called a *point at infinity* (or *vanishing point*) if the lines are parallel. The projective completion extends the Euclidean plane $\mathbb{R}^2$ to a projective plane [63], which enables to model, for example, the fact that a railway track, composed of two parallel rails, appears to cross at infinity. This projective completion also enables to express the (non-linear) transformation from a 3D coordinate (defined with respect to the camera's referencial) to the 2D homogeneous image plane coordinates $(X_{\text{CAM}}, Y_{\text{CAM}}, Z_{\text{CAM}})^\top \mapsto (fX_{\text{CAM}}/Z_{\text{CAM}}, fY_{\text{CAM}}/Z_{\text{CAM}}, 1)^\top$ as a linear operation:

$$\underbrace{\widetilde{\mathbf{X}}_{\text{CAM}} = \begin{pmatrix} X_{\text{CAM}} \\ Y_{\text{CAM}} \\ Z_{\text{CAM}} \\ 1 \end{pmatrix}}_{\text{3D homogeneous camera coordinates}} \quad \mapsto \underbrace{\begin{pmatrix} fX_{\text{CAM}} \\ fY_{\text{CAM}} \\ Z_{\text{CAM}} \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_{\text{CAM}} \\ Y_{\text{CAM}} \\ Z_{\text{CAM}} \\ 1 \end{pmatrix}}_{\text{2D homogeneous image's coordinates}}. \tag{2.1}$$

Equation (2.1) considers that the origin of the image's referencial is at the principal point $\mathbf{p} = (p_u, p_v) \in \mathbb{R}^2$ (see Figure 2.3), while conventions generally refers to the upper-left image's corner as the origin $(0,0)$.

To translate the origin to the upper-left corner, the mapping is transformed into:

$$\widetilde{\mathbf{X}}_{\text{CAM}} = \begin{pmatrix} X_{\text{CAM}} \\ Y_{\text{CAM}} \\ Z_{\text{CAM}} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX_{\text{CAM}} + Z_{\text{CAM}} \cdot p_u \\ fY_{\text{CAM}} + Z_{\text{CAM}} \cdot p_v \\ Z_{\text{CAM}} \end{pmatrix} = \begin{pmatrix} f & 0 & p_u & 0 \\ 0 & f & p_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_{\text{CAM}} \\ Y_{\text{CAM}} \\ Z_{\text{CAM}} \\ 1 \end{pmatrix}.$$

Also, the previous model assumes that the images coordinates are Euclidean coordinates and that the image's axes (u, v) are orthonormal. In practice, the CMOS (or CCD) sensors constituting the unit of reference, *i.e.*, the pixel, might be non-squared. For this reason, the image's axes u and v are respectively normalized by the abscissa pixel density m_u and ordinate pixel density m_v, transforming the mapping into:

$$\widetilde{\mathbf{X}}_{\text{CAM}} \mapsto \begin{pmatrix} f \cdot m_u & s & p_u \cdot m_u & 0 \\ 0 & f \cdot m_v & p_v \cdot m_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \widetilde{\mathbf{X}}_{\text{CAM}} \triangleq \begin{pmatrix} \alpha_u & s & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \widetilde{\mathbf{X}}_{\text{CAM}}, \tag{2.2}$$

where s is the skewing factor of the image's axes, allowing to generalize the sensor from rectangular to parallelogram shapes. Because the last column of the matrix is null, the added dimension in the homogeneous coordinates $\widetilde{\mathbf{X}}$ is not considered in the matrix multiplication, meaning that Equation (2.2) can be written in inhomogeneous coordinates:

$$\underbrace{\mathbf{X}_{\text{CAM}}}_{\text{3D inhomogeneous camera coordinates}} \mapsto \underbrace{\begin{pmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{X}_{\text{CAM}} \triangleq \mathbf{K}\mathbf{X}_{\text{CAM}} = \widetilde{\mathbf{x}}}_{\text{2D homogeneous image's coordinates}}. \tag{2.3}$$

Historically, the 5 degrees of freedom matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ has been called the *intrinsic matrix* of a pinhole camera, because these parameters are invariant for a fixed CMOS or CCD camera's sensor and a fixed focal length. Nowadays, many camera objectives allow to zoom by changing the focal length, meaning that the f parameter is not invariant anymore. However, such objectives usually numerically return this value, allowing to automatically adapt the intrinsic matrix.

Finally, we relax the assumption that the origin of the 3D world is considered on the camera's optical center $\mathbf{C}$, and looking towards the Z direction. Imagine that the inhomogeneous 3D coordinates of a point $\mathbf{X}$ are expressed relatively to an arbitrary 3D basis defined at the origin $\mathbf{O} = (0, 0, 0)^\top$, and that $\mathbf{X}_{\text{CAM}}$ represents the same point in the previously defined camera basis. These two basis are related via a 3D rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ and 3D translation $\mathbf{C}$, such that $\mathbf{X}_{\text{CAM}} = \mathbf{R}(\mathbf{X} - \mathbf{C})$ with

$$\mathbf{R} = \mathbf{R}_Z \cdot \mathbf{R}_Y \cdot \mathbf{R}_X,$$

with

$$\mathbf{R}_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma_X) & -\sin(\gamma_X) \\ 0 & \sin(\gamma_X) & \cos(\gamma_X) \end{pmatrix}$$

$$\mathbf{R}_Y = \begin{pmatrix} \cos(\gamma_Y) & 0 & \sin(\gamma_Y) \\ 0 & 1 & 0 \\ -\sin(\gamma_Y) & 0 & \cos(\gamma_Y) \end{pmatrix}$$

$$\mathbf{R}_Z = \begin{pmatrix} \cos(\gamma_Z) & -\sin(\gamma_Z) & 0 \\ \sin(\gamma_Z) & \cos(\gamma_Z) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where γ_X, γ_Y and γ_Z represent respectively the yaw (about the $\mathcal{X}$ axis), pitch (about the $\mathcal{Y}$ axis) and roll (about the $\mathcal{Z}$ axis) Euler angles.

In homogeneous coordinates, the relation $\mathbf{X}_{\text{CAM}} = \mathbf{R}(\mathbf{X} - \mathbf{C})$ can be expressed as:

$$\widetilde{\mathbf{X}}_{\text{CAM}} = \begin{pmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0}^\top & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \triangleq \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \widetilde{\mathbf{X}},$$

which is equivalent, in inhomogeneous coordinates, to:

$$\mathbf{X}_{\text{CAM}} = [\mathbf{R} \mid \mathbf{t}]\,\widetilde{\mathbf{X}} = \mathbf{R}\,[\mathbf{I} \mid -\mathbf{C}]\,\widetilde{\mathbf{X}}, \tag{2.4}$$

where $[\mathbf{I} \mid -\mathbf{C}]$ denotes the column concatenation of the 3×3 identity matrix $\mathbf{I}$ with the inhomogeneous 3D world coordinates of the camera's optical center $\mathbf{C}$. The part $\mathbf{R}\,[\mathbf{I}|-\mathbf{C}]$ defines the *extrinsic camera parameters*, because it englobes the external parameters of the camera, *i.e.*, its orientation and translation with respect to an arbitrary 3D basis.

Putting Equations (2.3) and (2.4) together, the pinhole camera projection of an arbitrary 3D homogeneous world coordinate $\widetilde{\mathbf{X}}$ to the homogeneous image's coordinates $\widetilde{\mathbf{x}}$ is given by:

$$\boxed{\widetilde{\mathbf{x}} = \mathbf{K}\mathbf{X}_{\text{CAM}} = \mathbf{KR}\,[\mathbf{I} \mid -\mathbf{C}]\,\widetilde{\mathbf{X}} \triangleq \mathbf{P}\widetilde{\mathbf{X}}.} \tag{2.5}$$

The $\mathbf{P} \in \mathbb{R}^{3\times4}$ matrix, known as the camera's *projection matrix*, fully describes the *projective geometry* in between a 3D point $\mathbf{X}$ and the position of $\mathbf{x}$, *i.e.*, the 2D coordinate that represents this 3D point on the image plane of a camera. It has 11 degrees of freedom and requires thus (at least) 6 point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ to be determined[3], since each inhomogeneous image's coordinate $\mathbf{x}_i$ leads to two independent equations (one in the abscissa coordinate u, the other in the ordinate v). The solution is obtained by solving the linear system $\mathbf{P}\widetilde{\mathbf{X}}_i = \widetilde{\mathbf{x}}_i$ for the 6 homogeneous points correspondences. Note that these equations involve homogeneous coordinates, meaning that $\mathbf{P}\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{x}}_i$ may differ by a non-zero factor while still satisfying the equation. Precisely,

[3]The reader is referred to the 3D generalization of the Chasles' theorem [92] for the definition of the degenerated configurations, *e.g.*, induced when two selected points lie on a twisted cubic.

equivalent vectors always have the same orientation, although they might differ in amplitude, and their cross product must thus be equal to zero. If we denote $\mathbf{P}_i^\top$ as the i^{th} row of the projection matrix ($i \in \{1,2,3\}$), the projection matrix can be determined by imposing:

$$\mathbf{P}\widetilde{\mathbf{X}}_i \times \widetilde{\mathbf{x}}_i = \begin{pmatrix} \mathbf{P}_1^\top \widetilde{\mathbf{X}}_i \\ \mathbf{P}_2^\top \widetilde{\mathbf{X}}_i \\ \mathbf{P}_3^\top \widetilde{\mathbf{X}}_i \end{pmatrix} \times \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} = \begin{pmatrix} w_i \mathbf{P}_2^\top \widetilde{\mathbf{X}}_i - v_i \mathbf{P}_3^\top \widetilde{\mathbf{X}}_i \\ u_i \mathbf{P}_3^\top \widetilde{\mathbf{X}}_i - w_i \mathbf{P}_1^\top \widetilde{\mathbf{X}}_i \\ v_i \mathbf{P}_1^\top \widetilde{\mathbf{X}}_i - u_i \mathbf{P}_2^\top \widetilde{\mathbf{X}}_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{2.6}$$

and can thus be written as:

$$\begin{pmatrix} \mathbf{0}^\top & w_i \widetilde{\mathbf{X}}_i^\top & -v_i \widetilde{\mathbf{X}}_i^\top \\ -w_i \widetilde{\mathbf{X}}_i^\top & \mathbf{0}^\top & u_i \widetilde{\mathbf{X}}_i^\top \\ v_i \widetilde{\mathbf{X}}_i^\top & -u_i \widetilde{\mathbf{X}}_i^\top & \mathbf{0}^\top \end{pmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.7}$$

Because the matrix in Equation (2.7) is of rank 2, each correspondence $\widetilde{\mathbf{X}}_i \leftrightarrow \widetilde{\mathbf{x}}_i$ provides 2 constraints on the projection matrix, and an exact solution for $\mathbf{P}$ can thus be obtained by solving the linear system $\mathbf{A}\mathbf{p}_v = \mathbf{0}$, *i.e.*, by determining the right null-space of the matrix $\mathbf{A}$, with $\mathbf{A} \in \mathbb{R}^{11\times 12}$ denoting the row concatenation of the constraints provided by the 6 correspondences (one of the constraint must be ignored), and $\mathbf{p}_v = vec(\mathbf{P}^\top)$ where $vec(.) : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{mn\times 1}$ is the matrix vectorization operator. However, the exactness of the estimated projection matrix depends on the accuracy of the measured coordinates $\mathbf{X}_i$ and $\mathbf{x}_i$. Because these measures are error-prone, *e.g.*, due to the precision of the manually annotated coordinates (u_i, v_i), which is limited to the pixel level, a least-squares minimization of the algebraic error of the over-determined system $\mathbf{A}\mathbf{p}_v = \mathbf{0}$, with $\mathbf{A} \in \mathbb{R}^{2n\times 12}$ englobing the constraints imposed by the $n \geq 6$ correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, generally leads to a more accurate estimation of $\mathbf{p}_v$. To determine a unique solution to this ill-posed problem and to reject the trivial solution $\mathbf{p}_v = \mathbf{0}$, multiple constraints on $\mathbf{p}_v$ have been proposed [92]. The most famous one is the normalization constraint $\|\mathbf{p}_v\|_2 = 1$ with $\|.\|_2$ the ℓ_2 norm. In this case, the solution $\mathbf{p}_v$ is the unit singular vector corresponding to the smallest singular value in the SVD decomposition of the matrix $\mathbf{A}$.

Instead of minimizing $\|\mathbf{A}\mathbf{p}_v\|_2$, one may want to minimize the geometric error between the selected 2D image's coordinates and the projection of the corresponding 3D coordinates onto the image plane:

$$\underset{\mathbf{P}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| \widetilde{\mathbf{x}}_i - \frac{1}{\lambda_i} \cdot \mathbf{P}\widetilde{\mathbf{X}}_i \right\|_2^2, \tag{2.8}$$

where $\lambda_i \in \mathbb{R}$ are equal to $\lambda_i = \mathbf{P}_3^\top \widetilde{\mathbf{X}}_i$ to ensure the equality between the homogeneous terms, and make Equation (2.8) a non-linear least-squares problem. Its solution corresponds to the maximum likelihood of $\mathbf{P}$, if we consider that the measurement errors are normally distributed [92], and can be estimated using an iterative algorithm, such as Levenberg-Marquardt, after having centralized and normalized the root mean square (RMS) of the 2D (3D) data points to $\sqrt{2}$ (respectively $\sqrt{3}$) and after having estimated the matrix $\mathbf{P}$

that minimizes the algebraic error to use it as initial point in the iterative algorithm. The procedure to obtain the camera's projection matrix $\mathbf{P}$ is called *camera calibration*. Although the two previous calibration methods are the most well-known, multiple other procedures have been proposed in the literature, by using for example the line of sight in the image [215].

The estimation of a projection matrix $\mathbf{P}$ does not only relate any 3D coordinate $\mathbf{X}$ to its 2D representation $\mathbf{x}$, but also defines uniquely (almost) all the intrinsic and extrinsic parameters of the camera. Indeed, according to the Equation (2.5), the projection matrix $\mathbf{P}$ can be seen as the horizontal concatenation of two sub-matrices:

$$\mathbf{P} = [\mathbf{KR} \mid -\mathbf{KRC}] \triangleq [\mathbf{M} \mid -\mathbf{MC}] .$$

Based on the fact that $\mathbf{R}$ is an orthogonal matrix and $\mathbf{K}$ is upper-triangular, two pairs of solutions for $(\mathbf{K}, \mathbf{R})$ are given by the RQ-decomposition of $\mathbf{P}$. The two acceptable solutions come from the fact that we can always multiply the k^{th} column of $\mathbf{K}$ by -1 and the k^{th} row of $\mathbf{R}$ also by -1 without changing the product $\mathbf{M} = \mathbf{KR}$. This ambiguity is removed by imposing that the diagonal entries of $\mathbf{K}$ are positive, due to the positivity of their physical interpretation. Specifically, according to Equation (2.3), the first diagonal element is $\alpha_u = m_u \cdot f$, where $m_u > 0$ and $f > 0$. Two tuples of possible Euler's angles (γ_X, γ_Y and γ_Z) are associated to the rotation matrix $\mathbf{R}$, obtained by RQ-decomposition of $\mathbf{M}$, requiring the knowledge of an additional 3D information (for example, the approximate direction of the observed 3D scene) to isolate the correct one. Some internal parameters of the cameras are also not uniquely defined. Indeed, while Equation (2.3) shows that the camera's skewing factor s is uniquely determined based on the intrinsic matrix $\mathbf{K}$, there exists a dependency between the value of the focal f and the pixel density factors m_u and m_v. To obtain a unique solution, it is usual to assume a uniform pixel density along one of the axis of the sensor, *e.g.*, $m_u = 1$, or to fix the focal length f based on the characteristics of the mounted lens. Finally, the camera's optical center can either be determined by solving the linear system $-\mathbf{MC} = \left((\mathbf{P})_4^\top\right)^\top$, or based on the fact that its projection determines the origin of the 3D referencial:

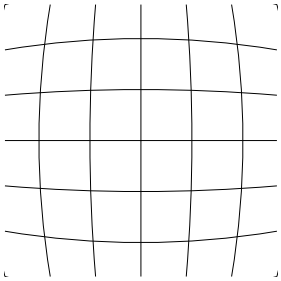
$$\mathbf{PC} = \mathbf{0}.$$

$\mathbf{C}$ is thus the right-null vector of $\mathbf{P}$, and can be obtained from its SVD decomposition.

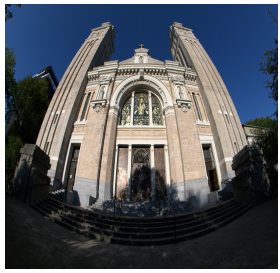
2.1.2 Lens distortion compensation

In the previous section, we have assumed the linearity of the light rays starting from an arbitrary 3D point and going through the camera's optical center. However, as illustrated in Figure 2.2(c), this assumption does not hold when the pinhole camera uses a lens, meaning that the projection of a straight line is not always represented as a straight line onto the image plane. One of the most important optical abberation is called the *radial distortion*, which can be classified either in *barrel distortion* (see Figure 2.4(a)) or in *pincushion distortion*

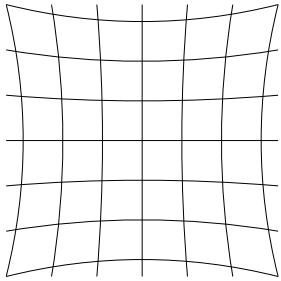
(see Figure 2.4(c)) and transform a straight line into a curved one. In practice, such kinds of distortions, as illustrated in Figure 2.4(b), are prominant in low-price and/or wide-angle (small focal length) lenses.



(a) Barrel distortion



(b) Barrel example



(c) Pincushion distortion

Figure 2.4: The projective geometry of a camera is disturbed by geometric abberations of the lens, which should be cancelled.

Because of the lens central symmetry, the distortion is uniform along the circles centered at the *lens center's point*, defined by the intersection of the camera optical axis with the lens surface. These distortions are thus expressed as a function of the radial ℓ_2 distance with respect to the lens center's point, *i.e.*, relatively[4] to $r = \sqrt{(u-u_c)^2 + (v-v_c)^2}$. If there were no distortion, an ideal lens should uniformely transform these circular level curves into other circular level curves. The scaling factor of this transformation is called the *lens magnification factor*, and is constant along r when there is no distortion. In barrel distortion, the lens magnification decreases with the distance from the lens center's point, while this magnification increases in pincushion distortion. The inverse magnification of these equipotentials is denoted by $L(r) : \mathbf{R}^+ \mapsto \mathbf{R}$, and defines the transformation of the pixel coordinates (u, v) with respect to its distance from the lens center's point. Hence, given the coordinates (u_c, v_c) of the lens center's point, the undistorted coordinates $(\hat{u}, \hat{v})$ are given by:

$$\hat{u} = u_c + L(r) \cdot (u - u_c) \qquad \hat{v} = v_c + \left(\frac{m_u}{m_v}\right)^{-1} \cdot L(r) \cdot (v - v_c)$$

where $\frac{m_u}{m_v}$ is the aspect ratio of the captured image. By estimating (u_c, v_c) and $L(r)$, the captured image can be unwarped, *i.e.*, its pixel coordinates can be transformed, in such a way to compensate those distortions.

Instead of determining the arbitrary analytical expression of $L(r)$, $L(r)$ is approximated by its Taylor k-th order expansion:

$$L(r) \approx L(0) + \kappa_1 r + \kappa_2 r^2 + \cdots + \kappa_k r^k,$$

in which $L(0) = 1$ to map the lens center's point onto the camera's principal point. The set of coefficients $\{\kappa_1, \kappa_2, \cdots, \kappa_k\}$ can either be estimated by

[4] Practically, the ℓ_2 distance with respect to the lens center's point $(u_c, v_c)^\top$ is normalized with respect to diagonal of the image, meaning that $r \in [0, 1]$.

including them in the minimization of the projection geometrical error (cfr. Equation (2.8)), or by maximizing iteratively a measure of the straightness of the corrected projection of 3D lines. In practice, only the first four coefficients $\{\kappa_1, \kappa_2, \kappa_3, \kappa_4\}$ are not negligeable ($> 10^{-6}$) even for low-quality lenses, and the estimation is restricted to these fourth ones. Lenses that produce a barrel distortion typically have $\kappa_1 < 0$, while the ones showing a pincushion effect have $\kappa_1 > 0$.

2.2 Multi-view geometry and 3D reconstruction

This section presents how calibrating (at least) two cameras imposes geometric dependencies between them, and how to exploit these dependencies to determine the scene's 3D. In this section, we assume that the observed 3D scene is composed only of Lambertian 3D surfaces, meaning that their surface's luminances are isotropic in all directions in the exterior half-space adjacent to the surface's tangent. Because the apparent brightness[5] of such a surface is the same regardless of the observer's angle of view, it is necessary that two observations of the same (infinitively small) 3D surface have the same brightness value. Section 2.3 shows how to relax this assumption by introducing some state-of-the-art image's representations that tend to describe geometrically and colorimetrically similar regions/pixels by a set of similar features, while representing dissimilar regions/pixels by dissimilar descriptors.

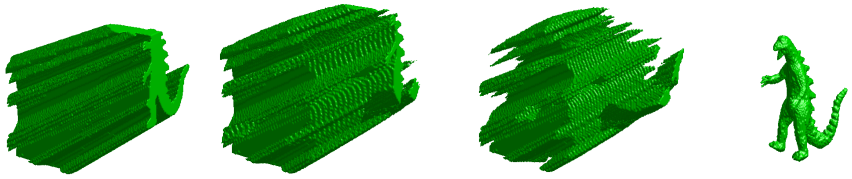
2.2.1 Space carving (3D to 2D)

Historically, the first steps of intermediate views synthesis have begun with the 3D reconstruction of a single object from a set of calibrated cameras. The first breakthrough method, called *space carving* [59], is based on the simple fact that the projection of the object's 3D surface is always included in the object's silhouette, such as observed by extracting the binary mask of the foreground object from the background in the reference images[6]. Complementarily, if a 3D coordinate does not project to the foreground mask of (at least) one view, it must belong to the 3D background. Instead of applying this *binary consistent test* on the set of all possible 3D coordinates $\mathbf{X} \in \mathbb{R}^3$, space carving methods discretize the 3D space, included in the convex-hull spanned by the cameras' optical centers, as a regular tesselation of non-overlapping cubes of predefined length, called *voxels* [50]. Each voxel is represented by the 3D coordinates of its centroid, that we denote $\mathbf{X}_i$ for the i^{th} voxel. A voxel for which any projection of its centroid $\widetilde{\mathbf{x}}_i^{(j)} = \mathbf{P}^{(j)}\widetilde{\mathbf{X}}_i$ on the j^{th} camera lies outside the foreground silhouette (or outside the image) of the j^{th} view is carved away, leaving only the voxels inside the 3D surface.

[5]The definition of the brightness has evolved gradually along the history, from the average of the (R,G,B) values to the *luminance* Y of the YUV color space, while passing through the *value* V of the HSV color space and the *lightness* L of the HSL color space.

[6]Generally, this foreground extraction is simplified by placing the 3D object in an environment with a controlled background, *e.g.*, in a green-screened studio, allowing to easily detect the background and thus its complementary foreground.

Figure 2.5 illustrates the space-carved 3D reconstruction from the Dino dataset [183] composed of 35 cameras uniformelly spanned along the 45° latitude of a sphere centered on a 3D figurine of a dinosaur.



(a) One view (b) Two small-baseline views (relative angle of 10°) (c) Two wide-baseline views (relative angle of 30°) (d) 35 views

Figure 2.5: N-views space-carving of a 4cm×4cm×16cm figurine discretized in voxels of approximatively 1.5mm×1.5mm×1.5mm.

Space-carving methods have four main limitations. As illustrated in Figure 2.5, the first important limitation is that space-carving requires a large density of cameras spreaded around the object to reconstruct, limiting the method to confined spaces. Matsuyama *et al.* [142] proposed, for example, to reconstruct the 3D volume of human beings captured in a 2m×2m×2m studio using 16 cameras and quantifing the 3D space into cubes of 1cm×1cm×1cm. The second main limitation is that space-carving cannot reconstruct the concavities of a 3D model. The third main limitation is that the accuracy of the 3D reconstruction directly depends upon the accuracy of the silhouette extraction and the fact that the foreground is entirely observed in all the views: any hole in a silhouette foreground mask will cause voxels to be carved away, drilling holes in the 3D object. As a consequence, a compact 3D model for which the foreground mask are noisy could be reconstructed as a set of unconnected 3D components, leading to a topologically incoherent reconstructed 3D model, for example human beings with their arms/legs/head detached from their body. Although surface smoothing methods [213], such as the marching cubes algorithm [133] [128], tend to reduce these discontinuities by fitting a continuous and smooth polygonal mesh [65] to the quantized voxel space, these methods do not use topological priors, others than the smoothness, meaning that they do not guarantee a topologically coherent final 3D model. To avoid that, Carranza *et al.* [31] have proposed to use a predefined 3D model of the object to reconstruct, and to determine its configuration by optimizing its degrees of freedom to maximize the overlap between the projection of the transformed 3D model and the 2D foreground silhouettes. Precisely, by determining, at time t, the optimal parameters of a 35 degrees 3D model of a human's actor[7], based on a grid-search around the optimal parameters obtained at time $t-1$, their method reconstructs a topologically coherent, but not necessary accurate, plausible 3D model of the human, captured in a controlled studio equiped with 7 calibrated cameras. Imposing such a strict temporal coherence, as exploited recently in [94], has the disavantage of propagating the

[7] *i.e.*, a fixed 3D human silhouette in which the rotation of the articulations (arms, legs, neck, etc.) can be modified.

error and being very sensitive to the initialization state, which generally imposes the model to be in a predefined configuration. The last main limitation of space-carving is its linear trade-off between computational time (or memory) and accuracy of the reconstruction with respect to the amount of voxels. To increase the accuracy[8] without increasing the number of tests, Seitz *et al.* [186] have proposed to replace the binary consistent test by a (necessary but not sufficient) *photo-consistency test*. They consider the richer color information (instead of a foreground mask) by defining a voxel $\mathbf{X}$ as occupied if it projects on pixels with similar brightness $\mathcal{I}$, *e.g.*, when $|\mathcal{I}\left(\mathbf{P}\widetilde{\mathbf{X}}\right) - \mathcal{I}\left(\mathbf{P}'\widetilde{\mathbf{X}}\right)| \leq \epsilon$ with an arbitrary but small threshold $\epsilon \in \mathbb{R}^+$ (when the photo-consistency test is evaluated in-between two cameras having respectively a projection matrix $\mathbf{P}$ and $\mathbf{P}'$). To speed up the algorithm, multi-resolution approaches based on octrees [210] that successively subdivide a 3D voxel if detected as occuped, have also been proposed. A more interesting approach has been the use of a *visibility constraint* [119]. This principle associate, to each pixel of a given camera, a binary value describing if a previously-tested voxel was projecting on this pixel, by testing the voxels $\mathbf{X}(\lambda)$ (expressed with respect to the camera's position) along the light ray

$$\widetilde{\mathbf{X}}(\lambda) = \mathbf{P}^+\widetilde{\mathbf{x}} + \lambda \cdot \widetilde{\mathbf{C}}, \tag{2.9}$$

where $\mathbf{P}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{P}$, *i.e.*, the matrix $\mathbf{P}^+ = \mathbf{P}^\top(\mathbf{P}\mathbf{P}^\top)^{-1}$ for which $\mathbf{P}\mathbf{P}^+ = \mathbf{I}$ in which $\mathbf{I}$ is the 3×3 identity matrix. With $\lambda \in \mathbb{R}^+$, starting from the ones which are the closest from the camera's view (smallest predefined λ, according to the whished quantization of the 3D space), there is no need to test for the projection, on this camera, of a next voxel further away on this light ray, if a previous voxel was already observed (*visibility* value set to one). Incorporating the 2D to 3D mappings $\mathbf{P}^+$ (instead of relying only on the 3D to 2D projections $\mathbf{P}$) in the 3D reconstruction, such as done by this last method, is the key principle underlying *triangulation* methods.

2.2.2 Triangulation (2D to 3D)

In the previous section, we have seen that space-carving methods investigate the coherence from the 3D world to the 2D image to estimate the 3D of a single object. Instead of a (computational) investigation of this 3D space, *triangulation* methods determine the 3D coordinates of a point lying on a 3D surface directly from the 2D coordinates of its representations in the two reference views.

The triangulation of a 3D point relies on Equation (2.9), which illustrates the fact that the 3D point $\mathbf{X}$ represented by the 2D (sub-)pixel $\mathbf{x}$ must belong to the light ray $\mathbf{X}(\lambda)$. The determination of this 3D point from one view is thus ill-posed, due to the fact that all the 3D points belonging to $\mathbf{X}(\lambda)$ project onto $\mathbf{x}$. Imagine now that this 3D point is observed with (at least) two cameras, having respectively the projection matrices $\mathbf{P}$ and $\mathbf{P}'$, and let $\mathbf{x}$ and $\mathbf{x}'$ denote the coordinates of these 2D observations. As long as these two views do not

[8] The accuracy of a 3D model can be, for example, defined by the Hausdorff distance between the reconstructed model and its ground-truth 3D measure.

share the same optical center, *i.e.*, $\mathbf{C} \neq \mathbf{C}'$, the 3D point $\mathbf{X}$ must belong to the intersection of the corresponding 3D light rays $\mathbf{X}(t)$ and $\mathbf{X}(t')$, in which $t \triangleq \frac{1}{\lambda}$ and $t' \triangleq \frac{1}{\lambda'}$:

$$\widetilde{\mathbf{X}}(t) = \widetilde{\mathbf{C}} + t \cdot \mathbf{P}^{+}\widetilde{\mathbf{x}} \qquad\qquad \widetilde{\mathbf{X}}(t') = \widetilde{\mathbf{C}}' + t' \cdot \mathbf{P}'^{+}\widetilde{\mathbf{x}}' \quad .$$

In general, because of the error-prone 3D to 2D calibration of $\mathbf{P}$ and $\mathbf{P}'$, the two 3D lines $\mathbf{X}(t)$ and $\mathbf{X}(t')$ might be skew. The closest 3D point to the two skew lines $\widetilde{\mathbf{X}}(t)$ and $\widetilde{\mathbf{X}}(t')$ is defined to be the center of the shortest segment separating the two lines, which is, by definition, orthogonal to these lines, and is thus given by:

$$\widetilde{\mathbf{X}} = \frac{\left(\widetilde{\mathbf{C}} + t^* \cdot \mathbf{P}^{+}\widetilde{\mathbf{x}}\right) + \left(\widetilde{\mathbf{C}}' + t'^* \cdot \mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)}{2},$$

in which the parameters t^* and t'^* are the ones that minimize the least squares distance separating the two light rays:

$$t^* = \frac{\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right) - \left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\cdot\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot(\mathbf{C}-\mathbf{C}')}{\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right) - \left(\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\right)^2}$$

$$t'^* = \frac{\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\cdot(\mathbf{C}-\mathbf{C}') - \left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\cdot\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot(\mathbf{C}-\mathbf{C}')}{\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right) - \left(\left(\mathbf{P}^{+}\widetilde{\mathbf{x}}\right)\cdot\left(\mathbf{P}'^{+}\widetilde{\mathbf{x}}'\right)\right)^2} \quad .$$

If the calibration is considered as noiseless, Hartley and Zisserman [92] have proposed another simple method, called the *linear triangulation method*, to approximate the 3D coordinate of a 3D point $\mathbf{X}$ observed at the 2D coordinates $\mathbf{x}$ and $\mathbf{x}'$ in the two reference views. It models the fact that the vector, defined by the homogeneous coordinates of $\widetilde{\mathbf{x}} = (u, v, 1)^\top$, must coincide with the light ray $\mathbf{P}\widetilde{\mathbf{X}}$, meaning that their cross-product $\widetilde{\mathbf{x}} \times \mathbf{P}\widetilde{\mathbf{X}}$ must be null (similarly, $\widetilde{\mathbf{x}}' \times \mathbf{P}'\widetilde{\mathbf{X}} = \mathbf{0}$). Because the homogeneous coordinates $\widetilde{\mathbf{X}}$ have 4 degrees of freedom, at least 4 of these constraints are required to determine $\widetilde{\mathbf{X}}$, which can be expressed as a linear system $\mathbf{A}\widetilde{\mathbf{X}} = \mathbf{0}$ (solved similarly to the one in Equation (2.6)), with for example:

$$\mathbf{A} = \begin{pmatrix} u\mathbf{P}_3^\top - \mathbf{P}_1^\top \\ v\mathbf{P}_3^\top - \mathbf{P}_2^\top \\ u'\mathbf{P}_3'^T - \mathbf{P}_1'^T \\ v'\mathbf{P}_3'^T - \mathbf{P}_2'^T \end{pmatrix}.$$

Visual hull

Visual hull methods also use the 2D to 3D back-projection, but determine the coordinates of the 3D points lying on a 3D surface simultaneously, as opposed to triangulation methods, which determine them independently. The concept of 3D visual hull was introduced by Laurentini [123]. The visual hull is defined as the smallest 3D convex-hull (in term of 3D volume) that, once projected on the reference camera views, fully overlaps the reference silhouettes. To determine the visual hull, the 2D to 3D mappings (cfr. Equation (2.9)) are used to back-project the shape of each 2D silhouette into the common 3D space.

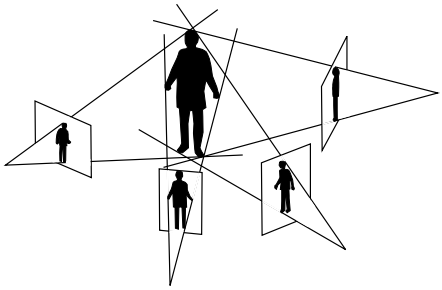


Figure 2.6: The intersection of the back-projections of the 2D foreground silhouettes forms the 3D visual hull [169].

As illustrated in Figure 2.6, each projection forms a visual cone encasing the 3D object and the intersection of these individual cones gives the approximated visual hull [77]. Although requiring much less projections than the space-carving methods, the computational bottleneck of the visual hull methods comes from the volumes intersections [193]. Matsuyama *et al.* [142] propose to restrain the back-projections of the silhouettes to the ground-plane, defined at $Z = 0$, and to reproject this ground-plane silhouette onto a set of predefined planes, or polyhedrons [145]. The computational estimation of the 3D intersections is thus simplified into the determination of planar cross-sections in the quantized 3D space. Visual hull methods also share three of their weaknesses with space carving methods, namely the (very) large number of views required to attain convincing rendering results [113], the inability to model concavities, and the dependence between the accuracy of the reconstructed model and the accuracy of the foreground silhouettes [112], easily corrupted for example by shadows [77]. To counter this last dependence, Guillemaut *et al.* [87, 86] have recently proposed to jointly optimize the foreground segmentation and its 3D reconstruction. Although their approach is leading to a more accurate 3D visual hull, it still requires a fairly large amount of 12 reference cameras. Stark *et al.* [201] reduce the amount of required cameras to 8 by imposing a spatio-temporal coherent 3D geometry during a post-processing step. However, due to the inaccuracy of the calibration of a wide-baseline camera setup when specifying 3D to 2D correspondences [84], the wide-baseline visual hull tends to produce topologically incoherent 3D models. This is due to the fact that a small calibration error generally produces a large error in the 3D reconstruction and that only a few intersecting cones define the 3D model. Given the (very) sparse amount of shape's prior insered in [201], imposing a spatio-temporal coherence between such topologically incoherent 3D models may also lead to a final topologically incoherent 3D model [43]. Matuski *et al.* [146] were the first ones to propose a method that performs the geometric computations in the 2D image space (instead of in the 3D space, that requires multiple error-prone projections, due to the inaccuracy of wide-baseline calibration). Their method, called *image-based visual hull*, relies on the simple fact that any 3D point $\mathbf{X}$ belonging to the visual hull's 3D surface and observed at the image point $\mathbf{x}$ must belong to the light ray $\mathbf{X}(\lambda)$ (cfr. Equation (2.9)), meaning that the projection $\widetilde{\mathbf{x}}' = \mathbf{P}'\widetilde{\mathbf{X}}$ of this 3D point onto another camera

view defined by the projection matrix $\mathbf{P}'$ must belong to the projection of the 3D light ray $\mathbf{l}' = \mathbf{P}'\widetilde{\mathbf{X}}(\lambda)$, as illustrated in Figure 2.7.

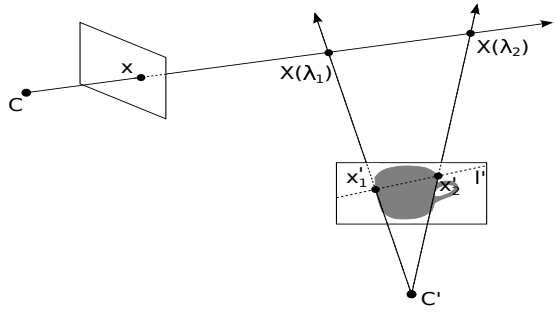


Figure 2.7: Image-based visual hull relies on the intersection between projected light rays and the reference foreground silhouettes to determine the 3D convex-hull of a given object.

For each pixel $\mathbf{x}_v$ in a virtual view (for which the calibration is detailed in Section 2.5.1), *i.e.*, a camera view that does not exist in reality, the associated light ray $\mathbf{X}_v(\lambda)$ is projected onto each of the reference images. If this projected light ray overlaps a foreground silhouette in a specific reference view, the overlapping segment represents the projection of the set of possible 3D points that lie in the visual hull and are represented by the pixel $\mathbf{x}$ in the virtual view. The depth (with respect to the image plane of the virtual camera) of all these 3D candidates, obtained separately on each reference view, are stored. Among these 3D points, the pixel $\mathbf{x}$ represents only the one that is the closest from the image plane of the virtual camera, because of the *visibility constraint* (cfr. Section 2.2.1), and its depth is associated to this pixel.

Image-based visual hull, and its color extension called image-based photo hulls [194], do not suffer from the computation complexity, limited resolution, or quantization artifacts of the previously presented volumetric approaches, while achieving lower average 3D errors than other visual hull methods [148]. This increase of performance comes from the fact that almost all the mentionned (error-prone) 2D to 3D projections can be avoided by exploiting the *epipolar geometry* that exists between the virtual view and each of the reference views, as explained in the next section.

2.2.3 Epipolar geometry

The *epipolar geometry* refers to the geometric constraint that associates, to a given 2D point $\mathbf{x}$ in a view, a restricted set of corresponding 2D points $\mathbf{x}'$ representing the 3D points that can be associated to $\mathbf{x}$. Such constraint is fundamental for general 3D reconstruction, due to the fact that the reconstruction of an arbitrary 3D scene, *e.g.*, composed of multiple objects, requires to determine pixel correspondences to reconstruct the 3D based on their triangulation (cfr. Section 2.2.2). As explained in the previous section, to each 2D image coordinates $\mathbf{x}$ in one view is associated a line in another view, which represents the projection, on this other view, of the light ray associated to the 2D point $\mathbf{x}$.

There exists thus a mapping $\mathbb{R}^3 \rightarrow \mathbb{R}^3 : \widetilde{\mathbf{x}} \mapsto \mathbf{l}'$, where $\mathbf{l}'$ is called the *epipolar line* of the 2D point $\mathbf{x}$. This mapping $\widetilde{\mathbf{x}} \mapsto \mathbf{l}'$ is represented by the fundamental matrix $\mathbf{F} \in \mathbb{R}^{3\times3}$, detailed in the next section.

Fundamental matrix

The *epipolar geometry*, *i.e.*, the projective geometry between two camera views, is fully represented by the 3×3 fundamental matrix $\mathbf{F}$, which maps a given 2D point $\mathbf{x}$ in one view onto a $1D$ line $\mathbf{l}'$, called an *epipolar line*, in the other view. The algebraic expression of this operator can be defined in two steps: first, Equation (2.9) is considered to express how a 2D coordinate $\mathbf{x}$ back-projects onto a light ray in the 3D space, and then, this 3D light ray is projected in the other view. Because the projective geometry conserves the lines, we focus on the projection of only two points of the 3D light ray onto the other view and determine the algebraic expression of the line $\mathbf{l}'$ passing through these two points. We arbitrary focus on the 3D points defined at $\lambda = \infty$ and $\lambda = 0$ in Equation (2.9). The first one corresponds to the 3D coordinates of the camera's optical center $\mathbf{C}$, and its projection $\widetilde{\mathbf{e}}' = \mathbf{P}'\widetilde{\mathbf{C}}$ is called the *epipole* of the second view. The projection of the second point $\mathbf{P}^+\widetilde{\mathbf{x}}$ onto the second view is given by $\mathbf{P}'\mathbf{P}^+\widetilde{\mathbf{x}}$, and the line joigning this 2D point with the first one (the epipole) is given by the homogeneous parameters of its normal vector $\mathbf{l}' = \widetilde{\mathbf{e}}' \times (\mathbf{P}'\mathbf{P}^+\widetilde{\mathbf{x}})$. By expressing this vector product by its skew-symetric matrix form, the line $\mathbf{l}'$ can be expressed as $\mathbf{l}' = [\widetilde{\mathbf{e}}']_\times \mathbf{P}'\mathbf{P}^+\widetilde{\mathbf{x}}$ with

$$[\widetilde{\mathbf{e}}']_\times = \begin{bmatrix} e'_1 \\ e'_2 \\ e'_3 \end{bmatrix}_\times = \begin{pmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{pmatrix}.$$

The fundamental matrix, which expresses the mapping $\mathbf{l}' = \mathbf{F}\widetilde{\mathbf{x}}$, is thus algebraically determined as

$$\mathbf{F} = [\widetilde{\mathbf{e}}']_\times \mathbf{P}'\mathbf{P}^+. \tag{2.10}$$

It is interesting to note that any epipolar line $\mathbf{l}'$ of the second view will always pass through the epipole $\mathbf{e}'$, due to the fact that any 3D light ray observed by the first camera will always pass through the camera's optical center $\mathbf{C}$, which is projected in the second view on $\mathbf{e}'$. The fundamental matrix $\mathbf{F}$ represents thus a mapping, in an inhomogeneous space, from $\mathbb{R}^2$ to the 1-dimensional pencil of lines passing through the epipole, and is thus of rank 2. Also, $\widetilde{\mathbf{e}}'^T$ is the left null-vector of $\mathbf{F}$, due to the fact that all epipolar lines $\mathbf{l}' = \mathbf{F} \cdot \widetilde{\mathbf{x}}$ intersect at the epipole $\mathbf{e}'$ and thus $\widetilde{\mathbf{e}}'^T\mathbf{l}' = \widetilde{\mathbf{e}}'^T (\mathbf{F} \cdot \widetilde{\mathbf{x}}) = \left(\widetilde{\mathbf{e}}'^T \cdot \mathbf{F}\right) \widetilde{\mathbf{x}} = 0$ for all $\widetilde{\mathbf{x}} \neq \widetilde{\mathbf{e}}$, imposing that $\widetilde{\mathbf{e}}'^T \cdot \mathbf{F} = \mathbf{0}$ [9].

Another very important property of the fundamental matrix concerns the fact that, if a 3D point $\mathbf{X}$ is imaged as $\mathbf{x}$ in the first view and as $\mathbf{x}'$ in the second view, $\mathbf{x}'$ must belong to $\mathbf{l}' = \mathbf{F}\widetilde{\mathbf{x}}$ and must thus satisfy $\widetilde{\mathbf{x}}'^T\mathbf{l}' = 0$, which defines the *fundamental relation* of the epipolar geometry:

$$\widetilde{\mathbf{x}}'^T\mathbf{F} \cdot \widetilde{\mathbf{x}} = 0. \tag{2.11}$$

[9] Similarly, $\mathbf{F} \cdot \widetilde{\mathbf{e}} = \mathbf{0}$ and $\widetilde{\mathbf{e}}$ if the right null-vector of $\mathbf{F}$

In other words, if $\mathbf{x}$ and $\mathbf{x}'$ represent the projections of the same 3D point $\mathbf{X}$, *i.e.*, $\mathbf{x}$ and $\mathbf{x}'$ are two *corresponding points*, they must satisfy the fundamental relation (2.11). If we want to find corresponding point to $\mathbf{x}$ in the second view, only the epipolar line $\mathbf{l}'$ needs to be searched, as opposed to the entire image. Reciprocally, the 2D point $\mathbf{x}$ must belong to the epipolar line $\mathbf{l}$ associated to the corresponding 2D point $\mathbf{x}'$ in the other view, imposing $\widetilde{\mathbf{x}}^\top \mathbf{l} = 0$ and thus

$$\widetilde{\mathbf{x}}^T \mathbf{F}' \widetilde{\mathbf{x}}' = 0. \tag{2.12}$$

By combining (2.11) and (2.12) together, we obtain

$$\mathbf{F}' = \mathbf{F}^\top. \tag{2.13}$$

Analogously to the fact that there exists correspondences between 2D coordinates $\mathbf{x} \leftrightarrow \mathbf{x}'$ in multi-view geometry, there also exists correspondences between the epipolar lines. Two epipolar lines are said to be *corresponding* if all the 2D coordinates belonging to the first epipolar line have correspondences that belong to the second one, and vice-versa, such as illustrated in Figure 2.8.

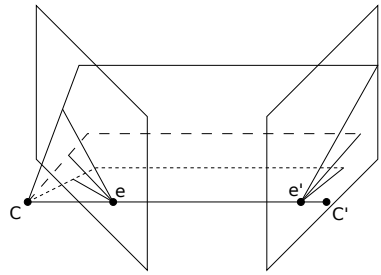


Figure 2.8: Epipolar lines are corresponding if all the 2D coordinates belonging to the first epipolar line have correspondences that belong to the second one, and vice-versa.

Although this definition relies on the knowledge of 2D correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$, the knowledge of the fundamental matrix is sufficient to determine the infinite set of corresponding epipolar lines. As a simple proof, let us imagine that the 2D coordinate $\mathbf{x}_1$ in a first reference view corresponds to $\mathbf{x}'_1$ in the second view. The epipolar constraint imposes that $\mathbf{x}'_1$ belongs to epipolar line $\mathbf{l}' = \mathbf{F} \cdot \widetilde{\mathbf{x}}_1$. Because this epipolar line passes through the epipole $\mathbf{e}'$ and the 2D coordinate $\mathbf{x}'_1$, its parametric equation is given by $\mathbf{l}'(\mu) = \widetilde{\mathbf{x}}'_1 + \mu \widetilde{\mathbf{e}}'$. Any 2D point $\mathbf{x}'_2$ belonging to $\mathbf{l}'$ corresponds to a specific value of μ, say $\mu^\star$, and can thus be expressed as $\widetilde{\mathbf{x}}'_2 = \widetilde{\mathbf{x}}'_1 + \mu^\star \widetilde{\mathbf{e}}'$. The correspondence of $\mathbf{x}'_2$ in the first view must belong to $\mathbf{F}^\top \widetilde{\mathbf{x}}'_2 = \mathbf{F}^\top \left(\widetilde{\mathbf{x}}'_1 + \mu^\star \widetilde{\mathbf{e}}' \right) = \mathbf{F}^\top \widetilde{\mathbf{x}}'_1 + \mathbf{F}^\top \widetilde{\mathbf{e}}' = \mathbf{F}^\top \widetilde{\mathbf{x}}'_1 = \mathbf{l}$. In summary, any 2D coordinate $\mathbf{x}'_2$ belonging to the line $\mathbf{l}'$ passing through an arbitrary 2D coordinate $\mathbf{x}'_1$ and the epipole $\mathbf{e}'$ (obtained as the right-null space of $\mathbf{F}^\top$) in the second image has a correspondence $\mathbf{x}_2$ that belongs to the epipolar line $\mathbf{l} = \mathbf{F}^\top \widetilde{\mathbf{x}}'_2$. By replacing $\mathbf{x}'_1$ with $\mathbf{x}_1$, $\mathbf{e}'$ with $\mathbf{e}$ and $\mathbf{F}$ with $\mathbf{F}^\top$ in the previous explanation, the reciprocal can be proved. Thus, by selecting any arbitrary coordinate $\mathbf{x}$, two corresponding lines $\mathbf{l} = \widetilde{\mathbf{x}} \times \widetilde{\mathbf{e}}$ and $\mathbf{l}' = \mathbf{F} \cdot \widetilde{\mathbf{x}}$ can be defined.

This thesis, and more generally stereo vision, extensively uses the three main advantages of the fundamental matrix, which are summarized here below:

- Given an image's coordinate $\mathbf{x}$, it restricts the set of possible correspondence $\mathbf{x}'$ in another view to the ones belonging to the corresponding epipolar line $\mathbf{l}' = \mathbf{F} \cdot \widetilde{\mathbf{x}}$. The general 2D search of correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$ is thus restricted to a $1D$ search along the corresponding epipolar lines.
- Given a set of correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$ approximated at the pixel level, the fundamental relation (Equation (2.11) or (2.12)) enables to refine these 2D coordinates to the sub-pixel level, by determination of a pair of corrected correspondences $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}}'$ that minimizes the objective function (cfr. the *golden triangulation method* in [92])

$$\begin{aligned} \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{x}}') = \quad & \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \left\|\mathbf{x}' - \hat{\mathbf{x}}'\right\|_2^2 \\ \text{subject to} \quad & \hat{\widetilde{\mathbf{x}}}'^T \mathbf{F} \hat{\widetilde{\mathbf{x}}} = 0 \end{aligned}$$

- The fundamental matrix fully describes the projective geometry in-between two (non-degenerated [92]) camera views, as does a pair of projection matrices, while its estimation requires much less efforts than calibrating $\mathbf{P}$ and $\mathbf{P}'$ [53].

The last point comes from the fact that the fundamental matrix can not only be estimated based on the projection matrices (as shown in Equation (2.10)), but also in terms of corresponding pairs of 2D image coordinates $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, which do not require to take 3D precise measurements, as it is the case for calibrating $\mathbf{P}$ and $\mathbf{P}'$. Indeed, the 7 degrees of freedom associated to the fundamental matrix $\mathbf{F}$ can be estimated based on (at least) 7 correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ satisfying the fundamental relation $\widetilde{\mathbf{x}}_i'^T \mathbf{F} \widetilde{\mathbf{x}}_i = 0$. If we denote $\mathbf{f}$ as the row-vectorization of $\mathbf{F}$, each of these constraint can be expressed as:

$$\begin{pmatrix} u'_i u_i & u'_i v_i & u'_i & v'_i u_i & v'_i v_i & v'_i & u_i & v_i & 1 \end{pmatrix} \mathbf{f} = 0,$$

and the concatenation of (at least) 7 correspondences can form a system $\mathbf{A}\mathbf{f} = 0$ which is sufficient to determine the rank-2 $\mathbf{F}$ matrix (up to a scaling factor), uniquely if $rank(\mathbf{A}) = 8$ or by imposing its singularity (null determinant) if $rank(\mathbf{A}) = 7$. However, if the 2D correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are corrupted by some noise, the rank of $\mathbf{A}$ might be equal to 9, implying that $\mathbf{f}$ must be estimated as a least-squared solution (*e.g.*, by a rank-2 approximation of singular vector corresponding to the smallest singular value in the SVD decomposition of $\mathbf{A}$ [92]).

Projective grid space (2D to 2D)

The main advantage of the fundamental matrix is that it constraints the 2D correspondences without explicitly considering the (error-prone) 3D information. For example, instead of applying a voxels occupancy test along a Euclidean 3D grid that requires a precise calibration of the projection matrices, the occupancy test can be investigated at the 2D pixel level. Indeed, Saito *et*

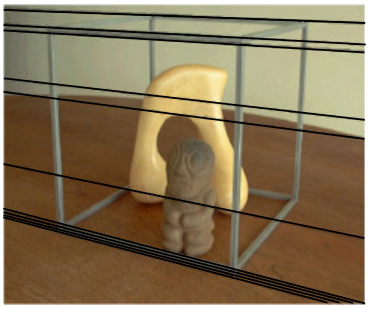
al. [177] have proposed to rely only on the fundamental matrices relating N camera views (with $N \geq 2$) to express the 3D coordinates directly in the image space and reconstruct a visual-hull of the observed 3D model. Analogously to Matuski *et al.* [146] who associate a 3D light ray to each pixel $\mathbf{x}$ and enquire the voxels' occupancies along this light ray (cfr. Section 2.2.2), [177] associates a 2D epipolar line $\mathbf{l}' = \left(l'_1, l'_2, l'_3\right)^\top = \mathbf{F} \cdot \widetilde{\mathbf{x}}$ to each pixel and enquires the voxels' occupancies along this 2D ray. Each voxel is thus not anymore defined based on the triangulation (cfr. Section 2.2.2) of two correspondences, *i.e.*, parametrized with respect to $(\mathbf{x}, \mathbf{x}', \mathbf{P}, \mathbf{P}')$, but is implicitly defined by the set of parameters $(\mathbf{x}, u')$ while knowing $\mathbf{F}$, where u' corresponds to the horizontal position of the pixel $\mathbf{x}'$ along the corresponding epipolar line[10]. Two of the N views are thus chosen to define the voxels' parametrizations $(\mathbf{x}, u', \mathbf{F})$ on the projective grid. The pixel representing the projection of the voxel $(\mathbf{x}, u')$ in any other reference view (if $N > 2$) can be found based on the fact that this pixel must belong to both the epipolar line $\mathbf{l}^{(i)} = \mathbf{F}^{(i)}\widetilde{\mathbf{x}}$ and $\mathbf{l}'^{(i)} = \mathbf{F}'^{(i)}\left(u', v', 1\right)^\top$, where $\mathbf{F}^{(i)}$ (respectively $\mathbf{F}'^{(i)}$) represents the predefined fundamental matrix from the first (respectively second) reference view to the i^{th} reference view (with $i > 2$). As done in [146], the voxel defined by $(\mathbf{x}, u')$ is considered as occupied if its projection belongs to all the object's foreground silhouettes in the camera views that could see this voxel, such as detailed in [118]. Although this method only requires a *weak calibration*, *i.e.*, the knowledge of the fundamental matrices linking the reference views, its efficiency still depends directly on the efficiency of the foreground extraction method, limiting the method to a controlled environment. Also, the projective grid method still requires an important amount of cameras (from 16 [241] to 51 [177]) to ensure the accuracy of the obtained 3D surface.

Epipolar rectification

Practically, any investigation along the corresponding epipolar lines, such as done in projective grid space, requires to define a sampling step. Because the orientation of the epipolar lines varies in function of $\mathbf{x}$, any constant sampling step might make the investigation fall in-between pixel entities. To avoid the computational overload of a *sub-pixel* interpolation at each investigation, the *epipolar rectification* is a pre-process that transforms the two stereo images in such a way that all their epipolar lines become parallel to the u axis (image's abscissa axis) of the image and that corresponding epipolar lines (cfr. Section 2.2.3) share the same v coordinate, as highlighted by the black lines in Figure 2.9. As a consequence, disparities between the images are only in the u-direction, *i.e.*, there is no v-disparity. These transformations make the image planes coplanar and in addition parallel to the baseline, simulating a pair of identical cameras placed side-by-side with their principal axes parallel.

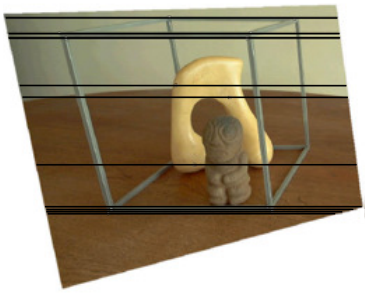
The general 2D correspondence investigation $\mathbf{x} = (u,\ v)^\top \leftrightarrow \mathbf{x}' = (u',\ v')^\top$ is thus simplified into a $1D$ research of correspondences $\mathbf{x}_R = (u_R,\ v_R)^\top \leftrightarrow \mathbf{x}'_R = (u'_R,\ v_R)^\top$ along the *corresponding rectified epipolar lines* having the same v-ordinate.

[10] Note that because any possible correspondence $\mathbf{x}'$ belongs, by definition, to the epipolar line $\mathbf{l}' = \mathbf{F} \cdot \widetilde{\mathbf{x}}$, its vertical coordinate is defined by $v' = -\left(l'_1 \cdot u' + l'_3\right)/l'_2$ to satisfy $\mathbf{l}'^\top \cdot \widetilde{\mathbf{x}}' = 0$.



(a) Original left view

(b) Original right view



(c) Epipolarly rectified left view

(d) Epipolarly rectified right view

Figure 2.9: Epipolar rectification transforms two arbitrary views into a pair of identical cameras, separated by a simple translation along the image's u-axis.

To estimate the general 2D projective image transformation $\mathbf{H} \in \mathbb{R}^{3\times3}$ (with $\mathbf{H}$ being a non-singular matrix) that makes the epipolar lines parallel, it is important to remember two points:

- Epipolar lines always cross at the epipole.
- Parallel lines cross at *a point at infinity*, defined by the homogeneous coordinates $(u, v, 0)^\top$ (cfr. Section 2.1.1).

Specifically, let us consider two transformed epipolar lines being parallel to the u-axis, and arbitrary defined at $v = c_1$ (with $c_1 \in \mathbb{R}$) and $v = c_2$ (with $c_2 \in \mathbb{R}$). Their inhomogeneous coordinates are thus given by $\mathbf{l}_1 = (0,\ 1,\ -1/c_1)^\top$ for the first line and $\mathbf{l}_2 = (0,\ 1,\ -1/c_2)^\top$ for the second one. They intersect at the 2D point defined, in homogeneous coordinates, by $\mathbf{l}_1 \times \mathbf{l}_2 = ((1/c_1 - 1/c_2)\,,\ 0,\ 0)^\top \triangleq (c,\ 0,\ 0)^\top$ with $c \in \mathbb{R}$.

The projective transformation $\mathbf{H}$ (respectively $\mathbf{H}'$ for the second image) must thus send the epipole $\mathbf{e}$ (respectively $\mathbf{e}'$) on the infinite point $(c,\ 0,\ 0)^\top$ in such a way to make them parallel, with c being arbitrary chosen in $\mathbb{R}$, due to the fact that $(c,\ 0,\ 0)^\top$ always represents the same 2D inhomogeneous point. The 8 degrees of freedom $\mathbf{H}$ must thus satisfy $\mathbf{H}\widetilde{\mathbf{e}} = (1,\ 0,\ 0)^\top$ (respectively, $\mathbf{H}'$ must satisfy $\mathbf{H}'\widetilde{\mathbf{e}}' = (1,\ 0,\ 0)^\top$) $\forall\, \widetilde{\mathbf{e}} \in \mathbb{R}^3$. Both the non-singular and this

last constraint lead to a transformation $\mathbf{H}$ with 4 degrees of freedom, meaning that determining $\mathbf{H}$ and $\mathbf{H}'$ independently is a 8 degrees of freedom problem. However, as it is the case for any stereo views, the rectified image coordinates $\widetilde{\mathbf{x}}_R = \mathbf{H}\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{x}}'_R = \mathbf{H}'\widetilde{\mathbf{x}}'$ must also fulfil the fundamental relation of epipolar geometry based on the rectified fundamental matrix $\mathbf{F}_R$:

$$\widetilde{\mathbf{x}}_R'^{\top}\mathbf{F}_R\widetilde{\mathbf{x}}_R = 0 \tag{2.14}$$

$$\left(u',\ v,\ 1\right)\ \mathbf{F}_R\ \left(u,\ v,\ 1\right)^{\top} = 0$$

for all u, v and u', which is only valid when

$$\mathbf{F}_R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}.$$

Based on the fundamental relation in the original image space,

$$\begin{aligned} \widetilde{\mathbf{x}}'^{\top}\mathbf{F}\widetilde{\mathbf{x}} &= 0, \\ \left(\mathbf{H}'^{\,-1}\widetilde{\mathbf{x}}'_R\right)^{\top}\mathbf{F}\left(\mathbf{H}^{-1}\widetilde{\mathbf{x}}'_R\right) &= 0, \\ \widetilde{\mathbf{x}}_R^{\top}\mathbf{H}'^{\,-\top}\mathbf{F}\mathbf{H}^{-1}\widetilde{\mathbf{x}}_R &= 0, \end{aligned}$$

which leads, based on Equation (2.14), to 2 additional constraints on the pair $(\mathbf{H}, \mathbf{H}')$,

$$\mathbf{H}^{-T\prime}\mathbf{F}_R\mathbf{H}^{-1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}.$$

The determination of the optimal set of transformations $(\mathbf{H}, \mathbf{H}')$ is thus a 6 degrees of freedom unconstrained problem, for which the solution is not unique. These 6 degrees of freedom are generally estimated by minimizing a measure of the distortions, or relative distortion, encountered by the two rectified images [141]. For example, the authors of [93] fix $\mathbf{H}$ to be a first-order approximation of a rigid transform around the image's origin and determine $\mathbf{H}'$ such as to minimize the u-disparity, *i.e.*, the distortion $\sum_{i=1}^{N}\left\|\mathbf{H}\widetilde{\mathbf{x}}_i - \mathbf{H}'\widetilde{\mathbf{x}}'_i\right\|_2^2$ of N manually selected corresponding points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. In [132], the homographies $\mathbf{H}$ and $\mathbf{H}'$ are decomposed into three successive operations: a specific projective transform, a similarity transform, and a shearing transform. Not only these successive operations generalize the transformations to an affine transformation, but the authors also use the additional degrees of freedom in the affine component to further reduce the distortions. The authors of [139] follow a similar idea by minimizing the Jacobian of an affine approximation of $\mathbf{H}$, that they interpret as the "creation and loss of pixels" when this approximation is applied. Fusiello *et al.* [69] estimate that any Euclidean epipolar transformation, obtained based on the knowledge of the cameras' fundamental matrix, overcomes the other types of transformations. Given the lack of knowledge of the intrinsic parameters, the Euclidean geometry can only be approximated, this is why they call their method *Quasi-Euclidean Uncalibrated*

Epipolar Rectification [11]. The approximation is done by non-linear minimization of a 6-parameters measure of the image's distortions.

Because none of the previous methods is clearly superior to the others in term of 2D v-discrepancy between rectified corresponding pixels, and because no consensus has been achieved on the optimal distortion criterion, this thesis follows the work of [156], who iteratively minimizes an intuitive measure based on the required rotations to align its epipolar lines. Their method can be decomposed into three main rotation steps:

1. Both the epipoles $\mathbf{e}$ and $\mathbf{e}'$ are send to infinity, and their v coordinates depend on the applied rotation, which is chosen as having a minimal rotation angle.

2. The two cameras are rotated so that both $[e_u\ e_v\ 0]^\top$ and $[e'_u\ e'_v\ 0]^\top$ are sent to $[1\ 0\ 0]$.

3. Both cameras are finally rotated around their baseline to compensate for their residual relative rotation.

This method has been chosen not only for its low rectification error[12], but more importantly because this error is almost constant whatever the configuration of the stereo pair (wide to narrow-baseline), while others [139] [93] tend to fail in wide-baseline cases. This advantage is paid at the price of restricting the usage of the method to cases for which the epipoles are outside the image domain, instead of treating arbitrary camera geometry as done in [170]. However, wide-baseline configurations tend to follow this geometry [52].

Reciprocity between fundamental and projective calibration

While the epipolar geometry fully describes the projective geometry in-between two views, their respective projection matrices are often directly required, *e.g.*, for the interpolation of the parameters of a (virtual) intermediate camera travelling smoothly in-between the two reference cameras (see Section 2.5.1). Fortunately, the projection matrices of the cameras can be retrieved from $\mathbf{F}$ up to a projective transformation of the 3D space. Indeed, compared to the fundamental matrix, which only describes the relations about the image coordinates, the projection matrices $\mathbf{P}$ and $\mathbf{P}'$ also englobe the chosen position for the 3D world basis (cfr. Equation (2.4)) and thus the Euclidean geometry of the 3D scene. In other words, the fundamental matrix $\mathbf{F}$ associated to any projective transformation (called *homography*, cfr. Section 2.2.4) $\mathbf{H} \in \mathbb{R}^{4\times4}$ of the 3D space, resulting in projection matrices $\mathbf{PH}$ and $\mathbf{P'H}$, is the same than the one corresponding to the camera pair $(\mathbf{P}, \mathbf{P}')$. Thus, although Equation (2.10) shows that a camera pair uniquely defines a fundamental matrix, the inverse is not true. To raise this ambiguity, it is common to fix the first view as the reference basis $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$. In this case, the pair $(\mathbf{P}, \mathbf{P}')$ is said to be expressed

[11] Note that the projective space can be upgraded to a metric one, *i.e.*, a Euclidean one, based on self-calibration [228].

[12] The rectification error is measured as the average and standard deviation of the v-disparity of rectified correspondences.

in its *canonical form*, and to form a *stereo rig*. As proved in [136], a good choice of stereo rig associated to the fundamental matrix $\mathbf{F}$ is given by:

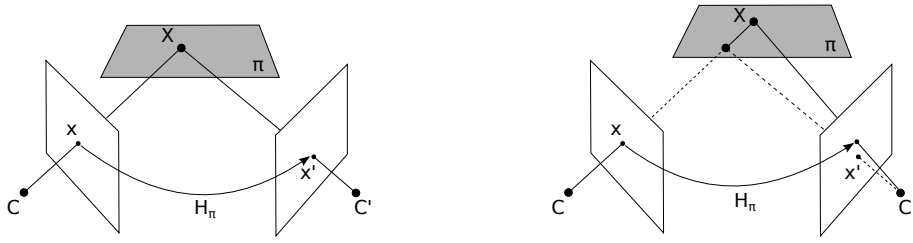
$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \qquad \mathbf{P}' = [[\widetilde{\mathbf{e}}']_{\times}\mathbf{F} \mid \widetilde{\mathbf{e}}'], \tag{2.15}$$

with $\mathbf{e}'$ the epipole of the second view.

2.2.4 Homographies

Given any fundamental matrix $\mathbf{F}$ relating two views, the previous section has shown how to determine their relative projection matrices $\mathbf{P}$ and $\mathbf{P}'$, which enable to constrain the investigation of correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$. Until now, such investigation has been considered independently for all $\mathbf{x} \in \mathbf{\Omega}_{\mathcal{I}}$, with $\mathbf{\Omega}_{\mathcal{I}}$ being the set of pixel's coordinates of the image $\mathcal{I}$ captured by the first reference view. However, because 3D structures generally impose local dependencies between the neighboring 3D points, the depth associated to two neighboring 2D coordinates might be highly correlated. This section considers planar models as underlying 3D structures, and how this assumption enables to estimate simultaneously the depth associated to a set of 2D coordinates.

The projective transformation from a 2D coordinate $\mathbf{x}$ in one view to another 2D coordinate $\mathbf{x}'$ in another view can always be described [52] [68] by a homography transformation $\mathbf{H} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, such that $\widetilde{\mathbf{x}}' = \mathbf{H}\widetilde{\mathbf{x}}$. As illustrated in Figure 2.10(a), any homography transformation $\mathbf{H}$ can be interpreted as the projection (called *transfer*) of the 2D point $\mathbf{x}$, via a 3D plane $ax + by + cz + d = 0$, onto the second image. For this reason, we denote the homography transfer of $\mathbf{x}$ onto $\mathbf{x}'$, via the 3D plane of homogeneous normal vector $\boldsymbol{\pi} = [a\ b\ c\ d]^{\top}$, by $\mathbf{H}_{\boldsymbol{\pi}}$.



(a) 3D plane well estimated (b) 3D plane wrongly estimated

Figure 2.10: Any point $\mathbf{x}$ will project on its corresponding point $\mathbf{x}'$ if the 3D homography approximating the (planary) 3D structure around $\mathbf{X}$ is correctly estimated.

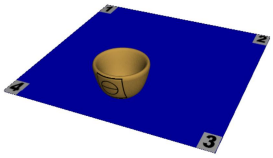
Reciprocally, as illustrated in Figure 2.10(b), if $\mathbf{x}$ is the projection, onto the first reference view, of a given 3D point $\mathbf{X}$ that does not belong to the 3D plane $\boldsymbol{\pi}$, the homographic transfer $\mathbf{H}_{\boldsymbol{\pi}}\widetilde{\mathbf{x}}$ will not be mapped onto $\mathbf{x}'$, defined as the projection of $\mathbf{X}$ onto the second reference view.

This principle can be extended to a set of points. For example, let us consider two 3D points $\mathbf{X}_1 \in \boldsymbol{\pi}$ and $\mathbf{X}_2 \notin \boldsymbol{\pi}$. Their projections onto the reference

views form respectively the sets $(\mathbf{x}_1, \mathbf{x}'_1)$ and $(\mathbf{x}_2, \mathbf{x}'_2)$, which must fulfil

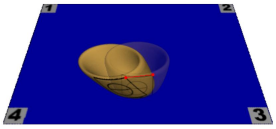
$$\widetilde{\mathbf{x}}'_1 = \mathbf{H}_{\boldsymbol{\pi}}\widetilde{\mathbf{x}}_1 \qquad\qquad \widetilde{\mathbf{x}}'_2 \neq \mathbf{H}_{\boldsymbol{\pi}}\widetilde{\mathbf{x}}_2.$$

In other words, $\mathbf{x}$ is correctly transferred onto its correspondence $\mathbf{x}'$ only if the intersection of the plane $\boldsymbol{\pi}$ with the 3D light ray associated to $\mathbf{x}$ perfectly falls onto $\mathbf{X}$, the 3D point projected at $\mathbf{x}$ and $\mathbf{x}'$ onto the reference views. Roughly speaking, we say that the 3D plane $\boldsymbol{\pi}$ correctly approximates a set of N 3D points $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N$ if $\mathbf{H}_{\boldsymbol{\pi}}$ correctly transfers $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ onto $\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_N$. Figure 2.11 illustrates this principle with different views of a 3D cup lying on a 3D planar blue panel. Figure 2.11(c) represents the transfer of the first view (Figure 2.11(a)) onto the second view (Figure 2.11(b)) via the plane $\boldsymbol{\pi}$ induced by the 3D planar blue panel.



(a) First reference view

(b) Second reference view



(c) Projection of the first reference view onto the second reference view via the homography induced by the blue 3D plane

Figure 2.11: Homography defined by the blue plane $\boldsymbol{\pi}$. Only the points $\mathbf{X}$ that belong to this tested 3D plane will have their 2D coordinates $\mathbf{x}$ correctly transferred on their corresponding points $\mathbf{x}'$. Image from the courtesy of [148].

As pointed by the red line in Figure 2.11(c), because the 3D of the cup is not correctly approximated by the 3D planar blue panel, its transfer via this plane does not match the cup in the second reference view. Instead of investigating separately the depth associated to each pixel $\mathbf{x}$, the 3D of a (piecewise-planar) 3D scene can thus be inferred by projecting one view onto another, via a set of plausible homographies $\mathbf{H}_{\boldsymbol{\pi}_i}$, and associating to $\mathbf{x}$ the depth of the ones which minimizes a measure of the photometric discrepancy, *e.g.*, $\|\mathcal{I}(\widetilde{\mathbf{x}}) - \mathcal{I}'(\mathbf{H}_{\boldsymbol{\pi}_i}\widetilde{\mathbf{x}})\|_2$, as further detailed in Section 2.2.4.

The tremedeous amount of methods estimating the 3D of a scene from a set of planes, called *planar proxies*, mainly differ in three points:

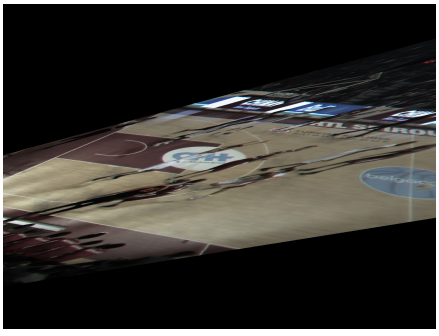
- The number of tested planes.
- The investigated orientations.
- How do they consider pixels that have no correspondence in another view (occluded pixels)?

Also, the division of the 3D object into a set of parts, such as proposed by pose estimation methods, might not be small enough to accurately approximate the

3D part by a 3D plane. In general, 3D reconstruction methods based on planar *proxies* have the advantage to require only a sparse set of correspondences (cfr. Section 2.3.3) to assign a penalty function to a given plane, but are limited to objects with a strong planarity [66] [191] [71] or to stereo setups in which the depth of the observed object is high compared to the baseline [72] [95], for which a simple image stitching [192] [27] is sufficient to interpolate an intermediate view in-between the reference ones. In other environments, the intermediate views of the object might render an object that looks flat [161], especially when the plane is not precisely estimated, as illustrated on Figure 2.12.

(a) Left view

(b) Right view



(c) Left → right homography via the ground plane

(d) Left → right homography via the back of the referee

Figure 2.12: When the 3D plane associated to a homography does not correctly represent the 3D of a (part of a) scene, the rendering of this last one might look flat. This is especially noticeable on Figure 2.12(c), where the ground plane does not accurately approximate the basketball players.

Germann *et al.* [73] have solved this problem by subdividing a planar proxy (chosen to be triangular and initialized based on Delaunay's triangulation [46]) recursively as long as the photometric discrepancy of this planar transfer overcomes a certain threshold.

Homographies in a calibrated stereo rig

The previous section has introduced the fact that a given 3D plane $\boldsymbol{\pi}$ defines uniquely[13] (up to the scaling factor) a homography $\mathbf{H}_{\boldsymbol{\pi}}$ which transfers one 2D point $\mathbf{x}$ from one view to another. This section assumes that these views are defined based on their projection matrices $\mathbf{P}$ and $\mathbf{P}'$ which can be expressed in their canonical form:

$$\mathbf{P} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}] \qquad \mathbf{P}' = \mathbf{K}'[\mathbf{R} \mid \mathbf{t}] \tag{2.16}$$

This form can either be obtained through the fundamental matrix, *i.e.*, based on Equation (2.15), or by decomposing the projection matrices (cfr. Section 2.1.1), and applying the inverse rotation and opposed translation of the first view on both of them. Given a calibrated stereo rig defined by Equation (2.16), Hartley and Zisserman have proven that the homography induced by the plane $\boldsymbol{\pi} = [a \quad b \quad c \quad d]$ is given by:

$$\mathbf{H}_{\boldsymbol{\pi}} = \mathbf{K}' \left(\mathbf{R} - \frac{\mathbf{t} \ [a \quad b \quad c]}{d} \right) \mathbf{K}^{-1}, \tag{2.17}$$

and that this homography is equivalently defined based only on the fundamental matrix $\mathbf{F}$ of the canonical stereo pair (cfr. Equation (2.15)):

$$\mathbf{H}_{\boldsymbol{\pi}} = \mathbf{A} - \frac{\widetilde{\mathbf{e}}'[a \quad b \quad c]}{d}, \tag{2.18}$$

with $\mathbf{A} = \left([\widetilde{\mathbf{e}}']_{\times}\right)^{-1} \mathbf{F}$.

Interestingly, to avoid numerical instabilities, the inverse homography $\mathbf{H}_{\boldsymbol{\pi}}^{-1}$ which transfers a 2D coordinate $\mathbf{x}'$ of the second view onto the first one, *i.e.*, $\widetilde{\mathbf{x}} = \mathbf{H}_{\boldsymbol{\pi}}^{-1}\widetilde{\mathbf{x}}'$, can be determined based on the Sherman-Morisson formula [188]:

$$\mathbf{H}_{\boldsymbol{\pi}}^{-1} = \mathbf{A}^{-1} \left(\mathbf{I} + \frac{\widetilde{\mathbf{e}}' \ [a \quad b \quad c] \ \mathbf{A}^{-1}}{1 - [a \quad b \quad c] \ \mathbf{A}^{-1}\widetilde{\mathbf{e}}'} \right).$$

Homographies by three points and F

Equations (2.17) and (2.18) still require to define the homographic plane $\boldsymbol{\pi} = [a \quad b \quad c \quad d]^{\top}$ in the 3D world's coordinates, *e.g.*, based on a set of three 3D points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \in \boldsymbol{\pi}$ and thus to define a 3D Euclidean space and calibrate the camera accordingly. To reduce this time-consuming work, Hartley and Zisserman [92] have proposed to derive the homography directly in the projective space, based on a set of three pairs of correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}_{i=1,2,3}$. The homography induced by the plane is given by:

$$\mathbf{H} = \mathbf{A} - \widetilde{\mathbf{e}}' \left(\mathbf{M}^{-1}\mathbf{b} \right)^{\top},$$

with $\mathbf{A} = \left([\widetilde{\mathbf{e}}']_{\times}\right)^{-1} \mathbf{F}$, $\mathbf{M} \in \mathbb{R}^{3\times3}$ having its i^{th} row defined by $\widetilde{\mathbf{x}}_i^{\top}$ and $\mathbf{b} \in \mathbb{R}^{3\times1}$ with

$$b_i = \frac{\left(\widetilde{\mathbf{x}}'_i \times (\mathbf{A}\widetilde{\mathbf{x}}_i)\right)^{\top} \left(\widetilde{\mathbf{x}}'_i \times \widetilde{\mathbf{e}}'\right)}{\left\|\widetilde{\mathbf{x}}'_i \times \widetilde{\mathbf{e}}'\right\|_2^2}.$$

[13] As long as this plane does not contain any of the camera centers.

Plane sweeping

Plane sweeping methods refer to the ones which investigate the depth of the 3D scene based on a set of planar *proxies* having their normal vectors orthogonal to the image plane of a (reference or virtual) view, as illustrated in Figure 2.13.

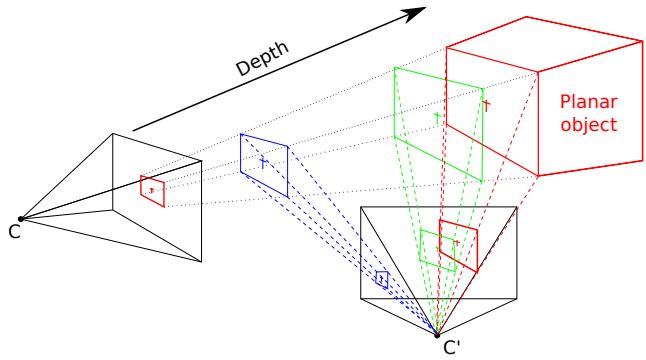


Figure 2.13: In order to estimate the depth of a 3D scene, plane-sweeping methods investigate successively multiple depth hypotheses by sweeping a plane through the 3D space. The optimal depths are estimated as the ones minimizing the discrepancy between the image observed in the other reference view and the homography projection (via the investigated swept plane) of the first reference image.

Although this kind of methods is highly parallelizable (and thus often pushed on GPU hardwares [242] [80] [82] [79]), they generally avoid the computational overhead of reconstructing the 3D of an entire scene, but instead generally focus on the 3D estimation of a well-localized object, by uniformly subsampling this restricted space with a set of planar proxies [36] [243]. The direct extension of this method to multiple well-localized objects tends however to produce ghosting artefacts (*e.g.*, the appearance of a third leg when reconstructing a human being), *i.e.*, the merge of non-corresponding textures during the reconstruction of the intermediate views, due to the mismatches, which are especially frequent when these objects display similar textures. Also, the quality of each reconstructed 3D model highly depends on the sampling step. Goorts *et al.* [82] have proposed to refine the depth obtained in a first coarse plane-sweeping phase [243], by investigating around the median depth of the analyzed object with an adaptative sampling step that is proportional to foreground density obtained by the first plane-sweeping phase. Although this method highly reduces the amount of investigate planar proxies, the accuracy of the reconstruction of the 3D model still highly depends upon the efficiency of the initial foreground extraction, especially due to the fact that any transfer falling into the background of only one reference view is considered as belonging to the background. While this limit is not restrictive in the soccer's environment, because of the simple and constant green background allowing to simply extract the foreground by thresholding a measure of the amplitude and color angle with respect to the green color [81], the generalization of the

method to more complex scene is not straightforward. Guillemaut *et al.* [87] have addressed the foreground mask issue by simultaneously optimizing the segmentation and the 3D estimation, leading to impressive results on soccer applications.

By considering multiple directions others than the ones parallel to an image plane, plane-sweeping methods have been extended to *billboarding* methods [215] [71]. Kitahara *et al.* [113] were among the first ones to propose to sample a Euclidean space parallely to the ground plane, defined at $Z = 0$. More interstingly, they adjust the sampling step and the resolution of planes (level of details) based on the relative locations of (i) the observer's viewing position, (ii) the multiple cameras, and (iii) the 3D object, based on a measure of the distortion of the projected 3D shape. This measure also confirms that the distortion of the projected 3D shape on the virtual viewpoint hardly decreases when the observing distance increases, allowing to simply approximate a 3D shape by an single plane when the object is "far away" from the reference (and virtual) cameras (or if the camera's relative angle is smaller than 10° [12]).

Billboarding has also been more recently used in [12] to reconstruct human beings in virtual view between two reference cameras. In this case, only 2 planes, parallel to two reference cameras, are placed in such a way that their intersecting line fit the principal direction of the human being they aim to reconstruct. Because two planes are insufficient to accurately approximate the 3D of the human being, the user is restricted to navigate from one reference view to another only when the colorimetric residue (considered only on foreground masks) is small enough.

2.3 Image representations for correspondences

Given the geometric constraints described in the previous sections, one can search for correspondences in-between viewpoints. To find such corrrespondences at the pixel level, it sounds natural to assume that the apparent brightness $\mathcal{I}(\widetilde{\mathbf{x}}) = \mathcal{I}\left(\mathbf{P}\widetilde{\mathbf{X}}\right)$ of the same 3D point $\mathbf{X}$ is constant whatever the reference camera $\mathbf{P}$. Such invariance requires that:

- Hypothesis 1: the 3D scene reflects the light rays isotropically and with equal intensity in all directions.
- Hypothesis 2: all the reference cameras measure the luminance of a (set of) light ray(s) equally.
- Hypothesis 3: the luminance measure is discriminative enough to distinguish two pixels that do not represent the same 3D point.

However, all of these three assumptions are in practice violated. Precisely, even for diffusely reflecting surfaces (*i.e.*, Lambertian surfaces), the luminous intensity of the reflected light rays depends on the angle between the incident ray at the normal of the 3D surface [34]. Because this angle can not be estimated without reconstructing the 3D surface, this chicken-egg problem is generally neglected. The first hypothesis is also generally assumed for

non-Lambertian surfaces, *e.g.*, partially reflecting and transparent surfaces, excepted in very narrow-baseline setups [233], when some specific properties of these surfaces can be used to estimate their 3D [233].

The second hypothesis assumes a perfect color balance between the reference cameras, while the camera's color perception varies not only from one version to another, but also for a same version of a low-cost camera (due to the high variability of low cost components). This color discrepancy can be reduced by precisely adjusting the camera's acquisition parameters, which is a labourious and time-consuming task. It has been automated in [161] based on a Macbeth color-checker [150], and/or by post-correction of the captured images. Among the most famous post-corrections, white-balancing [19] simply rescales the RGB components to transform a specific image's point into a pure white (no color temperature bias). Instead of independently rectifying the colors of the stereo images, the stereo images presented in this thesis have been systematically color rectified. This is explained in Section 2.3.1 below.

The second and third part of this section extend the pixel's luminance towards more discriminative similarity measures, by completing the observed colorimetric information with geometric and/or semantic informations surrounding the pixel. Specifically, the state-of-the-art dense descriptors, *i.e.*, that can be computed at each pixel's location, are first introduced. Their discriminativeness, obtained by completing the pixel colorimetry with informations about its neighborhood pixels in an area of predefined shape, is paid at the price of an increased sensitivity to occlusions and perspective changes. Next, it is shown how several state-of-the-art methods adapt the neighborhood to the image content, in such a way to construct region descriptors that are more robust to photometric and perspective changes.

2.3.1 Color calibration by histogram matching

Histogram matching [55], as its name suggests it, tries to equalize the color histogram of one image with respect to the color histogram of another reference view. This is done by determining a mapping function $f(\mathcal{I}') : \mathbb{R} \mapsto \mathbb{R}$ that transforms the luminance value of the first image into the luminance value of the second image. The procedure can be generalized to color images by applying it independently on all the color channels of its YCbCr representation. This mapping function is determined based on the cumulative density function of the histograms, notated $\mathrm{c}_{\mathcal{I}}$ and $\mathrm{c}_{\mathcal{I}'}$ and defines the bijective mapping of the intensities $\mathcal{I}'$ onto $\mathcal{I}$ by posing $f(\mathcal{I}')$ as:

$$\begin{aligned} f(j) &= i && \text{with } i \in \mathcal{I}, j \in \mathcal{I}' \\ \text{such that} \quad & \mathrm{c}_{\mathcal{I}}(i) \leq \mathrm{c}_{\mathcal{I}'}(j) < \mathrm{c}_{\mathcal{I}}(i+1) \end{aligned}$$

with $\mathrm{c}_{\mathcal{I}}(i) = \#\{p : \mathcal{I}(p) \leq i\}$ and # denotes the cardinality of a set. Figure 2.14 illustrates how the mapping function f is estimated and exploited. An example of color-calibrated stereo pair is illustrated in Figure 2.15.

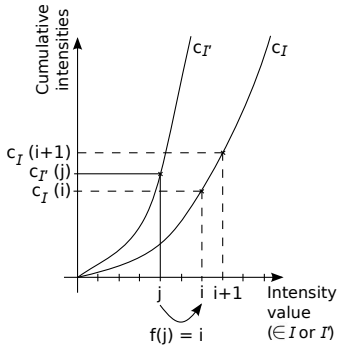


Figure 2.14: Color calibration is done by equalizing of the cumulative density function of the intensity histograms of the two views.

(a) First reference view

(b) Second reference view

(c) Color-corrected second reference view

Figure 2.15: Figure (c) illustrates the color-corrected version of Figure (b) based on the color histogram of Figure (a).

2.3.2 Dense representations

Instead of describing a pixel based only on its brightness, more discriminative descriptors consider its neighborhood [181]. The pixel comparison, required to determine correspondences, is then generalized into the measure of similarity (or dissimilarity) between two templates (regions) centered on the investigated pixels. In the literature, this procedure is often referred as *template matching*, and the comparison is done by translating a reference template, captured in one of the views, across the other view, and considering the position of maximum similarity as the corresponding one. The zero-mean normalized cross-correlation (ZNCC) [54] is one of the most appreciated similarity measure, due to its invariance to an illumination bias between the two reference views. Such kind of fixed-template matching assumes thus that the local transformation between the patches, observed in two different views, is a pure translation. This is almost true for very small-baseline setups, if the focal length of the two reference views are the same. However, if the focal lengths differ, the 2D area representing the same 3D region will be k times bigger in one reference view compared to the other, if this first one has a focal length equals to k times the focal length of the other camera (with $k \in]0;\infty[$). Because this focal length can not always be precisely estimated (as explained in Section 2.1.1), scale-invariant descriptors have been introduced.

SIFT [134] and SURF [14] are among the most well-known scale-invariant descriptors. Their invariance comes from the fact that the pixel neighborhood is defined at a certain scale that varies from patch to patch, in such a way to be optimal for detecting the specularities. This optimal scale is determined based on the Laplacian of Gaussian [23], which is approximated as a difference of Gaussian (DoG) for SIFT, and a difference of squared windows for SURF. Instead of collecting the absolute brightness of the pixels included in the region, SIFT and SURF describe a region based on its gradients (amplitude and direction for SIFT, analogeously computed as the response to Haar-wavelet filters in SURF), allowing to be invariant to a constant illumination bias between the two images.

SURF is generally preferred on SIFT for its fast computation speed, which comes from the intensive use of integral images [42]. Instead of relying on the intensity (brightness) image, Abdel *et al.* have extended SIFT to the RGB color space by defining an illumination invariant space, called the H invariant space [74], and determine the descriptors in this space.

When the reference views form a wide-baseline stereo pair, not only the change of scale should be considered, but also the plausible different rotations of the cameras, the affine perspective transformation or even the projective transformation that could make the two views completely different. Rotation invariance has been achieved in SIFT and SURF, by associating to each descriptor the most common direction of its gradients. A similar idea has been exploited in [28], in which the squared neighborhood is oriented according to orientation of a smoothed local gradient around the pixel of interest. Yu *et al.* [245] have extended the translation, scale and rotation invariance/robustness of SIFT to an affine invariance by simulating all the image views obtained by varying the two camera axis orientation parameters (latitude and longitude angles). The main disavantages of these descriptors are:

- their lack of discriminativeness generating multiple local minima in their associated matching function [218], which can attract any optimization process through erroneous matches.

- their important computation time, making them unadapted for dense computation.

To counter these facts, the correspondences are generally determined only on highly informative points, called *keypoints*. Those points are detected for example based on the Jacobian matrix [91] or on the Hessian matrix [14]. These sparse correspondences can then be used as seed points to propagate the matches to the other points.

In [129] and [130], correspondences are propagated in a predefined neighborhood around the keypoints, by considering only as plausible correspondences the ones that respect the epipolar constraint $\widetilde{\mathbf{x}}'^T\mathbf{F}\widetilde{\mathbf{x}} = 0$, while having a similar relative location with respect to their matched keypoints (defined based on a threshold) and maximizing the ZNCC. A similar idea has been exploited by Sun *et al.* [209], in which the keypoint correspondences are propagated along the scanline of the rectified stereo pair for pixels having a similar color. The propagation is extended to 2D affine models in [110], while Yao *et*

al. [244] drive the propagation in 2D by considering both an illumination invariant photoconsistency measure and the smoothness of the matching. However, the dense matches are generally only guaranted in small neighborhood around the keypoints, and their accuracies strongly depend on the truthfulness of the scene's prior. For example, the determination of correspondences based on affine transformations implicitly assumes an affine scene, which is not valid when it is composed of 3D slanted planes[14].

Recently, Tola *et al.* [217] have proposed a projective-robust descriptor that can be densely computed, while also being robust to the photometric and projective changes. This state-of-the-art descriptor, called Daisy, takes the best of the previously proposed ones (*e.g.*, SIFT, SURF, GLOH, etc.), based on the conclusions of their multiple comparisons [237] [238] [154]. Precisely, its outperforming matching performances [218] rely on the combination of four complementary features:

- The description of the pixel neighborhood by the oriented first derivative of its 2D gaussian smoothing, which ensures its photometric invariance to illumination bias. This oriented-gradient representation is obtained based on steerable filters, which are generally used in the best descriptors [237].

- The log-polar histograms of the oriented gradients. While SIFT and SURF define the pixel neighborhood based on a squared grid, and consider independently the oriented-gradients along this grid, Mikolajczyk and Schmid [154] have shown that circular grids (such as used in GLOH) have better localization properties [135]. As illustrated in Figure 2.16, Daisy extends this principle by using multiple circular grids. Its Daisy shape, which gives its name to the descriptor, has been shown not only to be optimal for sparse matching [237] but also to remain stable under projective distortions [238].

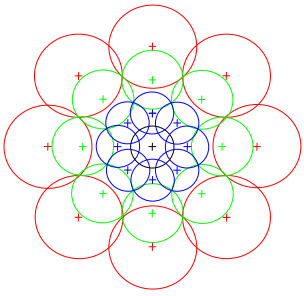


Figure 2.16: A Daisy description represents a keypoint (black cross) by a sampled version (taken on the crosses) of convolutions of the original image with several oriented derivatives of Gaussian filters. The radius of each circle is proportional to the standard deviation of the Gaussian kernels, meaning that high frequencies details of distant neighborhood tend to be neglected.

[14] Due to the fact that the 2D projection of any 3D slanted plane is not affine invariant, because its area varies with respect to the viewpoint.

An intuitive example concerns the perfect rotational symmetry of log-polar shapes, which makes the Daisy descriptor naturally robust to rotational perturbations [218]. This stability under perspective transformations makes the Daisy descriptor powerful, especially in wide-baseline setups, which suffer from strong projective distortions.

- The log-polar weighting and blurring of the oriented gradients. As illustrated in Figure 2.16, Daisy uses larger gaussian kernels to blur the oriented gradients belonging to its outer rings and gaussian kernels with smaller standard deviation near the keypoint. This adaptive blurring is strongly inspired from the geometric blur proposed by Berg and Malik [16], which has been proved to increase the robustness under affine distortions, always present in-between wide-baseline stereo views.

- Its fast computational speed (*e.g.*, more than 50 times faster than SIFT). This fast computation is achieved by convolving gradients maps using 2D Gaussian kernels, which have the advantage to be separable in two 1D kernels and to support multiscale computation. Moreover, the histograms of gradients can be computed only once per region and reused for all the neighboring pixels. It allows the dense description of high resolution images that exhibits highly textured and discriminative content at this resolution, while appearing uniform if captured at a smaller resolution.

Due to the unequalled robustness and discriminativeness of Daisy descriptors on wide-baseline stereo [237] [238] [218] [73], this thesis uses them as soon as a dense descriptor is required (cfr. Chapter 3 and Chapter 4).

While Daisy has been developped to be fast and robust to the projective transformations, its main weakness comes from its inconsideration of occluded areas, in which the neighboring pixels should not be taken into account because they are not visible in the other view. Estimating the occluded parts generally requires to estimate the 3D, and is thus a chicken-egg problem. In [218], occlusion masks are selected during the 3D estimation using an Expectation Maximization (EM) framework and are applied onto the Daisy descriptors to hide the occluded parts.

Because the pixels in the occluded parts generally share similar photometric/geometric/semantic properties, the next section shows how these groups can be extracted and described with a high repeatability based on distinguishing, invariant and stable properties in such a way to determine the correspondences not anymore at the pixel level, but at a superpixel (group of pixels) level.

2.3.3 Sparse representations and segments/regions matching

Segment-based stereo methods use 2D image regions, called *superpixels* or *segments*, as matching units instead of pixels. When the region is properly defined and described, segments have been shown to include richer and more discriminative content than pixels, pushing segment-based stereo matching systematically among the top-ranked algorithms in stereo [144] during the last years. Segment-based stereo methods generally define three main steps:

1. The determination of 2D regions, which can be equally detected (high repeatability) among the reference views, despite their strong perspective changes.

2. The description of these regions based on geometric and photometric invariant (or robust) features.

3. The method used to match these described regions.

There mainly exists two types of methods used to match these described regions. They mostly only rely on concepts that have already been introduced in the previous sections.

In the first type of methods, the region correspondence is defined with respect to their center pixel, allowing to use all the geometric constraint introduced in the previous sections. The disparity found for the center pixel, generally based on its region's properties, is assigned to all the pixels included in the region [103]. The 3D scene is thus implicitly approximated by a set of fronto-planar 3D surfaces [126]. For example, in [144], each region receives the maximum likelihood disparity, obtained based on [143] and [96].

The second type of methods assumes that the depth within each segment varies linearly. The 3D of each segment is thus explicitly represented as a (slanted) planar surface, which is estimated by robust plane fitting (*e.g.*, by means of RANSAC [58]). For example, in [114], a set of plausible horizontal and vertical slants of each plane are proposed based on a set of reliable pixel matches lying on the same horizontal (respectively vertical) line, and the orientation is then locally optimized based on the sum of absolute difference between the homography-transferred pixels.

Both of these methods thus assume that the 3D scene can be approximated by non-overlapping smooth regions, for which it is assumed that disparity/depth values within each superpixel are either constant or vary lineary, while depth-discontinuities occur along the segment boundaries. Since this hypothesis is only approximative [144], the depth maps estimated based on region-based methods are sometimes refined, by using it as a starting point in a slower depth optimizer [211], for example based on graph-cut [24] [115], PDE[15]-based matching [204], Expectation-Maximization algorithm [203], dynamic programming [20], belief-propagation [56] [207] or cooperative algorithms [232].

In the rest of this section, we first detail how to detect the 2D regions with high repeatability, and then explain how to describe them.

[15] Partial Differential Equations

To detect corresponding regions in two wide-baseline views, the region detector has to be robust to the change of perspective. Such detector has first been introduced in the seminal work of Tuytelaars and Van Gool, who have focused on affine invariant and projective-robust image patches, *i.e.*, that automatically deform with changing viewpoint as to keep on covering identical physical parts of a scene. Most of their preliminary works [224] rely of the affine-invariance of an ellitical-shaped area [92], and thus aim at detecting regions that are approximated by an ellipse. Since affine transformations do not fully cover the observed changes, this model will suffice for regions that are sufficiently small and planar [228]. For example, in [223], local intensity maxima are selected, and associated to a segment, defined by the closed shape covering the maximum intensities along axis spanning the 360° around each of these points. An ellipse having an area twice as big as the segment area is fitted onto each region, in such a way to capture more diversified texture patterns, to increase the distinctive power of the region descriptor.

Among the most famous perspective-robust region detectors, maximally stable extremal region (MSER) [140] has shown to overpass the other affine-robust region detectors, *e.g.*, Gradient-based detectors [179], Hessian-based detectors [153], edge-based detectors [222] or saliency-based region detectors [105], in term of matching performances [155]. Instead of detecting the maximally stable regions based only on the image intensity values, Forssen *et al.* [62] generalize the detection to the RGB color space.

However, this affine-invariant region detector, as many others, does not ensure that all the pixels within an image will be covered by such descriptor, or that a match will be found for each region. Two solutions to this problem have been proposed in the literature: *match propagation* and *pre-segmentation* of the reference images.

Among the first type of methods, Vergauwen *et al.* [228] have proposed to propagate the correspondences, obtained based on the match of elliptic approximation of a region [223], by imposing the smoothness of the correspondence map (disparity map). This smoothness is imposed by a non-linear diffusion process, which is a PDE-based solution of optical flows [205] [204] in which the correspondences search (and the derivative computations) is done at two different points in the two reference images.

In a recent work, the propagation proposed by Zhang *et al.* [248] describes the adjacency between the regions based on a pyramidal (tree-like) representation of each region, in which the k^{th} level of the pyramid includes the k^{th}-connectivity regions. If the analyzed region has not been matched, the correspondence is interpolated based on a trade-off between the closest region in the pyramid levels and the one that gives the highest pixel-based photoconsistency similarity.

The second kind of methods, *i.e.*, pre-segmentation methods, ensure the dense coverage of the entire reference views by first (over-)segmenting (based on k-means segmentation [137], anisotropic diffusion smoothing [168], mean-shift segmentation [37], soft-segmentation [138], etc.) the 2D images into a set of non-overlapping regions, and then using a projective-robust region descriptor and its associated similarity metric.

For example, in [223], each region is associated to an ellipse and is described based on an affine-invariant combinaison of elliptical shape moments (up to the second order).

Schaffalitzky *et al.* [178] have proposed a normalization of both the geometric and the photometric statistics (*e.g.*, covariance matrix of the gradient) aggregated over each region to make them affine-invariant. The dissimilarity measure is just computed as the ℓ_2 norm of these normalized descriptors.

In Zitnick *et al.* [252], the reference images are independently (over-)segmented and correspondences are found by minimizing the Euclidean distance on the normalized histogram of pixels included in each region. Occluded areas are isolated by imposing the correspondence as bijective, meaning that if a region of the first reference image has been associated to a specific region in the second reference image, this last region has also to be associated with the first region. This idea is an extension of the famous *disparity cross-consistency check*, imposed on pixels in [67].

More recently, Zhang *et al.* [248] have described each region by a set of affine-invariant features (Harris and Hessian features) [155] and matched them based on RANSAC, by minimizing a cost derived from the epipolar constraint (cfr. Section 2.2.3).

Among the most famous recent metrics to compare regions, AD-census [151] effectively combines the good representation of local structures, provided by the absolute differences (AD) measure, and the good discriminativeness of the census transform[16]. While the census has been proved to show the best overall results in small-baseline local and global stereo matching methods [98], due to its good capacity to disambiguate regions with similar color distributions, the AD-measure perfectly completes it due to its capacity to tolerate photometric changes and image noise.

One of the major problem with pre-segmentation stereo is that the accuracy of the region matching highly depends on the repeatability of the initial segmentation, *i.e.*, how well two segmented regions representing the same 3D region are precisely isolated.

Toshev *et al.* [220] have proposed to optimize simultaneously for the segmentation and the segment matching, based on a "co-saliency" matching score which enables to favor correspondences that are consistent with soft image segmentation [138]. Precisely, they express the problem as the determination of the dominant spectral component in a complete graph, called the Join-Image Graph (JIG), in which the nodes are the pixels of the two reference images. This spectral component is used as similarity metric in a positive feedback for updating and establishing new pixel correspondences.

As a conclusion, while dense image representation enables precise matching and thus precise 3D reconstruction, these dense descriptors are rarely discriminative enough to drive their matching towards any global optimum. At the opposite, sparse features, such as image segments, tend to be more discriminative but are weakly localized, and unprecise for 3D reconstruction. Recently, Braux-Zin *et al.* [25] have proposed a method that combines dense

[16]The census transform describes a pixel based on the superiority of its intensity compared to the ones of its neighborhood, as done in Local Binary Pattern [5], and computes the Hamming distance on these descriptors

features and sparse features in a variational framework. Precisely, they generalize the dense optical-flow optimization to wide-baseline views by showing that sparse features, such as line segments [231] [230], could enable to attract the optical-flow optimization outside local minima, generated by the (weakly discriminative) Absolute Difference-census (AD-census) measure. However, their method assumes an *a priori* known perfect match of (the borders of) the sparse features, which is only satisfied in presence of human-interaction. In contrast to this strong prior information, weaker priors, such as presented in the next section, push towards a fully automatic determination of the 3D of the scene.

2.4 Priors to disambiguate the correspondences

As explained in Section 1.3.1, data-fidelity measures are not sufficient to reconstruct the 3D of the scene without ambiguity. This is especially the case when the scene is composed of:

- *Non-lambertian surfaces*: The apparent colorimetry of an infinitesimal 3D section of such surfaces changes with regards to the observer's viewpoint.
- *Uniform and/or repetitive textures*: Because of the low-discriminativeness of such textures, any spatially-constrainted (local) matching process might swap the correspondences.
- *Foreshortening effects*: The foreshortening effect causes a distance or an object to appear shorter/wider than it is because it is angled toward one of the viewers (see Figure 2.17). Because the compaction ratio depends on the viewpoints, a given 3D object will be represented by a totally different number of pixels in different views.

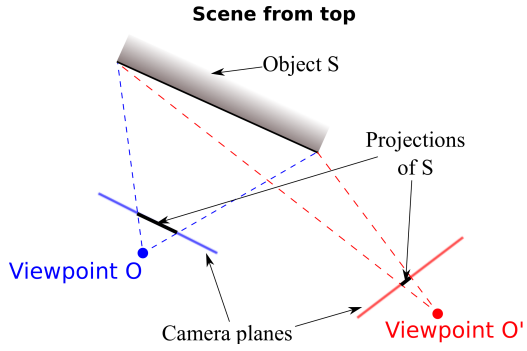


Figure 2.17: Illustration of the foreshortening effect. The projection of the object S is more compact in O' than in O.

- *Occlusions*: An occlusion occurs when a part of the scene can be observed in only one of the camera views, so that no correspondence can be found with the other reference views.

To disambiguate the ill-posedness of such scenes, multiple priors have been proposed:

- The photometric consistency considers that the apparent colorimetry of an infinitesimal 3D surface does not change regardless the observer's viewpoint. The validity of this prior depends on the Lambertian reflectances of the surfaces composing the 3D scenes, which could strongly affect the colorimetry of a given pixel from one point of view to the other. Previous works have reduced this impact by defining descriptors that characterize the relative photometric information of the pixel's neighbors. Unfortunately, the pixel's organization in the spatial area is strongly affected by the change of perspective, meaning that the problem is turned into the determination of geometric invariants. While dense affine invariants exist and have already been exploited [157], no dense invariance to the real word geometry, *i.e.*, the projective geometry, is known until now[17]. As a quite effective alternative, fast and dense projective-robust descriptors (*i.e.*, descriptors that do no change a lot when a projective deformation is applied) have been developed [218].

- The epipolar constraint [92] (described by the fundamental matrix $\mathbf{F}$) constraints each 3D point to belong to one of the 1D light ray associated to a pixel in a reference image.

- The unicity constraint assumes that the matching function is bijective [67] (one-to-one correspondences). This prior becomes invalid as the foreshortening effect gets significant, as generally encountered in wide-baseline stereo setups [178].

- The smoothness constraint models the depth values of the 3D scene as piecewise-constant [195] [144] or planar [70] [240]. This assumption is implicitly used in Chapter 5, in which a foreground object, that we aim to interpolate in virtual views, is represented by a set of line segments captured along the epipolar lines in the reference images. Once their positions in the intermediate views are determined based on a prior about the possible silhouette of the object, its textures can be linearly interpolated. This methodology is only valid when decomposing the 3D scene (and its projection) into infinitesimal 3D (2D) surfaces [252] [239]. To overcome this limitation, Li *et al.* [131] have regularized the depth based on both second and third-order priors, thereby generalizing to curved surfaces. However, their algorithm only approximates the solution of this computationally infeasible triple cliques problem, by optimizing a first-order prior on the surface's normals. Although $1D$ and 2D smoothness, often defined based on the TV (piecewise-constant model) or TGV norm (piecewise-affine model) [25], are considered among the state-of-the-art priors to alleviate the ambiguity of 3D estimation, they however lead to the loss of high-frequency details [60]. In Chapter 4, we propose

[17] The well-known projective-invariants [92], such as concurrency, collinearity, cross-ratio, etc., are defined along the projections of visible 3D lines, and not on all pixels. They are thus not dense.

to estimate the 3D of (man-made) scenes by a set of piecewise (projective) planar 3D model. Starting from a set of small 2D superpixels, for which the planar assumption is considered as valid, our method simultaneously estimates the normal of their associated 3D planes and merges the regions whose normals are similar. This joint optimization also ensures to represent the scene with a minimum set of plane models, leading to a light-weight piecewise-planar approximation of the 3D.

- The ordering constraint assumes that the left-right relation between the projection of two 3D points (belonging to the same epipolar plane) is preserved when changing the observer's point of view. However, strict ordering preservation, as implemented by earlier works [40], is violated in wide-baseline, and previous works are missing solutions to formulate and exploit the ordering as a weak and relaxed constraint. In Chapter 3, we propose to disambiguate the matches based on a relaxed version of the ordering constraint, which only favors the preservation of the order of the elements without necessary strictly forcing it.

2.5 Virtual view interpolation when correspondences are known

This section assumes that a dense correspondence map (expressed either as a depth map or equivalently as a disparity map) has been obtained by matching dense or sparse features (cfr. Sections 2.3 and 2.4). It shows how to process a map to generate, or *render*, images of the observed 3D scene from different viewpoints than the ones acquired by the reference cameras.

Precisely, the three most well-known types of methods for virtual view synthesis are presented:

- 3D *projection*: the virtual view is obtained by projecting a 3D representation of the observed scene (*e.g.*, a point-cloud or a meshed representation derived from the correspondences) onto a virtual view defined based on its (predefined) projection matrix $\mathbf{P}_v$.

- *View morphing*: parts for which a correspondence exists in the reference views are interpolated in the 2D image domain.

- *Light-field cut*: all the parts represented in at least one reference view is interpolated in the 2D image domain, based on the real observation of some other intermediate views.

Each of these methods, detailed in the following section, has its own advantages and disavantages, which are presented in Table 2.2.

[18] In practice, to avoid the projection of non-modeled parts (seen by only one or none of the reference cameras), the virtual view's pose is limited on the baseline between two reference cameras.

	3D *projection*	*View morphing*	*Light-field cut*
Required views	2 references	2 references	2 references + $\geq$ 1 intermediate(s)
Represents	Real 3D	Plausible 3D	Plausible 3D
Virtual view's pose	Arbitrary[18]	On the baseline	On the baseline
Synthesized parts	Corresponding	Corresponding	Corresponding + occluded

Table 2.2: Comparison between the three most well-known types of methods for virtual view synthesis.

2.5.1 3D projections

Based on a point-cloud or meshed [65] representation of the shape of the estimated 3D scene, 3D *projection* methods reconstruct a virtual view by fixing the 12 parameters of an arbitrary virtual view $\mathbf{P}_v$, and projecting the 3D model onto this view. Due to the arbitrary selection of the projection matrix $\mathbf{P}_v$, this principle, used since decades in computer graphics, allows the user to watch the dynamic scene from any desired viewpoint.

In case of meshed 3D model, a naive method for choosing the color of the projected points $\widetilde{\mathbf{x}}_v = \mathbf{P}_v\widetilde{\mathbf{X}}$ in the virtual view is to use the color of the reference camera that observes the 3D point with the smallest angle compared to the surface's normal. However, the lack of precision on the normal angle and the camera's photometric bias generally make the extracted pattern unsmooth and thus unpleasant for a viewer [142]. A first extension has been proposed by Seitz *et al.* [186], who have proposed to determine the color which is the "most consistent" (measured by a likelihood ratio test on a Chi-squared distribution of the color standard-deviations) with all the reference views, which often corresponds to the mean color. Instead of using the mean color as the representative one, state-of-the-art methods simply use a weighted combination of the colors, where each (normalized) weight is inversionnally proportional to the angle between a reference view and the plane normal of the 3D surface [219].

Both the geometric and colorimetric veracity of the arbitrary view depends on the accuracy of the 3D model. This model might not only be corrupted by mismatches during the 3D estimation, but might also be incomplete, *e.g.*, if a part of the scene is observed by only one reference view. These occluded areas appear as holes in the virtual image, and can be generally filled by inpainting methods [17] if a pattern similar to the occluded part is visible in the reconstructed image. However, because the amount of (dis-)occlusions increases with the distance of the virtual view from the original views [197], inpainting methods become inefficient when the pose (position and rotation)

of the virtual view highly differs from the one of the reference cameras. These drawbacks are generally solved by:

- Limiting the position of the virtual view on the baseline in-between two reference views. A generalization to three views has been proposed in [241]. Recently, Ballan *et al.* [12] have introduced additional constraints by defining the pose of the virtual camera in such a way that the projection of a 3D object of interest, assumed to be perfectly reconstructed, always stays at the center of the virtual image.

- Restricting 3D projection to simple 3D models only, *e.g.*, that can be approximated as a set of 3D planes, as applied in Chapter 4 to render simple backgrounds of 3D scenes, such as the obtained (light-weighted) piecewise-planar 3D approximation of man-made scenes.

The projection matrix $\mathbf{P}_v$ of a baseline virtual view is interpolated from the reference views' matrices $\mathbf{P}$ and $\mathbf{P}'$. While the intrinsic parameters and the position of the virtual camera can be linearly interpolated from the ones of the reference views (their extraction has been detailed in Section 2.1.1), Euler's angles can not be determined uniquely due to the Gimbal lock problem [83]. In this thesis, rotation matrices are interpolated based on their quaternion representation [90], which does not only avoid the degeneration in the rotation interpolation, but also strongly accelerates the computations by replacing computational trigonometric operations by simpler vector operations [166].

A quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)^\top \in \mathbb{R}^4$ is an algebraic structure imagined by William Rowan Hamilton [88] [89] to encode any rotation (3 degrees of freedom) in a 3D coordinate system as a four-components normalized vector ($q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$). Roughly speaking, the three last elements $(q_1, q_2, q_3)^\top$ can be thought of as Cartesian unit axes, expressed by the complex elements $(i, j, k) \in \mathbb{C}^3$, around which rotation should be performed, while the first element q_0 is the "scalar part" that specifies the amount of rotation that should be performed around the axes.

A quaternion can be represented as a linear combination of real and imaginery parts, $\mathbf{q} = q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k$, where i, j and k are called *hypercomplex numbers* that satisfy $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$ and form a non-commutative group ($ij = k$ while $ji = -k$).

The transformation from a quaternion vector $\mathbf{q} \in \mathbb{R}^4$ into a rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$ is given by [83]:

$$\begin{aligned}\mathbf{R} &= \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \\ &= \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2 \cdot (q_1 \cdot q_2 - q_0 \cdot q_3) & 2 \cdot (q_1 \cdot q_3 + q_0 \cdot q_2) \\ 2 \cdot (q_1 \cdot q_2 + q_0 \cdot q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2 \cdot (q_2 \cdot q_3 - q_0 \cdot q_1) \\ 2 \cdot (q_1 \cdot q_3 - q_0 \cdot q_2) & 2 \cdot (q_2 \cdot q_3 + q_0 \cdot q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix},\end{aligned} \tag{2.19}$$

which fulfils

$$\begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} q_0^2 \\ q_1^2 \\ q_2^2 \\ q_3^2 \end{pmatrix} = \begin{pmatrix} r_{11} \\ r_{22} \\ r_{33} \\ 1 \end{pmatrix}.$$

Therefore, the squared value of the elements of $\mathbf{q}$ can be estimated based on a rotation matrix $\mathbf{R}$ as:

$$\begin{pmatrix} q_0^2 \\ q_1^2 \\ q_2^2 \\ q_3^2 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} r_{11} \\ r_{22} \\ r_{33} \\ 1 \end{pmatrix}. \tag{2.20}$$

Because $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation matrix (cfr. Equation (2.19)), the sign of one element is arbitrary fixed[19], and the sign of the other elements results from this choice. For example, if we fix $sign(q_0) = +1$, Equations (2.19) and (2.20) lead to:

$$\begin{aligned} sign\,(q_1) &= sign\,(r_{32} - r_{23}) \\ sign\,(q_2) &= sign\,(r_{13} - r_{31}) \\ sign\,(q_3) &= sign\,(r_{21} - r_{12})\,. \end{aligned} \tag{2.21}$$

An intermediate rotation can be linearly interpolated from two reference quaternion vectors, notated $\mathbf{q}$ and $\mathbf{q}'$, based on *Spherical linear interpolation* (SLERP). Spherical linear interpolation takes its name for the fact that the interpolation follows the shortest arc on the unit sphere, which is considered as the optimal interpolation curve between two rotations [83].

Precisely, if $\alpha \in [0, 1]$ represents the relative distance from the starting configuration $\mathbf{q}$ to the target configuration $\mathbf{q}'$ along this arc, the interpolated quaternion at the relative position α is given by [189]:

$$\mathbf{q}_v(\alpha) = slerp(\mathbf{q}, \mathbf{q}', \alpha) = \mathbf{q} \cdot \left(\mathbf{q}^{-1} \cdot \mathbf{q}'\right)^{\alpha},$$

where $\mathbf{q}^{-1} = (q_0, -q_1, -q_2, -q_3)^\top$.

To avoid numerical instabilities from the power operator, $\mathbf{q}_v(\alpha)$ can be equivalently estimated as [189]:

$$\mathbf{q}_v(\alpha) = \begin{cases} \mathbf{q} \cdot (1-\alpha) + \mathbf{q}' \cdot \alpha & \text{if } (1-C) \leq \epsilon \\ \mathbf{q} \cdot \sin\left((1-\alpha) \cdot \frac{\pi}{2}\right) + \sin\left(\alpha \cdot \frac{\pi}{2}\right) \cdot (q_3, -q_2, q_1, -q_0)^\top & \text{if } (1+C) \leq \epsilon \\ \mathbf{q} \cdot \sin\left((1-\alpha) \cdot \theta\right) / \sin(\theta) + \mathbf{q} \cdot \sin(\alpha \cdot \theta) / \sin(\theta) & \text{otherwise} \end{cases} \tag{2.22}$$

where $C = \mathbf{q}^\top \cdot \mathbf{q}'$ and $\theta = \arccos(C)$ is the angle between the unit quaternions.

Once the intermediate rotation matrix interpolated based on Equations (2.20), (2.21) and (2.22), $\mathbf{P}_v$ can be reconstructed, and the textures of the intermediate view can either be directly projected via the 3D model, or rendered

[19] Fixing the sign of any other element leads to an equivalent (in term of equivalence class) representation of the quaternion.

by a weighted[20] average of the reference colors, transferred to the virtual view by homography [100] (cfr. Equation (2.17)).

Instead of projecting a predefined 3D model to reconstruct the virtual image, Woodford *et al.* [239] take the most of the knowledge of $\mathbf{P}_v$ by simultaneously optimizing the 3D model (depth information associated to the virtual camera) and the reconstructed virtual image. This is done by favoring, in the reconstructed image, the presence of image's patches that are consistent with the reference views [60]. In their paper, they show that this problem can be reduced as a depth optimization with occlusion reasoning, in which both the depth and an occlusion mask are retrieved by Quadratic Pseudo-Boolean Optimization [175].

2.5.2 View morphing

Instead of projecting a 3D representation of the scene, *view morphing* [185] interpolates the virtual images directly in the 2D image space, based on a correspondence map relating the two reference views. By focusing only on the reconstruction of a *visually plausible* virtual view [251], it allows to render more complex 3D scenes, at the price of limiting the position of the intermediate views to the baseline. Since their beginning [234], the methods based on (linear, bilinear, bicubic, etc.) image interpolation have shown promising results, especially due to the fact that holes can be filled during the interpolation phase, making them especially suited to the use of a sparse correspondence map. For example, while the determination of the 3D model associated to uniformly textured regions is ambiguous and often leads to ghosting artefacts when rendered by 3D projection, image interpolation methods can interpolate these uniform textures based only on the knowledge of edge correspondences [45].

However, although being sometimes blindly applied [101], generating intermediate views by simple (linear, bilinear, bicubic, etc.) interpolation inbetween correspondences does not guarantee to produce physically correct (*e.g.*, topologically coherent) representations of a scene [184] (cfr. Figure 2.1).

In their seminal work [185], Seitz and Dyer have demonstrated that topologically coherent intermediate views can be synthesized by (epipolarly) rectifying the reference images and applying a linear interpolation along the scanlines. This technique is called view morphing.

Because these methods mainly rely on the knowledge of the fundamental matrix, the quality of the rendered view is generally less impacted by the calibration inaccuracies [101].

This supports the generation of intermediate images by view morphing, which has been extensively used in Chapter 5.

Mathematically, view morphing relies on epipolar rectification to synthesize the intermediate textures by linear interpolation of the basis textures, such as:

$$\mathcal{I}_\alpha(u_\alpha, v) = (1-\alpha)\cdot\mathcal{I}(u,v) + \alpha\cdot\mathcal{I}'(u',v)$$

[20] The weights are generally inversionnally proportional to the angle in-between the principal axis of the virtual view and the one of the reference camera.

with $\mathcal{I}$ and $\mathcal{I}'$ the rectified reference images, $\mathcal{I}_\alpha$ the reconstructed intermediate image, u the abscissa of a pixel of $\mathcal{I}$, v its fixed ordinate (studied scanline), and u' the abscissa of the corresponding pixel in $\mathcal{I}'$. The pixel abscissa u_α is lineary interpolated as follow:

$$u_\alpha = (1-\alpha)\cdot u + \alpha \cdot u'$$

The transfer of the color from a reference image (*e.g.*, from $\mathcal{I}$ or $\mathcal{I}'$), to a target image (*e.g.*, $\mathcal{I}_\alpha$) is known in the literature as *forward warping* [38].

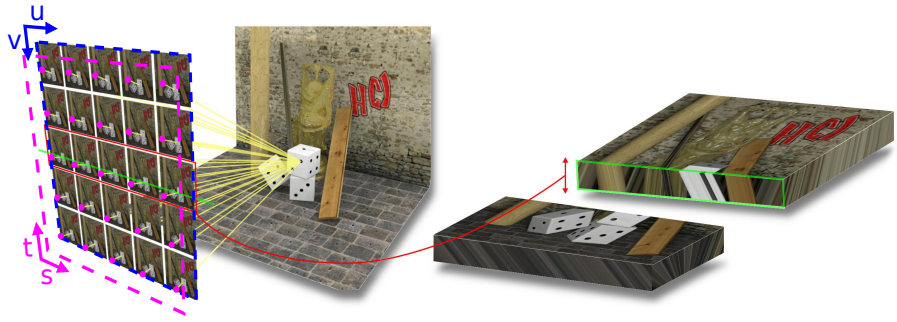
Forward warping projects the points from a source image onto a target image, based on the knowledge of a projective transformation (*e.g.*, a homography transfer $\widetilde{\mathbf{x}}' = \mathbf{H}\cdot\widetilde{\mathbf{x}}$) relating the two views. Because this projection $\mathbf{x}'$ might fail at the sub-pixel level, the coordinates of these projected points are generally rounded to their nearest pixel coordinates. As a consequence, multiple pixel from the source image might be projected on the same pixel of the target image and some pixels of the target image might not be filled. Due to these drawbacks, it is preferable to determine, for all the pixels in the target image to interpolate, the corresponding pixel(s) in the source image. This procedure, called *backward warping* [250], uses the inverse of the projective mapping (*e.g.*, $\mathbf{H}^{-1}\widetilde{\mathbf{x}}' = \widetilde{\mathbf{x}}$) to associate a (sub-)pixel coordinate $\mathbf{x}$ of the source image for each pixel $\mathbf{x}'$ in the target view. The color of each of the target pixels $\mathbf{x}'$ can then be interpolated from the colour values of the neighboring pixels around $\mathbf{x}$, *e.g.*, based on bilinear, cubic or bicubic interpolation [78] [111].

2.5.3 Light field cut

Light field rendering aims at modeling the compact set of light rays contained in a fixed 3D space, based on a simplification of the plenoptic function, introduced by Adelson and Bergen [2]. The plenoptic function defines, at a specific time t, the intensity of a light ray (of wavelength λ) passing through the 3D position $\mathbf{X} = (X, Y, Z)^\top$ and going towards the direction specified by the spherical angles ϕ and θ as a $7D$ function notated as $L(t, \lambda, X, Y, Z, \phi, \theta)$. Levoy and Hanrahan [127] have simplified this plenoptic function by assuming that the intensity of the light does not change along the ray (unless "blocked" by an occlusion [127]). In this case, they proved that, if the scene is observed by multiple views having their optical centers lying on a common 3D plane Ω and their image planes coplanar[21] in a common image plane denoted Ω' (as illustrated in Fig. 2.18(a)), the light field can be parametrized as 4D function. Indeed, the intensity of each light ray, projected on a pixel coordinate $(u, v) \in \Omega'$ of a camera having its optical center at $(s, t) \in \Omega$, can thus be represented as $L(s, t, u, v)$. By fixing the two coordinates $t^\star$ and $y^\star$, we restrict ourselves to the investigation of the light rays that belong to a 2D plane, called an *epipolar plane image*, whose intersection with Ω' is illustrated by the green line in Figure 2.18(a).

Interestingly, Bolles *et al.* [22] have shown that a sampled version of these light rays can be observed based on the reference views having the same t parameter. This observation is done in two steps. First, the reference views are

[21] Such camera configuration can always be obtained based on epipolar rectification (cfr. Section 2.2.3).



(a) (b)

Figure 2.18: After stacking (a) the reference views observing the scene into (b) a 3D cube and reconstructing its set of epipolar plane image (green rectangle), any transverse cross-section of the reconstructed 3D cube generates an intermediate view (images from the courtesy of S. Wanner [233]).

stacked on top of each other, in which the top image of the stack represents the first outermost reference view $\mathcal{I}$ while the bottom one represents $\mathcal{I}'$, *i.e.*, the other extreme one. An example of stack is illustrated in Figure 2.18(b). Second, the sampled version of an epipolar plane image is obtained by considering a cut through this stack, such as illustrated by the green rectangle in Figure 2.18(b).

In Figure 2.18(b), the scene is observed by a so dense collection of reference views (the authors of [127] mention hundreds to thousands) that each light ray becomes visible in a given epipolar plane image, and that their linear parameters can be easily estimated [41], allowing to reconstruct a continuous version of each epipolar plane image. Any intermediate viewpoint can then be generated by "sectioning" (horizontal cross-section in Figure 2.18(b)) the reconstructed set of epipolar plane images at an intermediate position $\alpha \in [0;1]$ in-between the two reference views.

However, in practice and as explained in Chapter 5, requiring both a tremendous amount of reference views and an important range of intermediate viewpoints (*i.e.*, the two outermost views forming a wide-baseline stereo pair) is not possible due to acquisition and storage limitations. This chapter also explains how these principles can be extended to wide-baseline setups, by showing that shape priors, that can be learned in advance, can help in interpolating the intermediate view, allowing to use only two reference views.

CHAPTER 3

Estimation of arbitrary 3D scenes under relaxed ordering constraint

Determining correspondences across 1D sequences is one of the principal challenges for highly parallelizable stereo matching/3D reconstruction. The problem is however ill-posed, and is thus generally solved under some additional prior constraints. One of the most popular prior enforces the strict preservation of the left-right relationships between the elements, which is known as the ordering constraint, and can directly be implemented based on *dynamic programming*. However, the strict preservation of this relative order becomes violated when the images are captured by two cameras that are far apart from each other. In this chapter, we propose to disambiguate the matches based on a relaxed version of the *ordering constraint*, which only favors the preservation of the order of the elements without necessary strictly forcing it. This is done by proposing a new objective function, whose coordinate descent maximization not only disambiguates the correspondences based on the order information, but also detects the occluded (no correspondence) elements. Simulations on synthetic data demonstrate that our proposed approach determines correspondences (and occluded elements) more accurately than *dynamic programming*. Validations on real images illustrate that the *relaxed ordering constraint* helps in finding correspondences (and thus 3D information) in wide-baseline stereo.

3.1 Introduction

Stereo matching is one of the oldest problems in computer vision, with numerous applications in 3D reconstruction, image interpolation, multi-view detection, tracking and content-based image retrieval. While multiple priors have been proposed to disambiguate this ill-posed problem when the views are captured by two cameras with similar poses (small-baseline configuration), most of those priors are violated when the two views are captured from very different viewpoints. This is the case for the ordering constraint, which forces a strict preservation of the left-right relationships between the elements to match. This prior is one of the most effective priors for narrow-baseline stereo matching, but appears to be regularly violated in wide-baseline setups (see Figures 1.6 and 2.17). Nevertheless, even if a strict constraint is too aggressive, favoring the preservation of left-right relationships between pairs of elements to match across images remains relevant in wide-baseline matching, since this order reflects the underlying spatial organization of the scene. For this reason, we propose a new prior, called the *relaxed ordering constraint*, which tends to preserve the spatial relationships between the elements to match without necessary strictly forcing them.

The chapter is organized as follows. In Section 3.2, we survey the main priors used in stereo matching. Then, Section 3.3 shows that determining the optimal correspondences under *relaxed ordering constraint* can be formulated as a labelling problem, for which an objective function is introduced in Section 3.4. The maximization of the objective function simultaneously detects the elements that have no correspondence and matches the others according to both their similarity and their relative order. Section 3.5 adopts a coordinate descent method to solve this maximization problem. Finally, Section 3.6 demonstrates the relevance of our approach. It first relies on synthetic data to show that the proposed objective function is well suited for stereo matching, and that the coordinate descent optimization scheme is as accurate as a brute-force search, while being computationally affordable. It then demonstrates, both on synthetic and real images, that our relaxed formulation of the ordering constraint outperforms the strict ordering constraint, especially when the distance between views increases. To the best of our knowledge, this work is the first one to introduce and manipulate a *relaxed ordering constraint*. It also confirms that such a weak constraint is especially suited to computer vision problems.

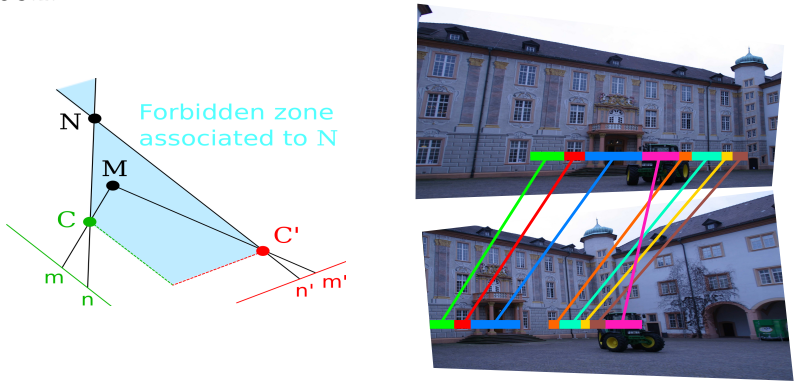
3.2 Related work

While the determination of sparse (and reliable) correspondences between two arbitrary views of a same 3D scene has been deeply investigated [223][224] [220], we focus on the determination of dense (pixel) correspondences between only two views, and assume that the stereo pair is calibrated. In this case, the 2D matching problem can be turned into the determination of correspondences along pairs of 1D epipolar lines [92]. This 1D problem is nevertheless an ill-posed problem, especially in presence of non-lambertian surfaces, uniform and/or repetitive textures, foreshortening effects and occlu-

sions. Multiple priors have been proposed to disambiguate the matches, and have been deeply detailed in Section 2.4.

In summary, the photometric consistency is the primary prior to consider when matching elements from different views, and perspective-robust descriptors (like Daisy [218]) are recommended to exploit this prior in wide-baseline setups. We have opted for the use of the Daisy descriptor [218] in our validation (see Section 3.6), due to its outperforming robustness/discriminativeness trade-off, compared to the usual descriptors such as SIFT, SURF, GLOH, etc (cfr. Section 2.3.2). Those descriptors are considered in our validation (see Section 3.6). Two other priors remain relevant in wide-baseline, namely the smoothness and the ordering. Interestingly, those two priors provide complementary hints about the matching: while the first one imposes a local constraint around the pixel to match (and would thus for example not help in determining dense correspondences between two views of a brick wall), the *ordering constraint* brings information about the spatial and more global organization of the scene.

Earlier works exploit this ordering prior by forcing a strict preservation of the order. This is typically done based on a dynamic programming framework [40]. However, despite the fact that the ordering constraint remains roughly valid in wide-baseline stereo, its strict preservation is violated, thereby preventing the use of conventional dynamic programming solutions. Precisely, the strict ordering is only valid when no 3D surface is included in the *forbidden zone* (induced by (dis)-occlusions) of another 3D surface [246], as illustrated in Figure 3.1.



(a) (b)

Figure 3.1: (a) The strict preservation of the order of the elements ("m is on the left of n") is violated because the 3D point **M** falls into the *forbidden zone* generated by **N**, with respect to the cameras' optical centers **C** and **C**′. (b) The violation of the order implies that the "left-right" relations between (sets of) pixels, captured along corresponding epipolar lines, are not preserved.

Hence, the contribution of this chapter is a relaxed version of the *ordering constraint*, which can be used in stereo matching to only favors the preservation of the order of the elements without necessary strictly forcing it. This means that it tolerates the inversion, as long as they are supported/promoted by other priors, such as the photometric consistency.

3.3 Problem definition

Turning the 2D stereo matching problem into the determination of correspondences along pairs of corresponding epipolar lines [92], we focus on the determination of the correspondences between the elements of two $1D$ sequences denoted $\mathbf{S}^1 = \{\mathbf{s}_1^1, \mathbf{s}_2^1, \cdots, \mathbf{s}_{N_1}^1\}$ and $\mathbf{S}^2 = \{\mathbf{s}_1^2, \mathbf{s}_2^2, \cdots, \mathbf{s}_{N_2}^2\}$. Each entry $\mathbf{s}_j^i \in \mathbb{R}^D$, with $i \in \{1,2\}$ and $j \in \{1, \cdots, N_i\}$, is a D-dimensional representation of the j^{th} element of the i^{th} sequence. As in [9], the correspondence problem is expressed as the determination of a labelling function l that labels each entry $\mathbf{s}_j^i \in \mathbf{S}^i$ with a label $l(\mathbf{s}_j^i) \triangleq l_j^i \in \{\mathcal{L} \cup \emptyset\}$ (with $|\mathcal{L}|$ being the total number of labels, as described here below), such that corresponding elements are labelled with the same label, and unmatched elements (no correspondence) are assigned to the empty set $\emptyset$. Let us define $\mathbf{L}^i = \{l_1^i, l_2^i, \cdots, l_{N_i}^i\}$ as the set of labels assigned to $\mathbf{S}^i$ and $\mathbf{\Phi} \in [0,1]^{N_1 \times N_2}$ the pairwise similarity between the elements of $\mathbf{S}^1$ and $\mathbf{S}^2$ (*e.g.*, the closeness between pixel descriptors). Because the correspondences between the labels impose only a relative dependency between $\mathbf{L}^1$ and $\mathbf{L}^2$, we proceed in two steps:

1. We fix $\mathbf{L}^1$ to the ordered sequence $\{l_1^1 = 1, l_2^1 = 2, \cdots, l_{N_1}^1 = N_1\}$, and determine the optimal $\mathbf{L}^2$, with $l_k^2 \in \{\mathbf{L}^1 \cup \emptyset\}, \ \forall k \in \{1, \cdots, N_2\}$. This step assumes that each element of $\mathbf{S}^2$ corresponds to at most one element of $\mathbf{S}^1$ or to the empty set, *i.e.*, $\mathbf{L}^2$ describes a surjective mapping to $\{\mathbf{L}^1 \cup \emptyset\}$, *i.e.*, $\sum_k \delta\left(l_k^2 \neq \emptyset\right) = N_1 = \sum_k \sum_j \delta\left(l_k^2 = l_j^1, l_k^2 \neq \emptyset\right)$.

2. Based on the optimal $\mathbf{L}^2$ determined in the previous step, $\mathbf{L}^1$ is corrected *a posteriori*, such that the elements $\mathbf{s}_k^1$ that have no correspondence (*i.e.*, $l_k^1 \notin \mathbf{L}^2$) are assigned either to the empty set $\emptyset$, or to one of the elements $\mathbf{s}_k^2$ whose label is not shared with any other element of $\mathbf{S}^2$. Symmetrically to step 1, this step assumes thus that each of those unmatched elements of $\mathbf{S}^1$ corresponds to at most one element of $\mathbf{S}^2$. This is implemented by reversing the algorithm used in step 1, keeping the elements of $\mathbf{S}^1$ already associated to $\mathbf{S}^2$ unchanged.

The asymmetry of this process and its implications are discussed in Chapter 6. Without loss of generality, we thus restrict ourselves to mapping $\mathbf{S}^2$ onto $\mathbf{S}^1$. The problem, which corresponds to step 1 above, is thus reduced to the determination of an optimal labelling $\mathbf{L}^2 \in \{\mathbf{L}^1 \cup \emptyset\}^{N_2}$, knowing $\mathbf{L}^1$. The optimization is done in a probabilistic framework, in which the optimal $\mathbf{L}^2$ is determined as the Maximum a Posteriori (MAP) of $p(\mathbf{L}^2|\mathbf{L}^1, \mathbf{\Phi})$, which is maximized in such a way to account for the noisiness that could affect the similarity metric $\mathbf{\Phi}$. According to Bayes' theorem,

$$\begin{aligned} \underset{\mathbf{L}^2}{\operatorname{argmax}} \ p(\mathbf{L}^2|\mathbf{L}^1, \mathbf{\Phi}) &= \underset{\mathbf{L}^2}{\operatorname{argmax}} \ \frac{p(\mathbf{\Phi}|\mathbf{L}^1, \mathbf{L}^2) \cdot p(\mathbf{L}^1, \mathbf{L}^2)}{p(\mathbf{\Phi})} \\ &= \underset{\mathbf{L}^2}{\operatorname{argmax}} \ p(\mathbf{\Phi}|\mathbf{L}^1, \mathbf{L}^2) \cdot p(\mathbf{L}^1, \mathbf{L}^2), \end{aligned}$$

because $p(\mathbf{\Phi})$ is independent of $\mathbf{L}^2$ and thus does not influence the optimization. While the first term represents a data-fidelity between the elements of $\mathbf{S}^2$ and those of $\mathbf{S}^1$, the second one expresses the joint probability distribution of $\mathbf{L}^2$. To formulate the MAP as an energy optimization problem, we propose to express the two terms as Gibbs distributions[1]:

$$\begin{aligned} p(\mathbf{\Phi}|\mathbf{L}^1,\mathbf{L}^2) &= \frac{1}{\mathcal{Z}_{E_F}} e^{-E_F\left(\mathbf{L}^1=\{l_1^1=1,\cdots,l_{N_1}^1=N_1\},\mathbf{L}^2,\mathbf{\Phi}\right)} \\ &\triangleq \frac{1}{\mathcal{Z}_{E_F}} e^{-\left(1-F\left(\mathbf{L}^1=\{l_1^1=1,\cdots,l_{N_1}^1=N_1\},\mathbf{L}^2,\mathbf{\Phi}\right)\right)} \\ p(\mathbf{L}^1,\mathbf{L}^2) &= \frac{1}{\mathcal{Z}_{\lambda E_G}} e^{-\lambda\cdot E_G(\mathbf{L}^1=\{l_1^1=1,\cdots,l_{N_1}^1=N_1\},\mathbf{L}^2)} \\ &\triangleq \frac{1}{\mathcal{Z}_{\lambda E_G}} e^{-\lambda\cdot\left(1-G\left(\mathbf{L}^1=\{l_1^1=1,\cdots,l_{N_1}^1=N_1\},\mathbf{L}^2\right)\right)}, \end{aligned}$$

where $\lambda \in \mathbb{R}^+$, $F\left(\mathbf{L}^1,\mathbf{L}^2,\mathbf{\Phi}\right) \in \mathbb{R}$ expresses the cost of assigning the set of labels $\mathbf{L}^2$ to $\mathbf{S}^2$ given the fact that the labels $\mathbf{L}^1 = \{l_1^1 = 1,\cdots,l_{N_1}^1 = N_1\}$ have been assigned to $\mathbf{S}^1$, $G\left(\mathbf{L}^1,\mathbf{L}^2\right) \in \mathbb{R}$ expresses a penalty of assigning a certain labelling $\mathbf{L}^2$, given the (ordered) sequence of labels defined by $\mathbf{L}^1$. $\mathcal{Z}_{E_F}$ and $\mathcal{Z}_{\lambda E_G}$ are partition functions. The MAP is thus given by:

$$\underset{\mathbf{L}^2}{\operatorname{argmax}}\ p(\mathbf{\Phi}|\mathbf{L}^1,\mathbf{L}^2)\cdot p(\mathbf{L}^1,\mathbf{L}^2) = \underset{\mathbf{L}^2}{\operatorname{argmax}}\ F(\mathbf{L}^1,\mathbf{L}^2,\mathbf{\Phi}) + \lambda\cdot G(\mathbf{L}^1,\mathbf{L}^2). \quad (3.1)$$

Both the data-fidelity $F(\mathbf{L}^1,\mathbf{L}^2,\mathbf{\Phi})$ and the regularization term $G(\mathbf{L}^1,\mathbf{L}^2)$ are defined in the following section.

3.4 Data-fidelity and order regularizer

We propose to define $F(\mathbf{L}^1,\mathbf{L}^2,\mathbf{\Phi})$ as a data-fidelity function that (i) enforces similar elements to receive the same labels, while (ii) pushing towards the assignment of the empty label for the elements of $\mathbf{S}^2$ that have a strong dissimilarity $(1-\max_{j=\{1,\cdots,N_1\}}\mathbf{\Phi}(\mathbf{s}_j^1,\mathbf{s}_k^2))$ with all the possible corresponding elements of $\mathbf{S}^1$. This is done by defining $F(\mathbf{L}^1,\mathbf{L}^2,\mathbf{\Phi})$ (notated as F) based on a weighted sum of these two criteria:

$$F = \sum_{k=1}^{N_2}\Bigg(\frac{\delta(l_k^2\neq\emptyset)}{\sum_{j=1}^{N_1}\delta_k(l_j^1)}\cdot\left(\sum_{j=1}^{N_1}\delta_k(l_j^1)\cdot\mathbf{\Phi}(\mathbf{s}_j^1,\mathbf{s}_k^2)\right) + \underbrace{\beta\cdot\delta_k(\emptyset)\cdot\left(1-\max_{j=\{1,\cdots,N_1\}}\mathbf{\Phi}(\mathbf{s}_j^1,\mathbf{s}_k^2)\right)}_{\triangleq F_k}\Bigg),$$

[1] As we will see in the next section, the two processes driving the label assignment, either based on descriptor similarity between the two images or based on local ordering within an image, can be considered as being Markovian in the sense that the probability of assigning a label to a given element primarily depends on a limited number of labels assigned to the other image or to the current image, respectively. According to the Hammersley-Clifford theorem [122], any probability measure that satisfies a Markov property is a Gibbs measure.

where $\delta(.)$ denotes the indicator function, $\delta_k(l_j^1) \triangleq \delta(l_k^2 = l_j^1)$ indicates if the k^{th} element of $\mathbf{S}^2$, *i.e.*, $\mathbf{s}_k^2$, has the same label than the j^{th} element of $\mathbf{S}^1$, *i.e.*, $\mathbf{s}_j^1$, and the parameter $\beta \in \mathbb{R}$ increases with the willingness to tolerate empty label assignment.

$G(\mathbf{L}^1, \mathbf{L}^2)$ is defined as a regularizer of the order of the sequences, which favors uncrossed associations, *i.e.*, avoids that $\mathbf{s}_j^1$ is associated with $\mathbf{s}_k^2$ knowing that $\mathbf{s}_{j+x}^1$ is associated to $\mathbf{s}_{k-y}^2$ ($x \in \{0, \cdots, N_1 - j\}$ and $y \in \{0, \cdots, k-1\}$). There exist two types of crossed associations. The first type, named *weakly crossed association*, comes when two elements of a sequence correspond to the same element in the other sequence, *i.e.*, when $x = 0$. The correspondences $\{\mathbf{s}_3^1 = \text{'b'}, \mathbf{s}_3^2 = \text{'b'}\}$ and $\{\mathbf{s}_3^1 = \text{'b'}, \mathbf{s}_4^2 = \text{'b'}\}$ in Figure 3.2 illustrate a *weakly crossed association*. The second type of crossed association, named *strongly crossed association*, occurs when $x \in \{1, N_1 - j\}$. The set of correspondences $\{\{\mathbf{s}_4^1 = \text{'c'}, \mathbf{s}_6^2 = \text{'c'}\}, \{\mathbf{s}_5^1 = \text{'e'}, \mathbf{s}_5^2 = \text{'e'}\}\}$ in Figure 3.2 describes a *strongly crossed association*.

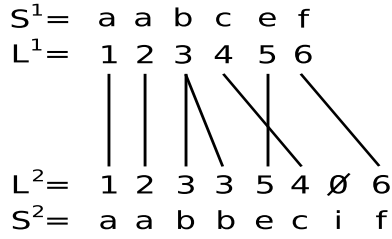


Figure 3.2: The *relaxed ordering constraint* tends to minimize the *weakly* (*e.g.*, involving 'b' letters) and *strongly crossed associations* (*e.g.*, involving 'c' and 'e' letters).

We propose to quantify the order of a sequence by associating, to each label $l_k^2 \in \mathbf{L}^2$, a measure $\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2)$ that counts the number of labels in $\mathbf{L}^2$ that do not induce a *strongly crossed association* with the k^{th} element of $\mathbf{S}^2$. As defined below, $\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2)$ measures thus how much the chosen label l_k^2 pushes $\mathbf{L}^2$ toward an ordered sequence:

$$\omega_k^{\text{ord}} \triangleq (1-\theta) \cdot \underbrace{\left(\sum_{n=1}^{k-1} o_{k>n}(l_k^2, l_n^2, \mathbf{L}^1) + \sum_{n=k+1}^{N_2} o_{k<n}(l_k^2, l_n^2, \mathbf{L}^1) \right)}_{\text{Ordering consistency}} + \theta \cdot \underbrace{\left(\sum_{n=1}^{k-1} \delta(l_k^2 = l_n^2) + \sum_{n=k+1}^{N_2} \delta(l_k^2 = l_n^2) \right)}_{\text{Ordering equality}}. \quad (3.2)$$

The parameter $\theta \in [0; 1]$ promotes the assignment of the same label to two elements of $\mathbf{S}^2$, and $o_{k>n}(l_k^2, l_n^2, \mathbf{L}^1)$ (respectively $o_{k<n}(l_k^2, l_n^2, \mathbf{L}^1)$), noted as $o_{k>n}$ (respectively $o_{k<n}$) for notation convenience, are defined only $\forall k \mid l_k^2 \neq \emptyset$, as:

$$o_{k>n} = \delta(l_k^2 > l_n^2) \cdot \sum_{i=1}^{N_1} \sum_{m=1}^{N_1} \delta(i > m \;\&\; l_i^1 = l_k^2 \;\&\; l_m^1 = l_n^2)$$
$$o_{k<n} = \delta(l_k^2 < l_n^2) \cdot \sum_{i=1}^{N_1} \sum_{m=1}^{N_1} \delta(i < m \;\&\; l_i^1 = l_k^2 \;\&\; l_m^1 = l_n^2).$$

The function $o_{k>n}$ (respectively $o_{k<n}$) counts how many matches of the n^{th} element of $\mathbf{S}^2$ do not cross the matches of the k^{th} element of $\mathbf{S}^2$, with $k > n$ (respectively $k < n$). For example, if we focus on the $k = 5^{\text{th}}$ element of the $\mathbf{S}^2$ sequence illustrated in Figure 3.2, *i.e.*, the element 'e', the ordering consistency of this element is decomposed into $\sum_{n=1}^{5-1} o_{5>n} = 4$, because 4 of its left elements do not cross its association with $\mathbf{s}_5^1$, and $\sum_{n=5+1}^{8} o_{5<n} = 1$ because, among all the matched elements at the right of $\mathbf{s}_5^2$, only one does not cross its association with $\mathbf{s}_5^1$.

We propose to use $\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2)$ to measure how much the chosen label l_k^2 favors the ordering, compared to all the other possible labels (from $\mathbf{L}^1$) that could be assigned to $\mathbf{s}_k^2$. This gain is expressed as $\left(\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2) - \mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]\right)$, where the operator $\mathbf{L} \circ_k l$ represents the assignment of the label l to the k^{th} component of $\mathbf{L}$, and $\mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]$ represents the mean (expected) cost when l is a uniform random variable. The higher this gain, the more assigning l_k^2 to the k^{th} element of $\mathbf{S}^2$ favors the ordering compared to any other label $l \in \{1, \cdots, N_1\}$. Finally, we propose to measure the ordering regularizer $G(\mathbf{L}^1, \mathbf{L}^2)$ of an entire labelling $\mathbf{L}^2$ as the (normalized) sum of these gains:

$$G = \sum_{k=1}^{N_2} \underbrace{\frac{\delta(l_k^2 \neq \emptyset) \cdot \left(\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2) - \mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]\right)}{\sum_{n \neq k} \delta(l_n^2 \neq \emptyset)}}_{\triangleq G_k}.$$

In summary, while maximizing the data-fidelity term $F(\mathbf{L}^1, \mathbf{L}^2, \mathbf{\Phi})$ pushes towards the association of each element of $\mathbf{S}^1$ to an element of $\mathbf{S}^2$ or to the empty set $\emptyset$, the maximization of $G(\mathbf{L}^1, \mathbf{L}^2)$, that corresponds to our *relaxed ordering constraint*, tends to reject the *crossed associations*. The importance of the regularization term over the data-fidelity term is steered by the parameter λ, such that the objective function is defined as:

$$\begin{aligned} f(\mathbf{L}^1, \mathbf{L}^2, \mathbf{\Phi}) &= F(\mathbf{L}^1, \mathbf{L}^2, \mathbf{\Phi}) + \lambda \cdot G(\mathbf{L}^1, \mathbf{L}^2) \\ &= \sum_{k=1}^{N_2} \underbrace{(F_k + \lambda \cdot G_k)}_{\triangleq f_k}. \end{aligned} \tag{3.3}$$

The objective function f describes thus two trade-offs, respectively controlled by the factors β and λ: a first one between the assignment of an element $\mathbf{s}_k^2$ to the empty set or its matching with an element of $\mathbf{S}^1$, and a second one to balance data-fidelity and ordering.

3.5 Optimization

Determining the optimal $\mathbf{L}^2$ among the $(|\mathcal{L}|+1)^{N_2} = (N_1+1)^{N_2}$ possibilities is a combinatorial problem. Our problem can be formulated as the optimization of a 3rd degree binary objective function, subject to equality constraints. A possible approach to solve this problem consists in turning the discrete optimization problem into a continuous optimization problem (*e.g.*, by relaxing the discrete labels into a labelling likelihood). However, the transformed (continuous) problem usually has an astronomically large number of local minima, making its global optimization challenging [159]. Because state-of-the-art methods, such as the combination of smoothing [99] with logarithmic barrier functions [57], do not guarantee to reach the global optimum, and because their convergence speed might be very slow, we have tested a simple coordinate descent optimization scheme, and demonstrate both its effectiveness and its efficiency in Section 3.6. We propose, as initial labelling, the one that maximizes the similarity. At each iteration, our optimization method selects an element $\mathbf{s}_k^2$ of $\mathbf{S}^2$, and computes the label $l_k^2 \in \{\mathbf{L}^1 \cup \emptyset\}$ that maximizes f (cfr. Equation (3.3)), the other labels $l^2_{i|i\neq k}$ being fixed. Fixing these labels makes the continuous version of each coordinate descent iteration a second degree problem. To accelerate the coordinate descent convergence process, we propose an adaptive scheduling strategy to avoid iterating over the elements that are less likely to improve the objective function. In practice, this is done by selecting the elements $\mathbf{s}_k^2$ randomly, according to a probability density function that takes the outcome of previous iterations into account to favor the selection of elements that most likely increase the objective function f. Those elements are the ones (i) that only bring a small contribution to f[2], *i.e.*, the ones for which $f_k = F_k + \lambda \cdot G_k$, as defined in Equation (3.3), is small, (ii) that have not been considered by a recent iteration of the coordinate descent process, and (iii) whose latest label optimization has led to an improvement of the objective function. As detailed in the Algorithm pseudo-code, this is achieved by defining the probability of selecting an element $k^\star$ to be inversely proportional to its gain $f_{k^\star}$, and by using a factor $(1-\exp(-\alpha_k \cdot (t' - t_k))$ (where t' represents the index of the current iteration and t_k the one of the previous process of the element k) to penalize the elements that have been recently investigated ($(t'-t_k)$ is small), especially when it failed to increase the objective f (α_k becomes smaller). The optimization stops once the system is in a steady state, *i.e.*, once no single label change can improve the objective function. The default value for the α_k parameters has been set in such a way that $(1-\exp(-\alpha_k \cdot (t'-t_k)) \approx 1$ until approximatively γ% other elements of $\mathbf{S}^2$ have been processed, *i.e.*, as long as $(t'-t_k) \leq \gamma \cdot N_2$. This is to make sure that enough other elements of $\mathbf{S}^2$ have been processed before coming back to a previously processed coordinate. Choosing $\gamma \cdot N_2 = \frac{4}{\alpha_k}$, *i.e* $\alpha_k = \frac{4}{\gamma \cdot N_2}$, fulfills this criterion, because $(1-\exp(-\alpha_k \cdot (t'-t_k))) = (1-\exp(-4)) \approx 0.98$.

[2]Because our coordinate descent optimization may update l_k^2 at each iteration, the value of f, defined by Equation (3.3), changes through time.

Algorithm Mapping under *relaxed ordering constraint*

Input: Matrix of similarities $\mathbf{\Phi} \in [0,\ 1]^{N_1 \times N_2}$
β, λ, θ and γ parameters (see text for details)
The function $f_k(\mathbf{L}^2) \leftarrow F_k + \lambda \cdot G_k$ (cfr. Equation 3.3)
Output: A surjective mapping $\mathbf{L}^2 \mapsto \{\mathbf{L}^1 \cup \emptyset\}$

Initialize: $\{l_j^1\}_{j=1}^{N_1} \leftarrow j; \quad \mathcal{E} = \{1, \cdots, N_2\}$
$\mathbf{L}^2 \leftarrow \{l_k^2 = \text{argmax}_{l \in \mathbf{L}^1}\ \mathbf{\Phi}\left(\mathbf{s}_l^1, \mathbf{s}_k^2\right)\}_{k=1}^{N_2}$;
$t' \leftarrow 0$; $\{t_k\}_{k=1}^{N_2} \leftarrow t'$; $\{\alpha_k\}_{k=1}^{N_2} \leftarrow \frac{4}{\gamma \cdot N_2}$;

while $\mathcal{E} \neq \emptyset$ **do**
  $t' \leftarrow t' + 1$;
  Evaluate $\{f_k(\mathbf{L}^2)\}_{k=1}^{N_2}$
  $\mathcal{P}_k \sim (f_k)^{-1} \cdot \left(1 - e^{-\alpha_k (t' - t_k)}\right) \qquad \forall k \ \in \ \mathcal{E}$
  $k^\star \leftarrow \text{argmin}_{k \in \mathcal{E}} \left| r - \sum_{k'=1}^{k} \mathcal{P}_{k'} \right|$ with $r \sim \mathcal{U}(0,1)$
  $l^\star \leftarrow \text{argmax}_{l \in \{\mathbf{L}^1 \cup \emptyset\}}\ f(\mathbf{L}^2 \ \circ_{k^\star} \ l)$
  if $l^\star \neq l_{k^\star}^2$ **then**
    $l_{k^\star}^2 \leftarrow l^\star$
    $\mathcal{E} \leftarrow \mathcal{E} \setminus k^\star$
    $\alpha_{k^\star} \leftarrow \frac{4}{\gamma \cdot N_2}$
    $t_{k^\star} \leftarrow t'$
  else
    $\mathcal{E} = \{1, \cdots, N_2\}$
    $\alpha_{k^\star} \leftarrow \alpha_{k^\star} / 2$
  end if
end while

The computational bottleneck of the algorithm is the estimation of G_k, which requires, for each $k \in [1, \cdots, N_2]$, (at most) $N_1 \times N_2$ estimations of ω_k^{ord}, due to the term $\mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]$. We have however observed that the normalized mean value $\frac{\mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]}{\sum_{n \neq k} \delta(l_n^2 \neq \emptyset)}$ is constant, and close to 0.5, whatever the choice of the θ parameter. This is due to the fact that, on average, low-value (respectively high-value) labels l make $\frac{\mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]}{\sum_{n \neq k} \delta(l_n^2 \neq \emptyset)}$ tend to 1 when k points to an element at the beginning (respectively at the end) of the sequence, while it tends towards 0 when k represents an element at the end (respectively at the beginning) of the sequence. $\frac{\mathbb{E}_l\left[\omega_k^{\text{ord}}(\mathbf{L}^1, \mathbf{L}^2 \circ_k l)\right]}{\sum_{n \neq k} \delta(l_n^2 \neq \emptyset)}$ is thus replaced by 0.5 in the estimation of G_k.

3.6 Validations

Our validation section is organized in three parts. The first two ones consider synthetic 3D scenes. Considering synthetic data allows us to compare the determined correspondences with the ground-truth correspondences. Using those data, we respectively assess (i) the benefit of introducing the *soft ordering term* in our objective function and (ii) the effectiveness of our proposed optimization scheme, compared to a brute force exhaustive search. We also show that our method, which uses the *relaxed ordering constraint* as regularizer, outperforms the well-known dynamic programming method, which implements the *strict ordering constraint*. In the third part of our validation, dense correspondence maps (depth maps) are generated from well-known wide-baseline real-life images. It demonstrates the relevance of our scheme in real life scenarios.

To generate the synthetic 3D scenes considered in the first and second parts of our validation, we simulate a stereo camera setup observing a set of fronto-planar 3D surfaces of uniform color. Multiple piecewise planar 3D scenes are synthesized by randomly placing 3D planes (of random sizes) parallel to the images'planes, in a cube of unit length, illustrated in Figure 3.3. Cameras are located in $X = 0.5$, $Y = 0.25$, $Z = 1$ and $X = 0.5$, $Y = 0.75$, $Z = 1$. Their intrinsic parameters have been fixed in such a way that the cameras' field of views cover $\frac{3}{4}$ of the 3D space. Without loss of generality, we assume that the stereo pair is rectified.

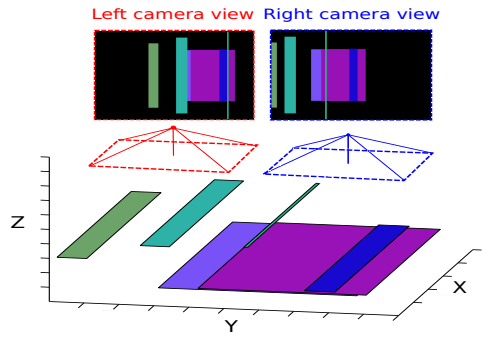


Figure 3.3: The objective function of Equation (3.3) is validated on synthetic wide-baseline fronto-planar 3D scenes.

To simulate the occlusions and ordering violations that appear when observing a 3D scene with a wide-baseline setup, the depth of the closest plane is chosen randomly but is forced to be at most twice the cameras' baseline. The depth of the other planes is chosen randomly, using an uniform distribution beyond the closest plane's depth.

To drive our matching based on colorimetric (dis)similarities, we propose to simply describe each of the planes by its *Lab* color, and to define $\mathbf{\Phi}$ by the (normalized) *CIEDE2000* [187] color similarity. The discriminativeness

of this metric makes it considered as the state-of-the-art perceptual color difference [167]. The similarity $\mathbf{\Phi}$ represents thus an "ideal" measure of correspondence, which is almost maximum when two elements match and minimum otherwise. In practice, only (photometric and geometric) invariant descriptors can provide such an "ideal" measure. However, as explained in Section 2.4, no dense (pixel) invariance to the real word geometry has been established until now, meaning that the perspective-robust descriptors currently used in wide-baseline provide a "noisy" version of this "ideal" similarity/dissimilarity measure. We propose to simulate this inaccuracy by adding white gaussian noise to this "ideal" similarity measure $\mathbf{\Phi}$, and to measure the noise level based on the SNR.
Given the above described synthetic dataset, as a first part of our validation, Figure 3.4 compares four 1D matching methods based on the percentage of elements of one image that have been correctly assigned to the second image or to the empty set (*i.e.*, detected as occluded), when the similarity metric $\mathbf{\Phi}$ is corrupted by such additive white gaussian noise. The four methods are the conventional dynamic programming approach, which forces strict order preservation [40], and three variants of our proposed scheme, respectively considering as an objective function only the similarity term in F, the whole function F, and the function f defined in Equation (3.3). To fairly compare the 4 methods, we always have selected, among a grid set of possible parameters (β, λ, θ, γ and the occlusion parameter of dynamic programming [40]) the ones that maximize the performances. This experiment has shown that the parameters $\beta = 5$, $\lambda = 1$, $\gamma = 1$ and $\theta = 0.49$ is a good choice for wide-baseline setups. The 0.49 value of the last parameter can be intuitively explained by the fact that (i) there is no clue about the presence or absence of foreshortening effect on an arbitrary element (which incites to tolerate the repetition of labels and thus $\theta \approx 0.5$), and (ii) it is slightly preferable to set θ below 0.5 to favor one to one mappings when possible. For a fair validation of our algorithm, this set of parameters is used in all our future validations, although we advice the reader to adapt them according to the observed scene (*e.g.* the likelihood of having occluded elements, of violating the ordering, etc.).

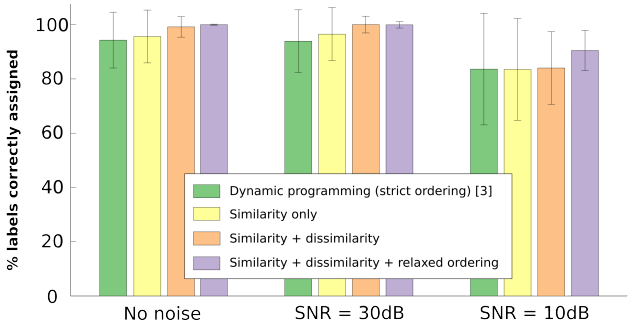


Figure 3.4: Average and standard deviation percentages (on 1200 runs) of correct labels, *i.e.*, correspondences or occlusions, when adding white gaussian noise to the similarity measure (to model the inaccuracy of photometric/geometric robust descriptors).

Figure 3.4 shows that, as soon as the similarity measure becomes inaccurate[3], *i.e.*, when the noise increases, the *relaxed ordering constraint* significantly helps to disambiguate the matching. Moreover, it does it much better than the strict preservation of the order implemented by dynamic programming [40].

In the second part of our validation, we still rely on the synthetic dataset to evaluate the effectiveness of the proposed optimization scheme. This is done by comparing the accuracy of the matching resulting from our coordinate descent approach to the one measured for the global optimum, obtained based on an exhaustive search. As shown in Table 3.1, the proposed optimization scheme reaches performances that are always close to the ones obtained by the global optimum.

	No noise	SNR = 30dB	SNR = 10dB
Brute-force search	$100 \pm 0\%$	$99.9 \pm 1.2\%$	$90.9 \pm 7.4\%$
Optimization scheme	$99.7 \pm 2.9\%$	$99.4 \pm 3.8\%$	$85.9 \pm 11.3\%$

Table 3.1: Our optimization scheme reaches performances that are close to the global optimum (determined by a brute-force search), while being computationally affordable.

This gives credit to the use of our coordinate descent strategy.

In the third and last part of our validation, we determine a dense correspondence between well-known pairs of wide-baseline stereo images [206], and convert them into (epipolary rectified) depth maps, such as illustrated in Figure 3.5. We highlight the fact that these results have been obtained based only on the two presented wide-baseline views. The last row of this figure illustrates the similarity measure between two corresponding epipolar lines of one of these wide-baseline stereo images, and respectively, from left to right, the ground-truth mapping, the mapping obtained by dynamic programming [40] and the labelling found after the convergence of our optimization scheme. As shown on this row, the strict preservation of the ordering constraint (such as done by dynamic programming [40]), which imposes that the mapping from $\mathbf{L}^2$ to $\mathbf{L}^1$ forms a strictly decreasing function, is not valid in wide-baseline conditions. By relaxing this prior, our method is adapted to wide-baseline conditions, and effectively approximates the ground-truth mapping.

[3]The trade-off between accuracy and robustness implies that $\mathbf{\Phi}$ is often chosen to be inaccurate, in such a way to robustify the matching (*e.g.*, by describing an element with an (scale, rotation, etc.) invariant descriptor).

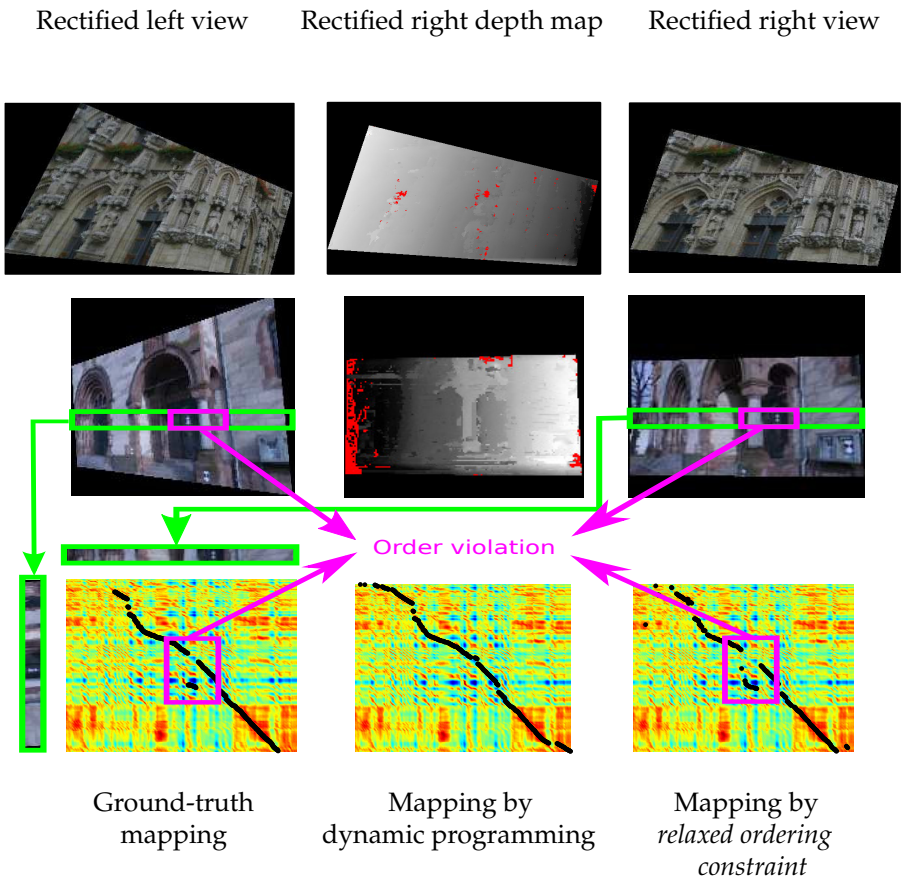


Figure 3.5: Dense depth maps (epipolary rectified) determined by our algorithm on well-known wide-baseline images provided by [206], based only on the two presented images. The red color represents the pixels that have been labelled as occluded. Structures can be easily observed when zooming in the figure. The color maps on the last row represents the pairwise similarity $\mathbf{\Phi}$ between the Daisy descriptors [218] captured along corresponding epipolar lines (at the center of the green rectangles). From the left to right image on this row, the black dots respectively present: the (manually generated) ground-truth mapping, the mapping determined by dynamic programming (which is incorrect since the ordering constraint is not strictly preserved in this scene (see the pink rectangle)) and the mapping found by our method, based on the *soft ordering* regularization.

3.7 Conclusion

The relative position between the elements constituting a 3D scene provides relevant informations that can be used to disambiguate the ill-posed problem of stereo matching. The strict preservation of the left-right order, such as imposed by dynamic programming in [40], appears to be a valid hypothesis

when the scene is observed from two very close viewpoints (narrow-baseline setups). However, this assumption becomes more and more violated as the viewpoints spread apart. This is the reason why this information has been rarely exploited until now in wide-baseline setups. To cope with this limitation, while still exploiting the relative ordering of the scene's elements, this chapter has introduced a framework for promoting the order preservation, without forcing it. The proposed approach computes dense correspondences in a wide-baseline stereo setup by maximizing the similarity of the elements captured along corresponding epipolar lines, while favoring correspondences that preserve the order, without necessary imposing it. The occlusions and the foreshortening effect, widely present in wide-baseline setups, are explicitly taken into account by our method. We have shown, on synthetic images, that the correspondences found by maximizing the proposed objective function are up to 10% more accurate than the ones obtained by dynamic programming. Because our formulation results in a combinatorial optimization problem, we have proposed an iterative optimization solution that has shown to reach close to the optimal performances, while being computationally tractable, allowing us to densely reconstruct the depth of well-known wide-baseline stereo datasets. Exhaustive validations on synthetic datasets have demonstrated quantitatively the benefit resulting from our framework in terms of matching accuracy, as well as the effectiveness of our proposed optimizer. Research questions include the inclusion of constraints favoring matching consistency across epipolar lines, and the combination of our relaxed ordering constraint with methods favoring the piecewise smoothness of the depth map.

CHAPTER 4

Automatic piecewise-planar 3D approximation from wide-baseline stereo

While the previous chapter has targeted the simultaneously reconstruction of both the background and the foreground of a 3D scene, this chapter focuses on the background reconstruction, assuming that its 3D can be reasonably approximated by the juxtaposition of a limited number of 3D plane models. It proposes a novel method that requires only two wide-baseline views to approximate the 3D model of a man-made scene[1] by a minimum number of 3D planes. Our method relies on the over-segmentation of one of the two reference images, and adopts a hypothesis testing process to assign a 3D plane to each region, when the region is not detected as occluded in the second view. It first produces a tremendous amount of 3D plane candidates, derived from 3D point triplets randomly picked in a dense but noisy 3D point cloud. It then extracts, among the set of plane candidates, a small number of plane hypotheses that correctly approximates the 2D regions delimited by the projection of each 3D triplet in the two reference views. Then, the reconstruction is formulated as an energy-driven plane-to-region assignment problem, which simultaneously optimizes a data-fidelity term, the labeling smoothness, and the number of assigned planar proxies. Targeting a minimal number of 3D planes guarantees a light-weight representation of the 3D scene. As another original contribution, we propose a novel data-fidelity term, that weights the 3D fitting error according to the accuracy and non-ambiguity of the reprojection of the region, via the investigated 3D plane, in the other reference view. To validate our approach, we generate free-viewpoints around 10 well-known wide-baseline datasets to demonstrate that our light-weight, piecewise-planar 3D reconstruction method approximates correctly the 3D of the scene.

[1] A man-made scene is composed of manufactured 3D objects, which are observed by real cameras.

4.1 Introduction

The estimation of the 3D model of a scene is an ill-posed problem. State-of-the-art Multi-View Stereo (MVS) methods generally disambiguate this reconstruction by relying on many views of the scene, captured by a small-baseline stereo network. However, due to physical constraints, it is not always possible to set up many cameras around the scene. Reducing this number of cameras does not only increase the number of challenging occluded areas, but also leads to stronger (projective) geometrical changes between the views. Beside occlusions and strong perspective changes, textureless and repetitive patterns might also lead to holes in a dense depth map. This is due to the difficulty of determining correspondences when there is a lack of specific/discriminative visual cues in the region. MVS methods that can suppress these artifacts via strong regularization generally tend to oversmooth the 3D surface [21] [25], especially due to the propagation of regularization constraints through adjacent 2D regions that correspond to discontinuous 3D surfaces.

Motivated by these drawbacks, this chapter considers a dense reconstruction, but limits itself to the reconstruction of 3D scenes that exhibit a piecewise-planar geometry. Such geometry is often encountered in man-made background scenes observed from the ground-level. Our piecewise-planar reconstruction is formulated as a 3D plane assignment problem over 2D regions, obtained based on a fast color segmentation [229] of one of the two reference images. We call this image the *source* image. In contrast to most previous works dealing with wide-baseline setups [21] [4] [191], our method builds upon a dense 3D point cloud[2], instead of a sparse set of correspondences between keypoints. Although dense point clouds are generally much more corrupted by noise and 3D outliers than sparse ones, they provide (noisy) cues about the 3D of challenging surfaces, *e.g.*, textureless or with repetitive patterns such as paved floors. Because such dense 3D point clouds are strongly affected by outliers, our method does not directly fit planes on these data, but rather proposes a set of planar hypotheses, and successively refines them while discarding the less reliable ones. We name this principle *plane hypothesis testing*.

The main contributions of the proposed *plane hypothesis testing* method are the following ones:

- An original approach to define a limited number of 3D plane hypotheses. The set of proposed 3D planes appears to include most of the planar surfaces composing the 3D scene, while being limited in size. The approach is presented in Section 4.4.
- A new plane-to-region data-fidelity term that modulates a region-based 3D point cloud fitting error based on the accuracy and non-ambiguity of the matching underlying the point cloud construction. Our experiments reveal that the proposed data-fidelity is robust to the numerous 3D outliers present in the dense point cloud. The definition of this new data-fidelity term is detailed in Section 4.5.
- An energy-driven formulation of the plane-to-region assignment problem, which maximizes the data-fidelity and the smoothness of the plane

[2]We define a dense point cloud as a set of 3D points whose projection fully covers (at least one of) the reference images.

assignment over the regions, while minimizing the number of assigned planes. This last term guarantees to approximate the 3D with a small set of planes, without having to fix this parameter *a priori* or to merge many similar plane models *a posteriori* [21]. To the best of our knowledge, this optimization scheme, presented in Section 4.6, is the first one to *densely* approximate the 3D of a scene based on the minimum set of 3D planes.

4.2 Related works

Previous works in 3D reconstruction of (man-made) scenes can mainly be categorized into four groups: (1) dense wide-baseline MVS (in the image domain), (2) dense small-baseline MVS (in the image domain), (3) extraction of geometric primitives from a sparse point cloud[3] and (4) extraction of geometric primitives from a dense point cloud.

Methods of the first group usually rely on multiple ($\gg 2$) wide-baseline images to estimate a dense depth map. Depth-maps are typically obtained from pairs of cameras, and then fused together for refinements [76] [247]. As a first drawback, these methods require tens [227] of views to obtain reliable correspondences, to mitigate the strong perspective and photometric changes present in wide-baseline configurations. As a second drawback, their strong smoothness regularization tends to oversmooth the depth [204] [203] [8], or even to propagate it to other surfaces when the amplitude of the image gradient is not sufficient at the surface's border [25]. As an alternative to depth maps fusion, plane-sweeping methods successively investigate multiple depth hypotheses by sweeping a plane [36] through the 3D space, either orthogonally to one of the camera's axis [12] [81] or along a few principal directions [70]. Although their GPU-based implementations demonstrate real-time performances [242] [171] [80], plane-sweeping assumes the Manhattan world hypothesis, *i.e.*, that the 3D surfaces are orthogonal to the sweeping directions.

The second group of methods relies on a few small-baseline images to estimate a dense depth/disparity map, and is often referred as dense two-frame stereo. Those methods have been evaluated and compared through the Middlebury challenge. Several of the top-ranked algorithms [181] [180] rely on image segmentation. Working at the region level has been proved to increase the robustness of the matching data-fidelity [97] [151] while effectively propagating depth information from textured to ambiguous regions [251].

The third group of methods produces sparse 3D point clouds [4] [199] [200] [198] on which 3D primitives can be fitted. The fitting can either be direct, *e.g.*, based on robust model fitting methods [58] or by fine-tuning the 3D parametrized primitives based on detection of line segments or vanishing directions [235]. Werner and Zisserman [235] proposed a fully automatic

[3]We define a sparse point cloud as a set of 3D points obtained based on the matching of discriminative 2D keypoints. In other words, none of the reference images is fully covered by the projection of the sparse point cloud onto this view.

approach that fits polyhedral models based on a sparse 3D point cloud and on a coarse prior model of the scene. However, their method requires multiple small-baseline input views. When considering views captured from a wider range of viewpoints, these fitting methods generally require manual interactions [120] to specify high-level scene information such as the relations between the 3D primitives (*e.g.*, adjacency and alignment) [160] or the regularity [225], typically present in man-made environments.
Instead of fitting directly the geometrical primitives on the 3D data, some recent works exploit the piecewise-planarity of man-made scene to choose, from a set of 3D plane candidates, the ones that support the best the 3D point cloud [191]. Global Markov-Random-Field (MRF) with multi-view constraints are then used to propagate the assignment through the pixels that are not represented by a 3D point of the sparse point cloud. To robustify the assignment, Bodis *et al.* [21] have recently proposed to lift-up the regularized assignment to the region level. A set of optimal planes are found over the regions, and the number of considered models is finally reduced, *a posteriori*, by merging the most similar ones. Their remarkable method strongly accelerates the reconstruction, from many minutes to a few seconds, due to the small amount of treated regions and their abstinence from using any expensive photoconsistency computation [152]. In practice, assigning planes to superpixels rather than to pixels suffers from a main drawback: it can not model the regions whose geometry is not described in the sparse point cloud. To adress this problem, Bodis *et al.* [21] consider large 2D regions and propagate plane models across regions, using a MRF formulation. Large 2D regions might however violate the region planarity assumption [251].
As a conclusion, methods relying on sparse 3D point clouds, such as the one of Bodis, can not approximate the 3D of a region that has no associated 3D point in the sparse cloud and has not a neighboring region with similar planar 3D. These two contraints are often met in challenging regions, such as in grass-floor planes. By relying on a dense point cloud, such as presented in the next paragraph, our method overcomes this problem.

The fourth group of methods generally relies on the piecewise-planar assumption of man-made scenes and fit multiple plane models on a dense 3D point cloud. A common problem of these methods is the trade-off between the reliability of the dense point cloud, and the amount of captured views. On one hand, the impressive work of [64] obtains a highly reliable dense point cloud by dense map fusion [247] of approximatively 3 million of images. Agarwal *et al.* reduced this number to a few hundred thousands images [3] [4]. On the other hand, Gallup *et al.* [71] use the depths obtained on ten images to fit planar hypotheses on segmented regions, based on RANSAC [58]. As illustrated on the Castle sequence [206] in Figure 4.1, independent robust fitting of 3D planes over the regions is however still too sensitive to the strong presence of noisy and/or 3D outliers (points) in a dense point cloud obtained from only two wide-baseline cameras. In their work, Gallup *et al.* propose to refine this initial set of inaccurate 3D planes, based on 10 multiview photoconsistencies as well as by learning priors about the textures and colors of the man-made scene. However, such priors, which are embeded in a classifier, still requires human interaction to be trained on application-dependent images.

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

Figure 4.1: Fitting 3D planes on a dense point cloud generated from a pair of wide-baseline images is a challenging problem due to the implicit noisiness of the associated point cloud. (a) Segmentation of the left image; (b) Right image; (c) Left-to-right view projection, via the piecewise-planar 3D model of the scene obtained by using RANSAC [58] to fit a plane to the 3D points associated to each region; (d) Left-to-right projection, when the 3D planar model is derived from the 3D points corresponding to the most accurate and unambiguous[4] 2D matches. Even in this case, RANSAC is insufficient to approximate the 3D model of the scene.

In contrast to existing solutions, our *plane hypothesis testing* method requires only two wide-baseline views to determine an accurate piecewise- planar approximation of man-made scenes. It relies on dense point cloud estimation to properly deal with the most challenging surfaces. It does not assume dominant directions, like in a Manhattan world hypothesis and does not require user interactions. Our method assigns plane hypotheses to regions, and minimizes the number of assigned models simultaneously to the plane assignment cost and the smoothness over regions, to obtain a dense piecewise-planar approximation of the 3D scene. It results in an accurate, low complexity 3D representation of the scene, perfectly adapted for light-weight storage and transmission.

4.3 Overview of the proposed method

To approximate the 3D of a scene captured by two wide-baseline reference views, our *plane hypothesis testing* method relies on (1) the proposition of a number of 3D plane hypotheses from the dense 3D point cloud computed from the two reference views, (2) the segmentation of one of the reference images into a complete set of non-overlapping regions and (3) the optimization of the plane assignment to each region. These three distinct steps are illustrated in Figure 4.2.

[4]The notions of accuracy and ambiguity of a matching are defined in Section 4.5.

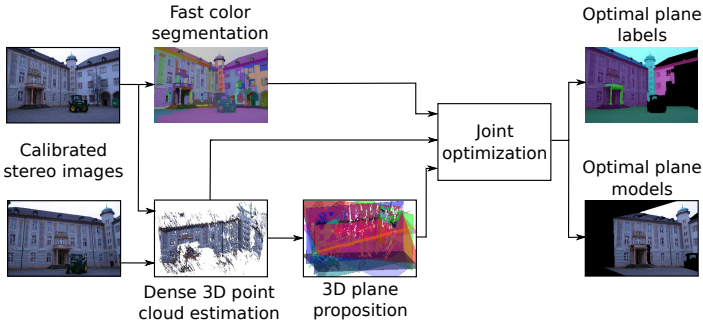


Figure 4.2: Our novel approach reconstructs a dense, piecewise-planar 3D model of the scene from only two wide-baseline image, by optimizing the assignment of proposed plane models over the image regions. The proposed models are obtained based on a dense (but noisy) 3D point cloud.

To segment the image, we rely on our previous work, presented in [229], to learn the dominant colors in the image. Given this set of C dominant colors, the segmentation problem is defined as the assignment of each pixel to one of the C classes. To impose the smoothness among neighboring pixels, this assignment problem is solved by graph-cut optimization [47], in which the data-fidelity term is defined as the ℓ_2 distance between the dominant colors and the pixel color, and the smoothness term is proportional to the inverse of the amplitude of the gradient of two neighboring pixels. This method results into a set of N regions, as illustrated in the upper block of the second column in Figure 4.2.

To propose a small set of plane candidates to describe the 3D of the scene, our method first generates a dense, but noisy 3D point cloud on which a tremendeous amount of M planes, each one supported by a random selection of three 3D points, are fitted. A measure of confidence is associated to each of these 3D planes, and exploited to reduce these M redundant (and possibly unreliable) plane candidates into $K \ll M$ reliable proposed planes. The generation of the dense 3D point cloud, as well as its associated plane proposition phase, are detailed in Section 4.4.

To assign a plane to each region, we derive a novel data-fidelity term to measure the cost of assigning the k^{th} plane to the n^{th} region. This cost measures the 3D proximity between the k^{th} 3D (plane) model and the 3D points that project within the n^{th} region or within its projection (through the k^{th} plane) onto the other reference view. We also propose to account for the consistency between the textures of those two regions by weighting the fitting error associated to the 3D points based on the accuracy and unambiguity of the matching underlying their definition. This is explained in detail in Section 4.5.

Finally, as detailed in Section 4.6, we simultaneously maximize the proposed data-fidelity and the smoothness of the plane attribution over the regions, while minimizing the amount of assigned planes based on PEARL optimization [102]. To the best of our knowledge, we are the first ones to densely reconstruct the 3D of a scene by explicitly targeting a minimum set of planar proxies.

4.4 3D planes proposition

This section describes how planar models are proposed from a dense cloud of 3D points. This is done in three consecutive steps.

In the first step, a dense 3D point cloud is generated by determining, for each pixel $\mathbf{x}$ belonging to the first view $\mathcal{I}$, the corresponding pixel $\mathbf{x}'$ in the second view $\mathcal{I}'$, and triangulating [92] these correspondences. A correspondence $\mathbf{x}'$ is determined for each $\mathbf{x} \in \mathbf{\Omega}_{\mathcal{I}}$ (where $\mathbf{\Omega}_{\mathcal{I}}$ is the spatial domain of the image $\mathcal{I}$) based on a simple "Winner-Takes-All" (WTA) [181] method, restricted to the epipolar line $\mathbf{l}' = \mathbf{F} \cdot \widetilde{\mathbf{x}}$ associated to $\mathbf{x}$:

$$\mathbf{x}' = \underset{\mathbf{y}' \in \mathbf{F} \cdot \widetilde{\mathbf{x}}}{\operatorname{argmin}} \left\| \mathbf{d}\left(\mathbf{x}\right) - \mathbf{d}\left(\mathbf{y}'\right) \right\|_2^2, \tag{4.1}$$

where $\mathbf{F}$ is the fundamental matrix of the calibrated stereo pair, $\widetilde{\mathbf{x}}$ are the homogeneous coordinates [92] of $\mathbf{x}$, $\mathbf{d}\left(\mathbf{x}\right)$ is a descriptor associated to this pixel, and $\|\cdot\|_2$ is the ℓ_2 norm. It is interesting to note that the correspondence between the 2D matched points $\mathbf{x}$ and $\mathbf{x}'$ and their associated 3D triangulated point $\mathbf{X}$ can be stored, thereby avoiding the need of costly projection in the rest of the method. In particular, note that a very fast access to the 2D to 3D correspondences is possible by storing them in a tree data-structure, where each node represents the label of the region containing the 2D point. In the rest of the paper, the term "extraction of 3D points from 2D coordinates" (or vice-versa) will refer to this type of access.

In the second step, we derive M planar models from this noisy 3D point cloud. Therefore, we randomly (uniformly) select M triplets of (non-colinear) 3D points $\mathbf{X}_t$ (with $t = \{1,2,3\}$ referring to the index of the 3D point in the Triplet) to generate M plane candidates $\boldsymbol{\pi}_m$ (with $1 \leq m \leq M$), each one parametrized as $\boldsymbol{\pi}_m = \begin{bmatrix} a_m & b_m & c_m & d_m \end{bmatrix}^\top$ to represent the plane $a_m x + b_m y + c_m z + d_m = 0$, or equivalently by

$$\boldsymbol{\pi}_m = \begin{bmatrix} a_m/d_m & b_m/d_m & c_m/d_m & 1 \end{bmatrix}^\top \triangleq \begin{bmatrix} \boldsymbol{\eta}_m^\top & 1 \end{bmatrix}^\top.$$

In the last step, we derive from the M plane candidates, a small number of $K \ll M$ planes that are expected to capture most of the representative 3D planar structures in the scene. Therefore, we first assign a quality value $q\left(\boldsymbol{\pi}_m\right)$ to each of the M plane candidates. This is done by considering the triangular patch $[\boldsymbol{\pi}_m]$ lying on the plane $\boldsymbol{\pi}_m$ and delimited by the triplet $\{\mathbf{X}_t\}_{t=\{1,2,3\}}$, such as illustrated by the red points and red patch in Figure 4.3.

The 2D region representing this triangular patch $[\boldsymbol{\pi}_m]$ in the first (respectively second) reference view is denoted $\boldsymbol{\Delta}_m$ (respectively $\boldsymbol{\Delta}'_m$), and is defined by:

$$\begin{aligned} \boldsymbol{\Delta}_m &= \left\{ \mathbf{x} \in \mathbf{\Omega}_{\mathcal{I}} \;\middle|\; \mathbf{x} \in \mathbf{P} \cdot [\boldsymbol{\pi}_m] \right\} \\ \boldsymbol{\Delta}'_m &= \left\{ \mathbf{x}' \in \mathbf{\Omega}_{\mathcal{I}'} \;\middle|\; \mathbf{x}' \in \mathbf{P}' \cdot [\boldsymbol{\pi}_m] \right\}, \end{aligned}$$

with $\mathbf{P} \in \mathbb{R}^{3\times4}$ (respectively $\mathbf{P}' \in \mathbb{R}^{3\times4}$) the projection matrix of the first (respectively second) reference view.

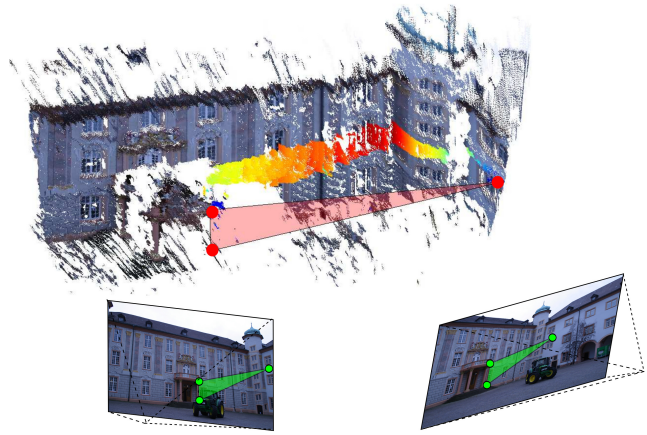


Figure 4.3: Each of the M 3D plane candidates $\boldsymbol{\pi}_m$ is generated by randomly selecting a triplet of 3D points (red points) in the dense point cloud. The quality value of a plane $\boldsymbol{\pi}_m$ is measured based on its proximity to the 3D points $\mathbf{X}$ (points in jet color) whose projections $\mathbf{P} \cdot \widetilde{\mathbf{X}}$ and $\mathbf{P}' \cdot \widetilde{\mathbf{X}}$ lie in the projections (green triangles in the figure) of $[\boldsymbol{\pi}_m]$ (red patch in the figure) in one of the reference views.

We then extract, from the point cloud, the set of 3D points $\mathbf{X}$ projecting in $\boldsymbol{\Delta}_m$ and/or in $\boldsymbol{\Delta}'_m$. For the sake of simplicity, we slightly abuse the notation in the rest of the paper and write $\mathbf{X} \in \boldsymbol{\Delta}_m$ when the projection $\mathbf{P} \cdot \widetilde{\mathbf{X}}$ of the 3D point $\mathbf{X}$ falls into the 2D triangle $\boldsymbol{\Delta}_m$. We write analogously $\mathbf{X} \in \boldsymbol{\Delta}'_m$.

Given those definitions, the proposed quality value $q(\boldsymbol{\pi}_m)$ quantifies how close is the plane candidate $\boldsymbol{\pi}_m$ from the 3D points $\mathbf{X}_j \in \{\boldsymbol{\Delta}_m \cup \boldsymbol{\Delta}'_m\}$, with $j \leq \mathcal{J}$, $\mathcal{J}$ being the number of 3D points projecting onto $\boldsymbol{\Delta}_m$ and/or $\boldsymbol{\Delta}'_m$. This is done by counting the fraction of 3D points $\mathbf{X}_j \in \{\boldsymbol{\Delta}_m \cup \boldsymbol{\Delta}'_m\}$ that are closer from the 3D plane $\boldsymbol{\pi}_m$ than a predefined threshold $T_d \in \mathbb{R}^+$:

$$q(\boldsymbol{\pi}_m) = \frac{1}{\mathcal{J}} \sum_{j=1}^{\mathcal{J}} \left(d(\boldsymbol{\pi}_m, \mathbf{X}_j) \leq T_d \right),$$

in which

$$d(\boldsymbol{\pi}_m, \mathbf{X}_j) = \frac{|\boldsymbol{\pi}_m^\top \cdot \widetilde{\mathbf{X}}_j|}{\|\boldsymbol{\eta}_m\|_2}$$

is the orthogonal distance between a 3D plane $\boldsymbol{\pi}_m$ and the 3D point $\mathbf{X}_j$.

The quality value $q(\boldsymbol{\pi}_m)$ of a candidate plane $\boldsymbol{\pi}_m$, which is the boundless version of the randomly selected triangular patch $[\boldsymbol{\pi}_m]$, is thus quantified based on the density of 3D points $\mathbf{X}_j \in \{\boldsymbol{\Delta}_m \cup \boldsymbol{\Delta}'_m\}$ surrounding it. Figure 4.4 confirms that the density of these 3D points is much higher in the neighborhood of the "ground-truth" 3D planes than around other planes. Precisely, Figure 4.4 represents the distribution of the orthogonal distance $d(\boldsymbol{\pi}_m, \mathbf{X}_j)$ between the 3D points $\mathbf{X}_j \in \{\boldsymbol{\Delta}_m \cup \boldsymbol{\Delta}'_m\}$ and the plane $\boldsymbol{\pi}_m$, for two kinds of

planes $\boldsymbol{\pi}_m$: the 3D planes approximating correctly the ground-truth 3D surface (in green), and the ones that are far away from this ground-truth (in red).

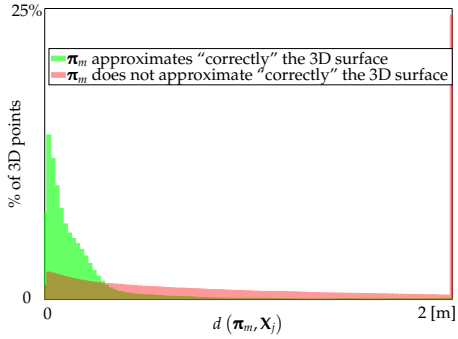


Figure 4.4: The quality $q(\boldsymbol{\pi}_m)$ of a plane candidate $\boldsymbol{\pi}_m$, associated to a randomly selected triangular patch $[\boldsymbol{\pi}_m]$, is quantified based on the fraction of 3D points $\mathbf{X}_j \in \{\boldsymbol{\Delta}_m \cup \boldsymbol{\Delta}'_m\}$ that are close to this 3D plane. We validate the relevance of this measure by showing that there are much more 3D points $\mathbf{X}_j$ close to the "ground-truth" (definition detailed in the text) 3D planes (green histogram) than around any arbitrary other one (red histogram).

In this figure, a plane candidate $\boldsymbol{\pi}_m$ is considered to approximate a ground-truth plane $\boldsymbol{\pi}_i^\star$ when its orientation is close to the one of the ground-truth plane and its distance to an arbitrary 3D point (*e.g.*, the optical center $\mathbf{C}$ of the source camera) is similar to the one of the ground-truth plane. In practice, $\boldsymbol{\pi}_m$ approximates "correctly" the ground-truth when there exists (at least) one ground-truth plane $\boldsymbol{\pi}_i^\star$ such that:

$$\Theta_{\boldsymbol{\pi}_m, \boldsymbol{\pi}_i^\star} = \left| \text{acos}\left(\frac{|\boldsymbol{\eta}_m^\top \cdot \boldsymbol{\eta}_i^\star|}{\|\boldsymbol{\eta}_m\|_2 \, \|\boldsymbol{\eta}_i^\star\|_2} \right) \right| \leq 5^\circ, \tag{4.2}$$

and

$$d(\boldsymbol{\pi}_m, \boldsymbol{\pi}_i^\star) = |d(\boldsymbol{\pi}_m, \mathbf{C}) - d(\boldsymbol{\pi}_i^\star, \mathbf{C})| \leq 50 \text{ [cm]}. \tag{4.3}$$

The concentration of the green distribution around small distance values in Figure 4.4, which has been generated from $M = 200000$ planes and with 7 ground-truth planes on the well-known Herz-Jesu-P8 dataset [206], confirms the appropriateness of the proposed plane quality value $q(\boldsymbol{\pi}_m)$.

Based on this observation, we select, from the M plane candidates $\boldsymbol{\pi}_m = \begin{bmatrix} \boldsymbol{\eta}_m^\top & 1 \end{bmatrix}^\top$, the $K \ll M$ most representative ones by applying a weighted k-means [44] on the $\boldsymbol{\eta}_m \in \mathbb{R}^3$ vectors. The weight associated to the plane candidate $\boldsymbol{\pi}_m$ in the weighted k-means is chosen to be its quality value $q(\boldsymbol{\pi}_m)$.

In summary, although we initially generate a tremendeous amount of M plane candidates to guarantee that this random selection includes the 3D ground-truth, our method avoids to compute $M \cdot N$ plane/region association metrics (with N being the number of regions), by reducing this computation to $K \cdot N$, with $K \ll M$. The proposed association metric is detailed in the next section.

4.5 Cost of assigning a 3D plane to a 2D region

This section proposes a novel data-fidelity metric to quantify how well a given 3D plane $\boldsymbol{\pi}$ approximates the 3D point cloud associated to a 2D image region $\mathcal{R} \subseteq \boldsymbol{\Omega}_{\mathcal{I}}$. Fundamentally, our data fidelity measures the proximity between the investigated (plane) model $\boldsymbol{\pi}$ and the 3D points that project into the 2D region $\mathcal{R}$ and/or its counterpart $\mathcal{R}'$, obtained in $\mathcal{I}'$ using the homography induced by the 3D plane $\boldsymbol{\pi}$. To modulate our 3D fitting error according to the discriminativeness of the textures observed in the 2D views, our proposed fitting error accounts for the inaccuracy and the ambiguity of the 2D descriptors associations that support the 3D points definition. Interestingly, this dependency is achieved without the need for a fine and unpractical tuning of some weighting factors associated to each one of the 3D points. Instead, it relies on the geometric average of a set of values, each value being associated to a subset of 3D points resulting from a sufficiently accurate and unambiguous matching, and reflecting how close those points are to the plane model.

We now introduce the notions and notations required to support the formal definition of our data-fidelity metric.

Since our fitting error considers both the 3D points associated to the 2D region $\mathcal{R} \subseteq \boldsymbol{\Omega}_{\mathcal{I}}$ and to its counterpart in $\mathcal{I}'$, we first introduce $\mathcal{R}_{\boldsymbol{\pi}}$ to denote the set of pixels $\mathbf{x}'_j \in \boldsymbol{\Omega}_{\mathcal{I}'}$ that are associated to the projection of $\mathcal{R}$ onto $\mathcal{I}'$, via a plane $\boldsymbol{\pi} = [a \quad b \quad c \quad d]^\top$. Note that $\mathcal{R}_{\boldsymbol{\pi}}$ might not define a compact set, *i.e.*, a 2D region, due to the image discretization. This set is fully determined by the pixels $\tilde{\mathbf{x}}'_j = \mathbf{H}_{\boldsymbol{\pi}} \cdot \tilde{\mathbf{x}}_j$ with $\mathbf{x}_j \in \mathcal{R}$, based on the homography $\mathbf{H}_{\boldsymbol{\pi}} : \mathbb{R}^3 \to \mathbb{R}^3$ defined as [92]:

$$\mathbf{H}_{\boldsymbol{\pi}} = \mathbf{K}' \left(\mathbf{R} - \frac{\mathbf{t} \, [a \quad b \quad c]}{d} \right) \mathbf{K}^{-1},$$

for two cameras expressed as a stereo rig configuration, *i.e.*, respectively described by the projection matrices $\mathbf{P} = \mathbf{K} \, [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = \mathbf{K}' \, [\mathbf{R} \mid \mathbf{t}]$, where $\mathbf{K} \in \mathbb{R}^{3\times3}$ and $\mathbf{K}' \in \mathbb{R}^{3\times3}$ describe the intrisic parameters of the cameras, $\mathbf{R} \in \mathcal{SO}(3)$ is the relative rotation between the cameras, and $\mathbf{t} \in \mathbb{R}^3$ their relative translation.

To measure whether a plane $\boldsymbol{\pi}$ correctly approximates the 3D surface associated to a region $\mathcal{R}$ in image $\mathcal{I}$, we quantify how far the 3D points that project in $\mathcal{R}$ and/or in $\mathcal{R}_{\boldsymbol{\pi}} = \left\{ H_{\boldsymbol{\pi}} \cdot \tilde{\mathbf{x}}_j : \mathbf{x}_j \in \boldsymbol{\Omega}_{\mathcal{I}'} \right\}$ are from the plane $\boldsymbol{\pi}$. As explained above, we propose to account for the confidence that we have in the matching that has defined each one of these 3D points.
Formally, a matching between a pair of 2D points $\mathbf{x} \in \boldsymbol{\Omega}_{\mathcal{I}}$ and $\mathbf{x}' \in \boldsymbol{\Omega}_{\mathcal{I}'}$ is expected to be reliable when the 2D point descriptors $\mathbf{d}(\mathbf{x})$ and $\mathbf{d}(\mathbf{x}')$ are (1) very similar, and (2) quite discriminant, which means they are different from most of the alternative matches along the epipolar line, *i.e.*, different from $\mathbf{d}(\mathbf{y}')$ with $\mathbf{y}' \in \mathbf{F} \cdot \mathbf{x}$ (see Equation (4.1)). Let $\mathbf{X}$ denote the 3D point associated to the triangulation of two matched pixels $\mathbf{x}$ and $\mathbf{x}'$, as determined during the matching associated to the dense matching process described in Section 4.4. We then introduce two metrics to estimate the inaccuracy and ambiguity of

the 3D point $\mathbf{X}$. Formally,

- the matching inaccuracy, denoted by $m_i(\mathbf{X})$, measures how dissimilar are the descriptors $\mathbf{d}(\mathbf{x})$ and $\mathbf{d}(\mathbf{x}')$ of the two corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$ associated to $\mathbf{X}$. The matching inaccuracy is defined by:

$$m_i(\mathbf{X}) = \frac{1}{\mathcal{D}} \left\| \mathbf{d}\left(\mathbf{P} \cdot \widetilde{\mathbf{X}}\right) - \mathbf{d}\left(\mathbf{P}' \cdot \widetilde{\mathbf{X}}\right) \right\|_2,$$

  where $\mathcal{D}$ is the size of the descriptor used during the matching phase.

- the matching ambiguity, denoted by $m_a(\mathbf{X})$, measures the percentage of pixel candidates $\mathbf{y}' \in \mathbf{F} \cdot \widetilde{\mathbf{x}}$ satisfying

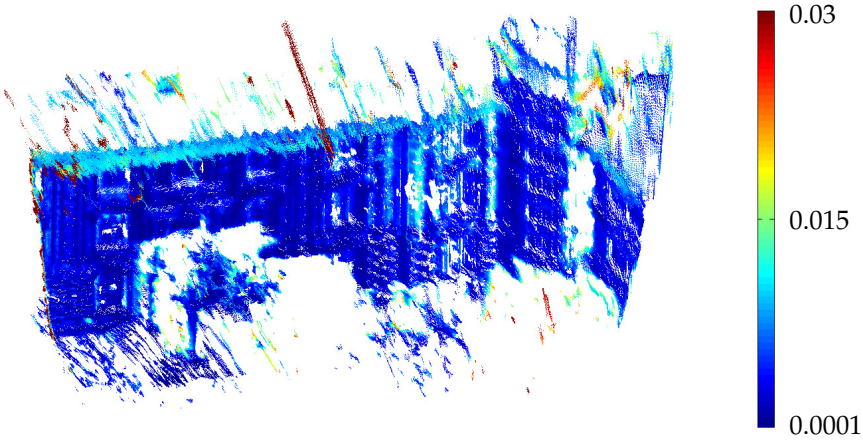
$$\frac{1}{\mathcal{D}} \left\| \mathbf{d}(\mathbf{x}) - \mathbf{d}(\mathbf{y}') \right\|_2 \leq \frac{m}{\mathcal{D}} \cdot \left\| \mathbf{d}(\mathbf{x}) - \mathbf{d}(\mathbf{x}') \right\|_2 + b,$$

  among the pixels $\mathbf{y}'$ lying on the epipolar line associated to $\mathbf{x}$. In this definition, m and b are respectively set to 1.5 and 0.002. Our experiments have revealed that these parameters do not strongly affect the performance of our method.

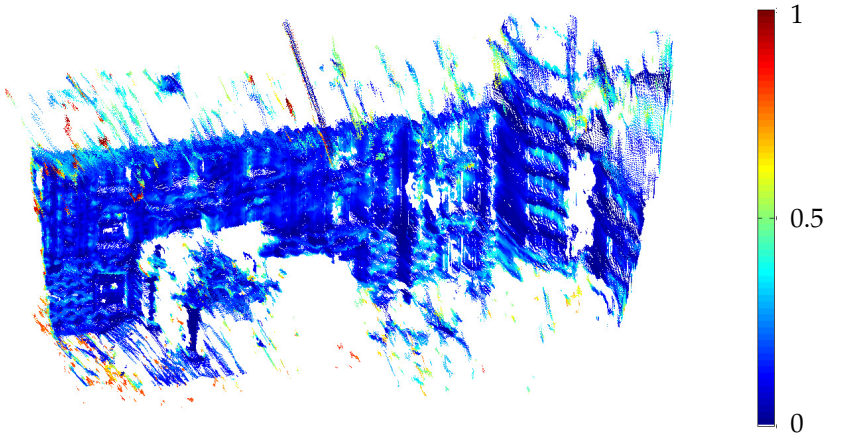
Figures 4.5(c) and 4.5(d) illustrate the matching inaccuracy $m_i(\mathbf{X})$ and matching ambiguity $m_a(\mathbf{X})$ of the 3D points in the dense point cloud associated to the well-known Castle sequence [206].

(a) Left view

(b) Right view



(c) Matching inaccuracy $m_i(\mathbf{X})$ of the 3D points $\mathbf{X}$.



(d) Matching ambiguity $m_a(\mathbf{X})$ of the 3D points $\mathbf{X}$.

Figure 4.5: The uncertainty of each 3D point $\mathbf{X}$ is quantified based on (c) the matching inaccuracy $m_i(\mathbf{X})$ and on (d) the matching ambiguity $m_a(\mathbf{X})$.

To evaluate the relevance of those metrics to decide whether a 3D point is likely to lie on the actual 3D surface of the scene (*i.e.*, being an inlier), Figure 4.6 plots their distributions for two classes of 3D points that project in a region for which a planar ground truth plane $\boldsymbol{\pi}^\star$ has been manually defined: (1) the green plot considers the "inliers" to the manual ground-truth plane $\boldsymbol{\pi}^\star$ associated to the region (*i.e.*, the $\mathbf{X}$ satisfying $d\left(\boldsymbol{\pi}^\star, \mathbf{X}\right) \leq 0.1$ [m]), and (2) the outliers (with distance $d\left(\boldsymbol{\pi}^\star, \mathbf{X}\right) > 1$ [m]) compared to this ground-truth plane.

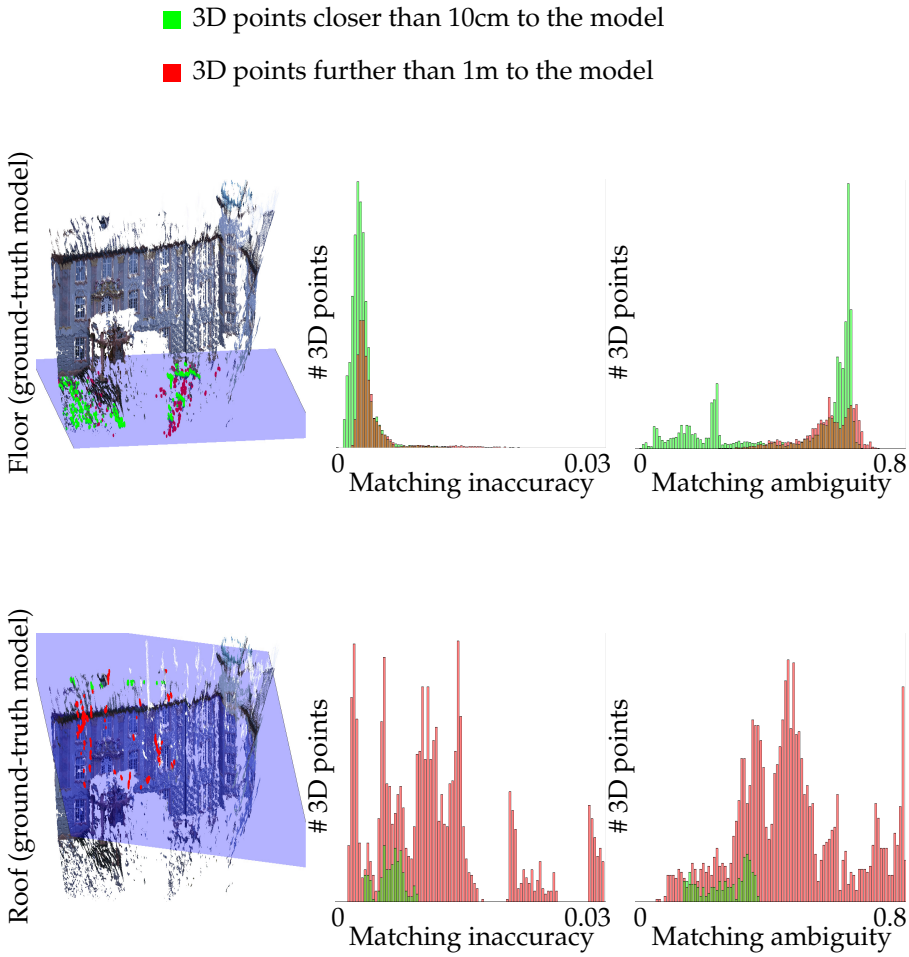


Figure 4.6: Distribution of the inaccuracy and ambiguity measures of the 3D points associated to two ground-truth 3D planar regions. The first one (floor) is textureless, while the second one (roof) is only composed of repetitive textures.

Figure 4.6 reveals that, whilst being different, the inliers and outliers distributions largely overlap each others. This prevents the accurate classification of the 3D points into an inlier and outlier class based on those two metrics.

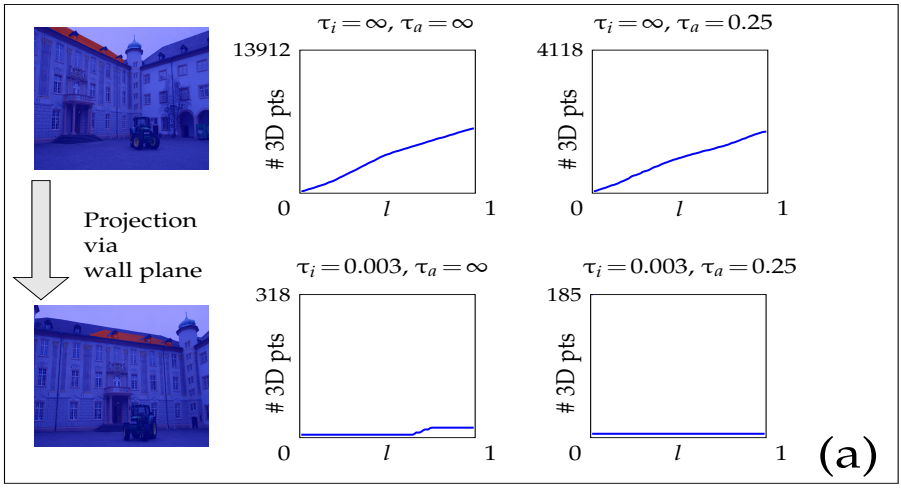
Since it is not possible to identify the inliers from the 3D points accuracy and ambiguity, we have adopted an indirect strategy to estimate whether a 3D plane correctly fits the 3D point cloud associated to an image region. In short, our approach consists in analyzing whether the 3D points tend to get statistically (on average) closer to the plane model when their inaccuracy and ambiguity decrease. If this is the case, the plane is likely to fit the actual 3D surface. Formally, let $\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ denote the set of 3D points $\mathbf{X}$ satisfying the three following criteria:

$$\begin{cases} \mathbf{X} \in \{\mathcal{R} \cup \mathcal{R}_{\boldsymbol{\pi}}\} \\ m_i(\mathbf{X}) \leq \tau_i \\ m_a(\mathbf{X}) \leq \tau_a \end{cases}$$

where $\boldsymbol{\tau} = \{\tau_i,\ \tau_a\}$ and $\tau_i \in \mathbb{R}^+$ and $\tau_a \in \mathbb{R}^+$ are investigated thresholds on the matching inaccuracy and ambiguity. As in Section 4.4, we abuse the notation, and write $\mathbf{X} \in \{\mathcal{R} \cup \mathcal{R}_{\boldsymbol{\pi}}\}$ to indicate that the 3D point $\mathbf{X}$ projects onto the 2D region $\mathcal{R}$, or its counterpart $\mathcal{R}_{\boldsymbol{\pi}}$ in $\mathcal{I}'$.

To analyze the scattering of the 3D points in $\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ around the investigated plane $\boldsymbol{\pi}$, we define $f_{\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}}(l, \boldsymbol{\pi})$ as a function describing the fraction of 3D points in $\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ whose distance to $\boldsymbol{\pi}$ is smaller than $l \in \mathbb{R}^+$, given a pair $\boldsymbol{\tau} = \{\tau_i,\ \tau_a\}$. Figures 4.7 ((a) and (b)) and 4.8 ((a) and (b)) illustrate multiple examples of scattering functions $f_{\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}}(l, \boldsymbol{\pi})$ for different types of regions, different proposed 3D planes and different pairs of inaccuracy/ambiguity thresholds. In each of the (a) or (b) figures, the left column shows how the 2D region $\mathcal{R}$ (in red) projects on the second view via a manually defined planar model (specified on the right of the arrow). The two other columns illustrate some scattering functions $f_{\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}}(l, \boldsymbol{\pi})$ for four pairs of investigated thresholds, ranging in $\tau_i = \{0.003; \infty\}$ and $\tau_a = \{0.25; \infty\}$. For each of the scattering curves, the upper ordinate value represents the number of 3D points in $\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ while the abscissa represents the distance l (in meters) to the investigated plane $\boldsymbol{\pi}$. The analysis of the scattering function is limited to $l \in [0; l_{\text{lim}}]$, where $l_{\text{lim}} = 1[\text{m}]$ has been empirically chosen accordingly to the scale of the Castle's 3D scene.

Figure 4.7 (a) considers the approximation of the roof of the Castle by the (ground-truth) 3D plane of the Castle's left wall. It is worth noticing that there is only a small percentage of the 3D points in $\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ that are close to this plane. This can be observed in the top-middle curve in Figure 4.7 (a), which indicates that more than 50% of the 3D points associated to the 2D roof region, are more than 1 meter away from the 3D wall plane. This observation is common to most of the cases for which the proposed plane does not correctly represent the 3D of the investigated region. Hence, a small area under the scattering curve appears to be a good indicator of the 3D plane incorrectness.



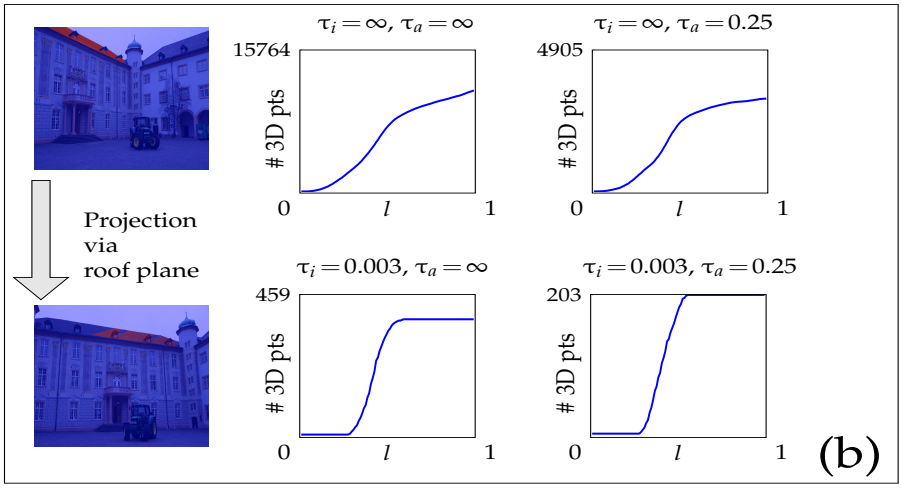


Figure 4.7: Our data-fidelity term considers the scattering (blue curves) of 3D points around the investigated plane $\boldsymbol{\pi}$. Generally, when the investigated 3D plane does not represent correctly the region (see (a)), the proportion of 3D points in $\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ near the investigated plane decreases when progressively considering more accurate and unambigious 3D points (lower values of τ_i and τ_a). When the investigated 3D plane represents correctly the region (see (b)), this proportion tends to increase especially when some error is tolerated in the distance to the plane (typically when $l > 30$ [cm] in the plots).

Figure 4.7 (b) illustrates the 3D approximation of the same roof region, but this time by the (ground-truth) 3D plane associated to the roof. The areas under the scattering curves are now larger than in Figure 4.7 (a), which confirms that this value might be a good indicator of the plane correctness. A deeper analysis of Figure 4.7 (b) also reveals that there might be a significative amount of 3D points that are still far away from this ground-truth plane. In particular, the top-right extremity of the top-middle curve shows that only barely more than 70% of the 3D points associated to this region are closer than 1 meter from this plane. This is explained by the low precision of the 3D points representing the roof region, due to its repetitive pattern nature, which makes the matching phase error-prone. However, when restricting ourselves to the more accurate and less ambiguous 3D points, *e.g.*, $\tau_i = 0.003$ and $\tau_a = 0.25$ (Figure 4.7 (b), curve in the bottom-right corner), we can observe that most of the 3D points are close from the ground-truth 3D plane.

Figure 4.8 (a) investigates the modeling of a 2D region $\mathcal{R}$ of the ground-plane (floor) by the planar model of the left wall of the Castle's sequence. Although the investigated wall plane is perpendicular to the (ground-truth) floor plane, the region $\mathcal{R}_{\boldsymbol{\pi}}$ is also located on the floor. This would make any kind of conventional reprojection error reasonably small, since the texture is relatively uniform on all the floor. In contrast, the (areas under the) scattering curves depicted in Figure 4.8 (a) reveal that the plane is not a valid model.

We conclude from these observations that the reliability of a proposed 3D plane can reasonably be inferred based on the analysis of the area under curve (AuC) observed for multiple sets of thresholds $\{\tau_i, \tau_a\}$.

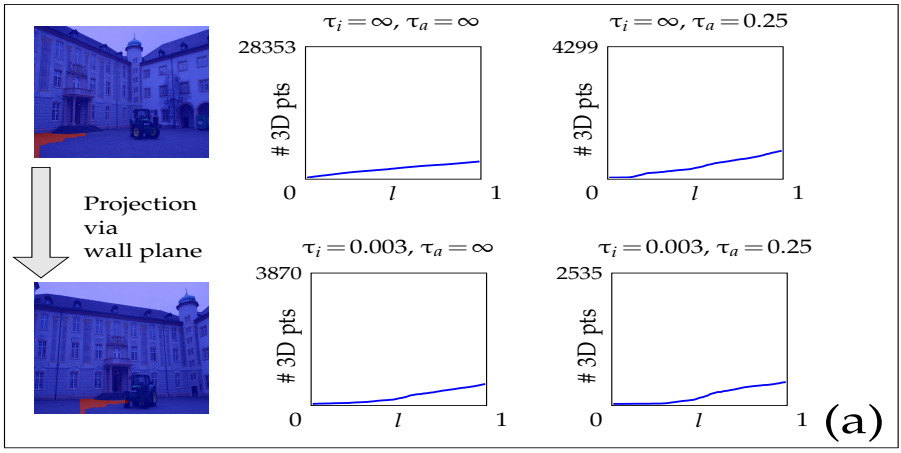
Since those AuC values correspond to different sets of 3D points, we propose to average them based on a geometric mean rather than the average mean. The geometric mean[5] is indeed generally better suited when deriving a single "figure of merit" for a set of items with different numerical ranges [39]. Hence, we define our data-fidelity metric as follows.

First, we introduce $A(\boldsymbol{\tau}, \boldsymbol{\pi}) \in [0;1]$ to denote the area under the curve of the scattering function $f_{\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}}(l, \boldsymbol{\pi})$:

$$A(\boldsymbol{\tau}, \boldsymbol{\pi}) = \int_0^{l_{\text{lim}}} f_{\mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}}(l, \boldsymbol{\pi}) \; dl.$$

Roughly speaking, this area reflects the likelihood that the 3D points $\mathbf{X} \in \mathcal{C}^{\boldsymbol{\tau}}_{\mathcal{R},\boldsymbol{\pi}}$ are "close" from the 3D plane $\boldsymbol{\pi}$.

[5]Given that more accurate and unambiguous matches are expected to describe the surface more reliably, one could imagine to assign a larger weight to their corresponding AuC when computing the geometric mean, *e.g.*, based on a weighted geometric mean. Our experiments have however shown that using a unitary weight for all AuC provides satisfying results.



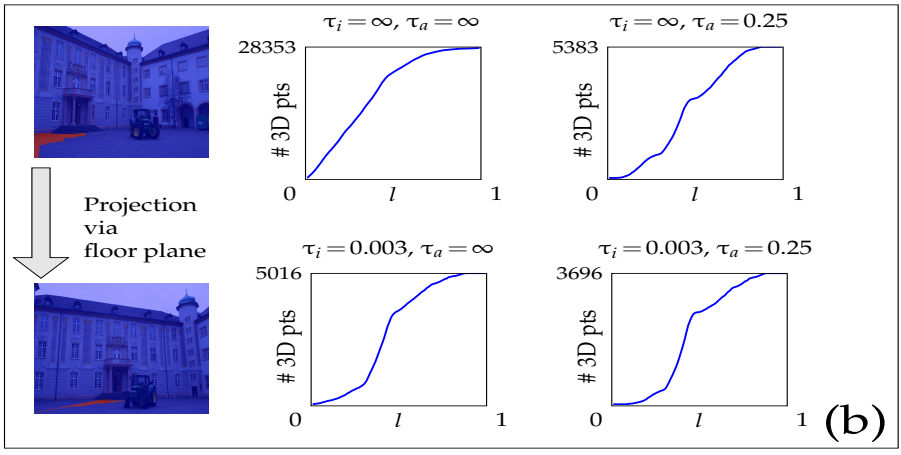


Figure 4.8: Evolution of the scattering function when a textureless region, *e.g.*, a floor region, is projected via the wall plane (90° away from the correct plane model). Although any data-fidelity based on an usual reprojection error would promote the plane as correct (because of the low image difference between the projected image and the image really captured), the very small areas under the scattering functions in (a) indicate that the 3D plane of the wall does not approximate correctly the 3D surface of the floor region. At the opposite, the high areas under the scattering functions in (b) indicate that the 3D plane of the floor correctly approximates the 3D surface of the floor region.

The data-fidelity $c(\mathcal{R}, \boldsymbol{\pi}) \in [0;1]$ of assigning a plane $\boldsymbol{\pi}$ to a region $\mathcal{R}$ is then defined, based on the geometric mean of a sequence of T tests $\boldsymbol{\tau}^{(t)} = \{\tau_i^{(t)}, \tau_a^{(t)}\}$ on the accuracy/unambiguity of the 3D points $\mathbf{X} \in \{\mathcal{R}, \mathcal{R}_{\boldsymbol{\pi}}\}$. Formally, the sequence of tests is defined as:

$$\boldsymbol{\tau}^{(t)} = \boldsymbol{\tau}^{(1)} - \frac{t-1}{T-1} \cdot \left(\boldsymbol{\tau}^{(1)} - \boldsymbol{\tau}^{(T)}\right) \qquad \forall t \in \{1, \cdots, T\},$$

with $\boldsymbol{\tau}^{(1)}$ (respectively $\boldsymbol{\tau}^{(T)}$) the set of maximum (respectively minimum) investigated thresholds and the data-fidelity is measured as:

$$c(\mathcal{R}, \boldsymbol{\pi}) = \begin{cases} 0 & \text{if } \mathcal{R}_{\boldsymbol{\pi}} \cap \boldsymbol{\Omega}_{\mathcal{I}'} = \emptyset \\ \exp\left(\frac{1}{T}\sum_{t=1}^{T} \log\left(A\left(\boldsymbol{\tau}^{(t)}, \boldsymbol{\pi}\right)\right)\right) & \text{otherwise.} \end{cases}$$

Finally, it is worth noting that we do not consider, in the computation of the data-fidelity, the area $A\left(\boldsymbol{\tau}^{(t)}, \boldsymbol{\pi}\right)$ which are computed on less than a certain number, set to 10 in practice, of 3D points $\mathbf{X} \in C_{\mathcal{R},\boldsymbol{\pi}}^{\boldsymbol{\tau}^{(t)}}$.

4.6 Joint optimization of data-fidelity, smoothness and amount of models

The assignment of a planar model to each of the N regions is formulated as a (model) label inference problem in a multi-label Markov Random Field (MRF). We propose to adapt the state-of-the-art *Propose, Expand and Re-Learn* (PEARL) algorithm [102], which iteratively infers the labels and refines the parameters of the proposed models, to our problem's specificity. As its name indicates, the PEARL inference optimization is composed of three steps: the proposition of a set of models ("propose stage"), the label inference ("expand stage") and the models reestimation ("re-learn stage").

In the "propose" stage, many hypothetical models are generated. While the original paper requires to generate several thousands of models candidates, we rely on Section 4.4 to limit ourselves to a few hundreds. This enables us to strongly accelerate the optimization, while keeping the same accuracy (as shown in Section 4.7).

In the "expand" stage, one planar model is assigned to each image region. A strong advantage of PEARL is that it allows us to explicitly consider the fact that a region can be occluded, by defining an extra label $L_\emptyset$ and its associated data-cost, in addition to the proposed model labels. The inference problem is expressed as an energy-driven minimization [102], which minimizes:

$$E(\mathbf{L}) = \sum_{n=1}^{N} \left(1 - c(\mathcal{R}_n, \boldsymbol{\pi}(L_n))\right) + \lambda \cdot \sum_{(p,q)\in\mathcal{N}} \omega_{pq} \cdot \delta\left(L_p \neq L_q\right) + \beta \cdot |\mathcal{L}_{\mathbf{L}}|, \quad (4.4)$$

where $\mathbf{L} = [L_1 \;\; L_2 \;\; \cdots \;\; L_N]^\top$ are the labels assigned to the N regions (each label L_n refers either to one of the K models, or to the occlusion label $L_\emptyset$), $c(\mathcal{R}_n, \boldsymbol{\pi}(L_n)) \in [0;1]$ is the cost of assigning the $\boldsymbol{\pi}(L_n)$ model to the n^{th} region, ω_{pq} is a weight associated to a pair of neighboring regions that encourages spatial coherence, $\delta(.)$ is the indicator function and the last term $|\mathcal{L}_\mathbf{L}|$ is the number of assigned models. This last term encourages parsimony, to describe the scene with as few plane models as possible. The data-fidelity cost $c(\mathcal{R}_n, \boldsymbol{\pi}(L_n))$ is defined in Section 4.5 and we define the weights w_{pq} as:

$$\omega_{pq} = \begin{cases} 1 - \mathbb{E}\left[|\nabla \mathcal{I}(\mathbf{x})|\right]_{\mathbf{x}\in\mathcal{B}} & \text{if } \mathcal{R}_p \text{ and } \mathcal{R}_q \text{ have a common border } \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

where the gradient amplitude $|\nabla \mathcal{I}(\mathbf{x})|$ is rescaled to $[0;1]$, by applying contrast stretching over the entire image, and $\mathbb{E}[.]$ represents the mean operator. Given those definitions, the labels are inferred by minimizing the energy (Equation (4.4)) based α-expansion optimization [24].

Eventually, in the "re-learn" stage, PEARL extracts, for each assigned label $L_n \neq L_\emptyset$, the set $\mathcal{P}_{L_n}$ of region assigned to this label, and reestimates the associated model. This reestimation is done by selecting the set of 3D points that project into one of the regions of $\mathcal{P}_{L_n}$ and applying RANSAC [58] (with inlier threshold τ) to robustly fit a new plane model to these 3D points. In their original paper, Fischler *et al.* have proposed to select the model having the maximum number of inliers. To discriminate between the RANSAC planes having the same number of inliers, we adopt the relaxed and more robust score proposed in [21] (Equation (1) in their paper).

In practice, the PEARL algorithm iterates sequentially between the three stages, until $E(\mathbf{L})$ reaches a minimum. It is worth noting that, in their original paper, Isack and Boykov have proposed not to consider anymore the data associated to the empty label in the next iterations. At the opposite, we compute their data-fidelity with respect to the updated models, so as to give the opportunity to assign a potentially more accurate updated model to those regions that have received an empty label in the initial stages of the algorithm.

4.7 Experiments

Our validations are divided into three parts.
First, we demonstrate the efficiency of our plane proposition phase by showing that it generates a small set of 3D plane hypotheses that includes the 3D ground-truth of the scene. Second, we show, on 10 well-known sequences representing diverse (man-made) building scenes, that our method is able to locate their principal 3D planes, as well as to detect their occluded regions. Finally, we generate free-viewpoints around the piecewise-planar reconstructed scenes.

Efficiency of the plane proposition phase

To assess the relevance of our plane proposition method, Figure 4.9(a) first depicts, as a function of M, how well a set of M plane candidates approximates a ground-truth 3D surface described by $I^\star$ ground-truth planes. Precisely, from each tested set of M plane candidates, we have selected the $I^\star$ planes candidates approximating the best the ground-truth planes $\{\boldsymbol{\pi}_i^\star\}_{i=\{1,\cdots,I^\star\}}$, *i.e.*, the ones minimizing $\|\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^\star\|_2$.

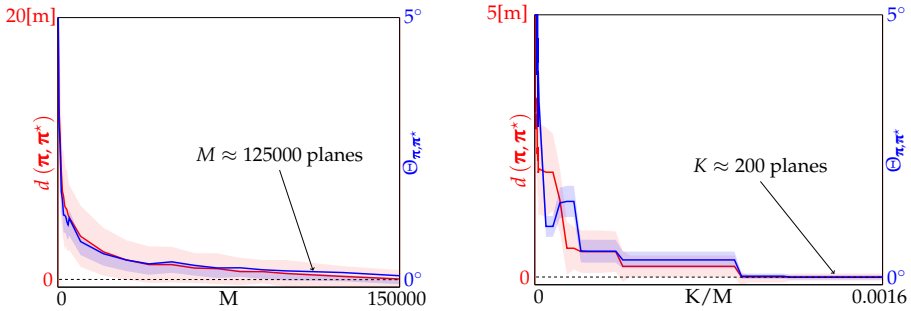


Figure 4.9: (a) Average of the maximum remoteness, with respect to a set of ground-truth planes $\{\boldsymbol{\pi}_i^\star\}_{i=\{1,\cdots,I^\star\}}$, of the best plane candidates $\boldsymbol{\pi}_m$ among the M plane candidates. (b) Average of the maximum remoteness of the best proposed plane $\boldsymbol{\pi}_k$ among the proposed K planes with respect to a set of ground-truth planes $\{\boldsymbol{\pi}_i^\star\}_{i=\{1,\cdots,I^\star\}}$.

The blue and red curves in Figure 4.9(a) respectively plot the worst relative angular and distance errors (cfr. Equations (4.2) and (4.3)), as measured with respect to $I^\star = 7$ ground-truth planes of the well-known Herz-Jesu-P8 dataset [206]. The performances have been averaged on $P = 1000$ iterations of the random triplets-based selection of the M planes. The blue and red shaded areas in this figure illustrate respectively the standard deviations of these two metrics. From Figure 4.9(a), we observe that multiple plane candidates ($M \geq 125000$) are required to ensure that the set of M candidate planes represent "correctly"[6] the ground-truth 3D surface.
Figure 4.9(b) depicts the same angular and distance errors as a function of K, when $M = 125000$. From this figure, we observe that our plane proposition phase effectively proposes a highly limited set of 3D plane (K is set to 200 in all our experiments), which greatly reduces the number of hypotheses (in practice, M had to be chosen 2 to 3 orders of magnitude higher than K to make sure to include all the ground-truth plane models of the scene) by eliminating redundant ones. This allows the PEARL optimization to work efficiently on a reduced set of proposed planes.

[6] Even if the plane candidates do not perfectly correspond to the ground planes, their parameters are later optimized in the "re-learn" phase of PEARL (see Section 4.6).

Piecewise-planar 3D reconstruction

We validate our method on 10 well-known and calibrated sequences representing street-level captures of (man-made) building scenes (indoor and outdoor). While these datasets provide multiple different views of each scene, we have arbitrarily selected two distant views among the available ones to define a set of wide-baseline stereo pairs. The segmentation of the left view has been obtained based on [229], and the required color dissimilarity threshold for learning the dominant colors consituting the image has been set to 20 in all the experiments. This parameter influences the number of obtained regions. Our experiments have revealed that it does not strongly affect the performance of our piecewise-planar 3D approximation.

The 3D point cloud estimated from Equation (4.1) is based on Daisy descriptors [218], which have been chosen for their robustness against wide-baseline geometric distortions, and their appropriateness for dense estimation [217] (cfr. Section 2.3.2).

Our method relies only on two types of parameters. First, the RANSAC inliers/outliers parameter τ and the parameter l_{lim} (representing the investigated orthogonal distance around the proposed 3D plane) are fixed, based on rough prior human knowledge about the depth variability in the scene. In all our experiments, l_{lim} has been chosen between 30cm and 1.5m, while τ has been fixed to $\tau = l_{\text{lim}}/5$. Second, for the parameters of the PEARL optimization, we have set the pairwise term to $\lambda = 0.1$ and the occlusion data-fidelity to $c(\mathcal{R}_n, \boldsymbol{\pi}(L_\varnothing)) = 0.5$ in all our experiments. This last parameter is a good trade-off between accepting plane assumptions on regions associated to noisy 3D points and discarding bad planes. The only highly varying parameter of the PEARL optimization is the labeling weight β. It is chosen between $[0.1; 0.5]$ according to the amount of expected 3D planes. In practice, multiple values have been tested within this range, and the one leading to a visual reasonable trade-off between number of planes and accuracy of representation has been kept for each dataset[7].

Figures 4.10 and 4.11 illustrate the results of the different steps of our algorithm, namely (a) the segmentation of one of the reference image of the wide-baseline stereo pair (images (a) and (b)), (c) the dense point cloud and (d) the regions assigned to the same 3D plane model. We also provide the projection of the first view (a) onto the second one (b) via the piecewise-planar approximated model. Occlusions are highlighted in black. From top to down, the used dataset are: CastleP19/FountainP11/HerzJesuP25 [206], Oxford Corridor/Model-house (in Figure 4.10), Oxford Library/Wadham/MertonI/MertonII/MertonIII [235] (in Figure 4.11).

As a first observation, we note that most of the 3D planes are correctly estimated (columns (d) and (e) in Figures 4.10 and 4.11), despite the presence of noise in the dense 3D point cloud. This performance is due to the

[7] One might also fix *a priori* the number of 3D planes representing the scene, and scan the specified range until this number of planes is reached.

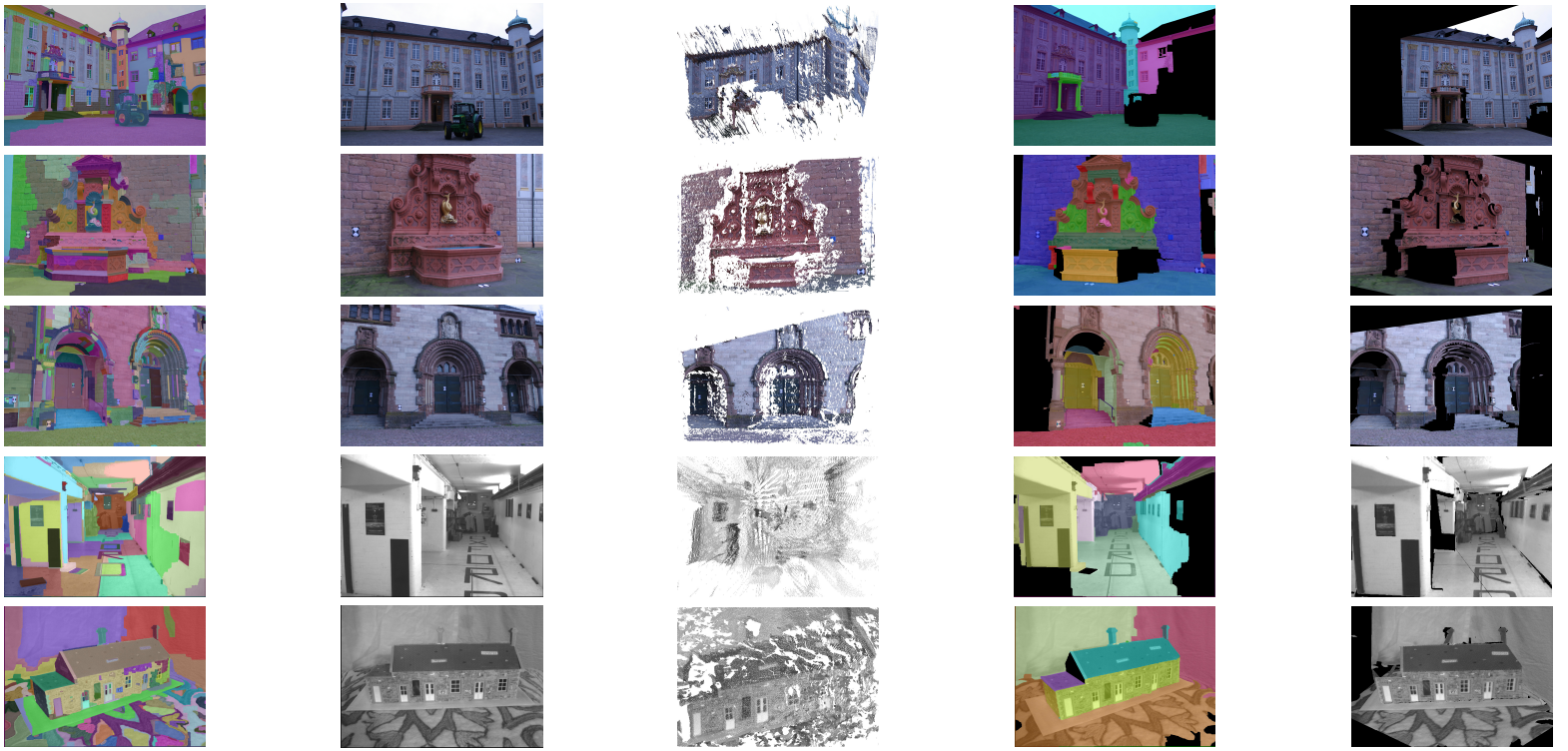
high robustness of the proposed data-fidelity metric $c(\mathcal{R}, \boldsymbol{\pi})$, which simultaneously considers the 3D points of $\mathcal{R}$ and $\mathcal{R}_{\boldsymbol{\pi}}$. On the one hand, because the 3D points have been determined based photometric/projective robust descriptors, they are less impacted by wide-baseline discrepancies than usual 2D color differences, such as done in [80]. On the other hand, even when considering photometric/projective robust descriptors instead of color in the pixel discrepancy, an irrelevant 3D plane candidate might lead to a very low discrepancy, as illustrated in the left column of Figure 4.8(a). In contrast, because our 3D points-based data-fidelity embeds additional information, typically related to the photometric rarety/ambiguity along the epipolar line (cfr. Section 4.5), it appears to be more discriminative than a simple 2D reprojection error. Also, the complementary information of the 3D points of $\mathcal{R}$ and $\mathcal{R}_{\boldsymbol{\pi}}$ enables to handle the cases where the distribution of the 3D points associated to a region $\mathcal{R}$ are strongly contamined with outliers (*e.g.*, $\mathcal{R}$ is textureless). In this case, if $\mathcal{R}_{\boldsymbol{\pi}}$ is discriminative (*e.g.*, highly textured), its accurate 3D points will push towards the rejection of the proposed plane (which project a textureless region onto a discriminative one). If both $\mathcal{R}$ and $\mathcal{R}_{\boldsymbol{\pi}}$ are challenging (*e.g.*, textureless), considering the union of both $\mathbf{X} \in \mathcal{R}$ and $\mathbf{X} \in \mathcal{R}_{\boldsymbol{\pi}}$ might reveal the correct plane. This is particularly true in wide-baseline stereo setups, in which the inliers spread closer around the ground-truth model than the inliers obtained based on small-baseline stereo setups. This is due to the more accurate 3D estimations provided by wide-baseline stereo setups, compared to the small-baseline ones [92] (cfr. Section 1.2.3).

As a second interesting observation, which can be observed at the level of the *D* symbol on the Oxford Corridor dataset (Figure 4.10, fourth row, column (d) and (e)), we note that our method might assign a 3D plane on partially occluded regions, based on the 3D points associated to the non-occluded part of this region. Under-segmentation might thus permit to model occluded parts, but also increases the risk of violating the planar assumption.

The failure cases of our approach are not frequent, and can be divided into four classes: wrong propagation of 3D plane model to other regions, assignement of visible regions to the occlusion label, and wrong 3D plane estimation.

The first class of errors appears either when all the 3D points of a region $\mathcal{R}$ (and/or $\mathcal{R}_{\boldsymbol{\pi}}$) are too strongly contaminated by 3D outliers, and the high value of β pushes towards a wrong propagation of the model (*e.g.*, in the sky regions in the first and second row of Figure 4.11), or when the initial segmentation assigns the same label to pixels that do not belong to the same plane (*e.g.*, on the gutter at the middle of the left wall of Merton II, fourth row, column (a) in Figure 4.11).

The second class of errors appears in the absence of 3D points in the interval $[0; l_{\text{lim}}]$ (considered interval around the plane model hypothesis), while the investigated plane correctly approximates the 3D ground-truth. This behavior can be observed on the tower of the Library dataset (Figure 4.11, second row, column (d)).



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

Figure 4.10: (Best viewed in color). Based on the segmentation (a) of one of the two wide-baseline views ((a) and (b)) and on their associated dense point cloud (c), our method approximates the 3D surface by the minimum set of 3D planes. In (d), regions assigned to the same plane are illustrated with the same color. The reprojection of the optimal piecewise-planar reconstruction, textured based on the first view (a) and projected in the second view (b), is represented in (e).

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

Figure 4.11: (Best viewed in color). Based on the segmentation (a) of one of the two wide-baseline views ((a) and (b)) and on their associated dense point cloud (c), our method approximates the 3D surface by the minimum set of of 3D planes. In (d), regions assigned to the same 3D plane are illustrated with a same color. The reprojection of the optimal piecewise-planar reconstruction, textured based on the first view (a) and projected in the second view (b), is represented in (e).

The third class of errors can affect either the large and challenging 2D regions, or the smallest ones. In the first case, the inaccuracy of the plane estimation originates from the planar re-estimation of PEARL, using RANSAC on a region that is contaminated by more than 50% of 3D "outliers". This behavior can be observed on the Oxford corridor image (Figure 4.10, fourth row, third column), or on the terrace of the Model-house sequence (Figure 4.10, fifth row, third column). The second case appears in very small regions with 3D surfaces that can not be correctly described based on the other optimized models. In this case, the spatial concentration of their associated 3D points makes the (RANSAC-based) fitted plane more error-prone to noise than if the 3D points were spatially spread (large region). The inaccuracy of the re-estimated 3D plane associated to these small parts can be detected by observing their projections from the reference image to another view, via their associated inaccurate 3D models, which project these small regions far away from their original location. This problem affects the roof of the windows of the Merton I dataset, which are projected on the grass, as illustrated in the third row of Figure 4.11, column (e).

The fourth limitation of our method concerns the reconstruction of arbitrary 3D shapes. As most of the state-of-the-art methods [160], our method either approximates the curved surfaces by planar patches, or considers their associated 3D points as outliers to a plane model. In our algorithm, this trade-off is controlled by the weight factor β, multiplying the number of labels in Equation 4.4. This behavior is similar to the one obtained recently by Oesau, Larfarge and Alliez [164], which relies on a local Manhattan world assumption in each cell of a 3D grid. As a perspective, other 3D primitives such as cylinders, spheres, ellipsoids and cones, could be considered to model more complex 3D structures. This idea has been explored in the recent work of [121], in which each tree is first detected and after modeled based on a 3D cylinder (for the trunk) and a 3D ellipsoid (for the upper part of the tree).

Regarding complexity, our method reconstructs each 3D scenes in a few minutes (from 4 to approximatively 20 minutes, according to the resolution of the reference images, Matlab implementation on a 2.4GHz Intel I5 CPU, 8Gb RAM machine), which are divided into three parts: approximatively 60% of the running time is dedicated to the dense point cloud generation (using currently a non-parallel implementation of the WTA method), 25% on the plane proposition phase (in which the location of the pixels in $\mathbf{\Delta}$ and $\mathbf{\Delta}'$ takes the most of the running time), and 15% for the rest.

View interpolation around the 3D reconstructed model

To complete our visual experimental results, Figures 4.12 and 4.13 demonstrates the effectiveness of our dense, piecewise-planar 3D approximation method, by projecting the textured 3D piecewise-planar models on virtual intermediate views.

Figure 4.12: Projection of the textured piecewise-planar approximation of the scene's 3D on virtual views in-between the two cameras of the wide-baseline stereo pair.

Figure 4.13: Projection of the textured piecewise-planar approximation of the scene's 3D on virtual views in-between the two cameras of the wide-baseline stereo pair.

4.8 Conclusion

State-of-the-art man-made scenes 3D reconstruction methods either approximate a sparse point cloud by a set of simple 3D proxies, or deliver complex meshes fitted on a dense point cloud. On the one hand, because sparse point clouds only represent 3D points associated to the more discriminative pixels, they do not allow to reconstruct regions with very few textures and/or composed of repetitive patterns. On the other hand, detailed meshes provides redundant informations on regular surfaces, are very computational and are vulnerable to noisy 3D points. To ensure a very low level of noise in the estimation of 3D points, these methods require to capture the 3D scene with a high number of (small-baseline) camera views. Our approach requires only two wide-baseline views and provides a dense, piecewise-planar approximation of the 3D scene. We express the 3D reconstruction as a generalized plane assignment problem over 2D image regions, in which the occluded regions are explicitly modeled. As opposed to [21], we rely on a dense, and thus implicitly highly corrupted, 3D point cloud to allow the approximation of challenging (*e.g.*, textureless or repetitively patterned) 2D regions, *e.g.*, grass floors. Therefore, we adopt a plane hypothesis testing framework. It relies on a limited number (*e.g.*, ≈ 200) of plane models to approximate the scene's 3D. It then formulates the plane assignment problem as an energy-driven formulation, which simultaneously optimizes a data-fidelity term, the smoothness of the plane assignment over the regions and the number of used models. Our main contributions have to do with (i) the computation of a small set of planar models that includes most of the models that are relevant to model the 3D scene, and (ii) the derivation of a data-fidelity metric that measures the fitting error while considering the errors associated to the 3D points resulting from an accurate and unambiguous matching. Also, to the best of our knowledge, by simultaneously optimizing the data-fidelity, the smoothness and the number of assigned models, our light-weight method is the first one to densely approximate a 3D scene while simultaneously targeting a minimal number of models. We have demonstrated the accuracy of the approximated 3D models by interpolating virtual views around a variety of man-made scene, on which traditional MVS methods fail [21]. We attribute the high performances of our algorithm to the robustness of the newly proposed data-fidelity term, which incorporates matching accuracy and matching ambiguity into a new 3D fitting error, instead of simply combining a conventional fitting error with a region-based projection error (*e.g.*, based on a weighted average whose ad-hoc weights are application dependent).

Future work will investigate the symmetrization of our method with respect to the two views, the propagation of the plane models to occluded regions, and their generalization to non-planar models.

CHAPTER 5

Object interpolation using shape prior regularization of epipolar plane images

While the previous chapter has targeted the 3D reconstruction of the (man-made) background of a scene, this chapter considers the synthesis of intermediate views of a foreground object captured by two calibrated and widely spaced cameras. Based only on those two very different views, this chapter proposes to reconstruct the object Epipolar Plane Image Volume (EPIV) [127], which describes the object transformation when continuously moving the viewpoint of the synthetic view in-between the two reference cameras. This problem is clearly ill-posed since the occlusions and the foreshortening effect make the reference views significantly different when the cameras are far apart. Our main contribution consists in disambiguating this ill-posed problem by constraining the interpolated views to be consistent with an object shape prior. This prior is learnt based on images captured by the two reference views, and consists in a nonlinear shape manifold representing the plausible silhouettes of the object described by Elliptic Fourier Descriptors. Experiments on both synthetic and natural images show that the proposed method preserves the topological structure of objects during the intermediate view synthesis, while dealing effectively with the self-occluded regions and with the severe foreshortening effect associated to wide-baseline camera configurations.

5.1 Introduction

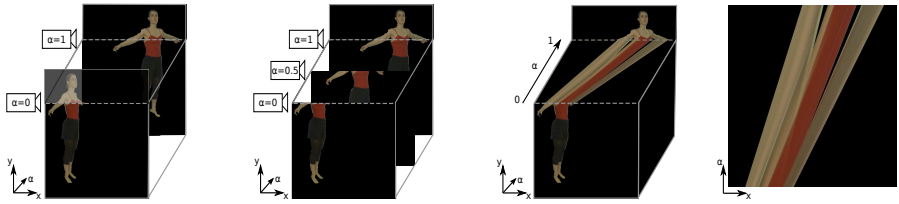
Virtual view synthesis aims at rendering images of a real scene from different viewpoints than the ones acquired by the cameras. This chapter restricts the general arbitrary view synthesis problem to the interpolation of images observed by a virtual camera located in an arbitrary position along the baseline connecting two reference cameras. The graceful transition between two reference viewpoints is a demanded feature, especially in the field of video production [165]. For example, in the rendering of cultural or sport events, conventional acquisition systems switch abruptly between the cameras, making the viewer uncomfortable. By generating a graceful transition between the reference viewpoints, view interpolation gives the ability to understand how the rendered viewpoint changes, *i.e.*, the feeling of being "inside the scene".

To synthesize intermediate views in-between reference cameras, state-of-the-art methods generally decompose the scene into its background and its dynamic foreground objects, and reconstruct them independently [196]. The interpolation of dynamic foreground object, situated relatively close to the pair of cameras, is the most complex question among both [109] [32], because the background can be reconstructed through projection of its 3D geometry [12]. Typically, the background 3D geometry can reasonably be acquired, based on state-of-the-art active 3D acquisition systems [174] [104] if it is still, or based on piecewise-planar 3D geometry approximations [21] [12] [113] when it is far from the cameras. The fundamental issues encountered to reconstruct a foreground object lie in (1) the availability of only two reference views, and (2) the object proximity to the cameras, compared to the distance between those cameras. The first factor prevents dense 3D estimation for the dynamic object, while the second causes many projective discrepancies between the two views (occlusions, foreshortening effects, etc.), which hamper the computation of dense correspondences and lead to holes in the interpolated views [32].

This chapter focuses on the reconstruction of foreground objects and assumes that the object silhouette can be extracted from the reference views[1], as generally assumed by state-of-the-art foreground synthesis methods [73] [80] [12]. Based only on two very different views captured by a pair of wide-baseline cameras, such as the ones shown in Figure 5.1(a), our scheme reconstructs intermediate views (see Figure 5.1(b)) along the baseline by reconstructing the object's Epipolar Plane Image Volume [127] (see Figure 5.1(c)), composed by the set of Epipolar Plane Images (see Figure 5.1(d)).

The specificities of the proposed method lie in the regularization of the ill-posed reconstruction of the Epipolar Plane Images (EPIs) based on a sequence of plausible intermediate object silhouettes. As illustrated in Figure 5.3, this sequence is derived from a low-dimensional manifold, learnt from the previous observations of the dynamic object by the wide-baseline stereo pair. Interestingly, the priors are used not only to disambiguate the matching, but also to determine how occluded parts vanish/appear while moving from one reference view to the other.

[1] In this chapter, the foreground is generally extracted based on a simple thresholding of the ℓ_2 color distance with a gaussian mixture model of the background [208].



(a) Epipolary rectified reference views

(b) Epipolary rectified reference views + 1 intermediate view

(c) Epipolary rectified reference views + continuum of intermediate views

(d) Epipolar Plane Image

Figure 5.1: Two (epipolary rectified) reference views (a) are available to generate intermediate views (b). For that purpose, our method estimates the Epipolar Plane Image Volume (c) made of the set of Epipolar Plane Images (d).

To the best of our knowledge, our work is the first one to reconstruct topologically consistent images from only two widely separated cameras, even for their occluded parts, while dealing effectively with the self-occluded regions and with the severe foreshortening effect associated to wide-baseline camera configurations.

The rest of this chapter is organized as follows. Section 5.2 surveys the recent advances in virtual view reconstruction, and identifies the limitations of earlier methods in our envisioned wide-baseline stereo acquisition setup. Section 5.3 introduces our proposed Epipolar Plane Images interpolation formalism. Section 5.4 explains how to capture and embed a prior about the plausible silhouettes of the object in a low-dimensional silhouette manifold, which can be exploited to constraint the reconstruction of the EPIs between two reference images, as detailed in Section 5.5. The view synthesis process is described in Section 5.6. Section 5.7 then validates our framework by generating topologically valid intermediate views on both real and synthetic images, captured by two cameras with very different viewpoints. The advantages induced by shape priors are further demonstrated by comparing our method with a set of conventional and state-of-the-art approaches.

5.2 Related work and challenges

The view synthesis techniques are generally categorized into two groups in the literature, namely model-based rendering and image-based rendering.

In model-based rendering, a 3D shape model of the observed scene is explicitly reconstructed from multi-view images. Adequate texture is then mapped on the model, and projected onto any arbitrary viewpoint. Methods such as projective grid space [177] [241], visual-hull [123] [146] [142] [194], 3D model adjustment [31], and shape from video [73] belong to this category. Those methods have the advantage to synthesize intermediate views repre-

senting the actual 3D scene. However, the quality of the virtual view is highly dependent on the accuracy of the estimated 3D model [182]. To obtain an accurate 3D model, the model-based rendering methods therefore rely on a dense coverage of the scene, which requires a large number of precisely calibrated video cameras [197]. The trade-off between the accuracy of the reconstruction and the amount of cameras is often relaxed when the distance between the object and the reference cameras is important compared to the baseline distance separating these cameras [77]. In this particular case, a simple (set of) planar model(s) (called *billboards*) permits to generate realistic intermediate views of the object. However, when the distance to the scene decreases, planar *proxies* become insufficient to approximate the 3D of the object [12]. This makes model-based rendering inappropriate to render close (dynamic) scenes between wide-baseline cameras.

In contrast, image-based rendering (IBR) methods [190] create the virtual view directly in the image color space without explicit reconstruction of a 3D piecewise smooth surface. Such methods are further classified into arbitrary-view and baseline interpolation approaches. On the one hand, arbitrary-view IBR approaches determine the pixel color values of each virtual view in a way that is geometrically and/or photometrically consistent with $N \geq 2$ reference views. These methods focus on optimizing multiple depth maps (either the ones of the virtual views [80], or the ones of the reference views [12]) and/or the virtual image's color [60]. However, the dense estimation of a depth map is only possible when all the 3D points corresponding to a pixel in the reconstructed view are observed with at least two reference views. This requires a sufficiently dense coverage of the scene with many cameras. On the other hand, baseline interpolation approaches determine region correspondences or pixel correspondences (disparity) between only two reference views and generate the intermediate views by interpolation [101] or morphing [185]. They are restricted to the reconstruction of images on the baseline between a pair of reference cameras, generally for small-baseline configurations, and rely on dense correspondence between the views. This trend culminates with light-field reconstruction approaches [127], which require tens or hundreds of narrow-baseline[2] cameras/lenses [163] to determine a continuous (sub-pixel) correspondence between the reference views. So far, image-based rendering techniques have thus been restricted to dense acquisition setups, where many images of the same 3D scene are captured by cameras that are close to each other, compared to their distance to the 3D scene. To the best of our knowledge, no image-based rendering method has been able to provide effective synthesis with a wide-baseline setup composed of only two reference cameras.

The main reason for the failure of rendering methods in wide-baseline stereo setups is that the more different the viewpoints, the more important the geometrical deformations (including projective distortions and occlusions), and the more difficult it is to find correspondences between images from different cameras. More precisely, the three following issues are specific to wide-baseline configurations:

- *The foreshortening effect* causes a distance or an object to appear shor-

[2]The reference views are separated from a few microns (microlens arrays) to a few centimeters in narrow-baseline setups.

ter/wider than it is because it is angled toward the viewer (see Figure 2.17). Because the compaction ratio depends on the viewpoints, a given 3D object will be represented by a totally different number of pixels in different views. This implies that finding correspondences with fixed-template matching methods fails [226]. The same holds when a pixel correspondence is optimized by graph-cut [24], belief propagation [56], or dynamic programming [40] approaches, which generally enforce the pixel uniqueness constraint, *i.e.*, a pixel in an image corresponds to at most one pixel in another image.

- *The self-occlusion effect* occurs when a part of an object hides another region of the same object. Parts of the object can thus be observed in only one of the camera views, so that no correspondence can be found with the other reference views. This problem drastically limits the correspondence-based interpolation methods [185] in a wide-baseline configuration.

- The lack of *sparse correspondences* and non-ambiguous correspondences induced by the large difference in viewpoints results in sparse disparity/depth maps, leading to large holes in the reconstructed intermediate view. Multiple methods exist to fill in these holes [144] [114] [252], but they are either based on globally non-valid hypothesis (*e.g.*, holes should contain patterns that are visible in the non-occluded parts), or on computationally expensive (post-)processings [18].

Our proposed method explicitly addresses those issues by computing correspondences between a continuous set of image segments (from which dense correspondences can be inferred, *e.g.*, through linear interpolation), and by constraining those correspondences to be consistent with a plausible deformation of the projected object silhouette between the reference views (guides the occlusion of segments, or their shrinkage/elongation due to the foreshortening effect).

5.3 Wide-baseline interpolation algorithm

This chapter adopts an EPI interpolation formalism to reconstruct the image of a foreground object between two widely spaced cameras. As depicted in Figure 5.1, the transformations of images between different viewpoints can be described by the object Epipolar Plane Image Volume [22] [41] (EPIV) (see Figure 5.1(c)). By definition, an EPIV is obtained by arranging in a 3D stack the images captured by a dense array of cameras that are uniformly distributed along a line with their image plane coplanar and vertically aligned. This is performed through epipolar rectification [92] of the reference images, which associates each horizontal line in one image to a row with the same ordinate in the other image, as illustrated in Figure 5.2.

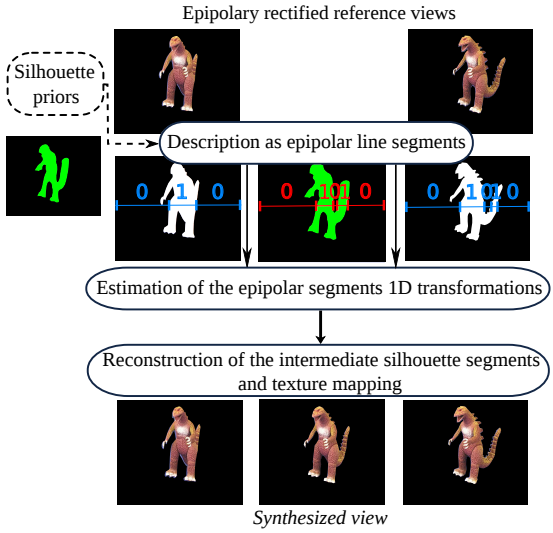


Figure 5.2: Our view interpolation method overview: foreground object silhouette segments are matched between the epipolar lines of two reference views, based on the prior about plausible silhouettes in intermediate views.

Roughly speaking, it implies that two corresponding pixels must belong to the same horizontal plane in the EPIV, and that any transverse cross-section of this 3D cube, *i.e.*, an Epipolar Plane Image (EPI) (see Figure 5.1(d)), describes how the pixels of one epipolar line in a view move to the other view. The light field theory [127] states that these transitions are always linear and that their slopes are inversionally proportional to the scene's depth. The EPIV is much richer than the depth information generally estimated by state-of-the-art wide-baseline stereo methods. Indeed, the EPIV additionally englobes the appearing/vanishment of occluded parts. However, its estimation has been limited so far to very narrow-baseline setups, which only permits to generate intermediate views in a very narrow range.

We adopt a new object-based approach to reconstruct the EPIV. After epipolar rectification of the reference views, we:

1. Learn a low-dimensional silhouette manifold. It describes prior plausible transformations of the object silhouette when changing the viewpoint along the baseline (see Figure 5.3, left side).

2. Use a sequence of plausible silhouettes to define how the object silhouette epipolar line segments are transformed (*i.e.*, through scaling/translation/vanishing) between the two reference views (Figure 5.3, right side).

3. Interpolate object textures based on the transformations, vanishments or appearance of the silhouette epipolar line segments.

The different blocks of our novel view interpolation algorithm are depicted in Figure 5.2 and described in detail in the next sections.

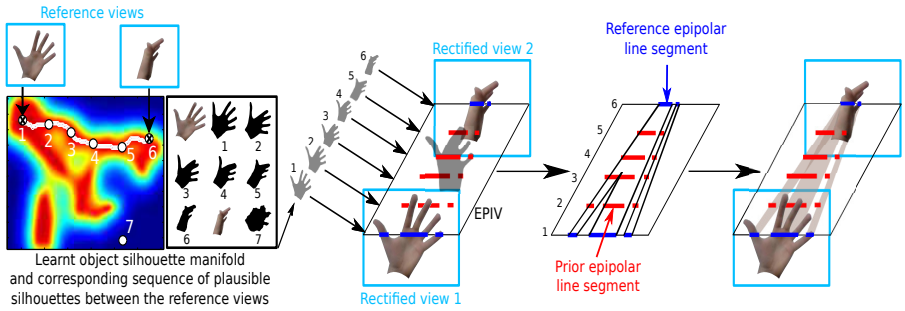


Figure 5.3: We propose to regularize the ill-posed problem of reconstructing the set of EPIs by incorporating prior knowledge about the plausible deformations of the object silhouette. This prior knowledge is learnt beforehand and is described by a low-dimensional space, from which intermediate 2D prior silhouettes can be extracted in-between the projected reference ones (left part on the figure, each point of this manifold represents a silhouette, while the color scale refers to the confidence about its plausibility). These intermediate 2D priors are then adequately placed in the EPIV (middle part of the figure) and are converted into a set of 1D priors to disambiguate the reconstruction of the set of EPIs (right part).

5.4 Object silhouette priors

This section describes the construction of priors on the plausible deformations undergone by an object silhouette during a viewpoint change. It aims at providing a sequence of approximated object silhouettes that defines *a priori* a plausible transition from the left reference silhouette to the right one. This sequence is then used to regularize the reconstruction of the set of EPIs.
We propose to generate these silhouette priors in four steps:

1. Learning a low-dimensional space representing the plausible silhouettes of the object.

2. Locating, in this low-dimensional space, the silhouettes observed in the reference views.

3. Interpolating, in this low-dimensional space, a sequence of low-dimensional silhouettes that likely represent the deformation of the object silhouette in-between the reference views.

4. Converting these low-dimensional intermediate silhouette representations into high-dimensional prior images for view synthesis.

The main challenge of our approach lies in the definition of a low-dimensional space that ensures that the interpolation step results in a smooth and topologically coherent sequence of silhouette priors. We propose to follow the pioneer approach of Prisacariu and Reid [172], who avoid the curse of dimensionality problem [15] by splitting the low-dimensional manifold construction into two parts: the first part describes the shape of a silhouette as a set of

high-dimensional features, and the second part maps those high-dimensional descriptors to a lower dimensional latent space. The different steps of the construction of the prior silhouettes are described in detail in the rest of this section.

5.4.1 High-dimensional silhouette description

We first propose to use Elliptic Fourier Descriptors (EFD) [116] as high-dimensional features for object silhouettes. Elliptic Fourier Descriptors represent the shape of a silhouette, given as a set of 2D coordinates $(u(t), v(t))$, as a sum of N elliptic harmonics, based on:

$$u(t) = a_0 + \sum_{n=1}^{N} \Big(a_n \cos \frac{2\pi n t}{T} + b_n \sin \frac{2\pi n t}{T} \Big),$$

where T is the perimeter of the contour and:

$$\begin{aligned} a_0 &= \frac{1}{T} \sum_{p=1}^{K} \Big(\frac{\Delta u_p}{2\Delta t_p} (t_p^2 - t_{p-1}^2) + \xi_p \, (t_p - t_{p-1}) \Big) \\ a_n &= \frac{T}{2n^2\pi^2} \sum_{p=1}^{K} \Big(\frac{\Delta u_p}{\Delta t_p} \big(\cos \frac{2\pi n t_p}{T} - \cos \frac{2\pi n t_{p-1}}{T} \big) \Big) \\ b_n &= \frac{T}{2n^2\pi^2} \sum_{p=1}^{K} \Big(\frac{\Delta u_p}{\Delta t_p} \big(\sin \frac{2\pi n t_p}{T} - \sin \frac{2\pi n t_{p-1}}{T} \big) \Big), \end{aligned}$$

where

$$\xi_p = \sum_{j=1}^{p-1} \Delta u_j - \frac{\Delta u_p}{\Delta t_p} \sum_{j=1}^{p-1} \Delta t_j,$$

with K being the number of sampling points in the contour, t_p the curvilinear coordinates on the shape, u_p the abscissa projection of t_p, $\Delta u_p = u_p - u_{p-1}$ and $\Delta t_p = \sqrt{(\Delta u_p)^2 + (\Delta v_p)^2}$. The second coordinate of the shape contour, $v(t)$, is defined completely analogously in terms of coefficients c_0, c_n and d_n, by exchanging Δu_p by Δv_p. Each harmonic is thus described by four coefficients, which have an intuitive geometrical interpretation: a_n (b_n) corresponds to the projection on the u-axis of the semi-major (minor) axis of the n^{th} elliptic harmonic and c_n (d_n) to their projections on the v-axis. We thus propose to describe the shape of an object silhouette as a high dimensional feature vector, composed of N sets of harmonic coefficients (a_n, b_n, c_n, d_n). In our validations, the parameter N is simply chosen based on a visual inspection of the discrepancy between the original shape (plain blue shape in Figure 5.4) and the one obtained by backward transformation (dashed red shape in Figure 5.4) of the Elliptic Fourier Descriptors of the original shape. It is worth noticing that the number of harmonics can also be fixed based on the wished maximum ℓ_1 error (in the u or v image dimensions) [116], based on the error bound provided by Giardina *et al.* [75]. In our validations, we have observed than the results do not change much when $N \geq 50$, and values of $N = 50$ and 70 have been empirically chosen.

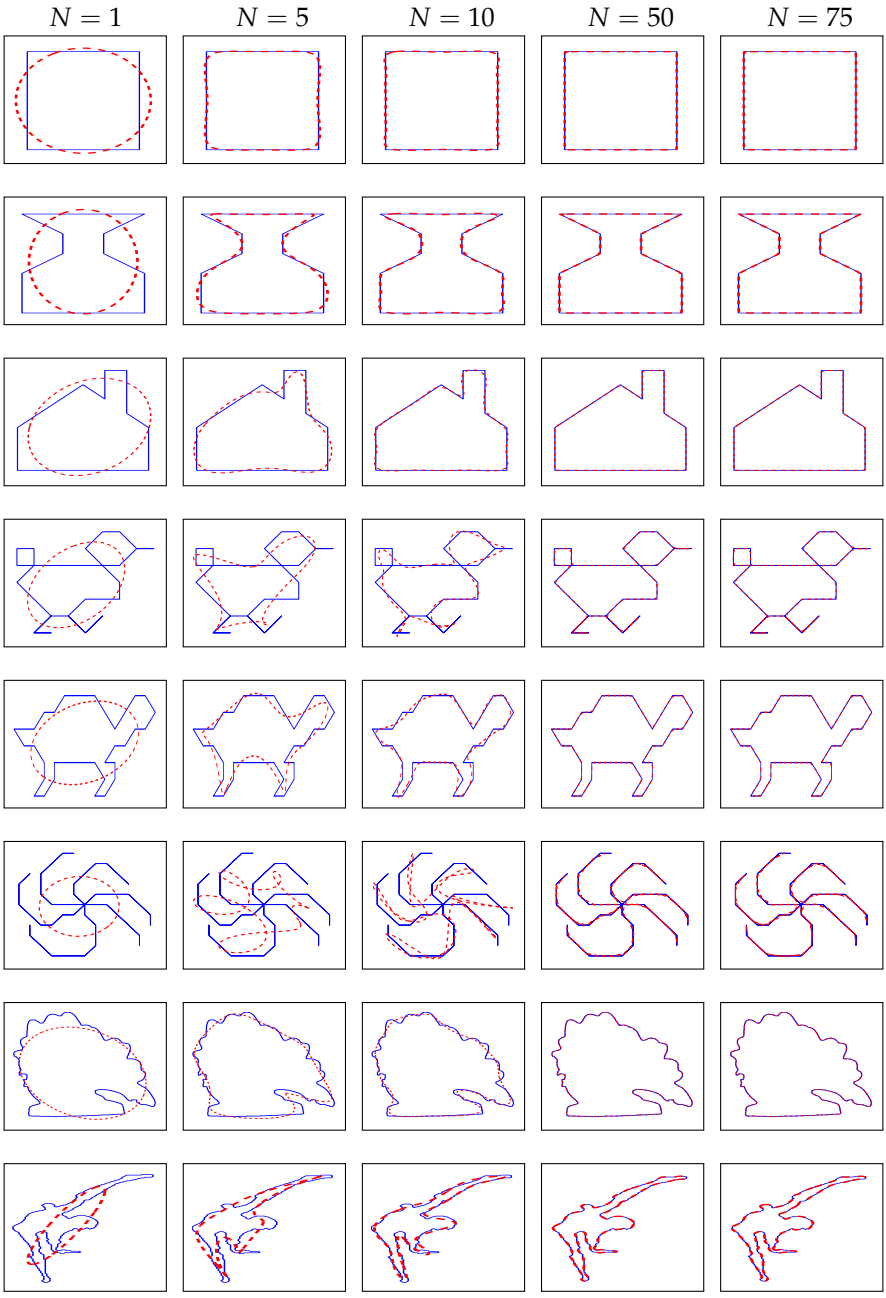


Figure 5.4: Elliptic Fourier Descriptors (EFD) are used to describe the foreground silhouettes based on a restricted number of coefficients/harmonics. The number of used harmonic N is set to $N \geq 50$ in such a way to have a low discrepancy between the original shape (plain blue shape) and the one obtained by backward transformation of the EFD (dashed red shape).

The description of a 2D shape into a restricted set of coefficients might have been done based on other descriptors than Elliptic Fourier descriptors [116]. It is worth noting that the description of a 2D shape based on Elliptic Fourier Descriptors is not considered as a contribution of this thesis, and EFD could be replaced by any other shape descriptor, as long as this descriptor is reversible (with or without losses). However, as shown in the qualitative and quantitative comparison provided in Appendix A, Elliptic Fourier Descriptors have been chosen for their good trade-off between accuracy and computation time.

5.4.2 Learning a silhouette manifold using GPLVM

We then map M instances of high-dimensional EFD feature vectors to a low-dimensional latent space that represents the different plausible silhouettes. We use a nonlinear dimensionality reduction technique called *Gaussian Process Latent Variable model* (GPLVM) [124]. This technique is used because the shape spaces are often nonlinear. Moreover, since GPVLM makes no assumption about the distribution of the latent space, it permits to work with a low dimension, while still capturing most of the shape variance.

In more details, GPLVM represents a data set $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_M]^T$, composed of M original data points (*e.g.*, M reference silhouettes represented with EFD) collected in a D dimensional space ($D = 4 \cdot N$ here), with a lower dimensional set of latent variables $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_M]^T$, where each variable is a latent point of dimensionality d, with $d \ll D$. GPLVM can be considered as a generalization of the probabilistic PCA [216] to less restrictive covariance functions, by replacing the inner product kernel, denoted $\mathcal{K}$, with nonlinear functions. Generally, the popular radial basis function kernel is used for the nonlinear mapping. GPLVM represents this mapping as a Gaussian process and determines the parameters of the mapping function in such a way that the distribution of the corresponding target data can be optimally approximated as a normal distribution. The hyperparameters $\boldsymbol{\theta}^\star$ of this Gaussian mapping are obtained based on the following optimization:

$$\begin{aligned}\{\mathbf{X}^\star, \boldsymbol{\theta}^\star\} &= \underset{\mathbf{X},\boldsymbol{\theta}}{\operatorname{argmin}} \left(P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) - \prod_{i=1}^{M} \mathcal{N}(\mathbf{y}_i|0, \mathcal{K}(\boldsymbol{\theta})) \right) \\ &= \underset{\mathbf{X},\boldsymbol{\theta}}{\operatorname{argmax}} \underbrace{\left(1 - \left(P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) - \prod_{i=1}^{M} \mathcal{N}(\mathbf{y}_i|0, \mathcal{K}(\boldsymbol{\theta})) \right) \right)}_{\text{Approximation precision}}, \end{aligned} \tag{5.1}$$

which maximizes the approximation precision of the reprojected low-dimensional points $\mathbf{X}$ with respect to the high-dimensional data $\mathbf{Y}$ [124]. As a result of this optimization from the latent space to the original data space, GPLVM keeps apart in the latent space the points that are far apart in the data space, but nothing guarantees that points that are close in the data space will also be close in the latent space. Hence, to push GPLVM to also preserve local distances, we impose *back-constraints* [125] in the computation of the latent variables. In particular, we constrain each latent variable to be a smooth mapping

from its high-dimensional counterpart. As a result, the learnt latent space becomes more adapted to our interpolation purpose, since it guarantees that the transition between two close points in the latent space maps to a smooth and topologically coherent silhouettes transition in the high-dimensional space.

As an example, on the left part of Figure 5.3, the GPLVM optimization has learnt a 2-dimensional latent space from a set of $M = 150$ shapes of silhouettes captured on video sequence representing hands' gestures and described by 35 elliptic harmonics. The colormap of Figure 5.3 represents the optimum approximation precision (cfr. Equation 5.1) of this learned latent space, where the regions with the warmest colors are more likely to represent the shape of a hand.

The set of silhouettes used for training are captured by one of the reference cameras before[3] the time at which the intermediate view synthesis is generated. In practice, the approach only requires a small amount of training samples; around 100 samples are used on average in our validation.

5.4.3 Interpolating intermediate silhouettes on the manifold

To obtain a sequence of plausible 2D silhouettes between the reference views, we first project the left and right reference silhouettes on the latent space (points 1 and 6 on the left part of Figure 5.3), based on the mapping function learnt by GPLVM [124]. Then, we use a shortest path algorithm to interpolate a plausible transition between these low-dimensional reference silhouettes, and obtain the corresponding high-dimensional silhouette prior by back-projection of this path, from the latent space to the image space. The black silhouettes on the left of Figure 5.3 illustrates the silhouettes obtained by back-projection (from the latent space to the shape space) of the latent points represented in white, on the left part of the figure. More precisely, because the transition in the intermediate views must represent the 2D appearance of a 3D object, we constrain it to the latent points for which the approximation precision is the highest. Practically, the shortest geodesic path (white path in Figure 5.3) is computed using Dijkstra's algorithm [48] with a transition cost c_{ij} from node i to j that is inversely proportional to the precision of j ($c_{ij} = -\log(\text{precision}_j + \epsilon)$, where ϵ avoids numerical instabilities).

5.4.4 Registering the silhouette priors with the reference ones

The set of prior foreground silhouettes obtained in the previous section represent a smooth and topologically consistent interpolation between the projections of the two reference silhouettes on the latent space. However, these priors describe the 2D shapes of the silhouettes, but not their position, scale and rotation. To exploit them during the EPIV reconstruction, we have thus to approximatively register them in the EPIV. This alignment is performed in three consecutive steps by:

1. Orientating the prior shapes with respect to the silhouettes observed in the reference views. The orientation of each silhouette is approximated

[3] The actual time-windows used in our validation are specified in Section 5.7.

by the angle of the first principal component of its PCA decomposition. Each shape is then rotated in such a way that its relative angle coincides with the linear interpolation of the angles of the two reference silhouettes.

2. Translating the oriented prior shapes, in such a way that their centers of mass coincide with the linear interpolation of the centers of mass of the two reference silhouettes.

3. Scaling the translated and orientated prior shapes, based on the linear interpolation of the height of the object between the two reference silhouettes.

Figure 5.5(a) shows some aligned versions of the prior shapes (in white) extracted from a linear sampling along the shortest path in a latent space representing a dinosaur. As illustrated by the red segments on this figure, the resulting silhouette priors provide, for a given epipolar line, a set of silhouette borders. Hence, they describe *a priori* a smooth transition of the reference epipolar line segments, up to the alignment inaccuracies between the blue and red segments. In the following sections, those alignment inaccuracies are considered explicitly by using translation-robust metrics when comparing the reference epipolar line segments with the prior ones.

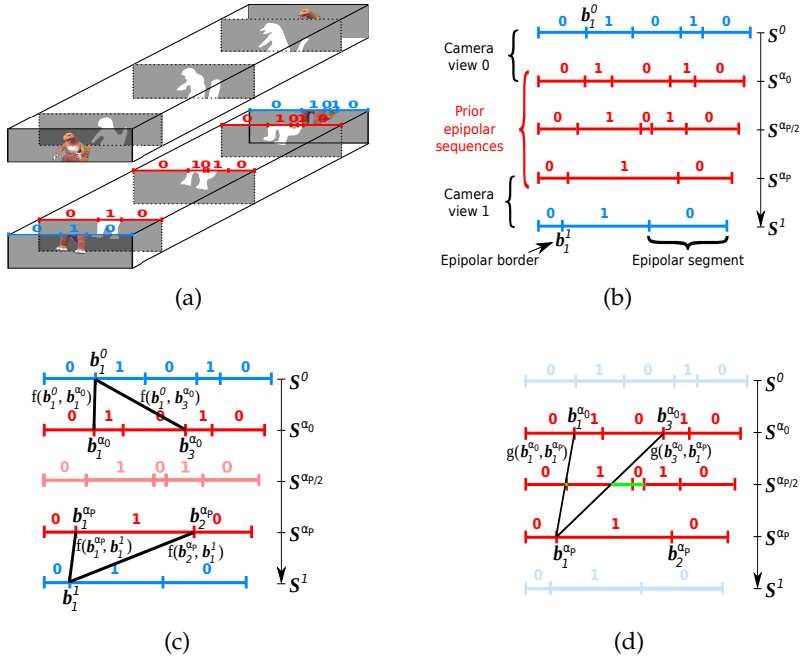


Figure 5.5: (a) Prior information about the plausible deformations of the object silhouette is used to determine the cost of matching the left epipolar border $\mathbf{b}_1^0$ to the right epipolar border $\mathbf{b}_1^1$ (b). This cost is defined by minimizing the sum of (c) the cost f of moving from a reference border to a prior one and (d) the discrepancy g with the prior (see the text for details).

5.5 Transformations of epipolar line segments

This section explains how to disambiguate the ill-posed reconstruction of the object EPIs based on a sequence of 2D silhouette priors, as obtained in the previous section. As illustrated on the right side of Figure 5.3, our approach estimates how the object epipolar line segments evolve when moving the viewpoint from one reference view to the other. Due to the epipolar rectification of the reference images, the set of possible geometric transformations of a foreground (background) epipolar line segment is restricted to the combination of an horizontal translation, a 1D scaling and a potential split-up or merge with other foreground (background) epipolar line segments. In the following, without loss of generality, we represent those combined transformations based on the displacement and potential fusion of the segments' borders. We first introduce some notations.

Let $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{\mathcal{S}}]$ denote a sequence of consecutive foreground and background epipolar line segments, defined along a rectified epipolar line as illustrated on one of the blue or red lines of Figure 5.5(a). For more clarity, in the following, these reference (blue) and prior (respectively red) epipolar line segments will be represented as a front view, as shown in Figure 5.5(b). The number of segments constituting the rectified epipolar line is denoted by $\mathcal{S} = |\mathbf{S}|$. Each segment $\mathbf{s}_k \in \mathbf{S}$ (with $k \in \{1, 2, \cdots, \mathcal{S}\}$) is characterized by a binary value, denoted $v(\mathbf{s}_k)$, depending if it corresponds to foreground (1) or background (0) information, and by its normalized length $l(\mathbf{s}_k)$, relative to the length of the entire sequence $\mathbf{S}$.

We associate a sequence of epipolar borders $\mathbf{B} = [\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{\mathcal{S}}]$ to each epipolar sequence $\mathbf{S}$, where $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$ respectively represent the beginning and the end of the epipolar segment $\mathbf{s}_k$ ($\forall k \in \{1, \cdots, \mathcal{S}\}$). The position of a border is then defined as $p(\mathbf{b}_k) = \sum_{x=0}^{k} \delta(x > 0) \cdot l(\mathbf{s}_x)$, with $k \in \{0, 1, \cdots, \mathcal{S}\}$ and $\delta(.)$ being the unit function. The modality $m(\mathbf{b}_k)$ of the border $\mathbf{b}_k$ defines the kind of transition (foreground to background or background to foreground) that it supports, *i.e.*, $m(\mathbf{b}_k) = v(\mathbf{s}_k)$ if $k \in \{1, 2, \cdots, \mathcal{S}\}$, and $m(\mathbf{b}_k) = 0$ otherwise.

To determine how the 2D object silhouette, represented in each EPI by a set of epipolar borders, evolves when changing the viewpoint, we proceed in two steps:

1. We identify and match the reference epipolar borders (blue borders in Figure 5.5(a)) that have a corresponding border in the other reference view. This is done by introducing an original cost-function to drive the matching process in a way that is consistent with the available silhouette priors (see Section 5.5.1).

2. We approximate the vanishing trajectories of all the unmatched borders in a way that is consistent with the prior information (see Section 5.5.2).

These two steps are described in detail in the next sections.

5.5.1 Matching epipolar borders

For a given EPI, let $\mathbf{B}^0$ and $\mathbf{B}^1$ denote the two sequences of reference epipolar borders that delimit the epipolar segments of the left and right reference silhouettes, respectively. Thus, as illustrated on Figure 5.5(b), $\mathbf{b}_i^0$ refers to the i^{th} epipolar border in the first reference view (starting at index 0). Similarly, $\mathbf{b}_j^1$ is the j^{th} epipolar border in the second reference view. We match pairs of borders with the algorithm of Needleman and Wunsch [162] and adapt its underlying cost functions to account for our problem specificities.

The Needleman and Wunsch (NW) algorithm has been extensively used to compare sequences of characters [162]. Given an alphabet of characters $\mathcal{C}$, and a measure of dissimilarity $d(.,.)$ between any pair of characters in $\mathcal{C}$, the NW algorithm aligns two sequences of characters in a way that (1) preserves the order of the characters within each sequence [181], (2) matches the most similar characters together by minimizing the sum of dissimilarities between matched characters and (3) tolerates unmatched characters at the cost of some skipping penalty $w(.)$. Its optimization scheme, which determines the associations and unmatched characters based on the matching cost $d(.,.)$ and skipping cost $w(.)$, is described in the Appendix B of this thesis.
We now define the borders matching and skipping costs ($d(.,.)$ and $w(.)$ respectively), so as to capture the specificities of our problem, as well as to take advantage of the available intermediate prior silhouettes. In particular, we want to ensure that:

- long segments are less likely to vanish than shorter ones. In other words, borders that delimit long reference epipolar segments have less chance to be unmatched. Therefore, the skipping cost $w(\mathbf{b}_k)$ of the reference border $\mathbf{b}_k$ is defined to be equal to $\max\left(l(\mathbf{s}_k), l(\mathbf{s}_{k+1})\right)$;

- reference borders are unmatched by pairs of consecutive borders, so that their skipping can be interpreted as a vanishing/appearing segment. Since, by definition, a border separates two segments having a different foreground/background value, the modes of consecutive borders are different. Skipping borders by pairs is thus equivalent to constraining each border to only match borders having the same modality. Hence, the distance between two borders with different modalities in the two camera views should be set to ∞.

- the matching of reference borders between the two reference views shall be consistent with the prior that is available about the plausible deformation of the silhouette between the two views. The rest of this section explains how this is achieved through proper definition of the distance metric $d(.,.)$ between borders of the same modality.

Recall that the silhouette priors are represented by a sequence of $P+1$ foreground images, in which the p^{th} image, with $p \in [0; P]$, describes *a priori* the silhouette of the object as observed at a relative intermediate position $\alpha_p = \frac{p}{P}$ between the left and the right reference views. Those $P+1$ silhouette priors represent thus *a priori* a linear sampling of the continuous smooth

transformation of the silhouette from the left to the right reference views. As illustrated in Figure 5.5(a) and (b), they provide, for a given epipolar line, a set of intermediate sequences of segments $\{\mathbf{S}^{\alpha_0}, \cdots, \mathbf{S}^{\alpha_p}, \cdots, \mathbf{S}^{\alpha_P}\}$ and their associated sequences of borders $\{\mathbf{B}^{\alpha_0}, \cdots, \mathbf{B}^{\alpha_p}, \cdots, \mathbf{B}^{\alpha_P}\}$. We define the cost of matching a border in $\mathbf{B}^0$ with a border in $\mathbf{B}^1$ by measuring how it is in-line with the prior sequences $\mathbf{B}^{\alpha_p}$ (with $p \in [0; P]$).

To account for the fact that the alignment of the prior silhouettes in the EPIV is prone to a translation error (as discussed in Section 5.4.4), we decompose the cost of matching the i^{th} border of $\mathbf{B}^0$ with the j^{th} border of $\mathbf{B}^1$, *i.e.*, $d(\mathbf{b}_i^0, \mathbf{b}_j^1)$, into two metrics. The first metric measures the quality of the alignment, in each reference view, between the prior and the reference borders. It is defined to be independent of a global and rigid translation of the prior. The second metric estimates how well the association of two prior borders that are extracted from the left and right viewpoints (corresponding to α_0 and α_P respectively), is supported by the intermediate prior borders ($0 < \alpha_p < 1$).

Precisely, the **first metric**, illustrated in Figure 5.5(c), quantifies the likelihood of matching each reference epipolar border of $\mathbf{B}^0$ (respectively $\mathbf{B}^1$) with each of the prior borders of $\mathbf{B}^{\alpha_0}$ (respectively $\mathbf{B}^{\alpha_P}$) observed from a reference viewpoint. To define the associativeness $f(\mathbf{b}_i^0, \mathbf{b}_k^{\alpha_0})$ between the i^{th} reference border of $\mathbf{B}^0$, *i.e.*, $\mathbf{b}_i^0$, and the k^{th} border of $\mathbf{B}^{\alpha_0}$, *i.e.*, $\mathbf{b}_k^{\alpha_0}$, we rely on the fact that two borders are likely to be in correspondence when they share similar neighborhood. Because $\mathbf{S}^{\alpha_0}$ and $\mathbf{S}^0$ are seen from the same camera viewpoint, the foreshortening effect does not influence the length of their epipolar segments. This cost can be measured by the complementary of the normalized Hamming correlation (detailed in the Appendix C of this thesis), *i.e.*, the number of positions in which the reference and prior sequences have identical values when they are aligned on the borders of interest. We highlight the fact that this metric is invariant to a rigid translation and is thus adapted to consider the translation error-prone prior. The metric $f(\mathbf{b}_l^{\alpha_P}, \mathbf{b}_j^1)$ to match the l^{th} prior border in $\mathbf{B}^{\alpha_P}$ with the j^{th} reference border in $\mathbf{B}^1$, observed in the other reference view, is defined similarly.

The **second metric** evaluates the cost of associating a border of the first prior $\mathbf{B}^{\alpha_0}$ with a border of the last prior $\mathbf{B}^{\alpha_P}$, as illustrated on Figure 5.5(d). We assume a linear displacement between two corresponding borders. With rectified images, the linearity is strictly verified when the silhouette borders correspond to the same physical 3D point [41], independently of the viewpoint. In other cases, since the actual 3D point supporting the silhouette border generally does not move a lot when changing the viewpoint, the linearity assumption is also reasonably valid. Hence, we evaluate the discrepancy between a linear displacement and the actual transformations given by the priors $\mathbf{B}^{\alpha_p}$ (with $p \in \{1, \cdots, P-1\}$). Formally, we define the prior deformation cost $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$ of matching the k^{th} border of $\mathbf{B}^{\alpha_0}$ with the l^{th} border of $\mathbf{B}^{\alpha_P}$, to be the sum of the ℓ_1 interpolation residues, *i.e.*, the distance between the linear interpolation of $\mathbf{b}_k^{\alpha_0}$ and $\mathbf{b}_l^{\alpha_P}$ in the intermediate views α_p, and the closest prior borders having the same modality in $\mathbf{B}^{\alpha_p}$ (with $p \in \{1, \cdots, P-1\}$). This is illustrated with green color codes in Figure 5.5(d). The formal derivation of the prior deformation cost $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$ is given in Appendix D.

Finally, the cost $d(\mathbf{b}_i^0, \mathbf{b}_j^1)$ to match the i^{th} border of $\mathbf{B}^0$ with the j^{th} border of $\mathbf{B}^1$ is defined as:

$$d(\mathbf{b}_i^0, \mathbf{b}_j^1) = \min_{k,l} \left(f(\mathbf{b}_i^0, \mathbf{b}_k^{\alpha_0}) + g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}) + f(\mathbf{b}_l^{\alpha_P}, \mathbf{b}_j^1) \right), \tag{5.2}$$

where the minimum is determined by the Dijkstra's algorithm [48]. By construction, a small $d(\mathbf{b}_i^0, \mathbf{b}_j^1)$ reflects the existence of a prior border that moves smoothly while going from one extreme prior view to the other (*i.e.*, small $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$), and a good coherence between the prior and the actual reference borders in each reference view (*i.e.*, small $f(\mathbf{b}_i^0, \mathbf{b}_k^{\alpha_0})$ and $f(\mathbf{b}_l^{\alpha_P}, \mathbf{b}_j^1)$ values). Thereby, a small $d(\mathbf{b}_i^0, \mathbf{b}_j^1)$ promotes the matching of the borders $\mathbf{b}_i^0$ and $\mathbf{b}_j^1$.

Using $d(\mathbf{b}_i^0, \mathbf{b}_j^1)$ and $w(\mathbf{b}_i^0)$, the NW algorithm determines the optimal borders associations, and identifies (pairs of) unmatched borders.

5.5.2 Appearing/vanishing trajectories

We now present an original method to handle vanishing trajectories of unmatched borders. This is equivalent to analyzing how occluded parts vanish or appear when changing the viewpoint. As one of the most original contribution of this research, we now show that it is possible to estimate how occluded parts vanish/appear when changing the viewpoint in-between the reference views. Since we know in which reference view the occluded epipolar segment[4] is visible, we consider the vanishing when moving from this view to the other, and assume that the learnt latent space embeds an instance of vanishment of this occluded part. As illustrated in Figure 5.6, our method estimates, from the prior, the speed at which each occluded segment shrinks (vanishes) when changing the viewpoint.

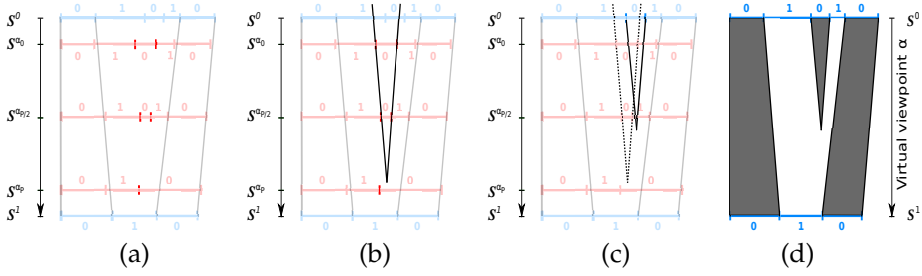


Figure 5.6: The vanishing trajectories are estimated by (a) identifying the occluded prior borders (in dark red), (b) fitting linear trajectories to these prior borders and (c) associating the slope of these trajectories (dotted lines) to the occluded reference borders (plain lines). Finally, by (d) adding the vanishing trajectories to the set of trajectories describing the transitions between the associated reference borders, the EPI of the object silhouette is reconstructed.

Since the borders displacements along the EPI are linearly proportional to α [41], we only have to evaluate the two constant border displacement speeds

[4] An occluded epipolar segment is defined by two consecutive occluded epipolar borders.

and propagate this prior information to the occluded reference segments. This is done as follows:

1. Identifying the prior borders that correspond to a segment that is subject to occlusion. We name them occluded prior borders (dark red borders in Figure 5.6(a)).

2. Fitting linear trajectories to these prior borders (Figure 5.6(b)).

3. Associating the slope (vanishing speed) of each of these linear trajectory to one of the occluded reference border (Figure 5.6(c)).

We present each of these steps in detail in the following.

Identification of prior borders defining the occluded prior segments

Obviously, only the prior borders that do not support one of the association/matching of reference borders computed by the algorithm presented in Section 5.5.1 should be considered to explain the vanishing of occluded segments. Hence, we first select as *occluded prior borders* the prior borders that are sufficiently far from the linear trajectories followed between the pairs of associated reference borders, or more specifically between their corresponding priors $\mathbf{B}^{\alpha_0}$ and $\mathbf{B}^{\alpha_P}$ at the reference viewpoints. In our experiments, we have used a simple heuristic threshold, set to 5% of the image width, to decide whether a prior border is sufficiently far from the linearly interpolated trajectories. This may however lead to many false positive. Hence, the following section proposes a robust way to estimate the vanishing/appearing paths from this initial set of occluded prior borders.

Robust fitting of linear trajectories

This section shows how to determine the linear trajectories of the l occluded reference borders from an imperfect set of prior occluded borders. Precisely, the set of l occluded reference borders can be divided into l_0 occluded reference borders representing a transition from foreground to background (*i.e.*, having a mode value of 0) and l_1 borders representing a transition from background to foreground, such that $l_0 + l_1 = l$. Hence, we propose to divide the set of prior occluded borders into two sets, based on their modes. Then, l_0 linear trajectories (respectively l_1 linear trajectories) are estimated on the subset of occluded prior borders having a mode of 0 (respectively 1). This is done by sequentially applying l_0 times (respectively l_1 times) the RANSAC algorithm [58], *i.e.*, by estimating a linear trajectory on the subset of occluded prior borders of mode 0 (respectively 1), removing the prior borders that are inlier to this estimated model, estimating a new linear trajectory on this new subset, and so on. At each RANSAC iteration, two borders are randomly selected from the set of occluded prior borders, and the linear trajectory passing through these borders is estimated. All the prior borders located in a small and conservative ℓ_1 distance (*e.g.*, 5% of the width of the image) are considered as inliers to the trajectory model. This simple greedy algorithm appears to work well in practice, due to the relatively small amount of outliers in the

set of occluded prior borders. The linear model that maximizes the amount of inliers is considered as the optimal model of the l_i^{th} sequential application of RANSAC.

Assignment of linear trajectories to the reference occluded borders

We want to assign the trajectories computed from the prior occluded borders to the unmatched borders in the reference views, so as to transfer their slope, *i.e.*, the constant speed at which the borders move along the EPI when the viewpoint index α changes. The process is illustrated in Figure 5.6(c). The cost of assigning a prior trajectory to a reference border is simply defined to be the ℓ_1 distance between the border and the position defined by the trajectory prior in the reference view (compensated with a linear interpolation of the translation error indicated by the matches of the NW algorithm). The assignment problem is then solved using the Hungarian algorithm [117], so as to assign one and only one trajectory to each unmatched border while minimizing the sum of assignment costs. Finally, as illustrated on Figure 5.6(d), these vanishing trajectories are added to the set of trajectories describing the transitions between the associated reference borders to form the EPI of the object silhouette.

5.6 View synthesis

This section describes how a view is synthesized based on the estimation of the trajectories followed by the reference epipolar borders. We propose to synthesize the intermediate views by combining the textures of matched epipolar line segments and by propagating the texture of occluded line segments from the reference view in which those segments are visible.

Texturing an intermediate view by combining the textures of its corresponding elements in the both views has been deeply investigated in the past [185] [51]. By favoring the piecewise smoothness of the intermediate texture, most of these state-of-the-art methods permit to generate pleasant intermediate views despite corrupted matches. In contrast, in order to fairly validate our contribution, *i.e.*, the estimation of the geometric transformations of the epipolar line segments, we propose to simply rely on view morphing [185], which is not robust to corrupted matches. Indeed, it does not impose piecewise smoothness of the texture, so that any wrong border match results in highly noticeable discontinuities in textures.

More precisely, view morphing relies on epipolar rectification to synthesize the intermediate textures by linear interpolation of the reference textures, such as:

$$\mathcal{I}_\alpha(u_\alpha, v) = (1-\alpha)\cdot\mathcal{I}_0(u_0, v) + \alpha\cdot\mathcal{I}_1(u_1, v),$$

with $\mathcal{I}_0$ and $\mathcal{I}_1$ the rectified reference images, $\mathcal{I}_\alpha$ the reconstructed intermediate image, u_0 the u coordinate of a pixel of $\mathcal{I}_0$, v its fixed ordinate (studied

scanline). The pixel abscissa u_α and u_1 are computed as follow:

$$\begin{aligned} u_\alpha &= (1-\alpha)\cdot u_0 + \alpha \cdot u_1 \\ u_1 &= \frac{l(\mathbf{s}_j^1)}{l(\mathbf{s}_i^0)} \cdot \Big(u_0 - p(\mathbf{b}_i^0)\Big) + p(\mathbf{b}_j^1), \end{aligned}$$

with $\mathbf{s}_i^0$ denoting the epipolar line segment including u_0, $\mathbf{s}_j^1$ denoting the epipolar line segment matched to $\mathbf{s}_i^0$. Then $p(\mathbf{b}_i^0)$ is the position of the left border of the epipolar line segment including u_0, and $p(\mathbf{b}_j^1)$ defines the position of the corresponding matched border.

The proposed method propagates the texture for occluded segments with a similar principle, although the interpolation is done between a pixel in a reference view and the segment's vanishing point determined by the intersection (u_v, α_v) of the vanishing trajectories surrounding this occluded segment. This synthesis is defined as follows:

$$\mathcal{I}_\alpha\Big(\frac{\alpha}{\alpha_v}\cdot(u_v - u_1) + u_0, v\Big) = \mathcal{I}_0(u_0, v),$$

if the vanishing epipolar line segment belongs to the left reference image (occlusion), or

$$\mathcal{I}_\alpha\Big(\frac{(\alpha-1)\cdot(u_v-u_1)}{\alpha_v - 1} + u_1, v\Big) = \mathcal{I}_1(u_1, v)$$

if the vanishing segment belongs to the right reference image (disocclusion). In contrast to conventional morphing strategies, the synthesized images represent both the parts that are visible in the two reference views, and the parts that are visible in a single reference view.

5.7 Results

In this section, we demonstrate the performance of our approach on well-known datasets, namely the synthetic *Kung-Fu Girl* sequence [149], the real *Dino* [183] and *Ballet* sequences [252]. Although these multi-view datasets contain numerous images acquired by multiple (small-baseline) cameras, we only consider a pair of widely separated cameras from these sets to learn our shape priors model, and to reconstruct the intermediate views.

For each dataset, we interpolate five intermediate views uniformly sampled in-between the left and right reference views[5]. To show the advantage of using epipolar line segments as basis matching element, we provide reconstructed views when pixels are chosen as basis matching elements. To demonstrate the benefit of the silhouette priors, we also provide the views that have been reconstructed without silhouette priors to disambiguate the epipolar segments matching. We also compare the reconstructed intermediate views resulting from our method with the ones obtained by three other conventional and state-of-the-art methods.

[5]We encourage the reader to refer to videos provided at `http://infoscience.epfl.ch/record/200492` to observe the continuous transition from the left to the right cameras.

5.7.1 The *Kung-Fu Girl* dataset

For the *Kung-Fu Girl* dataset, we have selected two wide-baseline cameras separated by an angular difference of 45°. The view captured by the left camera (or right camera) is shown on the left (respectively right) of the first row in Figure 5.7. The image shown in-between corresponds to the ones captured by a camera situated approximatively at the middle ($\alpha \simeq 0.5$) in-between these two reference views and represents thus the ground-truth.
The second row in Figure 5.7 represents the intermediate views generated by a conventional visual-hull reconstruction [146], in which the two foreground silhouettes are projected back in the 3D world, forming two cones whose intersection defines the 3D boundary of the object. The intermediate views are obtained by projecting and texturing this 3D model onto an arbitrary viewpoint [101]. The reconstructed intermediate views perfectly show the limitations of model-based approaches in our wide-baseline stereo, namely the requirement of observing the object with a large amount of reference cameras to avoid an imprecise 3D model, leading to corrupted intermediate views.
The third row in Figure 5.7 represents the intermediate views generated when morphing [185] a dense (pixel) correspondence obtained by dynamic programming [40] [181] on corresponding epipolar lines. The matching cost is simply defined as the ℓ_2 norm of the pixels' colors and the skipping penalty $w(.)$ is arbitrary set to 0.5. Two kinds of artefacts can be observed on these reconstructed views. First, they are topologically incoherent. This can be observed in-between the legs of the Kung Fu girl, near her neck and on her left hand, where some members get apart from her body. This is due to the ill-posedness of the wide-baseline matching problem, leading to wrong pixel correspondences. Second, because the resulting matching is not smooth, holes appear in the reconstructed intermediate views. This artefact, caused by the foreshortening effect, is generally avoided by imposing a smooth disparity/depth map [190], at the price of a slow matching process.
To explicitly impose the smoothness along the epipolar lines, in the 4^{th} row of Figure 5.7, we use epipolar line segments as matching elements. The method extends the one of [40] by considering epipolar line segments (and not pixels) as basic image elements. It corresponds to the approach we have introduced in Section 5.5.1, but without prior silhouettes knowledge. Hence, each epipolar border is matched, by NW [181], only considering the $f(.,.)$ terms in the definition of $d(\mathbf{b}_i^0, \mathbf{b}_j^1)$ (see Equation (5.2)). We observe in Figure 5.7 that the reconstructed intermediate views are smoother, but still exhibiting some topologically incoherent transitions, such as shown at the level of her head.
To regularize the reconstruction of the EPIs in such a way that they provide topologically coherent intermediate views, a latent space representing plausible silhouettes of the Kung-Fu girl has been learnt on a total of 60 silhouettes captured by the two wide-baseline cameras, and observed uniformly in a time-window starting from the first frame of the sequence to 20 frames before the required transition. We thus highlight the fact that the training silhouettes do not include the silhouettes to be reconstructed. These training silhouettes have been described using 70 elliptic harmonics, and the fifth row in Figure 5.7 illustrates this latent space.

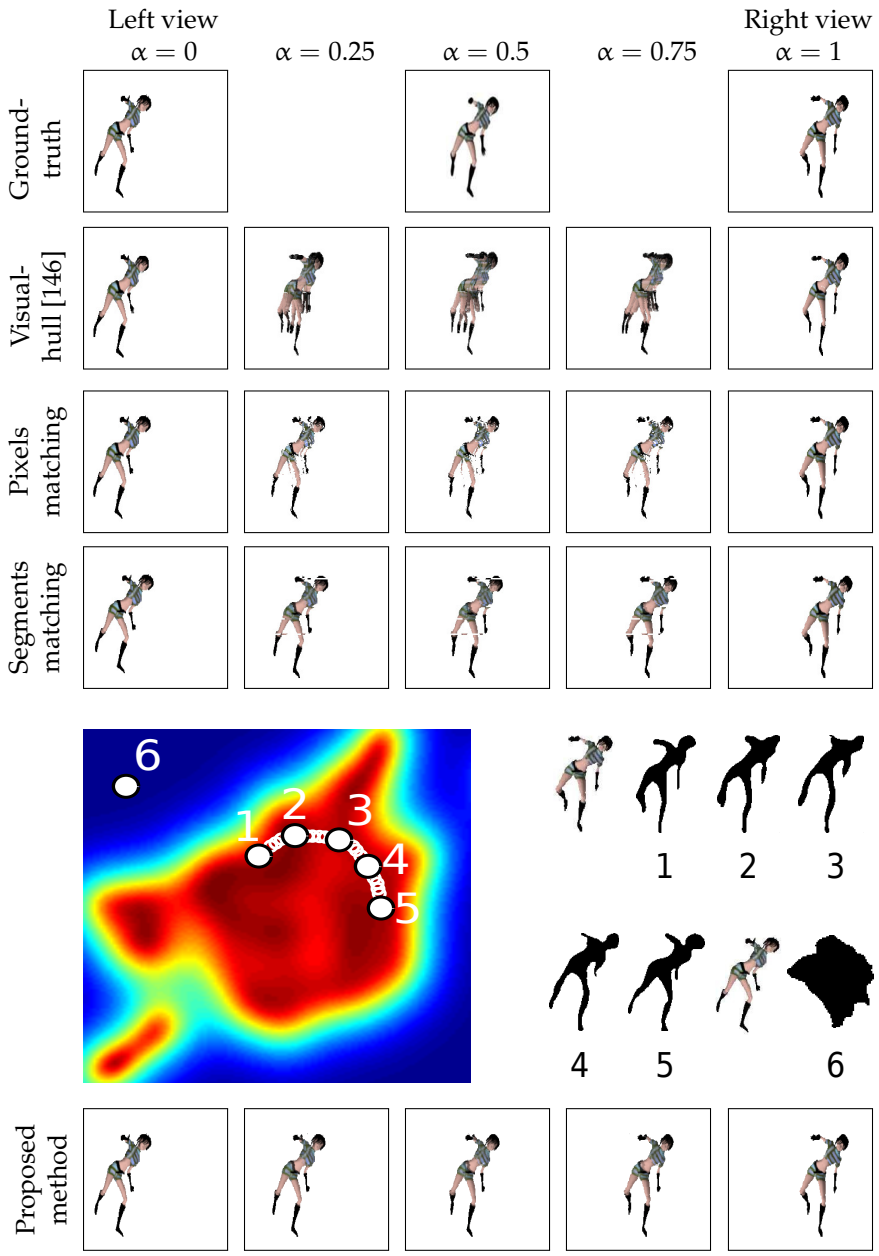


Figure 5.7: Instead of projecting an estimated 3D model [146] (second row) or determining a dense (pixel) match (third row), epipolar line segments are used as basic matching elements (fourth row). In the last row, our method regularizes the epipolar segments matching so that the shapes of the intermediate silhouettes are topologically consistent with the plausible deformations of the object silhouette, learnt and described by a low-dimensional latent space (fifth row).

The advantage of considering these priors is illustrated on the last row in Figure 5.7, where intermediate views have been generated by the method proposed in this chapter. Only 5 intermediate priors have been used to reconstruct the EPIs. We observe that the intermediate views reconstructed by our method shows a topogically coherent transition of the Kung-Fu girl from the left to the right reference view.

5.7.2 The *Ballet* sequence

The second sequence, called *Ballet* [252], has been captured using eight cameras placed along a $1D$ arc spanning about 30° end-to-end. While two neighbor cameras of this array constitute a small-baseline stereo pair, the outer cameras represent a wide-baseline configuration. Indeed, because of the small depth of the foreground dancer, strong self-occlusions and foreshortening effects can be observed between these two external viewpoints (especially on the dancer's arms). In Figure 5.8, we compare the reconstructed images at intermediate viewpoints with five methods, using only the two wide-baseline reference views (in contrast to the use of the small-baseline multi-views pairs, as done in [252]). On the first row, the intermediate views are generated by view morphing, based on multiple depth maps as proposed in [252][6]. Since the depth map estimated from the extreme wide-baseline views is very poor, we provide the images reconstructed from the textures in the two extreme views, based on the depth maps computed with neighbor cameras (small-baseline configuration). Even with this additional information, the small depth inaccuracies (equivalently weak pixel correspondences) lead to merging of non-corresponding textures, *i.e.*, ghosting artefacts. The second row of Figure 5.8 illustrates the intermediate views obtained by a state-of-the-art stereo method [158], top-ranked in February 2015 in the well-known Middlebury Stereo Evaluation [180] [97] [181]. By combining a cost-filtering approach, especially adapted to manage the occlusions, with a global (fully connected Markov Random Field) optimization, which imposes the smoothness of the disparity map, their method achieves impressive results on small-baseline stereo setups. However, as expected, the strong geometrical and photometric changes, as well as the foreshortening effects affecting our wide-baseline stereo setup make this algorithm pretty vulnerable, especially due to the over-smoothing of the disparity map. In the third row, we use only the external views and test wide-baseline stereo matching by applying the Needleman-Wunsch algorithm (dynamic programming [181]) on pixels, as done in [40]. We observe that the strong foreshortening effect produces holes in the reconstructed intermediate views. By applying dynamic programming on the segment representation, dense correspondences have been found, but topological inconsistencies subsist (see fourth row on Figure 5.8). Because of the ill-posed nature of the problem, the lowest cost match does not necessarily give the optimal match in terms of topological consistency of the silhouette.

The last row in Figure 5.8 illustrates the result obtained by our complete method using silhouette priors. The latent space has been learnt on 40 sil-

[6]The pixel correspondences are obtained by projection of the pixels of one reference view at the depth indicated by the depth map, and back-projection of these 3D points in the other reference view.

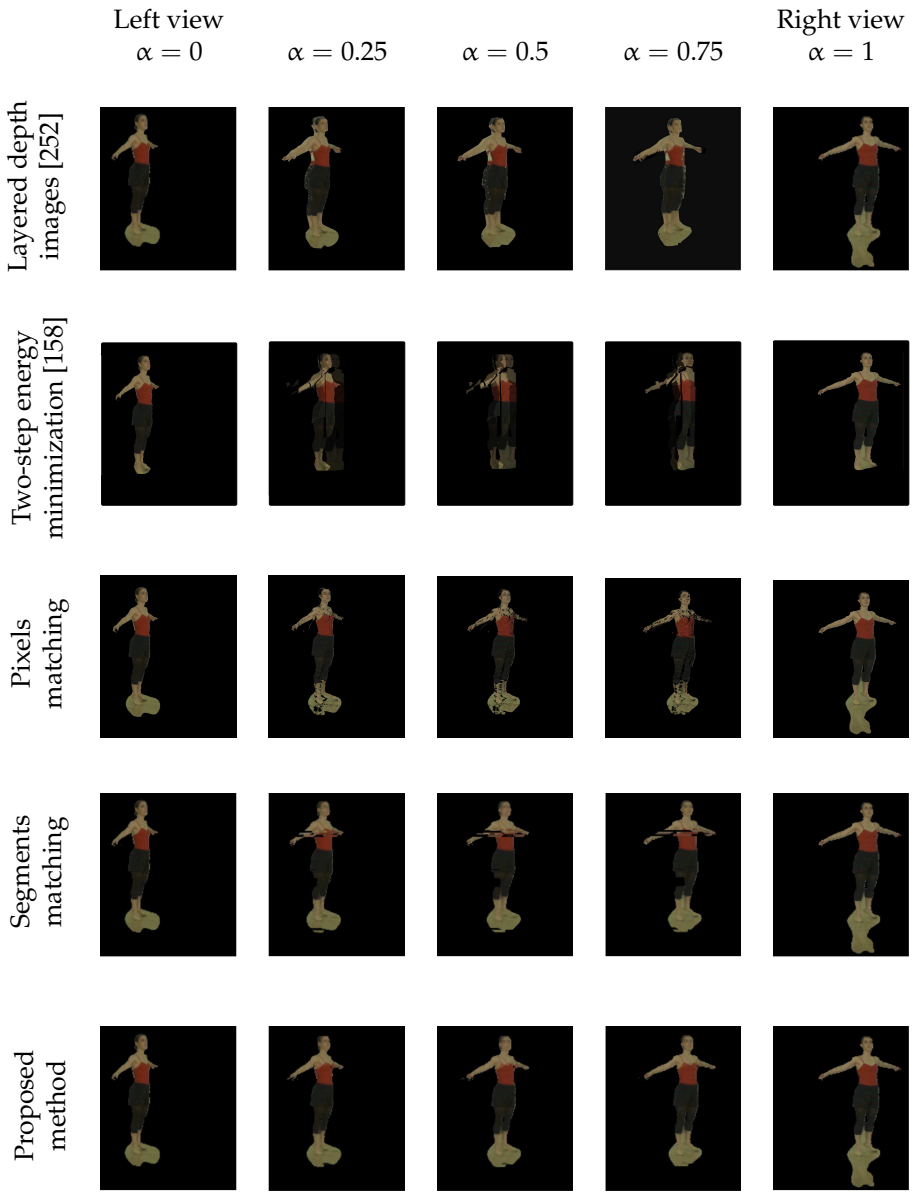


Figure 5.8: Comparison between the interpolated intermediate views generated based on matching of layered representation [252] (first row), which exploits intermediate depth maps in addition to external views, a state-of-the-art method [158] which is top-ranked in the Middlebury Stereo Evaluation [180] (second row), dynamic programming on pixels [40] (3^{rd} row), dynamic programming on our proposed epipolar line segment representation (4^{th} row) and our method (5^{th} row).

houettes observed by the two outermost cameras in a time-window 20 frames away from the transition time. These silhouettes have been described with 50 harmonics of Elliptic Fourier Descriptors, and 6 intermediate priors are used to regularize the determination of the segments' transformations.

5.7.3 The *Dino* sequence

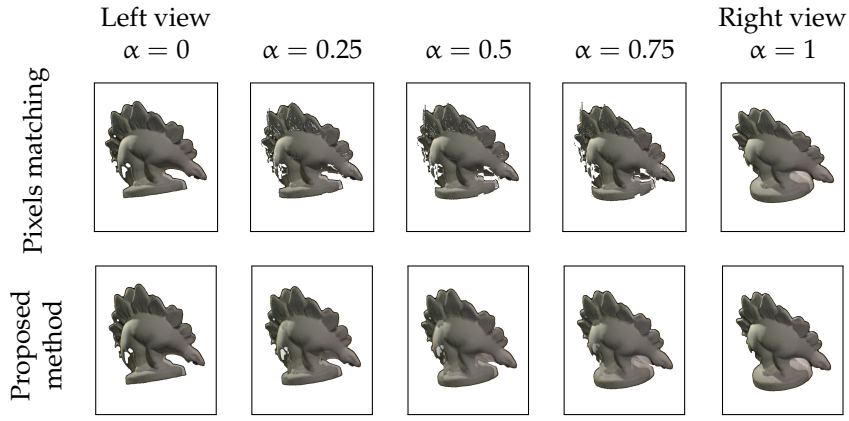


Figure 5.9: The intermediate views have been reconstructed based on only two wide-baseline cameras (relative angle of 31°).

For the *Dino* dataset, we have selected a stereo pair having a relative angle of 31°. The first row of Figure 5.9 illustrates the reconstructed views obtained by matching the color pixels by dynamic programming, as done in [40]. As observed on the other datasets, these intermediate views are affected by holes and form a topologically incoherent transition from the left to the right reference view. The second row in Figure 5.9 illustrates three intermediate views obtained by the proposed reconstruction of the EPIs. Because only one image is captured per camera in this dataset, a set of plausible Dinosaur's silhouettes can not be observed from images captured by extreme cameras at previous timestamps. We thus learn the latent space based on 160 Dinosaur's silhouettes observed by cameras situated at least 15° away from the baseline of the stereo pair. We highlight the fact that we don't use, for the training, cameras that could lie on the interpolated path. Each of these silhouettes have been described based on 70 elliptic harmonics, and 5 prior intermediate silhouettes have been extracted by linearly sampling the optimal transition on the object manifold, as explained in Section 5.4.4.

As observed on the other datasets, the intermediate views obtained by regularizing the matching of epipolar line segments (illustrated on the second row in Figure 5.9) provide a topologically coherent transition from the left to the right reference view, as opposed to the ones obtained by simply matching the color pixels with the NW algorithm [40] (first row in Figure 5.9).

5.7.4 Discussion

In contrast to the previous methods, we obtain topologically coherent intermediate views, thanks to the additional silhouette prior obtained from the latent space. Our method also efficiently deals with the foreshortening effect that is typical in wide-baseline configurations, as it can be seen on the front part of the chest of the dancer, which is severly slanted in the left reference view, while almost fronto-planar in the right one. Finally, to the best author's knowledge, the work presented in this chapter is the first one to infer the trajectories of occluded parts, allowing to interpolate their content in intermediate views, as shown by the left shoulder of the Kung-Fu girl or the space in-between the legs of the dancer. Next to those very encouraging results, four limitations of our approach however deserve to be pointed. The first one can be observed on the reconstructed fingers of the dancer's right hand in Figure 5.8. Indeed, when changing the viewpoint from the left to the right reference view, her fingers detach from her hand, showing a topologically incoherent transition of her right hand. This is due to the limited accuracy of the priors, determined from a low-dimensional space representing the approximations of training shapes as a set of N smooth harmonics (ellipses). When the high frequencies details, such as the dancer's fingers, are not represented by the priors, their matching can not be regularized, and their transition might become topologically incoherent.

As a second limitation, it is worth noting that the extraction of a sequence of intermediate silhouettes from the latent space rely on the assumption that most pairs of object silhouettes observed at the same time from the reference viewpoints are different (and project in distinct points on the manifold). Enriching the silhouettes descriptors so as to account for its local temporal evolution could be a way to reduce the risk to observe very similar (spatio-temporal) silhouettes from the two reference viewpoints. In other words, introducing some temporal consistency in the learning and interpolation procedure could be a path to mitigate this requirement, and would be an interesting topic for future research.

The third weakness of our method comes from the choice of using epipolar line segments as matching units. Indeed, while it permits to explicitly take into account the foreshortening effect and the sparse correspondence problem, the precise correspondence of their inner pixels is not known, and can only be inferred from the knowledge of the matches of their borders. The simple linear interpolation of inner textures, as detailed in Section 5.6, may result in wrong matches of the inner pixels if the surface described by this epipolar line segment is not planar. This limit can be observed on the pixels representing the straps on the chest of the dancer, which are not correctly matched by linearly interpolating inside the (correctly matched) borders of the curved chest or on her face, and results in the ghosting artefact. This artefact could be reduced by generalizing the texture interpolation to convex surfaces, *e.g.*, based on floating textures [51].

The last weakness of our method is that it associates the borders of the (silhouette) epipolar segments, even when they actually do not correspond to the same 3D point. When combined with linear texture interpolation, such erroneous associations of border segments generally lead to a stretching/shrinking

of the inner textures, which leads to a (slight) horizontal magnification (shrinkage) of the object. However, because these magnifications (shrinkages) affect smoothly the object along its height, they are unnoticeable when the inner textures are uniform. For non-uniform inner textures, we propose to circumvent this problem by decoupling the matching (and interpolation) of pixels between the views from the matching of silhouette segments. Specifically, we propose to drive the interpolation of the intermediate views based on a (dense) pixels association inside the matched epipolar segments, *e.g.*, based on the strict preservation of the pixels' order. Such approach to interpolate the texture between matched segments supports the occlusion of pixels between the two views when relevant, and consequently does not force the association of pixels located on corresponding silhouette borders. This follows the principle exploited in the floating textures [51], in which an optical-flow strategy is used to refine the pixel matching after the coarse texture association.

Finally, we note that the processing time of our algorithm (Matlab implementation, Intel I5 CPU 2.4GHz and 8Gb of RAM) shows encouraging performances (on average 4.2s to describe a 768×1024 image into epipolar line segments, 0.06s to match all the epipolar lines independently and 0.16s to render an intermediate view). Moreover, because the epipolar lines are processed independently, real-time implementation is within reach, *e.g.*, based on GPU parallelization.

5.8 Conclusion

In this chapter, we have proposed a new and original interpolation technique for intermediate view synthesis between cameras in wide-baseline configurations. We also notice that although this coherence is imposed independently on each epipolar line, the fact that these constraints are derived from 2D priors favors consistency along the epipolar lines. Our method relies on prior information about the silhouettes of objects in the intermediate views to guarantee consistency between the synthesized silhouettes and the ones present in the two reference viewpoints. As a first contribution, these silhouette priors are learnt by reducing the dimensionality of Elliptic Fourier shape Descriptors, accumulated over a training set of representations of the objects under consideration, typically from earlier observations of the object moving in front of the wide-baseline camera pair. This additional information is then exploited to determine the 1D transformation of epipolar line segments when moving from one view to the other. As a second contribution, this new framework has not only the advantage of generating consistent and smooth virtual transitions between the viewpoints where correspondences can be found in the two basis images, but it can also handle the vanishing of occluded informations. Finally, we have demonstrated that our method outperforms state-of-the-art view interpolation methods by generating topologically coherent intermediate views of an object, despite the multiple occlusions and severe foreshortening effect that are typical in wide-baseline configurations.

CHAPTER 6
Overall conclusion of this thesis

Nowadays, video viewers are restricted to observe a filmed scene from the point of view of one of the cameras recording it. This thesis proposes solutions to extend the range of possible viewpoints to the baseline separating two calibrated cameras observing the scene, including when they are widely separated. This wide-baseline stereo setup offers the viewer a wider range of viewpoints than its small-baseline counterpart, which have been studied for more than 30 years. However, the price to pay for this increased flexibility is an increased ambiguity in the view interpolation process.

This thesis investigates the three main groups of view interpolation approaches, detailed in Chapter 2, namely *image-based rendering*, *model-based rendering* and *light-field* methods. Although these three types of reconstructions, respectively exploited in Chapter 3, Chapter 4 and Chapter 5, are based on different concepts, they all share the same difficulty: they require to determine a tremedeous amount of 2D (pixel) correspondences in-between the two reference views. Indeed, the determination of 2D correspondences (or equivalently the determination of the scene's 3D) is an ill-posed problem due to the photometric changes as well as the projective geometric transformations affecting the 2D views of a single 3D region. This thesis proposes three different methods to alleviate this ambiguity.

Chapter 3 has revisited one of the most popular prior used in small-baseline configurations: the preservation of the left-right relations between the image's elements. This strict prior, which is known as the ordering constraint, is often violated in wide-baseline stereo setups, due to their multiple inherent occlusions. In this chapter, we have proposed a framework that only favors the preservation of the order of the image elements without necessary strictly forcing it. Our method does not only disambiguate the correspondences based on the order information, but also detects the occluded (no correspondence) elements. The main benefit of this prior is its wide applicability, which is paid at the price of inaccuracies in the depth estimation.

For improved accuracy, Chapters 4 and 5 have proposed to reconstruct the background and the foreground of the scene separately. This decomposition relies on the fact that the 3D geometry of the background can generally be approximated by simple piecewise-planar proxies, especially in case of a man-made scene, or when this background is far away from the cameras. Moreover, in still background cases, this geometry can be estimated only once, while the moving foreground requires to estimate an accurate and time-varying 3D model.

In Chapter 4, we have formulated the piecewise-planar approximation of the background's 3D as a plane assignment problem over image's regions, whose boundaries define the planes' borders. We relied on our own fast-color segmentation algorithm to extract these regions, and proposed an approximated dense 3D point cloud to determine the parameters of the 3D planes. We showed that the inaccuracy of these 3D data, inherent to wide-baseline stereo setups, makes any straightforward plane fitting method inappropriated, even when considering only the most reliable 3D points and random consensus-based approaches. In contrast, our method builds on a set of candidate plane models and either associates one of them to each region, or detect the region as occluded. We have proposed an effective method to define a set of planar models that includes most planar surfaces appearing in the scene, while being reasonably small in size. We also have introduced a new data-fidelity that measures how well a plane hypothesis fits a dense point cloud. Our metric is shown to be robust to noisy 3D points. The plane assignment problem is then solved jointly over the regions, while simultaneously maximizing the proposed data-fidelity and minimizing the number of planar proxies. This discrete optimization process is inspired from the state-of-the-art PEARL method [102]. Such a light-weighted, minimalist piecewise-planar representation of the background is then rendered on a virtual view by simple and fast homography projections.

Chapter 5 tackles the interpolation of the foreground using a light-field formalism. We have proposed to disambiguate the ill-posed reconstruction of a sparse light-field by constraining the interpolated views to be consistent with an object shape prior. This prior is learnt based on a small number of frames captured by the two reference views, and consists in a nonlinear shape manifold representing the plausible silhouettes of the object. We showed that such an object-specific prior can disambiguate the 2D correspondence problem and also allows to determine how the numerous occluded parts vanish/appear when changing the viewpoint. Finally, we showed that the proposed framework generates topologically consistent and smooth virtual transitions of the froeground when changing the viewpoint in-between the two reference views.

Altogether, while Chapter 3 enables the reconstruction of an arbitrary scene for which the background/foreground separation is not obvious, Chapter 4 and Chapter 5 support a more precise reconstruction based on more specific priors.

Future works

In this last section, we first suggest possible paths to improve each contribution of this thesis. After, we build upon our experience to propose novel directions of research.

Let us start with the relaxed ordering constraint, presented in Chapter 3. Just like its well-known strict version, it regularizes the matching independently on each pair of corresponding epipolar lines. This independent process along one dimension of the image guarantees the high degree of paral-

lelization of our algorithm. However, it neglects a possible continuity of the matching along the second image dimension. By neglecting this prior information, we have formulated the determination of correspondences as the research, in the similarity matrix associated to corresponding epipolar lines, of a piecewise-smooth 2D path that optimizes a cost function reflecting the quality of matching and the preservation of the relative order between the image components. To include a possible continuity of the correspondences along the second dimension of the image, we could envision to analyze the problem in the 3D cube formed by the stack of 2D similarity matrices related to the different pairs of associated epipolar lines. Instead of determining a piecewise-smooth 2D path maximizing a measure of the order, the problem is then generalized to the search of a piecewise-smooth surface in this stack. In this case, maximizing the global order of the reconstruction might be formulated as the maximization of the sum of the 2D order measures defined in Chapter 3, each one captured along a slice of this stack. The coherence along successive pairs of epipolar lines could be favored by maximizing the piecewise-smoothness, *e.g.*, the TV [33] or the Total-Generalized-Variation norm [26], along the depth of the stack. Analogously, our method could also be generalized to more than 2 views by imposing smoothness along the stack composed of the set of epipolary rectified reference views.

Another drawback of the proposed method is its asymmetry: the correspondences are determined for all the elements of one of the two views, knowing the labels assigned to the elements of the other view. To mitigate the impact of assigning a distinct label to each element of the first view before optimizing the labels of the second view, Section 3.3 has proposed to run the label assignement algorithm a second time, in the other direction (*i.e.*, from the second to the first view). However, a more elegant solution to this issue would consist in formulating the association problem in a symmetric manner. This could be done by formulating the problem directly in terms of associations between the elements of each sequence. Hence, instead of defining the label of a sequence element in terms of the labels assigned to the elements of the other sequence, one element would be associated to an element of the other sequence through the definition of a so-called *match* between them. The energy to minimize would then be defined directly in terms of (a sum over the) matches, rather than in terms of (a sum over the) labels. This work is in progress and should hopefully be published in the coming months.

Our piecewise-planar approximation of the 3D of the background, presented in Chapter 4, relies on a 3D point cloud. This point cloud can be considered as a sparse and noisy sampling of the real dense 3D of the background. Sparsity and accuracy form unfortunately a trade-off. Whatever the choice, this sparse 3D information is not sufficient to disambiguate the dense 3D, especially in the regions with uniform textures. We have chosen to add two very strong priors, namely the smoothness and a minimum number of planar proxies, to help the proposed data-fidelity to disambiguate the reconstruction. On one hand, our smoothness prior is defined based on the (normalized) gradient of the 2D image. When working on real images, this gradient is not always sufficiently outlined at the pixel's locations of the border of a surface, leading to a propagation of a 3D model through regions that do not share the same

3D model. We mitigate this drawback by assigning a very small weight to the smoothness prior, with respect to the data-fidelity term. On the other hand, the optimal planes fitted on textureless (or repetitively patterned) regions are highly depending on the weight associated to the number of assigned models. This sensitivity originates from the fact that the 3D points associated to these regions are sometimes so noisy that other planar structures support their distributions. One way to address the problem could be to not only consider the inner part of a region, but also its surrounding content. Precisely, we are investigating the incorporation of a region description, that describes robustly the spatial and colorimetric organization of its surrounding regions, in the data-fidelity term. As a last remark about Chapter 4, the proposed framework could be straightforwardly extended to more than two references views. On the one hand, using more camera views could enable to increase both the robustness of the generated 3D point cloud (*e.g.*, based on imposing the epipolar constraint by pairs, cfr. "Projective grid space" in Section 2.2.3), and the reliability of the matching inaccuracy and matching ambiguity measures (cfr. Section 4.5). On the other hand, only minor changes will be required: the data-fidelity should be computed on each segmented region in each view, and the inter-view smoothness could be applied as done in [21].

The shape prior presented in Chapter 5 allows to disambiguate only the shape of the foreground object. We simply interpolate linearly the inner textures along the epipolar lines. Such a simple linear interpolation might produce the ghosting artefact if the inner parts are not planars. Such artefact can straightforwardly be avoided by interpolating the textures based on projective texturing algorithms, such as the floating textures [51]. A second solution could rely on sparse but accurate correspondences found in the inner parts, to formulate the problem as the estimation of the deformation of a non-rigid surface given fixed boundaries. Another, more challenging, way to circumvent the problem is by learning shape priors about the inner parts of the object. This requires to (i) temporally track the inner regions and (ii) assume that two observations of the same 3D region are segmented similarly. However, these two constraints are still active and challenging areas of research (called respectively part-based tracking and 3D segmentation). As a last remark about Chapter 5, the proposed framework could be straightforwardly extended to more than two reference views, by considering the really observed epipolary rectified intermediate views as given (instead of estimated) priors in the EPIV, forcing the shape borders to pass through it.

Finally, the rest of this thesis presents more general directions of research that are worth investigating in the domain of view interpolation from wide-baseline stereo setups. We propose to classify them into three groups, namely:

1. The definition of metrics to quantify the level of ill-posedness in view interpolation.

2. The improvement of the invariance/discriminance trade-off in image descriptors.

3. The exploitation of scene-specific priors.

First, as explained in the beginning of this thesis, the notion of wide-baseline itself is not well defined, and is often linearly related to the notion of "difficulty of reconstruction". Indeed, while some experts consider only the distance separating the reference cameras as an appropriate measure of the "difficulty", some others only consider the relative angle between the reference cameras. We sincerely believe that these two criteria should be complete with 3D informations, such as the minimum depth of the scene to reconstruct, as deeply explained in Chapter 1.

Second, the main challenges of 3D reconstruction in wide-baseline stereo setups are (i) the colorimetric changes, (ii) the geometric (projective) transformations, (iii) the presence of occlusions and (iv) the ill-posedness of the reconstruction. State-of-the-art methods counter (i), (ii) and (iii) by introducing robust (pixel or region) descriptors, at the price of a decreased discriminativeness in the matching process. About the colorimetry, most of the most well-known descriptors (SIFT [134] [135], SURF [14], BRIEF [30], ORB [176], GLOH [154], etc.) focus on the image's texture, and thus systematically consider the intensity images instead of the full colorimetry. This full colorimetry is nevertheless a discriminative information, as proved by the performances of GIST [49] and CSIFT [1]. Occlusions also strongly affect the image's description. Engin Tola [218] has proposed a time-consuming solution, which replicates N times the descriptor and masks each of them differently. However, it mutliplies the matching complexity by N^2, which is often untractable. We believe that exploiting segmentation to construct appearance descriptors that are robust to occlusions, as proposed recently in [221], is an appropriate solution for the future.

Finally, this thesis pushes towards the learning and the use of object-specific priors to regularize the ill-posed 3D reconstruction. We believe that such kinds of approaches might have an impact in the future, in particular due to the recent increase of performance in object recognition. More generally, these two different tasks can be merged by autonomous (deep-)learning the features/priors, that are discriminative to (implicitly or explicitly) reconstruct the 3D of a specific type of region/object, as suggested, very recently, in [61].

CHAPTER 7
Publications

The list here-below enumerates the publications which have either been published in full-length peer-reviewed conference papers/journals or are currently under submission.

- C. Verleysen and C. De Vleeschouwer. Piecewise-planar parsimonious approximation of wide-baseline stereo scenes, under submission, 2015.
- C. Verleysen and C. De Vleeschouwer. Wide-baseline stereo matching under relaxed ordering constraint, under submission, 2015.
- C. Verleysen, T. Maugey, P. Frossard and C. De Vleeschouwer. Wide-baseline object interpolation using shape prior regularization of epipolar plane images, under submission, 2014.
- C. Verleysen and C. De Vleeschouwer. Learning and Propagation of Dominant Colors for Fast Video Segmentation, In Advanced Concepts for Intelligent Vision Systems, pages 657-668, Springer, 2013.
- C. Verleysen and C. De Vleeschouwer. Using shape priors to regularize intermediate views in wide-baseline image-based rendering, In ICVSS, 2013.
- C. Verleysen and C. De Vleeschouwer, Recognition of sport players' numbers using fast-color segmentation, In IS&T/SPIE Electronic Imaging, pages 250-360, International Society for Optics and Photonics, 2012.
- C. Verleysen, N. Merlin and C. De Vleeschouwer, Adapting JPEG2000 bit allocation to preserve features of interest, In 4th International Congress on Image and Signal Processing (CISP), volume 2, pages 601-606, IEEE, 2011.

APPENDIX A

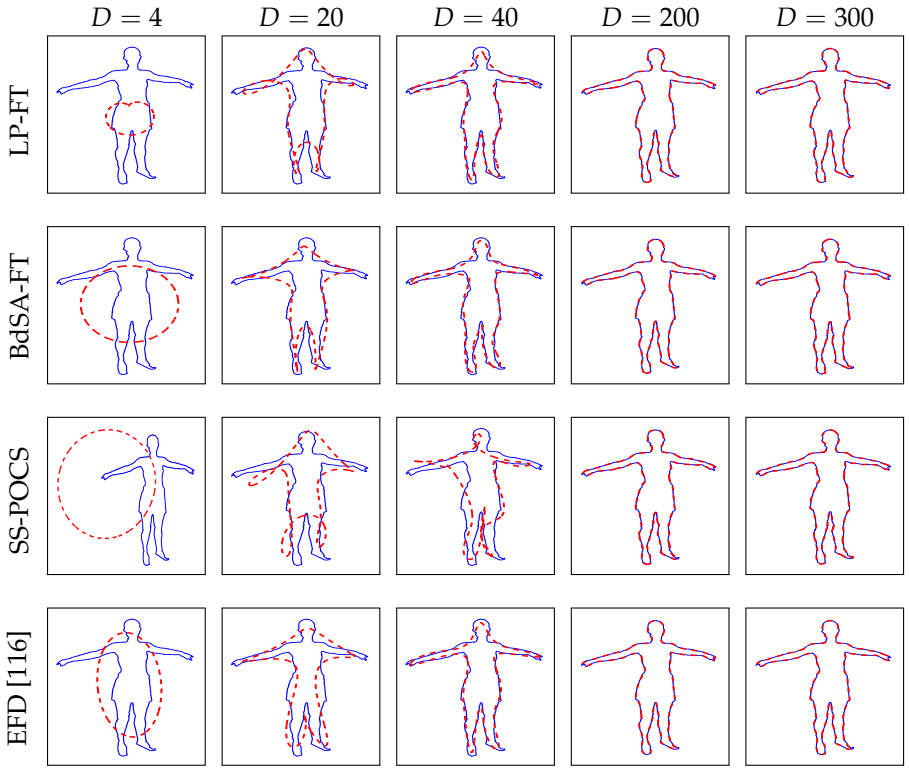
Exhaustive comparison of 2D shape descriptors (in Section 5.4.1)

The description of a 2D shape into a restricted set of coefficients might have been done based on other descriptors than Elliptic Fourier descriptors [116]. It is worth noting that the description of a 2D shape based on Elliptic Fourier Descriptors is not considered as a contribution of this thesis, and EFD could be replaced by any other shape descriptor, as long as this descriptor is reversible (with or without losses). Here below, we propose to compare both qualitatively and quantitatively (in term of SNR) the ability of EFD to represent a 2D shape based on D coefficients with three other well-known Fourier-based methods:

- The low-pass filtering of the Fourier Transform (denoted LP-FT) [7]: the 2D (closed, and thus periodic) shape is first represented in the complex domain based on its M pixel coordinates $(u_m, v_m) \in \mathbb{C}$ (with $m \in \{1, \cdots, M\}$). Then, only the Fourier (complex) coefficients associated to the $D/2$ lowest frequencies of the shape are considered, forming a description vector of D elements. Finally, the 2D shape is reconstructed by inverse Fourier transformation of these selected coefficients.
- The best D-sparse approximation of the Fourier Transform (denoted BdSA-FT): the 2D (closed, and thus periodic) shape is first represented in the complex domain based on its M pixel coordinates $(u_m, v_m) \in \mathbb{C}$ (with $m \in \{1, \cdots, M\}$). Then, only the $D/2$ Fourier (complex) coefficients having the highest amplitudes are not truncated to 0, leading to a D-sparse vector. Finally, the 2D shape is reconstructed by inverse Fourier transformation of this truncated vector.
- The spread spectrum signature with POCS[1] reconstruction (denoted SS-POCS): the 2D (closed, and thus periodic) shape is first represented in the complex domain based on its M pixel coordinates $(u_m, v_m) \in \mathbb{C}$ (with $m \in \{1, \cdots, M\}$). The spectrum of the Fourier transform of the shape is spread in the frequency domain (by applying random shifts), and $D/2$ frequencies are randomly selected to describe the 2D shape as a description vector of D real coefficients. This shape is then non-linearly reconstructed, based on a variant of the POCS method proposed in [173] (with 1000 iterations), in which the soft-thresholding is replaced by a simpler selection of the $D/4$ lowest frequencies.

Figure A.1 provides both a visual and quantitative comparison (in term of SNR and computation time) of a D-dimensional description of a shape between EFD (last row, where $D = 4 \cdot N$, N being the number of harmonics, cfr. Section 5.4.1) and the three other methods.

[1] Projections on Convex Sets



Method	SNR [dB]					Time [s]
	D=4	D=20	D=40	D=100	D=300	
LP-FT	11.76	28.55	37.3	54.61	57.62	0.0004
BdSA-FT	16.84	29.89	38.37	55.95	59.17	0.0006
SS-POCS	6.01	19.96	30.89	55.31	57.97	3.65
EFD [116]	21.77	30.89	39.52	56.78	59.83	0.0009

Figure A.1: Elliptic Fourier Descriptors (EFD, last row) offer a good trade-off between accuracy of the reconstructed shape and computation time.

The fact that BdSA-FT outperforms LP-FT in term of SNR indicates that high frequencies must also be considered in the description of arbitrary 2D shapes. By randomly selecting frequencies in the spread-spectrum, the shape description of SS-POCS also considers these high frequencies, but its low performances seem to point the inaccuracy of its associated non-linear reconstruction. Although Fourier descriptors based on spread spectrum might contain richer informations about a 2D shape than others[2], their slow and inaccurate reconstructions make them inappropriate candidates in our application. Elliptic Fourier Descriptors have then been chosen for their good trade-off between accuracy and computation time.

[2]This verification is out-of-the-scope of this thesis.

APPENDIX B

The Needleman-Wunsch algorithm (in Section 5.5.1)

The Needleman-Wunsch algorithm [162] builds a two-dimensional matrix M, whose element $M(i,j)$ measures the smallest cost to align the first i characters of the first sequence with the first j characters of the second sequence.
Let us consider two sequences of characters $C_1 = \{c_1(1), ..., c_1(L_1)\}$ and $C_2 = \{c_2(1), ..., c_2(L_2)\}$, with $c_k(l) \in \mathcal{C}$ (with $\mathcal{C}$ a predefined alphabet of characters). Let also $d(c_1(m), c_2(n))$, with $m \in \{1, \cdots, L_1\}$ and $n \in \{1, \cdots, L_2\}$, denote the distance between two characters $c_1(m)$ and $c_2(n)$ in $\mathcal{C}$. Let further $w(c)$ be the penalty induced by leaving the character $c \in \mathcal{C}$ unmatched during sequence alignment. This penalty is often called the skipping cost, and its definition is problem specific. The initialization and the recursive step of the dynamic programming algorithm that computes the $(L_1 + 1) \times (L_2 + 1)$ elements of matrix M are then defined as follows :

$$
\begin{aligned}
M(0,j) &= \textstyle\sum_{k=1}^{j} w(c_2(k)) \\
M(i,0) &= \textstyle\sum_{k=1}^{i} w(c_1(k)) \\
M(i,j) &= \min \big(\begin{array}{ll} M(i-1,j-1) & + \; d(c_1(i), c_2(j)), \\ M(i-1,j) & + \; w(c_1(i)), \\ M(i,j-1) & + \; w(c_2(j)) \; \big). \end{array}
\end{aligned}
$$

The three options in the recursive computation of $M(i,j)$ respectively correspond to matching $c_1(i)$ and $c_2(j)$, skipping $c_1(i)$, or skipping $c_2(j)$. Once M has been computed, $M(L_1, L_2)$ gives the minimal score among all possible alignments. The alignment that gives this score can be retrieved by starting from position (L_1, L_2) and observing recursively backwards which of the three decisions has been taken ($c_1(i)$ matches $c_2(j)$, $c_1(i)$ is unmatched or if $c_2(j)$ is unmatched).

APPENDIX C

Derivation of $f(\mathbf{b}_i^0, \mathbf{b}_k^{\alpha_0})$ (in Section 5.5.1)

To define the associativeness $f(\mathbf{b}_i^0, \mathbf{b}_k^{\alpha_0})$ between the i^{th} reference border of $\mathbf{B}^0$, *i.e.*, $\mathbf{b}_i^0$, and the k^{th} border of $\mathbf{B}^{\alpha_0}$, *i.e.*, $\mathbf{b}_k^{\alpha_0}$, we rely on the complementary of the normalized Hamming correlation, which is a translation-invariant metric. It measures the number of positions in which the reference and prior sequences have identical values when they are aligned on the borders of interest, and is expressed as:

$$f(\mathbf{b}_i^0, \mathbf{b}_k^{\alpha_0}) = 1 - \sum_{u \in \mathcal{E}} \frac{\mathcal{I}_0\left(u - p(\mathbf{b}_i^0), v\right) \oplus \mathcal{I}_{\alpha_0}\left(u - p(\mathbf{b}_k^{\alpha_0}), v\right)}{|\mathcal{E}|},$$

with

$$\mathcal{E} = \left\{ \min\left(p(\mathbf{b}_i^0), p(\mathbf{b}_k^{\alpha_0})\right), \cdots, \min\left(w_0 \cdot \left(1 - p\left(\mathbf{b}_i^0\right)\right), w_{\alpha_0} \cdot \left(1 - p\left(\mathbf{b}_k^{\alpha_0}\right)\right)\right) \right\}$$

where w_0 and w_{α_0} represent the width of the reference image $\mathcal{I}_0$ and prior image $\mathcal{I}_{\alpha_0}$ respectively, $\oplus$ is the binary XOR operator, u refers to an image's abscissa coordinate and v to an ordinate.

APPENDIX D

Derivation of $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$ (in Section 5.5.1)

We derive the prior deformation cost, *i.e.*, the cost of associating a border of the first prior $\mathbf{B}^{\alpha_0}$ with a border of the last prior $\mathbf{B}^{\alpha_P}$. To determine the prior deformation cost $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$ of matching the k^{th} border of $\mathbf{B}^{\alpha_0}$ with the l^{th} border of $\mathbf{B}^{\alpha_P}$, we first define $r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_q^{\alpha_p})$ as the discrepancy between the q^{th} border of the p^{th} prior and the linear transition from $\mathbf{b}_k^{\alpha_0}$ to $\mathbf{b}_l^{\alpha_P}$:

$$r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_q^{\alpha_p}) = \begin{cases} \infty & \text{if } \; m(\mathbf{b}_k^{\alpha_0}) \neq m(\mathbf{b}_l^{\alpha_P}) \\ & \text{or } \; m(\mathbf{b}_k^{\alpha_0}) \neq m(\mathbf{b}_q^{\alpha_p}) \\ & \text{or } \; m(\mathbf{b}_l^{\alpha_P}) \neq m(\mathbf{b}_q^{\alpha_p}) \\ \\ \big((1-\alpha_p)\cdot p(\mathbf{b}_k^{\alpha_0}) + \alpha_p \cdot p(\mathbf{b}_l^{\alpha_P})\big) - p(\mathbf{b}_q^{\alpha_p}) & \text{otherwise} \end{cases}$$

and $q^*(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{B}^{\alpha_p})$ as the index of the border of $\mathbf{B}^{\alpha_p}$ with the smallest discrepancy:

$$q^*(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{B}^{\alpha_p}) = \underset{q}{\operatorname{argmin}} \left| r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_q^{\alpha_p}) \right|,$$

and $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$ as the weighted sum of these discrepancies in the intermediate views:

$$g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}) = \begin{cases} \infty & \text{if } \; m(\mathbf{b}_k^{\alpha_0}) \neq m(\mathbf{b}_l^{\alpha_P}) \\ \sum_{p=1}^{P-1} \left(1 - \frac{w_p}{\sum_{n=1}^{P} w_n}\right) \left| r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_{q^*}^{\alpha_p}) \right| & \text{otherwise,} \end{cases}$$

with

$$w_p = \min\Big(\Big| \sum_{x=1}^{p} \frac{\Delta r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_{q^*}^{\alpha_x})}{\Delta x} \Big|, \\ \Big| \sum_{x=1}^{p} \frac{\Delta r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_{q^*}^{\alpha_{P-1-x}})}{\Delta x} \Big|\Big).$$

The cost $g(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P})$ is small when the prior validates a smooth (linear) displacement from $\mathbf{b}_k^{\alpha_0}$ to $\mathbf{b}_l^{\alpha_P}$, and thereby supports their matching. The weight w_p enables to relax the constraint on the smoothness of the transition of epipolar borders, and thus of the prior shapes.

Indeed, as described in Section 5.4, the prior shapes are extracted from a smooth transition in a space of lower dimension, *i.e.*, on a silhouette manifold. However, due to the impossibility to perfectly preserve local distances while switching back to the high-dimensional image space, part of the smooth transition on the manifold might be corrupted with topologically different silhouettes. This might result in a sharp increase of the distance to the prior $r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_q^{\alpha_p})$ in a few intermediate prior views. In order to mitigate the impact of those rare (but possible) sharp discontinuities in the sequence of prior silhouettes, we propose to weight the distance $r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_q^{\alpha_p})$ by a factor $\left(1 - \frac{w_p}{\sum_{n=1}^{P} w_n}\right)$ that becomes small in case of a sharp increase (high gradient) of $r(\mathbf{b}_k^{\alpha_0}, \mathbf{b}_l^{\alpha_P}, \mathbf{b}_q^{\alpha_p})$, so as to favor the priors that reflect smooth transitions in the image space.

Bibliography

[1] A. E. Abdel-Hakim and A. A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1978–1983. IEEE, 2006.

[2] E. H. Adelson and J. R. Bergen. *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[3] S. Agarwal, Y. Furukawa, N. Snavely, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing Rome. *International Journal of Computer Vision (IJCV)*, (6):40–47, 2010.

[4] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112. IEEE, 2011.

[5] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(12):2037–2041, 2006.

[6] L. B. Alberti. *De pictura*. Major arts, 1435.

[7] J.-P. Antoine, D. Barachea, R. Cesar, and L. da Fontoura Costa. Shape characterization with the wavelet transform. *Signal Processing*, 62(3):265–290, 1997.

[8] L. Bagnato, P. Frossard, and P. Vandergheynst. Optical flow and depth from motion for omnidirectional images using a TV-L1 variational framework on graphs. In *International Conference on Image Processing (ICIP)*, pages 1469–1472. IEEE, 2009.

[9] J. Bai, Q. Song, O. Veksler, and X. Wu. Fast dynamic programming for labeling problems with ordering constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1728–1735. IEEE, 2012.

[10] S. Baker, T. Sim, and T. Kanade. A characterization of inherent stereo ambiguities. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 428–435. IEEE, 2001.

[11] S. Baker, T. Sim, and T. Kanade. When is the shape of a scene unique given its light-field: A fundamental theorem of 3D vision? *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(1):100–109, 2003.

[12] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *Transactions on Graphics (TOG)*, 29(4):87, 2010.

[13] B. Bartczak and R. Koch. Dense depth maps from low resolution time-of-flight depth and high resolution color views. In *Advances in Visual Computing*, pages 228–239. Springer, 2009.

[14] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[15] R. Bellman and R. Corporation. *Dynamic Programming*. Princeton University Press, 1957.

[16] A. C. Berg and J. Malik. Geometric blur for template matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 599–607. IEEE, 2001.

[17] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Computer graphics and Interactive Techniques (SIGGRAPH)*, pages 417–424. ACM, 2000.

[18] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: Patchmatch belief propagation for correspondence field estimation. *International Journal of Computer Vision (IJCV)*, pages 1–12, 2012.

[19] S. Bianco, F. Gasparini, and R. Schettini. Combining strategies for white balance. In *Electronic Imaging*, pages 652–660. International Society for Optics and Photonics, 2007.

[20] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision (IJCV)*, 33(3):181–200, 1999.

[21] A. Bodis-Szomoru, H. Riemenschneider, and L. Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 469–476. IEEE, 2014.

[22] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision (IJCV)*, 1(1):7–55, 1987.

[23] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):55–73, 1990.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001.

[25] J. Braux-Zin, R. Dupont, and A. Bartoli. A general dense image matching framework combining direct and feature-based costs. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 185–192. IEEE, 2013.

[26] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *Journal on Imaging Sciences*, 3(3):492–526, 2010.

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

[28] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 510–517. IEEE, 2005.

[29] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8):993–1008, 2003.

[30] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(7):1281–1298, 2012.

[31] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *Transactions on Graphics (TOG)*, 22(3):569–577, 2003.

[32] C. D. Castillo and D. W. Jacobs. Wide-baseline stereo for face recognition with large pose variation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 537–544. IEEE, 2011.

[33] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[34] G. Chartier. *Introduction to optics*. Springer, 2005.

[35] P. Collinet and C. a. Verleysen. Evaluation des fonctions motrices á l'aide d'une kinect. Master's thesis, Université catholique de Louvain, 2014.

[36] R. T. Collins. A space-sweep approach to true multi-image matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 358–363. IEEE, 1996.

[37] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619, 2002.

[38] K. R. Connor and I. D. Reid. Novel view specification and synthesis. In *British Machine Vision Conference (BMVC)*, pages 1–10, 2002.

[39] T. P. P. Council. Tpc-d - frequently asked questions (faq). `http://www.tpc.org/tpcd/faq.asp#anchor1140017`, 2013. [Online; accessed 16-July-2015].

[40] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding (CVIU)*, 63(3):542–567, 1996.

[41] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding (CVIU)*, 97(1):51–85, 2005.

[42] F. C. Crow. Summed-area tables for texture mapping. *Computer Graphics and Interactive Techniques (SIGGRAPH)*, 18(3):207–212, 1984.

[43] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *Transactions on Graphics (TOG)*, 27(3):98, 2008.

[44] R. C. De Amorim and B. Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering. *Pattern Recognition*, 45(3):1061–1075, 2013.

[45] J. de la Torre Lara. 3D visualization using virtual view generation for stereoscopic hardware. Master's thesis, Universitat Politècnica de Catalunya, 2011.

[46] B. Delaunay. Sur la sphère vide à la memoire de Georges Voronoï. *Bulletin de l'Académie des Sciences de l'URSS.*, 6:793–800, 1934.

[47] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision (IJCV)*, 96(1):1–27, 2012.

[48] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[49] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *International Conference on Image and Video Retrieval*, page 19. ACM, 2009.

[50] C. R. Dyer. Volumetric scene reconstruction from multiple views. In *Foundations of Image Understanding*, pages 469–489. Springer, 2001.

[51] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. De Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating textures. In *Computer Graphics Forum*, volume 27, pages 409–418, 2008.

[52] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[53] O. Faugeras. Stratification of three-dimensional vision: projective, affine, and metric representations. *Josa*, 12(3):465–484, 1995.

[54] O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Moll, G. Berry, J. Vuillemin, et al. Real time correlation-based stereo: algorithm, implementations and applications. Technical report, Inria, 1993.

[55] U. Fecker, M. Barkowsky, and A. Kaup. Improving the prediction efficiency for multi-view video coding using histogram matching. In *Picture Coding Symposium*, 2006.

[56] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision (IJCV)*, 70(1):41–54, 2006.

[57] A. V. Fiacco and G. P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM, 1990.

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

[59] A. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In *3D Structure from Multiple Images of Large-Scale Environments*, pages 155–170. Springer, 1998.

[60] A. Fitzgibbon, Y. Wexler, A. Zisserman, et al. Image-based rendering using image-based priors. In *International Conference on Computer Vision (ICCV)*, volume 3, pages 1176–1183. IEEE, 2003.

[61] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. arXiv:1506.06825, 2015.

[62] P.-E. Forssén. Maximally stable colour regions for recognition and matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[63] D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[64] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building rome on a cloudless day. In *European Conference on Computer Vision (ECCV)*, pages 368–381. Springer, 2010.

[65] J. Franco and E. Boyer. Efficient polyhedral modeling from silhouettes. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(3):414–427, 2009.

[66] F. Fraundorfer, K. Schindler, and H. Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image and Vision Computing*, 24(4):395–406, 2006.

[67] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, 1993.

[68] A. Fusiello, S. Caldrer, S. Ceglie, N. Mattern, and V. Murino. View synthesis from uncalibrated images using parallax. In *International Conference on Image Analysis and Processing*, pages 146–151. IEEE, 2003.

[69] A. Fusiello and L. Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[70] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[71] D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1418–1425. IEEE, 2010.

[72] M. Germann, A. Hornung, R. Keiser, R. Ziegler, S. Würmlin, and M. Gross. Articulated billboards for video-based rendering. *Computer Graphics Forum*, 29(2):585–594, 2010.

[73] M. Germann, T. Popa, R. Keiser, R. Ziegler, and M. Gross. Novel-view synthesis of outdoor sport events using an adaptive view-dependent geometry. In *Computer Graphics Forum*, volume 31, pages 325–333. Wiley Library, 2012.

[74] J.-M. Geusebroek, R. Van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(12):1338–1350, 2001.

[75] C. R. Giardina and F. P. Kuhl. Accuracy of curve approximation by harmonically related vectors with elliptical loci. *Computer Graphics and Image Processing (GCIP)*, 6(3):277–285, 1977.

[76] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 1–8. IEEE, 2007.

[77] B. Goldlücke and M. Magnor. Real-time free-viewpoint video rendering from volumetric geometry. In *Visual Communications and Image Processing (VCIP)*, pages 1152–1158. International Society for Optics and Photonics, 2003.

[78] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Prentice Hall Upper Saddle River, 2002.

[79] P. Goorts. *Real-time, Adaptive Plane Sweeping for Free Viewpoint Navigation in Soccer Scenes*. PhD thesis, Hasselt University, 2014.

[80] P. Goorts, C. Ancuti, M. Dumont, S. Rogmans, and P. Bekaert. Real-time video-based view interpolation of soccer events using depth-selective plane sweeping. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2013.

[81] P. Goorts, M. Dumont, S. Rogmans, and P. Bekaert. An end-to-end system for free viewpoint video for smooth camera transitions. In *International Conference on 3D Imaging*, pages 1–7. IEEE, 2012.

[82] P. Goorts, S. Maesen, M. Dumont, S. Rogmans, and P. Bekaert. Optimization of free viewpoint interpolation by applying adaptive depth plane distributions in plane sweeping-a histogram-based approach to a non-uniform plane distribution. 2013.

[83] C. Gramkow. On averaging rotations. *Journal of Mathematical Imaging and Vision*, 15(1):7–16, 2001.

[84] O. Grau, G. A. Thomas, A. Hilton, J. Kilner, and J. Starck. A robust free-viewpoint video system for sport scenes. In *International Conference on 3DTV*, pages 1–4. IEEE, 2007.

[85] M. Gross, S. Wuermlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. Vande Moere, and O. Staadt. Blue-c: A spatially immersive display and 3D video portal for telepresence. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM, 2003.

[86] J.-Y. Guillemaut and A. Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International Journal of Computer Vision (IJCV)*, 93(1):73–100, 2011.

[87] J.-Y. Guillemaut, J. Kilner, and A. Hilton. Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 809–816. IEEE, 2009.

[88] W. R. Hamilton. On a new species of imaginary quantities connected with a theory of quaternions. In *Proceedings of the Royal Irish Academy*, volume 2, pages 424–434, 1844.

[89] W. R. Hamilton. On quaternions or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13, 1844.

[90] A. J. Hanson. Visualizing quaternions. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, page 1. ACM, 2005.

[91] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[92] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[93] R. I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision (IJCV)*, 35(2):115–127, 1999.

[94] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 224–231. IEEE, 2009.

[95] A. Hilton, J.-Y. Guillemaut, J. Kilner, O. Grau, and G. Thomas. 3D-TV production from conventional cameras for sports broadcast. *Transactions on Broadcasting (TOB)*, 57(2):462–476, 2011.

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

[97] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[98] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(9):1582–1599, 2009.

[99] R. Horst and H. Tuy. *Global optimization: Deterministic approaches*. Springer, 1996.

[100] N. Inamoto and H. Saito. Arbitrary viewpoint observation for soccer match video. In *European Conference on Visual Media Production*, pages 21–30. Citeseer, 2004.

[101] N. Inamoto and H. Saito. Virtual viewpoint replay for a soccer match by view interpolation from multiple cameras. *Transactions on Multimedia (TOM)*, 9(6):1155–1166, 2007.

[102] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97(2):123–147, 2012.

[103] S. Ivekovic and E. Trucco. Dense wide-baseline disparities from conventional stereo for immersive videoconferencing. In *International Conference on Pattern Recognition (ICPR)*, volume 4, pages 921–924. IEEE, 2004.

[104] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Symposium on User Interface Software and Technology*, pages 559–568. ACM, 2011.

[105] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *European Conference on Computer Vision (ECCV)*, pages 228–241. Springer, 2004.

[106] T. Kanade, P. Narayanan, and P. W. Rander. Virtualized reality: Concepts and early results. In *Computer Vision Workshops (in conjunction with ICCV)*, volume 1, pages 69–76. IEEE, 1995.

[107] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *Transactions on Multimedia (TOM)*, 4(1):34–47, 1997.

[108] S. B. Kang. *A survey of image-based rendering techniques*. Digital, Cambridge Research Laboratory, 1997.

[109] S. B. Kang, Y. Li, X. Tong, and H.-Y. Shum. Image-based rendering. *Foundations and Trends in Computer Graphics and Vision*, 2(3):173–258, 2006.

[110] J. Kannala and S. S. Brandt. Quasi-dense wide baseline matching using match propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[111] R. Keys. Cubic convolution interpolation for digital image processing. *Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160, 1981.

[112] J. Kilner, J. Starck, A. Hilton, and O. Grau. Dual-mode deformable models for free-viewpoint video of sports events. In *International Conference on 3D Digital Imaging and Modeling*, pages 177–184. IEEE, 2007.

[113] I. Kitahara and Y. Ohta. Scalable 3D representation for 3D video display in a large-scale space. In *International Conference on Virtual Reality*, pages 45–52. IEEE, 2003.

[114] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 15–18. IEEE, 2006.

[115] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision (ECCV)*, pages 82–96. Springer, 2002.

[116] F. P. Kuhl and C. R. Giardina. Elliptic fourier features of a closed contour. *Computer Graphics and Image Processing (CGIP)*, 18(3):236–258, 1982.

[117] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[118] K. N. Kutulakos and S. M. Seitz. What do photographs tell us about 3D shape? Technical report, Technical Report TR692, Computer Science Dept., U. Rochester, 1998.

[119] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision (IJCV)*, 38(3):199–218, 2000.

[120] F. Lafarge and C. Mallet. Building large urban environments from unstructured point data. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 1068–1075. IEEE, 2011.

[121] F. Lafarge and C. Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision (IJCV)*, 99(1):69–85, 2012.

[122] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, volume 1, pages 282–289, 2001.

[123] A. Laurentini. The visual hull concept for silhouette-based image understanding. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(2):150–162, 1994.

[124] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems (NIPS)*, 16(3):329–336, 2004.

[125] N. D. Lawrence and J. Quiñonero-Candela. Local distance preservation in the gp-lvm through back constraints. In *International Conference on Machine Learning (ICML)*, pages 513–520. ACM, 2006.

[126] S.-B. Lee, K.-J. Oh, and Y.-S. Ho. Segment-based multi-view depth map estimation using belief propagation from dense multi-view video. In *International Conference on 3DTV*, pages 193–196. IEEE, 2008.

[127] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 31–42. ACM, 1996.

[128] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of graphics tools*, 8(2):1–15, 2003.

[129] M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(8):1140–1146, 2002.

[130] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(3):418–433, 2005.

[131] G. Li and S. W. Zucker. Surface geometric constraints for stereo in belief propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2355–2362. IEEE, 2006.

[132] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 125–131. IEEE, 1999.

[133] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics and Interactive Techniques (SIGGRAPH)*, 21(4):163–169, 1987.

[134] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157. IEEE, 1999.

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

[136] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding (CVIU)*, 64(2):193–229, 1996.

[137] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. California, USA, 1967.

[138] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2008.

[139] J. Mallon and P. F. Whelan. Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23(7):643–650, 2005.

[140] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[141] M. Matousek. *Epipolar Rectification Minimising Image Loss*. PhD thesis, Czech Technical University, 2007.

[142] T. Matsuyama and T. Takai. Generation, visualization, and editing of 3D video. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 234–245. IEEE, 2002.

[143] S. Mattoccia. A locally global approach to stereo correspondence. In *Computer Vision Workshops (in conjunction with ICCV)*, volume 1, pages 1763–1770. IEEE, 2009.

[144] S. Mattoccia. Accurate dense stereo by constraining local consistency on superpixels. In *International Conference on Pattern Recognition (ICPR)*, pages 1832–1835. IEEE, 2010.

[145] W. Matusik, C. Buehler, and L. McMillan. *Polyhedral visual hulls for real-time rendering*. Springer, 2001.

[146] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 369–374. ACM, 2000.

[147] W. Matusik and H. Pfister. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *Transactions on Graphics (TOG)*, volume 23, pages 814–824. ACM, 2004.

[148] N. F. Maunder and G. de Jager. *Virtual View Synthesis Using Visual Hulls*. PhD thesis, University of Cape Town, 2005.

[149] Max-Planck-Institut Informatik Independent Research Group 3. A synthetic test sequence for multi-view reconstruction and rendering research, 2005.

[150] C. S. McCamy, H. Marcus, and J. Davidson. A color-rendition chart. *Journal of Applied Photography Engineering*, 2(3):95–99, 1976.

[151] X. Mei, X. Sun, M. Zhou, H. Wang, X. Zhang, et al. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (in conjunction with ICCV)*, volume 1, pages 467–474. IEEE, 2011.

[152] B. Mičušík and J. Košecká. Multi-view superpixel stereo in urban environments. *International Journal of Computer Vision (IJCV)*, 89(1):106–119, 2010.

[153] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision (ECCV)*, pages 128–142. Springer, 2002.

[154] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005.

[155] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1-2):43–72, 2005.

[156] P. Monasse, J.-M. Morel, Z. Tang, et al. Three-step image rectification. In *British Machine Vision Conference (BMVC)*, pages 89–100, 2010.

[157] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *Journal on Imaging Sciences*, 2(2):438–469, 2009.

[158] M. Mozerov and J. van de Weijer. Accurate stereo matching by two-step energy minimization. *Transactions on Image Processing (TIP)*, 24(3), 2015.

[159] W. Murray and K.-M. Ng. An algorithm for nonlinear optimization problems with binary variables. *Computational Optimization and Applications*, 47(2):257–288, 2010.

[160] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer. A Survey of Urban Reconstruction. *Computer Graphics Forum*, 32, 2013.

[161] S. Narasimhe Gowda, C. Verleysen, and C. De Vleeschouwer. Virtual image reconstruction in a multi-view camera network. Master's thesis, Université catholique de Louvain, 2012.

[162] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[163] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report*, 2(11), 2005.

[164] S. Oesau, F. Lafarge, and P. Alliez. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, pages 1–15, 2015.

[165] J. Owens. *Television sports production*. Taylor & Francis, 2012.

[166] J. Park. Quaternion-based camera calibration and 3D scene reconstruction. *International Conference on Computer Graphics, Imaging and Visualisation (CGIV)*, pages 89–92, 2007.

[167] O. Pele and M. Werman. Improving perceptual color difference using basic color terms. arXiv:1211.5556, 2013.

[168] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(7):629–639, 1990.

[169] B. Petit, J.-D. Lesage, C. Menier, J. Allard, J.-S. Franco, B. Raffin, E. Boyer, and F. Faure. Multicamera real-time 3D modeling for telepresence and remote collaboration. *International Journal of Digital Multimedia Broadcasting*, 2010, 2009.

[170] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 496–501. IEEE, 1999.

[171] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision (IJCV)*, 78(2-3):143–167, 2008.

[172] V. A. Prisacariu and I. D. Reid. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 2185–2192. IEEE, 2011.

[173] G. Puy, P. Vandergheynst, R. Gribonval, and Y. Wiaux. Universal and efficient compressed sensing by spread spectrum and application to realistic fourier imaging techniques. *Journal on Advances in Signal Processing (EURASIP)*, 2012(1):1–13, 2012.

[174] M. Reynolds, J. Dobos, L. Peel, T. Weyrich, and G. J. Brostow. Capturing time-of-flight data with confidence. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 945–952. IEEE, 2011.

[175] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[176] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 2564–2571. IEEE, 2011.

[177] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2. IEEE, 1999.

[178] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 636–643. IEEE, 2001.

[179] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *European Conference on Computer Vision (ECCV)*, pages 414–431. Springer, 2002.

[180] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*, pages 31–42. Springer, 2014.

[181] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47(1-3):7–42, 2002.

[182] S. M. Seitz. *Image-based transformation of viewpoint and scene appearance*. PhD thesis, University of Wisconsin Madison, 1997.

[183] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 519–528. IEEE, 2006.

[184] S. M. Seitz and C. R. Dyer. Physically-valid view synthesis by image interpolation. In *Computer Vision Workshops (in conjunction with ICCV)*, volume 1, pages 18–25. IEEE, 1995.

[185] S. M. Seitz and C. R. Dyer. View morphing. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 21–30. ACM, 1996.

[186] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision (IJCV)*, 35(2):151–173, 1999.

[187] G. Sharma, W. Wu, and E. N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005.

[188] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, pages 124–127, 1950.

[189] K. Shoemake. Animating rotation with quaternion curves. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, volume 19, pages 245–254. ACM, 1985.

[190] H.-Y. Shum, S.-C. Chan, and S. B. Kang. *Image-based rendering*. Springer, 2008.

[191] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 1881–1888. IEEE, 2009.

[192] A. M. Siu and R. W. Lau. Image registration for image-based rendering. *Transactions on Image Processing (TIP)*, 14(2):241–252, 2005.

[193] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *Eurographics Conference on Volume Graphics*, pages 81–101. Eurographics Association, 2001.

[194] G. Slabaugh, R. Schafer, and M. Hans. Image-based photo hulls. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 704–862. IEEE, 2002.

[195] B. M. Smith, L. Zhang, and H. Jin. Stereo matching with nonparametric smoothness priors in feature space. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 485–492. IEEE, 2009.

[196] A. Smolic. 3D video and free viewpoint video-from capture to display. *Pattern recognition*, 44(9):1958–1968, 2011.

[197] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand. 3D video and free viewpoint video-technologies, applications and mpeg standards. In *International Conference on Multimedia and Expo (ICME)*, pages 2161–2164. IEEE, 2006.

[198] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. 27(3):15, 2008.

[199] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *Transactions on graphics (TOG)*, 25(3):835–846, 2006.

[200] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*, 80(2):189–210, 2008.

[201] J. Starck and A. Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications*, 27(3):21–31, 2007.

[202] T. Stich, C. Linz, G. Albuquerque, and M. Magnor. View and time interpolation in image space. *Computer Graphics Forum*, 27(7):1781–1787, 2008.

[203] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 552–560. IEEE, 2004.

[204] C. Strecha, T. Tuytelaars, and L. Van Gool. Dense matching of multiple wide-baseline views. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 1194–1201. IEEE, 2003.

[205] C. Strecha and L. Van Gool. PDE-based multi-view depth estimation. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 416–425. IEEE, 2002.

[206] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2008.

[207] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(7):787–800, 2003.

[208] L. Sun, Q. De Neyer, and C. De Vleeschouwer. Multimode spatiotemporal background modeling for complex scenes. In *European Signal Processing Conference (EUSIPCO)*, pages 165–169. IEEE, 2012.

[209] X. Sun, X. Mei, M. Zhou, H. Wang, et al. Stereo matching with reliable disparity propagation. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 132–139. IEEE, 2011.

[210] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image understanding*, 58(1):23–32, 1993.

[211] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov Random Fields with smoothness-based priors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(6):1068–1080, 2008.

[212] M. Tanimoto. FTV: Free-viewpoint television. *Signal Processing: Image Communication*, 27(6):555–570, 2012.

[213] G. Taubin. Curve and surface smoothing without shrinkage. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 852–857. IEEE, 1995.

[214] D. V. Taylor. System for producing time-independent virtual camera movement in motion pictures and other media, Dec. 18 2001. US Patent 6,331,871.

[215] G. A. Thomas. Real-time camera pose estimation for augmenting sports scenes. In *European Conference on Visual Media Production*, pages 10–19. IEEE, 2006.

[216] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[217] E. Tola. *DAISY: A Fast Descriptor for Dense Wide Baseline Stereo and Multiview Reconstruction*. PhD thesis, EPFL, 2010.

[218] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(5):815–830, 2010.

[219] E. Tola, C. Zhang, Q. Cai, and Z. Zhang. Virtual view generation with a hybrid camera array. *CVLAB-Report-2009-001 (École Polytechnique Fédérale de Lausanne, 2009)*, 2009.

[220] A. Toshev, J. Shi, and K. Daniilidis. Image matching via saliency region correspondences. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[221] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer. Dense segmentation-aware descriptors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 2890–2897. IEEE, 2013.

[222] T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Visual Information and Information Systems*, pages 493–500. Springer, 1999.

[223] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference (BMVC)*, volume 412, 2000.

[224] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision (IJCV)*, 59(1):61–85, 2004.

[225] A. Van Den Hengel, A. R. Dick, T. Thormählen, B. Ward, and P. H. Torr. Building models of regular scenes from structure and motion. In *British Machine Vision Conference (BMVC)*, pages 197–206, 2006.

[226] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 547–556. IEEE, 2003.

[227] M. Vergauwen and L. Van Gool. Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(6):411–426, 2006.

[228] M. Vergauwen, F. Verbiest, V. Ferrari, C. Strecha, and L. Van Gool. Wide-baseline 3D reconstruction from digital stills. *International Workshop on Visualization and Animation of Reality-based 3D Models*, 2003.

[229] C. Verleysen and C. De Vleeschouwer. Learning and propagation of dominant colors for fast video segmentation. In *Advanced Concepts for Intelligent Vision Systems*, pages 657–668. Springer, 2013.

[230] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1311–1318. IEEE, 2009.

[231] Z. Wang, F. Wu, and Z. Hu. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 42(5):941–953, 2009.

[232] Z.-F. Wang and Z.-G. Zheng. A region based stereo matching algorithm using cooperative optimization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2008.

[233] S. Wanner and B. Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *Pattern Recognition*, pages 1–10. Springer, 2013.

[234] T. Werner, R. D. Hersch, and V. Hlavac. Rendering real-world objects using view interpolation. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 957–962. IEEE, 1995.

[235] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *European Conference on Computer Vision (ECCV)*, pages 541–555. Springer, 2002.

[236] C. Wheatstone. Contributions to the physiology of vision on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, pages 371–394, 1838.

[237] S. A. Winder and M. Brown. Learning local image descriptors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–8. IEEE, 2007.

[238] S. A. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 178–185. IEEE, 2009.

[239] O. J. Woodford, I. D. Reid, P. H. Torr, and A. Fitzgibbon. On new view synthesis using multiview stereo. In *British Machine Vision Conference (BMVC)*, pages 1–10, 2007.

[240] O. J. Woodford, P. H. Torr, I. D. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(12):2115–2128, 2009.

[241] S. Yaguchi and H. Saito. Arbitrary viewpoint video synthesis from multiple uncalibrated cameras. *Transactions on Systems, Man, and Cybernetics*, 34(1):430–439, 2004.

[242] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 204–211. IEEE, 2003.

[243] R. Yang, M. Pollefeys, H. Yang, and G. Welch. A unified approach to real-time, multi-resolution, multi-baseline 2D view synthesis and 3D depth estimation using commodity graphics hardware. *International Journal of Image and Graphics*, 4(04):627–651, 2004.

[244] J. Yao and W.-K. Cham. 3D modeling and rendering from multiple wide-baseline images by match propagation. *Signal Processing: Image Communication*, 21(6):506–518, 2006.

[245] G. Yu and J.-M. Morel. ASIFT: an algorithm for fully affine invariant comparison. *Image Processing*, 2:438–469, 2011.

[246] A. L. Yuille and T. Poggio. *A generalized ordering constraint for stereo correspondence*. PhD thesis, MIT, Cambridge, A.I. Laboratory Memo 777, Mass., 1984.

[247] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l 1 range image integration. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 1–8. IEEE, 2007.

[248] Q. Zhang and K. N. Ngan. Dense stereo matching from separated views of wide-baseline images. In *Advanced Concepts for Intelligent Vision Systems*, pages 255–266. Springer, 2010.

[249] F. Zilly, C. Riechert, M. Muller, and P. Kauff. Generation of multi-view video plus depth content using mixed narrow and wide-baseline setup. In *International Conference on 3DTV*, pages 1–4. IEEE, 2012.

[250] S. Zinger, L. Do, and P. de With. Free-viewpoint depth image based rendering. *Journal of visual communication and image representation*, 21(5):533–541, 2010.

[251] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision (IJCV)*, 75(1):49–65, 2007.

[252] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. 23:600–608, 2004.