Multi-view Shape Modeling from Images: Contributions to Photometric-based Reconstruction using Deformable Meshes

Amaël Delaunoy

To cite this version:

Amaël Delaunoy. Multi-view Shape Modeling from Images: Contributions to Photometric-based Reconstruction using Deformable Meshes. Computer Vision and Pattern Recognition [cs.CV]. Université de Grenoble, 2011. English. NNT : . tel-00654188v1

HAL Id: tel-00654188
https://theses.hal.science/tel-00654188v1

Submitted on 21 Dec 2011 (v1), last revised 10 Jul 2012 (v3)

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE
Spécialité : **Mathématiques, Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Amaël Delaunoy

Thèse dirigée par **Peter Sturm**
et codirigée par **Emmanuel Prados**

préparée au sein de l' **Inria Grenoble, Laboratoire Jean Kuntzmann**
et de l'école doctorale **Mathématiques, Sciences et Technologies de l'Information, Informatique**

Multi-view Shape Modeling from Images

Contributions to Photometric-based Reconstruction using Deformable Meshes

Thèse soutenue publiquement le **2 Décembre 2011**,
devant le jury composé de :

Prof. Augustin Lux
Professeur, Université de Grenoble, Président
Prof. Marc Pollefeys
Professeur, ETH Zürich, Rapporteur
Prof. Jean Ponce
Professeur, École Normale Supérieure, Rapporteur
Prof. Daniel Cremers
Professeur, Technische Universität München, Examinateur
Dr. Peter Sturm
Directeur de recherche, Inria Grenoble, Directeur de thèse
Dr. Emmanuel Prados
Chargé de recherche, Inria Grenoble, Co-Directeur de thèse

Abstract

Understanding, analyzing and modeling the 3D world from 2D pictures and videos is probably one of the most exciting and challenging problem of computer vision. In this thesis, we address several geometric and photometric aspects to 3D surface reconstruction from multi-view calibrated images.

We first formulate multi-view shape reconstruction as an inverse rendering problem. Using generative models, we formulate the problem as an energy minimization method that leads to the non-linear surface optimization of a deformable mesh. A particular attention is addressed to the computation of the discrete gradient flow, which leads to coherent vertices displacements. We particularly focus on models and energy functionals that depend on visibility and photometry. The same framework can then be equally used to perform multi-view stereo, multi-view shape from shading or multi-view photometric stereo.

Then, we propose to exploit different additional information to constraint the problem in the non-Lambertian case, where the appearance of the scene depends on the view-point direction. Segmentation for instance can be used to segment surface regions sharing similar appearance or reflectance. Helmholtz reciprocity can also be applied to reconstruct 3D shapes of objects of any arbitrary reflectance properties. By taking multiple image-light pairs around an object, multi-view Helmholtz stereo can be performed. Using this constraint acquisition scenario and our deformable mesh framework, it is possible to reconstruct high quality 3D models.

Keywords: Computer Vision, Multi-view Shape Reconstruction, Deformable Meshes, Photometry, Visibility, Shape from Shading, Variational Methods, Multi-view Stereovision.

Résumé

Comprendre, analyser et modéliser l'environement 3D à partir d'images provenant de caméras et d'appareils photos est l'un des défis majeurs actuels de recherche en vision par ordinateur. Cette thèse s'intéresse à plusieurs aspects géométriques et photometriques liés à la reconstruction de surfaces à partir de plusieurs caméras calibrées.

La reconstruction 3D est vue comme le problème inverse de rendu graphique, et vise à minimiser une fonctionnelle d'énergie afin d'optimiser un maillage triangulaire représentant la surface à reconstruire. L'énergie est définie via un modèle génératif faisant naturellement apparaître des attributs tels que la visibilité ou la photométrie. Ainsi, l'approche présentée peut indifférement s'adapter à divers cas d'applications tels que la stéréovision multi-vues, la stéréo photométrique multi-vues ou encore le *shape from shading* multi-vues.

Plusieurs approches sont proposées afin de résoudre les problèmes de correspondances pour des scènes non Lambertiennes, dont l'apparence varie en fonction du point de vue. La segmentation, la stéréo photométrique ou encore la réciprocité d'Helmholtz sont des éléments étudiés afin de contraindre les algorithmes de reconstruction. L'exploitation de ces contraintes dans le cadre de reconstruction multi-vues permet de reconstruire des modèles complets 3D avec une meilleure qualité.

Keywords: Vision par ordinateur, Reconstruction 3D de formes, Maillages déformables, Photométrie, Visibilité, Shape from Shading, Méthodes variationnelles, Stéréovision multi-vues.

Acknowledgements

Je tiens tout d'abord à remercier Augustin Lux, Jean Ponce, Marc Pollefeys et Daniel Cremers d'avoir accepté d'être dans mon jury de thèse et d'avoir fait le déplacement jusqu'à Grenoble. Merci à Jean et Marc d'avoir été les relecteurs de ma thèse, et Augustin et Daniel d'en avoir été les examinateurs. Merci à vous quatre pour l'effort réalisé et vos compliments ainsi que pour vos commentaires sur mon travail.

Ensuite, un grand merci à Peter et Manu d'avoir été d'excellents encadrants tout au long de ces années passées à l'Inria. Merci de m'avoir guidé dans le métier de chercheur. Vous m'avez donné la possibilité de découvrir le monde fabuleux de la recherche en vision par ordinateur, et vous avez toujours été présents quand j'en avais besoin. Merci à tous les deux pour votre disponibilité, votre optimisme et vos excellents conseils. Bonne réussite avec Steep et dans vos vies familiales!

Un grand merci à tous mes collègues de bureau de l'Inria, pour tous les échanges que nous avons eu ensemble, scientifiques ou non. Merci donc à Pau Gargallo, Daniel Weinland, Clément Ménier, Kuk-Jin Yoon, Jamil Draréni, Visesh Chari, Simone Gasparini, Régis Perrier ainsi que tous les membres de Perception et Steep, permanents, stagiaires, doctorants ou ingénieurs que j'ai la chance de côtoyer durant ces années. Que de bons souvenirs pendant les pauses cafés, parties de Volley, les divers pots et bien sûr les BBQ!

Merci également à Stephen Aylward et Julien Jomier pour m'avoir ouvert les yeux au monde de la recherche lors de mon passage à UNC Chapel Hill quelques années en arrière. Merci à CPE-Lyon de m'avoir permis de faire ce stage, et merci à Catherine Mennessier et Jean-Marie Becker d'avoir été d'excellents enseignants durant mes années à Lyon.

Merci à Shree Nayar et Peter Belhumeur de m'avoir donné la possibilité de passer quelques temps à New York pendant ma thèse. J'y ai beaucoup appris et apprécié l'expérience passée là-bas. Merci à Hiroshi Kawasaki de m'avoir donné la possibilité de découvrir le Japon.

Je n'oublie pas tous mes amis de Lyon et de Grenoble. Merci pour tous les bons moments passés durant mes études. Bon courage et bonne continuation à tous. Un grand merci particulier à toute la troupe des cakes au beurre, Régis Perrier, Antoine Letouzey, Benjamin Petit, Lamiae Azizi, Gaëtan Jansens, Michel Amat, Xavi Alameda, Kiran Varanasi, Pierre Landes et j'en oublie surement plusieurs, pour les bons moments passés, les sorties ciné, escalade, ski, rando, ski de randonnée, bières du mercredi, pains au chocolat et tant d'autres! Antoine, merci pour les quiches et cette super colocation, et Régis, merci d'avoir été un excellent compagnon de voyage pendant ces conférences!

Enfin, un grand merci à toute ma famille, et particulièrement mes parents Michel et Cécile et mes frères et soeur Thibaud, Maïté et Gabriel, sans qui je ne serais pas là aujourd’hui. Merci d’être venus à ma soutenance et d’avoir toujours été présents!

Bonne continuation à tous!

Contents

CHAPTER I

Multi-view Shape Reconstruction: Background and Introduction

In this chapter we introduce the problem of multi-view shape reconstruction from images in computer vision. Inferring the 3D shape of objects given a set of several pictures of that object is an ill-posed problem. It aims at solving the inverse problem of the image rendering process. Therefore, it is necessary to understand this process in order to be able to recover the 3D model. Image formation is a complex process that includes projective geometry, scene geometry, appearance, materials, lighting and other complex photometric phenomena. In order to constraint the problem, assumptions and simplified models are required. This chapter overviews such simplifications. We will also present standard approaches to stereo reconstruction, in particular in the multi-view setting where cameras are placed all around the object of interest. We will see how generative models can lead to generic formulations for multi-view reconstruction. We also detail the organization of the thesis and expose the contributions of our work.

Contents

I.1 The Shape from Images Reconstruction Problem

Nowadays, who does not hold a camera? Being in your phone, in your laptop, as a personal camera or a video camera, digital cameras are everywhere and in everyone's hand. This has contributed making computer vision a fast growing field of research, with new perspectives and research opportunities. Among this field, 3D reconstruction has called the attention of many researchers in computer vision, and particularly during the last two decades. This research aims at understanding and/or capturing the tri-dimensional information of a scene thanks to two-dimensional images taken from (digital) cameras, by exploiting the redundant information among those images.

I.1.1 Overview

Motivation

3D models are often needed in a large variety of applications and domains. The recent number of 3D movies and interactive video games using 3D information such as the Nintendo Wii or Microsoft Kinect are good examples of recent interest towards 3D models. Whereas those general public applications are mainly focus in the multimedia and entertainment domain, 3D information is important and sometimes critical in a lot of different fields such as industrial control, robotics, medical imaging, space industry or military applications.

Automatically reconstructing such 3D models is very difficult and challenging. It first requires acquisition devices that can extract a 3D information of the scene. Then further modeling, computation and algorithms are usually required. For a long time, Laser scans (or similar techniques) have been extensively used, for example to produce 3D models in computer graphics. Usually associated with geometry post-processing (such as recovering smooth polygonal surfaces from point clouds), they provided nice results that are still used nowadays as references [1] or ground truth data [136, 145].

On the other hand, cameras provide more information such as *color* and *textures*. Used in stereo pairs or in multi-view settings with correct geometry modeling, they provide useful information. This can be used to reconstruct shapes in environments where active systems would not work, such as the reconstruction of large scale objects or environments [49, 154]. Moreover, used in appropriate manners, cameras can deal with objects of more diverse materials such as particular surface geometry texture, reflectance, transparency or refraction. Cameras are more flexible and easier to use, and nowadays even outperforms Laser scans for normal-sized objects in terms of resolution [105]. Even though multi-view 3D reconstruction has been extensively used and developed recently (See Seitz et al. [136, 145] for comparisons of recent techniques, as well as the Chapter 11 and 12 of [146]), there still remains open doors and challenges.

Problem Description

Stereo reconstruction techniques often refer to binocular stereo reconstruction, where a scene is observed by a pair of two cameras, originally inspired by biological vision and perception. In general correspondences between the left and right images allow to compute what is called a disparity map or a depth map of the scene. The baseline between the two cameras is usually small and the system only allow to recover partial 3D information due to the limited visible field of view. In order to get a complete overview of the geometry, multiple views are required. Extending binocular stereo to multi-view stereo is not straight forward. This has lead to many specific research developments those last decades. We can for instance refer to the multi-view geometry and 3D reconstruction parts of the following books: Faugeras and Luong [43], Forsyth and Ponce [45], Hartley and Zisserman [63] or Szeliski [146]. Those books give a good overview of the field as well as good pointers to related publications.

Figure I.1: Overview of multi-view shape from images: On the left: input images taken from the GRImage platform at Inria, that allows to acquire up to 32 synchronized and calibrated video data. On the middle we show the calibrated pictures in the $3D$ space at time t. On the right is an expected output: a tri-dimensional model representing the scene or object of interest obtained with [31].

Multi-view shape from images is the process of reconstructing a tri-dimensional scene given a set of images or pictures of that scene. The output of that process is a tri-dimensional structure such as points or surfaces. Figure I.1 illustrates an example of acquisition system used at Inria, along with the input of the algorithm (calibrated images) and one expected result (a 3D triangle mesh). Those images first need to be calibrated, meaning that all camera positions and parameters are known, and that there is a link between the 3D world and the 2D images. In the rest of this thesis, we will consider the camera positions to be known in advance, that is the system is geometrically calibrated. We focus on the description and modeling of the 3D reconstruction part only.

Then to achieve the reconstruction, many cues on images can be used, such as motion, shading, occluding boundaries or texture information. Where acquisition techniques such as Laser scans provide (up to calibration and noise) direct 3D measurements, multi-view stereo rely on image processing and correspondences. The problem becomes finding dense correspondences between the pixels in the different images. Then two pixels in two images that corresponds to the same 3D location can be used to retrieve that point using triangulation, assuming that the camera positions are known. However finding a dense matching between images is difficult because of occlusion problems as well as matching ambiguities and change of appearance between the views. The difference in the appearance may come from as diverse photometric phenomena as reflection, refraction or diffusion among others. Because of the complexity of the image rendering process, new techniques have been developed and are still being improved. The large disparity in image appearance and representations makes 3D reconstruction a very hard and challenging problem. This creates needs to fully understand and model images in order to be able to process information for building 3D models. Because the physical models of cameras are too complex to use, one needs further assumptions.

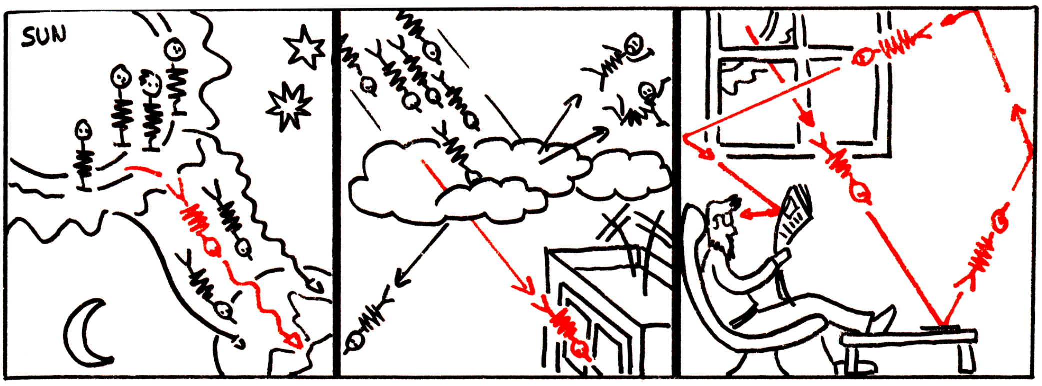

Figure I.2: Life of a photon, courtesy of Pau Gargallo [55]. This figure illustrates the complexity of the light energy received at a sensor: the eye in this case, the camera in computer vision. Light is emitted by the sun or other light sources, then diffused, scattered, refracted and reflected on different media and objects before finally reaching a sensor.

I.1.2 Geometric and Photometric Assumptions

Because 3D reconstruction from 2D images is the inverse problem of the image formation process, it is necessary to understand that step. Digital camera create images of the world by capturing for each pixel the incoming set of light rays coming to it through an optical device. An image can be captured out of this world, but is just a 2D projection (a two dimensional signal) of that 3D information. This involves complex photometric phenomena due to scene geometry, materials, optical device and lighting conditions. In this thesis we define an image as the result of following components:

- The acquisition device: its optical properties (along with projective geometry) and the properties of the sensor itself (photometric and geometric aspects).
- The illumination: light sources, positions and intensity.
- The scene or object of interest: its appearance (object materials, object reflection/-refraction) and its geometry (3D shape, self occlusions and visibility).

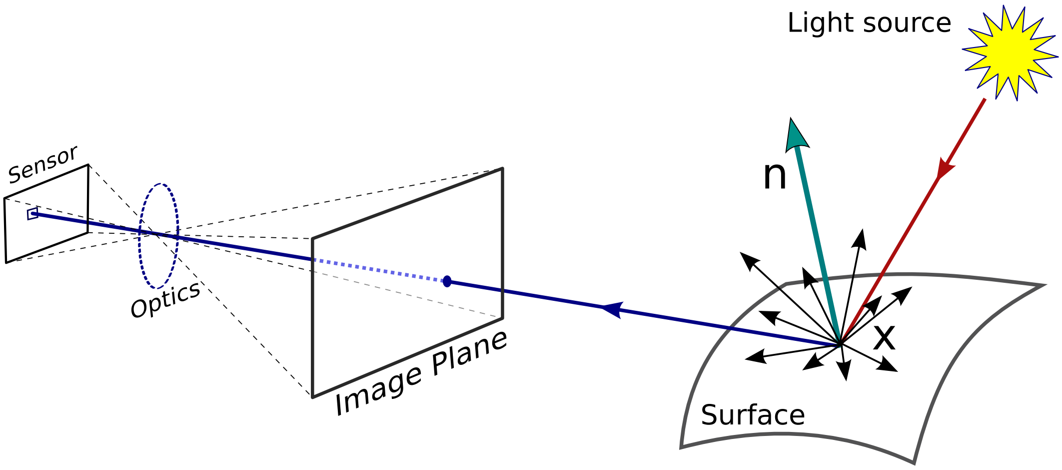

Figure I.3: Simplified image formation process.

The Figure I.3 shows a simple example of the generation of images. In the following, we discuss elements and information that are going to be useful in the context of this thesis. For a more complete information, we refer the readers to [41, 45, 63, 146] and all the references therein. Those books give a good overview of the computer vision research field [45, 146] and its associated multi-view geometry in particular [43, 63].

I.1.2.1 Multi-view Geometry

Camera models and their associated projective geometry are fundamental in computer vision, since it links the 3D world with its 2D projection in the images. Given a set of images, we have the following relationship between a 3D point $\mathbf{x}$ of a scene and its associated 2D projected pixel $\mathbf{p}$:

$$\mathbf{p} = \Pi_i(\mathbf{x}) \; , \tag{I.1}$$

where $\Pi_i : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ corresponds to the projection model of a camera i. Then calibrating a camera i corresponds to find its function Π_i. If this model can be parametrized, then the calibration consists in estimating the appropriate parameters of the camera model. The camera position and orientation are called the *extrinsic parameters*. The *intrinsic parameters* depend on the camera device itself, and correspond to the projection model parameters as well as the radiometric response of the camera (For a pinhole camera model, the geometric intrinsic parameters are typically the focal length, principal point, sensor size and resolution, or lens distortions). Thanks to recent work those last decades [63], we have good and accurate calibration. In the rest of this thesis we will consider those parameters to be known in advance, and we will focus on the reconstruction part only.

In 3D reconstruction, it is often assumed that a camera can be modeled using the pinhole camera model. While this assumption is not true using real world data, in practice the pinhole camera model is a good approximation and a simple model to use. We will see in the following that recent 3D reconstruction techniques give reasonable 3D models using that assumption. From now on, we will only consider parametrized camera models such as the pinhole model or push-broom cameras

Scene Geometry

For reconstruction, the scene geometry is represented by a shape model, that can have different forms [87]. This model is really important since it conditions the underlined algorithms used for reconstruction. Even though reconstruction methods can sometimes be applied to a large variety of shapes, in general (numerical) algorithms are adaptive to particular subset of shapes. In multi-view reconstruction, it is often represented as one of the following shapes:

- Point clouds or set of oriented patches. [48, 95]
- Depth map or Height maps. [54]
- Model-based representations. [16, 151]
- Implicit models - for example Level Sets or occupancy grids. [42, 82, 91, 130, 141, 160]
- Explicit surfaces such as meshes. [9, 31, 36, 154]

In the following, we will consider the shape to be a closed surface. It is a natural and intuitive way of representing the shape of the object of interest to reconstruct. Moreover, the visibility is clearly defined using a surface, whereas it is not clear how to define it on point clouds or occupancy grids. We will also make the following assumption to the kind of scene we can reconstruct: we will consider that the scene to be reconstructed can be modeled by a piecewise planar surface. As a result, gas, fluids or objects that could not be represented as closed surfaces won't be addressed in this thesis. Let $\mathcal{S}$ be a $2D$ surface embedded into $\mathbb{R}^3$, and $\mathrm{d}s$ be the element of area of the surface.

I.1.2.2 Image Appearance

Bidirectional Reflectance Distribution Function

Here we describe the image formation process that is used by cameras to generate images. The appearance of a pixel is proportional to the amount of light received at the sensor, called the *irradiance* and measured in $[W.m^{-2}]$. The irradiance is the total incoming light reaching a surface patch. The *radiance* is the amount of light emitted by a surface patch in a particular direction. It is expressed as the power of light by unit area, by solid angle $[W.m^{-2}.sr^{-1}]$. The image irradiance is actually proportional to the scene radiance.

The scene radiance depends on many factors which are mainly due to the illumination, the reflectance of the scene and the position from which it is viewed. The reflectance is the function that describes the surface material properties. If one consider that the reflectance does not change over time, does not depend on the wavelength and does not have any scattering effects, then the reflectance is called the *bidirectional reflectance distribution function*

(BRDF). The BRDF indicates how bright a surface point appears from a particular direction, when illuminated from an other direction. It is the ratio between the incoming irradiance and the outgoing radiance. Let us define $\beta(\hat{\mathbf{v}}_i, \hat{\mathbf{v}}_o)$ to be the surface's BRDF, where $\hat{\mathbf{v}}_i$ parametrizes the incoming light direction and $\hat{\mathbf{v}}_o$ the outgoing (view) direction. One particularity of the BRDF is that it is *reciprocal*, meaning that the BRDF is the same if the input direction and outgoing one are switched and that: $\beta(\hat{\mathbf{v}}_i, \hat{\mathbf{v}}_o) = \beta(\hat{\mathbf{v}}_o, \hat{\mathbf{v}}_i)$. This property is known as *Helmholtz reciprocity*, and we will see later in Chapter V that it can be used as a cue for multi-view reconstruction.

The BRDF is very difficult and complex to acquire [156], even if the object's shape is known. Indeed, one need simplifications to be able to mix it with shape reconstruction. A lot of simplified models and approximations have been proposed in computer graphics to render images, see for instance Ngan [121] for a recent survey on the subject. Those models rely either on physics models [26, 148] or empirical models [122, 127]. In general, BRDF models are defined using a dichromatic model composed of a *diffuse* term and a *specular* term. Depending on the kind of materials one wants to model, those diffuse and specular term may have different formulations. For example, the *Lambertian* model usually describes perfectly smooth and diffuse surfaces, whose appearance is independent on the view-point direction. If the surface has micro-structures, then diffuse surfaces are better explained using the Oren and Nayar reflectance model [122], whereas specular models can be modeled using the Cook and Torrance reflectance model [26]. Unfortunately, there does not exist a single reflectance model that encompasses all kind of object materials.

Illumination Models

Even though the most common illumination models used in 3D reconstruction are point light sources or directional light, there exist many different ways to represent the illumination. A classical approach to model low frequency illumination is to use spherical harmonics. To control the illumination, projectors are an easy way to go, since the can be modeled using camera geometry and then calibrated similarly (a projector model is homologous to a camera).

Estimating the illumination is very complex. Indeed, most 3D reconstruction works assume simple light sources or even consider the scene to be Lambertian so that modeling the illumination is not required (See next paragraphs).

Shape and Reflectance

Estimating both the shape and reflectance of objects is challenging due to different reasons. The two main reasons are that the model of the reflectance has to be known in advance, and that the reflectance estimation is an under-constrained problem due to the number of parameters to estimate compared to the number of observations. A lot of effort has been done in that direction, showing promising techniques and results, see for example [9, 115, 133, 163]. Most of those methods rely on alternated optimization between the shape of the scene $\mathcal{S}$ and the parameters of the associated reflectance model $\mathbf{R}$. They also assume that the illumination conditions are known and that the class of objects to reconstruct fall in the chosen reflectance model.

The Constant Brightness Assumption

In computer vision applications, it is usually easier to consider that the appearance of an object is independent on the view-point direction. In that case the reflectance model is Lambertian and corresponds to the surface's *albedo*. It can usually be represented as a cosine law such that: $I = \rho\, \mathbf{n} \cdot \mathrm{L}$, where I is the intensity value, ρ the surface albedo, and L the light vector.

In the real world, very few objects are truly Lambertian. However, some objects are close to Lambertian, and some other can be considered as Lambertian under ambient illumination. If one assumes a Lambertian world and if the illumination is fixed when several images from different view-points are taken, then the constant brightness assumption is valid. Furthermore, the intensities values should be the same across images I_i and then modeling the illumination is unnecessary. This makes Lambertian approaches very popular since estimating the illumination is a rather difficult task. In that case, matching between views can be exploited to use stereo, structure-from-motion and 3D reconstruction algorithms. The Lambertian assumption is used in most of photometric multi-view stereo reconstruction techniques, though a lot of techniques use robust matching scores for more flexibility in appearance variations.

I.2 Reconstruction Techniques

This section addresses how the multi-view shape from images problem can be solved given the assumptions described previously. While there is a full literature on that subject, we focus our attention in techniques that automatically recover full 3D models. We first review direct approaches, that extract information in the images in 2D, and use it to retrieve 3D information. Then we describe methods that see 3D reconstruction as an inverse problem. In particular, we use the Bayesian inference by describing how images are generated and how this process can be inverted to infer the unknown geometry of the scene. This way of reasoning, previously described in [112] and applied for multi-view stereo in [55], allows to rigorously model the shape from images reconstruction problem. This is usually formulated as a non-linear optimization problem. We will see this can be solved by minimizing an energy functional explaining the scene.

I.2.1 Rapid Overview of Existing Approaches

There exist many different methods for reconstructing shapes from 2D images that have been extensively developed during the last decades (See for instance Chapter 12 of [146]). The differences between these approaches rely on the assumptions they make of the scene (either on its geometry, illumination or image formation model), on the choice of the shape representation, on their ability to deal with visibility, on their precision or applicative goal (visualization, interaction, scientific computing). In this section, we briefly describe some of those existing approaches.

Many methods extract 3D information directly from 2D images, for example by matching points between the different views or by merging depth maps obtained thanks to classical binocular stereo algorithms between image pairs. Binocular stereo algorithms have been widely used [134] to extract such information. However their extensions to multi-view settings are not straightforward and have been extensively researched so far. We refer for example to [136, 145] for comparisons of recent techniques, which show the recent interest in multi-view stereo techniques. The first category of algorithms use direct information from the images, in order to estimate 3D points, 3D patches, lines, or other features. If points are robustly matched across images, 3D model can be obtained and one can fit a surface to the reconstructed points. Many multi-view stereo approaches follow that principle and use it as an initialization for more complex algorithm by expanding it to either dense 3D reconstruction or dense stereo matching. If a large amount of images is available, robust algorithms can be applied to keep the best candidates for matching. This allows to initialize more complex algorithms that can reconstruct dense 3D models with large photo collections. Other works consider particular scenarios in order to make the matching easier, by considering specific acquisition setups (using for instance known illumination, changing illumination, etc.), or imposing constraints on the scene (known materials, shape symmetry, etc.). We detail some of those approaches in the next section.

Those previous methods can be described as direct methods, as opposed as to methods that tend to estimate the shape directly or optimize it. One typical approach is to represent the scene as a 3D volume, and then remove points that are not photo-consistent. This can be done by using voxel coloring [137] or space carving algorithms [53, 94]. One major issue in that case is to carefully account for visibility. This photo-consistency, as well as many other cues that will be developed in the next sections, can be used to define a quality measure of a reconstructed model. This model can be optimize by minimizing this error measure.

Energy Minimization Methods

Variational methods are commonly used in computer vision and computer graphics to compute, improve and process surface interfaces. Such approaches consist in defining an energy whose minimum is reached by the surface of the object of interest. The idea is to start from an initial shape and optimize it to decrease the energy until it reaches its minimum. It is usually performed through gradient descent techniques. This generic approach can be equally applied to different kind of problems and has been for example successfully applied in image segmentation [29]. During the last decades, this framework has been widely used in 3D reconstruction problems as well to optimize surfaces, see for example [42, 82, 130, 141, 160].

In general, the total energy of the shape $\mathcal{S}$ (typically a $2D$ surface embedded into $\mathbb{R}^3$) can be defined as a sum of a regularization term and a data term (the next section describes how to obtain those terms using the Bayesian approach). This energy functional quantifies the quality of the geometric model to optimize with respect to the observed data. The regularization corresponds to the prior probability and describes for instance the elasticity and rigidity of the model. This is typically a smoothness term toward some prior. The data term corresponds to the likelihood energy term, and measures the model's fidelity compared to the input data. It quantifies how close the current shape is from the observed images.

Let $\mathbf{I} = \{I_1, I_2, \ldots, I_m\}$ be the set of the observed images. If the problem is well-defined, then the solution of the problem is the surface $\hat{\mathcal{S}}$ that minimizes the following equation:

$$\hat{\mathcal{S}} = \arg\min_{\mathcal{S}} \left\{ \underbrace{E_{data}(\mathbf{I}, \mathcal{S})}_{Data\,term} \quad + \quad \underbrace{E_{reg}(\mathcal{S})}_{Regularization\,term} \right\} . \tag{I.2}$$

This energy minimization problem can be more generally described as minimizing the following energy functional:

$$E(\mathcal{S}) = \int_{\Omega} f(\mathbf{x}) \,\mathrm{d}\mathbf{x} \; + \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}_{\mathcal{S}}) \,\mathrm{d}s \, , \tag{I.3}$$

where $g : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^+$ is a weighted area functional over the surface $\mathcal{S}$ and $f : \mathbb{R}^3 \to \mathbb{R}^+$ is a volume potential (sometimes called ballooning term) inside the shape $\mathcal{S}$, where here Ω represents the interior of the surface $\mathcal{S}$. In multi-view shape reconstruction, $g(\mathbf{x}, \mathbf{n}_{\mathcal{S}})$ is typically a photo-consistency metric at the surface point $\mathbf{x}$ with its associated normal $\mathbf{n}_{\mathcal{S}}$.

Minimizing this kind of energy functional can be done using different techniques, depending on the concerned application and on the shape representation one uses. It is usually minimized through gradient descent techniques and therefore requires the computation of the corresponding gradient [141]. Then, starting from an initial surface $\mathcal{S}^0$, the surface optimization scheme follows the gradient descent flow:

$$\begin{cases} \mathcal{S}(0) = \mathcal{S}^0 \, , \\ \dfrac{\partial \mathcal{S}(t)}{\partial t} = -\nabla_M E(\mathcal{S}(t)) \, , \end{cases} \tag{I.4}$$

where $\nabla_M E(\mathcal{S})$ is the gradient that depends on a metric defined on a Riemannian manifold M such that $\mathcal{S} \in M$. Changing the metric of the gradient may result in more coherent gradient descent flow [24, 174]. Many tools are available to implement surface evolution, such as for example Level Sets, Graph cuts, convex relaxation or deformable meshes. Those techniques will be detailed in Section I.4.

One problem of this kind of energy is that the empty set is an obvious global minima. Moreover, minimizing Equation (I.3) usually introduces bias toward small surfaces that are usually compensated by adding additional terms. This classical problem is known as the *minimal surface bias* [5], and can be reduced in multi-view reconstruction by considering a Bayesian approach of the problem in order to properly define the energy functional. More details are available in Gargallo et al. [56] and in Section I.2.2 that follows.

I.2.2 Bayesian Formulation and Generative Models

Here, we describe the Bayesian rationale to multi-view reconstruction. This section is a summary of several works already using this formulation [55, 163] on which the following of this thesis is based. The Bayesian approach aims at defining the probability of a solution given the observed data and obtain a solution that best maximizes this probability.

Let $\mathbf{\Omega}$ be the unknown scene to recover (composed of geometry, appearance and illumination), and $\mathbf{I} = \{I_1, I_2, \ldots, I_m\}$ be the set of given images previously defined. Ideally, we would like to find the best scene $\hat{\mathbf{\Omega}}$ that corresponds to the images $\mathbf{I}$. This is called the maximum a posteriori in the machine learning and statistics literature [10, 37]. Then $\hat{\mathbf{\Omega}}$ can be found using the joint probability of a scene given the images:

$$\hat{\mathbf{\Omega}} = \arg\max_{\mathbf{\Omega}} \{p(\mathbf{\Omega}|\mathbf{I})\} \ . \tag{I.5}$$

The joint probability of the scene and the images can be decomposed into a prior probability on the scene $p(\mathbf{\Omega})$ and the likelihood probability $p(\mathbf{I}|\mathbf{\Omega})$ (which represents how likely a given scene is to reproduce the input images).

- The term $p(\mathbf{I}|\mathbf{\Omega})$ corresponds to the probability of having a set of images given the parameters of a scene, and explain how likely is a scene to be the solution. In the case of multi-view reconstruction, this can be seen as a generative model, where the likelihood measures the error between the input images $\mathbf{I}$ and images that would be generated by the scene $\mathbf{\Omega}$.
- The prior term $p(\mathbf{\Omega})$ imposes constraints over the set of all possible solutions, since the non unique solution due to the huge space of possible scenes $\mathbf{\Omega}$ makes it a difficult and ill-posed problem.

Using the Bayes rule, one can rewrite Equation (I.5) as its negative logarithmic form:

$$\hat{\mathbf{\Omega}} = \arg\min_{\mathbf{\Omega}} \{\log p(\mathbf{I}|\mathbf{\Omega}) + \log p(\mathbf{\Omega})\} \ . \tag{I.6}$$

The scene $\mathbf{\Omega}$ is composed of the shape $\mathcal{S}$, the associated reflectance $\mathbf{R}$ and the illumination $\mathbf{L}$. If we know the illumination, and that we consider the shape and reflectance to be independent, then Equation (I.6) becomes:

$$\hat{\mathbf{\Omega}}(\mathcal{S}, \mathbf{R}) = \arg\min_{\mathbf{\Omega}(\mathcal{S},\mathbf{R})} \{\log p(\mathbf{I}|\mathcal{S}, \mathbf{R}, \mathbf{L}, \Pi) + \log p(\mathcal{S}) + \log p(\mathbf{R})\} \ . \tag{I.7}$$

The terms $p(\mathcal{S})$ and $p(\mathbf{R})$ are respectively the prior terms on the shape and the reflectance. $p(\mathbf{I}|\mathcal{S}, \mathbf{R}, \mathbf{L}, \Pi)$ corresponds to the likelihood for a given shape $\mathcal{S}$ and reflectance $\mathbf{R}$ where the illumination $\mathbf{L}$ and the camera models Π are fixed. $p(\mathbf{I}|\mathcal{S}, \mathbf{R}, \mathbf{L}, \Pi)$ measures the similarity between the generated images and the observed images. This part directly depends on the image formation model chosen for the modeling.

Let $\overline{I}$ be one generated image of the scene $\mathbf{\Omega}$ that would correspond to an input image I, based on some image formation model. Ideally, we would like I to be equal to $\overline{I}$ up to noise and small deviations from the model. In the literature, the probability density function p is often defined as a normal (or Gaussian) distribution. Then a natural choice for the likelihood term is:

$$\begin{aligned} p(\mathbf{I}|\mathcal{S}, \mathbf{R}, \mathbf{L}, \Pi) &= \prod_i p(I_i|\overline{I}_i) , \\ &\propto \prod_i \prod_{\mathbf{p}} e^{-(I_i(\mathbf{p})-\overline{I}_i(\mathbf{p}))^2} , \end{aligned} \tag{I.8}$$

where this is a product on all pixels $\mathbf{p}$ of all images i. Then the likelihood energy to minimize can be expressed as:

$$E_{data}(\mathbf{\Omega}) = \sum_i \int_{\mathcal{I}_i} (I_i(\mathbf{p}) - \overline{I}_i(\mathbf{p}))^2 \, \mathrm{d}\mathbf{p} + \text{const} . \tag{I.9}$$

The prior terms in Equation (I.7) are necessary because the set of all possible solutions is too large, and input data only corresponds to a sparse set of images. Therefore, there are not enough observations to explain all the shape and reflectance which makes the problem under-determined. Moreover Equation (I.7) is rather difficult to optimize directly and people generally split the optimization process by alternatively optimizing the Shape $\mathcal{S}$ for a fixed $\mathbf{R}$ and vice-versa.

From Equation (I.9), one may notice that the energy is defined over the whole image domain. As shown in [31, 56, 160], this has the advantage to reduce the minimal surface bias present in most variational approaches without any additional terms or constraints. However, one may also notice that the energy is not only defined on the part corresponding to the region covered by the projection of the surface, but also on the background parts. This means that prior information about the background is needed: either it has to be known in advance (for example using background subtraction, silhouettes or simply by taking images without the object of interest) or should be easy to model [56]. In such a case, Energy I.9 can be split into two terms, the one where the surface projects (the image region denoted by $\Pi_i(\mathcal{S})$) and the one corresponding to the background. In that case the energy can be rewritten as an energy over the surface, and then corresponds to standard energy minimization techniques (See Equation (I.3)). This part will be detailed in Chapter II.

I.3 Information in Multi-view Reconstruction: Exploiting Cues and Priors

While generative models offer a generic approach to reconstruct 3D models, estimating the full radiance of a scene might be too complex. Therefore one may prefer to use different cues in order to express the likelihood and prior energies. While it is often assumed that stereo correspondence is the more common cue to use, this is just one among many that one has at his disposal. The following section describes some of them, and how they can be used for recovering 3D models from multiple images.

We divide those cues into three categories. First, cues that are related to the images such as color, texture, radiance as well as contours. They are usually closely related to the data energy term previously defined since they directly depend on the input images. Secondly, we describe how different constraints on the 3D shape can be used, and often be seen as a prior term on the shape or on the initialization. Finally, we describe how changes in the acquisition setups can lead to additional constraints in order to better pose the problem.

I.3.1 Image-based Cues

I.3.1.1 Correspondence and Matching

The natural way to see stereo reconstruction is to think about correspondences and matching between images. If two points of two different images correspond to the same 3D point, and if the cameras are calibrated, then the coordinates of a 3D point can be easily computed [62, 63]. Image correspondence and matching has a long history in computer vision [84]. Since interest points are easier to handle, a full literature covers feature definition, detection and tracking such as: the Lucas-Kanade tracker, the Sift [104] and more recently for denser interest points Daisy [147]. See surveys on keypoint detector [135] and feature descriptions [110] for further information. In that context, the goal is to define and detect robust point (or patch or line) features to find a sparse set of correspondences. This matching is a pre-requisite in order to automatically calibrate images, do object or camera tracking, structure from motion or align images for stitching. However, such problem is clearly badly defined since two 2D points sharing the same color may correspond to the projection of two different 3D points. This is usually due to the appearance of a 3D point that depends on the viewpoint (if the object is not Lambertian, which is usually the case), or on points that are wrongly matched (because the texture is repetitive, or not discriminative enough, or on textureless objects). In that case robust estimations, and techniques to remove outliers such as RANSAC [44] are usually used. Nevertheless recent algorithms allow to track, reconstruct and map features in real-time such as PTAM [86] or DTAM [120].

In computer vision, in particular in 3D reconstruction, features are widely used because they are easy to handle and fast to compute and match. Indeed, images can be matched to millions of others in seconds [77]. This allows to match images from large image collections and calibrate them using optimization such as bundle adjustment. Finally 3D models can be recovered from large image databases [4, 46]. However, sparse features are mostly used to calibrate cameras and to initialize dense multi-view stereo techniques. Indeed additional steps are required to reconstruct dense 3D models such as patch expansion techniques [4, 52], dense depth maps estimate and merging [46], or surface refinement [154].

Dense image matching

Dense matching can also be performed directly between images. Image registration aims at finding a transformation between two images, and like many non-linear problems, are mainly based on energy minimization techniques. Given two images $\mathcal{I}_1$ and $\mathcal{I}_2$, the goal is to find a transformation $\mathcal{W}$ such that:

$$\hat{\mathcal{W}} = \arg\min_{\mathcal{W}} \{\mathcal{M}(\mathcal{W}(\mathcal{I}_1), \mathcal{I}_2)\} \tag{I.10}$$

where $\mathcal{M}(A, B)$ is an image metric measuring the similarity between two images A and B. Typically $\mathcal{M}$ is the sum of squared differences between the pixel intensity values in $\mathcal{I}_2$ and the transformed image $\mathcal{W}(\mathcal{I}_1)$. One may prefer to use more robust measurements such as the Normalized Cross-Correlation or the Mutual Information [150]. Those metric account for the image dynamic, and then are more robust to intensity variations, even for registering images coming from different modalities [155]. Though originally developed to align 3D volumes [155] and medical imaging applications, image registration can be applied to solve computer vision problems since for example optical flow (usually applied to slow motion) and dense stereo matching (constrained to epipolar geometry) are particular cases of deformable image registration.

Dense image alignment, or deformable/non-rigid image registration, has been successfully applied to solve 3D reconstruction problems. For example, recently Newcombe and Davison [119] use an initial surface recovered from structure from motion. Then Image registration is performed between each back-projection of the views in a reference frame and the image of the reference frame itself, using optical flow [15]. The surface displacement field is then integrated on the surface to optimize it, allowing to produce live dense reconstruction [119]. In the multi-view context, Pons et al. [131] use the same idea and show how surfaces can be optimized using the registration between an image and the projection of the back-projection of a second image onto the surface. Compared to Newcombe and Davison [119] that work on the 2D image space, Pons et al. [131] explicitly use a 3D surface model, and then optimize it to make it matches with the data. In this case the warp function $\mathcal{W}$ is the surface itself to recover. It makes the result more coherent in the multi-view setting since it implicitly exploits the redundancy of information in the different views. Moreover it is also more robust to occlusions since the optimization is done directly in 3D instead of in the 2D images only.

Multiview Photo-Consistency

The generative models for 3D reconstruction use the similar "matching" idea, but in the inverse order. Instead of starting from 2D positions in images and go for 3D coordinates, one may think of the following problem: What is the appropriate 3D position such that the color of its projections in images is the same across images? This way of thinking exactly corresponds to the Bayesian rationale described previously. One advantage that multi-view stereo algorithms have over depth maps or binocular stereo approaches is that they are scene-based. Therefore they enjoy properties in principled manners such as being able to exploit the redundancy of information across multiple views as well as visibility and occlusion handling.

In the ideal Lambertian world, such a point is called to be photo-consistent. The photo-consistency of a point is its ability to project with the same (consistent) color in all images it is visible from. Please note that like the previous case, a $3D$ point that is photo-consistent is not necessary a proper solution. The photo-consistency term is frequently used in multi-view stereo and may have different forms and criteria [42, 81, 91, 130, 137].

We consider the function $g : \mathbb{R}^3 \mapsto \mathbb{R}$ to be the photo-consistency metric. A simple choice for g is to check if the radiance of a point $\mathbf{x}$ is the same on all input images I_i by using the simple following function:

$$g(\mathbf{x}) = \sum_i \frac{1}{2} \left(I_i(\pi_i(\mathbf{x})) - C(\mathbf{x}) \right)^2 \nu_{\mathcal{S},i}(\mathbf{x}) \,, \tag{I.11}$$

where $\nu_{\mathcal{S},i}(\mathbf{x})$ is a visibility function that is 1 if $\mathbf{x}$ is visible from the camera i, and 0 otherwise.

If g could be computed on the whole space $\mathbb{R}^3$, then finding the shape of the object would be a segmentation problem, where the surface to recover would be the one such that all its points satisfy $g(\mathbf{x}) = 0$. However, the visibility function makes g only easily computable for one given surface $\mathcal{S}$.

Using generative models as explained previously in Section I.2.2, the photo-consistency measure can be adapted in order to minimize the observed intensity in the image I with the radiance $\overline{I}$ of the current model:

$$g(\mathbf{x}) = \sum_i^n \frac{1}{2} \left(I_i(\pi_i(\mathbf{x})) - \overline{I}_{\mathcal{S},i,\pi_i}(\mathbf{x}) \right)^2 . \tag{I.12}$$

In that case the error measure is computed in the image domain and therefore more closely corresponds to a Bayesian vision of the problem. In contrast to Equation (I.11), the observed radiance $\overline{I}$ may depends here on the view-point direction, and eventually on some reflectance and illumination models. There is a major problem in that formulation: the function that generates the image $\overline{I}$ is non linear and depends on the shape itself and in particular the *visibility* of that point with respect to the shape. The same dependency appears in Equation (I.11). We will address this visibility issue in the following Section I.3.1.2.

I.3.1.2 Contours and Visibility

Contours or boundaries often give precious information about the shape of an object. Contours in images are usually due to two main reasons:

- Occluding contours: which are related to the visibility of the surface, due to self-occlusions or the presence of additional objects.

- Contours due to changes in the object's appearance: this can be due to an object composed of different materials, having different colors, having projected shadows on it, or having brightness discontinuities (due for example to edges having discontinuous normal directions).

If the latest can be used as a matching cue, this is not necessarily true for occluding contours. In fact if the baseline between two images is large enough, the apparent contours in the two images do not necessarily correspond to the same occluding contour. This is the main reason why stereo matching between wide baseline stereo pairs often fail or contains errors, because features may be detected on parts that do not perfectly match in 3D.

In 3D reconstruction techniques, one classical approach using contour cues is to use image's silhouettes. Silhouettes are typically used to apply visual hull reconstruction [47, 100], which gives a good approximation of the object's shape if enough images are available and if the object does not contain too many concavities. The main advantage is that it is fast to compute and therefore can be used for interaction for instance. Visual hull can also be obtained by applying stereoscopic segmentation [160], assuming that the object's radiance and the background radiance can be discriminatingly segmented. In that case the idea is to segment the 3D space as being inside or outside the shape. Visual hulls are usually used as initial conditions for more sophisticated multi-view stereo techniques or are used as additional boundary constraints such as in [138].

Similarly as with the visual hull, additional constraints on apparent contours and occluding contours can be used to improve multi-view stereo algorithms [66, 153]. Many other authors have been working on the shape from contours problem so far, which leads to a good understanding of the visibility.

Visibility is a major issue and a fundamental problem in computer vision and multi-view reconstruction in particular. If a 3D point is a good candidate for the reconstruction, then that means that all the space between that point and the camera center should be free [55]. In their paper, the authors show how to compute the gradient of energy functional depending on the visibility, such as the one in Equation (I.12). This kind of energy is known as reprojection error functionals. This makes their surface evolution technique matches the recovered shape with the apparent contours in the images. We extend that work to be applied with polygonal meshes (See Chapter II and [31]), which is one of the contributions of this thesis.

I.3.1.3 Photometric Information

For now on, we presented works that use color and texture for matching and reconstruction. However, the radiance captured by a camera carries more information than the intensity value itself. Let us first remember that the image radiance comes from light reflection as illustrated in Figures I.2 and I.3. As a consequence intensity is the result of light transport, and is a product of light, reflectance, surface orientation and position. If assumptions about some of those parameters are known or can be estimated, then information about the others can be inferred.

Shading

One cue that has been exploited for years is the shading of objects. The shading on one image is a big cue on the shape of an object, and is useful to distinguish concave and convex parts, or appreciate its smoothness or roughness. This idea has even been used for hundreds of years in painting to create the impression of 3D from flat 2D image representations.

From the beginning of computer vision, a large area of research has been exploiting shading to recover 3D shapes from a single image. This is known as the *shape from shading* problem [71, 132]. The key idea is that the 3D perception of shapes highly depends on the shading, which gives information on the surface normals orientations. Under the constant brightness assumption, corresponding to the Lambertian reflectance model, the observed radiance is proportional to the dot product between the normal of the surface and the lighting vector L. The idea is to find for each pixel its associated surface normal using a Lambertian model:

$$I = \rho \, \mathbf{n} \cdot \mathrm{L} \,, \tag{I.13}$$

where I is observed intensity value (radiance) at the current pixel, ρ is the surface's albedo, L is the light vector and $\mathbf{n}$ the surface normal to recover.

If a set of normals can be estimated at each pixel of the image, then a 3D surface can be inferred from those information. However because of illumination estimation, Bas-relief ambiguity [7] and varying surface albedo, it is often difficult to estimate the normals by applying shape-from-shading. If several images are acquired while moving the images and/or the light sources, ambiguities can be reduced and the normal estimation becomes more robust and accurate. If the camera is fixed and the light is moving, this is known as the *photometric stereo* problem, which will be detailed in Section I.3.3.3. In the multi-view settings, several approaches have been proposed and most of them use energy minimization and surface evolution techniques [79, 163]. Those algorithms even provide a unique framework to combine both multi-view stereo and shape-from-shading, similarly as what we will see in Chapter III.

Towards Complete Photometric Modeling

Shading is usually exploited under the Lambertian assumptions. But when the scene is not Lambertian, which is the case for most (if not all) real world scenes, cues such as Lambertian shading or stereo correspondences are not valid and algorithms fail to reconstruct accurately the surface. However, photometry can still be exploited to estimate the surface normals even for non-Lambertian surfaces. In the multi-view setting, non-Lambertian assumption results in color changes across images. Therefore, correspondences can not be straight-forwardly found in that context.

One solution to that problem is to consider robust error measurements. Instead of measuring difference on pixel intensity measures, one may consider to match the dynamic of the texture by using for instance (normalized) cross correlation (NCC) or mutual information (MI). This way of reasoning allows more robust matching and better multi-view stereo algorithms [130, 154]. One other approach consists in considering points that do not satisfy the constant brightness assumption as outliers. For examples, points in shadows (typically dark pixels) or points in strong highlights (for example saturated pixels) are not used [9, 68]. Robust estimators can also be used to remove outliers or attenuate their influence. In that case more samples are usually needed to compensate the lack of information. Some others modify the input images to have specular free images [162], or to be photometric invariant [106, 173]. An alternative way is to use material samples in order to match them with target objects [69, 60].

In the rest of this thesis, we will denote by $\mathbf{R}$ the parameters of the reflectance model, and assume objects can be modeled using BRDF's. In particular transparent and purely specular object reconstruction techniques that are nicely reviewed in [73] are out the scope of this thesis work.

Previous works tried to estimate both the shape and the reflectance using several strategies. Some propose to use more general reflectance / radiance parametric models. See for instance [78, 79, 164, 163]. In practice, these approaches suffer from several limitations. First of all the reflectance model has to be known in advance and this constrains scene to be composed by materials consistent with the chosen reflectance model. Such algorithms tend to solve non-linear systems of thousands of variables (one reflectance / radiance model per surface point), or need additional assumptions (single or fixed number of materials, single specular component, etc.). Those models are difficult to be optimized and generally require to alternatively estimate the reflectance and the shape [115, 163]. They are numerically unstable and easily tend to get stuck in local minima [79]. Moreover, the algorithms are generally ill-posed, and then require strong regularization which over-smooth the obtained results. Finally, the reflectance and illumination models need also to be approximated. Although such algorithms show reasonable results for perfect synthesized scenes, their application to real-world scenes is complex and requires accurate camera and light calibration, and is still an open research area. Mainly based on generative models, they however offer good ways to refine surfaces at a last stage of 3D reconstruction.

I.3.1.4 Motion Information

In computer vision, motion is a strong cue that can help analyze images and videos. We already saw that optical flow can be used to estimate the displacement flow between two images, and a fortiori, estimate the motion between consecutive frames as well.

In the 3D space, the equivalent of optical flow is known as *scene flow*. Instead of estimating a dense velocity field on the images like optical flow, the goal here is to estimate a dense displacement field in the 3D space or attached to a 3D shape (often a 2D surface embedded in $\mathbb{R}^3$). This motion information can be used to track the surface points. This can be used either to improve the quality of stereo reconstructions [72], multi-view $3D$ reconstruction [27, 131], or for interaction purposes [126]. An other approach in exploiting motion cues is to perform tracking on a reconstruction. In that case a 3D template is deformed to match the images in order to get a temporally consistent deformable model across a whole sequence, see for example [17].

Recently, authors have been exploiting motion in order to build temporally coherent 3D models. In their papers, Aganj et al. [3] for $4D$ visual hull models and Courchay et al. [27] for $3D + t$ multi-view stereo exploit this information. In fact, we also propose to exploit motion in order to perform dynamic photometric stereo in [76]. In the paper, we combine optical flow with variational methods in order to estimate temporally coherent normal maps of textured dynamic scenes. However, this is out the scope of this thesis work. Although motion information could also be added in our framework similarly as done in [27, 131], in this thesis we only focus on 3D reconstruction of static scenes - or eventually static reconstructions of dynamic scenes (independent shapes at time t).

I.3.2 Shape Prior, Regularization and Initialization

Multi-view reconstruction is an ill-posed problem, not only due to matching difficulties and ambiguities, but also because of the visibility problem. Points of scene that are seen by no cameras, or just by one, cannot be explained using matching. Then more information are required. As explained earlier, with the Bayesian formulation of the problem, in addition to the likelihood term, there is a prior information term. Then, the energy minimization problem often has a regularization term based on the surface itself.

Initialization

In non-linear optimization problems, the optimization starts from an initial condition. In gradient descent based methods, the algorithms are subject to local minimas. To avoid those solutions, it is common to start from a good initialization. In multi-view stereo reconstruction, those initial conditions can be of different natures.

The simplest idea is to take the visual hull as the initial surface. The best advantage is that the visual hull is usually easy and fast to compute. The main idea is to segment the object of interest to obtain its silhouette. This is usually done by background subtraction, but one may use recent segmentation algorithms. The intersection of the cones formed by the silhouettes and their associated camera center is called the visual hull, and can be exactly computed by silhouettes. Visual hulls are commonly used as initial surfaces in variational methods for 3D reconstruction [81, 82, 140, 161, 163]. Moreover, visual hulls have interesting properties. If points lying on the visual hull are photo-consistent, they are likely to belong to the surface itself. Furthermore, the surface's normal at those points is the one of the visual hull and can be used to estimate the light [68], or serve as a basis for multi-view stereo carving [51].

If one has enough images in his disposal and can extract feature points and match them, then those features can be used to extract an initial surface. Furukawa et al. [52] propose to extend those interest points by patches, and then grow and interpolate those patches by checking the multi-view consistency. An other approach is to directly construct a triangular mesh representation of the 3D point cloud coming from the features. In their paper, Labatut et al. [95, 97] use a Delaunay tetrahedrization of the space, and recover the surface thanks to a graph-cut algorithm based on visibility constraint. They show how to incorporate their algorithm in a complete multi-view stereo pipeline to reconstruct large-scale 3D models [154].

Shape Prior

An other approach is to use a template model and deform it to match the data. This is generally used to reconstruct dynamic human bodies, since a strong prior can be used on an animated human model. While allowing more robust results in concerned applications where the template model matches the data, they focus on specific applications and can not be used for general 3D modeling of generic scenes. Similarly, in variational image segmentation, a lot of effort has been recently done in adding shape prior in minimizations. While this significantly improves the robustness of the segmentation (for example with respect to noise or occlusions), those methods have two main drawbacks: first they are limited by the prior template, and second their extension to 3D is not straightforward. For example, one could minimize the following energy functional:

$$E(\mathcal{S}, T)_{\text{prior}} = D(\mathcal{S}, T(Q)) \,, \tag{I.14}$$

where Q is the reference (prior) model, and T is a transformation that align the reference model to the target. $D(A, B)$ is a distance function between two surfaces A and B. D can for instance be the Hausdorff distance between the two surfaces. The main difficulty is the estimation of T.

While prior models are used now in several domains to recover the camera pose estimation, or register images to CAD models, 3D shape prior for multi-view reconstruction algorithms is still an open research area.

Regularization

In the previous section, we defined our energy as a sum of a data term and a regularization term. The surface regularization term is often necessary to better constrain the optimization and be more robust to noise in the energy functional. In general, variational methods use a smoothness term as the regularization. A classical regularizer used in many variational approaches is to minimize the total surface's area using the following energy functional:

$$E_{\text{reg}}(\mathcal{S}) = \int_{\mathcal{S}} \mathrm{d}s. \tag{I.15}$$

Its associated gradient flow is the so called *Mean Curvature Flow*. This energy functional tends to help the optimization to get to the closest minimal surface area solution. This regularization is used in many multi-view stereo problems.

Other smoothness terms can also be used. Instead of minimizing the surface area, one may add a smoothness term to penalize variations on the surface normals. Smoothing the normals can be applied by minimizing the following energy functional:

$$E_{\text{reg}}(\mathcal{S}) = \int_{\mathcal{S}} |\mathbf{n}(\mathbf{x}) - \mathbf{h}(\mathbf{x})|^2 \,\mathrm{d}s, \tag{I.16}$$

where $\mathbf{n}(\mathbf{x})$ is the normal at the surface point $\mathbf{x}$, and $\mathbf{h}(\mathbf{x})$ is a unit vector. Typically $\mathbf{h}(\mathbf{x})$ is the mean normal on a local neighborhood around the point $\mathbf{x}$. In that case the smoothing algorithm directly depends on how $\mathbf{h}$ is defined: for example, this can be a simple mean Gaussian normal vector or or more complicated one used for example to perform anisotropic smoothing.

An other very popular regularizer is to use the *Total Variation* of a characteristic function representing the surface. Let H_ε be a regularized Heaviside function that models a characteristic function [21], then the total variation of this function can be expressed using the following energy function:

$$\int_{\Omega} |\nabla H_{\varepsilon}(\mathbf{x})| \, \mathrm{d}\mathbf{x} \, . \tag{I.17}$$

Total variational regularizers have been successfully applied in order to solve convex segmentation problems [14, 22], in the binary segmentation case as well as in multi-label segmentation problems [19, 102, 166]. Recently, they also have successfully been applied to multi-view shape reconstruction problems using convex relaxation methods, see [91]; as well as in real-time structure from motion using dense tracking and mapping DTAM [120].

The approaches presented above work to regularize the shape. However, this can also be applied in regularizing the reflectance on the shape in order to penalize variations in the surface materials. Several works use this strategy by considering that materials are locally smooth on the object of interest.

I.3.3 Setup-based Constraints

If cameras are good to provide information and appearances of objects, this is not the only device one has to his disposal. For many years people have used for instance Laser scans, which, despite limitations such as noise and material restrictions, provide good 3D models after geometry processing steps or even after being combined with vision algorithms.

Most of those devices that do not use cameras cannot recover the appearance and therefore cannot exploit photometry. However, cameras used in particular setups, or along with additional information such as lighting, can lead to powerful 3D scanning systems. Exploiting lighting conditions, for instance, allows in many vision applications to remove ambiguities and simplify the problems to solve. In the following, we describe some of those setups with their advantages and their use in multi-view settings.

I.3.3.1 Active Illumination Systems

For decades, active vision systems have been widely used and developed. They provide efficient and easy way to reconstruct 3D scenes and usually rely on active illumination systems, but are difficult to use in a multi-view setting.

Methods based on structured light are probably the most commonly used, and recent methods now use video projectors. The main reason is that the market now provides cheaper video-projector at reasonable prices, and recent methods have been proposed to easily calibrate such systems [35]. Then the problem becomes the one of matching the projected pattern with the image of the camera observing the object.

In order to solve the matching problem, encoded patterns are usually used. These methods can be divided into two categories, temporally coded patterns and spatially coded patterns. The first category consists in changing the pattern across time. Because matching can be easily performed, it provides an easy and fast method for 3D reconstruction [99]. However, because several patterns are needed to perform the matching, it makes this method impossible to use for moving scenes and difficult to reconstruct the entire shape of objects. On the other hand, spatially coded patterns approaches use a single static pattern, coded to reduce matching ambiguities. Such systems allow fast reconstruction suitable for real time 3D reconstruction. However, the main drawback is that it cannot straightforwardly be applied to entire shape acquisition in the multi-view setting, because of interferences between the different patterns projected on the object, see [50].

As explained previously in Section I.3.1.2, visual hull is an easy way to approximate and compute a 3D shape. However it relies on image segmentation and background subtraction techniques. An easy and simple way to solve this problem is to use a moving light source, and a camera to acquire the cast shadow projected onto a flat surface (for example a sheet of paper). Then the segmentation of the projected silhouette is made easy and can be used directly to construct the visual hull. Such techniques are referred as shadow-grams [158], and are fast and accurate because the silhouette extraction is made easy by the shadows.

Recent advances in technology let also appear devices that directly compute a depth map. This is for example the case of Time-of-Flight (ToF) cameras, that emit infra-red light and estimate the depth from the time the signal is reflected on the surface and captured back on the camera. An other example is devices such as Kinect, where the technology relies on structured light as described previously but in the infra-red domain. Those systems offer practical and fast depth map estimation. However, there extensions to multiple devices settings are not straight forward because of possible interference between the sources or the lighting environment.

I.3.3.2 Computational Cameras

Recently, there has been a lot of interest in computational photography. This domain consists in adapting the acquisition device to the goal it is designed for [114, 117]. Even though most of recent research in computational photography has been done in enhancing images and improving their quality, some computational cameras techniques are related to 3D scene reconstruction such as depth from defocus techniques.

Recently, powerful cameras have been designed to exploit focus changes in images in order to perform shape from defocus. They usually change the camera aperture in such a way that the image structure is preserved. By finding an appropriate deconvolution method, one can achieve good results of 3D reconstruction from a single shot, or a set of shots with different apertures [103, 169]. Shape from diffusion consider a similar idea. Instead of changing the aperture and put a binary (or gray scaled) pattern filter, the authors [168] propose to add a diffuser. The diffusion angle is spatially varying on the aperture, and depth can be estimated using deconvolution techniques.

Although computational cameras offer promising perspectives, in the rest of this thesis we focus on standard cameras. Those are easier to get, easy to use and do not need post-processing steps that could alter the image formation process. However, because we focus our attention on the modeling of the problem, these algorithms could be successfully adapted to those kind of camera.

I.3.3.3 Photometric Stereopsis

In multi-view stereo, it is sometimes difficult to find correspondences, like for instance if the object is textureless. To avoid this problem, one can use photometric stereo. Photometric stereo is a reconstruction technique that uses a single camera and a moving light source. The goal is to take different images of an object under different illumination conditions with the same view-point. Since the camera is fixed, no matching is required and a single pixel can be seen under different lighting conditions. Then photometric information can be extracted, which in general, leads to the estimation of the surface normal (Figure I.4 illustrates the photometric stereo approach). By repeating this process for all pixels, it is possible to estimate a normal map and then infer the 3D shape through normal integration techniques. If the scene is Lambertian, photometric stereo becomes a very compelling method to estimate both the shape and the albedo. Even if Lambertian assumptions are not fully satisfied, this can leads to good normal estimations and reconstruction results, see [67, 68, 76].

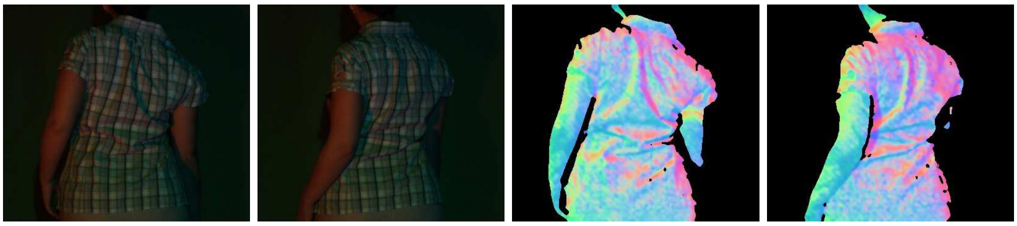

Figure I.4: Example of Dynamic Photometric Stereo [76]. Left: Two images from a fixed camera taken under three color light sources; Right: Estimated normal maps.

When the object is not Lambertian, several approaches have been proposed. One neat idea is to take sample images of a reference object with the same material as the target object to reconstruct the scene under the same illumination conditions. The surface's normals are estimated by matching the intensities with the one of the reference images, see [69]. This can even be extended to be able to deal with multiple materials or even when reference objects are unknown but the number of fundamental materials is small [60]. Although it can deal with many materials and anisotropic BRDF, material samples are needed. This can be a restrictive assumption in concerned applications. Another approach is to look for highlights (points of maximum intensity), where the surface's normal should be the unit bisector vector between the light source direction and the viewing direction. In that case surface's normals can be easily computed [25], but since only saturated pixels are accounted for, a large number of images is required to estimate the normal at each image pixel. Other methods have also been proposed, the recent work of [70] for instance shows impressive results.

On the other hand, the main problem of photometric stereo is that it usually requires a precise calibration of light sources. Some work have been extended in order to work with complex lighting [40] or even unknown one [6], but most works have been designed to work using a single calibrated light source. This makes this method not very flexible to use for outdoor scenes or even dynamic scenes. However dynamic photometric stereo can still be performed by exploiting temporal smoothness, and by using colored light sources [67]. Even though this works for gray textureless surfaces, we have extended it by combining it with optical flow [15] in order to perform colored dynamic photometric stereo [76].

Multi-view photometric stereo can be performed in order to recover full 3D models by exploiting multi-view settings. Several techniques have been applied, for example by optimizing a mesh starting from the visual hull using multiple views/light configuration [68] or by estimating normal maps from a light stage [40] and finally integrate them in 3D to obtain full 3D reconstructions of dynamic scenes [152]. Those methods [68, 152] have been proved to provide high quality 3D reconstructions. In this thesis, we will show that integrating several normal maps in a multi-view setting can be used using a generative based approach to recover full, dense and high quality 3D models, see Chapter III.

I.3.3.4 Helmholtz Stereopsis

As we have seen previously, most of multiview reconstruction algorithms rely on image correspondences (as done for instance in multiview stereo [57]) or shading (using the normal information in multiview shape from shading [80, 79] or multiview photometric stereo [68]). When texture information (stereo case) is good enough or Lambertian assumption is sufficiently verified, those methods have been proved to give good results with surfaces that are nearly. They then obtain either accurate correspondences or accurate normal estimates. When objects are not Lambertian, several methods can be performed as presented in previous sections. However, those approaches are not generic enough to be able to reconstruct surfaces of any spatially varying BRDF, even under constrained scenarios.

Helmholtz Reciprocity

The bidirectional reflectance distribution function (β) is the same if the light input direction $\hat{\mathbf{v}}_l$ and the camera direction $\hat{\mathbf{v}}_c$ are switched: $\beta(\hat{\mathbf{v}}_l, \hat{\mathbf{v}}_c) = \beta(\hat{\mathbf{v}}_c, \hat{\mathbf{v}}_l)$. This particularly interesting property is known as *Helmholtz Reciprocity* [65, 171]. In practice, the idea behind Helmholtz reciprocity is to exploit a pair of images under a single light source, where camera centers and light positions are exchanged at each shot. It uses the fact that in this particular setup, for a single reciprocal pair, the relationship between two radiances of a single surface point is independent on the reflectance β. Then, a constraint that links the two observed radiances can be formulated, which leads to the following equation:

$$I_c \, \frac{\mathbf{v}_c}{|\mathbf{v}_c|^3} \cdot \mathbf{n} \;=\; I_l \, \frac{\mathbf{v}_l}{|\mathbf{v}_l|^3} \cdot \mathbf{n} \,, \tag{I.18}$$

where I_c (respectively I_l) is the intensity observed at the camera of view direction $\mathbf{v}_c$ (respectively $\mathbf{v}_l$) and light direction $\mathbf{v}_l$ (respectively $\mathbf{v}_c$); $\mathbf{n}$ is the unit surface normal. During the last decades, previous authors have proposed to use Helmholtz reciprocity in order to perform 3D reconstruction [61, 149, 170, 171, 172].

Figure I.5 illustrates a reciprocal image pair that shares interesting image properties. That can be used to estimate the normals of the surface using Equation (I.18).

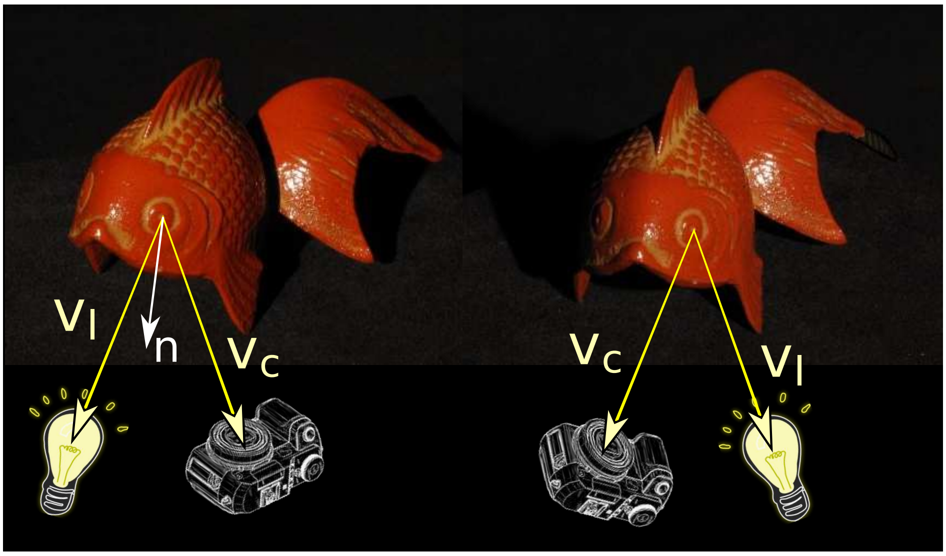

Figure I.5: Helmholtz reciprocal image pair. Those two images are taken such that the camera and light positions are switched. Those two images share important properties: for example specular highlights appear at the same locations in both images, or cast shadows boundaries in one image correspond to occluding contours in the second image. Helmholtz reciprocity allows to constrain surface normals in reconstruction algorithms.

Contrary to works described in previous paragraphs and in particular in Section I.3.3.3 about photometric stereo, methods based on Helmholtz reciprocity [61, 149, 170, 171, 172] allow to accurately estimate the normals at one point, independently on the reflectance model. This can be used to obtain a 3D surface. In this context, modeling the reflectance, having material samples or being photometric invariant is not required. Nevertheless, contrary to most of multiview stereovision algorithms, the state of the art in Helmholtz reconstruction is limited to depth map reconstructions. In this thesis, we will see in Chapter V that we push the envelope by proposing an entire shape 3D multi-view Helmholtz stereovision method.

I.3.4 Conclusion

Many cues are available to infer the 3D shape from images. Contrary to works that are usually limited to particular scenarios, the goal of this thesis is to propose a framework that can equally be applied to a number of classical scenarios such as for example multi-view stereovision, photometric stereo or multi-view shape from shading. Similarly as done in Yoon et al. [163], we combine different cues into a single framework. Then we naturally account for classical cues in 3D shape reconstruction such as stereo matching, shading as well as apparent contours in images, such as in Gargallo [55]. In addition, we apply it to different scenarios where the setup could eventually help constraining the problem. This way, the same framework can also be used to apply multi-view photometric stereo or multi-view Helmholtz stereovision.

I.4 Reconstruction Tools for Deformable Surfaces

In this section, we explore surface evolution solutions for shape reconstruction. Deformable models are often used in computer vision to address those kinds of problems. In Section I.2, we have seen that the multi-view shape reconstruction problem could be seen as an optimization problem. This can be performed by minimizing an appropriate energy functional defined using the different cues and priors presented in Section I.3. Until now, we focused more on the modeling part as well as the definition of the objective function. In this section we present tools and in particular deformable models that are generally used to solve those non linear optimization problems.

In the following of this thesis we will focus on deformable meshes. Then in this section we first briefly review some of the techniques that do not fall in this category and that are used in multi-view reconstruction. In particular, we will focus on deformable models that use surface representations. Then we will focus on mesh-based surfaces and motivate that choice for the rest of this thesis.

I.4.1 Rapid Overview of Previous Work

Deformable models are usually referred as Snakes [85], Active Contours [18] or deformable surfaces [111], and often describe a surface of co-dimension 1 (a 2D surface in $\mathbb{R}^3$, a curve in $\mathbb{R}^2$ and so on) that is deformed from an induced velocity field from an initial surface. That velocity field depends on the application, and in our case follows the gradient descent scheme presented in Section I.2. Then it can be arbitrarily applied to multi-view stereo, multi-view photometric stereo or any other computer vision tasks depending on the cues one wants to use (See Section I.3) to compute the gradient or velocity field.

Minimizing an energy functional directly depends on the surface representation one has at his disposal, and has been widely studied over the past decades. We will overview some of those optimization methods. Graphical models such as Markov random fields are usually represented using occupancy grids. The 3D space is discretized and represented by variables and edges connecting them. The energy is discretized in term of those variables with probability density functions. The shape can be obtained for example by Graph Cuts algorithms that will extract the surface by finding the minimal cut of the variable edges. Under certain conditions, the global optimal solution can be found [93]. Graph cuts are commonly used in multi-view stereo [138, 153].

Implicit representations are usually referred as the Level Set approach [123]. The surface is represented by the 0-level of an implicit function defined on the 3D space that is negative inside the shape and positive outside. Approaches using this representation have been extensively used in multi-view stereo [42, 56, 131] as well as in photometric-based multi-view reconstruction [79, 163].

Convex relaxation methods have been recently applied to multi-view reconstruction [28, 91]. The idea is similar to the level set approach in the sense that an implicit function is optimized on the 3D space. In that case, the problem is relaxed on a function defined on a convex set (between 0 and 1) and regularized using Total Variation. Since their function is convex, a global optima can be found.

All of those methods are implemented on 3D voxel grids and therefore contain limitations due to space discretization. The chosen resolution then directly affects the model precision, and fine resolution requires high memory usage. In order to reconstruct high resolution models and use limited memory, several approaches have been proposed for instance using narrow-bands to store only useful information or octrees that allow non-uniform (but still fixed) discretization of the 3D space. Finally, Eulerian methods cannot be used to track surface properties such as textures or reflectance, and therefore are not suitable for dynamic reconstruction and surface tracking.

I.4.2 Deformable Meshes

Recently, deformable models based on explicit representations have become more popular. In the variational framework, this type of discrete representations leads to the Lagrangian setting, which are generally contrasted with Eulerian approaches which are mainly based on implicit representations such as the level set representation. In computer graphics for instance, surface representations based on polyhedral and especially triangular meshes are the most commonly used. Moreover, the design of graphics hardware makes triangular meshes the most natural way to represent surfaces in a large number of applications [11].

During the last decades, Eulerian methods have become very popular mainly because they allow us to naturally deal with topological changes. In particular, this last setup has been extensively used for 3D reconstruction problems. Nevertheless, recent advances in mesh processing allow Lagrangian methods to enjoy the same facilities [98, 129, 167]. Even, the same idea can be done in higher dimensions in order to build $4D$ models [3] or $3D + t$ temporally coherent 3D models [27]. Moreover, the level set representation used in the works mentioned above has usually high computation cost and very poor memory efficiency when reconstructing high resolution 3D models. Polygonal meshes have a better precision since they are not limited to a fixed spatial resolution, and then can be used to reconstruct thin structures. Optimization highly depends on the surface representations one chooses, and polygonal meshes seem to be an appropriate choice. Moreover, recent advances in computer graphics allow to enjoy interesting properties and computations on polygon meshes, we refer to [11] and the references therein for further information.

Lagrangian methods may be classified into two different categories:

- The first strategy consists in:
    1. formulating the problem with continuous surfaces and computing the gradient in the continuous framework,
    2. then discretizing the continuous gradient in order to apply it to the discrete surface representation.

- The second strategy consists in:
    1. formulating directly the problem with the discrete surface representation, by discretizing the energy accordingly,
    2. then computing the exact gradient of the formulated discretized energy.

In fact in the first strategy we do not know if finally in practice the discrete representation minimizes something; in the best case, we do not know which exact energy the computed solution really minimizes since it does not necessarily corresponds to the shape representation. Also, surprisingly, a number of works follow the first strategy, see for example [9, 36, 167]. In particular, in this case the computed gradient assumes normal velocities whereas the second approach may have tangential components which leads to more coherent flow.

In this thesis, for those reasons we follow the second approach that has been previously referred as a *discretize then optimize* strategy by [154] which seems more appropriate because it is consistent with the representation. Then our approach is similar as other previous authors works: for example, Slabaugh and Unal [139] who deal with surface segmentation, Eckstein et al. [39] who are interested in shape matching in order to register scanned meshes, or Vu et al. [154] who propose a complete multi-view stereo-vision algorithm whose final surface refinement step is performed with a deformable mesh that is similar surface evolution to ours. We also use the similar strategy presented by Debreuve et al. [30] who deal with discrete parametric active contours for segmentation or Dziuk and Elliott [38] using finite element modeling on evolving surfaces. In this thesis, we will represent the surface $\mathcal{S}$ by a triangular mesh $\mathbf{X}$, where topology changes can be handled by the methods described in [129, 167]. Following those works, we will show how to compute the gradient flows for the kind of energy functionals presented in Section I.2. The details of this computation will be presented in Chapter II, whereas its applications to multi-view shape reconstruction will be shown in Chapter III.

I.5 Contributions and Organization of this Thesis

In this thesis, we provide practical solutions for the shape from multi-view cameras problem. We exploit different cues as diverse as color and texture information, segmentation prior, constraint multi-view setups or photometric cues. In all the following contributions, we use generative approaches based on Bayesian rationale in order to recover the 3D model that best corresponds to the observed images.

- In Chapter II we propose to model the 3D world as represented by a 3D surface. Such 3D representation allows to directly link the 3D shape to the input images in a coherent way and avoid redundancy in the representation. We propose to see the 3D world as piecewise planar such as represented by triangle meshes. We show how to optimize such shape representation in order to recover 3D models via surface evolution techniques. We use gradient descent techniques and we show how to compute the gradient of a generic likelihood energy often used in computer vision and related fields. By adding different cues such as stereo-vision correspondence or photometric information to this optimization, one may adapt the energy functional depending on the application. We show different cases for solving the multi-view shape from images problem.

We focus our attention on two particular points toward optimizing deformable meshes. First we rigorously compute the gradient of generic energy functionals depending on visibility. This ensures that the occluding boundaries of the surface match with the occluding contours in the observed images. Second, we also account for functionals depending on the normals. This is needed to nicely account for photometric information (for example the shading) and then allows to produce high quality 3D models.

- After showing how to optimize such deformable meshes in Chapter II, Chapter III demonstrates how this can be applied for 3D reconstruction from images. This is done by adapting an appropriate energy functional depending on the concerned application. In particular the approach is successfully applied for multi-view stereo reconstruction as well as multi-view reconstructions based on photometry, where we present reconstruction results on several data. The coherent gradient flow of Chapter II is suitable for recovering accurate and detailed 3D models.

- In Chapter IV, we propose to segment a shape into several meaningful parts. This provides semantic information about the scene, from a geometric aspect as well as a photometric one. We present a continuous optimization scheme based on recent image segmentation techniques that we adapt for the case of segmenting data on surfaces. We propose a multi-region segmentation algorithm based on convex relaxation. Such algorithm can be useful to decompose the mesh into regions of uniform property such as surface radiance, surface reflectance or any geometric information. For example in the case of reflectance, only one needs to be estimated for one particular region. Since those points all share the same reflectance, this last one is easier to estimate.

- In Chapter V, we address the problem of recovering the shape of object of any arbitrary BRDF model. Instead of exploiting photometric cues directly, we change the acquisition setup using Helmholtz reciprocity. This particular setup allows to constrain the problem in such a way that it becomes independent on the reflectance model. This is done by taking reciprocal views, which are a pair of images that consists in an image taken under a single source light illumination, and its reciprocal view taken by switching the camera and light positions). Then we can define a likelihood energy functional to minimize and use the work presented in Chapter II to recover the 3D shape. It is then possible to reconstruct full and high accurate 3D models of objects of arbitrary reflectance model.

To summarize, we provide tools for solving variational problems of computer vision using deformable triangle meshes. By defining appropriate energy functionals based on visual cues from multiple images, we show how to automatically obtain dense and full 3D models. We give a discussion about the thesis and future work in Section VI, which also concludes this thesis. The list of related article published based on this thesis work is available in Section B.

CHAPTER II

Gradient Flows for Optimizing Triangular Meshes: Applications to 3D Reconstruction Problems dealing with Visibility

This chapter tackles the problem of using variational methods for evolving 3D deformable surfaces. As shown in the introduction, deformable surfaces are of wide interest to solve 3D reconstruction problems. We give an overview of gradient descent flows when the shape is represented by a triangular mesh-based surface, and we detail the gradients of two generic energy functionals which embody a number of energies used in mesh processing, computer vision and related fields. In particular, we show how to rigorously account for visibility in the surface optimization process. The gradient correctly takes into account the visibility changes that occur when a surface moves; this forces the contours generated by the reconstructed surface to match with the apparent contours in the input images. In the next chapter we will present different applications regarding 3D reconstruction from multiple views for which the visibility is fundamental.

Contents

II.1 Introduction

As we have seen in the introduction of this thesis, variational methods are commonly used in computer vision and computer graphics to compute, improve and process surface interfaces. Such approaches consist in defining an energy whose minimum is reached by the surface of the object of interest. In particular, this framework has been widely used in 3D reconstruction problems, see for example [23, 42, 56, 81, 82, 89, 91, 96, 131, 143, 142, 157, 160, 163]. In mesh processing, geometric flows have been extensively used in different applications such as texture synthesis [8], mesh denoising [33, 108] and shape matching [39].

This chapter focuses on the optimization of *2D surfaces of* $\mathbb{R}^3$ represented by *triangle meshes*, via gradient descent methods. We rigorously establish and detail the *gradient flows* of some *generic* energies which encompass a large number of energies used in computer vision and for which the normal to the surface and the *visibility* appear in their formulation. We demonstrate the interest of this contribution by illustrating it via several applications in the next chapter, in particular applications in 3D reconstruction from multiple calibrated cameras. But before we will first focus on the theoretical part and gradient computation.

II.1.1 Considered Energies: from Weighted Area Functionals to Functionals that account for Visibility

In this paper, we first consider the following generic energy:

$$E(\mathcal{S}) = \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, \mathrm{d}s\,, \tag{II.1}$$

where $\mathcal{S}$ is a $2D$ surface embedded into $\mathbb{R}^3$. Here $g : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$ is a scalar function defined on the surface that eventually depends on the normal $\mathbf{n}$ to the surface $\mathcal{S}$; $\mathbb{S}$ being the unit sphere of $\mathbb{R}^3$. $\mathrm{d}s$ is the element of area of the surface. This generic energy is very classical and is called a *weighted area functional*. It has already been studied in the literature in the continuous framework, see in particular [58, 59, 141]. A number of energies proposed in the computer vision, image processing and mesh processing literature are particular cases of this energy.

In this paper we are also considering more complex families of energies. Generally in inverse problems, all rests on a prior knowledge (models, regularization etc) and the data fidelity. A common solution to inverse problems is provided by minimizing a criteria which compares the real input data to the synthetic data generated by the models. Also, to be complete, the comparisons must be done on the data space. In particular, as explained in

[56, 140], in 3D reconstruction problems from image data, a natural solution would be a surface such that the images generated from the model are more similar to the observed images (i.e. the data). This naturally leads to formulate the problem as the minimization of an error measure between the observed and predicted values of pixels, carried out over all pixels in all input images. It corresponds to a Bayesian vision of the problem. This thus brings us to minimize an energy of the form [56, 140]:

$$E(\mathcal{S}) = \int_{\mathcal{I}} g\big(\,\pi_{\mathcal{S}}^{-1}(\mathbf{p}),\, \mathbf{n}(\pi_{\mathcal{S}}^{-1}(\mathbf{p}))\,\big)\, \mathrm{d}\mathbf{p}\;, \tag{II.2}$$

where $\mathcal{I}$ is the set on which data is defined (the set of all pixels in the image, in the case of the 3D reconstruction problems from images), and $\pi_{\mathcal{S}}^{-1}(\mathbf{p})$ is the point of the surface corresponding to the pixel $\mathbf{p}$ (the surface point which is viewed in pixel $\mathbf{p}$, in the 3D reconstruction problems from images). In the case where $\pi_{\mathcal{S}}^{-1}(\mathbf{p})$ does not reproject onto the surface $\mathcal{S}$, $\pi_{\mathcal{S}}^{-1}(\mathbf{p})$ is a point on the background B. $\mathrm{d}\mathbf{p}$ is the area measure on the image. g gives the error measure for the data point $\mathbf{p}$. Such functionals are generally called reprojection errors. One of the major properties in image formation, also one of the major problems in computer vision, is that only visible (i.e. unoccluded) elements are present in the image. Functionals (II.2) can then be rewritten as an integral over the surface (instead of the image) by counting only the visible points [130, 140, 160]. This gives, by a simple change of variables:

$$E(\mathcal{S}) = \int_{\mathcal{S}\cup B} \overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\; \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\;. \tag{II.3}$$

This involves adapting the measure on the surface [140] and the expression of $\overline{\mathbf{g}}$ which directly depends on the projection model being used. In most cases, e.g. for orthographic, perspective or linear pushbroom cameras, we have:

$$\overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}) = -g(\mathbf{x}, \mathbf{n}) k(\mathbf{x}) \mathbf{d}(\mathbf{x})\;,$$

where $k(\mathbf{x})$ is a specific scalar function of $\mathbf{x}$ and $\mathbf{d}(\mathbf{x})$ is the projection vector of $\mathbf{x}$ according to the camera. For example, for a perfect pinhole camera model, the adequate $\overline{\mathbf{g}}$ is given by $\overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}) = -g(\mathbf{x}, \mathbf{n})\,\frac{1}{\mathbf{x}_z^3}\mathbf{x}$, see [130, 140, 160]. For a linear pushbroom camera, we have $\overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}) = -g(\mathbf{x}, \mathbf{n})\,\frac{1}{v\cdot\mathbf{d}(\mathbf{x})\,\mathrm{d}_z(\mathbf{x})}\mathbf{d}(\mathbf{x})$, where $\mathbf{d}(\mathbf{x})$ is the vector joining $\mathbf{x}$ and the optical center of the sensor at corresponding time; v is a vector depending on the speed of the satellite, see [64]. In Equation (II.3), B is the surface behind $\mathcal{S}$ that corresponds to the background (i.e. the points on the data set which do not correspond to any point of the surface model of the object of interest). $\nu_{\mathcal{S}}(\mathbf{x})$ is the visibility function $\nu_{\mathcal{S}} : \mathbb{R}^3 \mapsto [0, 1]$ such that:

$$\nu_{\mathcal{S}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is visible from the camera,} \\ 0 & \text{otherwise.} \end{cases} \tag{II.4}$$

Finally, by using the separation technique proposed by [160], we can rewrite energy (II.3)

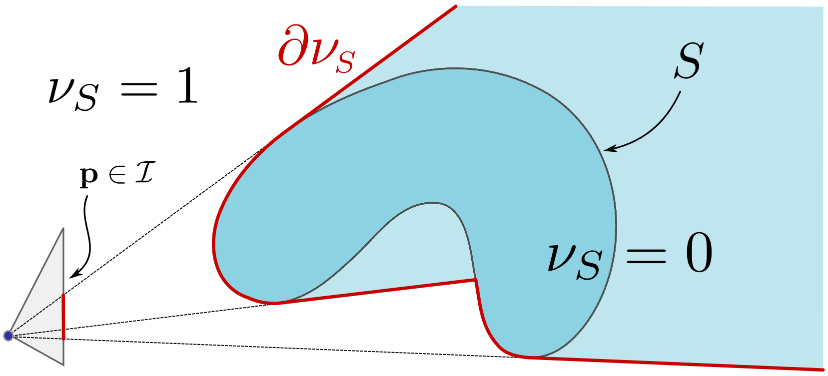

Figure II.1: *Banana Shape* seen from a vantage point (See [55]). The energy defined over the image explains the visibility interface $\partial \nu_S$ (in red) of the surface $\mathcal{S}$.

as an integral over only the visible surface:

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) \, \nu_{\mathcal{S}}(\mathbf{x}) \, \mathrm{d}s \, . \tag{II.5}$$

To obtain g from $\overline{\mathrm{g}}$, we refer the reader to Section III.1.2 and especially Equation (III.4), as well as [160], which comprehensively details this step. Figure II.1 illustrates the case of an energy defined over a visible volume.

Let us emphasize that handling properly the visibility term is a non-trivial undertaking. In particular, this is one of the major difficulties in the stereo-vision problem. Previous works cope with this difficulty more or less elegantly. Most often the authors approximate the visibility in pre-processing steps which can be completely prior to the whole algorithm or else inside the iterations of the minimization process [9, 42, 68, 81, 130, 140, 165].

Only recently, some authors [56, 160] manage to rigorously and fully account for visibility in the optimization process. In their recent work, Yezzi and Soatto [160] (for convex surfaces) and Gargallo et al. [56] (in the general case) provide the exact gradient of the reprojection error. This computation is done in the mathematical framework of continuous (smooth) surfaces as used in [141]. This work has shown that a proper handling of the visibility automatically forces the apparent contours generated by the reconstructed surface to perfectly match with the apparent contours in the observed data. This makes the use of additional energy terms like ballooning [153], visual hull [66, 138] or contours unnecessary and significantly reduces the minimal surface bias present in many other problem formulations.

In this chapter, we rigorously minimize energies (II.1) and (II.5). But contrary to the works [56, 160] which deal with continuous surfaces, here we consider triangle mesh-based surfaces.

II.1.2 Gradient Descent Optimization

Energy (II.5) is rather difficult to optimize. In particular, the complexity is due to the dependency on the normal and the visibility term. There exist several tools to minimize energy functionals. Recent methods, such as graph-cuts for example, allow us to find global minimum. At the present time these global optimization techniques are limited to rather simple energies [93] and, based on the current state-of-the-art, it seems extremely difficult to apply them for minimizing energies such as (II.5). Recent advances also allow such minimization via total variation and convex relaxation methods [28, 89, 91], however it is difficult to apply those methods for functionals depending on the surface normal. The recent work of [92] allows to take the normal into account but is not directly applicable to general energy functionals like (II.1) and (II.5). Taking the normal into account in the minimization allows to produce high quality 3D models [118]. In this chapter we compute the derivatives of the generic functionals (II.1) and (II.5) which allow us to minimize it via gradient descents [24, 42, 58, 141], see Section II.2.

II.1.3 Triangle Mesh-based Representation

Surface representation based on polyhedral and especially triangular meshes are the most commonly used in graphics. Moreover, the design of graphics hardware makes triangular meshes the most natural way to represent surfaces in a number of applications. In the variational framework, this type of discrete representations leads to the Lagrangian setting. Lagrangian methods are generally contrasted with Eulerian approaches which are mainly based on the level set representation [123]. During the last decades, Eulerian methods have become very popular mainly because they allow us to naturally deal with topological changes. In particular, this last setup has been extensively used for 3D reconstruction problems [23, 42, 56, 81, 82, 90, 96, 131, 143, 142, 157, 163]. Nevertheless, recent advances in mesh processing allow Lagrangian methods to enjoy the same facilities [129, 167]. In other respects, in the Eulerian methods, the gradient is computed in the continuous framework. Technically, computing the gradient in the continuous framework is more complicated than in a discrete framework, since the first one requires functional analysis when the second one only needs differential calculus. Furthermore, in practice in Eulerian approaches, one finally needs to discretize the continuous gradient flow since the level set function is also discretized on a grid. Also, this discretization (which is usually obtained using discrete differential operators [108] or finite element modeling [38]) is sometimes difficult to obtain, as we can see for example in Sections II.3 and II.4. By directly considering discrete surfaces, this last step is not necessary in our case.

Now, let us note that Lagrangian methods may be classified into two different approaches: The first strategy consists in 1) formulating the problem with continuous surfaces and computing the gradient in the continuous framework, then 2) discretizing the continuous gradient in order to apply it to the discrete surface. The second strategy consists in 1) formulating directly the problem with the discrete surface representation, then 2) computing the exact gradient of the formulated energy. Clearly, the second strategy is better than the first one: in fact in the first strategy we do not know if finally in practice the discrete representation minimizes something; in the best case, we do not know which exact energy the computed solution really minimizes since it does not necessarily corresponds to the representation. Also, surprisingly, a number of works follow the first strategy, see for example [9, 36, 167]. In particular, the gradient is computed using normal velocities whereas the second approach may have tangential components which leads to more coherent flow. In this chapter, we follow the second strategy as other authors: for example, Slabaugh and Unal [139] who deal with surface segmentation, Eckstein et al. [39] who are interested in shape matching, or Vu et al. [154] who proposes a complete multi-view stereo-vision algorithm. We also use the similar strategy presented by Debreuve et al. [30] who deal with discrete parametric active contours for segmentation or Dziuk and Elliott [38] using finite element modeling on evolving surfaces. More exactly, we detail the exact gradient flow of energy (II.1) in which the surface $\mathcal{S}$ is explicitly a discrete surface based on triangular meshes, and also extend it to visibility-driven energies (II.5).

II.1.4 Contributions

In this chapter, we first give an overview of gradient descent flows with deforming surfaces, when represented by triangular surface meshes. Here, the gradient is the one of the energy defined with the discrete surface; we do not need to approximate and discretize it when we finally evolve the surface. Even though the visibility plays a key role in computer vision, until now it has been managed more or less elegantly. It is clearly a key difficulty in the field, which has been recently solved in the theory of continuous surfaces [55, 56]. Here, we show how to rigorously deal with the visibility in the framework of discrete surface representations and we give the gradient flow of generic energies which encompasses a large class of energies used in computer vision (see Section II.4).

We then illustrate the presented results by giving the gradient of commonly used functionals in computer vision and graphics, and we emphasize the 3D reconstruction applications for which the visibility is fundamental; we thus show in Chapter III how one can apply mesh evolution techniques to 3D reconstruction applications from multiple views. In particular in Section III.1 we focus on the multi-view stereo-vision problem and we propose a successful algorithm which, since it fully accounts for visibility, automatically aligns contour generators with image contours.

This chapter generalizes our previous conference paper [32] in which the considered energy does not depend on the normal. This allows us to present here a larger spectrum of applications including multi-view shape from shading and multi-view photometric stereo (see Section III.3).

II.2 Gradient Flows

In this section, we are interested in minimizing energy functionals defined on surfaces with respect to the surface representation. Whatever surface representation one chooses, the energy minimization should be consistent with it in order to be sure the energy minimized is the right one. It has been common to minimize such energy by performing gradient descent. Computing an adequate gradient corresponding to the representation of the surface is not trivial, and the following gives a way for computing generalized gradient flows for an energy

$E(\mathcal{S})$, where $\mathcal{S}$ is a 2-dimensional surface in $\mathbb{R}^3$, g is a differentiable scalar function and $\mathbf{n}$ is the Gauss map for $\mathcal{S}$.

In the following, we explain how to obtain such gradients firstly in the theory of continuous smooth surfaces, and finally using triangular meshes.

II.2.1 Gradient Descent in the Continuous Case

Let M denote the set of all admissible 2D-manifolds embedded in $\mathbb{R}^3$, and $\mathcal{S} \in M$. Let $\mathbf{v}$ be a vector in the tangent space of M, denoted by $T_F M$, associated with an inner product $\langle \cdot, \cdot \rangle_F$. Let $E(\mathcal{S}) : M \to \mathbb{R}$ be a surface functional as defined previously (II.1) such that its Gâteaux Derivative in the direction $\mathbf{v}$ can be expressed as : $DE(\mathcal{S}, \mathbf{v}) \equiv \frac{\mathrm{d}}{\mathrm{d}\tau} E(S + \tau \mathbf{v}) \Big|_{\tau=0}$. Then the gradient of E at $\mathcal{S}$ is the unique vector $\nabla_M E(\mathcal{S}) \in T_F M$ such that $DE(\mathcal{S}, \mathbf{v}) = \langle \nabla_M E(\mathcal{S}), \mathbf{v} \rangle_F$ for all $\mathbf{v} \in T_F M$. See for instance Solem and Overgaard [141] for a more detailed explanation.

Then, the gradient descent flow of an energy $E(\mathcal{S})$ as the form:

$$\begin{cases} \mathcal{S}(0) = \mathcal{S}^0 \, , \\ \dfrac{\partial \mathcal{S}(t)}{\partial t} = -\nabla_M E(\mathcal{S}(t)) \, . \end{cases} \tag{II.6}$$

II.2.2 Gradient Descent for Polyhedral Meshes

In practice, we often deal with discrete representations of the surface. Also whichever this representation is, computing exactly the gradient of the energy including directly the discrete representation of the surface is much more suitable than computing the gradient in an ideal continuous framework (with continuous surfaces) and then discretizing the continuous gradient accordingly to the discrete representation. In fact, the second strategy has two significant drawbacks. First, the continuous gradient lets often appear terms that are very difficult to discretize and which sometimes do even not really make sense in the case of discrete surfaces. For example the notion of surface curvature on mesh representation has a lot of different approximations. Second, since the discrete object we are practically handling is not deformed following the exact gradient but by an approximation of it, finally, we do not exactly know what we are minimizing. In this chapter, we then compute the exact gradient of the energy including directly the discrete representation of the surface in the same way as [38, 39].

Let the mesh $\mathbf{X} = \{\mathbf{x}_1 \ldots \mathbf{x}_n\}$ be the piecewise planar polyhedral representation of $\mathcal{S}$. Vertices of $\mathbf{X}$ are denoted by $\mathbf{x}_k$ and $\mathcal{S}$ is deformed by moving vertices $\mathbf{x}_k$. We denote by $\phi_k : \mathcal{S} \to \mathbb{R}$ the piecewise linear, interpolating basis function such that $\phi_k(\mathbf{x}_k) = 1$ and $\phi_k(\mathbf{x}_i) = 0$ if $i \neq k$. Then any point $\mathbf{x}$ on the surface $\mathcal{S}$ can be defined such that $\forall \mathbf{x} \in \mathcal{S}$, $\mathbf{x} = \sum_k \mathbf{x}_k \phi_k(\mathbf{x})$, where we also have $\forall \mathbf{x} \in \mathcal{S}$, $\sum_k \phi_k(\mathbf{x}) = 1$.

Let the set $\{\mathbf{V}_k\}$ be a parametrized vector field defined on all the vertices $\mathbf{x}_k$ of the mesh $\mathbf{X}$ representing the surface deformation. $\{\mathbf{V}_k\}$ can be naturally extended on $\mathcal{S}$ by a piecewise linear vector field on $\mathcal{S}$. For convenience, we denote this extension $\mathbf{V} : \mathbf{V}(\mathbf{x}) = \sum_k \mathbf{V}_k \phi_k(\mathbf{x})$.

Then, evolving $\mathbf{X}$, by moving its vertices $\mathbf{x}_k$ following $\mathbf{V}_k$ is equivalent to deform the surface $\mathcal{S}$ by the dense deformation following $\mathbf{V}$.

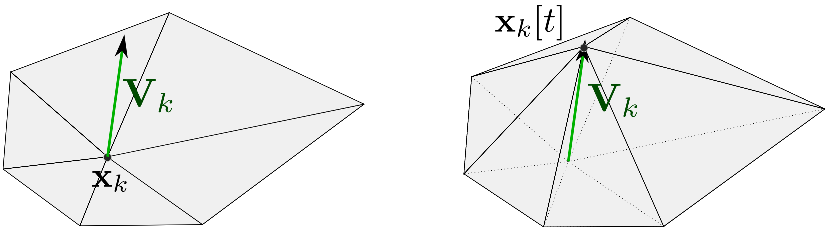

Figure II.2: Local evolution of a surface point $\mathbf{x}_k$ under induced velocity $\mathbf{V}_k$ used to compute the gradient.

The gradient of the energy is computed using shape gradient [30]. We consider the evolution of this energy according to the deformation $\mathbf{V}$. In other words, we assume that the vertices $\mathbf{x}_k[t]$ of $\mathbf{X}[t]$ are moving according to $\mathbf{x}_k[t] = \mathbf{x}_k^0 + t\mathbf{V}_k$ (See Figure II.2). The method for computing the gradient of $E(\mathcal{S})$ consists in computing the directional derivative of $E(\mathcal{S}[t])$ for this deformation and then in rewriting it as a scalar product of $\mathbf{V}$, i.e. as $\langle \mathbf{V}, \mathbf{G} \rangle$, $\mathbf{G}$ being independent of $\mathbf{V}$. The obtained vector $\mathbf{G}$ is called the *gradient* and the energy necessarily decreases when deforming the surface according to its opposite direction $-\mathbf{G}$. Indeed, for $\mathbf{x}_k[t] = \mathbf{x}_k^0 - t\mathbf{G}_k$, we have

$$[E \circ \mathcal{S}]'(0) = -\langle \mathbf{G}, \mathbf{G} \rangle \ \leq 0,$$

see [39].

Let us recall now that, as underlined by [24, 39], the notion of gradient depends on the underlying scalar product. In this work we will only consider the L^2 inner product which has the advantage of taking into account the area of the triangles contrarily to the pointwise scalar product, which is necessary if the surface is not a regular mesh. Let $A = \{\mathbf{a}_k\}$ and $B = \{\mathbf{b}_k\}$ be vector fields on the mesh $\mathbf{X}$. Let a and b be their linear extension on the whole surface $\mathcal{S}$. Then their L^2 scalar product is:

$$\begin{aligned} \langle A, B \rangle_{L^2} &= \int_{\mathcal{S}} \langle a(\mathbf{x}), b(\mathbf{x}) \rangle \, \mathrm{d}s \\ &= \int_{\mathcal{S}} \left\langle \sum_k \mathbf{a}_k \phi_k(\mathbf{x}), \sum_k \mathbf{b}_k \phi_k(\mathbf{x}) \right\rangle \mathrm{d}s \\ &= A^\top \, M \, B \,, \end{aligned} \tag{II.7}$$

where $M = \{m_{ij}\}$ is the mass matrix defined by $m_{ij} = Id_3 \int_{\mathcal{S}} \phi_i(x)\phi_j(x)\mathrm{d}s$. In the last line of Equation (II.7), A and B are the matricial representations of the vector fields. They are column vectors containing successively $\mathbf{a}_k$ and $\mathbf{b}_k$ vectors. Then the gradient becomes:

$$\nabla E(\mathbf{X}) = M^{-1} \frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}) \,, \tag{II.8}$$

where

$$\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}) = \left[\frac{\partial E}{\partial \mathbf{x}_1}(\mathbf{X})\ \frac{\partial E}{\partial \mathbf{x}_2}(\mathbf{X}) \dots \frac{\partial E}{\partial \mathbf{x}_n}(\mathbf{X})\right]'$$

corresponds to the gradient associated with the pointwise inner product $< A, B >= \sum_k \mathbf{a}_k \cdot \mathbf{b}_k$.

One classically approximates M by the diagonal mass lumping $\tilde{M}$, where $\tilde{m}_{ii}$ is the area of the Voronoi dual cell of $\mathbf{x}_i$ times the identity matrix Id_3, see e.g. [39]. It follows that the L^2 gradient descent flow is:

$$\begin{cases} \mathbf{X}[0] = \mathbf{X}^0\,, \\ \dfrac{\partial \mathbf{X}[t]}{\partial t} = -\tilde{M}^{-1}\dfrac{\partial E}{\partial \mathbf{X}}(\mathbf{X}[t])\,. \end{cases} \tag{II.9}$$

II.2.2.1 Triangle Mesh Representation and Notations

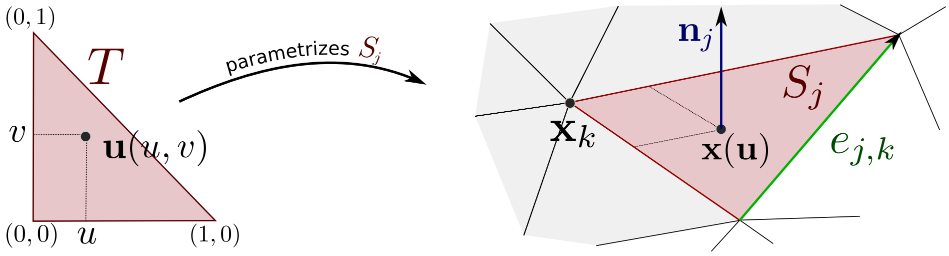

Figure II.3: Local parametrization of the discrete representation of the surface into a triangle mesh. Figure on the left shows the local parametrization $\mathbf{u}(u, v)$ of a surface point. On the right, we show the notations used in the thesis where $\mathbf{x}_k$ is the current vertex of $\mathcal{S}$, $\mathcal{S}_j$ is the current facet around $\mathbf{x}_k$ and $\mathbf{e}_{j,k}$ is the opposite edge of $\mathbf{x}_k$ in $\mathcal{S}_j$. The gray area represent the set of triangles around $\mathbf{x}_k$ denoted by $\mathcal{J}_k$.

The previous results are valid for any polyhedral representation. Also in practice and in the following, we will focus on triangle representation that are easier to understand and more simple to handle using computers.

Let $\mathcal{S}_j$ be the j^{th} triangle of the mesh and $\mathbf{x}_k$ be a vertex of $\mathcal{S}_j$. Let us consider the parametrization on the triangle $\mathcal{S}_j$ such that

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}_k + u\,\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} + v\,\overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\,, \tag{II.10}$$

where $\mathbf{x}_{k1}$ and $\mathbf{x}_{k2}$ are the two other vertices of the triangle $\mathcal{S}_j$ such that $(\mathbf{x}_k, \mathbf{x}_{k1}, \mathbf{x}_{k2})$ is a counter-clockwise triangle. Here, $\mathbf{u} = (u, v) \in T$, where

$$T = \{(u, v) | u \in [0, 1] \text{ and } v \in [0, 1 - u]\}\,.$$

Figure II.3 illustrates this representation and parametrization. Too be rigorous, we should write $\mathbf{x}_k^j(\mathbf{u})$ instead of $\mathbf{x}(\mathbf{u})$, since the parametrization depends on j and k. Nevertheless, in order to simplify equations and improve the clarity of the thesis, we remove these indexes in the rest of the thesis. In the following, when we use $\mathbf{x}(\mathbf{u})$, the choice of the associated j and k is directly given by the context. On each triangle $\mathcal{S}_j$, we denote by ϕ_k the linear interpolating basis function that verifies $\phi_k(\mathbf{x}(\mathbf{u})) = (1-u-v)\ ,\ \forall (u,v) \in T$.

We denote by A_j the surface area of triangle $\mathcal{S}_j$ and by $\mathbf{n}_j$ its outward surface normal. A_j and $\mathbf{n}_j$ can easily be defined with respect to the triangle vertices such that:

$$A_j = \frac{1}{2}\left|\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\right| \quad \text{and} \quad \mathbf{n}_j = \frac{\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}}{2\,A_j}\ ,$$

where the operator $\wedge$ denotes the cross product. Indeed it is easy to show that the area surface measure on the surface $\mathrm{d}s$ can be written using the parametrization $\mathbf{u}$ such that:

$$\mathrm{d}s = 2\,A_j\,\mathrm{d}\mathbf{u}\ . \tag{II.11}$$

In the following, we also denote by $\mathbf{e}_{j,k}$ the opposite edge of the vertex $\mathbf{x}_k$ in the triangle $\mathcal{S}_j$ such that $\mathbf{e}_{j,k} = \overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}$.

$\mathcal{J}_k$ represents the set of triangles containing vertex $\mathbf{x}_k$ and the set $\mathcal{K}_j$ is the set of indexes of the three vertices of triangle $\mathcal{S}_j$ (See Figure II.3.).

II.3 Gradient of Weighted Area Functionals

II.3.1 Continuous Case

Let $\mathcal{S} \in M$ be the surface to deform in order to minimize the following classical weighted area functional (II.1):

$$E(\mathcal{S}) = \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\mathrm{d}s\ ,$$

where g is a differentiable scalar function defined all over the surface. Then using shape gradient as described previously, one can rewrite the differential of $E(\mathcal{S})$ under a linear deformation V in order to find the expression of $\nabla E(\mathcal{S})$. As used in [56] and shown in [141], the gradient descent flow of the functional defined in (II.1) has the form:

$$\nabla E(\mathcal{S}) = \nabla \cdot (g_{\mathbf{n}} + g\mathbf{n})\ , \tag{II.12}$$

where $g_{\mathbf{n}}$ is the gradient on the unit sphere $\mathbb{S}$.

II.3.2 Gradient for Triangle Mesh-based Surfaces

In this paragraph, we consider the discretization on the surface and the gradient descent flow described in the previous section, when the surface is represented by a triangular mesh. Also we consider the case where the energy functional to be minimized is:

$$
\begin{aligned}
E(\mathcal{S}) &= \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\,\mathrm{d}s \\
&= \sum_j 2\,A_j \int_T g(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)\,\mathrm{d}\mathbf{u}\,,
\end{aligned}
\tag{II.13}
$$

where A_j and $\mathbf{n}_j$ are defined in Section II.2.2.1.

Let us focus on the evolution of $E(S)$ under the induced velocity $\mathbf{V}$ on triangle $\mathcal{S}_j$ only. We have

$$
E(\mathcal{S}_j[t]) = 2\,A_j[t] \int_T g(\mathbf{x}(\mathbf{u}) + t\mathbf{V}(\mathbf{x}(\mathbf{u})), \mathbf{n}_j[t])\,\mathrm{d}\mathbf{u}\,,
\tag{II.14}
$$

where $A_j[t]$ is the area of the triangle $\mathcal{S}_j[t]$ and $\mathbf{n}_j[t]$ is its normal at time t. By simple derivation we get

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}_j[t])\Big|_{t=0} = 2\,A'_j[0] &\int_T g(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}\mathbf{u} \\
+ 2\,A_j &\int_T \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j) \cdot \mathbf{V}(\mathbf{x})\,\mathrm{d}\mathbf{u} \\
+ 2\,A_j &\int_T \nabla_{\mathbf{n}} g(\mathbf{x}, \mathbf{n}_j) \cdot \mathbf{n}'_j[0]\,\mathrm{d}\mathbf{u}\,.
\end{aligned}
\tag{II.15}
$$

Above, in order to simplify equations, we have removed the dependency in $\mathbf{u}$ by writing $\mathbf{x}$ instead of $\mathbf{x}(\mathbf{u})$. In the sequel, we will use this abuse of notation. In order to rewrite $\frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}_j[t])\Big|_{t=0}$ as a scalar production of $\mathbf{V}$, we use also the fact that $\forall \mathbf{x} \in \mathcal{S}_j$, $\mathbf{V}(\mathbf{x}) = \sum_{k \in K_j} \mathbf{V}_k \phi_k(\mathbf{x})$, and then use the expressions of $A'_j[0]$ and $\mathbf{n}'_j[0]$ computed in appendixes A.2.1 and A.2.2:

$$
\begin{aligned}
A'_j[0] &= \frac{\mathrm{d}}{\mathrm{d}t} A_j[t]\Big|_{t=0} = \frac{1}{2}\left(\mathbf{n}_j \wedge \overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}\right) \cdot \mathbf{V}_k\,, \\
\mathbf{n}'_j[0] &= \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{n}_j[t]\Big|_{t=0} = \frac{\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k - \left(\left(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k\right) \cdot \mathbf{n}_j\right) \mathbf{n}_j}{2\,A_j}\,.
\end{aligned}
\tag{II.16}
$$

It follows that:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}_j[t])\Big|_{t=0} = \sum_{k \in K_j} \mathbf{V}_k \cdot \Bigg\{ &\frac{\mathbf{n}_j \wedge \mathbf{e}_{j,k}}{2A_j} \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}s \\
&+ \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j) \phi_k(\mathbf{x})\,\mathrm{d}s \\
&- \frac{\mathbf{e}_{j,k}}{2\,A_j} \wedge \int_{\mathcal{S}_j} g_{\mathbf{n}}(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}s \Bigg\}\,,
\end{aligned}
\tag{II.17}
$$

where we define $g_{\mathbf{n}} = \nabla_n g(\mathbf{x}, \mathbf{n}_j) - \langle \nabla_n g(\mathbf{x}, \mathbf{n}_j), \mathbf{n}_j \rangle\, \mathbf{n}_j$, where $\nabla_n g(\mathbf{x}, \mathbf{n}_j)$ is the gradient of g with respect to the second variable (i.e. $\mathbf{n} \in \mathbb{R}^3$).

It then immediately follows that

$$\frac{\mathrm{d}}{\mathrm{d}t}E(\mathcal{S}[t])\Big|_{t=0} = \sum_k \mathbf{V}_k \cdot \left[\sum_{j\in\mathcal{J}_k} \left\{ \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x},\mathbf{n}_j)\phi_k(\mathbf{x})\,\mathrm{d}s \right.\right. \\ \left.\left. -\frac{\mathbf{e}_{j,k}}{2A_j} \wedge \int_{\mathcal{S}_j} g(\mathbf{x},\mathbf{n}_j)\,\mathbf{n}_j + g_{\mathbf{n}}(\mathbf{x},\mathbf{n}_j)\,\mathrm{d}s \right\}\right], \tag{II.18}$$

Finally the part in brackets gives the k^{th} component of $\dfrac{\partial E}{\partial \mathbf{X}}$ and one has to use Equation (II.9) in order to get the gradient and optimize the mesh $\mathbf{X}$.

Discussion about the continuous gradient

By looking at both gradients, one may note similarities. First it is worth it to notice that the discrete gradient is written with respect to well defined quantities that can be easily expressed, such as edges vectors, or triangle area.

On the other hand the continuous gradient has terms depending on the curvature for instance, which is well defined in the theory of continuous differential surfaces, but is hard to discretize on mesh representations. By making a piecewise linear surface assumption, the obtained gradient directly accounts for those intrinsic properties.

II.3.3 Applications

II.3.3.1 Mean Curvature Flow

As described in the introduction, one of the most commonly used gradient flow is the mean curvature flow, that minimizes the surface area. This energy is often used to perform mesh smoothing, or more often as a smoothing energy term. The associated energy functional is simply

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathrm{d}s\,, \tag{II.19}$$

and its associated continuous gradient is $\nabla E = \kappa$, where κ is the surface's curvature.

However, one needs approximations in order to apply it to discrete surface representations since notion of discrete curvature is not clear. On triangular meshes, different approximated flows have been proposed like Laplacian approximation or the umbrella operator. Meyer et al [108] have computed the discrete gradient flow that minimizes Equation (II.19) with respect to the triangle mesh representation. It is easy to show that the same result can be obtained using the presented approach. Following Section II.2, we have

$$\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = \sum_k \mathbf{V}_k \cdot \sum_{j\in\mathcal{J}_k} \frac{1}{2}\mathbf{n}_j \wedge \mathbf{e}_{j,k}\,. \tag{II.20}$$

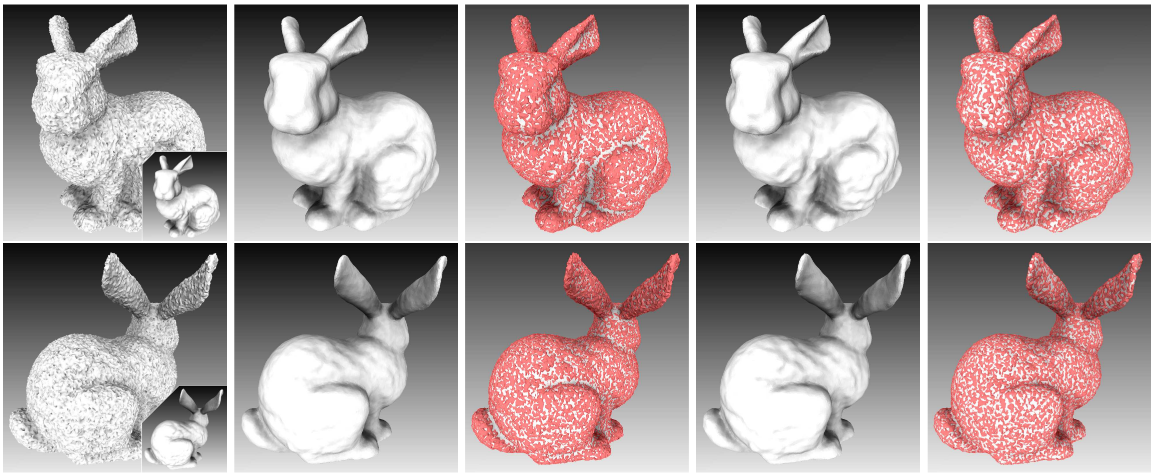

Figure II.4: Evolution of smoothing algorithms on the Stanford Bunny data. From left to right: input noisy data; denoised mesh obtained using the mean curvature flow (MCF) of Section II.3.3.1; MCF result versus the noisy input (in red); denoised mesh obtained using the normal smoothing algorithm (NS) of Section II.3.3.2; NS result versus the noisy input (in red). Smoothing the normals help preserving details better than the mean curvature flow and does not over-smooth the result at concavities and convexities as much as the mean curvature flow.

The evolution of one vertex k then follows

$$\frac{d\mathbf{x}_k}{dt} = \frac{1}{A_k} \sum_{j \in \mathcal{J}_k} \frac{1}{2}\, \mathbf{e}_{j,k} \wedge \mathbf{n}_j \ , \tag{II.21}$$

where A_k is the area of the neighborhood $\mathcal{J}_k$ of vertex k. Note that this result is exactly the same as the results in [108], but is just expressed differently. The given formulation can be useful for applications where edges and surface normals have been previously computed. Figure II.4 illustrates this algorithm on the Stanford Bunny data [1].

II.3.3.2 Normal Field Integration

One of the applications is to align a surface with respect to an external normal field. For instance, in 3D reconstruction, one recover the surface by integrating photometric normals [23, 68, 152]. Let $\mathbf{h}$ be a unit vector field in $\mathbb{R}^3$. Integrating this vector field $\mathbf{h}$ such that the surface normals $\mathbf{n}$ correspond to it involves minimizing the following functional

$$\begin{aligned} E(\mathcal{S}) &= \frac{1}{2} \int_{\mathcal{S}} |\mathbf{h} - \mathbf{n}|^2 \, \mathrm{d}s \\ &= \sum_j \int_{\mathcal{S}_j} (1 - \mathbf{h} \cdot \mathbf{n}_j) \ \mathrm{d}s \ . \end{aligned} \tag{II.22}$$

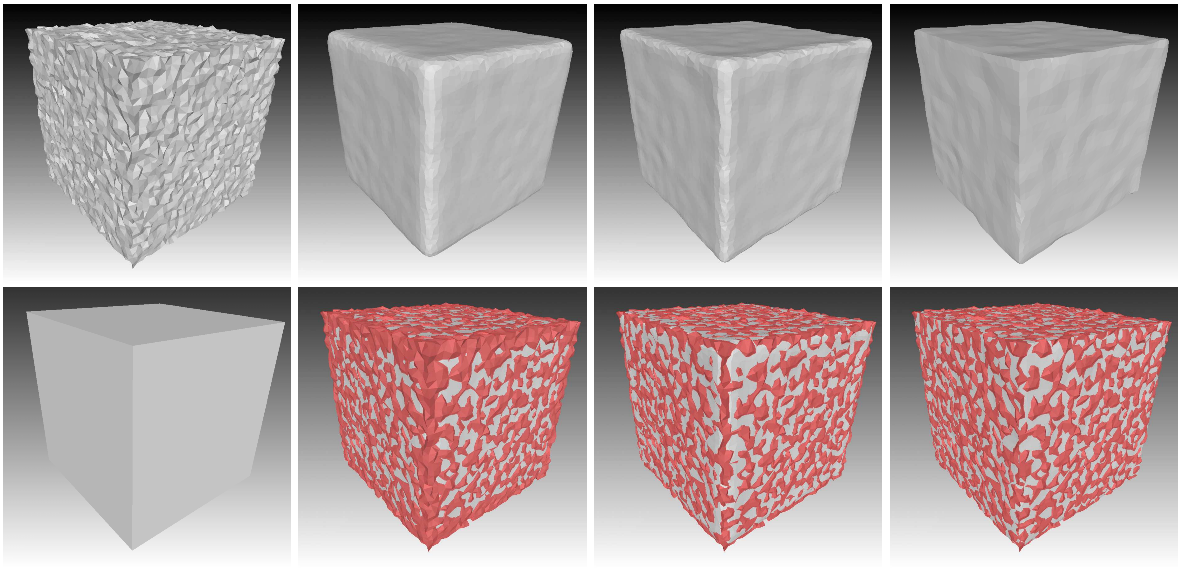

Figure II.5: Evolution of smoothing algorithms on simple cube data. From left to right: noisy input and corresponding original surface; mean curvature flow smoothing; normal smoothing; median filtering. The red part corresponds to the input noisy mesh being displayed together with the results.

As explained in Section II.3, it is easy to show that the following gradient descent flow is:

$$\frac{d\mathbf{x}_k}{dt} = \frac{1}{A_k} \sum_{j \in \mathcal{J}_k} \frac{1}{2}\, \mathbf{e}_{j,k} \wedge \int_T (\mathbf{n}_j - \mathbf{h})\, \mathrm{d}\mathbf{u}\,. \tag{II.23}$$

Normal Smoothing

As an example we show now that this approach can be easily applied to normal smoothing with efficiency, and is barely more time consuming than the mean curvature flow. Let $\mathbf{h}_j$ be the weighted average normal of the triangle j and its neighborhood N_j:

$$\mathbf{h}_j = \frac{\sum_{l \in N_j} \alpha_l\, \mathbf{n}_l}{|\sum_{l \in N_j} \alpha_l\, \mathbf{n}_l|}\,, \tag{II.24}$$

where the weights α_l can be chosen depending on the application. It could for instance be the area A_l of the triangle l, a Gaussian weight or more simply one can set $\alpha_l = 1$. Therefore the energy functional to be minimized can be expressed as:

$$E(\mathcal{S}) = \sum_j \int_{\mathcal{S}_j} (1 - \mathbf{h}_j \cdot \mathbf{n}_j)\ \mathrm{d}s\,, \tag{II.25}$$

where $\mathbf{h}_j$ is constant over the surface. Note that we cannot directly apply the results presented in Section II.3.2 to energy (II.25) since it depends on the normal at several surface points.

To minimize energy (II.25), we consider $\mathbf{h}_j$ to be fixed while updating the surface. In other words, we alternately update the surface and the vector field $\mathbf{h}$. The gradient descent flow corresponding to the normal smoothing energy with respect to the surface is

$$\frac{d\mathbf{x}_k}{dt} = \frac{1}{A_k} \sum_{j \in \mathcal{J}_k} \frac{1}{2}\, \mathbf{e}_{j,k} \wedge (\mathbf{n}_j - \mathbf{h}_j)\ . \tag{II.26}$$

Note the similarity of the above equation with the mean curvature flow (Equation (II.21)), and the fact that the gradient flow with respect to the surface is barely more time consuming. Results are shown in Figure II.4, where the evolution is stopped once the noise is no longer visible. The figure also shows a comparison between the mean curvature flow. In particular, when displayed with the input noisy mesh, one may notice the different density of noise in the mesh (shown in red). Since this is a Gaussian noise, the quantity of noise should be uniform all over the surface, as it is almost the case for normal smoothing. However, proportion of the visible noisy mesh in the mean curvature flow example is much denser in the convex parts and disappears in concavities. While mean curvature flow is popular for surface regularization and smoothing, normal smoothing preserve details better and does not over-smooth as much as the mean curvature flow.

Median Filtering

Similarly, one may choose the external vector field $\mathbf{h}_j$ to be the median vector of the neighborhood N_j of $\mathcal{S}_j$ to perform median filtering on the mesh. $\mathbf{h}_j$ can be computed by first computing the spherical coordinates (θ_m, ϕ_m) of each normal $\mathbf{n}_m \in N_j$. Then by sorting the spherical coordinates (θ_m, ϕ_m) along each component, one can obtain the median normal by getting back to Cartesian coordinates. Results on a noisy cube and comparison with the previous smoothing algorithms are shown in Figure II.5.

II.4 Gradient of Functionals defined on Visible Surface

As described in the introduction, computer vision applications are most likely able to deal with what the camera sees, i.e. the projected image. Such projection imply $3D$ information such as depth and occlusion as well as a camera model. An energy functional can be expressed accordingly as in Equation (II.5), where the energy is defined on the visible interface (See also Figure II.1). Often some quantities can be estimated (it can be for instance color, photometric normals, reflectance, etc) and used to compare them with data in the input images. In the sequel we denote such energies reprojection error functionals. As shown for example in [56, 159], an accurate Bayesian formulation of computer vision problems (as e.g. in 3D reconstruction) yields to the minimization of reprojection error functionals instead of classical weighted area functionals.

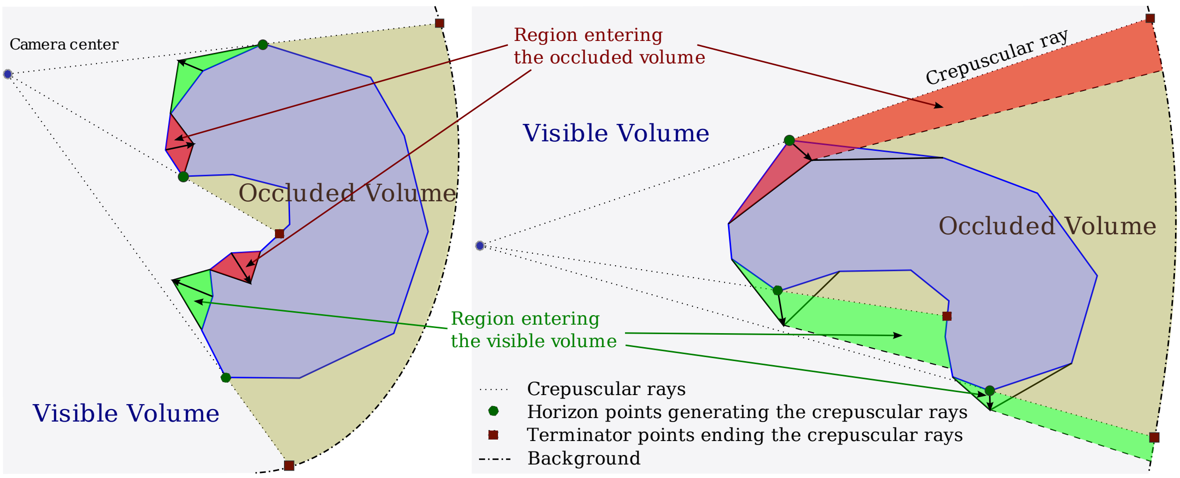

Figure II.6: Geometric representation of the change of visibility when moving the mesh. Contrary to the interior (left), movements of the horizon (right) strongly affect the movement of the visible interface between visible and occluded volumes by creating a movement of the crepuscular rays.

Let us then consider the energy functional (II.5)

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\ .$$

In this section, we compute the gradient of this functional with respect to the shape $\mathcal{S}$.

In practice, the direction of the 3D vector $\mathbf{g}$ corresponds to the direction of the viewing ray. Thus the reprojection error functionals generally verify $\mathbf{g}(\mathbf{x}, \mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})) \cdot \mathbf{n}_{\partial\mathcal{V}}(x) = 0$ for all points $\mathbf{x}$ on the horizon of the visibility interface ($\partial\mathcal{V}$ being the visibility interface and $\mathbf{n}_{\partial\mathcal{V}}$ its normal, see Figure II.1). Then, by Gauss' divergence theorem, we can rewrite $E(\mathcal{S})$ as an integral over $\mathbb{R}^3$, see [55, 56].

$$\begin{aligned} E(\mathcal{S}) &= \int_{\partial\mathcal{V}} \mathbf{g}(\mathbf{x}, \mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})) \cdot \mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})\, \mathrm{d}s \\ &= -\int_{\mathbb{R}^3} \nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x}\ . \end{aligned} \tag{II.27}$$

Let $\mathcal{S}[t]$ be a variation of $\mathcal{S}$ such that $\mathcal{S}[t] = \mathcal{S} + t\,V$. By the product rule, the derivative of the energy with respect to t is:

$$\begin{aligned} \frac{d}{dt} E(\mathcal{S}[t])\Big|_{t=0} = &-\int_{\mathbb{R}^3} \nabla \cdot \frac{d}{dt}\Big(\mathbf{g}(\mathbf{x}, \mathbf{n}[t](\mathbf{x}))\Big)\Big|_{t=0} \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x} \\ &-\int_{\mathbb{R}^3} \nabla \cdot \Big(\mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))\Big)\, \frac{d}{dt} \nu_{\mathcal{S}[t]}(\mathbf{x})\Big|_{t=0} \mathrm{d}\mathbf{x}\ . \end{aligned} \tag{II.28}$$

For notation simplicity we have denoted $\mathcal{S} = \mathcal{S}[0]$, $\mathbf{n} = \mathbf{n}[0]$.

In Section II.4.2.1, we write the first integral of Equation (II.28) linearly with respect to $\mathbf{V}$. We then describe the corresponding part of the gradient of $E(\mathcal{S})$. We denote it $\mathbf{G}^{norm}$.
In the second integral of Equation (II.28), the normal does not depend on t. Thus, this term is the derivative of a quantity (which does not depend on $\mathcal{S}$ and does not vary in time) integrated over the visible volume. In other words, the second term of Equation (II.28) is the derivative (with respect to the shape $\mathcal{S}$) of an energy:

$$\tilde{E}(\mathcal{S}) = \int_{\mathbb{R}^3} f(\mathbf{x}) \nu_{\mathcal{S}}(\mathbf{x}) \, \mathrm{d}\mathbf{x} \,,$$

where $f(\mathbf{x})$ does not depend on $\mathcal{S}$;

$$f(\mathbf{x}) = -\nabla \cdot \Big(\mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \Big).$$

It is the derivative of a scalar field integrated over a visible volume.

Until now all equations are valid for both the continuous case and the discrete case. In order to find the gradient expression, the idea is to see what happens to the energy under an induced velocity on the surface. In the continuous case under a continuous vectorial field V, and under a piecewise linear vectorial field $\mathbf{V}$ defined by the discrete field $\{V_k\}$ in the other case. In both cases we have the following geometric interpretation of the changes in the energy when the surface is deforming.

Since this energy $\tilde{E}(\mathcal{S})$ is an integral over the visible volume, its variations are only due to the variations of the visible volume. Also, as illustrated by Figure II.6 (reduced to the 2D case for simplicity), when the vertex $\mathbf{x}_k$ is moving according to $\mathbf{V}_k$ we have to separate two cases:

1. when all the triangles adjacent to $\mathbf{x}_k$ are visible, the variation of the visible volume is just the sum of the tetrahedron formed by the adjacent triangles and the moved point $\mathbf{x}_k + \mathbf{V}_k$ (see Figure II.6, left). The corresponding gradient computation is detailed in Section II.4.2.2. By replacing $f(\mathbf{x})$ by $-\nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))$, this gives a second part of the gradient of $E(\mathcal{S})$; we denote it $\mathbf{G}^{int}$.

2. when $\mathbf{x}_k$ is generating occluding contours in images, i.e. when it is a *horizon* point, its movement affects the visibility of other points located behind it (called *terminator* points). So the variation of the visible volume is the sum of the first case term, plus the volume swept out by the crepuscular rays generated by the horizon movement (see Figure II.6, right). The corresponding gradient computation is detailed in Section II.4.2.3. Again, by replacing $f(\mathbf{x})$ by $-\nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))$, this gives a third part of the gradient of $E(\mathcal{S})$ that we denote it $\mathbf{G}^{horiz}$.

Now let us first remind the expression of the gradient in the continuous case (Section II.4.1) and then compute it when using triangle meshes (Section II.4.2).

II.4.1 Gradient in the Continuous Case

The differential of the energy (II.5) in the case of continuous surfaces is the work of Gargallo [55] and is:

$$\nabla \cdot \Big(\mathbf{g}_{\mathbf{n}} \cdot \mathbf{n} + \mathbf{g} \Big) \nu_{\mathcal{S}} - \mathbf{x}^t \nabla \mathbf{n} \, (\mathbf{g} - \mathbf{g}') \delta(\mathbf{x} \cdot \mathbf{n}) \nu_{\mathcal{S}} \, , \tag{II.29}$$

where δ is a Dirac distribution function and $\mathbf{g}'$ is the value of $\mathbf{g}$ at the terminator of the current point. In the presence of a discrete surface, terms like the curvature $\nabla \mathbf{n}$ are difficult to handle and have to be approximated. The following section shows that by computing the gradient with respect to the true representation of the surface, one can use a new formulation using intrinsic surface properties.

II.4.2 Gradient for Triangle Meshes

Here, we derive the exact gradient (with respect to the shape) of these functionals in the case where the surface is represented by a triangular mesh. Here, contrary to our previous conference article [32], the function $\mathbf{g}$ may also depends on the normal of the surface.

Then we can summarize the way gradient descent is performed in that case. The variation of an energy that depends on the visibility and the surface normal can be decomposed into three different cases. The three terms are:

1. the one due to the change in the normal $\mathbf{G}^{norm}$ (Equation (II.33)) that corresponds to Section II.4.2.1,

2. the term due to the movement of points on the fully visible areas $\mathbf{G}^{int}$ (Equation (II.38)) of Section II.4.2.2,

3. and finally the term due to the movement of points on occluding contours $\mathbf{G}^{horiz}$ (Equation (II.44)) that makes global (in the sense on the whole surface) changes of the energy (Described in Section II.4.2.3).

II.4.2.1 Term due to the Variation of the Normal

In this section, we are going to rewrite the first integral term appearing in equation (II.28), linearly with respect to $\mathbf{V}$. This will directly give us the first term of the gradient of $E(\mathcal{S})$ defined in (II.5). First, using Gauss' divergence theorem again, let us rewrite it as an integral over the surface. This gives:

$$\begin{aligned}
-\int_{\mathbb{R}^3} \nabla \cdot \frac{d}{dt}\mathbf{g}(\mathbf{x}, \mathbf{n}[t](\mathbf{x}))\Big|_{t=0} \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x} & \\
&= \int_{\mathcal{S}} \frac{d}{dt}\mathbf{g}(\mathbf{x}, \mathbf{n}[t](\mathbf{x}))\Big|_{t=0} \cdot \mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s \\
&= \sum_j \int_{\mathcal{S}_j} \frac{d}{dt}\mathbf{g}(\mathbf{x}, \mathbf{n}_j[t])\Big|_{t=0} \cdot \mathbf{n}_j\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s \\
&= \sum_j \int_{\mathcal{S}_j} (D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)\mathbf{n}'_j[0]) \cdot \mathbf{n}_j\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\ ,
\end{aligned} \tag{II.30}$$

where $D_{\mathbf{n}}$ is the differential with respect to the second variable and where $\mathbf{n}'_j[0]$ is the derivative of the normal $\mathbf{n}_j[t]$ of the triangle $\mathcal{S}_j[t]$ at time $t = 0$. Here we have assumed that the reprojection error functional verifies

$$\left(\frac{d}{dt}\mathbf{g}(\mathbf{x}, \mathbf{n}_{\partial\mathcal{V}}[t](\mathbf{x}))\right) \cdot \mathbf{n}_{\partial\mathcal{V}}(x) = 0$$

for all points $\mathbf{x}$ on the horizon lines of the visibility interface; it is generally the case in practice, see [55, 56].

The computation of $\mathbf{n}'_j[0]$ is detailed in appendix A.2.2. We have

$$\mathbf{n}'_j[0] = \frac{1}{2A_j}\left(\left(\sum_{k\in K_j} \mathbf{e}_{j,k} \wedge \mathbf{V}_k\right) - \left((\sum_{k\in K_j} \mathbf{e}_{j,k} \wedge \mathbf{V}_k) \cdot \mathbf{n}_j\right)\mathbf{n}_j\right). \tag{II.31}$$

So

$$\begin{aligned}
\int_{\mathbb{R}^3} \nabla \cdot \frac{d}{dt}\mathbf{g}(\mathbf{x}, \mathbf{n}[t](\mathbf{x}))\Big|_{t=0} \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x} & \\
&= \sum_j \int_{\mathcal{S}_j} \mathbf{n}'_j[0] \cdot D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j\, \nu_{\mathcal{S}}(\mathbf{x}) \mathrm{d}s \\
&= \sum_k \mathbf{V}_k \cdot \Big[\sum_{j\in J_k} (-1)\frac{\mathbf{e}_{j,k}}{2\,A_j} \wedge \\
&\qquad \int_{\mathcal{S}_j} P_{n_j^\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j)\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s \Big]\ ,
\end{aligned} \tag{II.32}$$

where $P_{n_j^\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j)$ is the projection on the orthogonal plane to $\mathbf{n}_j$ of $D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j$. The edge $\mathbf{e}_{j,k}$ is defined in appendix A.2.1. Roughly $\mathbf{e}_{j,k}$ is the opposite edge of the vertex $\mathbf{x}_k$ in the triangle $\mathcal{S}_j$. Finally, writing the previous expression on the mesh parametrization $\mathbf{x}(\mathbf{u})$

yields: which can be rewritten as :

$$\mathbf{G}_k^{norm} = \sum_{j\in\mathcal{J}_k} \mathbf{e}_{j,k} \;\wedge\; \int_T P_{n_j^\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}(\mathbf{u}),\mathbf{n}_j)^\top \mathbf{n}_j)\; \nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u}))\,\mathrm{d}\mathbf{u}. \tag{II.33}$$

II.4.2.2 Term Due to the Tetrahedra of the Visible Adjacent Triangles

In this paragraph, let us focus on the variation of the energy caused by the variation of the visible volume corresponding to the tetrahedra formed by the visible adjacent triangles of vertex $\mathbf{x}_k$ and $\mathbf{x}_k + \mathbf{V}_k$. (See Figure II.2)

In the sequel, we denote $\mathcal{S}_j$ the j^{th} triangle of the mesh. Following the variation of the energy caused by the visible adjacent triangles of $\mathbf{x}_k$, we have

$$\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0]) = \sum_j \varpi_{j,\mathbf{V}_k} \int_{Vol[j,\mathbf{V},t]} f(\mathbf{x})\,\mathrm{d}\mathbf{x} \;+ \ldots\;, \tag{II.34}$$

where $Vol[j,\mathbf{V},t]$ is the volume of the tetrahedron formed by the vertices of the *visible* triangle $\mathcal{S}_j$ and the point $\mathbf{x}_k + \mathbf{V}_k$. (See Figure II.2) The sign $\varpi_{j,\mathbf{V}_k}$ specifies if matter has been added to or removed from the object volume and it is equal to the sign of $\mathbf{n}_j \cdot \mathbf{V}_k$, where $\mathbf{n}_j$ is the outward surface normal of triangle $\mathcal{S}_j$. The dots part "$\ldots$" on the right of equation (II.34) is null except if the vertex $\mathbf{x}_k$ is a horizon point, which means that this point is on the occluding contour; this additional part will be detailed in the next paragraph.

Now, we parametrize the volume $Vol[j,\mathbf{V},t]$ by the point $\mathbf{x}(u,v,w) = \mathbf{x}(u,v) + w\,t\mathbf{V}_k$, where $\mathbf{x}(u,v) = \mathbf{x}(\mathbf{u})$ parametrizes the triangle $\mathcal{S}_j$ as defined previously and shown in figure II.3; The local parametrization is such that $\mathbf{u}(u,v) \in T = \{(u,v) \mid u \in [0,1]$ and $v \in [0,1-u]\}$ and $w \in [0,1-u-v]$. By a change of variables, Equation (II.34) becomes

$$\sum_j \varpi_{j,\mathbf{V}_k} \int_T \int_0^{\phi_k(\mathbf{u})} \Big\{ f(\mathbf{x}(\mathbf{u}) + wt\mathbf{V}_k) \times |det(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}}, \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}, t\mathbf{V}_k)| \Big\}\,\mathrm{d}w\mathrm{d}\mathbf{u} + \ldots \tag{II.35}$$

Let A_j be the area of the triangle $\mathcal{S}_j$. It is easy to show that $\varpi_{j,\mathbf{V}_k}|det(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}}, \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}, t\mathbf{V}_k)| = 2t\,A_j\mathbf{V}_k \cdot \mathbf{n}_j$. Then Equation (II.35) becomes

$$2t\,\mathbf{V}_k \cdot \left[\sum_j A_j\mathbf{n}_j \int_T \int_0^{\phi_k(\mathbf{u})} f(\mathbf{x}(\mathbf{u}) + wt\mathbf{V}_k)\,\mathrm{d}w\mathrm{d}\mathbf{u} \right] + \ldots \tag{II.36}$$

It follows that the limit of $\frac{\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0])}{t}$ when t tends to zero is

$$\mathbf{V}_k \cdot \left[2\sum_j A_j\mathbf{n}_j \int_T f(\mathbf{x}(\mathbf{u}))\,\phi_k(\mathbf{x}(\mathbf{u}))\,\mathrm{d}\mathbf{u} \right]. \tag{II.37}$$

Now the derivative of the energy can be expressed as a scalar product between the velocity $\mathbf{V}_k$ and a quantity that corresponds to the gradient as explained previously. The part in square brackets corresponds then to the interior term of the gradient of $E(\mathcal{S})$ with respect to $\mathbf{x}_k$:

$$\mathbf{G}_k^{int} = 2 \sum_j A_j \mathbf{n}_j \int_T \nabla \cdot \mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)\, \phi_k(\mathbf{x}(\mathbf{u}))\, \mathrm{d}\mathbf{u}\,, \tag{II.38}$$

where the sum is on all the (completely) visible triangles containing vertex $\mathbf{x}_k$.

II.4.2.3 Term due the movement of the crepuscular cone

In this section we are then going to compute the additional term which appears when $\mathbf{x}_k$ is a horizon point on the occluding contour. In this case the energy variation during a surface movement is due to the volume created by the crepuscular cones. This movement is not affected by the dependency in $\mathbf{n}(\mathbf{x})$ since this is purely due to the visibility changes.

Let $H_{k,j}$ be the vector such that $[\mathbf{x}_k, \mathbf{x}_k + H_{k,j}]$ is the edge of the triangle $\mathcal{S}_j$ generating the horizon. The volume corresponding to the movement of the horizon can be parametrized by the points $\mathbf{y}(u,v)$ of the triangle $\{\mathbf{x}_k, \mathbf{x}_k + H_{k,j}, \mathbf{x}_k + v\, t\mathbf{V}_k\}$ generated by the movement of the horizon. More rigorously, it can be parametrized as the set of points $\mathbf{x}(u,v,r) = r\,\mathbf{y}(u,v)$ where $\mathbf{y}(u,v) = \mathbf{x}_k + u\, H_{k,j} + v\, t\mathbf{V}_k$; r corresponds to the depth of $\mathbf{x}$ in the view point direction; $r \in [1, \mathcal{T}_{(u,v)}]$, where it corresponds to the terminator of $\mathbf{x}_k$ when $r = \mathcal{T}_{(u,v)}$.
Let us note that $\mathbf{y}(u,v)$ depends on t; we emphasize this dependency by denoting $\mathbf{y}_t(u,v)$. By a change of variable, we get $\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0]) =$

$$\cdots - \frac{1}{2} \int_T \int_{r \in [1, \mathcal{T}_{(u,v)}]} \left\{ f(r\,\mathbf{y}_t(u,v))\, (H_{k,j} \wedge\, t\mathbf{V}_k) \cdot \frac{\mathbf{y}_t(u,v)}{|\mathbf{y}_t(u,v)|} r^2 \right\} \mathrm{d}r \mathrm{d}u \mathrm{d}v\,, \tag{II.39}$$

where "..." corresponds to the part described in the previous paragraph. It follows that the limit of $\frac{\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0])}{t}$ when t tends to zero is the term in Equation (II.38) plus the following term

$$-\frac{1}{2} \int_T \int_{[1, \mathcal{T}_u]} f(r\,\mathbf{y}(u))\, (H_{k,j} \wedge \mathbf{V}_k) \cdot \frac{\mathbf{y}(u)}{|\mathbf{y}(u)|} r^2 \mathrm{d}r \mathrm{d}u \mathrm{d}v\,, \tag{II.40}$$

where $\mathbf{y}(u) = \mathbf{x}_k + u\, H_{k,j}$ and $\mathcal{T}_u = \mathcal{T}_{(u,0)}$. Let us denote $\mathcal{L}(u)$ such that

$$\mathcal{L}(u) = \int_{[1, \mathcal{T}_u]} f(r\,\mathbf{y}(u)) r^2 \mathrm{d}r\,. \tag{II.41}$$

The right-hand part of equation (II.40) can be rewritten as

$$-\mathbf{V}_k \cdot \frac{1}{2} \int_{u \in [0,1]} \mathcal{L}(u) \left(\frac{\mathbf{y}(u)}{|\mathbf{y}(u)|} \wedge H_{k,j} \right) (1-u) \mathrm{d}u\,, \tag{II.42}$$

where $\mathcal{L}(u) = \int_{1,\mathcal{T}_u} f(r\,\mathbf{y}(u)) r^2 dr$. Here we have $f(\mathbf{x}) = -\nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))$. Here we can explicit $\mathcal{L}(u)$ as :

$$\begin{aligned}\mathcal{L}(u) &= -\int_{1,\mathcal{T}_u} \nabla g(r\mathbf{y}(u)) \cdot \frac{\mathbf{y}(u)}{\mathbf{y}(u)_z^3} \mathrm{d}r \\ &= -\left[g(T(\mathbf{y}(u))) - g(\mathbf{y}(u))\right] \frac{1}{\mathbf{y}(u)_z^3} \ ,\end{aligned} \tag{II.43}$$

where we have denoted $\mathbf{g}(\mathbf{x}) = g(\mathbf{x}, \mathbf{n})\dfrac{\mathbf{x}}{\mathbf{x}_z^3}$ and $T(\mathbf{y}(u))$ is the point located behind $\mathbf{y}(u)$ in the direction of the viewpoint (i.e. its terminator point).

This gives the third part of the k^{th} component of the gradient of $E(\mathcal{S})$ defined in Equation (II.5):

$$\begin{aligned}\mathbf{G}_k^{horiz} = \sum_{H_{k,j}} \frac{1}{2} \int_0^1 \Big\{ &\big(g\left(T\left(\mathbf{y}(u)\right)\right) - g\left(\mathbf{y}(u)\right)\big) \\ &\times \left(\frac{\mathbf{y}(u) \wedge H_{k,j}}{|\mathbf{y}(u)||\mathbf{y}(u)|_z^3}\right)(1-u)\Big\} \mathrm{d}u \ ,\end{aligned} \tag{II.44}$$

where the sum is on the edges containing $\mathbf{x}_k$ and which generate horizons. $H_{k,j}$ are the horizon edges around vertex $\mathbf{x}_k$.

II.4.2.4 Total Gradient Descent Flow

Finally, as explained in Section II.2.2, the gradient descent flow depends on the choice of an inner product for the gradient. Here we use the L^2 gradient that allows to account for triangle area variations. As described previously and shown in [24, 39], changing the metric can result in more coherent gradient flows, but this study is out the scope of this thesis. In this case the gradient descent flow for a point $\mathbf{x}_k$ corresponding to our energy functional defined on a visible domain is:

$$\begin{cases} \mathbf{x}_k(0) = \mathbf{x}_k^0 \ , \\ \dfrac{d\mathbf{x}_k}{dt} = -\dfrac{1}{A_k} \left\{ \mathbf{G}_k^{int} + \mathbf{G}_k^{horiz} + \mathbf{G}_k^{norm} \right\} \ . \end{cases} \tag{II.45}$$

II.5 Conclusion

In this chapter we compute the shape gradient of general energy functionals which account for normals and visibility changes and which embody a number of energies used in mesh processing and computer vision. Gradient computation is done directly with respect to the discrete representation of the surface based on triangular meshes. This allows for coherent gradient flows that tend to place the mesh vertices to their correct locations and make triangle edges match with the data.

In particular we presented a mesh evolution technique for 3D vision problems, when the cost functional depends on an image based score and a camera model. For instance, our approach can be used to solve multi-view shape reconstruction problems from images using a discrete representation. The following chapter illustrates the use of the presented framework for different application scenarios and examples. Contrary to most previous works, during the evolution, *we correctly deal with visibility changes* by expressing the exact gradient of the reprojection error functional. In particular, exactly as in the continuous case [56], this forces the contour generators of the surface to appear at their correct location in the images and reduces the minimal surface bias from which most of variational methods suffer.

In the next Chapter (III) we present examples and experiments on various multi-view reconstruction cases, which are straightforward implementations of the results shown in this Chapter. Additional discussions and comments are developed in the conclusion of Chapter III.

Chapter III

Applications to 3D Shape Reconstruction from Images

Recently many multi-view stereo techniques have emerged to produce high quality 3D models from two or multiple images. In the previous chapter, we have seen how to optimize 3D deformable meshes from energy functionals. In this chapter, we will see how this framework can be applied to a family of 3D reconstruction problems not limited to stereo correspondence. As described in Chapter I, many other cues can be used to describe the objective functional. In order to illustrate Chapter II, we present different examples of multi-view shape from images problems dealing with visibility issues and exploiting some of those cues such as stereo matching or shading. In particular, we present results for multi-view stereo, multi-view shape-from-shading and multi-view photometric stereo applications. Both synthesized and real experiments are presented in order to validate our approach.

Contents

This chapter shows how the gradient descent framework presented in Chapter II can be used to perform mesh evolutions. In particular several concrete examples that are common in computer vision and especially 3D reconstruction are presented. For more clarity and comparison purposes, the following examples use the L^2-gradient descent scheme presented in the previous chapter. Since the minimization is performed via gradient descent, the algorithm needs to start from some initialization. In most examples the initial shape is the visual hull of the scene, where we assume silhouettes can be easily computed or are given, or that stereoscopic segmentation [160] can be easily performed. It is also important to notice that the horizon term is performed only on existing contour generators, so that the topology has to be close to the final one, or that contour generators can be created thanks to the interior term influence. Also, since it is based on energy differences one may add more or less importance to it. In particular in the following experiments, we add a weight λ_H to control this amount, which is empirically determined. If λ_H is too big, the horizon term will make the contour generators of the surface oscillates around their corresponding apparent contours.

In the following examples, the mesh evolution algorithm that includes the remeshing and the topology changes is done using the Delaunay topology-adaptive meshes proposed by [129] or using TransforMesh [167], both written using CGAL [2]. A multi-resolution coarse-to-fine scheme is used to avoid local minima and make the optimization faster. Also, all figures shown in this section are displayed using flat shading rendering. This means that the shading is computed using the facet's normals and not the weighted normals estimated at each vertex as used in many other works (this makes the model to appear smoother in the images). In that way, one can better estimate the real surface's shape and see triangle edges.

III.1 Multi-view Stereovision

In this example, we detail how the gradient defined in Section II.4 can be used in three-dimensional surface reconstruction from images.

Multi-view stereo is the problem of recovering the shape of scenes using multiple cameras, by assuming that matching or correspondences between different views can be obtained (by usually considering Lambertian constant brightness assumptions). Given a set of images of a scene taken from different camera positions, the goal is to reconstruct the shape $\mathcal{S}$, and optionally the appearance, of the object. Since it is the inverse problem of image rendering, this problem can be modeled in a Bayesian framework by minimizing the difference between the images of the reconstructed model and the observed ones, i.e. the reprojection error.

The multi-view stereovision algorithm we propose here is based on the same modeling as the one proposed by Gargallo et al. [56]. As [160], Gargallo et al. use the Eulerian formulation and implement their algorithm in the level-set framework. Here, we use a triangle mesh-based representation, as described in the previous chapter. Below, we describe the modeling of the problem and then detail the exact gradient for our discrete representation obtained by using the results developed in Section II.4.2.

III.1.1 Modeling of the Reprojection Error

In order to be able to compare the whole observed images (data) with the images generated by the model, it is crucial to define and model the background. Also, as shown by [56, 160], this allows us to be sure that the estimated foreground surface does not shrink to an empty set (which is the global optimum for most cost functionals used in other work). Whereas most of the previous work assumes that the background is known (e.g. simply modeling it as uniformly black, or by exploiting given silhouette images), here we also estimate the background images $B_i : \mathcal{I}_i \to \mathbb{R}^3$, under the single assumption that these images are smooth, similarly as [160]. Here i corresponds to the index of the camera and $\mathcal{I}_i$ is its image domain.

Now, let us assume that the scene surface $\mathcal{S}$ is Lambertian and the illumination static. Let $C : \mathcal{S} \to \mathbb{R}^3$ be the radiance function that associates colors to the points on the surface.

Ideally, the color $I_i(\mathbf{p})$ observed at pixel $\mathbf{p}$ of image I_i should be equal to the color $C(\pi_{i,\mathcal{S}}^{-1}(\mathbf{p}))$ of its backprojection $\pi_{i,\mathcal{S}}^{-1}(\mathbf{p})$ onto the surface or, in the case where $\mathbf{p} \notin \pi_i(S)$, to the color $B_i(\mathbf{p})$ of the same pixel on the background images (π_i denoting the projection associated with camera i). Thus, the reprojection error of the surface is

$$E_{data} = \frac{1}{2} \sum_i \left[\int_{\pi_i(\mathcal{S})} \left(I_i(\mathbf{p}) - C(\pi_{i,\mathcal{S}}^{-1}(\mathbf{p})) \right)^2 \mathrm{d}\mathbf{p} + \right. \\ \left. \int_{\mathcal{I}_i - \pi_i(\mathcal{S})} \left(I_i(\mathbf{p}) - B_i(\mathbf{p}) \right)^2 \mathrm{d}\mathbf{p} \right] . \quad \text{(III.1)}$$

Finally, in order to well pose the problem, we use as a prior on $\mathcal{S}$ an additional smoothing area energy. As described previously, smoothing the normals gives a better prior than the commonly used mean curvature flow. It helps not to over-smooth the surface and to preserve geometric details better. The considered energy is

$$E_{RS} = \int_{\mathcal{S}} (1 - \mathbf{h}(\mathbf{x}) \cdot \mathbf{n}) \, \mathrm{d}s \, ,$$

where $\mathbf{h}$ is the average normal of the considered surface point. We also assume that the background images are smooth by adding the total variation term

$$E_{RB} = \sum_i \int_{\mathcal{I}_i} |\nabla B_i(\mathbf{p})| \mathrm{d}\mathbf{p}.$$

The total variation helps preserving edges and does not over-smooth the object boundaries. In practice, the background images can be identified before the surface optimization by giving the silhouettes or doing stereoscopic segmentation.

III.1.2 Minimization of the Total Energy

For optimizing our total energy

$$E_{total} = E_{data} + \lambda_{\mathcal{S}} E_{RS} + \lambda_B E_{RB} \, , \quad \text{(III.2)}$$

we perform gradient descents alternately with respect to B_i and $\mathcal{S}$.

The computation of the gradients with respect to B_i is classical since it is image-based. For a fixed shape $\mathcal{S}$ and a fixed C, we have

$$\nabla E_{total}(B) = - \sum_i \left[(I_i - B_i)(1 - h) + \lambda_B \nabla \cdot \frac{\nabla B_i}{|\nabla B_i|} \right], \quad \text{(III.3)}$$

where h is the characteristic function that indicates if $\mathbf{u}$ is covered by the projection of the surface $\mathcal{S}$ ($h(\mathbf{p}) = 1$) or not ($h(\mathbf{p}) = 0$, $\mathbf{p}$ being explained by the background); λ is the smoothness parameter; $C(\mathbf{x})$ is computed by taking the mean color of the projection in the image I_i where $\mathbf{x}$ is visible.

Let us now detail the gradient of E_{total} with respect to $\mathcal{S}$. Since E_{RB} do not depend on $\mathcal{S}$ and E_{RS} is classical [33, 160]), the main point is to compute the gradient of E_{data} with respect to $\mathcal{S}$. To apply the results presented in the previous sections, we first need to rewrite the energy E_{data} as an integral over only the visible surface. For simplicity of notation, we are going to give the gradient for a single camera and so we remove the dependency on i. For several cameras, the gradient will be the sum of the gradients associated with each camera.

As explained in the introduction, the first step, to be able to apply our previous results, is to rewrite the energy as an integral over the surface instead of the image. This change of variable implies the geometric model of the camera which we assume to be a pinhole perspective camera model in this work. It also involves adapting the measure on the surface [140] and in counting only the visible points [56, 130, 160]. This can be achieved by $d\mathbf{u} = -\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\nu_{\mathcal{S}}(\mathbf{x})\mathrm{d}s$ where $\mathrm{d}s$ is the classical surface area measure and $\mathbf{x}_z$ is the depth of $\mathbf{x}$. Thus, by using the separation technique proposed by [160], the energy functional becomes (for a *single* image):

$$\begin{aligned} E_{data}(\mathcal{S}) &= -\int_{\mathcal{S}} g_I(\mathbf{x})\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\nu_{\mathcal{S}}(\mathbf{x})\mathrm{d}s \\ &\qquad\qquad + \int_{\mathcal{I}-\pi(\mathcal{S})} g_B(\mathbf{p})\mathrm{d}\mathbf{p}\,, \\ &= -\int_{\mathcal{S}} [g_I(\mathbf{x}) - g_B(\pi(\mathbf{x}))]\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\nu_{\mathcal{S}}(\mathbf{x})\mathrm{d}s \\ &\qquad\qquad + \int_{\mathcal{I}} g_B(\mathbf{p})\mathrm{d}\mathbf{p}\,, \end{aligned} \tag{III.4}$$

where $g_I(\mathbf{x})$ is $\frac{1}{2}\left[I(\pi(\mathbf{x})) - C(\mathbf{x})\right]^2$ and $g_B(\mathbf{p})$ is $\frac{1}{2}\left[I(\mathbf{p}) - B(\mathbf{p}))\right]^2$. The right-hand term of equation (III.4) does not depend on $\mathcal{S}$, so in the following we intentionally omitted it as it does not contribute to the gradient expression (with respect to $\mathcal{S}$). Hence, denoting $g(\mathbf{x}) = g_I(\mathbf{x}) - g_B(\pi(\mathbf{x}))$ for convenience, the energy to be minimized with respect to $\mathcal{S}$ is

$$E_{data}(\mathcal{S}) = -\int_{\mathcal{S}} g(\mathbf{x})\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\nu_{\mathcal{S}}(\mathbf{x})\mathrm{d}s. \tag{III.5}$$

Now, the energy functional $E_{data}(\mathcal{S})$ is of the form of Equation (II.5) with

$$\mathbf{g}(\mathbf{x}) = -\frac{g(\mathbf{x})}{\mathbf{x}_z^3}\mathbf{x}.$$

The gradient descent flow for the shape is then directly given by (II.45)

$$\begin{cases} \mathbf{x}_k(0) = \mathbf{x}_k^0\,, \\ \dfrac{d\mathbf{x}_k}{dt} = -\dfrac{1}{A_k}\left\{\, \mathbf{G}_k^{int} + \mathbf{G}_k^{horiz} \,\right\}\,, \end{cases} \tag{III.6}$$

where $\mathbf{G}_k^{int}$ and $\mathbf{G}_k^{horiz}$ are respectively obtained from Equations (II.38) and (II.44) where $\mathrm{g}(\mathbf{x})$ is replaced by $-\dfrac{g(\mathbf{x})}{\mathbf{x}_z^3}\mathbf{x}$.

Expression of $C(\mathbf{x})$: As mentioned above, $C : \mathcal{S} \rightarrow \mathbb{R}^3$ is the estimated radiance on the surface. Then $C(\mathbf{x})$ should satisfy the following equation:

$$\sum_i \int_{\mathcal{S}} [I_i(\pi(\mathbf{x})) - C(\mathbf{x})]^2 \, w_i(\mathbf{x}) \, \nu_{i,\mathcal{S}}(\mathbf{x}) = 0 \,. \tag{III.7}$$

Here $\mathbf{x}_i$ denote the coordinate of the point $\mathbf{x}$ in the camera coordinate system of camera i (which in this thesis is often omitted since equations are often written for a single image). $w_i(\mathbf{x})$ is a weight function that depends on the camera model. As described previously, for a pinhole camera model then $w_i(\mathbf{x}) = \frac{\mathbf{x}_i \cdot \mathbf{n}(\mathbf{x})}{\mathbf{x}_{i,z^3}}$. Then an obvious choice for the color $C(\mathbf{x})$ is to take the weighted mean color at point $\mathbf{x}$ of the image intensity values from the images where $\mathbf{x}$ is visible. This leads to:

$$C(\mathbf{x}) = \frac{\sum_i I_i(\pi_i(\mathbf{x})) \, w_i(\mathbf{x}) \nu_{i,\mathcal{S}}(\mathbf{x})}{\sum_i w_i(\mathbf{x}) \, \nu_{i,\mathcal{S}}(\mathbf{x})} \,. \tag{III.8}$$

III.1.3 Experiments for the multi-view stereovision application

As previously described, the mesh evolution algorithm is implemented using the Delaunay topology-adaptive meshes. The visibility is computed using OpenGL Z-buffer. The evolution is done using a multi-resolution scheme and starting from the visual hull. The discretization over the triangles is automatically adapted to the image resolution. Horizons are located using the changes of signs of the dot products of facet normals and viewpoint directions. The *terminator* error metric is computed using OpenGL Shader Language.

As in [56], we first reconstruct three uniformly colored balls arranged on a plane (20 images of resolution 640×480), see Figure III.1. This way we ensure the importance of the *horizon* term as the color gradient is null over the surface except at the interfaces between objects in images (which correspond to object boundaries). Using only the *interior* term (Section II.4.2.2), the surface shrinks due to the minimal bias. By using the horizon term only (given in Section II.4.2.3), we correctly reconstruct and separate the balls, and occluding contours correctly reproject in the images. Then we tested our algorithm on synthetic Lambertian data for the Stanford dragon mesh (Figure III.2) composed of 32 images of resolution 640×480. The result shows the correct reconstruction of the dragon, even though the texture is smooth and some parts in shadow are dark (See Table III.1). Here the initial shape was a visual hull automatically computed from a stereoscopic segmentation algorithm [160].

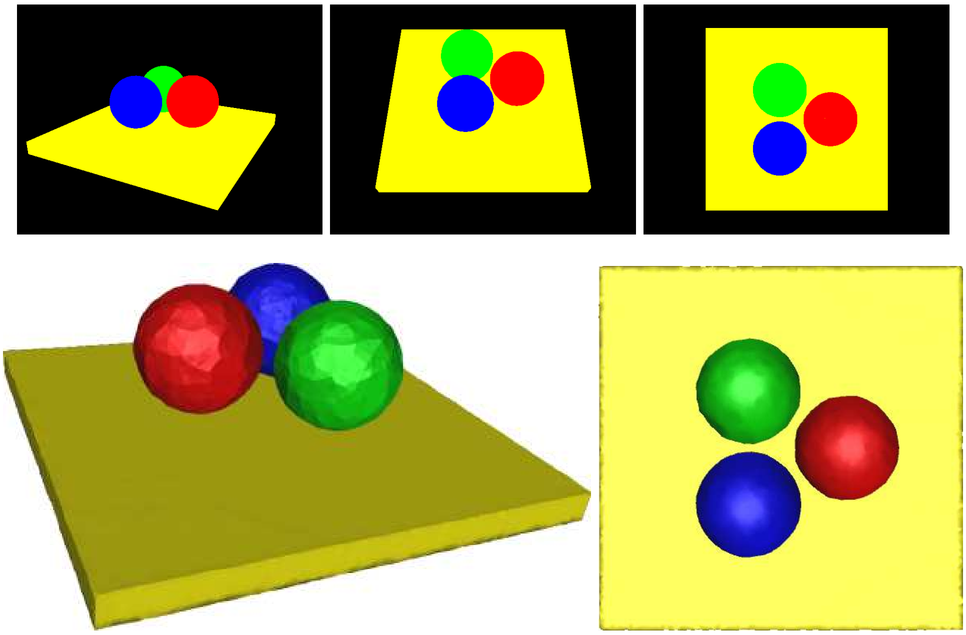

Figure III.1: The balls sequence. Top row: 4 of 20 input images. Bottom row: results with the horizon term computed in Section II.4.2.3 from different viewpoints.

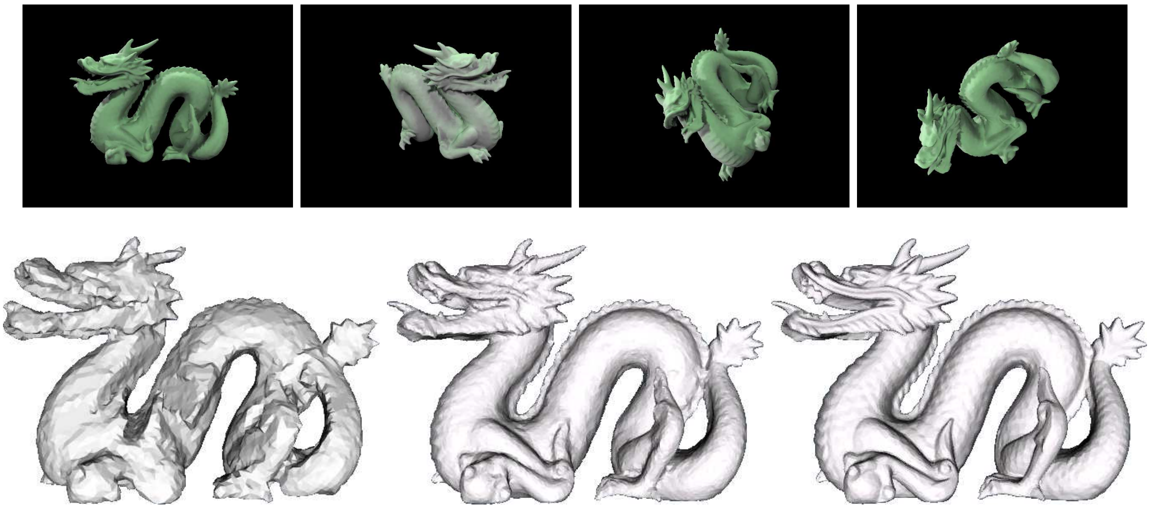

Figure III.2: Synthesized dragon sequence. Top row: 4 of 32 diffuse input images. Bottom row: Initial shape; recovered shape by our algorithm; ground truth model.

Then, we tested our method on the classical Dino (16 images of 640 × 480) and Temple (16 images of 640 × 480 images) datasets from the Middlebury repository (Figure III.3). The results and a comparison with selected approaches that motivated our work is presented Table III.2. The results of [55, 56] are done in the continuous domain using level set implementation. We refer to the Middlebury benchmark website [136] for evaluation with state-of-the-art

	Dragon Diffuse images (Figure III.2)
Acc. 95%	0.241mm
Comp. 0.5mm	98.3%

Table III.1: Numerical evaluation of the proposed method for the dragon sequence that shows accuracy at 95% and completeness at 0.5mm following [136]. (compared to ground truth)

	templeSparseRing		dinoSparseRing	
	accu. (mm)	compl. (%)	accu. (mm)	compl. (%)
Gargallo [56]	1.05	81.9	0.76	90.7
Gargallo [55]	0.79	96.8	0.50	97.7
Our approach	0.73	95.9	0.89	93.9

Table III.2: Results for the temple and dino datasets. For each dataset, accuracy and completeness scores are given. Results of Gargallo et al. are shown for comparison purposes.

reconstruction. We can see that our method is comparable to state-of-the-art, but the main contribution here results in giving a unified framework for photo-consistency optimization that correctly handle visibility using triangular meshes. Additional terms like ballooning forces or silhouettes terms can now be understood, by only dealing with the reprojection error criteria.

Figure III.3: Dino and Temple sequence (input data courtesy of [136]). From left to right: 1 of 16 input images; estimated background images (scaled by 2 for visualization); estimated radiance; estimated mesh seen from a different viewpoint.

Comparison with Structured Light Scanning using 6 Cameras

Structured light is a 3D reconstruction technique that has proved to provide accurate results. In that case a designed pattern is projected onto the object (for example using a projector). This makes the correspondence between the pattern and capture images easier, and 3D models can be reconstructed by simple matching algorithms. Recent techniques allow those approaches to work in multi-view settings [50]. In [50], the authors extract 3D points by direct matching between projectors and cameras, and by checking the consistency in multiple views. Finally a surface is extracted via Graph cuts from those points and silhouette information.

The following Figures III.4 and III.5 show results for multi-view 3D reconstruction of dynamic scenes, without time consistency. The projected pattern are composed of color coded lines. In this system there are only 6 cameras along with 6 projectors, that are placed in circle around the object of interest in order to capture the full 3D shape. From a multi-view stereo point of view, the projected patterns can also be exploited as additional texture information onto the images. Then if the pattern is designed to be locally unique, it will result in a more discriminative photo-consistency, and multi-view stereo can be performed even though there are only 6 cameras.

From the Figures III.4 and III.5, we can see that our reconstruction is not perfect but still in some areas performs better than using the structured light system. Please notice that only 6 cameras where used here, which is a very critical case for multiview stereo. In fact, each surface point is seen by 1, 2 or 3 cameras at most, since the relative angle between two cameras is approximately 60 degrees. Then points seen by 3 cameras are watched by two camera whose angles are 120 degrees, which is an extreme case for stereo and would make any feature based technique fail to reconstruct points (note that the structured light system uses nearby camera/projector pairs and therefore performs matching with a 30 degrees of baseline). In that context, we believe our approach performs very well and that extra cameras could reduce errors and improve the reconstruction significantly. In our approach, since consistent points are search directly in 3D and that the projected pattern helps making the photo-consistency discriminative enough, 3D models can be recovered. Also, it surprisingly even outperforms structured light systems, See Figures III.4 and III.5. However, without much effort on both cases on parallelization, GPU computing and optimization, the algorithm of [50] takes around 20 seconds where our optimization method takes between 5 and 10 minutes for one frame (where most time is spent on last fine resolution iterations).

One other extension of this work is to use motion as an additional reconstruction cue in order to obtain temporally consistent reconstructions. This would improve the 3D models and avoid flickering in the results. In this case texture information is hard to extract, since the texture is mostly the one coming from the projected patterns, and is not attached to the surface. Then approaches such as [27] can not be straightforwardly applied in this case since texture cannot be tracked. Finally photometry could also be exploited in this case since the illumination sources can be easily calibrated (since they are coming from a projector). While dealing with photometry will be addressed in the next sections using directional or point light

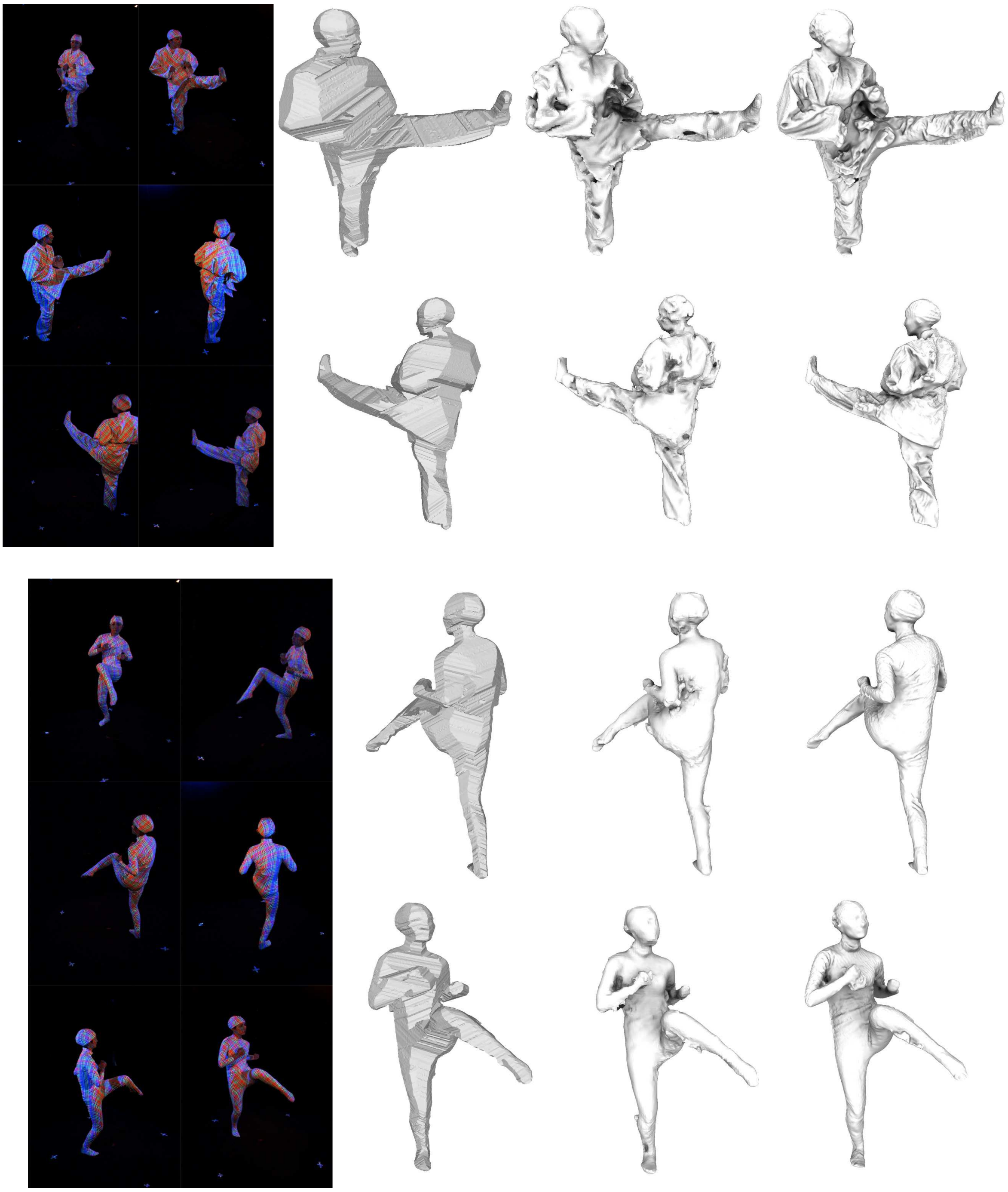

Figure III.4: Comparison with a structured light multi-view system. From left to right: the 6 input images; the visual hull; the reconstructed shape from [50]; the estimated mesh using our multi-view stereo approach.

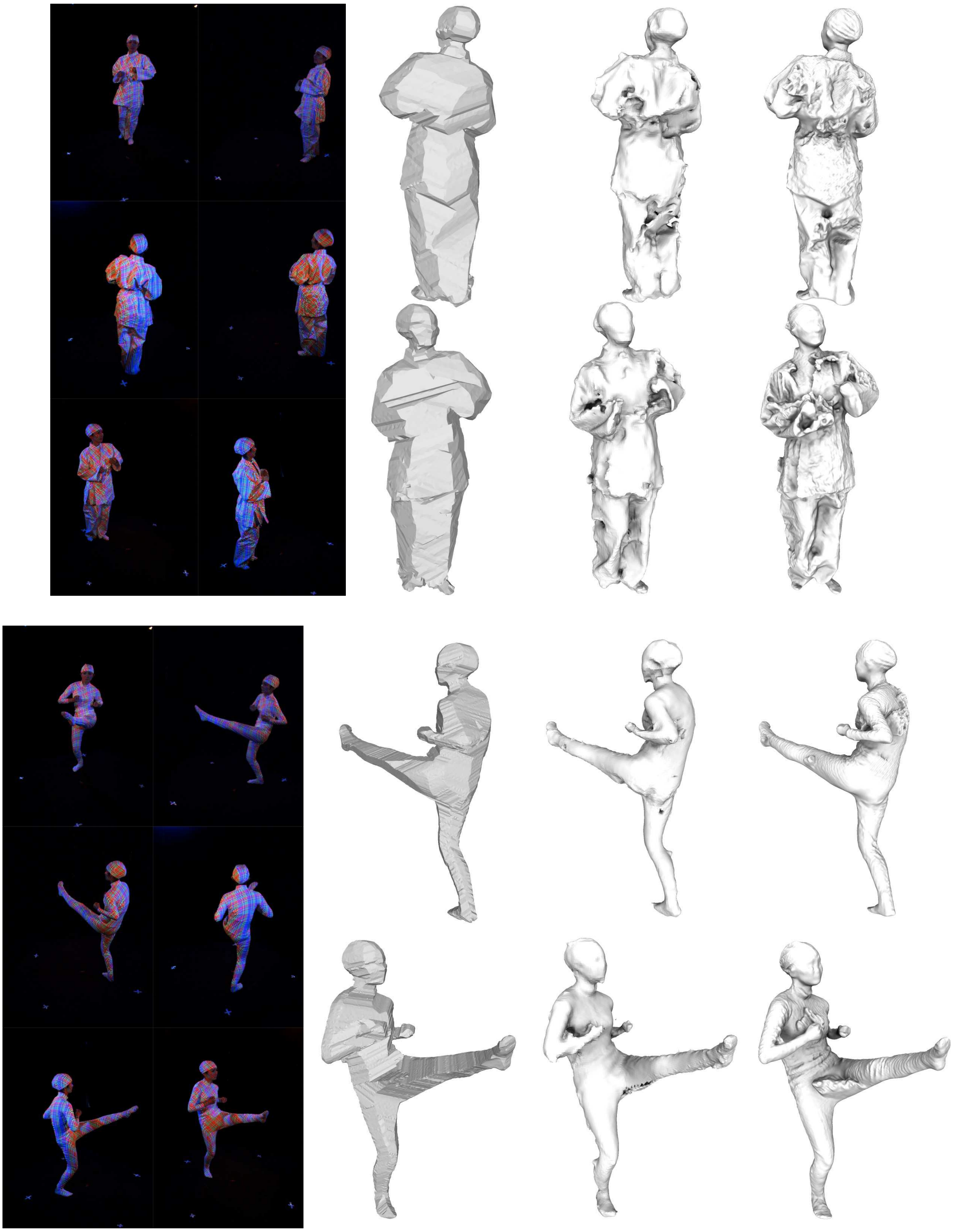

Figure III.5: Comparison with a structured light multi-view system. From left to right: the 6 input images; the visual hull; the reconstructed shape from [50]; the estimated mesh using our multi-view stereo approach.

sources, dealing with projected light pattern would be worth to try and might be part of future work.

III.2 Lambertian 3D Reconstruction using Illumination

In the previous case, the illumination was not taken into account. In fact the estimated color of the surface compared to the input images was the estimated radiance of the object (which was supposed to be Lambertian). However, radiance contains shading, inter-reflections, cast shadows and other non-Lambertian phenomena. The illumination adds additional cues that can be used to estimate surface normals. Considering this, shading can be used in order to recover the geometry of textureless regions. This also allows to separate the surface albedo from the radiance and allows to do more realistic object relighting.

In this section, we only consider the case of Lambertian surfaces. In particular, this allows to consider the multi-view Shape From Shading problem (SFS) and the multi-view Photometric Stereo problem (PS). These problems consist in recovering the 3D shape of a scene by exploiting the information contained in the shading of the corresponding images. In contrast to multi-view stereo, the illumination conditions may vary between different views, and has to be known in advance or has to be estimated. Basically in multi-view photometric stereo, the images are generated with varying lighting (typically, each point of the surface must be seen with three different lights). Whereas in multi-view shape from shading, the lighting is the same for all the images. The SFS problem is therefore less well-posed than the PS problem. It then needs additional constraints. In SFS, we classically assume that the reflectance properties are homogeneous over the whole scene.

A solution to such problems would be a surface $\mathcal{S}$ such that the images generated from that surface are very similar to the observed images (i.e. the data). This naturally leads to formulate the problem as the minimization of an error measure between the observed and predicted values of pixels. For simplicity, here we are going to consider only Lambertian scenes illuminated by point light sources. This work can nevertheless be extended to other parametric reflectance models and to more realistic lighting conditions for example as done in [79, 163]. For a point $\mathbf{x}$ of the surface $\mathcal{S}$, the radiance equation for the i^{th} image is then

$$\begin{aligned} I_i(\pi_i(\mathbf{x})) &= \rho(\mathbf{x}) \left(\sum_{l=1}^{n_L^i} L_l^i \nu_{l,\mathcal{S}}^i(\mathbf{x}) (\mathbf{n}(\mathbf{x}) \cdot \mathbf{l}_l^i(\mathbf{x})) + E_0 \right) \\ &= R(\mathbf{x}, \mathbf{n}(\mathbf{x}), \mathcal{S}) \ . \end{aligned} \tag{III.9}$$

Above L_l^i and $\mathbf{l}_l^i$ are respectively the light intensity color and the light direction of the l^{th} light in the i^{th} image. $\nu_{l,\mathcal{S}}^i(\mathbf{x})$ is the visibility of the l^{th} light source of the i^{th} image at point $\mathbf{x}$ according to $\mathcal{S}$. The additional term E_0 corresponds to the ambient lighting.

A natural energy functional to be considered can be written as :

$$E(\mathcal{S}) = \frac{1}{2}\sum_i \int_{\mathcal{I}_i} \big(I_i(\mathbf{p}) - R(\pi^{-1}_{i,\mathcal{S}}(\mathbf{p}), \mathbf{n}(\pi^{-1}_{i,\mathcal{S}}(\mathbf{p})), \mathcal{S})\big)^2 \mathrm{d}\mathbf{p}, \tag{III.10}$$

see Jin et al. [79] for details.

To minimize this energy, we alternately minimize it with respects to the shape $\mathcal{S}$ and to the albedo ρ.

Let us first fix the albedo and optimize energy (III.10) with respects to the shape. To simplify, we are going to neglect the variations of the visibility of the light sources $\nu_{l,\mathcal{S}}(\mathbf{x})$ when the shape is deforming. These variations are null almost everywhere. They are Dirac functions with a support restricted to the shadow boundaries. In other words, we are neglecting cast shadows information. Practically in the following, this is equivalent to assuming that $R(\mathbf{x}, \mathbf{n}, \mathcal{S})$ does not depend on $\mathcal{S}$.
In other respects, as previously done in other applications, let us note that the gradient of the energy (III.10) is the sum of the gradients associated with each one of the images. In the sequel we only compute the gradient associated with one image. By assuming that the camera is a pinhole, we can rewrite this energy for a single image as:

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\ \nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s\ , \tag{III.11}$$

where

$$\mathbf{g}(\mathbf{x}, \mathbf{n}) = -\frac{1}{2}\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n})\right)^2 \frac{\mathbf{x}}{\mathbf{x}_z^3} \tag{III.12}$$

and where

$$R(\mathbf{x}, \mathbf{n}) = \rho(\mathbf{x}) \left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x})(\mathbf{n} \cdot \mathbf{l}_l(\mathbf{x})) + E_0\right)\ . \tag{III.13}$$

We can then directly use the results of Section II.4. The gradient is split into three parts: one for the term depending on normals, one for the interior term, and a last one for the horizon term.

For the term depending on normals, we have :

$$\mathbf{G}_k^{norm} = -\sum_{j \in \mathcal{J}_k} \mathbf{e}_{j,k} \wedge \int_T P_{n_j^\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)^\top \mathbf{n}_j)\ \nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u}))\mathrm{d}\mathbf{u} \tag{III.14}$$

where $P_{n_j^\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j)$ is the projection of $D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j$ on the tangent plane of the surface. It can be re-written as $D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j - \big(\mathbf{n}_j \cdot D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^\top \mathbf{n}_j\big)\, \mathbf{n}_j$. Here we have

$$D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j) = \rho(\mathbf{x})\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n}_j)\right) \frac{\mathbf{x}}{\mathbf{x}_z^3} \left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) \mathbf{l}_l(\mathbf{x})^\top\right)\ ; \tag{III.15}$$

$\frac{\mathbf{x}}{\mathbf{x}_z^3}\left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) \mathbf{l}_l(\mathbf{x})^\top\right)$ being a 3 matrix, so

$$D_{\mathbf{n}}\mathbf{g}(\mathbf{x},\mathbf{n}_j)^\top \mathbf{n}_j = \rho(\mathbf{x})\left(I(\pi(\mathbf{x})) - R(\mathbf{x},\mathbf{n}_j)\right)\left(\frac{\mathbf{x}}{\mathbf{x}_z^3}\cdot \mathbf{n}_j\right)\left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x})\mathbf{l}_l(\mathbf{x})\right). \tag{III.16}$$

The k^{th} component of the *interior* term is:

$$\mathbf{G}_k^{int} = \sum_j A_j \mathbf{n}_j \int_T \nabla \cdot \mathbf{g}(\mathbf{x}(\mathbf{u}),\mathbf{n}_j)\, \phi_k(\mathbf{u})\, \mathrm{d}\mathbf{u}\,, \tag{III.17}$$

where the sum is on the set of the (completely) *visible* triangles $\mathcal{S}_j$ containing the vertex $\mathbf{x}_k$, and where

$$\begin{aligned}\nabla \cdot \mathbf{g}(\mathbf{x}(\mathbf{u}),\mathbf{n}_j) = &- \left(I(\pi(\mathbf{x})) - R(\mathbf{x},\mathbf{n})\right)\\ &\times \left(D\pi(\mathbf{x})^\top \nabla I(\mathbf{x}) - \nabla_{\mathbf{x}} R(\mathbf{x},\mathbf{n}_j)\right)\cdot \frac{\mathbf{x}}{\mathbf{x}_z^3}\,.\end{aligned} \tag{III.18}$$

Above, all the terms are explicit at the exception of $\nabla_{\mathbf{x}} R(\mathbf{x},\mathbf{n}_j)$. In fact $\nabla_{\mathbf{x}} R(\mathbf{x},\mathbf{n}_j) = \nabla\rho(\mathbf{x})\mathrm{L}(\mathbf{x},\mathbf{n}_j) + \rho(\mathbf{x})\nabla_{\mathbf{x}}\mathrm{L}(\mathbf{x},\mathbf{n}_j)$ where we denote $\mathrm{L}(\mathbf{x},\mathbf{n}_j) = \sum_{l=1}^{n_L} L_l\nu_l(\mathbf{x})(\mathbf{n}_j \cdot \mathbf{l}_l(\mathbf{x})) + E_0$. The computation of the term $\nabla_{\mathbf{x}} R(\mathbf{x},\mathbf{n}_j)$ and $\nabla_{\mathbf{x}}\mathrm{L}(\mathbf{x},\mathbf{n}_j)$ are detailed and discussed in Appendix A.2.3.

The k^{th} component of the *Horizon* term is:

$$\mathbf{G}_k^{horiz} = -\sum_{H_{k,j}} \frac{1}{2}\int_0^1 \mathcal{L}(u)\left(\frac{\mathbf{y}(u)}{|\mathbf{y}(u)|} \wedge H_{k,j}\right)(1-u)\mathrm{d}u\,, \tag{III.19}$$

where $\mathcal{L}(u) = -\left[h(T(\mathbf{y}(u))) - h(\mathbf{y}(u))\right]\frac{1}{\mathbf{y}(u)_z^3}$ with $h = \frac{1}{2}\left(I(\pi(\mathbf{x})) - R(\mathbf{x},\mathbf{n}(\mathbf{x}))\right)^2$ and $T(\mathbf{y}(u))$ is the *terminator* point of $\mathbf{y}(u)$.

Then, for a fixed shape $\mathcal{S}$, the optimal albedo $\rho(\mathbf{x})$ is obtained using :

$$\rho(\mathbf{x}) = \frac{\sum_i I_i\left(\pi_i(\mathbf{x})\right) w_i(\mathbf{x})\, \nu_{\mathcal{S},i}(\mathbf{x})}{\sum_i \left(\sum_{l=1}^{n_L^i} L_l^i \nu_{l,\mathcal{S}}^i(\mathbf{x})(\mathbf{n}(\mathbf{x})\cdot \mathbf{l}_l^i(\mathbf{x})) + E_0\right) w_i(\mathbf{x})\, \nu_{\mathcal{S},i}(\mathbf{x})}\,, \tag{III.20}$$

where similarly as in Equation (III.8), the weight function w_i depends on the camera model, here a pinhole model: $w_i(\mathbf{x}) = \frac{\mathbf{n}(\mathbf{x})\cdot\mathbf{x}}{\mathbf{x}_z^3}$.

The gradient descent flow is the same as previously defined, where we alternatively update the shape $\mathcal{S}$ (using Equations (III.14),(III.17) and (III.19)) and the albedo ρ (using Equation (III.20)). If the albedo is known in advance, this last step is unnecessary. When we assume that the albedo is homogeneous (single material objects), the denominators and numerators of the above equation have to be integrated on the whole surface $\mathcal{S}$.

III.2.1 Experimental Results on Asteroid Reconstruction

Reconstructing 3D objects in Space such as asteroids and planets is of broad interest in the research space industry. 3D models allow researchers to study the structure of asteroids in order to understand their history and trajectory. In such a context, active illumination vision systems can not be applied. However, probes can be sent in orbit around those objects in order to take pictures of them. Vision in Space has interesting properties. In particular the simplified image formation model presented in the introduction (Section I.1) is almost valid and asteroid's reflectance is very close to Lambertian. In space the single point source illumination is also valid since the only light source is the sun, and no scattering and diffusion effects are observable since there is no atmosphere in front of the camera. If sufficiently far from the asteroids, then the directional light source assumption can also be exploited.

In order to illustrate the multi-view shape from shading approach presented above, we show an example of a 3D reconstruction of an asteroid. The images were generated from a dedicated simulator Pangu [125] and given by the company EADS Astrium. First we show an example where the illumination is fixed with respect to the object. In that case it corresponds to the multi-view stereovision case presented in Section III.1. Since the sun is fixed with respect to the asteroid, only the illuminated part of the object can be reconstructed since the rest is occluded by the light.

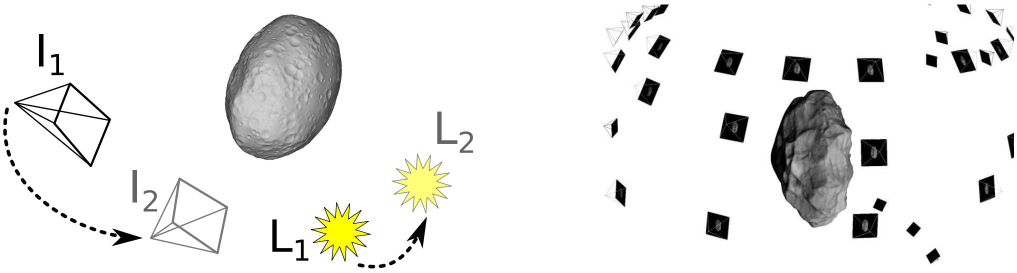

Figure III.6: Asteroid reconstruction scenario. Left: images taken under different lighting conditions. Right: Example of images taken around the asteroid.

Finally, asteroids usually turn around themselves, and as a consequence, light conditions are different for each of the taken images (Figure III.6 illustrates that scenario). Each image is taken under a single but different light source (with respect to the 3D shape) that can be modeled as directional if the sun is sufficiently far from the asteroid. Then, the constant brightness assumption is not valid and multi-view stereo cannot be applied since correspondence will fail. It is in that case necessary to exploit shading using the gradient computed above in this Section. Figure III.8 shows a reconstructed asteroid by minimizing a similar energy as Equation (III.11) that has been simplified to match our assumptions:

$$E(\mathcal{S}) = \sum_i \int_{\mathcal{S}} (I_i(\pi_i(\mathbf{x})) - \rho(\mathbf{x})\, \nu_{\mathcal{S},\mathbf{l}_i}(\mathbf{x})\, \mathbf{n}(\mathbf{x}) \cdot \mathbf{l}_i)^2\, \frac{\mathbf{n}(\mathbf{x}) \cdot \mathbf{x}}{\mathbf{x}_z^3}\, \nu_{\mathcal{S},i}(\mathbf{x})\, \mathrm{d}s\,. \tag{III.21}$$

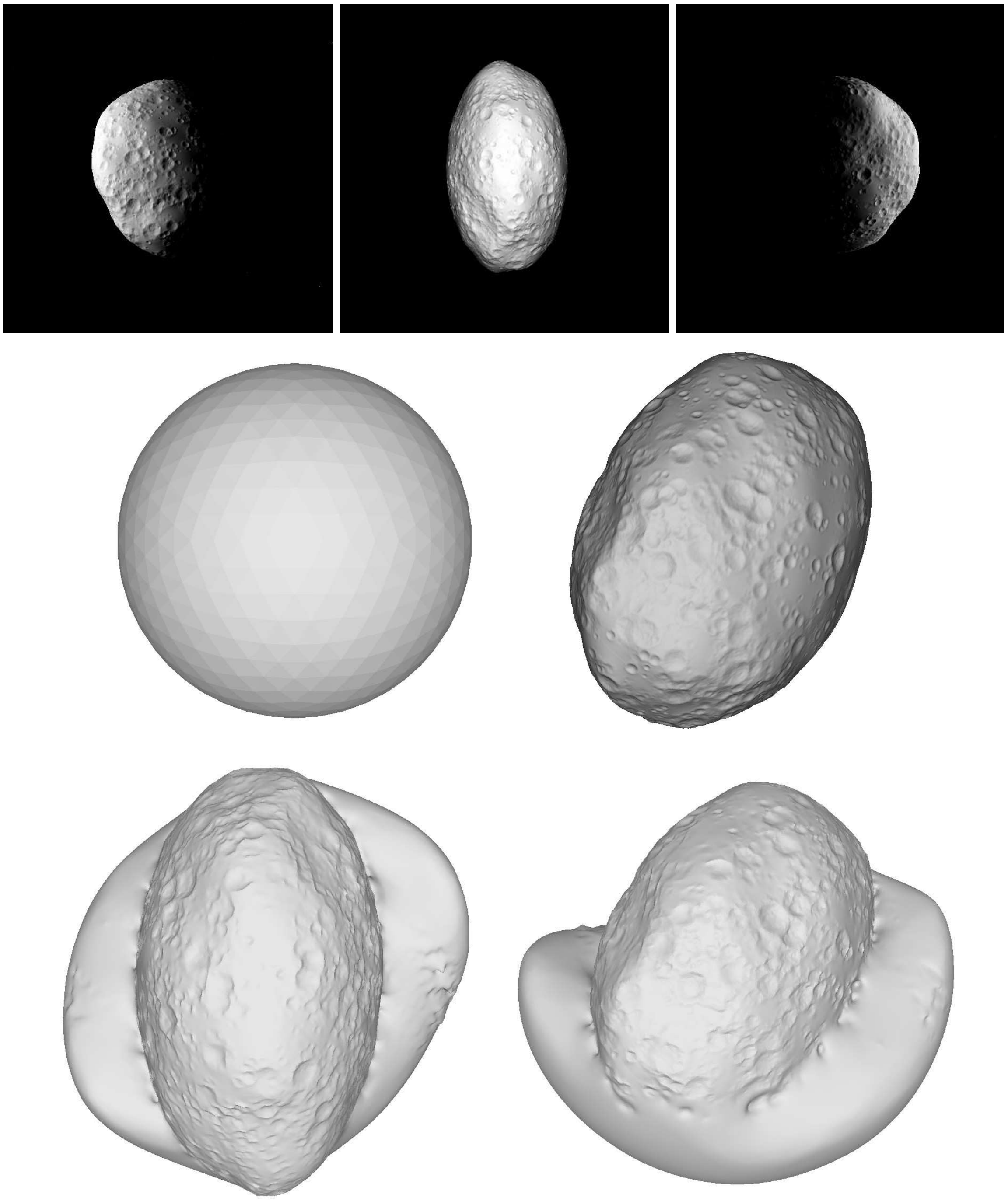

Figure III.7: Asteroid 3D reconstruction example with fixed illumination (stereo case). Top: 3 of 30 input images; Middle: Initial shape (sphere) and ground truth data; Bottom: two views of the reconstructed shape using multiview stereo. Parts occluded by the sun cannot be recovered.

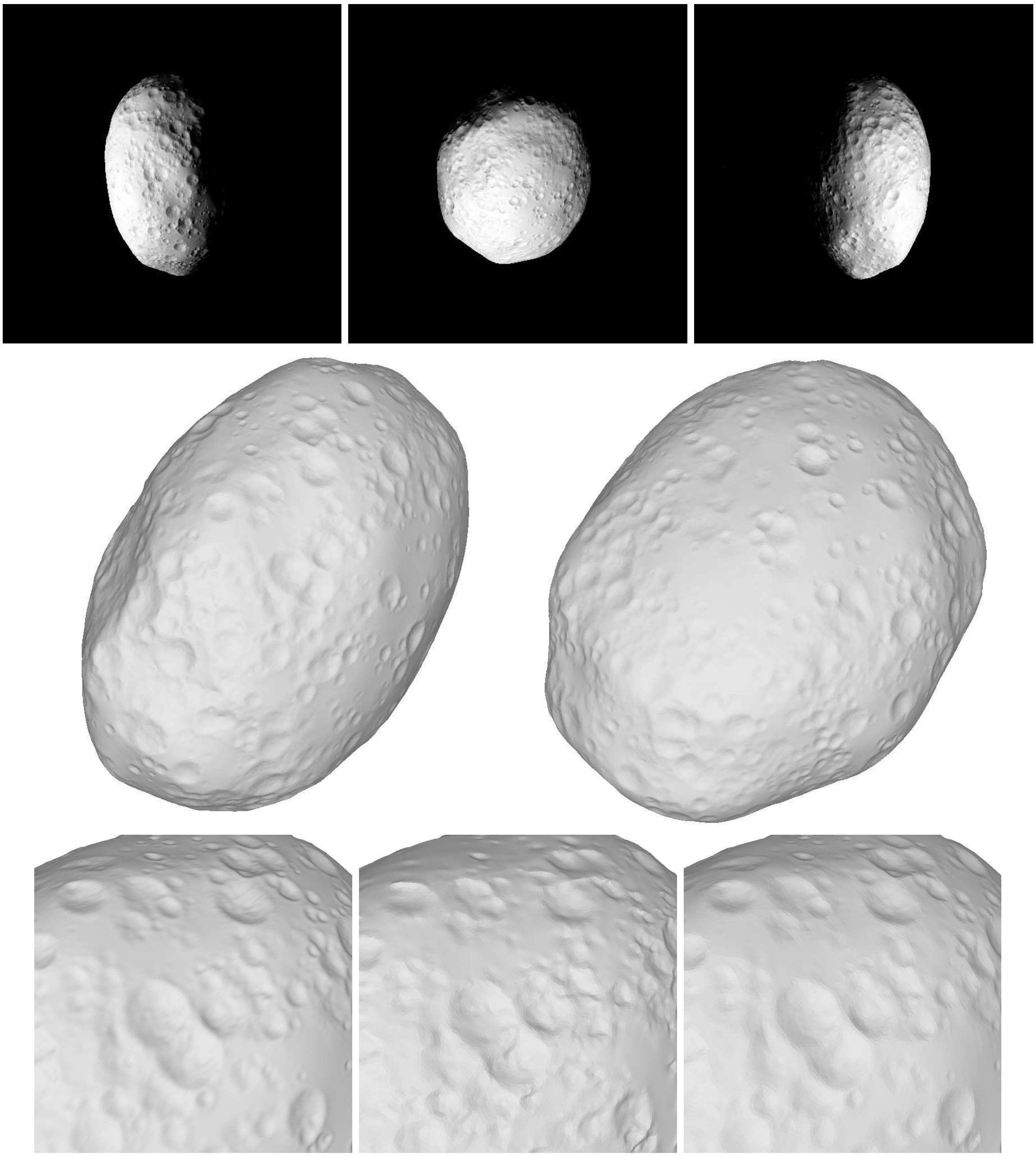

Figure III.8: Asteroid 3D reconstruction example with illumination changes. Top: 3 of 40 input images; Middle: two views of the reconstructed shape using multiview shape-from-shading and starting from the visual hull. Bottom: close-up views and details on the ground truth data (left), the multiview stereo results (Figure III.7) and the multi-view shape from shading approach (right). Since in this case the light source position is different for each view, the whole asteroid can be reconstructed by exploiting photometry.

III.3 Multi-view Normal Integration

In this section, we present an application for integrating surfaces from multiple normal maps like for instance the one of Chang et al.[23] developed in the level sets framework. Such normals can for instance be obtained via photometric stereo that uses a single fixed camera and a moving light source [68, 152]. Having different illumination conditions for one particular view-point allows to estimate the surface normals (See Chapter I for more details on this). By integrating this normal field, it is possible to recover the 3D geometry of the scene. This can be done using the previously described method for normal field integration. However, since photometric stereo is a vision-based application that allows to recover normals for each pixel in the image, the energy functional is based on camera modeling and therefore the energy can be expressed as a reprojection error functional. The gradient descent corresponding to this problem then directly follows the approach presented in Section II.4.

The problem can be solved by minimizing the following energy functional:

$$E(\mathcal{S}) = \sum_i \int_{\mathcal{I}} \frac{1}{2} \left(N_i(\mathbf{p}) - \mathbf{n}(\pi_{\mathcal{S}}^{-1}(\mathbf{p})) \right)^2 \, \mathrm{d}\mathbf{p} \,, \tag{III.22}$$

where $N(\mathbf{p})$ is the normal in input image and $\mathbf{n}(\mathbf{x})$ is the normal of the surface $\mathcal{S}$ at point $\mathbf{x}$. As the norms of N and $\mathbf{n}$ are equal to 1, for simplicity one can rewrite Equation (III.22) for a *single* image as:

$$E(\mathcal{S}) = \int_{\mathcal{I}} \left(1 - N(\mathbf{p}) \cdot \mathbf{n}(\pi_{\mathcal{S}}^{-1}) \right) \mathrm{d}\mathbf{p} \,. \tag{III.23}$$

Rewriting it as an integral over the visible surface, we have:

$$E(\mathcal{S}) = \int_{\mathcal{S}} (1 - N(\pi(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})) \, \frac{\mathbf{x} \cdot \mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \, \nu_{\mathcal{S}}(\mathbf{x}) \, \mathrm{d}s \,, \tag{III.24}$$

which has a similar form as the one used previously for the reprojection error with $g(\mathbf{x}, \mathbf{n}) = (1 - N \cdot \mathbf{n})$. This way one can use previous results when the energy functional also depends on the normal. The differential of the energy with respect to a vertex $\mathbf{x}_k$ for the term due to the normal is:

$$\mathbf{V}_k \cdot \sum_j \mathbf{e}_{j,k} \wedge \int_{T_j} \left\{ \Big(N - (N \cdot \mathbf{n}_j)\, \mathbf{n}_j \Big) \, \frac{\mathbf{x} \cdot \mathbf{n}_j}{\mathbf{x}_z^3} \, \nu_{\mathcal{S}}(\mathbf{x}) \, \phi_k(\mathbf{x}(\mathbf{u})) \right\} \mathrm{d}\mathbf{u} \,. \tag{III.25}$$

Then, to get the complete gradient, one has to sum (the gradient corresponding to Equation (III.25)) with the term due to the differential of a quantity integrated over a visible volume (containing the *interior* term and the *horizon* term). The *interior* term is null because on the triangle $\nabla \cdot \mathbf{n}_j = 0$ and because $(\nabla N) \cdot \mathbf{x} = 0$. The Horizon term is:

$$\mathbf{V}_k \cdot \sum_{H_{k,j}} \frac{1}{2} \int_0^1 \left\{ \Big(\mathbf{n}_j - \mathbf{n}(T(\mathbf{y}(u)))\Big) \cdot N(u) \right. \\ \left. \frac{\mathbf{y}(u) \wedge H_{k,j}}{|\mathbf{y}(u)|\,[\mathbf{y}(u)]_z^3} (1-u) \right\} \mathrm{d}u\,, \tag{III.26}$$

where $T(\mathbf{y}(u))$ is the *terminator* of the current point $\mathbf{y}(u)$ (located behind $\mathbf{y}(u)$ in the view point direction). Note that compared to the gradient in the continuous case, we have here a lower complexity (we are missing the divergence operator and have instead the vectorial product with the opposite edge of the triangle). It is more natural to implement on triangular meshes than the previous continuous case [55] :

$$\nabla\Big((N - (N \cdot \mathbf{n})\mathbf{n}) \frac{\mathbf{x} \cdot \mathbf{n}}{\mathbf{x}_z^3} + (N - \mathbf{n})^2 \frac{\mathbf{x}}{\mathbf{x}_z^3}\Big) \nu_{\mathcal{S}} \\ + \big((N - \mathbf{n})^2 - (N - \mathbf{n}')^2\big) \frac{\mathbf{x}^t \nabla \mathbf{n} \mathbf{x}}{\mathbf{x}_z^3} \delta(\mathbf{x} \cdot \mathbf{n}) \nu_{\mathcal{S}}\,. \tag{III.27}$$

As described previously, this can be extended to multiview photometric stereo methods, where normals are estimated using reflectance and lighting conditions. Then it is possible to integrate this normal field estimated for each image pixel in order to recover the full 3D shape [23, 68, 152]. In the following, we illustrate the approach with different example, where we also add a smoothness term . The corresponding energy is:

$$E_{R\mathcal{S}} = \int_{\mathcal{S}} (1 - \mathbf{h}(\mathbf{x}) \cdot \mathbf{n})\, \mathrm{d}s\,,$$

where $\mathbf{h}(\mathbf{x})$ corresponds to the mean of all the normals viewed from each camera at point $\mathbf{x}$. The corresponding gradient is a straight forward application of Section II.3.3.2.

Figure III.9, III.10 and III.11 illustrates our method on synthetic examples for multi-view normal field integration. First, we tested the multiview normal integration algorithm on the simple Ellipse dataset. By using only the term depending on the normal, the surface shrinks. The horizon term allows to constraint the surface such that it matches the image contours. It naturally gives boundary conditions for the normal integration and allows to start from surfaces that does not fully contain the object of interest.

The second experiment shown in Figure III.10 illustrates the approach on a CAD designed mesh. The original mesh as twice more vertices and triangles than the reconstructed one. Moreover, since we use a coherent gradient descent flow with respect to the mesh representation, we do not assume normal velocity like in [9, 36], making vertices move to appropriate locations. This makes the recovered triangles nicely matches image edges even though the mesh resolution is not very high. Our method can then be used to reconstruct surfaces with sharp edges which is, as far as we know, not possible using implicit surface representations.

The third experiment (Figure III.11) shows the efficiency of the proposed method for

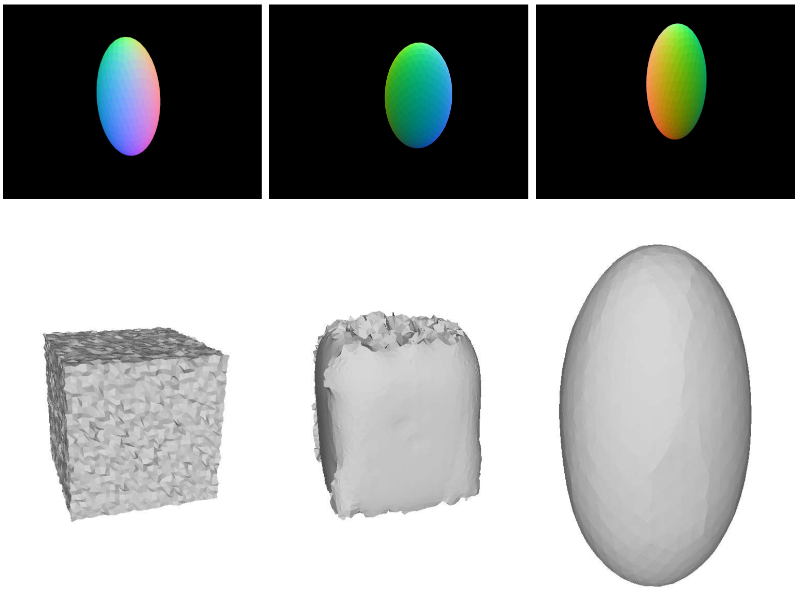

Figure III.9: The ellipse sequence. Top row: 3 of 24 input images showing normal maps of the object of interest; Bottom row: Initial surface; Intermediate result during the evolution; Reconstructed surface.

handling complex surfaces. The initial surface is the visual hull and a coarse to fine approach is used for the evolution. Details of the result are well recovered, and the final shape is very similar to the ground truth even though the input images have low resolution (640×480). Using photometric information, our method can then be used in order to obtain high quality meshes.

In order to compare ours results to state-of-the-art methods, we tested our approach on the dynamic photometric stereo dataset provided by Vlasic et al. [152]. It is composed of 8 images associated with 8 normal maps. The presented approach can directly be applied to their dataset by performing normal integration. Figure III.12 illustrates those results. One of the images shows the shape obtained without using the *horizon* term. In this case the surface shrinks toward the empty set. The result on the right is obtained using the same flow plus the one of the *horizon* term, which yields the expected result. Integrating the normal field gives good high frequency and details, but is poor for low frequency due to the integration. In these conditions, mixing multiview stereo and multiview normal field integration will provide powerful 3D reconstruction algorithms [118, 152]. Figure III.13 shows the results obtained by [152] along with ours. In their paper, the authors compute several normal maps from each view, and then register and merge the different integrations in order to obtain the final mesh. In this context they have troubles in recovering parts where the normal maps contain occluding contours. Since our approach is surface-based, we can better exploit the multi-view system in the reconstruction process and the 3D position of the surface is more accurate even though both methods nicely recover shape details (note that there are *only* 8 images). In fact, Figure III.13 also shows the textured meshes obtained by reprojecting camera images onto the

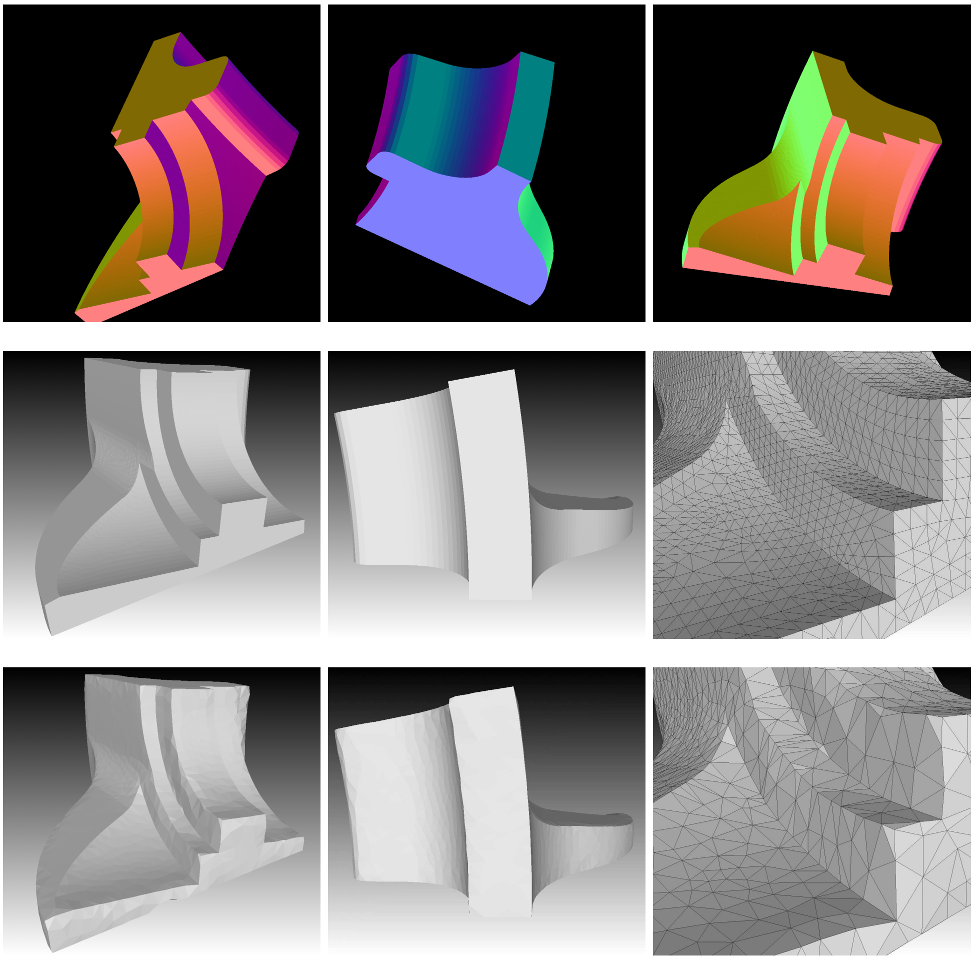

Figure III.10: The Fandisk sequence. From Top to Bottom rows: 3 of 24 input images showing normal maps of the object of interest; original CAD model; reconstructed mesh. The last column shows details of the meshes with the associated triangulation. It shows that the coherent gradient flow makes triangle edges match with the data.

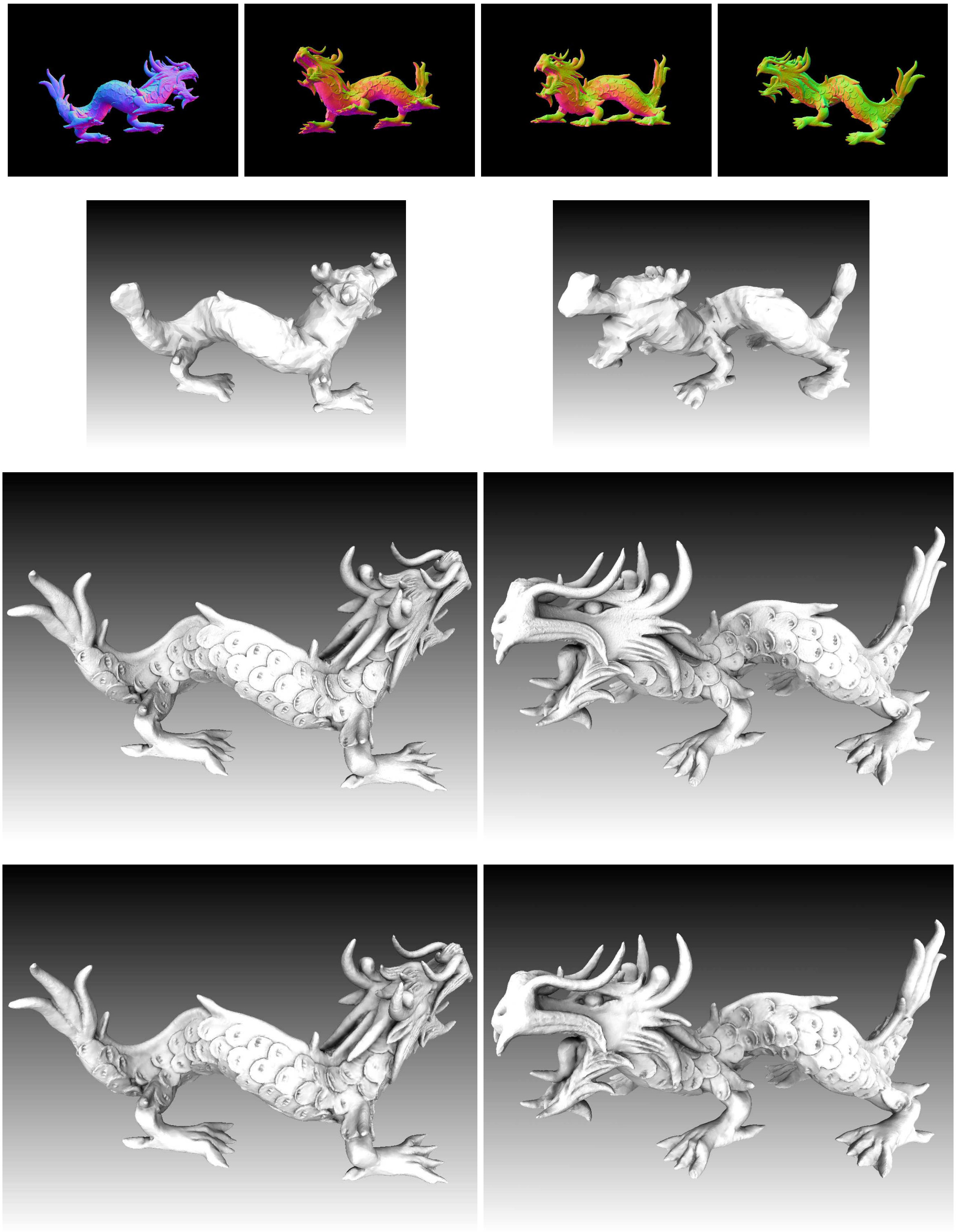

Figure III.11: The dragon sequence (mesh obtained from Stanford repository [1]). Top row: 4 of 24 input images (640×480) showing normal maps of the object of interest; From top to bottom: Initial surface; Ground truth shape; Reconstructed surface. Rendering is performed using ambient vertex occlusion and flat shading so that it displays the surface without missing any details.

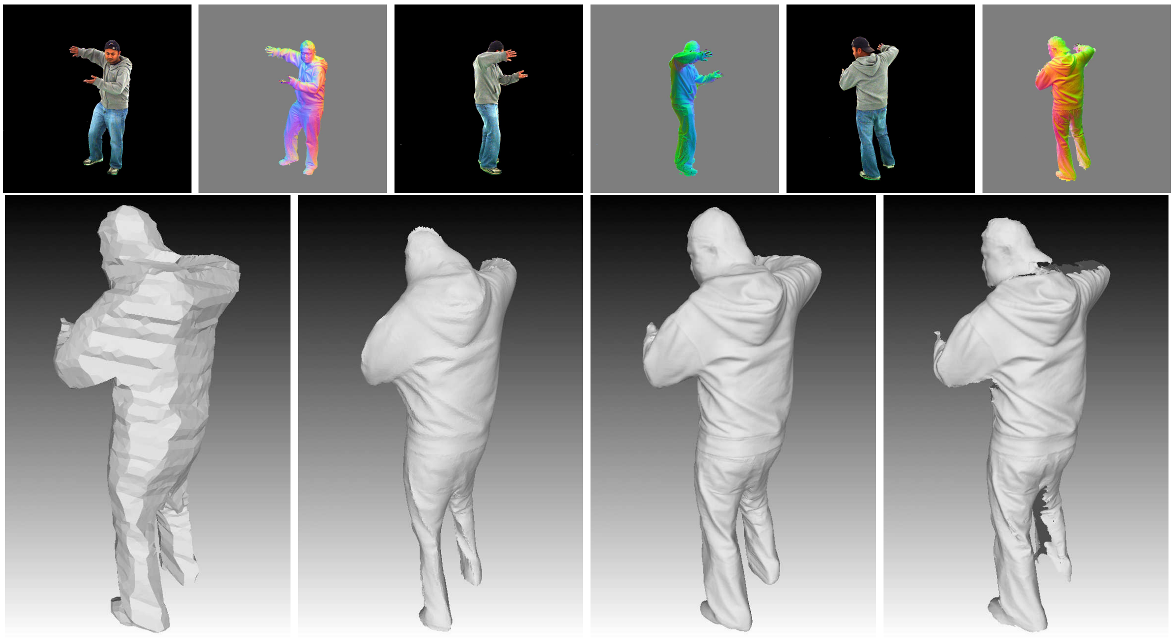

Figure III.12: MIT Sequence (Courtesy of [152]). Top row: 3 of 8 input images and normals (1024×1024). Bottom row, from left to right: visual hull; result using Section III.3 *without* using the horizon term; result using Section III.3 *with* the horizon term; result of [152].

mesh. This emphasize the fact that even though the recovered surface of [152] visually looks really nice, they suffer from the integration bias and registration errors which leads to slightly incorrect 3D positions, as well as incomplete surface recovery. For example, the two images seeing the right ear reproject in different locations, creating a non-coherent textured surface - other problems are shown in red. This might result in wrong visual artifacts for relighting purposes. In contrast, our approach naturally takes advantage of the multi-view information. Also, our approach is purely image-based and does not use pre or post-processing such as re-estimating (and smoothing) the normal maps or performing hole filling like in [152]. Some part like the cap are not well recovered in our case, mainly due to noise and missing normals in the input images, but also by the fact we use a closed surface. Those are still promising results for reconstruction high quality 3D models.

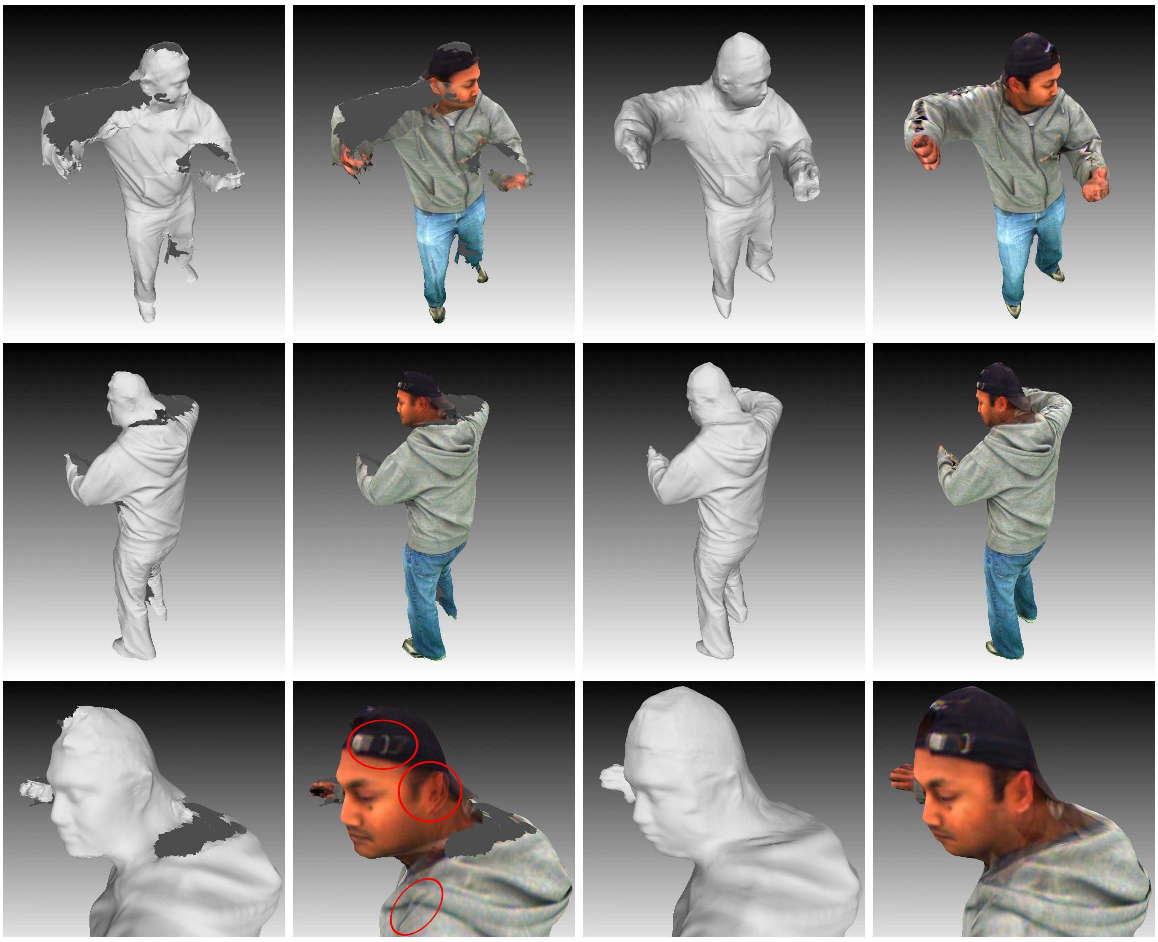

Figure III.13: Results and comparison with [152]. From left to right: final shape of [152]; corresponding mean texture from visible cameras; result with the proposed approach; corresponding mean texture from visible cameras.

III.4 Conclusion

In this chapter we implement deformable meshes in order to solve 3D reconstruction from images problems. The gradient flows described in Chapter II is used in order to optimize triangular meshes via gradient descent. The proposed approach presents several benefits:

- The same framework can equally be applied to a large number of multi-view reconstruction algorithms.
- The energy functional that is minimized can be easily adapted to a particular application, by defining an appropriate error measure exploiting stereo matching, shading, occluding contours or any other cue.
- A coarse-to-fine multi-resolution scheme is adapted so that even being a gradient descent based method, good results can be achieved.

- If the error metric is discriminative enough, a small number of images can be considered to recover full dense and high quality models. We present two example, one for stereo using only six cameras and an other one exploiting photometric stereo using height cameras. In both cases full dense models are reconstructed.

- The approach is flexible with the camera model being used. Even though pinhole camera models were used here, this can straightforwardly be applied to many others such as push-broom cameras.

- Contrary to previous works, during the evolution, *we correctly deal with visibility changes* by expressing the exact gradient of the reprojection error functional. In particular, exactly as in the continuous case [56], this forces the contour generators of the surface to appear at their correct location in the images and reduces the minimal surface bias from which variational methods suffer. This is particularly important in photometric approaches where flows tend to oversmooth the solution.

- Gradient computation is done directly with respect to the discrete representation of the surface based on triangular meshes. This allows for coherent gradient flows that tend to place the mesh vertices to their correct locations and make triangle edges match with the data.

- Triangular meshes offer more flexibility and allow to obtain higher quality models than previous similar approaches using Level Sets, while using the same variational models [80, 79, 163].

Discussion

While not being perfect, the results presented in this chapter illustrate the fact 3D models can be recovered with simple energy functionals and simple gradient descent. This leads to several difficulties and limitations, but also to place for further improvements:

- The presented error measures are simple since this is just the sum of squared differences between pixel intensities. One may define a more robust cost measure to improve robustness such as for example [131] who already proposed a global error metric based on cross-correlations that has even been used using triangle meshes [154].

- While all experiments start from a simple sphere or a visual hull, many approaches have been proposed to better initialize 3D meshes, for example from point clouds [75, 95]. This would speed up the optimization and avoid local minimas.

- The way the surface is optimized might be improved since it uses simple L^2 gradient descent. One may change the gradient metric, make the functional convex, or change the optimization algorithm in order to improve robustness or speed. Changing the gradient metric allows to get more coherent gradient descent flows [24, 174]. This can be easily applied since that corresponds to a straightforward extension of [39] that already does it using triangle meshes.

- While this work focus more on the modeling part, there is place for implementation improvements especially via robust estimators, parallelism and GPU computations.

- The thesis however shows that it can be successfully applied for mesh refinement in a variety of cases, including geometric flows that depends on the surface's normal and/or visibility. Moreover one may adapt its error metric g for concerned applications - for example adding volumetric flows such as silhouettes constraints or edge attachment weight as done in geodesic active contour methods. In all cases adapting the metric is straight forward and its minimization via L^2 gradient descent is a direct application of the presented approach.

- Finally, in this work we used deformable meshes which imply remeshing (in particular for topology changes) during the optimization. In this context, removing points might slightly change the objective functional. One additional improvement would be to change the remeshing algorithm in order to completely ensure the spatial consistency in the optimization process, but this is out of the scope of this thesis and is still an open research area.

CHAPTER IV

Segmentation as a 3D Reconstruction Prior

Multi-view shape reconstruction usually exploits the redundancy of information in the images, either by the fact several views of a same point are available, or that appearances under different lighting conditions can be observed. In this chapter we take a more surface-centered strategy, where we have a look at the redundancy of information such as color, texture or reflectance that can be exploited directly on the surface itself. By using the fact that objects are defined by a finite set of materials, we can segment those properties and classify them on the shape.

This chapter reviews recent advances in variational multi-labels segmentation using convex relaxation. Those global methods have been proved to be robust to both initialization and noise. We apply those methods to segment piecewise constant regions on 3D shapes. We show in particular how this can be implemented on triangular surface meshes via gradient descent methods. We finally discuss how this could be integrated in photometric-based multi-view reconstruction in order to constrain reflectance estimation for instance.

Contents

Being on the image domain or the shape itself, segmentation offers semantic information about the scene. This decomposition of the scene can be used to improve 3D reconstruction algorithms. For example, visual hull based techniques rely on silhouettes, which is the segmentation of the object of interest over images. Visual hull can also be seen as a segmentation of the 3D space [159]. One good example is body reconstruction. The skin of the person, on his arms, hands, or on his face, share the same reflectance properties. Then global segmentation approaches offer significant advantage towards local methods because all those regions are considered in the same cluster. Following the same idea, a shirt or a pant is usually composed of one or two reflectance models.

In the previous chapters, we have seen that 3D reconstruction from images can be solved by minimizing a proper energy functional. This energy depends on an error metric, which is generally designed for particular reconstruction purposes. For example, one could use a proper photo-consistency measure for textured objects, or use a photometric cost designed for smooth uniform Lambertian surfaces. However, real-world objects are not composed of perfect textured surfaces, or purely uniform Lambertian materials. Real-world objects are a mix of those type of reflectance since they are usually composed of a finite number of materials. In that context, segmentation can be used on the scene to tell the algorithm what kind of information to use. For example in the Lambertian case, a binary segmentation algorithm can distinguish textured and textureless areas. Then textured parts could be recovered using multi-view stereo, while textureless areas could exploit shape from shading techniques.

The same idea could be applied to reflectance models instead of reconstruction models. Unfortunately, there does not exist a single reflectance model that encompasses all kind of object materials. We saw in Section I.1 that depending on the kind of objects one wants to reconstruct, models are valid under certain assumptions (mainly depending on specular/diffuse and smooth/rough surface properties). Then, if segmentation can be performed to identify what models are best to use, algorithms can be adapted to concerned parts of the scene. Though this is our main motivation to exploit segmentation as an additional prior for models and algorithms, this chapter focuses on segmentation of data on surfaces. We describe and give ideas on how to exploit this for improving 3D reconstruction algorithms, but further experimental results and analysis will be part of future work.

In the following we describe recent advances in image segmentation, as well as how this can be applied to segment surfaces in 3D. We also describe how incorporating semantic information can be useful to improve 3D reconstruction, or make some algorithms possible, such as multi-view shape from shading of scenes composed of several albedos, like for instance in Jin et al. [83]. In particular we are interested in global segmentation approaches, from which recent techniques offer convex solutions. This is important because if the reflectance models are known, it assures consistency in the segmentation (for example across multi-view images or in the temporal domain). After describing how those methods can be applied on images, we show one way to solve it on arbitrary 2D manifolds. In particular, since triangular meshes are now popular, and is the base of our work in previous chapters, we show how to perform segmentation using gradient descent based approaches.

IV.1 Global Minimization Models for Image Segmentation

Image segmentation aims to partition a given image into several meaningful regions based on certain attributes such as intensity, texture, color, etc. This problem is one of the most challenging and important problems in computer vision, as well as many other fields of image processing such as medical imaging. Many approaches have been proposed to solve image segmentation problems. In particular, via gradient descents, variational methods for image segmentation have had a great success, such as snakes [85], geodesic active contours [18], geodesic active regions [124] and the Chan-Vese models [21]. See for example Cremers et al. [29] for a recent review of different approaches. Yet, the main drawback of those methods is the existence of local minima due to the non-convexity of the energy functionals. As a consequence, minimizing those functionals by gradient descent methods makes the initialization critical.

To obtain global minima, some previous image segmentation works have used different optimization techniques: For example the graph-cuts in a fully discrete setting, see [13, 88, 93] and the references therein. Nevertheless, while binary segmentation methods based on graph-cuts assure to get a global minima, multi-region segmentation algorithms are based on sequences of graph-cuts which cannot guarantee a global optimization.

Recently, some authors have tried to handle the problem in another direction. Instead of working on the optimization techniques in order to compute the minima of non-convex problems, they have reformulated the energy in order to get a convex problem [5, 14, 19, 20, 22, 102, 128, 166]. These segmentation techniques are based on TV-regularizers and aim at finding characteristic functions that minimize the objective functions. Obtaining *global minima* becomes easy and can be done by simply performing a gradient descent. Also the *initialization problem vanishes*: the algorithm can start from any initialization and obtains the same result. The multi-region segmentation models proposed by [19, 102, 166] are rather similar, the work of Pock et al. [128] (inspired form Ishikawa's [74]) differs from the fact that it deals with ordered labels and uses a regularization term which favors transitions between nearby labels. This makes sense in their stereo application where the ordering is due to depth, but it is not the case when we deal with independent labels. Here, we adopt the model of [19, 102, 166] which is more appropriate to the applications we have in mind (the fact that real-life scenes are made by a finite number of independent materials in 3D reconstruction problems) and we adapt this *image* labeling model to *manifolds*.

IV.1.1 Convex Binary Segmentation

In this section, we describe the convex image segmentation model we will use in this thesis. To make this model comprehensible and intuitive, let us first remind of the region-based active contour model of Chan and Vese [21]. Here we show that the energy functional of Chan and Vese, which is the piecewise constant case of the Mumford-Shah model ([113]), can be recast as a convex functional in order to find the global minimizer of the original energy functional.

The Chan-Vese model [21], which is formulated in the level set framework, partition a given image into two subregions. For a given image I, the idea is to find a subset Σ of a bounded domain $\Omega \subset \mathbb{R}^N$, whose boundary $\partial\Sigma$ is represented by the zero level set of function $\phi : \Omega \to \mathbb{R}^N$. This is done by minimizing the energy functional:

$$\min_{\phi,c_1,c_2} \int_\Omega \Big\{ H_\varepsilon(\phi)(I(\mathbf{p}) - c_1)^2 + (1 - H_\varepsilon(\phi))(I(\mathbf{p}) - c_2)^2 + \lambda \, |\nabla H_\varepsilon(\phi)| \Big\} \, \mathrm{d}\mathbf{p} \,, \qquad \text{(IV.1)}$$

where $\lambda \in \mathbb{R}$, $c_1, c_2 \in \mathbb{R}$ and H_ε is a regularized Heaviside function, which models a characteristic function (see [21]).

Since the energy functional (IV.1) is not convex, minimizing it by gradient descent methods may result in reaching local minima. By relaxing the characteristic function $H_\varepsilon(\phi)$ by an arbitrary function y bounded between 0 and 1, Chan et al. [22] showed that minimizing (IV.1) can be rewritten as the following convex minimization problem:

$$\min_{0\leq y\leq 1} \Big\{ \int_\Omega \Big\{ y(\mathbf{p})(I(\mathbf{p}) - c_1)^2 + (1 - y(\mathbf{p}))(I(\mathbf{p}) - c_2)^2 \Big\} \, \mathrm{d}\mathbf{p} + \lambda \int_\Omega |\nabla y| \, \mathrm{d}\mathbf{p} \Big\}, \qquad \text{(IV.2)}$$

c_1 and c_2 being fixed, in $\mathbb{R}$. As proved in [14, 22], if $y(x)$ is a minimizer of (IV.2), then for a.e. $\mu \in [0, 1]$, the set $\Sigma(\mu) = \{x \in \Omega, y(x) > \mu\}$ is a minimizer of the Mumford-Shah functional [113], implying that *the solution to (IV.1) can be obtained by thresholding y at any arbitrary threshold between* 0 *and* 1. The last term of Equation (IV.2) is called the Total Variation of y.

IV.1.2 Multi-Region Segmentation of an Image

Recently, several authors [19, 102, 166] have extended the convex formulation (IV.2) to multi-region segmentation:

$$\min_{\mathbf{y}\in K} \Big\{ \int_\Omega < \mathbf{y}(\mathbf{p}), \mathbf{s}(\mathbf{p}) > + \lambda \, |\nabla \mathbf{y}(\mathbf{p})| \, \mathrm{d}\mathbf{p} \Big\}, \qquad \text{(IV.3)}$$

where K^m is the set of function $\mathbf{y} : \Omega \to \mathbb{R}^m$ such that

$$\Big\{ \forall \, \mathbf{p} \in \Omega, \, \forall \, p \in [1..m] : \; \mathbf{y}_p(\mathbf{p}) \geq 0, \, \sum_{p=1}^{m} \mathbf{y}_p(\mathbf{p}) = 1 \Big\} \,. \qquad \text{(IV.4)}$$

K^m is defined on a standard m-simplex and basically represents the convex hull of $(m + 1)$ points in an Euclidean space of dimension m or higher. $|\nabla \mathbf{y}(\mathbf{p})|$ corresponds to $\sqrt{\sum_p |\nabla \mathbf{y}_p(\mathbf{p})|^2}$, where $|.|$ denotes the L^2 norm. m denotes the number of labels and $\mathbf{s}(\mathbf{p})$ is an m-dimensional

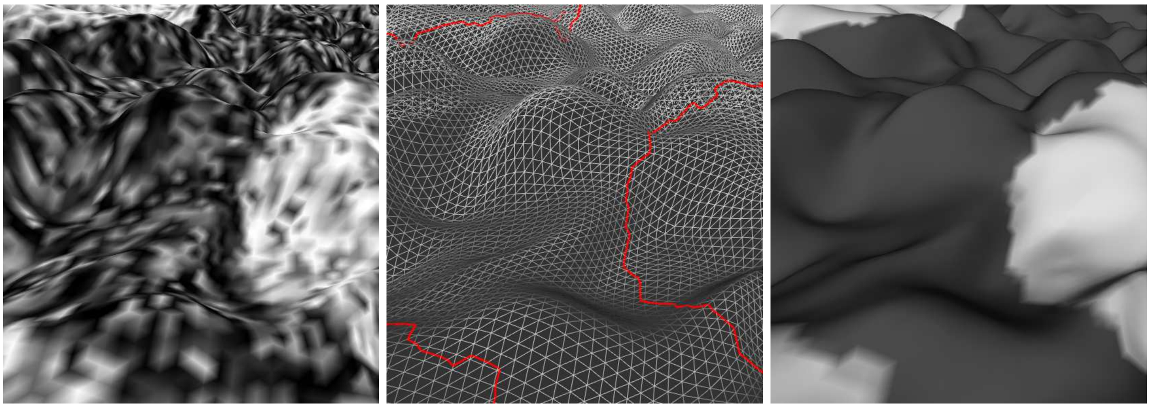

Figure IV.1: Example of segmentation on a manifold. (a) The Input textured surface. (b) The triangular representation of the surface with the retrieved contour (in red). (c) Surface colored with the mean values of the segmented regions (and surface shading).

vector; $s_p(\mathbf{p})$ indicates the affinity of the data at point $\mathbf{p}$ with class p. From a statistical point of view, it represents the likelihood probability to belong to that class. The convex domain naturally allows direct competition between the labeling.

IV.2 Segmentation on Manifolds via Convex Relaxation

In this section, we address the problem of segmenting *data* defined on *manifolds* (typically a 2-surface in $\mathbb{R}^3$) into a set of multiple regions of uniform properties such as piecewise constant attributes. The ability to solve such a problem offers significant new possibilities in a number of applications. For example, in 3D reconstruction (see Jin et al. [83]), a segmentation into piecewise constant data of the reconstructed surface allows to naturally introduce constraints on the materials of the scene.

In particular, we propose a numerical method when the manifold is represented by a triangular mesh. Based on recent image segmentation models, our method minimizes a convex energy and then enjoys significant favorable properties: it is robust to initialization and avoid the problem of the existence of local minima present in many variational models. The contributions of this chapter are threefold: firstly we adapt the convex image labeling model to manifolds; in particular the total variation formulation. Secondly we show how to implement the proposed method on triangular meshes, and finally we show how to use and combine the method in other computer vision problems, such as 3D reconstruction. We demonstrate the efficiency of our method by testing it on various data.

IV.2.1 A Convex Multi-Region Segmentation Formulation on Manifolds

We extend the multi-region convex model (IV.3) defined in the previous Section IV.1.2 on a manifold, and we show how to optimize the associated energy for a surface represented by a mesh. To our best knowledge, these convex formulations (IV.2,IV.3) have been defined only on open subsets of $\mathbb{R}^2$ which correspond to image domains, as described in the previous section.

Let $\mathcal{S}$ be a Riemannian manifold. Typically, $\mathcal{S}$ could be a smooth 2D surface of $\mathbb{R}^3$. Energy (IV.3) is adapted as follows:

$$\min_{\mathbf{y} \in K} \left\{ \int_{\mathcal{S}} < \mathbf{y}(\mathbf{x}), s(\mathbf{x}) > + \lambda |\nabla_{\mathcal{S}}\, \mathbf{y}(\mathbf{x})| \,\mathrm{d}s \right\}, \tag{IV.5}$$

where now the functions $\mathbf{y}$ are defined on $\mathcal{S}$ instead of Ω, $|.|$ is the Riemannian norm, $\nabla_{\mathcal{S}}$ is the intrinsic gradient on $\mathcal{S}$ and $\mathrm{d}s$ is the manifold's element measure (surface's area measure for 2D manifolds).

Now, let us consider a manifold represented by a mesh. The following results apply to manifolds with any topology. Let $\mathbf{X}$ be a (piecewise linear) polyhedron representation of the surface $\mathcal{S}$, defined by a set of vertices $\mathbf{x}_k : \mathbf{X} = \{\mathbf{x}_k\}$ and let l be the cardinality of $\mathbf{X}$ (the number of vertices). As in the finite elements literature, we define $\phi_k : \mathcal{S} \to \mathbb{R}$ as the piecewise affine, interpolating basis function such that $\phi_k(\mathbf{x}_k) = 1$ and $\phi_k(\mathbf{x}_i) = 0$ if $i \neq k$. The vector valued field $\mathbf{Y} = \{\mathbf{y}_k\}$ is defined on all vertices $\mathbf{x}$ of the polyhedron $\mathbf{X}$. $\mathbf{Y}$ can be naturally extended on $\mathcal{S}$ by a piecewise affine vector valued field on $\mathcal{S}$. We denote this extension $\mathbf{y}(\mathbf{x}) = \sum_k \mathbf{y}_k \phi_k(\mathbf{x})$. To make the thesis easier to read and because of space limitations, we assume that the manifold is a 2D surface of $\mathbb{R}^3$. However, the following method applies to any dimension. Let $\mathcal{S}_j$ be the j^{th} triangle of the mesh. The multi-region segmentation energy can then be rewritten as

$$\sum_j \sum_k \left\langle \mathbf{y}_k, \int_{\mathcal{S}_j} \phi_k(\mathbf{x}) s(\mathbf{x}) \mathrm{d}s \right\rangle + \lambda \int_{\mathcal{S}_j} |\nabla_{\mathcal{S}}\, \mathbf{y}(\mathbf{x})| \mathrm{d}s, \tag{IV.6}$$

where $\mathbf{y}$ is constrained to be in K. The first term of (IV.6) is explicitly written with respect to $\mathbf{Y}$. In order to make the total variation term explicit with respect to $\mathbf{Y}$, we first consider a local paramtrization (u, v) on the manifold. Following [34, 78], we rewrite the right term of Equation (IV.6) using fundamental forms:

$$\nabla_{\mathcal{S}}\, \mathbf{y} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial u} & \frac{\partial \mathbf{x}}{\partial v} \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_v \end{bmatrix} \text{, and then}$$

$$|\nabla_{\mathcal{S}}\, \mathbf{y}| = \sqrt{\begin{bmatrix} \mathbf{y}_u & \mathbf{y}_v \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_v \end{bmatrix}},$$

where $\mathsf{E} = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial \mathbf{x}}{\partial u}$, $\mathsf{F} = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial \mathbf{x}}{\partial v}$ and $\mathsf{G} = \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial \mathbf{x}}{\partial v}$ are coefficients of the first fundamental form (see [34, 78]). $\mathbf{y}_u$ and $\mathbf{y}_v$ are partial derivatives of $\mathbf{y}$ with respects to u and v respectively. Considering the mesh representation, we parametrize the triangle $\mathcal{S}_j$ by $\mathbf{x}(u, v) = \mathbf{x}_{j,1} + u\, \overrightarrow{\mathbf{x}_{j,1}\mathbf{x}_{j,2}} + v\, \overrightarrow{\mathbf{x}_{j,1}\mathbf{x}_{j,3}}$ where $\mathbf{x}_{j,1}$, $\mathbf{x}_{j,2}$ and $\mathbf{x}_{j,3}$ are the three vertices associated with the triangle $\mathcal{S}_j$ and where $(u, v) \in T = \{(u, v) | u \in [0, 1] \text{ and } v \in [0, 1-u]\}$.

We then have $\int_{\mathcal{S}_j} |\nabla_{\mathcal{S}}\, \mathbf{y}(\mathbf{x})| \mathrm{d}s =$

$$\int_T \sqrt{\sum_p \mathbf{y}_{p_u}^2 \mathsf{G} - 2\mathbf{y}_{p_u} \cdot \mathbf{y}_{p_v} \mathsf{F} + \mathbf{y}_{p_v}^2 \mathsf{E}}\ \mathrm{d}u\mathrm{d}v\ . \tag{IV.7}$$

$\mathbf{y}_{p_u}$ and $\mathbf{y}_{p_v}$ are partial derivatives of $\mathbf{y}_p$ with respects to u and v respectively. Here the reader will easily verify that E, F, G, $\mathbf{y}_{p_u}$ and $\mathbf{y}_{p_v}$ are constant functions on $\mathcal{S}_j$ and that their respective values are equal to $\mathsf{E}_j = |\mathbf{x}_{j,2} - \mathbf{x}_{j,1}|^2$, $\mathsf{F}_j =< \mathbf{x}_{j,2} - \mathbf{x}_{j,1}, \mathbf{x}_{j,3} - \mathbf{x}_{j,1} >$, $\mathsf{G}_j = |\mathbf{x}_{j,3} - \mathbf{x}_{j,1}|^2$, $\mathbf{y}_{p_u}^{j} = \mathbf{y}_{j,2_p} - \mathbf{y}_{j,1_p}$ and $\mathbf{y}_{p_v}^{j} = \mathbf{y}_{j,3_p} - \mathbf{y}_{j,1_p}$, where $\mathbf{y}_{j,1}$, $\mathbf{y}_{j,2}$ and $\mathbf{y}_{j,3}$ are the values of $\mathbf{y}$ at vertices $\mathbf{x}_{j,1}$, $\mathbf{x}_{j,2}$ and $\mathbf{x}_{j,3}$ respectively. Now the term inside the integral of (IV.7) does not depend on u and v. The convex multi-region segmentation energy on the meshed manifold becomes:

$$\begin{aligned} E(\mathbf{Y}) = \sum_j \sum_k \left\langle \mathbf{y}_k, \int_{\mathcal{S}_j} \phi_k(\mathbf{x}) s(\mathbf{x}) \mathrm{d}s \right\rangle & \\ + \frac{\lambda}{2} \sum_j \sqrt{\sum_p {\mathbf{y}_{p_u}^{j}}^2 \mathsf{G}_\mathsf{j} - 2\mathbf{y}_{p_u}^{j} \cdot \mathbf{y}_{p_v}^{j} \mathsf{F}_\mathsf{j} + {\mathbf{y}_{p_v}^{j}}^2 \mathsf{E}_\mathsf{j}}\ . \end{aligned} \tag{IV.8}$$

IV.2.2 Gradient Descent Method

When the (surface) manifold is represented by a mesh, the convex multi-region segmentation model then leads to optimizing the convex energy (IV.8) with respect to $\mathbf{Y} \in \mathbb{R}^{l \times m}$, with the convex constraint $\mathbf{Y} \in K$; K being the set $\{\mathbf{Y} \text{ s.t. } \forall k,\ \sum_p \mathbf{y}_{kp} = 1 \text{ and } \forall p,\ \mathbf{y}_{kp} \geq 0\}$. This convex constrained optimization problem on $\mathbb{R}^{l \times m}$ can be solved by the projected gradient method [12], which consists in generating the sequence U^t via:

$$\mathbf{Y}^{t+1} = Proj_K(\mathbf{Y}^t - \tau \nabla E(\mathbf{Y}^t))\ , \tag{IV.9}$$

for a fixed time step $\tau > 0$, until $|\mathbf{Y}^t - \mathbf{Y}^{t-1}|_\infty \leq \delta$, a small constant. $Proj_K$ is the projection on the convex set K. In other words, we iteratively process gradient descent steps and projections of the $\mathbf{y}_k$ on the set K. One interesting property of simplicies like the set K is that they are composed of simplicies of lower dimensionality, and then their projection can be computed recursively using a low dimensional set where projection is easy, such as triangle, segments or points. These projections can be done via Michelot's algorithm [109] that uses that property. From energy (IV.8) we easily obtain:

$$\nabla \frac{\partial E}{\partial \mathbf{y}_{kp}}(\mathbf{Y}) \quad = \quad \sum_{j \in N(k)} \left[\int_{\mathcal{S}_j} \phi_k(\mathbf{x}) s(\mathbf{x}) \mathrm{d}s \right]_p \quad - \quad \frac{\lambda}{2} \quad Q \quad (\xi \quad + \quad \varepsilon)^{-\frac{1}{2}}, \tag{IV.10}$$

where $Q = (\mathbf{y}_{j,2_p} - \mathbf{y}_{kp})(\mathsf{G}_j - \mathsf{F}_j) + (\mathbf{y}_{j,3_p} - \mathbf{y}_{kp})(\mathsf{E}_j - \mathsf{F}_j)$, ξ is the term in the squared root of (IV.8), and $N(k)$ is the 1-ring neighborhood of vertex k. Figure IV.2 shows an example of a 3-simplex, with random points and their projections onto the simplex. As in [22], we regularize the term ξ by incorporating a small value ε inside the squared root to avoid instabilities when the gradient of $\mathbf{y}$ is 0.

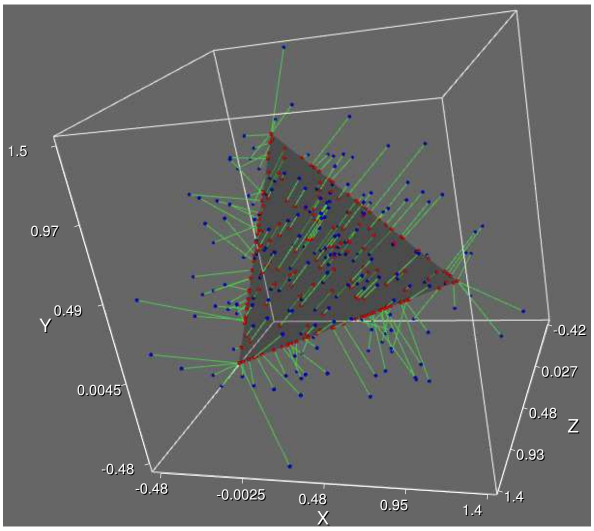

Figure IV.2: An example of a 3-dimensional convex set (a standard 3-simplex). Random points in $\mathbb{R}^3$ (blue) are projected onto the simplex (into red points) using recursive projection on sub-simplicies.

Then from Equation (IV.10), we can obtain the complete L^2 gradient by following Chapter II.

IV.3 Applications with Image-Based Segmentation Cues

IV.3.1 Applications

In the previous sections the data term of the segmentation model s is assumed to be known (Equation IV.5). This term depends on parameters than can eventually be estimated beforehand, in cases where the distribution models are known in advance (such as mean colors, Gaussian radiance distribution and so on). In the applications, this term can also be estimated, and depends on some parameters that have to be optimized. However we will consider that the number of regions to segment is known in advance, similarly as in K-Means approaches. The convex problem can be solved by alternating optimization of the parameters in a bi-convex way. For fixed parameters of s we update $\mathbf{y}$ and vice-versa. $\mathbf{y}$ is updated according to the method presented in section IV.2.2. In practice we update the parameters of s every r update iterations of $\mathbf{y}$ (r is chosen arbitrary; we fix $\mathsf{r} = 10$ in our experiments).

Piecewise Constant Data Segmentation

Let us consider the case where the data we want to segment are assumed to be piecewise constant. Here a natural expression for $s_p(\mathbf{x})$ is to use the squared error between the scalar or vector-valued data $C(\mathbf{x})$ at the point $\mathbf{x}$ and the value μ_p associated with the label p (μ_p having the same dimension as the data):

$$s_p(\mathbf{x}) = (C(\mathbf{x}) - \mu_p)^T (C(\mathbf{x}) - \mu_p) \ .$$

The optimization of the energy (IV.5) with respect to μ_p gives:

$$\mu_p = \frac{\int_{\mathcal{S}} \mathbf{y}_p(\mathbf{x})\ C(\mathbf{x}) \mathrm{d}s}{\int_{\mathcal{S}} \mathbf{y}_p(\mathbf{x}) \mathrm{d}s} \ ,$$

which corresponds to the mean value of the data of the associated region. Note that the previous model can be easily extended to any probability density function D_p. For example, D_p can be a multivariate Gaussian density function of mean μ_p and covariance Σ_p, and then we would have:

$$s_p(\mathbf{x}) = -\ln(D_p(\mathbf{x}, \mu_p, \Sigma_p)) \ , \text{ with}$$
$$D_p(\mathbf{x}, \mu_p, \Sigma_p) = \frac{1}{\sqrt[m]{2\pi |\Sigma|}} e^{-\frac{1}{2}(C(\mathbf{x}) - \mu_p)^T \Sigma_p^{-1} (C(\mathbf{x}) - \mu_p)} \ .$$

Segmentation in 3D Reconstruction Problems

Such segmentation framework can be incorporated in 3D Reconstruction applications. In such applications, it can be interesting to segment a particular region, or all parts of the surface sharing the same reflectance properties. In 3D reconstruction, most of the variational methods yield to minimizing an energy of the form

$$E(\mathcal{S}) = \sum_i \int_{\mathcal{S}} g(\mathbf{x}) \frac{\mathbf{x}_i \cdot \mathbf{n}}{\mathbf{x}_{i,z}^3} \nu_{\mathcal{S}}(\mathbf{x}) \ \mathrm{d}s \ , \tag{IV.11}$$

see for example [32, 79, 78]. Moreover, if we choose

$$g(\mathbf{x}) = \sum_{p=1}^{m} \mathbf{y}_p(\mathbf{x}) (I_i(\pi_i(\mathbf{x})) - \mu_p)^T (I_i(\pi_i(\mathbf{x})) - \mu_p) \ ,$$

where $\pi_i(\mathbf{x})$ is the projection of the surface point $\mathbf{x}$ into the i^{th} image and $I_i : w \mapsto I_i(w)$ is the function which associates to each pixel w, its color on the i^{th} image. We then get an extension of the stereoscopic segmentation method proposed by [159] to the case where the surface is composed of more than two regions of piecewise constant radiance. Also, contrary to our method, the segmentation approach proposed in [159] is subject to local minima. Finally, the optimization of the energy (IV.5) with respect to μ_p gives:

$$\mu_p = \frac{\int_{\mathcal{S}} \mathbf{y}_p(\mathbf{x})\ \sum_i I_i(\pi_i(\mathbf{x})) \frac{\mathbf{x} \cdot \mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) \mathrm{d}s}{\int_{\mathcal{S}} \mathbf{y}_p(\mathbf{x}) \sum_i \frac{\mathbf{x} \cdot \mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) \mathrm{d}s} \ .$$

If we chose $g(\mathbf{x}) =$

$$\sum_{p=1}^{m} \mathbf{y}_p(\mathbf{x})(I_i(\pi_i(\mathbf{x})) - \rho_p\, \mathbf{n}(\mathbf{x}) \cdot \mathbf{L})^T (I_i(\pi_i(\mathbf{x})) - \rho_p\, \mathbf{n}(\mathbf{x}) \cdot \mathbf{L}),$$

where $\mathbf{N}(\mathbf{x})$ is the normal to the surface at the point $\mathbf{x}$ and $\mathbf{L}$ is the vector corresponding to the light source illuminating the scene, then we get an extension of the (Lambertian) multi-view shape from shading method proposed by [83] for surfaces with piecewise constant albedo. In the same way, contrary to our approach, the method proposed by [83] is limited to two regions segmentation and is strongly subject to local minima. The optimization of energy (IV.5) with respect to the albedo gives:

$$\rho_p = \frac{\int_{\mathcal{S}} \mathbf{y}_p(\mathbf{x}) \ \sum_i I_i(\pi_i(\mathbf{x}))\mathbf{n}(\mathbf{x}) \cdot \mathbf{L}\dfrac{\mathbf{x} \cdot \mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) \mathrm{d}s}{\int_{\mathcal{S}} \mathbf{y}_p(\mathbf{x})(\mathbf{n}(\mathbf{x}) \cdot \mathbf{L})^2 \sum_i \dfrac{\mathbf{x} \cdot \mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) \mathrm{d}s} \ .$$

The theoretical and experimental study of these algorithms will be the topic of forthcoming works.

IV.3.2 Experiments

In order to validate the proposed multi-region segmentation approach on meshes, we present different experiments on synthetic as well as realistic data. In practice as explained in previous section, the segmentation is solved by alternating between region parameters and the segmentation variable $\mathbf{Y}$, with a known number of regions. The algorithm complexity is linearly dependent on the number of facets and the number of classes. Experiments have been performed on a 2.66GHz linux machine and take about 200 seconds on a mesh of $200{,}000$ facets and for a 4 regions segmentation. The values of λ have been manually chosen in each example but a value of 0.01 gives reasonable results in most cases.

The Two Region Case

Figures IV.3 and IV.4 show examples of our algorithm using a synthetic image mapped onto a mesh for the Stanford bunny model. Noise has been added to the image. Here, we show that our algorithm performs well on the given example and that the final solution is binary. Moreover it is robust to the initialization of the scalar function $\mathbf{Y}$. Note that the retrieved solution that has been displayed is the auxiliary value $\mathbf{y}$, and not the segmented constant values μ_1 and μ_2. Also because the energy functional is convex in $\mathbf{y}$ only and the values μ_1 and μ_2 are optimized during the evolution, they can be assigned to the region corresponding to either $\mathbf{y} = (1, 0)$ or $\mathbf{y} = (0, 1)$, this explains why the last initialization do not show the same values of $\mathbf{y}$ but an inverted one. In practice, although the total functional is not fully convex, we obtain the same results and really similar μ_1 and μ_2 for each example. We respectively obtain ($\mu_1 = 140.778$, $\mu_2 = 231.003$), ($\mu_1 = 140.746$, $\mu_2 = 231.01$), ($\mu_1 = 140.75$, $\mu_2 = 231.03$) and ($\mu_1 = 230.992$, $\mu_2 = 140.765$) for the four different

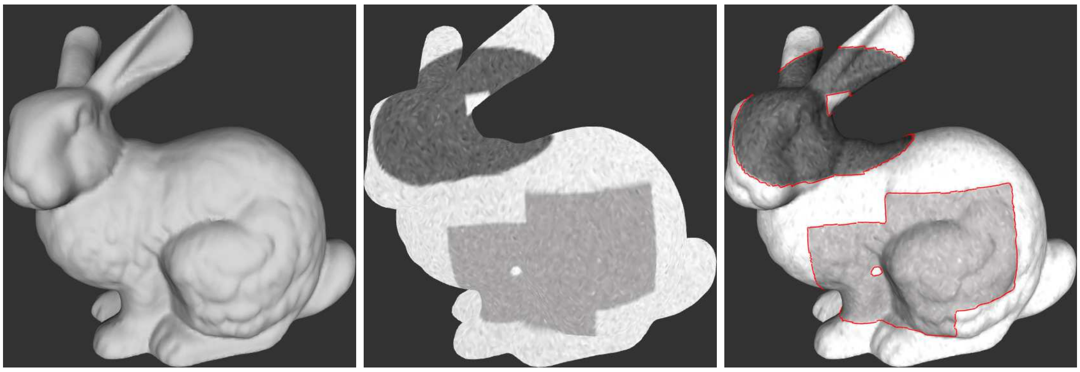

Figure IV.3: Segmentation result on the synthesized Stanford bunny surface. (a) Input shaded object. (b) Input mesh with synthetic texture mapping. (c) Input textured mesh (shaded visualization) and final contour (in red).

initializations. Note that in the last column, values of μ_1 and μ_2 are inverted and the solution $\mathbf{y}$ as well. In this example, geodesic active contours or level sets methods would tend to the closest local minima from the initialization as the texture is not clearly binary. Nevertheless, as the method is global here, segmenting a particular region should be done using additional cues.

Figure IV.5 present segmented surfaces from real-world textures that have been mapped onto a mesh, in the case of the two-phases segmentation. We show different examples from classical images used in segmentation. Note that the segmentation is done on the mesh using the method described in this chapter and not on an image. The experiments show three different non binary images and their segmentation into two different regions. As expected the results are binary even though the initial values of the segments are random values. The mean values of each region is estimated during the process as described before, and the parameters λ can be adjusted to add more smoothness to the segmentation. As shown by experiments, even though the initialization is random and the parameters of each region are computed during the evolution, the algorithm still converges to the desired solution as a binary solution.

Dealing with Multiple Regions

Here we show the efficiency of the proposed method when dealing with multiple regions. Different examples are shown, first with synthetic textures on which noise has been added, and then on meshes textured by real-world images like previous examples. Note that the number of regions is initially given and is not automatically estimated.

In Figure IV.6, the experiment shows noisy texture on meshes, the segmentation result using K-Means, and the result of our TV-based algorithm on meshes. Because the K-Means algorithm does not take into account the spatial coherence of points, the result is noisy. On the other hand, the TV regularization allows coherence in the scene and the segmentation is close to the expected solution. In addition to be robust to initialization, our approach is

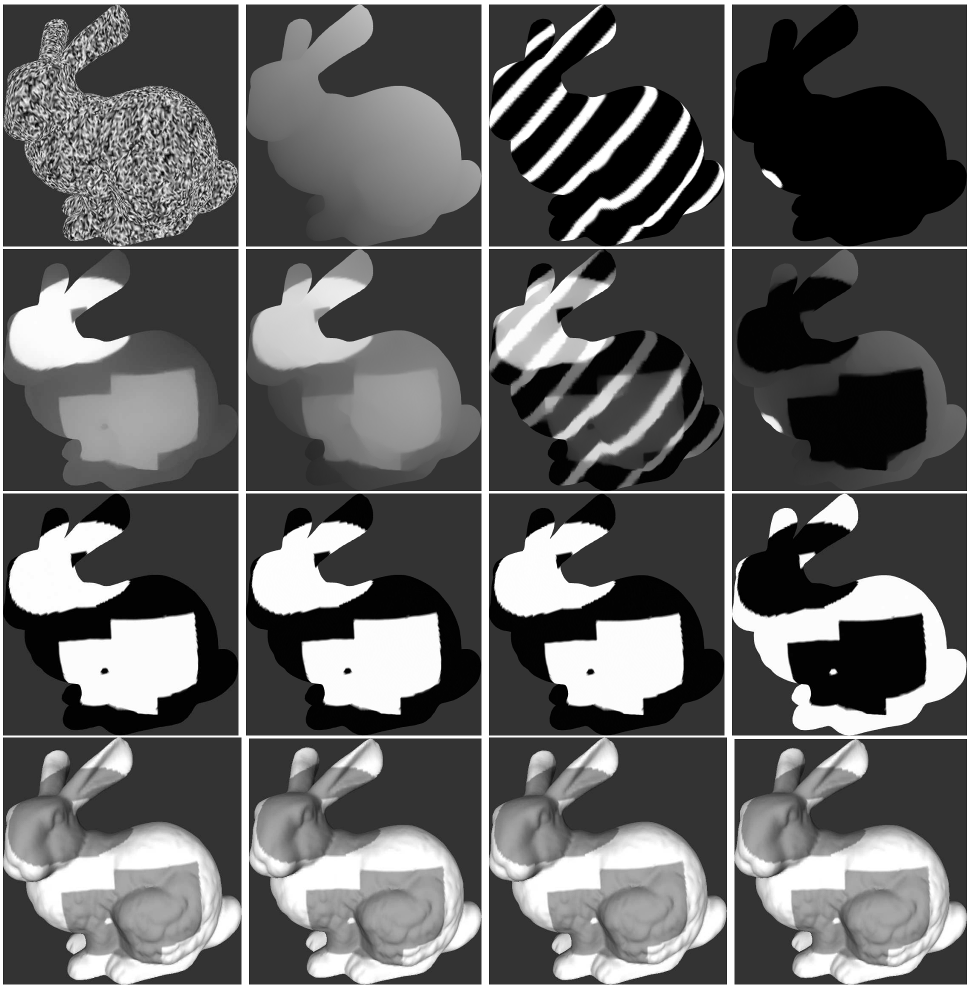

Figure IV.4: The evolution on the synthesized bunny surface. Different initialization of $\mathbf{Y}$ (first row); Intermediate values of $\mathbf{Y}$ (second row); The obtained solution $\mathbf{Y}$ (third row); The obtained mean values (fourth row) with shading.

robust to noise.

We then tested our multi-region segmentation approach on various data from real-world images [107], see Figure IV.7.

Figure IV.8 also shows result on more image regions, in that case six regions. On that figure we also put the histogram of the convex variable $\mathbf{Y}$ onto the convex domain K. Since a 6-Simplex can only be represented in dimension 5 or higher, we represent it as a 6-node connected graph on a 2D plane. Each node represents the vertices of the simplex, and since the final values should be binary, pics should appear at the vertices locations like it is the case in Figure IV.8. Each node represents each class ot fhe segmentation.

As an example, we applied our approach to segment mean curvature on a mesh using three different regions. Figure IV.9 shows that we are able to segment concave and convex parts of the mesh.

Figure IV.5: Segmentation results on meshes in the two-region segmentation on three different examples. From top to bottom: Input textured mesh; Mesh shape where the segmentation is performed and the initial random value of one component of $\mathbf{Y}$; Recovered mean values of each region; Segmented object.

Finally, in Figure IV.10, we show the examples of a 3D mesh obtained by 3D reconstruction algorithms, as the one in [144, 167]. The last row shows the obtained color-based labeling (into three regions). Even though the texture is far from being binary, the segmentation is the expected one. For instance in the result, we nicely recover the skin, the pant and the shirt. Here again, initialization was random. Figure IV.10 shows a graph of the histogram evolution of $\mathbf{Y}$ during the optimization.

For comparison of the convex *image* multi-region segmentation model (IV.3) with other methods, we refer to [102] which shows quantitative and qualitative comparisons with belief propagation, sequential belief propagation, graph cuts with alpha-expansion, graph cuts with alpha-beta swap and sequential tree reweighted belief propagation methods. The experiments show that the generated labeling is comparable to state-of-the-art discrete optimization methods.

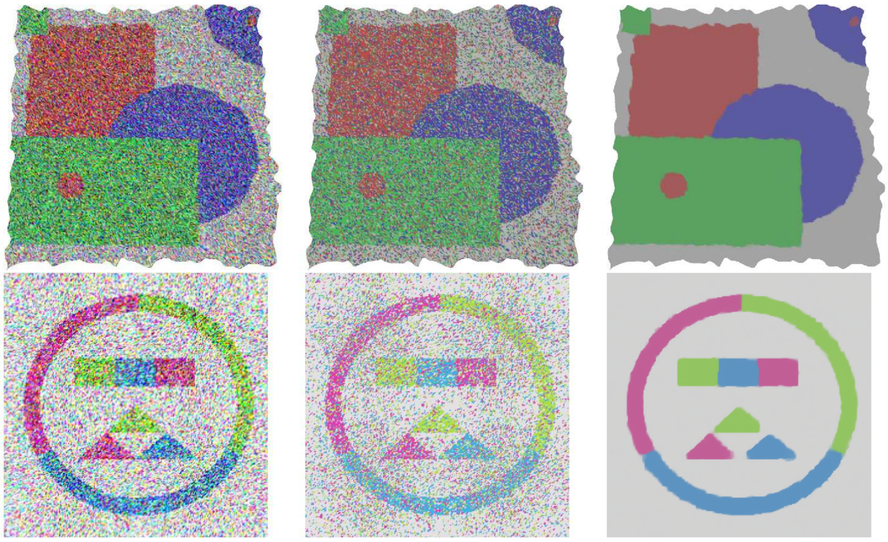

Figure IV.6: Segmentation results on meshes in the multi-region case on synthesized examples inspired from [102]. (a) Input textured mesh (same shape as previous Butterfly and Horses data). (b) Clustering using K-Means algorithm. (c) Recovered mean values of each region obtained by our approach.

Figure IV.7: Segmentation results on meshes in the multi-region case. Top row: Horse data set and its segmentation for three regions. Bottom row: Four regions labeling of the Butterfly data. (a) Input textured mesh; (b) Recovered mean values of each region obtained by our approach; (c) One of the segmented regions.

This chapter shows a proof of concept of multi-labeling on triangular meshes. It works well on the presented examples, however there is still a long way to go to jointly use it with

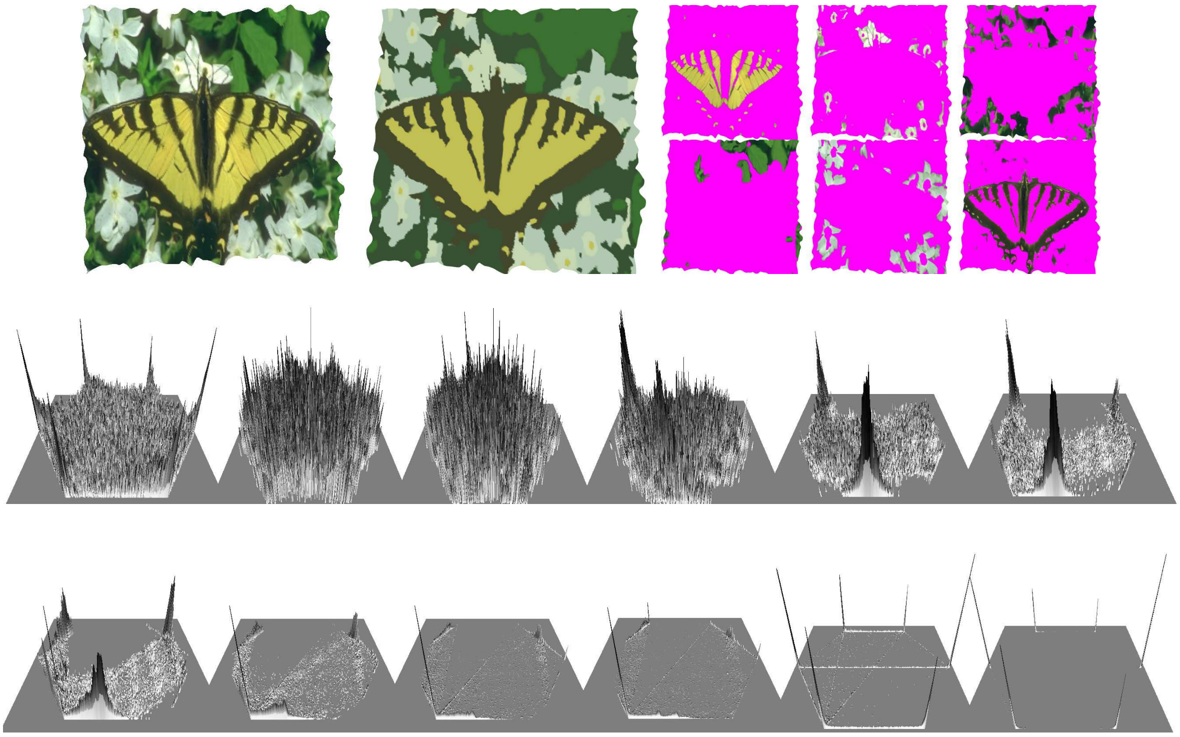

Figure IV.8: Segmentation results and histogram evolution of the convex variable onto a graph representation of the convex set K (a standard 6-simplex here). The six nodes of the graph corresponds to the six values of the mean colors of each region.

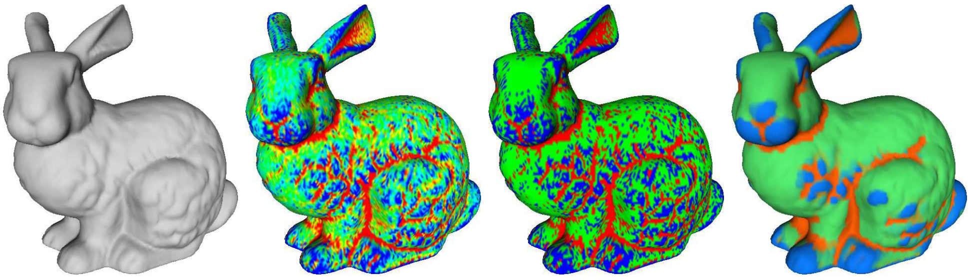

Figure IV.9: Segmentation results on mesh curvature in 3 regions. (a) Input mesh. (b) Mean Curvature visualization. (c) Simple thresholding of the mean curvature. (d) Segmentation result of the mean curvature into three regions with our approach.

3D reconstruction as originally motivated. Unfortunately, the optimization is too slow to alternatively update the shape, the reflectance and the segmentation. This is a direct consequence of using a gradient descent flow, known to be slow for such energy. In fact, recent methods allow to compute the same energies defined over images in the same way more efficiently. In particular Primal-Dual approaches have been recently very promising and fast methods [20, 101]. Alternatively, a mesh being a graph structure, one may suggest to use Graph-cut techniques, whose recent developments in multi-labeling allow optimal solutions

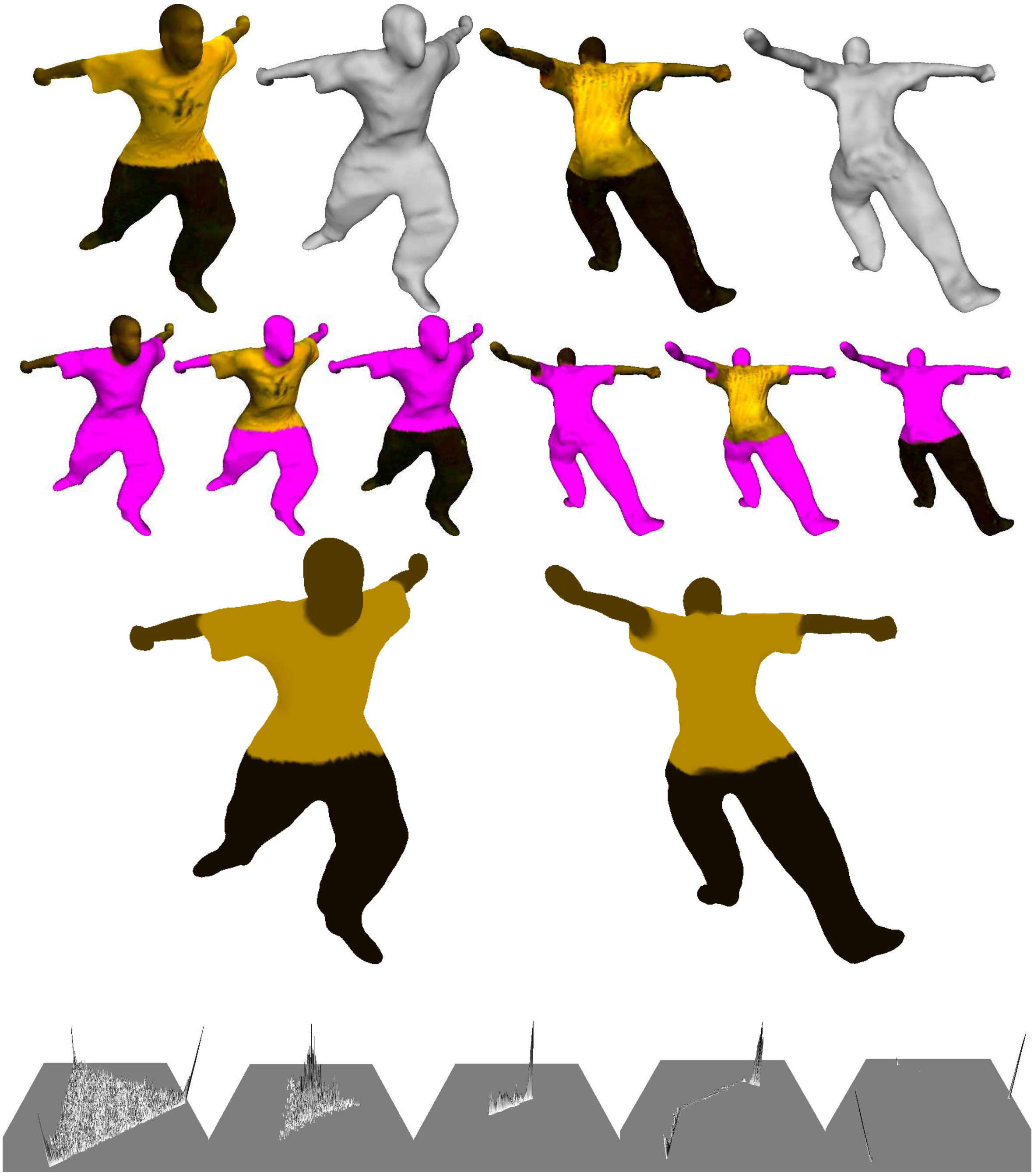

Figure IV.10: Segmentation result on a colored mesh obtained by multiview stereo algorithm. Front view (left) and back view (right). From top to bottom: Original input colored mesh and the associated 3D shape; Result of the segmentation into three regions obtained by our algorithm; Recovered mean values displayed for each region; Histogram of the convex variable.

as well. This would be worth to extend our work to those methods in future experiments.

One other benefit of our work is to show how Total Variation can be performed on triangle meshes. While used in this work as a segmentation regularizer, this could be used as a general regularization for surface properties such as reflectance, scene flow or other. Let $\mathbf{w}$ be a m-dimensional vector defined over a surface $\mathcal{S}$, then this corresponds to minimizing the following equation:

$$E(\mathbf{w}) = \int_{\mathcal{S}} |\nabla_{\mathcal{S}}\, \mathbf{w}(\mathbf{x})| \,\mathrm{d}s \,.$$

Then its gradient flow with respect to $\mathbf{w}$ is a straightforward application of this chapter.

IV.4 Conclusion

In this chapter we propose a variational method for segmenting data on manifolds into regions of constant properties. Based on recent image processing procedures, we adapt methods based on convex relaxation onto surfaces. The convex formulation makes the proposed model robust to initialization. Moreover, the total variation regularizer makes the method robust to noise while preserving region boundaries. We show how to implement the method, in particular how to compute the gradient of the total variation term, when the surface has a discrete representation such as triangular meshes. We have demonstrated the efficiency of our method by testing it on various synthetic and realistic data from computer vision applications.

CHAPTER V

Multi-view Helmholtz Stereovision: Towards 3D Modeling of Arbitrary Objects

Reconstructing objects of non-Lambertian reflectance is a difficult task as discussed earlier and seen in previous chapters. Several approaches have been proposed to face those limitations more or less elegantly. One interesting approach is to exploit BRDF properties and in particular the fact that it is reciprocal. This is called *Helmholtz reciprocity* and has been previously exploited in 3D reconstruction algorithms and referred as Helmholtz Stereopsis.

Helmholtz stereovision algorithms are limited to binocular stereovision or depth maps reconstruction. In this chapter, we extend these methods to recover the full 3D shape of the objects of a scene from multiview Helmholtz stereopsis. Thus, we are able to reconstruct the complete three-dimensional shape of objects made of any arbitrary and unknown bidirectional reflectance distribution function. Unlike previous methods, this can be achieved using a full surface representation model. In particular occlusions (self occlusions as well as cast shadows) are easier to handle in the surface optimization process. As described in previous chapters, we use a triangular mesh representation which allows to naturally specify relationships between the geometry of a point of the scene and its surface normal. We show how to implement the presented approach using the coherent gradient descent flow previously presented in Chapter II.

Contents

V.1 Introduction

Reconstructing shape and appearance of objects from images is still one of the major problems in computer vision and graphics. In this work, we are interested in recovering a full and dense representation of the object's three-dimensional shape. Multi-view reconstruction systems are commonly used to estimate such a model, as they provide information from many viewpoints around the object of interest. Among these approaches, variational methods have been popular because they can be used to solve a wide variety of vision problems. The idea is to minimize an energy functional that depends on the considered object surface and on the input images, whose minima is reached at the object of interest. Many methods have been proposed in order to solve this problem, but these approaches are often limited in the kind of appearance they can handle. To overcome these limitations, we exploit Helmholtz reciprocity and propose a single framework for normal estimation and normal integration using a triangular mesh-based deformable model. In contrast with Chapter III, we do not make any assumption about the nature of the reflectance of objects to be reconstructed.

V.1.1 Helmholtz Stereovision: a Reconstruction Approach for Real World Objects

Most of the multiview reconstruction algorithms rely on image correspondences (as done for instance in multiview stereo [57]) or shading (using the normal information in multiview shape from shading [80, 79] or multiview photometric stereo [68]). When texture information (stereo case) is good enough or Lambertian assumption is sufficiently verified, those methods have been proved to give good results with surfaces that are nearly. They then obtain either accurate correspondences or accurate normal estimates. But when the scene is not Lambertian, which is the case for most (if not all) real world scenes, such cues are not valid and algorithms fail to reconstruct accurately the surface. In order to solve this problem, many alternatives have been proposed. Some authors consider specular highlights as outliers [68], and consider a large number of images in order to compensate. Some others modify the input images to have specular free images, and be photometric invariant [106, 173]. Another approach is to have a robust similarity measurement [131, 154]. All these methods try to compensate the non-Lambertian components in order to run reconstruction algorithms designed for the Lambertian case.

Some authors consider another strategy: they propose to use more general reflectance / radiance parametric models. See for instance [78, 164, 163]. In practice, these approaches suffer from several limitations. First of all the reflectance model has to be known in advance and this constrains scene to be composed by materials consistent with the chosen reflectance model. Such algorithms tend to solve non-linear systems of thousands of variables (one reflectance / radiance model per surface point), or need additional assumptions (single or fixed number of materials, single specular component, etc.). Those models are difficult to be optimized and generally require to alternatively estimate the reflectance and the shape. They are numerically unstable and easily tend to get stuck in local minima [79]. Moreover, the algorithms are generally ill-posed, and then require strong regularization which over-smooth the obtained results. Finally, the reflectance and illumination models need also to be approximated. Although such algorithms show reasonable results for perfect synthesized scenes, their application to real-world scenes is complex and requires accurate camera and light calibration.

A different approach introduced by [69] uses radiance samples of reflectance exemplars to match it with the observed images. A direct matching allows to estimate the surface normal in order to reconstruct the shape by normal integration. Although it can deal with many materials and anisotropic BRDF, material samples are needed. This can be a restrictive assumption in concerned applications.

During the last decade, some authors have proposed to use Helmholtz reciprocity in order to perform 3D reconstruction [171, 172, 61, 170, 149]. In practice, Helmholtz reciprocity is exploited by taking a pair of images under a single light source, where camera centers and light positions are exchanged at each shot. It uses the fact that in this particular setup, for a single reciprocal pair, the relationship between two radiances of a single surface point is independent of the reflectance. Contrary to works described in previous paragraphs, methods based on Helmholtz reciprocity [171, 172, 61, 170, 149] allow to accurately estimate the normals at one point, independently of the reflectance model. This can be used to obtain a 3D surface. In this context, modeling the reflectance, having material samples or being photometric invariant is not required. Nevertheless, contrary to most of multiview stereovision algorithms, the state of the art in Helmholtz reconstruction is limited to depth map reconstructions. In this chapter, we push the envelope by proposing a Full 3D multiview Helmholtz stereovision method.

V.1.2 A Surface-Based Approach

Until now, all the previous Helmholtz reconstruction methods were camera-view centered. On the contrary, we propose to change the surface representation and to adopt an object-centered strategy. Thus, instead of using a 2.5D surface as it used to be done previously, we represent the object's surface by a closed and dense 2D manifold embedded in the euclidean 3D space. The interest of this choice is two-fold. First this allows to naturally recover a full 3D surface when the other approaches only recover a depth map (or a needle map). Secondly, this allow to easily and properly handle visibility (shadows as well as self occlusions). Figure V.1 sheds light on these advantages: the picture on the left illustrates the case of conventional Helmholtz Stereovision methods, that recover a surface based on a virtual view (here in red). The drawing on the right illustrates the proposed approach which is surface-based instead of view-based reconstruction. The red contours show the optimal surface each method can recover. Clearly, the conventional approaches are confronted with difficulties with visibility since it generates discontinuities in the depth maps and the needle maps. Also, this difficulty spreads at the integration steps which behave badly in presence of discontinuities.

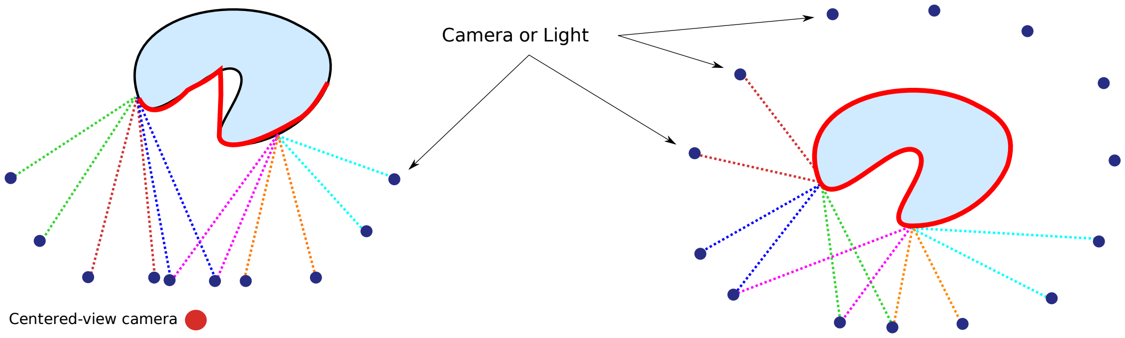

Figure V.1: Recovered surfaces by conventional Helmholtz Stereopsis versus proposed approach (surface in red). Left: case of conventional Helmholtz Stereo methods which recovers the shape based on a virtual view (in red). Right: the proposed surface-based approach.

In this work, as we will discuss in section V.3, we represent the 2D surface using a triangle mesh. In our context, where the surface normals play a key role, this representation offers significant advantages. In particular, it intrinsically links a depth to its surface normal which is, in the case of piecewise planar surfaces like triangular meshes, well defined on each facet. Also, this allows us to naturally combine both the normal integration and normal estimation when these were two separate stages in most of previous approaches [171, 172, 61].

V.1.3 Contributions

This chapter presents several improvements to state-of-the-art 3D reconstruction techniques exploiting Helmholtz reciprocity. First, we model the problem as an energy minimization problem which is completely surface-based (when all previous methods were camera based). Secondly, we present a method for solving this problem and show how to implement it on triangular surface meshes by using a coherent discrete gradient flow as shown in Chapter V. Finally, since we optimize a full surface, the method is able to handle visibility. This also allows to fully reconstruct dense 3D surfaces of complex objects in a single framework.

V.2 Helmholtz Stereopsis: A Variational Formulation

As described previously, Helmholtz reciprocity exploits the fact that BRDFs are generally symmetric and therefore, for any incoming angle $\hat{\imath}$ and outgoing direction $\hat{o}$, we have $\beta(\hat{\imath}, \hat{o}) = \beta(\hat{o}, \hat{\imath})$ where β is the BRDF function. By interchanging light and camera positions, one can exploit the constraint on β in radiance equations in order to solve 3D reconstruction problems.

Given a camera – light pair, one can write the radiance equation I_c of a scene seen from a camera c as:

$$I_c = \alpha \beta(\mathbf{v}_c, \mathbf{v}_l) \frac{\mathbf{v}_l \cdot \mathbf{n}}{|\mathbf{v}_l|^3} , \tag{V.1}$$

where $\mathbf{v}_c$ is the vector from the camera center to the point, $\mathbf{v}_l$ is the vector from the light center to the view-point and α is a constant. $\mathbf{n}$ is the surface normal of the considered point and $\beta(\mathbf{v}_c, \mathbf{v}_l)$ is the BRDF at the surface point (See Figure V.2). The same radiance equation can be written for modeling the radiance I_l with the BRDF $\beta(\mathbf{v}_l, \mathbf{v}_c)$. Using Helmholtz reciprocity allows us to write $\beta(\mathbf{v}_c, \mathbf{v}_l) = \beta(\mathbf{v}_l, \mathbf{v}_c)$. This equality then defines the Helmholtz stereopsis constraint for all the point of the surface $\mathcal{S}$:

$$\left[I_c \frac{\mathbf{v}_c}{|\mathbf{v}_c|^3} - I_l \frac{\mathbf{v}_l}{|\mathbf{v}_l|^3} \right] \cdot \mathbf{n} = 0 \,. \tag{V.2}$$

Now, we are going to formulate this constraint in the variational framework via a weighted area functional defined over the surface of the object.
We denote by $\pi_c(\mathbf{x})$ (resp. $\pi_l(\mathbf{x})$) the projection of a point in space $\mathbf{x}$ in the camera c (light l respectively), and I_c (or I_l) its corresponding intensity value in the image. For more clarity, we also denote

$$\mathbf{h}(\mathbf{x}) = \left[I_c(\pi(\mathbf{x})) \frac{\mathbf{v}_c}{|\mathbf{v}_c|^3} - I_l(\pi(\mathbf{x})) \frac{\mathbf{v}_l}{|\mathbf{v}_l|^3} \right] \,. \tag{V.3}$$

In this case, the surface that "best" verifies Equation (V.2) can be obtained by minimizing the following energy functional defined over the surface, with respect to the surface itself:

$$E_{HS}(\mathcal{S}) = \int_{\mathcal{S}} (\mathbf{h}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}))^2 \;\nu_{\mathcal{S},c,l}(\mathbf{x}) \,\mathrm{d}s \,, \tag{V.4}$$

where $\nu_{\mathcal{S},c,l}$ is the characteristic function such that $\nu_{\mathcal{S},c,l}(\mathbf{x}) = 1$ if $\mathbf{x}$ is visible from both images, or 0 otherwise. $\mathrm{d}s$ is the element of area of the surface. This problem formulation allows thus to naturally integrate both multiview geometry and normal constraint. The functional (V.4) constrains the surface normals to be on the orthogonal plane of $\mathbf{h}(\mathbf{x})$. This is an ill-posed problem since there is an unlimited choice for the normal. In this context several reciprocal pairs are needed in order to better pose the problem and Energy (V.4) has to be adapted to multiview settings. At the end, we then consider the energy to minimize as the sum of all energies for all Helmholtz pairs i:

$$E_{HS}(\mathcal{S}) = \sum_i \int_{\mathcal{S}} (\mathbf{h}_i(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}))^2 \;\nu_{\mathcal{S},c_i,l_i}(\mathbf{x}) \,\mathrm{d}s \,, \tag{V.5}$$

where i is the i^{th} camera / light pair.

In the next section, we show how to minimize this energy via gradient descent when the surface is represented by a triangle mesh. To simplify the notations, we will just consider and compute the gradient of the functional (V.4). The gradient of (V.5) is then obtained by summing the gradients of all the camera / light pairs.

V.3 Optimization for Triangle Mesh Representation

V.3.1 Choice of Representation

In section V.1.2 we show the interest of using an intrinsic and full surface representation. In practice, there exists several possibilities for such representation. We propose to minimize energy (V.5) using a gradient descent algorithm. The choice of the representation must then be consistent with the surface evolution technique which is the base of gradient descent methods. Here we chose to use the Lagrangian framework and to represent the surface by a triangle mesh. Several reasons motivate this choice. In recent years, Lagrangian methods have taken advantage of significant advances in mesh processing allowing these methods to enjoy practical properties such as topological changes [129, 167]. In Lagrangian methods, the gradient is computed directly from the discrete representation, whereas in the Eulerian framework the continuous gradient is computed and then discretized. Performing gradient descent in the context of discrete representations allows to make the minimization coherent with the handled numerical object. In other respects, the visibility of a point from a vantage point is well defined and easy to compute with a mesh representation (by using graphic hardware). In practice, it is easier to check the visibility with such a representation than with a level-set representation. All these reasons make nowadays Lagrangian methods more and more popular. Also these methods have recently proved their strong potential for 3D applications [154, 39, 32].

V.3.2 Shape Gradient and Evolution Algorithm

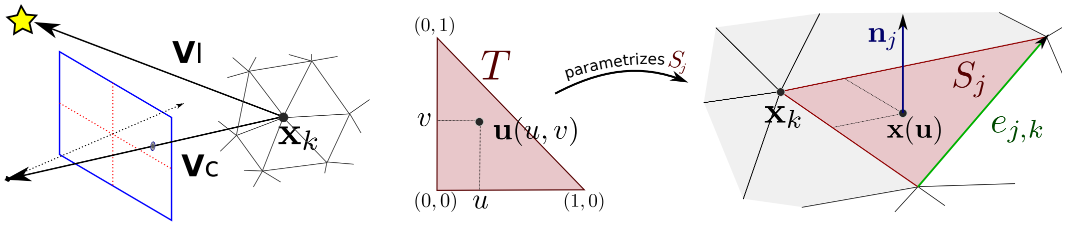

Figure V.2: Notations. Parametrization of the discrete representation of the surface into a triangle mesh.

In this section we first summarize Chapter II for surface evolution. Like previously, we consider that the surface is a piecewise planar triangular mesh. Let $\mathbf{X} = \{\mathbf{x}_1 \ldots \mathbf{x}_n\}$ be a discrete mesh, $\mathbf{x}_k$ being the k^{th} vertex of $\mathbf{X}$, and let $\mathcal{S}_j$ be the j^{th} triangle of $\mathbf{X}$. With such a representation, functional (V.4) can be discretized and be rewritten as:

$$E_{HS}(\mathcal{S}) = \sum_j \int_{\mathcal{S}_j} (\mathbf{h}(\mathbf{x}) \cdot \mathbf{n}_j)^2 \; \nu_{\mathcal{S},c,l}(\mathbf{x}) \, \mathrm{d}s_j \; , \tag{V.6}$$

where $\mathbf{n}_j$ is the normal to $\mathcal{S}_j$ and where the sum is over all the triangles of the mesh $\mathbf{X}$. Figure V.2 illustrates these notations.

We propose to optimize E_{HS} with respect to $\mathcal{S}$ using the shape gradient [30]. Let $\mathbf{V}$ be a vector field defined on all the vertices $\mathbf{x}$ of the mesh $\mathbf{X}$ representing the surface deformation. Let us consider the evolution of E_{HS} according to the deformation $\mathbf{V}$. In other words, we assume that the vertices $\mathbf{x}_k[t]$ of $\mathbf{X}[t]$ are moving according to $\mathbf{x}_k[t] = \mathbf{x}_k^0 + t\mathbf{V}_k$. The method for computing the gradient of E_{HS} with respect to $\mathcal{S}$ consists in computing the directional derivative of $E(\mathcal{S})$ for this deformation $\mathbf{V}$, i.e., $\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0}$, and then in rewriting it as a scalar product of $\mathbf{V}$, i.e. as $\langle \mathbf{V}, \mathbf{G} \rangle = \sum_k \mathbf{V}_k \cdot \mathbf{G}_k$. The obtained vector $\mathbf{G}$ is called the *gradient* and the energy necessarily decreases when deforming the surface according to its opposite direction $-G$. Indeed, for $\mathbf{x}_k[t] = \mathbf{x}_k^0 - t\mathbf{G}$, we have $\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = -\langle \mathbf{G}, \mathbf{G} \rangle \leq 0$, see [39].

To obtain the gradient of our energy E_{HS}, we have then to calculate the expression of $\frac{d}{dt}E_{HS}(\mathcal{S}[t])\Big|_{t=0}$ and express it as a scalar product of $\mathbf{V}$. In Section II.3, we have detailed this calculus in the general case where a functional $\int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\mathrm{d}s$ is minimized. By replacing $g(\mathbf{x}, \mathbf{n}(\mathbf{x}))$ by $(\mathbf{h}_i(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}))^2$, we get:

$$\begin{aligned} \frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = \sum_j \sum_{k\in K_j} \mathbf{V}_k \cdot \Bigg\{ & \frac{\mathbf{e}_{j,k}}{A_j} \wedge \int_{\mathcal{S}_j} \left((\mathbf{h}\cdot\mathbf{n}_j)^2\,\mathbf{n}_j - 2\,(\mathbf{h}\cdot\mathbf{n}_j)\,\mathbf{h} \right) \mathrm{d}s_j \\ & - \int_{\mathcal{S}_j} 2(\mathbf{h}\cdot\mathbf{n}_j)\nabla_{\mathbf{x}}(\mathbf{h}\cdot\mathbf{n}_j)\;\phi_k(\mathbf{x})\;\mathrm{d}s_j \Bigg\}, \end{aligned} \tag{V.7}$$

where A_j is the area of $\mathcal{S}_j$, $\mathbf{e}_{j,k}$ is the edge of $\mathcal{S}_j$ that is at the opposite of vertex k; K_j is the set of the indexes of the three vertices of the triangle $\mathcal{S}_j$ and $\phi_k : \mathcal{S} \to \mathbb{R}$ is the piecewise linear interpolating basis function such that $\phi_k(\mathbf{x}_k) = 1$ and $\phi_k(\mathbf{x}_i) = 0$ if $i \neq k$.

Since, $\nabla_{\mathbf{x}}(\mathbf{h}\cdot\mathbf{n}_j) = D_{\mathbf{x}}\mathbf{h}\;\mathbf{n}_j$, $D_{\mathbf{x}}\mathbf{h}$ being the 3×3 Jacobian matrix, from (V.3) we get:

$$\begin{aligned} \nabla_{\mathbf{x}}(\mathbf{h}\cdot\mathbf{n}_j) = \nabla I_c \frac{\hat{\mathbf{v}}_c\cdot\mathbf{n}_j}{\|\mathbf{v}_c\|^2} + \frac{I_c}{\|\mathbf{v}_c\|^3}\left(\mathbf{n}_j - 3(\hat{\mathbf{v}}_c\cdot\mathbf{n}_j)\hat{\mathbf{v}}_c\right) & \\ - \nabla I_l \frac{\hat{\mathbf{v}}_l\cdot\mathbf{n}_j}{\|\mathbf{v}_l\|^2} - \frac{I_l}{\|\mathbf{v}_l\|^3}\left(\mathbf{n}_j - 3(\hat{\mathbf{v}}_l\cdot\mathbf{n}_j)\hat{\mathbf{v}}_l\right)\;, & \end{aligned} \tag{V.8}$$

where $\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.

Then the L^2 gradient descent flow using the triangular mesh uses the gradient $\nabla E(\mathbf{X}) = M^{-1}\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X})$, where M is the *mass* matrix and $\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X})$ is directly given by the part in braces of Equation (V.7). One classically approximates M by the diagonal mass lumping $\tilde{M}$, where $\tilde{M}_{ii}$ is the area of the Voronoi dual cell of $\mathbf{x}_i$ times the identity matrix Id_3; for more details, see e.g. [39, 108]. The evolution algorithm used here is:

$$\begin{cases} \mathbf{X}[0] = \mathbf{X}^0\;, \\ \mathbf{X}[t+1] = \mathbf{X}[t] - dt\;\tilde{M}^{-1}\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}[t])\;, \end{cases} \tag{V.9}$$

where $\mathbf{X}^0$ is some initial mesh.

V.4 Experiments

V.4.1 A New Practical Setup

Switching light and camera positions for real experiments is not trivial. However, easy practical setups can be designed to satisfy the Helmholtz stereo restrictive configuration. In their original paper, Zickler et al. [172] have designed such a setup, where a turning wheel is used to reconstruct the shape of an object. The camera and the light are diametrically opposed on the wheel. By turning the wheel, the light and camera position can be exchanged easily. In their configuration, a fixed object is disposed in front of the acquisition device and images can only be captured from one side of the object. This setup is shown in the left part of Figure V.3.

Although the setup of [172] can acquire 3D shape of arbitrary appearance, it is difficult to extend it to reconstruct entire 3D shapes. Since our main concern in this work is the recovery of full 3D shapes, we propose a new practical solution for that case. Instead of moving the camera and the light, we propose to place the object on a rotating table and turn it of an angle θ. We place two light L_1 and L_2 sources respectively at angles $-\theta$ and θ from the center of rotation, and at the same height and distance as the camera's one. Then the reciprocal image of a first image taken with the light L_1 corresponds to a second image taken under the light L_2 and a rotation angle of the object of θ. Then by taking pictures that way all around the object, one can capture multiview Helmholtz stereo data. The drawing Figure V.3 illustrates the method. A possible extension is to use several cameras and more lights using the same principle, at different heights. This way one could capture and reconstruct higher quality 3D models. The following section shows results using the presented acquisition setup.

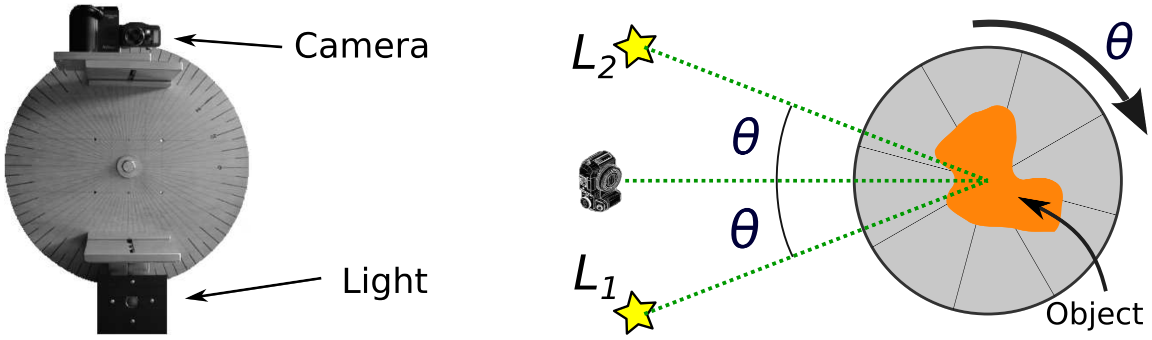

Figure V.3: Two practical solutions for Helmholtz Stereo: On the left, the acquisition wheel of the setup proposed by Zickler et al. [172]; on the right, the proposed configuration. Our configuration allows more easily to acquire entire 3D shapes, whereas the setup on the right requires the object or the acquisition device to be moved during the scanning of the object.

V.4.2 Experimental Results

In this section we present results using the gradient developed in section V.3 (Equation (V.7)) that directly corresponds to the minimization of our original Energy (Equation (V.6)).

Let us emphasize here that the gradient descent flow used to obtain the results is exactly the one described previously. In particular, it does not require additional terms or parameters such as surface smoothing present in most variational formulations like the one in Chapter III. Of course, adding such a term would help in being more robust to noise and calibration errors. The only parameters used by us are for numerical computations, like for instance the numerical integration over the triangles. All those can be easily estimated automatically. Since this is a gradient descent approach, one needs some reasonable initialization of the surface such as the visual hull to avoid local minima.

The experiments were implemented in C++ and OpenGL using the CGAL library for mesh computation running on a standard 2.4GHz Linux machine – and the topology adaptive meshes of [129]. The optimization starts from an initial condition which is the visual hull in our case. A coarse-to-fine approach is applied to help prevent from local minima. The rendered results use only one constant normal per facet (flat shading).

We first apply our method to synthesized data. Figure V.4 shows an example of simple objects disposed on a plane, where images were generated using non-Lambertian reflectance. This dataset is composed of 32 reciprocal pairs with images of resolution 800×600, placed all around the object of interest. This example shows that our method is able to recover the surface whereas previous Helmholtz stereo methods, where visibility is not accounted for, would fail. In order to solve this problem, they would need to cluster the camera positions to find several view-centered cameras, integrate multiple normal maps and then merge the final reconstructions into a single surface. Our method is simpler in the sense that it works without additional steps.

Even though the number of vertices is low, the gradient flows tends to place them in their correct location. In particular triangle edges perfectly match the one in the images. Again, this is made possible because the discrete gradient is computed with respect to the discrete representation. Also the approach is suitable for reconstructing objects with sharp edges, having depth discontinuities or self-occlusions.

The example in figure V.5 shows that our method can be applied to reconstruct full and high quality object shapes. Even though the input resolution of the images is low (1024×768), the recovered surface nicely matches the ground truth model. The images were generated using a mixture of different specular models so that it looks realistic and non Lambertian. Details are well recovered, and the quality of the mesh is good enough for further object relighting.

Following the evaluation presented in [136], we perform a quantitative evaluation of examples from Figures V.4 and V.5 where we consider the object in a 2m diameter bounding box. For Figure V.4, the completeness at 10mm is 87.51%, and the accuracy at 95% is 9.301mm. For the Buddha, the completeness at 10mm is 95.766%, and the accuracy at 95% is 5.29mm.

The next two figures V.6 and V.7 show real data used in previous work [171, 172], that

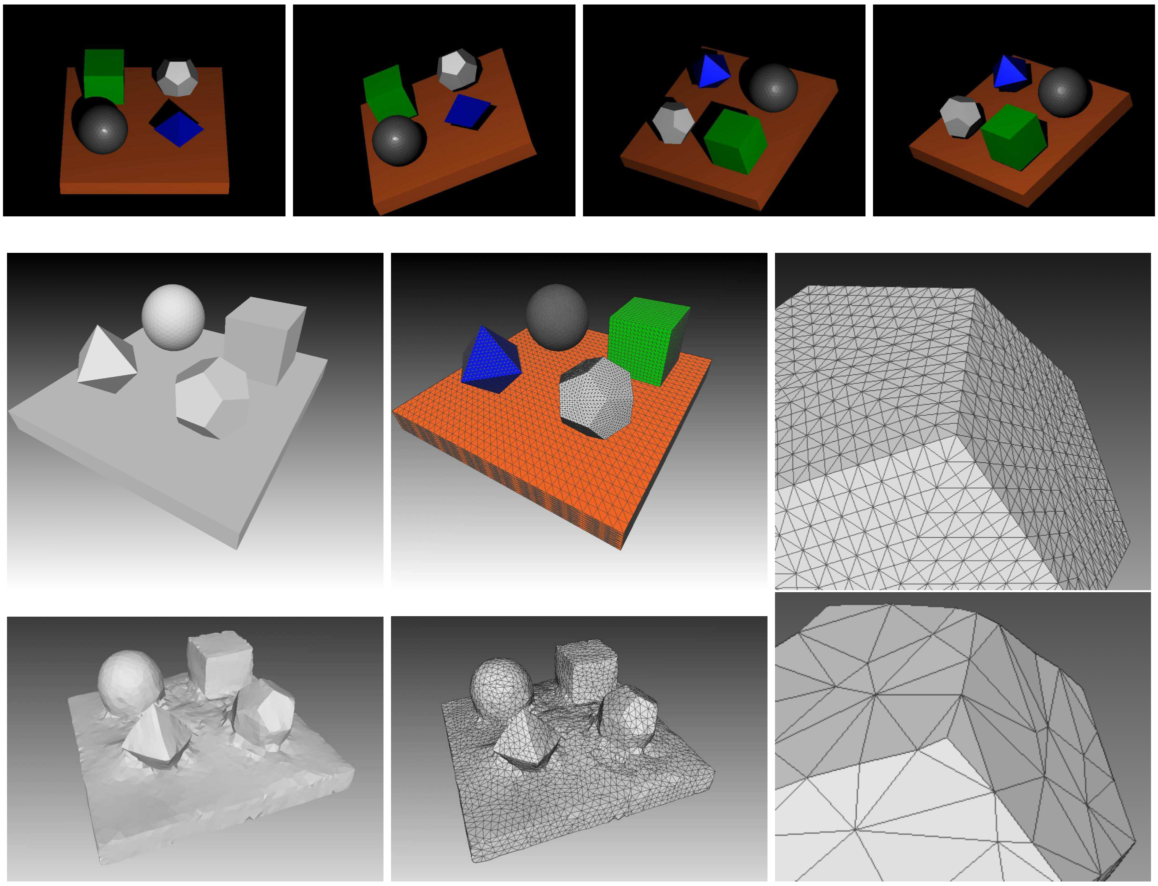

Figure V.4: A simple synthetic example. Top row: 2 camera / light pairs out of the 32. Middle row: ground truth surface; ground truth mesh representation; Bottom row: final result; mesh representation of the result; details on the mesh.

have been capture only from one side. We then cannot reconstruct the full surface of the object since images behind the object are missing. Figure V.6 shows the 3D reconstruction of a textured box. Since we can choose the mesh resolution, having large triangles compared to the image resolution allows to integrate the gradient over the triangle and have a correct gradient flow for the vertices of the triangle. Then we can reconstruct objects with textured or rough surfaces using Helmholtz reciprocity similarly as in [61]. This figure also shows different optimizations taken from different initial conditions that finally give similar results.

Figure V.7 shows results for two real datasets, the mask containing 18 reciprocal pairs, and the mannequin containing 8 reciprocal pairs.

Finally, the approach was tested on real dataset on full 3D objects (Figure V.8). It consists of 18 reciprocal pairs (using 1104×828 resolution images) taken on a ring around the object slightly on top of it. The two data show a "Fish" reconstruction highly specular with fine changing surface structure, whereas the second one "Dragon" has strong self-occlusions and complex shape. Starting from the visual hull, we are able to recover details on the surface even though the input image resolution is not too high. Results are illustrated in Figure

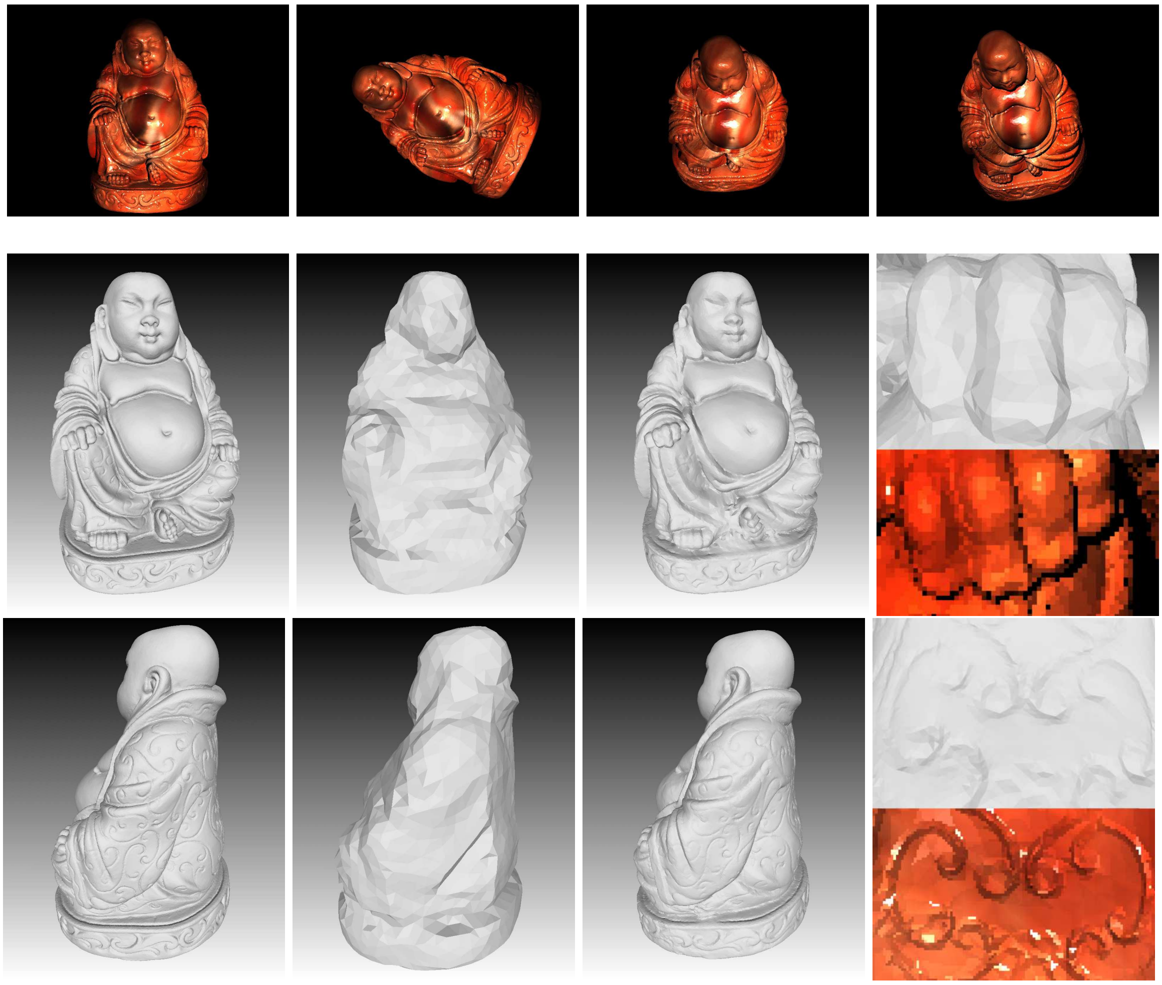

Figure V.5: Buddha dataset. Top row: 2 camera – light pairs out of the 32. Bottom row: ground truth surface; initial visual hull; estimated mesh; input image zoom and corresponding recovered mesh details.

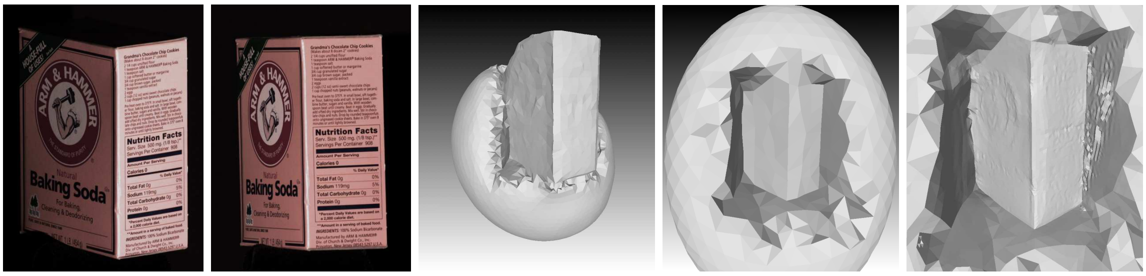

Figure V.6: Small box data: results of the proposed method with different initial conditions and mesh resolution. From left to right: 1 camera – light pair out of the 8 pairs; result from a small non-encompassing initial surface; result from an encompassing initial surface; result from an encompassing initial surface with a more dense resolution.

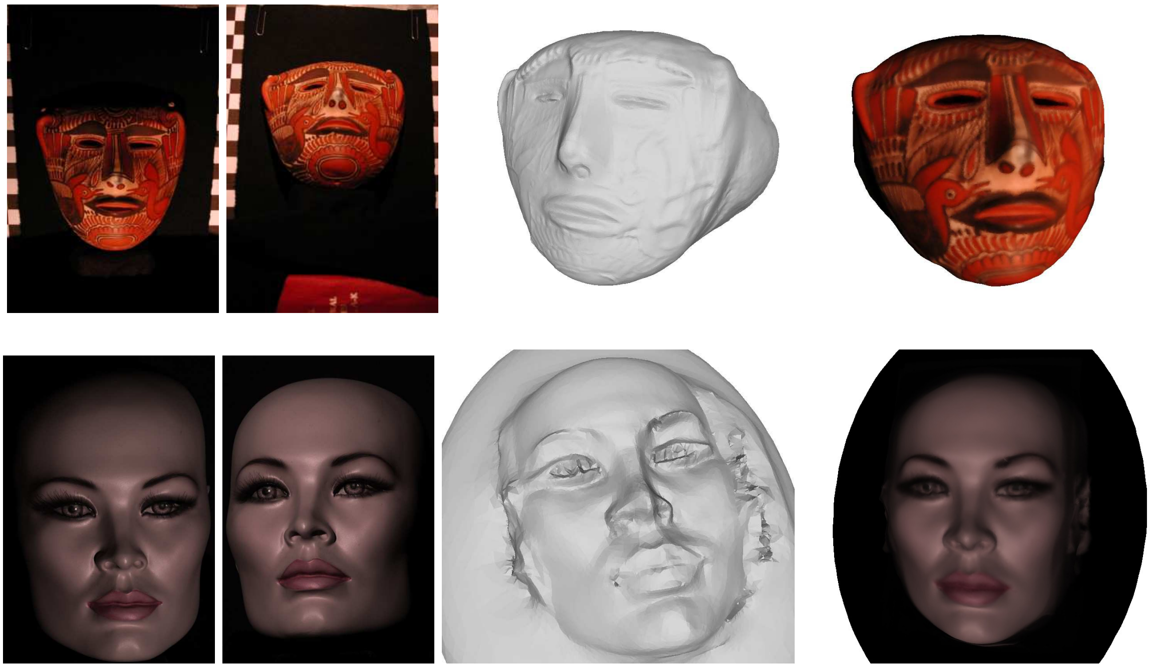

Figure V.7: Mannequin (8 reciprocal pairs) and Mask (18 reciprocal pairs) data: results of the proposed method for two real dataset containing varying complex appearance. From left to right: 1 camera / light pair out of the 18 (respectively 8); final result; final mesh textured with the mean of the reprojected colors from the cameras.

V.8 and present the recovered full 3D surface. Some parts of the surface are not visible from the images and thus cannot be recovered. Camera calibration was performed using a checker board without distortion corrections and light positions were empirically positioned and calibrated. Images are taken around the object so some parts are occluded from images and objects also contain self-occlusions. For those reasons, previous approaches using a camera-center view do not apply – since it requires depth continuity – whereas our surface-based approach can recover the full 3D shape.

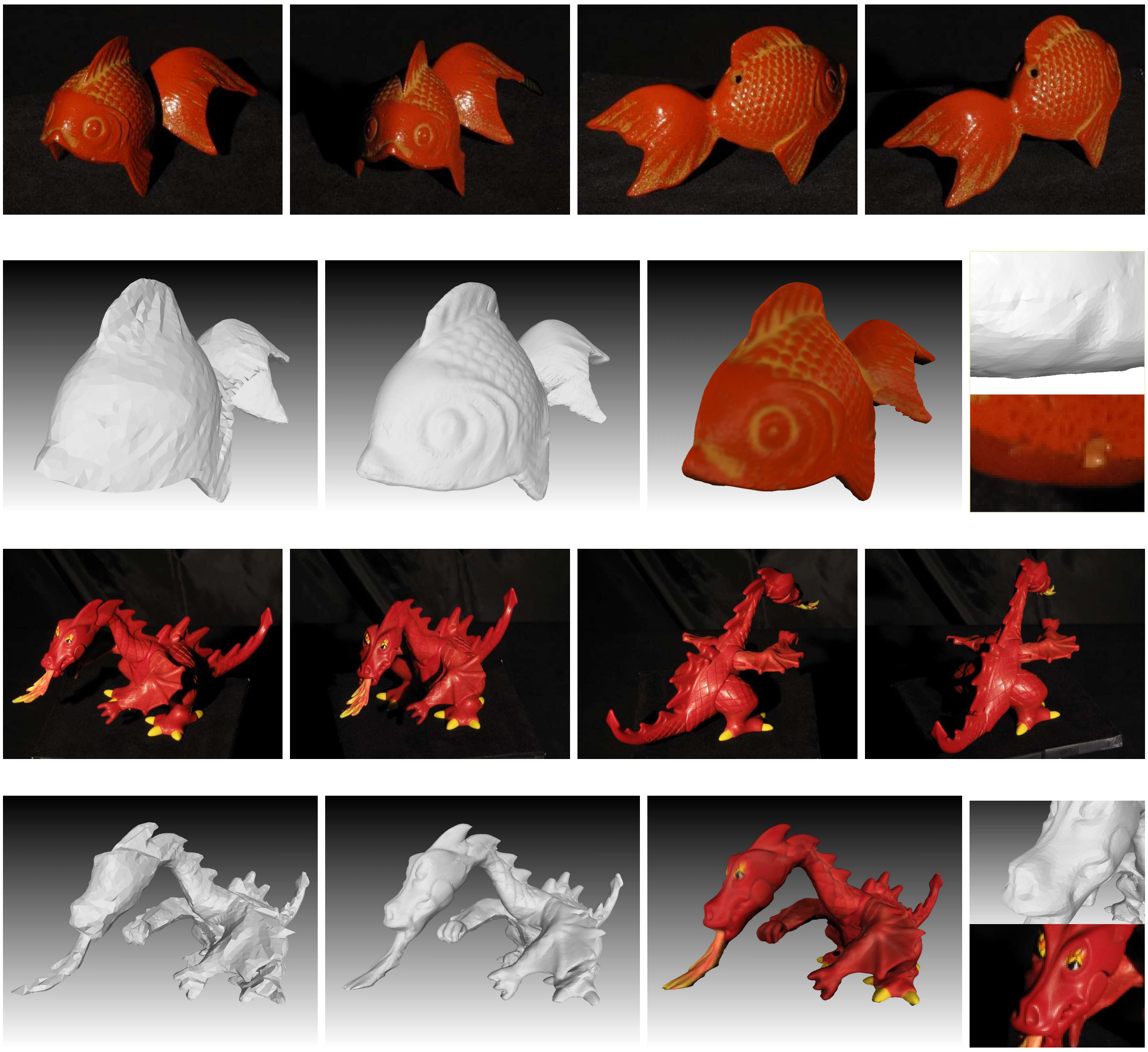

Figure V.8: Fish and Dragon data (18 reciprocal pairs of 1104×828): Results of the proposed method for two full 3D real world datasets. Top and third row: 2 camera / light pairs out of the 36. Second and last row: initial visual hull; recovered 3D mesh; final mesh textured with the mean of the reprojected colors from the cameras; input image zoom and corresponding recovered mesh details.

All these examples illustrated the advantage of having a mesh representation. It allows to preserve edges, and we show that having a coherent discrete gradient flow allows vertices to be placed at their correct locations. Possible extensions to the approach may use automatic adaptive meshes, where big triangles would fill planar surface parts and high curvature would have a more dense mesh.

V.5 Multi-view Helmholtz Stereo as a Reprojection Error

V.5.1 Problem Modelling

Second order minimization is known to recover well the higher frequency of the surface, but poorly recover the lower ones. Also, due to the integration process, the optimization might be slow if the surface is too far from the solution. A good initial condition and a coarse to fine approach significantly help to prevent these problems, as illustrated in the experiments. In particular, the gradient flow tends to shrink the surface and introduce a minimal surface bias, as explained in Chapter III.

Moreover, since here the method minimizes a weighted area functional defined over the surface, one of its global minima (in addition to the real surface) is the empty set. To prevent from this effect, one can add an additional term, start closer to the solution, or fix boundary conditions. A more elegant way is to see the problem as a reprojection error as introduced in Chapter II. Instead of minimizing a weighted area functional, one can reformulate the problem (V.2) by minimizing the following energy functional:

$$E(\mathcal{S}) = \int_{\mathcal{I}_c} \left(I_c - \left(I_l \circ \pi_{\mathcal{S},I_l} \circ \pi^{-1}_{\mathcal{S},I_c} \right) \frac{\hat{\mathbf{v}}_l \cdot \left(\mathbf{n} \circ \pi^{-1}_{\mathcal{S},I_c} \right)}{\hat{\mathbf{v}}_c \cdot \left(\mathbf{n} \circ \pi^{-1}_{\mathcal{S},I_c} \right)} \right)^2 \mathrm{d}\mathbf{p} \, , \tag{V.10}$$

where $\pi^{-1}_{\mathcal{S},I_c}(\mathbf{p})$ is the reprojection of an image point $\mathbf{p}$ of the image I_c on the surface $\mathcal{S}$. $\mathrm{d}\mathbf{p}$ represents the pixel area measure. This energy is image-based and so is closer to a Bayesian vision of the problem as described in the Introduction I.2.2.

This formulation has been already presented by [149] in case of Helmholtz stereo. However it has not been minimized as a reprojection error, like for instance in the case of [55] for continuous surfaces, and of [31] using deformable meshes like in Chapter II. Such formulation lets appear an additional term, that turns out to behave like a visual hull constrain on the silhouette occluding contours, or as a contour matching term for self occlusions. Such a term will give boundary conditions to prevent from shrinkage, and will help the method to be more robust to initial conditions.

As shown in Chapter II, Equation (V.10) can be seen as a reprojection error functional, and can then be rewritten as an energy functional defined over the visible surface. The only difference here is that both camera and light visiblity need to be carefully accounted for. Following the explanations in Chapter II, we can compute the gradient and separate it into the interior term and the horizon term. The difference here is that the horizon term does not only appear at occluding contours, but also at horizon points created by the light crepuscular rays. To understand this, let us first rewrite Equation (V.10) as an energy functional defined over the visible surface:

$$E(\mathcal{S}) = -\int_{\mathcal{S}} \Big(I_c \circ \pi_{\mathcal{S},I_c}(\mathbf{x}) - I_l \circ \pi_{\mathcal{S},I_l}(\mathbf{x})\; \alpha(\mathbf{x},\mathbf{n}(\mathbf{x})) \Big)^2 \, \frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \, \nu_{\mathcal{S},I_l}(\mathbf{x})\, \nu_{\mathcal{S},I_c}(\mathbf{x})\, \mathrm{d}s \,, \tag{V.11}$$

where $\alpha(\mathbf{x},\mathbf{n}(\mathbf{x}))$ is the quantity $\dfrac{\hat{\mathbf{v}}_l(\mathbf{x})\cdot\mathbf{n}(\mathbf{x})}{\hat{\mathbf{v}}_c(\mathbf{x})\cdot\mathbf{n}(\mathbf{x})}$ and as before, $\mathrm{d}s$ is the surface area measure such that:

$$\mathrm{d}\mathbf{p} = \frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \, \nu_{\mathcal{S},I_l}(\mathbf{x})\, \nu_{\mathcal{S},I_c}(\mathbf{x})\, \mathrm{d}s \,.$$

Then instead of having Equation II.5 like in Chapter II, we have an energy of the form:

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x},\mathbf{n}(\mathbf{x}))\cdot\mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S},I_l}(\mathbf{x})\, \nu_{\mathcal{S},I_c}(\mathbf{x})\, \mathrm{d}s \,. \tag{V.12}$$

This equation can then be minimized the same way as it is done in Chapter II. In Equation (V.11), we only consider a single image. However the full energy is a sum over all images. In particular, this is a sum over all image reciprocal pairs. Then for a given reciprocal pair, the energy is the sum of two reciprocal terms as in Equation (V.11), and corresponds to Equation (V.12) with the function $\mathbf{g}(\mathbf{x},\mathbf{n}(\mathbf{x}))$ equals to:

$$\begin{aligned}\mathbf{g}(\mathbf{x},\mathbf{n}(\mathbf{x})) = {} & \Big(I_c \circ \pi_{\mathcal{S},I_c}(\mathbf{x}) - I_l \circ \pi_{\mathcal{S},I_l}(\mathbf{x})\; \alpha(\mathbf{x},\mathbf{n}(\mathbf{x})) \Big)^2 \, \frac{\mathbf{x}_c}{\mathbf{x}_{c,z}^3} \\ & + \Big(I_l \circ \pi_{\mathcal{S},I_l}(\mathbf{x}) - I_c \circ \pi_{\mathcal{S},I_c}(\mathbf{x})\; \frac{1}{\alpha(\mathbf{x},\mathbf{n}(\mathbf{x}))} \Big)^2 \, \frac{\mathbf{x}_l}{\mathbf{x}_{l,z}^3} \,, \end{aligned} \tag{V.13}$$

where $\mathbf{x}_c$ and $\mathbf{x}_l$ are the coordinates of the point $\mathbf{x}$ in the camera coordinates system of respectively cameras c and l.

Finally by following Chapter II, then the gradient descent flow is composed of three terms: one interior term which applies to points visible by both cameras, and two horizon terms corresponding to each of the two camera's visibility.

V.6 Conclusion and Discussion

In this chapter we have presented a surface-based method to estimate the 3D shape of objects from multiview Helmholtz stereo pairs. This leads to several benefits:

- Unlike examples treated in Chapter III, no assumptions are made about the reflectance of the object of interest. As a consequence, we can estimate 3D models of objects of any arbitrary BRDF.

- As far as we know, this is the first time Helmholtz stereopsis can be used to recover dense and full 3D models into a single framework. This is made possible thanks to the compact surface representation that allows to easily compute surface point visibility.

Moreover, the mesh based representation used allows to naturally exploit geometric relationships between a point of the scene and its surface normal.

- Unlike many variational methods, we do not use any additional regularization term here such as surface smoothing as used in some examples in Chapter III.
- We propose a new acquisition setup for Helmholtz Stereovision, that easily allows to acquire entire 3D shapes of static objects. This requires a single fixed camera and two light sources, correctly positioned with respect to a rotating table. This simplify Helmholtz stereo vision acquisitions scenarios to recover entire shapes. Extensions using more cameras and lights could be done by placing them at different heights. This would give a more complete set of images around the object in order to improve the reconstructions.
- This acquisition scenario requires a dark room. While being very restrictive, a simple solution toward this problem is to separate direct and indirect illumination from the light source, as done in [116]. This would have two significant advantages: the first one is being able to recover a larger class of objects, and the second is to remove inter-reflection effects and therefore improve the result's accuracy.
- Like in Chapter II, multi-view Helmholtz stereo can be modeled as a reprojection error functional, and as a consequence enjoy the same gradient descent flows that correctly account for visibility changes during the optimization. Further experiments are required to fully validate this step.

CHAPTER VI

Conclusion

In this chapter, we conclude this thesis work on multi-view shape reconstruction. We first summarize the main contributions of our work and then discuss further directions of research.

VI.1 Summary and Discussion

We can summarize the multi-view shape reconstruction problem as identifying and exploiting information, cues and priors (obtained from 2D images and knowledge of the scene) that along with proper numerical tools and algorithms, allows to recover a 3D shape of a scene or object of interest. This process can be addressed using generative models and can be recovered using Bayesian inference. This leads to the energy minimization of an energy functional defined over all images that minimizes the error between generated images from a given model and the observed images. This functional is called the reprojection error. This error depends on image formation processes that are very complex to model, due to illumination, reflectance, projection models used as well as occlusions. As a consequence, its minimization is not trivial. We solve the problem using gradient descent for surface evolution.

This thesis has two main contributions. The first one is to exploit the surface's representation one has in its hand, and the importance of having an optimization algorithm consistent with it. The other is how, given such a representation, one can model different multi-view reconstruction algorithms that can handle different cases: textured, Lambertian, textureless, specular or generic scenes. We detail some of those contributions:

- **Importance of shape representation**

  The choice of a scene representation in multi-view reconstruction algorithms is very important. The type of object to be modeled directly depends on it. In Chapter I, we reviewed some of them and listed the advantages in existing surface models for multi-view stereo techniques. We have seen in the following that a large variety of objects can be modeled using mesh-based surfaces. This is due to the flexibility meshes offer from a practical point of view, either by their non-fixed spatial resolution, their compact representation and memory efficiency or their Graphics-oriented nature. As a surface-centered approach, they allow to reconstruct general objects from multiple sides and allow to easily model occlusions and deal with them.

  The representation not only conditions the class of objects one may or not reconstruct, but it more importantly conditions the algorithms that are going to be used in order to optimize or retrieve that representation. Again here, the use of triangular meshes has been driven by several choices, and has been proved to be of particular interest in all the examples that we have shown in this thesis. In Chapter II we show how to optimize a deformable mesh by minimizing reprojection error functionals, even when this energy depends on the normals or the visibility. We compute the exact gradient of the discretized reprojection error over the mesh, which leads to coherent gradient flows. As a consequence, this leads to coherent vertex displacement that makes edges match with apparent contours in images, and triangle tend to orient themselves correctly. In addition, the gradient of the reprojection error accounts for visibility changes during the surface evolution similarly as in [55], which reduce minimal surface bias in this problem. Computing the differential of the reprojection error using deformable meshes is the main contribution of this thesis.

- **Lambertian reconstruction**

  In Chapter III we show different applications of 3D reconstruction including multi-view stereo and multi-view shape-from-shading, where we consider a constant brightness assumption across views. In those cases a simple generative model is adopted, but still in this case simple gradient descent can be performed and can achieve reasonable results. This can be obtained by the correct minimization of the reprojection error. If image correspondence is very popular, exploiting photometry and surface orientations has been proved efficient for recovering high quality 3D models. In those cases the generative model carries terms depending on the normal that help in getting better accuracy and more precise 3D models. Then stereo and shading cues are accounted for in a unique framework.

- **Towards non-Lambertian modeling**

  When the constant brightness assumption is not valid, reconstruction can still be achieved. If normals can be estimated using for example photometric stereo techniques, then they are sufficient to build consistent 3D models. In Chapter III we show how to integrate multiple normal fields in order to reconstruct hight quality 3D models. This can be obtained by adapting the generative model of the reprojection error.

  When normals cannot be easily detected or for non-Lambertian reflectance, both the shape and the reflectance have to be estimated. This makes the problem ill-posed. Additional smoothness terms on reflectance such as the one based on Total Variation described in Chapter IV can be applied. Alternatively, segmentation could be used in order to reduce the number of reflectance parameters to estimate by using the fact several surface points share the same reflectance.

  In order to estimate the shape of objects of any arbitrary BRDF, we introduce Multi-view Helmholtz Stereo in Chapter V. Helmholtz reciprocity is exploited and adapted as an energy functional minimized thanks to the approach described in Chapter II. The mesh representation makes it possible to reconstruct the dense entire shape of the object, where previous methods were limited to partial reconstructions. Finally we propose a simple acquisition setup that makes Hemlholtz stereo practical.

VI.2 Future Research Directions

As discussed in each of the chapters, there is a large place for improvements. Those can be used in different stages of the reconstruction, from the physical model used to the additional cues to exploit, from the optimization tools to the acquisition setup. We discuss here different directions in which this thesis work can be extended and pursued.

- The choice of a projection model, an illumination model or a reflectance model conditions the kind of scenes one wants to reconstruct. Several improvements on the generative model could be done in order to make it more robust or more generic. In this thesis, the cost of a 3D point is computed independently from each other. Then more prior information could be used from the images, such as adaptive smoothness terms depending on the context, geometric prior, or segmentation prior as described in Chapter IV.

- A better initialization: In most examples, the surface evolution starts from the visual hull. However, a lot of recent approaches in multi-view stereo allow to reconstruct dense point clouds accurately and robustly. By extracting a surface out of it, one could use our deformable mesh framework as a last surface refinement stage to gain accuracy and robustness.

- Improving the optimization: as described in this thesis, the surface evolution is based on gradient descent techniques. Therefore, this is a local method and may fall into local minimas. This effect is reduced by the fact we have a coarse-to-fine strategy, but then coarse meshes cannot approximate complex shapes very well. To improve that, one can work on the model as described above, but one can also change the algorithm. One straightforward idea is to change the gradient metric to have a more coherent gradient flow, similarly as [24, 39]. Also gradient descent schemes such as Levenberg-Marquart or BFGS could be applied to speed up the optimization. Convex models such as the ones described in [28] is an alternative technique which is worth to study.

- Generative vs. non-generative. While generative approaches offer a nice theoretical and Bayesian motivated framework, direct methods or heuristic based approaches offer very good alternatives. For example, one could find a term that acts the same as the horizon term but would be more efficient to compute (which is actually what many people already do to prevent from minimal surface bias). Similarly, heuristic-based terms can be added or can replace existing ones. Generative models are more general but are sometimes more difficult to optimize as well. Recent 3D reconstruction applications, while being less general, offer faster algorithms that turn out to be very accurate. There is a trade-off to find between physically motivated models and heuristic or empirical models. Continuing research on both sides is a necessity to fully understand 3D reconstruction and propose better algorithms.

- Temporal consistency. In this thesis, only static reconstructions have been performed. Dynamic scenes have been reconstructed as independent reconstructions at each time instant. However, motion is a very important cue and could be easily added in our models. That would require to develop $4D$ or $3D + t$ mesh tools, such as the work of [3, 27] and looks more promising and appropriate than methods that require implicit functions as the surface representation.

- In this work we assume that the image information is available thanks to usual digital images. Since our primary goal is to reconstruct a 3D shape, those might not be the best tools to do so. Recent advances in computational cameras allow to design systems for their purpose. A multi-view system could be designed that way, by exploiting for instance coded aperture cameras. Moreover, using the appropriate generative models, this could directly be integrated in our framework, and could lead to promising results.

APPENDIX A

Computational Details

In this appendix, we detail computations used in this thesis, in particular in Chapter II.

Contents

A.1 Camera Modeling and Differential Properties in Images

A.1.1 Pinhole Camera

Here we briefly describe some calculations regarding the pinhole camera model that are required in the context of this thesis. In particular, we show how to deal with change of variables from the surface $\mathcal{S}$ to the image I.

In Chapter II, we rewrite the energy functional on the image domain as an integral over the surface (See Equations (II.2) and (II.5)). This implies defining the change of variable in the integrals. Basically we want to estimate the pixel element area in images $\mathrm{d}\mathbf{p}$ in terms of the surface element area $\mathrm{d}s$. The following result is well known in the literature [56, 130, 140, 160]:

$$\mathrm{d}\mathbf{p} = -\frac{\mathbf{n}\cdot\mathbf{x}}{\mathbf{x}_z^3}\,\mathrm{d}s\,, \tag{A.1}$$

where $\mathbf{x}$ is a surface point and $\mathbf{n}$ the surface normal at that point. $\mathbf{x}_z$ is the third coordinate of $\mathbf{x}$ in the camera coordinate system. Let us note that we forgot to put the visibility term in Equation (A.1) on purpose.

Equation (A.1) can be shown easily. Let $\mathrm{d}\mathbf{w}$ be the solid angle unit measure that would see a surface patch $\mathrm{d}s$ at a point $\mathbf{x}$. Then it is easy to show that:

$$\mathrm{d}\mathbf{w} = \frac{\cos(\theta)}{|\mathbf{x}|^2}\,\mathrm{d}s = -\frac{\mathbf{n}\cdot\mathbf{x}}{|\mathbf{x}|^3}\,\mathrm{d}s\,, \tag{A.2}$$

where θ is the angle between the viewing direction and the surface's normal $\mathbf{n}$. Similarly, one can write the unit solid angle measure $\mathrm{d}\mathbf{w}$ with respect to the image unit area $\mathrm{d}\mathbf{p}$:

$$\mathrm{d}\mathbf{w} = \frac{\cos(\alpha)}{|\mathbf{y}|^2}\,\mathrm{d}s\,, \tag{A.3}$$

where $\mathbf{y}$ Finally we have:

$$\mathrm{d}\mathbf{w} = \frac{\mathbf{x}_z^3}{|\mathbf{x}|^3\,f^2}\,\mathrm{d}\mathbf{p}\,. \tag{A.4}$$

Finally, from Equations (A.2) and (A.4), and by adding the visibility function $\nu_{\mathcal{S}}$ we have the following result:

$$\mathrm{d}\mathbf{p} = f^2\,\frac{\mathbf{n}\cdot\mathbf{x}}{\mathbf{x}_z^3}\,\nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s\,. \tag{A.5}$$

A.2 Computational Details on Gradient Flows for Triangular Meshes

A.2.1 Expression of $A'_j[0]$

We have to explicit $A'_j[0]$ linearly in function of the $\mathbf{V}$. The area $A_j[t]$ of a triangle $\mathcal{S}_j[t]$ is $A_j[t] = \frac{1}{2}\,|\overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k1}[t]} \wedge \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k2}[t]}|$ where $\mathbf{x}_k[t],\ \mathbf{x}_{k1}[t],\ \mathbf{x}_{k2}[t]$ are the vertices of triangle i at time t. For more convenience, we express the squared area A_j^2 to avoid squared root while computing the differential at $t = 0$. Then we have :

$$A_j^2[t] = \frac{1}{4}\left(\overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k1}[t]} \wedge \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k2}[t]}\right)\cdot\left(\overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k1}[t]} \wedge \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k2}[t]}\right)$$

$$\begin{aligned}\frac{d}{dt}A_j^2[t]\Big|_{t=0} = \frac{1}{2}\,&\left(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\right)\\ &\cdot\left(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k + \overrightarrow{\mathbf{x}_{k2}\mathbf{x}_{k}} \wedge \mathbf{V}_{k1} + \overrightarrow{\mathbf{x}_{k}\mathbf{x}_{k1}} \wedge \mathbf{V}_{k2}\right).\end{aligned}$$

Using $\frac{d}{dt}A_j^2[t]\Big|_{t=0} = 2\,A_j[0]\,\frac{d}{dt}A_j[t]\Big|_{t=0}$, we get

$$A'_j[0] = \sum_{k\in K_j}\mathbf{V}_k\cdot\left(\frac{1}{2}\mathbf{n}_j \wedge \mathbf{e}_{j,k}\right)\,. \tag{A.6}$$

If we move only one vertex at once (meaning V_{k1} & V_{k2} are null for vertex k), we have :

$$A'_j[0] = \frac{1}{2}\,\left(\mathbf{n}_j \wedge \overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}\right)\cdot\mathbf{V}_k\,. \tag{A.7}$$

A.2.2 Expression of $\mathbf{n}'_j[0]$

We have to explicit $\mathbf{n}'_j[0]$ linearly in function of the $\mathbf{V}$. Considering $\mathbf{n}_j = \frac{\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}}{2\,A_j}$, we have :

$$\mathbf{n}'_j[0] = \frac{1}{2\,A_j[0]^2}\Big((\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}})'[0]\,A_j[0] \\ - \big(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\big)\,A'_j[0]\Big).$$

$$\mathbf{n}'_j[0] = \frac{1}{2\,A_j}\Big(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k + \overrightarrow{\mathbf{x}_{k2}\mathbf{x}_k} \wedge \mathbf{V}_{k1} + \overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \mathbf{V}_{k2} \\ - \big((\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k + \overrightarrow{\mathbf{x}_{k2}\mathbf{x}_k} \wedge \mathbf{V}_{k1} + \overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \mathbf{V}_{k2}) \cdot \mathbf{n}_j\big)\,\mathbf{n}_j\Big).$$

So

$$\mathbf{n}'_j[0] = \frac{1}{2\,A_j}\left(\left(\sum_{k\in K_j} \mathbf{e}_{j,k} \wedge \mathbf{V}_k\right) - \left((\sum_{k\in K_j} \mathbf{e}_{j,k} \wedge \mathbf{V}_k) \cdot \mathbf{n}_j\right)\mathbf{n}_j\right). \tag{A.8}$$

Therefore $\mathbf{n}'_j[0]$ is the projection of $\sum_{k\in K_j} \mathbf{e}_{j,k} \wedge \mathbf{V}_k$ on the orthogonal plane to $\mathbf{n}_j$, divided by $2\,A_j$.

In the case where we consider moving only one vertex at once (meaning $\mathbf{V}_{k1}$ & $\mathbf{V}_{k2}$ are null for vertex k), we have :

$$\mathbf{n}'_j[0] = \frac{\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k - \big((\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k) \cdot \mathbf{n}_j\big)\,\mathbf{n}_j}{2\,A_j}\,. \tag{A.9}$$

A.2.3 Details on the Lambertian Case Using Illumination

All the terms in Equation (III.18) are explicit at the exception of $\nabla_{\mathbf{x}} R(\mathbf{x}, \mathbf{n}_j)$. In fact $\nabla_{\mathbf{x}} R(\mathbf{x}, \mathbf{n}_j) = \nabla\rho(\mathbf{x})\mathrm{L}(\mathbf{x}, \mathbf{n}_j) + \rho(\mathbf{x})\nabla_{\mathbf{x}}\mathrm{L}(\mathbf{x}, \mathbf{n}_j)$ where we denote

$$\mathrm{L}(\mathbf{x}, \mathbf{n}_j) = \sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x})(\mathbf{n}_j \cdot \mathbf{l}_l(\mathbf{x})) + E_0\,.$$

In the case of a homogeneous albedo (typically in shape from shading) we have $\nabla\rho(\mathbf{x}) = 0$.

According to (III.20), for $\mathbf{x}$ in $\mathcal{S}_j$, we have $\nabla\rho(\mathbf{x}) = \dfrac{b\,\nabla a - a\,\nabla b}{b^2}$, where a and b are defined by

$$\begin{aligned} a &= \sum_i I_i(\pi_i(\mathbf{x}))\nu_{\mathcal{S},i}(\mathbf{x})\,, \\ b &= \sum_i \left(\sum_{l=1}^{n_L^i} L_l^i \nu_{l^i,\mathcal{S}}(\mathbf{x})(\mathbf{n}_j \cdot \mathbf{l}_l^i(\mathbf{x})) + E_0\right)\nu_{\mathcal{S},i}(\mathbf{x}) \end{aligned} \tag{A.10}$$

and where ∇a and ∇b are

$$\begin{aligned}\nabla a &= \sum_i D\pi(\mathbf{x})^\top \nabla I_i(\pi_i(\mathbf{x}))\nu_{\mathcal{S},i}(\mathbf{x})\ , \\ \nabla b &= \sum_i \left(\sum_{l=1}^{n_L^i} L_l^i \nu_{l^i,\mathcal{S}}(\mathbf{x})(\mathbf{n}_j \cdot \nabla \mathbf{l}_l^i(\mathbf{x})) \right) \nu_{\mathcal{S},i}(\mathbf{x}).\end{aligned} \tag{A.11}$$

(We assume here that visibilities are the same for all the points $\mathbf{x}$ on the triangle $\mathcal{S}_j$, or we neglect their variations). For scenes illuminated by far light sources we have $\nabla \mathbf{l}_l^i(\mathbf{x}) \approx 0$, and so $\nabla \rho \approx a'/b$. Moreover, if the light sources are same for all the image, then

$$b = (\sum_i \nu_{\mathcal{S},i}(\mathbf{x})) \left(\sum_{l=1}^{n_L} L_l(\mathbf{n}_j \cdot \mathbf{l}_l)\nu_{l,\mathcal{S}}(\mathbf{x}) + E_0 \right)\ .$$

Finally, neglecting the variations of the light visibility, we have

$$\nabla_{\mathbf{x}} \mathrm{L}(\mathbf{x}, \mathbf{n}_j) = \sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) D\mathbf{l}_l(\mathbf{x})^\top \mathbf{n}_j,$$

where $D\mathbf{l}_l(\mathbf{x})^\top$ is the transposition of the differential of $\mathbf{l}_l$ (3×3 matrix). In the case of far light sources, we have $\nabla_{\mathbf{x}} \mathrm{L}(\mathbf{x}, \mathbf{n}_j) \approx 0$.

APPENDIX B

Related Publications of the Author

International Journal Papers

- **Amaël Delaunoy** and Emmanuel Prados.
  Gradient Flows for Optimizing Triangular Mesh-based Surfaces: Applications to 3D Reconstruction Problems dealing with Visibility.
  The ***International Journal of Computer Vision***, Volume 95, page 100–127 - 2011.

International Conferences Papers

- **Amaël Delaunoy**, Emmanuel Prados and Peter N. Belhumeur.
  Towards Full 3D Helmholtz Stereovision Algorithms.
  Proceedings of the 10th ***Asian Conference on Computer Vision***, Queenstown, New-Zealand - November 2010.

- Zsolt Janko, **Amaël Delaunoy** and Emmanuel Prados.
  Colour Dynamic Photometric Stereo for Textured Surfaces.
  Proceedings of the 10th ***Asian Conference on Computer Vision***, Queenstown, New-Zealand - November 2010.

- Peter Sturm, **Amaël Delaunoy**, Pau Gargallo, Emmanuel Prados and Kuk-Jin Yoon.
  3D and Appearance Modeling from Images.
  Proceedings of the 14th ***Iberoamerican Congress on Pattern Recognition***, Guadalajara, Mexico, Volume 5856, page 694–704 - November 2009.

- **Amaël Delaunoy**, Ketut Fundana, Emmanuel Prados and Anders Heyden.
  Convex Multi-Region Segmentation on Manifolds.
  Proceedings of the 12th ***International Conference on Computer Vision***, Kyoto, Japan - September 2009.

- **Amaël Delaunoy**, Emmanuel Prados, Pau Gargallo, Jean-Philippe Pons and Peter Sturm.
  Minimizing the Multi-view Stereo Reprojection Error for Triangular Surface Meshes.
  Proceedings of the 19th ***British Machine Vision Conference***, Leeds, UK - September 2008.

International Workshops Papers and Technical Reports

- Ryo Furukawa, Ryusuke Sagawa, **Amaël Delaunoy** and Hiroshi Kawasaki.
  Multiview Projectors/Cameras System for 3D Reconstruction of Dynamic Scenes.
  Proceedings of the First IEEE Workshop on ***Dynamic Shape Capture and Analysis*** (in conjunction with ICCV 2011) - November 2011.

- Kuk-Jin Yoon, **Amaël Delaunoy**, Pau Gargallo and Peter Sturm.
  Toward Global and Model based Multiview Stereo Methods for Shape and Reflectance Estimation.
  Proceedings of the First International Workshop on ***Photometric Analysis For Computer Vision*** (in conjunction with ICCV'07) - October 2007.

- Kuk-Jin Yoon, Emmanuel Prados, Peter Sturm, **Amaël Delaunoy** and Pau Gargallo.
  Shape and Reflectance Recovery using Multiple Images with Known Illumination Conditions.
  Technical Report RR-6309, INRIA Rhône-alpes, Number RR-6309 - September 2007.

National Conferences Papers and Others

- **Amaël Delaunoy**, Emmanuel Prados, Ketut Fundana and Anders Heyden.
  Segmentation convexe multi-région de données sur les surfaces.
  Actes du 17ème Congrès de ***Reconnaissance des Formes et Intelligence Artificielle***, Caen, France - RFIA 2010 - Janvier 2010.

- **Amaël Delaunoy**, Emmanuel Prados, Pau Gargallo, Jean-Philippe Pons and Peter Sturm.
  Stéréo multi-vues: erreur de reprojection et maillages triangulaires.
  Actes des Journées ***ORASIS***, Trégastel, France - Juin 2009.

- **Amaël Delaunoy**.
  Vers une Méthode Globale de Reconstruction Multi-vues pour l'Estimation de Surface et d'Apparence.
  Master thesis, ***Université de Lyon*** *(INSA and CPE)* - September 2007.

Bibliography

[1] The stanford 3d scanning repository - http://graphics.stanford.edu/data/3dscanrep/. 2, 43, 76

[2] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org/. 56

[3] E. Aganj, J.-P. Pons, F. Ségonne, and R. Keriven. Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, Oct 2007. 18, 27, 118

[4] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *International Conference on Computer Vision*, 2009. 13

[5] B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 28(1):106–118, 2006. 10, 83

[6] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72:239–257, 2007. 10.1007/s11263-006-8815-7. 23

[7] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, 1999. 17

[8] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, 174(2):759–780, 2001. 32

[9] N. Birkbeck, D. Cobzas, P. Sturm, and M. Jägersand. Variational shape and reflectance estimation under changing light and viewpoints. In *European Conference on Computer Vision*, volume 1, pages 536–549. Springer, may 2006. 6, 7, 17, 28, 34, 35, 73

[10] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006. 11

[11] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. AK Peters / CRC press, 2010. 27

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003. 87

[13] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *IEEE International Conference on Computer Vision*, 2001. 83

[14] X. Bresson, S. Esedoglu, P. Vandergheynst, J. P. Thiran, and S. J. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, June 2007. 21, 83, 84

[15] T. Brox, A. A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, 2004. 14, 24

[16] C. Cagniart, E. Boyer, and S. Ilic. Iterative deformable surface tracking in multi-view setups. In *3DPVT*, 2010. 6

[17] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. In *ECCV*, 2010. 18

[18] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, (1), 1997. 26, 83

[19] A. Chambolle, D. Cremers, and T. Pock. A convex approach for computing minimal partitions. Technical Report R.I.649, CMAP CNRS, November 2008. 21, 83, 84

[20] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011. 10.1007/s10851-010-0251-1. 83, 95

[21] T. Chan and L. Vese. Active contour without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 20, 83, 84

[22] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *Journal of Applied Mathematics*, 2006. 21, 83, 84, 87

[23] J. Y. Chang, K. M. Lee, and S. U. Lee. Multiview normal field integration using level set methods. In *IEEE Conference in Computer Vision and Pattern Recognition*, 2007. 32, 35, 43, 72, 73

[24] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven, and O. Faugeras. Generalized gradients: Priors on minimization flows. *International Journal of Computer Vision*, 73(3):325–344, Jul 2007. 10, 35, 38, 52, 80, 117

[25] T. Chen, M. Goesele, and H.-P. Seidel. Mesostructure from specularity. In *CVPR (2)*, pages 1825–1832, 2006. 23

[26] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, pages 7–24, 1982. 7

[27] J. Courchay, J.-P. Pons, P. Monasse, and R. Keriven. Dense and accurate spatio-temporal multi-view stereovision. In *Proc. Asian Conference on Computer Vision*, pages 11–22, 2009. 18, 27, 63, 118

[28] D. Cremers, T. Pock, K. Kolev, and A. Chambolle. Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, 2011. 26, 35, 117

[29] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, Apr. 2007. 9, 83

[30] E. Debreuve, M. Gastaud, M. Barlaud, and G. Aubert. Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation. *Journal of Mathematical Imaging and Vision*, 2007. 28, 36, 38, 105

[31] A. Delaunoy and E. Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International Journal of Computer Vision*, 95:100–123, 2011. 10.1007/s11263-010-0408-9. 3, 6, 12, 16, 112

[32] A. Delaunoy, E. Prados, P. Gargallo, J.-P. Pons, and P. Sturm. Minimizing the multiview stereo reprojection error for triangular surface meshes. In *British Machine and Vision Conference, Leeds, UK*, 2008. 36, 48, 89, 104

[33] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics*, 33(Annual Conference Series):317–324, 1999. 32, 58

[34] M. P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, February 1976. 86

[35] J. Drareni, P. Sturm, and S. Roy. Methods for geometrical video projector calibration. *Machine Vision and Applications Journal*, 2011. 21

[36] Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. pages Vol III: 238–251, 2004. 6, 28, 35, 73

[37] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001. 11

[38] G. Dziuk and C. Elliott. Finite elements on evolving surfaces. *IMA journal of numerical analysis*, 27(2):262–292, 2007. 28, 35, 36, 37

[39] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. J. Kuo, and M. Desbrun. Generalized surface flows for mesh processing. In *Eurographics Symposium on Geometry Processing*, 2007. 28, 32, 36, 37, 38, 39, 52, 80, 104, 105, 117

[40] P. Einarsson, C.-F. Chabert, A. Jones, W.-C. Ma, B. Lamond, T. Hawkins, M. Bolas, S. Sylwan, and P. Debevec. Relighting human locomotion with flowed reflectance fields. In *Eurographics Symposium on Rendering*, 2006. 23, 24

[41] O. Faugeras. *Three-dimensional computer vision: A geometric viewpoint*. The MIT press, Cambridge, MA, 1993. 5

[42] O. Faugeras and R. Keriven. Variational-principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE TIP*, 7(3):336–344, 1998. 6, 9, 15, 26, 32, 34, 35

[43] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images*. The MIT press, Cambridge, MA, 2001. 3, 5

[44] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 13

[45] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. 2002. 3, 5

[46] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Jonhson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *European Conference on Computer Vision (ECCV)*, 2010. 13

[47] J.-S. Franco and E. Boyer. Efficient polyhedral modeling from silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:414–427, 2009. 16

[48] P. Fua. Reconstructing complex surfaces from multiple stereo views. In *International Conference on Computer Vision*, pages 1078–1085, 1995. 6

[49] R. Furukawa, H. Kawasaki, R. Sagawa, and Y. Yagi. Shape from grid pattern based on coplanarity constraints for one-shot scanning. *IPSJ Transaction on Computer Vision and Applications*, 1:139–157, 2009. 2

[50] R. Furukawa, R. Sagawa, A. Delaunoy, and H. Kawasaki. Multiview projectors/cameras system for 3d reconstruction of dynamic scenes. In *IEEE Workshop on Dynamic Shape Capture and Analysis (in conjonction with ICCV)*, 2011. 22, 63, 64, 65

[51] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. In *European Conference on Computer Vision*, Graz, Austria, May 2006. 19

[52] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 13, 19

[53] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. *International Journal of Computer Vision*, 81(1):53–67, 2009. 9

[54] D. Gallup, J.-M. Frahm, and M. Pollefeys. A heightmap model for efficient 3d reconstruction from street-level video. In *Int. Conf. on 3D Data Processing, Visualization and Transmission*, 2010. 6

[55] P. Gargallo. *Contributions to the Bayesian approach to Multi-view Stereo*. PhD thesis, Institut National Polytechique de Grenoble, France, February 2008. 4, 8, 11, 16, 25, 34, 36, 46, 48, 49, 60, 61, 73, 112, 116

[56] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *International Conference on Computer Vision*. IEEE Computer Society Press, 2007. 10, 12, 26, 32, 33, 34, 35, 36, 40, 45, 46, 49, 53, 56, 58, 59, 60, 61, 79, 119

[57] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2402–2409, Washington, DC, USA, 2006. IEEE Computer Society. 24, 100

[58] B. Goldlucke, I. Ihrke, C. Linz, and M. Magnor. Weighted minimal hypersurface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1194–1208, 2007. 32, 35

[59] B. Goldlücke and M. A. Magnor. Weighted minimal hypersurfaces and their applications in computer vision. In *European Conference on Computer Vision*, pages 366–378, 2004. 32

[60] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying brdfs from photometric stereo. In *International Conference on Computer Vision*, pages 341–348, Washington, DC, USA, 2005. IEEE Computer Society. 17, 23

[61] J.-Y. Guillemaut, O. Drbohlav, R. Sara, and J. Illingworth. Helmholtz stereopsis on rough and strongly textured surfaces. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 10–17, Washington, DC, USA, 2004. IEEE Computer Society. 24, 25, 101, 102, 108

[62] R. Hartley and P. Sturm. Triangulation. In *Computer Analysis of Images and Patterns*, volume 970 of *Lecture Notes in Computer Science*, pages 190–197. Springer Berlin / Heidelberg, 1995. 13

[63] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004. 3, 5, 13

[64] R. I. Hartley and R. Gupta. Linear pushbroom cameras. In *European Conference on Computer Vision '94*, pages 555–566. Springer-Verlag, 1994. 33

[65] H. Helmholtz. *Treatise on Physiological Optics*. Dover: New York, 1925. 24

[66] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004. 16, 34

[67] C. Hernandez, G. Vogiatzis, G. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Proc. ICCV'07*, pages 1–8, 2007. 23

[68] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008. 17, 19, 23, 24, 34, 43, 72, 73, 100

[69] A. Hertzmann and S. Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 27(8):1254–1264, 2005. 17, 23, 101

[70] T. Higo, Y. Matsushita, and K. Ikeuchi. Consensus photometric stereo. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1157 – 1164, june 2010. 23

[71] B. K. P. Horn and M. J. Brooks. The variational approach to shape from shading. *Comput. Vision Graph. Image Process.*, 33(2):174–208, February 1986. 17

[72] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *Proc. Intl. Conf. on Computer Vision*, Rio de Janeiro, Brasil, Oct. 2007. IEEE. 18

[73] I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Computer Graphics Forum*, 29(8):2400–2426, 2010. 18

[74] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25, 2003. 83

[75] A.-L. Jachiet, P. Labatut, and J.-P. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, Jun 2010. 79

[76] Z. Janko, A. Delaunoy, and E. Prados. Colour Dynamic Photometric Stereo for Textured Surfaces. In Springer, editor, *Asian Conference on Computer Vision*, Queenstown, Nouvelle-Zélande, Nov. 2010. 18, 23, 24

[77] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, jun 2010. 13

[78] H. Jin. *Variational Methods for Shape Reconstruction in Computer Vision*. PhD thesis, Electrical Engineering Department, Washington University, USA, August 2003. 18, 86, 89, 100

[79] H. Jin, D. Cremers, D. Wang, E. Prados, A. Yezzi, and S. Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, 76(3), March 2008. 17, 18, 24, 26, 66, 67, 79, 89, 100, 101

[80] H. Jin, D. Cremers, A. J. Yezzi, and S. Soatto. Shedding light on stereoscopic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 01, pages 36–42, 2004. 24, 79, 100

[81] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005. 15, 19, 32, 34, 35

[82] H. Jin, A. Yezzi, and S. Soatto. Variational multiframe stereo in the presence of specular reflections. In *3DPVT*, pages 626–630, 2002. 6, 9, 19, 32, 35

[83] H. Jin, A. J. Yezzi, and S. Soatto. Mumford-shah on the move: Region-based segmentation on deforming manifolds with application to 3-d reconstruction of shape and appearance from multi-view images. *Journal of Mathematical Imaging and Vision*, 29(2-3):219–234, 2007. 82, 85, 90

[84] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 16(9):920 – 932, September 1994. 13

[85] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988. 26, 83

[86] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007. 13

[87] J. Koenderink. *Solid shape*. MIT Press, 1990. 6

[88] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Dec. 2007. 83

[89] K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. 2008. 32, 35

[90] K. Kolev and D. Cremers. Continuous ratio optimization via convex relaxation with applications 3d multiview reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 35

[91] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 2009. 6, 15, 21, 26, 32, 35

[92] K. Kolev, T. Pock, and D. Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In *European Conference on Computer Vision*, 2010. 35

[93] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 26(2):147–159, February 2004. 26, 35, 83

[94] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 9

[95] P. Labatut. *Labeling of data-driven complexes for surface reconstruction*. PhD thesis, Université Paris Diderot-Paris VII, Sep 2009. 6, 19, 79

[96] P. Labatut, R. Keriven, and J.-P. Pons. Fast level set multi-view stereo on graphics hardware. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 774–781, Washington, DC, USA, 2006. IEEE Computer Society. 32, 35

[97] P. Labatut, J.-P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *International Conference on Computer Vision*, 2007. 19

[98] J.-O. Lachaud and B. Taton. Deformable model with adaptative mesh and automated topology hanges. In *International Conference on 3-D Digital Imaging and Modeling*, 2003. 27

[99] D. Lanman, D. Crispell, and G. Taubin. Surround structured lighting for full object scanning. In *The 6th*. IEEE, 2007. 21

[100] S. Lazebnik, Y. Furukawa, and J. Ponce. Projective visual hulls. *International Journal of Computer Vision*, 74:137–165, 2007. 10.1007/s11263-006-0008-x. 16

[101] J. Lellmann, D. Breitenreicher, and C. Schnörr. Fast and exact primal-dual iterations for variational problems in computer vision. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 494–505. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-15552-9_36. 95

[102] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In X.-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie, editors, *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 150–162. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-02256-2_13. 21, 83, 84, 93, 94

[103] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 22

[104] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 10.1023/B:VISI.0000029664.99615.94. 13

[105] Z. Lu, Y.-W. Tai, M. Ben-Ezra, and M. S. Brown. A framework for ultra high resolution 3D imaging. In *2010 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1205–1212, June 2010. 2

[106] S. P. Mallick, T. Zickler, D. J. Kriegman, and P. N. Belhumeur. Beyond lambert: Reconstructing specular surfaces using color. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 619–626, 2005. 17, 100

[107] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–423, July 2001. 92

[108] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. 2002. 32, 35, 42, 43, 105

[109] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of rn. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986. 87

[110] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 27(10):1615–1630, October 2005. 13

[111] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: Topology, geometry and deformation. *Image and Vision Computing*, 2001. 26

[112] D. Mumford. Bayesian rationale for energy functionals. In *Geometry-driven diffusion in Computer Vision*, pages 141–153. 1994. 8

[113] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *Comm. on Pure and Applied Math.*, XLII(5):577–685, 1988. 83, 84

[114] S. Nayar. Computational Cameras: Approaches, Benefits and Limits. Technical report, Jan 2011. 22

[115] S. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *Robotics and Automation, IEEE Transactions on*, 6(4):418 –431, aug 1990. 7, 18

[116] S. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)*, Jul 2006. 114

[117] S. K. Nayar. Computational Cameras: Redefining the Image. *IEEE Computer Magazine, Special Issue on Computational Photography*, pages 30–38, Aug 2006. 22

[118] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. In *SIGGRAPH*, 2005. 35, 74

[119] R. Newcombe and A. Davison. Live dense reconstruction with a single moving camera. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2010. 14

[120] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, 2011. 13, 21

[121] A. Ngan, F. Durand, and W. Matusik. Experimental analysis of brdf models. In *Proceedings of the Eurographics Symposium on Rendering*, pages 117–226. Eurographics Association, 2005. 7

[122] M. Oren and S. K. Nayar. Generalization of lambert's reflectance model. In *SIGGRAPH*, pages 239–246, New York, NY, USA, 1994. ACM Press. 7

[123] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988. 26, 35

[124] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97:259–282, 2005. 83

[125] S. Parkes, M. Dunstan, I. Martin, M. McCrum, and O. Dubois-Matra. Testing advanced navigation systems for planetary landers and rovers. In *International Astronautical Congress, Paper IAC-09-A3.1.6*, 2009. 69

[126] B. Petit, A. Letouzey, E. Boyer, and J.-S. Franco. Surface Flow from Visual Cues. In *Vision, Modeling and Visualization Workshop*, Berlin, Germany, Oct. 2011. 18

[127] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 7

[128] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision*, pages III: 792–805, 2008. 83

[129] J. Pons and J. Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In *International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 27, 28, 35, 56, 104, 107

[130] J. P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 822–827 vol. 2, 2005. 6, 9, 15, 17, 33, 34, 58, 119

[131] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007. 14, 18, 26, 32, 35, 79, 100

[132] E. Prados. *Application of the theory of the viscosity solutions to the Shape From Shading problem*. PhD thesis, University of Nice Sophia-Antipolis, Oct. 2004. 17

[133] Y. Sato. *Object shape and reflectance modeling from color image sequence*. PhD thesis, The Robotics Institute, Carnegie Mellon University, 1997. 7

[134] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 9

[135] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37:151–172, 2000. 10.1023/A:1008199403446. 13

[136] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006. 2, 9, 60, 61, 62, 107

[137] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151 – 173, 1999. 9, 15

[138] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation. In *IEEE International Conference on Computer Vision*, pages 349–356, 2005. 16, 26, 34

[139] G. Slabaugh and G. Unal. Active polyhedron: surface evolution theory applied to deformable meshes. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:84–91 vol. 2, 2005. 28, 36

[140] S. Soatto, A. J. Yezzi, and H. Jin. Tales of shape and radiance in multi-view stereo. In *IEEE International Conference on Computer Vision*, pages 974–981, 2003. 19, 33, 34, 58, 119

[141] J. Solem and N. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. pages 419–430, 2005. 6, 9, 10, 32, 34, 35, 37, 40

[142] J. E. Solem. *Variational Problems and Level Set Methods in Computer Vision - Theory and Applications*. PhD thesis, Lund University, 2006. 32, 35

[143] J. E. Solem, H. Aanaes, and A. Heyden. A variational analysis of shape from specularities using sparse data. In *3DPVT*, pages 26–33, Washington, DC, USA, 2004. IEEE Computer Society. 32, 35

[144] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 2007. 93

[145] C. Strecha, W. V. Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition*, 2008. 2, 9

[146] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. 2, 3, 5, 8

[147] E. Tola, V. Lepetit, and P. Fua. Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. volume 32, pages 815–830, May 2010. 13

[148] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.*, 57(9):1105–1112, Sep 1967. 7

[149] P. Tu and P. R. S. Mendon. Surface reconstruction via helmholtz reciprocity with a single image pair. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:541, 2003. 24, 25, 101, 112

[150] P. Viola and W. M. Wells, III. Alignment by maximization of mutual information. *Int. J. Comput. Vision*, 24:137–154, September 1997. 14

[151] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3):97, 2008. 6

[152] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 28(5), Dec. 2009. 24, 43, 72, 73, 74, 77, 78

[153] G. Vogiatzis, C. Hernández, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 2007. 16, 26, 34

[154] H.-H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Jun 2009. 2, 6, 13, 17, 19, 28, 36, 79, 100, 104

[155] W. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 03 1996. 14

[156] T. Weyrich, J. Lawrence, H. Lensch, S. Rusinkiewicz, and T. Zickler. Principles of appearance acquisition and representation. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 80:1–80:119, New York, NY, USA, 2008. ACM. 7

[157] R. T. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998. 32, 35

[158] S. Yamazaki, S. G. Narasimhan, S. Baker, and T. Kanade. Coplanar shadowgrams for acquiring visual hulls of intricate objects. In *IEEE International Conference of Computer Vision*, 2007. 22

[159] A. Yezzi and S. Soatto. Stereoscopic segmentation. *IEEE International Conference on Computer Vision*, 2001. 45, 82, 89

[160] A. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal of Computer Vision*, 53(1):31–43, 2003. 6, 9, 12, 16, 32, 33, 34, 55, 56, 58, 59, 119

[161] K.-J. Yoon, A. Delaunoy, P. Gargallo, and P. Sturm. Toward global and model based multiview stereo methods for shape and reflectance estimation. In *Proceedings of the First International Workshop on Photometric Analysis For Computer Vision (in conjunction with ICCV'07)*, 2007. 19

[162] K.-J. Yoon and I.-S. Kweon. Correspondence search in the presence of specular highlights using specular-free two-band images. In *Asian Conference on Computer Vision*, pages 761–770, 2006. 17

[163] K.-J. Yoon, E. Prados, and P. Sturm. Joint estimation of shape and reflectance using multiple images with known illumination conditions. *International Journal of Computer Vision*, 2009. 7, 11, 17, 18, 19, 25, 26, 32, 35, 66, 79, 100

[164] T. Yu, N. Xu, and N. Ahuja. Recovering shape and reflectance model of non-lambertian objects from multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 226–233, 2004. 18, 100

[165] T. Yu, N. Xu, and N. Ahuja. Shape and view independent reflectance map from multiple views. *International Journal of Computer Vision*, 73(2):123–138, 2007. 34

[166] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Proceedings of Vision, Modeling, and Visualization*, 2008. 21, 83, 84

[167] A. Zaharescu, E. Boyer, and R. P. Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *Asian Conference on Computer Vision*, LNCS, Tokyo, Japan, November 2007. Springer. 27, 28, 35, 56, 93, 104

[168] C. Zhou, O. Cossairt, and S. Nayar. Depth from Diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2010. 22

[169] C. Zhou, S. Lin, and S. Nayar. Coded aperture pairs for depth from defocus and defocus deblurring. *International Journal of Computer Vision*, 93:53–72, 2011. 10.1007/s11263-010-0409-8. 22

[170] T. Zickler. Reciprocal image features for uncalibrated helmholtz stereopsis. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1801–1808, 2006. 24, 25, 101

[171] T. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2-3):215–227, 2002. 24, 25, 101, 102, 107

[172] T. Zickler, J. Ho, D. Kriegman, J. Ponce, and P. Belhumeur. Binocular helmholtz stereopsis. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1411–1417 vol.2, Oct. 2003. 24, 25, 101, 102, 106, 107

[173] T. Zickler, S. P. Mallick, D. J. Kriegman, and P. Belhumeur. Color subspaces as photometric invariants. *International Journal of Computer Vision*, 2008. 17, 100

[174] D. Zikic, A. Kamen, and N. Navab. Natural gradients for deformable registration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 10, 80