Reconstruction de scène dynamique à partir de plusieurs vidéos mono et multiscopiques par hybridation de méthodes “ silhouettes ” et “ multi-stéréovision ”

Muhannad Ismael

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

ÉCOLE DOCTORALE SCIENCES TECHNOLOGIE SANTE (547)

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Discipline : ***INFORMATIQUE***

Présentée et soutenue publiquement par

Muhannad ISMAEL

Le 12 juillet 2016

RECONSTRUCTION DE SCÈNE DYNAMIQUE À PARTIR DE PLUSIEURS VIDÉOS MONO ET MULTISCOPIQUES PAR HYBRIDATION DE MÉTHODES « SILHOUETTES » ET « MULTI-STÉRÉOVISION »

Thèse dirigée par **Céline LOSCOS et Yannick REMION**

JURY

M. El Mustapha MOUADDIB,	, Professeur,	à l'Université d'Amiens Picardie Jules Verne,	**, Président**
Mme. Raphaëlle CHAINE,	, Professeur,	à l'Université de Lyon 1 Claude Bernard,	**, Rapporteur**
M. Jacques VERLY,	, Professeur,	à l'Université de Liège,	**, Rapporteur**
M. Mathieu DESBRUN,	, Professeur,	CALTECH, USA,	**, Examinateur**
Mme. Céline LOSCOS,	, Professeur,	à l'Université Reims Champagne-Ardenne,	**, Examinateur**
Mme. Stéphanie PREVOST,	, Maître de Conférences HDR,	à l'Université Reims Champagne-Ardenne,	**, Examinateur**
M. Yannick REMION,	, Professeur,	à l'Université Reims Champagne-Ardenne,	**, Invité**

Résumé

La reconstruction précise d'une scène 3D à partir de plusieurs caméras offre un contenu synthétique 3D à destination de nombreuses applications telles que le divertissement, la télévision et la production cinématographique. Cette thèse propose une nouvelle approche pour la reconstruction 3D multi-vues basée sur l'enveloppe visuelle et la stéréovision multi-oculaire. Cette approche nécessite en entré l'enveloppe visuelle et plusieurs jeux d'images rectifiées issues de différents *unités multiscopiques* constituées chacune de plusieurs caméras alignées et équidistantes. Nos contributions se situent à différents niveaux. Le premier est notre méthode de stéréovision multi-oculaire qui est fondé sur un nouvel échantillonnage de l'espace scénique et fournit une *carte de matérialité* exprimant la probabilité pour chaque point d'échantillonnage 3D d'appartenir à la surface visible par l'unité multiscopique. Le second est l'hybridation de cette méthode avec les informations issues de l'enveloppe visuelle et le troisième est la chaîne de reconstruction basée sur la fusion des différentes enveloppes creusées tout en gérant les informations contradictoires qui peuvent exister. Les résultats confirment : i) l'efficacité de l'utilisation de la carte de matérialité pour traiter les problèmes qui se produisent souvent dans la stéréovision, en particulier pour les régions partiellement occultées ; ii) l'avantage de la fusion des méthodes de l'enveloppe visuelle et de la stéréovision multi-oculaire pour générer un modèle 3D précis de la scène.

Mots-clés : Reconstruction 3D à partir de multiples vues, Stéréovision multi-vue, Enveloppe visuelle, Géométrie épipolaire parallèle décentrée, Reconstruction basée silhouette.

Abstract

Accurate reconstruction of a 3D scene from multiple cameras offers 3D synthetic content to be used in many applications such as entertainment, TV, and cinema production. This thesis is placed in the context of the RECOVER3D collaborative project, which aims is to provide efficient and quality innovative solutions to 3D acquisition of actors. The RECOVER3D acquisition system is composed of several tens of synchronized cameras scattered around the observed scene within a chromakey studio in order to build the visual hull, with several groups laid as *multiscopic units* dedicated to multi-baseline stereovision. A multiscopic unit is defined as a set of aligned and evenly distributed cameras. This thesis proposes a novel framework for multi-view 3D reconstruction relying on both multi-baseline stereovision and visual hull. This method's inputs are a visual hull and several sets of multi-baseline views. For each such view set, a multi-baseline stereovision method yields a surface which is used to carve the visual hull. Carved visual hulls from different view sets are then fused iteratively to deliver the intended 3D model. Furthermore, we propose a framework for multi-baseline stereo-vision which provides upon the Disparity Space (DS), a *materiality map* expressing the probability for 3D sample points to lie on a visible surface. The results confirm i) the efficient of using the materiality map to deal with commonly occurring problems in multi-baseline stereovision in particular for semi or partially occluded regions, ii) the benefit of merging visual hull and multi-baseline stereovision methods to produce 3D objects models with high precision.

Keywords: Multiview 3D reconstruction, Multi-baseline stereovision, Visual hull, Decentered parallel geometry, Shape from silhouette.

Remerciements

Je souhaite remercier en premier lieu tous les membres du jury pour avoir accepté de juger mes travaux.

Je souhaite ensuite remercier mes encadrants M. Yannick Rémion, Mme. Céline Loscos, et Mme. Stéphanie Prévost qui ont su conseiller, diriger, et parfois canaliser mes ardeurs scientifiques, afin de mener à bien ce travail, dont je suis fier aujourd'hui.

Je tiens également à remercier M. Philippe Souchet et M. Cédric Niquin de l'entreprise XD Production pour leur collaboration fructueuse et passionnée au sein du projet RECOVER3D.

Un grand merci également à tous les membres du laboratoire CReSTIC M. Laurent Lucas, M. Frédéric Blanchard, Mme. Sylvia Chalençon Piotin, M. Aassif Benassarou, M. Gilles Valette, M. Nicolas Passat, M. Cyril de Runz, M. Éric Desjardin, Mme. Barbara Romaniuk pour leur amabilité.

Je tiens à remercier mes amis et collègues du laboratoire CReSTIC particulièrement Ludovic, Jennifer, Jimmy, Mohammed, Maryam, Francisco, Joël, Pierre, Jonathan, Romain, Ulysse, Idriss, Exaverine,... et bien sûr les autres car j'en oublie beaucoup!

Enfin j'adresse ma plus profonde et chaleureuse gratitude à ma chère Sanna, à mes parents, et à ma famille, qui ont tous joué un rôle déterminant dans le développement de ma curiosité et de ma passion pour la science, de par leur confiance et leur soutien inconditionnel.

Table des matiéres

Table of contents

List of figures

List of tables

Introduction générale

Le travail présenté dans ce manuscrit s'inscrit dans le projet RECOVER3D (Realtime Environnement for COmputational Video Editing and Rendering in 3D) labellisé dans le cadre des « Investissements d'avenir » dont l'objectif est d'élaborer le premier système intégré de vidéo virtuelle pour le marché de la télévision et du cinéma. L'innovation apportée par RECOVER3D vise à libérer la création d'images vidéo des contraintes matérielles classiques liées à la prise de vue multi-caméras grâce à un nouveau système de « clonage virtuel » d'acteurs et de décor, basé sur des captures vidéo 3D intelligentes délivrant nativement une information de profondeur. L'université de Reims Champagne-Ardenne, via le laboratoire CReSTIC, participe au projet RECOVER3D en partenariat avec son porteur, la société XD-Production. L'objectif majeur de cette thèse vise à l'amélioration des solutions de reconstruction 3D de la scène.

D'une part, la reconstruction 3D d'une scène à partir de plusieurs images est depuis longtemps un problème majeur de la recherche en vision par ordinateur et de nombreuses approches telles que la reconstruction basée silhouette, stéréovision ou scanner 3D à lumière structurée ont été proposées. Elles sont généralement classées en deux groupes principaux : « active » et « passive ». Les méthodes dites « active » nécessitent une acquisition avec un matériel autre qu'une caméra comme un laser ou un vidéoprojecteur pour celles basées lumière structurées. Bien que la reconstruction obtenue soit de meilleure qualité que pour celles dites « passives », elles ont pour principaux inconvénients d'imposer des contraintes sur l'éclairage de la scène, de restreindre le champs de déplacement des éléments dynamiques de la scène et de gêner l'acquisition des textures réelles. Face au contexte de RECOVER3D, la restriction des mouvements des acteurs et une illumination contrôlée ne sont pas envisageables et cela nous amène donc à exclure toutes les méthodes dites « actives ».

D'autre part, préalablement à ce projet, la société XD-Production a développé un système de reconstruction 3D basé silhouette à partir de plusieurs cameras monoscopiques (à un seul point de vue) afin de modéliser une scène 3D. A l'issue de cette reconstruction, l'enveloppe visuelle obtenue est texturée avec les informations colorimétriques extraites

des images acquises. L'objectif majeur de cette thèse vise donc à proposer une nouvelle approche dite « passive » de reconstruction 3D d'une scène mono ou multi-objets acquise dans un studio dédié. Ce studio est composé de plusieurs caméras placées en cercle, sur deux niveaux, tout autour de la scène et regroupées en unité monoscopique (une seule caméra) et multiscopique (plusieurs caméras). Notre approche se base sur l'exploitation d'une part des techniques « basées silhouette » [64] et d'autre part de celles liées à la stéréovision multi-vues [73].

La reconstruction 3D basée silhouette est très utilisée dans les environnements multi-caméras. Elle est simple à implémenter, robuste, efficace et délivre une surface fermée. Cependant ses principaux désavantages sont le manque de précision du modèle et son incapacité à retrouver les zones concaves. En revanche la reconstruction 3D basée stéréovision produit une modélisation de haute résolution (zones convexes et concaves incluses) mais elle est plus complexe à implémenter et manque de robustesse. Ainsi les méthodes de stéréovision et les méthodes de silhouette s'avèrent être complémentaires. Bien que la littérature propose déjà des méthodes fusionnant ces deux techniques, nous présentons, dans cette thèse, un nouveau procédé pour les fusionner. Tout d'abord, nous commencerons par décrire une nouvelle méthode de stéréovision multi-vues, basée sur un système de capture aux centres optiques alignés que représente une unité multiscopique du studio RECOVER3D. Spécifiquement construite pour profiter du contexte multi-oculaire en géométrie parallèle décentrée, et en exploiter la géométrie multi-épipolaire simplifiée et régulière. Ensuite, nous en proposerons une hybridation, capable d'exploiter les informations des silhouettes, améliorant ainsi sa robustesse et son efficacité. Enfin nous terminerons par la description de la méthode de fusion des techniques « silhouettes » et « stéréovision » attendue par le projet RECOVER3D. Ces contributions sont exposées dans les chapitres 3 et 4 du manuscrit.

Ce manuscrit est constitué de 4 chapitres. Le chapitre 1 présente le projet RECOVER3D où nous détaillerons les spécificités du studio vidéo 3D et la problématique industrielle. Le chapitre 2 introduit le système de capture utilisé et la géométrie qui y est liée. Il revient sur le modèle sténopé d'une caméra et sur la géométrie épipolaire dans un contexte binoculaire avant d'étendre ces notions à la géométrie épipolaire multiple dans notre contexte de capture multi-oculaire. Une fois ces notions introduites, nous présenterons un état de l'art sur les techniques existantes de reconstruction basée stéréovision, basée silhouette et celles basées sur une fusion de ces deux approches. Le chapitre 3 aborde la reconstruction 3D partielle de la scène avec les informations issues d'une seule unité multiscopique. Il est consacré à notre méthode de reconstruction stéréovision multi-vues basée sur un système de capture multi-oculaire aligné et parallèle. Contrairement aux techniques existantes, notre approche délaisse l'espace image pour travailler principalement dans l'espace 3D et repose

sur notre carte de matérialité. Cette dernière exprime, pour chaque des points 3D de la scène, la probabilité d'appartenir à la surface reconstruite. Notre méthode de stéréovision multi-vues est dite basée scène. Le chapitre 4 traite de la reconstruction entière de la scène. Dans la première partie, une hybridation de notre méthode de stéréovision multi-vues basée sur un système de capture multi-oculaire aligné et parallèle est présentée. Cette hybridation se fait par la prise en compte des informations de l'enveloppe visuelle. La deuxième partie est consacrée à une nouvelle approche pour la fusion des modèles reconstruits nés des informations de chacune des unités multiscopiques et de l'enveloppe visuelle. Enfin la denière partie est dédiée à la présentation des résultats obtenues suivi d'une comparaison avec d'autres méthodes existantes et d'une discussion. Le chapitre 5 résume et conclut ce manuscrit et apporte quelques pistes et perspectives concernant nos travaux dans et en dehors du contexte RECOVER3D.

Chapter 1

Introduction

According to the increasing fragmentation of the TV audience due to the multiplication of channels and the appearance of new consumption behaviors (VOD, Internet ...), broadcasters and producers seek differentiated and content of quality, produced in optimal economic conditions. Among all paths considered in this regard, the use of 4D reconstruction studios are a sound alternative in the sense that it provides controlled environments generally based around a large room with uniform background equipped with multiple synchronized calibrated video cameras and appropriate illumination. The main application areas of 4D studios are currently dedicated to computer games, movies, TV productions, interactive media and motion analysis. The "4D studios" term refers to the spatio-temporal domain where 3D reconstructions of non-rigid moving objects are calculated. Most of these systems require a temporal sequence of simultaneous image shots from multiple viewpoints in addition to suitable software solutions to produce a static set of 3D models at each time step.

1.1 Context

This thesis presents the 3D reconstruction part of a broader project called RECOVER3D (acronym for Real-time Environment for COmputational Video Editing and Rendering in 3D). This project is born to fulfill needs of the broadcast industry of economically sustainable 3D post-production capabilities. More precisely, it aims at providing a new "virtual cloning" system of actors based on smart multi-video capture, natively delivering full 4D textured models of actors' performance. The RECOVER3D consortium is based on the partnership between academic researchers in computer vision and industrial integrators and producers from the broadcast world. Together, we designed and implemented a prototype of what could be a suitable shooting facility for the industrial production of 4D images. The constraint is not only to improve the overall esthetical quality of the resulting models, but

also to produce them in real time or in a reduced post processing delay, providing a credible alternative to standard 2D studios.

Figure 1.5 shows the pipeline of the RECOVER3D project for 4D reconstruction system purpose. It consists in four blocks, identified by four different colors in figure 1.5.

- The blocks with gray color in figure 1.5 represent the "Studio setting". This latter delivers a convenient studio whose layout is optimized according to the scenarist needs concerning useful scenic space where the actors and objects can be moved without leaving the intersection of all viewing frustum cameras. The studio setting begins with an interactive virtual configuration yielding a convenient layout. Then, the real configuration step places each camera in the studio according to this specification. Afterwards, the calibration process delivers extrinsic, intrinsic, and deformation parameters for each camera. These parameters are mandatory for the incoming shooting and reconstruction processes.
- In the red color block, the capture system provides synchronized corrected videos from all cameras for each rush.
- The modules in blue blockes implement the reconstruction of one 3D model of a scene for each frame. At each time-stamp, they combine reconstructed visual hull with the results of our multi-baseline stereovision.
- The sequence of these 3D models is then transmitted to the last blocks "4D model tracking" (green modules) in which motion flows are estimated in order to animate a dynamic mesh.

This thesis focuses on the 3D reconstruction from visual hull and multi-baseline stereovision methods. It concerns two blue blocks in figure 1.5 "Multi Stereo Matching" and "Fusion".

1.2 Problem statement

Reconstructing 3D objects from multiple views has long been a major research problem in computer vision. Many techniques such as multi-stereovision, shape from silhouette, shape from shading, and structured-light 3D scanner have been proposed for 3D reconstruction. They are usually classified as active or passive reconstruction. The active ones require controlled illumination such as a laser or a structured light. The passive ones rely only on the information contained in captured images. The main advantages of passive approaches are less restriction on the movement of the actors and the possibility of capturing actual textures. A main disadvantage is the lower visual quality of the 3D modeling compared to the precision obtained from some active approaches. Our project has to use passive reconstruction as live shooting of actual performances makes controlled illumination not desirable for our 4D textured model reconstruction. In this thesis, we propose a new multi-view passive approach

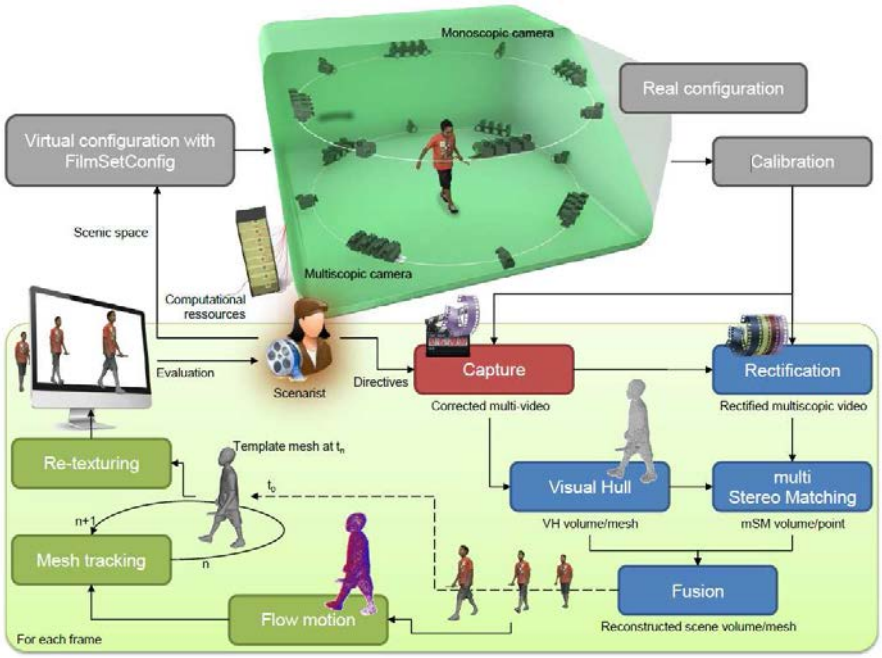


Fig. 1.1 Pipeline of the RECOVER3D project.

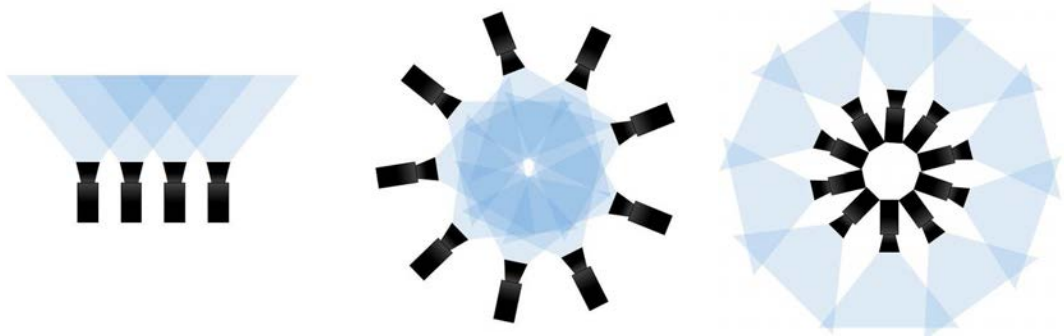
whose aim is to reach the visual quality and the precision of active approaches. Our method merges results from shape from silhouette and multi-baseline stereovision reconstructions.

Multiocular stereovision methods such as [39][60] conveniently reconstruct surface details and concave regions. However, they fail for textureless surfaces or repetitive textures because their core computational process relies on image texture. Shape from silhouette methods such as [13][68] are very useful in a multi-camera environment [12] and handle conveniently textureless and specular surfaces. However, their reconstruction quality is somehow limited as the produced visual hull (VH) cannot recover concave regions laying inside the optical beam passing through the silhouette for each camera. Thus, multi-stereovision and shape from silhouette are complementary to each other and numerous hybrid methods have already been published as we will describe in details in the next chapter. In this thesis, we propose a novel framework for 3D reconstruction combining both approaches using our proposed acquisition system and a novel multi-baseline stereovision framework.

1.3 Multi-camera Systems for 3D Video Production

One of the most important design factor of a 3D video studio is how to determine their spatial arrangement to achieve high 3D shape reconstruction accuracy. In general, the cam-

eras are spaced uniformly around the object(s) so that the captured images cover the entire object surface. If we do not have any specific knowledge about the object shape or motion, or if we want to capture a variety of objects in the same studio, one reasonable solution is to employ a circular ring camera arrangement, where a group of cameras placed evenly along the ring observes the object performing actions at the ring center. We may call it a converging acquisition system. Figure 1.2 illustrates three typical acquisition systems: diverging multi-camera arrangement for omni-directional image capture in 1.2c, parallel multi-camera arrangement for multi-baseline stereo and lightfield modeling 1.2a, and converging multi-camera arrangement 1.2b. Many research laboratories and companies are equipped with studio containing multiple cameras in converging acquisition system. Among those, the Kinovis room at INRIA Rhône-Alpes [7] is illustrated in figure 1.3a. A similar multi-camera system is also deployed at Surrey University in London [66] (see figure 1.3b). Note that it is often hard to satisfy a requirement of full observation coverage of the object surface. Some parts of the surface are occluded by others even when capturing a single object. Moreover, heavy occlusions become unavoidable when capturing multiple objects in action. Thus in order to produce a 3D video, there is a need for methods that cope with self and mutual occlusions. Many methods for 3D shape reconstruction from a set of multi-view images have been developed. One of the most popular methods is silhouette-based reconstruction. As pointed out before, since this method utilizes only silhouette information, many concave parts of the object cannot be reconstructed as we will describe in the next chapter. Contrary to existing studios, we propose to work with a novel acquisition system (see figure 1.4) that is composed of two typical multi-camera arrangements: parallel and converging which permits to exploit two kinds of reconstruction methods: i) multi-baseline stereovision, ii) visual hull.



(a) (b) (c)

Fig. 1.2 Acquisition systems for multi-view camera: a) parallel arrangements, b) converging, and c) diverging.

(a) (b)

Fig. 1.3 Chromakey studios: a) Kinovis capture studio at INRIA Rhône-Alpes b) Capture studio of the university of Surrey.

1.3.1 RECOVER3D studio layout and processing

The RECOVER3D studio is installed in the premises of the industrial partner XD Productions. It is a green chromakey studio of 100 square meters, which is 4.5 meter high. The results shown in this thesis were produced by 24 full HD cameras (1920x1080 pixels), at 25 frames per second, but the system is designed to be scalable up to 40 cameras, recording 60 frames per second. The combination of visual hull (VH) and multi-baseline stereovision requires views from a wide variety of angles for visual hull extraction and from distinct but close points of view for stereo matching. The project thus relies on a studio composed of many synchronized and time-stamped cameras. As we mentioned previously, our acquisition system consists in converging cameras in order to build the VH called *monoscopic* cameras and several groups of multi-camera arrangements called *multiscopic units* and each dedicated to one multi-baseline stereovision reconstruction. Multiscopic units (see figures 1.5, 1.4) are laid with aligned and evenly distributed optical centers. We chose to group four cameras per multiscopic unit which seems, according to experience, a good compromise between robustness, relying on views redundancy, and computational efficiency [50]. The cameras are calibrated in geometry and colorimetry in a pre-shooting step. For each time stamp, every image is matted thanks to pre-calibrated chromakey (RGB space related to background in views) and resulting silhouettes are used to compute the VH. For each multiscopic unit, captured images are then rectified to match simplified epipolar geometry [43].

1.4 Contributions of this thesis

This thesis makes two major contributions. The first one is a novel scene-based framework for direct multi-view stereovision reconstruction. Our proposition aims at building a new *materiality map* on the disparity space to optimize it according to a relevant energy function and finally to use its optimized content for deciding where the reconstructed surfaces

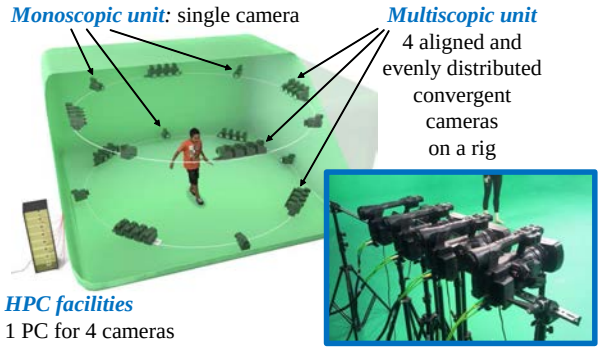


Fig. 1.4 Dedicated multiview studio.

Fig. 1.5 RECOVER3D studio at XD-production company with multiscopic and monoscopic units.

lie in the disparity space (DS). The second contribution is a novel framework for multi-view 3D reconstruction relying on both multi-baseline stereovision and visual hull. This method's inputs are a visual hull and several sets of multi-baseline views. For each such view set, a multi-baseline stereovision method yields a surface which is used to carve the visual hull. Carved visual hulls from different view sets are then fused iteratively to deliver the intended 3D model.

1.5 Layout of this thesis

This thesis is organized by the following chapters :

- Chapter 2 presents the state of art of 3D reconstruction methods. Firstly, we describe the geometry for one camera and multiple cameras. Different geometry constraints are exploited. In this chapter we explain in details two methods for the 3D reconstruction, multi-stereovision and shape from silhouettes. Secondly, we propose to classify the methods merging the silhouette-based and stereovision-based reconstruction into three groups: i) stereovision guided by visual hull methods, ii) collaborative methods applying simultaneously criteria borrowed from both visual hull and stereovision, iii) separate application of both methods with further merging of their results.

- Chapter 3 describes a novel method for multi-baseline stereovision. We introduce the concept of materiality map to represent the probability of the 3D points (called target points) to belong to the reconstructed surface. The method consists of different steps which can be summarized as follows:
  - scene sampling to determine the digital domain where the objects can be reconstructed.
  - identification the similarity, confidence, and visibility for each target point.
  - definition of the cost function and application of gradient descent as optimization method.
  - binarization of the materiality values of each target points in order to extract the reconstructed surface.
- Chapter 4 proposes an innovative framework of 3D reconstruction from fusion of silhouette-based reconstruction and multiple results of our chapter 3 multi-baseline stereovision approach. It first enhances multi-baseline stereovision process thanks to visual hull data. Then it merges different volumes resulting from multiple multiscopic units. The goal of this method is then to produce a single 3D model representing the 3D pose of the object to reconstructed.
- Chapter 5 presents a detailed summary and conclusion of the thesis, and discusses opened problems to tackle in future work.

1.6 Résumé : Introduction et contexte

Dans un contexte de fragmentation de l'audience TV due á la multiplication du nombre de chaînes et á la concurrence de nouveaux modes de consommation (VOD, Internet, ...), les diffuseurs et les producteurs sont plus que jamais à la recherche de contenu différencié de qualité, produit dans des conditions économiques optimales. Cette thèse présente la partie de la reconstruction 3D d'un projet plus vaste appelé RECOVER3D (Real-time Environnement for COmputational Video Editing and Rendering in 3D) qui développe, pour les industries du cinéma et de la télévision, un système complet allant de la capture de performances d'acteurs ou d'autre objets en médias 4D de haute qualité á leurs utilisations multiples et variées (duplication, édition spatiale, géométrique, temporelle, texturale, ré-éclairement, ...) en régie virtuelle. Le système d'acquisition proposé pour la reconstruction de la scène, repose sur un studio multi-caméras spécifique. Le studio chromakey développé comprend jusqu'à 40 caméras HD synchronisées réparties, autour et au dessus de l'espace scénique désiré, isolément (unités monoscopiques), ou par bloc de 4 (unités multiscopiques). Les caméras d'une unité multiscopiques sont disposées avec des centres optiques alignés et équidistants pour permettre, par rectification, de délivrer des vidéos 4-vues en géométrie épipolaire simplifiée. L'ensemble de ces unités est utilisé dans un premier temps, via une méthode basée silhouette, á reconstruire l'enveloppe visuelle de la scène. Dans cette thèse, nous proposons tout d'abord une nouvelle méthode de stéréovision multi-vue alignée appliquée sur les images acquises par chaque unité multiscopique. Puis nous décrivons comment améliorer sa robustesse et son efficacité en intégrant des informations issues des silhouettes. Enfin, dans l'objectif de générer un modèle 3D de précision de la scène, nous présentons une hybridation de notre méthode multiscopique et notre pipeline de reconstruction 3D multi-vues, intégrant les résultats issus de l'enveloppe visuelle et la stéréovision multi-vue.

Chapter 2

Multiview 3D reconstruction: a review

In this chapter, we present the concept of 3D scene modeling from multiple images and some of its applications. In section 2.1, we present the geometrical models and tools implied in multi-view computer vision. Starting with the monocular pinhole camera model, we further focus on binocular and multiocular geometries. In section 2.2, we describe existing 3D reconstruction techniques that use multiple views. We introduce the concept of binocular stereovision and describe the multi-view stereovision and shape-from-silhouettes methods. Since we work within the RECOVER3D project which aims at hybridizing shape from silhouettes and multi-view stereovision, we propose in section 2.3 to classify such hybrid techniques into three major groups: i) stereovision guided by visual hull methods, ii) collaborative methods applying simultaneously criteria borrowed from both techniques, iii) separate application of both methods with further merging of their results.

2.1 Multiple view geometry: definitions and notations

Before discussing the multiview 3D reconstruction, it is important to know how the images are obtained. In this section, we describe the single camera shooting geometry and geometric constraints existing between multiple views of a same scene with no prior constraint on layout of cameras.

2.1.1 Monocular geometry

The process of evaluating the relationship between the scene and captured image coordinates is called camera calibration. It is a necessary step for many computer vision applications especially for the 3D reconstruction methods. It requires some parametrical model of the coordinate transformation process which relies on the projection model used for the camera. The most usual perspective camera model corresponds to the pinhole camera

model expressing a perfect projective camera with infinitesimal and instantaneous aperture. Such an ideal model yields a simple mathematical relationship between the coordinates of a 3D point in a reference frame tied to the camera and its projection onto the planar image domain indexed in pixel coordinates. The more global relationship between the 3D scene coordinates and their corresponding image coordinates is usually expressed using two groups of parameters:

- Extrinsic camera parameters: they define the relative position and orientation of the scene frame in the camera frame. They describe the positioning of the 3D scene in the camera frame (see section 2.1.1.1).
- Intrinsic camera parameters : they relate to internal geometric and optical characteristics of the camera. They are linked to the projection step and given by the pinhole model (see section 2.1.1.2).

Thank to the definition of intrinsic (section 2.1.1.1) and extrinsic (section 2.1.1.2) camera parameters, we will identify the projection matrix in section 2.1.1.3.

2.1.1.1 Intrinsic camera parameters

The usual pinhole camera model expresses a perfect perspective camera (see figure 2.1a) in which a visible 3D point **M** is projected onto the 2D point **m** of the image plane via a single optical ray passing through the optical center **C**. The intrinsic camera reference frame of the pinhole camera is usually positionned on the optical center (projection center) with $\mathbf{Z}^c$ axis orthogonal to the image plane and oriented towards the scene and $\mathbf{X}^c$ axis parallel to sensor rows. The sensor plane is set at focal distance f from the optical center. The perfect perspective projection of point **M** with coordinates $\mathbf{M}_c = (x_c, y_c, z_c)^t$ in camera frame onto the image plane point $(x_p, y_p)^t$ is expressed by:

$$x_p = f\,\frac{x_c}{z_c} \quad y_p = f\,\frac{y_c}{z_c} \iff \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}. \tag{2.1}$$

It should be noted that the symbol $\sim$ refers to the equality of vectors with a nonzero scaling factor (this is due to the use of homogeneous coordinates).
The conversion of metric coordinates $(x_p, y_p)^t$ in sensor plane to pixel coordinates $\mathbf{m}(u, v)^t$ in the image depends on sensor geometry and position (see figure 2.1.b). These sensor parameters encompass:

- its horizontal $p_h = w/nc$ and vertical $p_v = h/nr$ pitches expressing the distances between adjacent columns and rows, or their inverses $k_u = p_h^{-1}$ and $k_v = p_v^{-1}$;

- a column bias γ taking into account the fact that row and column may not be orthogonal due to some manufacturing skew error;
- the pixel coordinates $(u_0, v_0)^t$ of the intersection **c** of the optical axis with the image plane.

These parameters yield the needed conversion:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u & \gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}. \tag{2.2}$$

Using the equations 2.1 and 2.2, pixel coordinates are then obtained from metric coordinates in the camera frame by:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u & \gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}. \tag{2.3}$$

The equation 2.3 can be re-written as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} \alpha_u & s & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}, \tag{2.4}$$

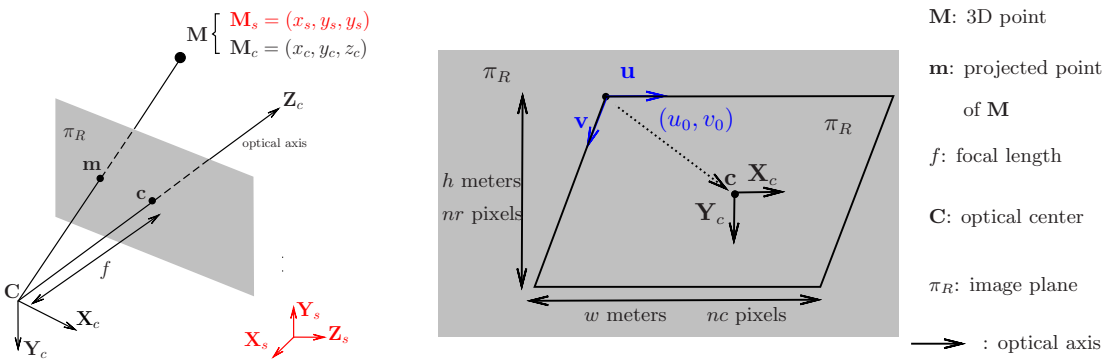
$$\text{with:} \quad \alpha_u = k_u\, f, \quad \alpha_v = k_v\, f, \quad s = \gamma\, f.$$

Introducing the 3x3 identity matrix $\mathbf{I}_3$, equation 2.4 becomes:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}. \tag{2.5}$$

In short hand notation, we write equation 2.5 as:

$$\begin{pmatrix} \mathbf{m} \\ 1 \end{pmatrix} \sim \mathbf{K} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{M}_c \\ 1 \end{pmatrix}, \tag{2.6}$$



(a) (b)

Fig. 2.1 Pinhole camera: a) Perspective projection, b) Transformation from metric coordinates in sensor plane to pixel coordinates.

where $\mathbf{m}$ represents the pixel coordinates, $\mathbf{K}$ is the perspective projection matrix, and $\mathbf{M}_c$ is the vector of coordinates of a point measured in the camera frame. The parameters α_u, α_v, s, u_0, and v_0 do not depend on the orientation and the position of the camera in the scene. Therefore, they are called intrinsic parameters.

2.1.1.2 Extrinsic camera parameters

The relationship between the scene and camera frames , respectively $(\mathbf{C}^s, \mathbf{X}^s, \mathbf{Y}^s, \mathbf{Z}^s)$ and $(\mathbf{C}^c, \mathbf{X}^c, \mathbf{Y}^c, \mathbf{Z}^c)$, both supposed orthonormal and direct, is defined as a rigid transformation. This is described by a translation, which represents the displacement between origins of the scene and camera frames, and by a rotation, which defines the scene frame orientation with respect to the camera frame. With $\mathbf{V}_s$ and $\mathbf{V}_c$ expressing the coordinates of vector $\mathbf{V}$ respectively in scene and camera frames, $\mathbf{C}_s^c$ referring to optical center translation from camera to scene frame written in camera frame, and $(\mathbf{X}_c^s \ \ \mathbf{Y}_c^s \ \ \mathbf{Z}_c^s)$ representing the rotation of scene frame into camera frame, we obtain:

$$\begin{aligned}(\mathbf{M}_c - \underbrace{\mathbf{C}_c^c}_{\mathbf{0}}) &= \underbrace{\begin{pmatrix} \mathbf{X}_c^s & \mathbf{Y}_c^s & \mathbf{Z}_c^s \end{pmatrix}}_{\mathbf{R}} (\mathbf{M}_s - \underbrace{\mathbf{C}_s^c}_{\mathbf{t}}), \\ \mathbf{M}_c &= \mathbf{R}\,\mathbf{M}_s \underbrace{-\mathbf{R}\,\mathbf{t}}_{\mathbf{T} \equiv \mathbf{C}_c^s}.\end{aligned} \tag{2.7}$$

Finally, from equation 2.7, we obtained the scene/camera transformation:

$$\begin{pmatrix} \mathbf{M}_c \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ & 1 \end{pmatrix} \begin{pmatrix} \mathbf{M}_s \\ 1 \end{pmatrix}. \tag{2.8}$$

2.1.1.3 Projection matrix

Combining equations 2.6 and 2.8, we get the transformation from scene coordinates to pixel coordinates (see equation 2.9).

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ & 1 \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix}. \tag{2.9}$$

This transformation relies on a *projection matrix* **P** which includes intrinsic and extrinsic camera parameters defined as follows:

$$\mathbf{P} \sim \mathbf{K}(\mathbf{I}_3 \;\; 0) \begin{pmatrix} \mathbf{R} & -\mathbf{Rt} \\ & 1 \end{pmatrix}. \tag{2.10}$$

P can be re-written:

$$\begin{aligned} \mathbf{P} &\sim (\mathbf{KR} \quad -\mathbf{KRt}) \\ \mathbf{P} &\sim \mathbf{KR}(\mathbf{I}_3 \quad -\mathbf{t}). \end{aligned} \tag{2.11}$$

Finaly the equation 2.9 can be written in short notation as follows:

$$\begin{pmatrix} \mathbf{m} \\ 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} \mathbf{M}_s \\ 1 \end{pmatrix}. \tag{2.12}$$

Using the projection matrix **P** described in 2.10 yields an irreversible loss of local depth (in camera frame) information. One possibility to overcome this problem is, as we will show in the next chapter, to add some depth related value to pixel coordinates in order to get a square and invertible *expanded projection matrix*.

2.1.1.4 Calibration process

In order to achieve 3D reconstruction, one has to evaluate the projection matrix parameters to quantify the projection of 3D points to pixels. The calibration process of a single camera thus estimates its extrinsic $(\mathbf{R}, \mathbf{T})$ and intrinsic matrices (**K**) from actual views. The rotation matrix **R**, although consisting of 9 elements, has only 3 degrees of freedom as it has to fulfill 6 constraints linked to its orthonormality. The translation vector **T** obviously has 3 parameters. Therefore, this leads in total to 11 unknown parameters: 6 extrinsic and 5 intrinsic.

Fig. 2.2 Multi-planar chessboard 3D reference object-based calibration, source: [85].

In order to compute these unknowns, one has to build equations from relations between some values measured in image space (coordinates, distances,...) and corresponding values known in 3D scene space. The parametric relation between these values rely on the projection matrix **P** which has 12 unknown parameters. The overall process of computing $(\mathbf{R},\mathbf{T})$ and **K** usually uses two steps: (i) computing the projection matrix **P** which best satisfies the equations provided by the relations between the chosen sets of corresponding values in image and scene spaces and (ii) extracting **K**, **R**, and **T** from **P** according to equation 2.11 and properties of **K** (upper triangular), and **R** (orthonormal).

2.1.1.4.1 First step: computing projection matrix

We need $ne \geq 12$ scalar equations to obtain a solution for the projection matrix **P**. We propose to classify calibration methods into two major groups according to their choice of corresponding values yielding the necessary *ne* equations:

- 3D reference object-based calibration: a reference object (see figure 2.2) with distinguishable and calibrated 3D feature points is placed in a known pose in scene space (for instance two or three orthogonal planes laid as material representation of the chosen scene frame). The coordinates $\mathbf{M}_s^i$ of each 3D feature point in scene space and the corresponding pixels are thus known and measurable. After finding for each point of the set of 3D feature points, its coordinates in scene frame $\mathbf{M}_s^i$ and its corresponding pixel in image coordinate system $\mathbf{m}^i$, we can write the following equation for all *i* and then compute **P**:

$$\begin{pmatrix} \mathbf{m}^i \\ 1 \end{pmatrix} \sim \overbrace{(\underbrace{\mathbf{KR}}_{\mathbf{A}} \quad \underbrace{\mathbf{KT}}_{\mathbf{B}})}^{\mathbf{P}} \begin{pmatrix} \mathbf{M}_s^i \\ 1 \end{pmatrix}; \tag{2.13}$$

- rigid pattern-based calibration: a rigid planar pattern (chessboard) shown at a few different orientations [10] or a rigid set of collinear points moved around a fixed point

provide unknown but related 3D points [10]. Thus, in order to find **P**, for each frame with such a pattern in unknown pose p (orientation $\mathbf{R}_p$ and translation $\mathbf{T}_p$), we define the following equation for each feature point indexed by i in the pattern. Using the coordinates $\mathbf{M}_o^i$ of the feature point in the pattern frame and those of its corresponding pixel $\mathbf{m}^{p,i}$ in pose p gives the following equation:

$$\begin{pmatrix} \mathbf{m}^{p,i} \\ 1 \end{pmatrix} \sim \mathbf{P} \begin{pmatrix} \mathbf{R}^p & \mathbf{T}^p \end{pmatrix} \begin{pmatrix} \mathbf{M}_o^i \\ 1 \end{pmatrix}. \tag{2.14}$$

2.1.1.4.2 Second step: decomposing projection matrix

The left 3x3 sub-matrix **A** (defined in 2.13) of projection matrix **P** is a product of upper-triangular matrix **K** and orthogonal matrix **R**. Any non-singular square matrix **G** can be decomposed into the product of an upper-triangular matrix R and an orthogonal matrix Q using the RQ factorization [56]. When this factorization has yielded **K** and **R**, **T** is easily computed as $\mathbf{T} = \mathbf{K}^{-1}\mathbf{B}$.

2.1.2 Binocular geometry

The relative geometry of two different perspective views of the same 3D scene is called *epipolar geometry*. The two perspective views may be acquired simultaneously, for example in a stereo rig, or sequentially if the scene is static, for example by a moving camera. In this section, we expose and describe the geometrical relationship existing between corresponding pixels in two images of the same scene. This relationship depends only on the intrinsic parameters of the two cameras and their relative translation and rotation which may be obtained from their extrinsic parameters. It expresses that both points are projections of a single visible 3D point of the scene and, thus, that their optical rays must intersect each other on this 3D point. We introduce the concept of epipolar geometry in section 2.1.2.1 and simplified epipolar geometry in section 2.1.2.2.

2.1.2.1 Epipolar Geometry

2.1.2.1.1 Concept

If a point of the scene is seen by two different cameras, then consequently a geometrical relationship is defined between the 3D point and its projections in the images. The relationship introduces a constraint between matching points in the left (l) and right (r) images called epipolar constraint. It represents the necessary coplanarity of the 3D point **M**, its projections onto both images $\mathbf{m}_l$ and $\mathbf{m}_r$ and the optical centers $\mathbf{C}_l$ and $\mathbf{C}_r$ of both cameras. The epipolar constraint reduces the search space dimension for any left pixel $\mathbf{m}_l$ to the 1D intersection of the plane $(\mathbf{C}_l, \mathbf{C}_r, \mathbf{m}_l)$ with the other right image plane and therefore the time spent searching for matching points between two images.

Let's consider for example the system illustrated in figure 2.3 composed of two cameras. Points $\mathbf{e}_l$ and $\mathbf{e}_r$ represent the *epipolar centers* (or *epipoles*) defined as the intersections of the straight line passed through ($\mathbf{C}_l$, $\mathbf{C}_r$) with each image plane. The epipole $\mathbf{e}_r$ (or $\mathbf{e}_l$ respectively) refers to the position in its image plane where the projection center $\mathbf{C}_l$ (or $\mathbf{C}_r$) of the other image is observed. Considering $\mathbf{m}_l$ a pixel of the left image, the point $\mathbf{v}_r^{\mathbf{m}_l}$ is the projection on right image plane of the vanishing point $\mathbf{V}^{\mathbf{m}_l}$ of the ray of $\mathbf{m}_l$. Moreover, the 3D point $\mathbf{M}$ projected on $\mathbf{m}_l$, is necessarily located on the ray passing through $\mathbf{C}_l$ and $\mathbf{m}_l$. Its projection $\mathbf{m}_r$ on the right image is thus mandatory on the plane $(\mathbf{C}_l, \mathbf{C}_r, \mathbf{m}_l)$ called *epipolar plane* of $\mathbf{m}_l$.

We can therefore restrict the search for matching points of any left pixel $\mathbf{m}_l$ within the intersection of its epipolar plane with the right image plane. Moreover, we notice that the 3D point projected on $\mathbf{m}_l$ lies on half line $[\mathbf{C}_l, \mathbf{V}^{\mathbf{m}_l})$. The associated right rays sweep the area of the epipolar plane limited by half lines $[\mathbf{C}_r, \mathbf{C}_l)$ and $[\mathbf{C}_r, \mathbf{V}^{\mathbf{m}_l})$. Corresponding right pixels lie on the intersection of this plane with the right image. This defines the *epipolar segment* $[\mathbf{e}_r, \mathbf{v}_r^{\mathbf{m}_l}]$ of $\mathbf{m}_l$. The line extending the epipolar segment of $\mathbf{m}_l$ is called the *epipolar line* of $\mathbf{m}_l$.

If two left pixels $\mathbf{m}_l$ and $\mathbf{m}'_l$ are aligned with the epipole $\mathbf{e}_l$, they define the same epipolar plane and thus share the same right epipolar line. Their epipolar segments differ only according to their right ends $\mathbf{v}_r^{\mathbf{m}_l}$ and $\mathbf{v}_r^{\mathbf{m}'_l}$. One may thus think of epipolar geometry as a set of epipolar planes rotating around the baseline $(\mathbf{C}_l, \mathbf{C}_r)$ and defining couples of associated epipolar lines in both images. For every pixel lying on one of these epipolar lines, one has to search its homologue on the associated line. Therefore, it is important to define the epipolar line corresponding to a given pixel in the other image. One needs a dedicated and practical tool to ease the identification of this search space. This tool is usually provided by the *fundamental matrix* as we will show in the next section.

However, when the image planes are parallel to the baseline, epipoles are at infinity in baseline direction and epipolar lines are then also both parallel to the baseline. This configuration facilitates the matching process and is introduced by the simplified epipolar geometry explained in details in the section 2.1.2.2.

2.1.2.1.2 Fundamental matrix

As previously mentioned, the epipolar geometry expresses a mandatory geometrical relationship between corresponding pixels in separate views. The epipolar constraint reduces the research space for homologous pixels to 1D epipolar segments or lines where these homologous must be located. Therefore, it is important to identify the specific epipolar line or segment corresponding to a given pixel of an image. In this section, we describe the mathematical construction of an epipolar line equation from the given pixel coordinates $\mathbf{m}_l$

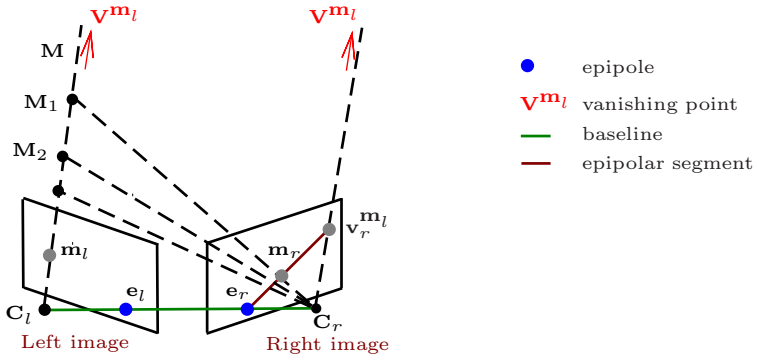


Fig. 2.3 Schematic representation of the epipolar geometry: corresponding point $\mathbf{m}_r$ of pixel $\mathbf{m}_l$ has to lie on segment $[\mathbf{e}_r, \mathbf{v}_r^{\mathbf{m}_l}]$, intersection of its image plane with half planar stripe $(\mathbf{C}_l, \mathbf{C}_r, \mathbf{V}^{\mathbf{m}_l})$.

choosing for convenience the left image as reference. This equation will be expressed by forcing to be zero the dot product between homogeneous right pixel coordinates $(\mathbf{m}_r^t \ \ 1)^t$ and a 3D vector $\mathbf{l}_r(\mathbf{m}_l)$ containing the equation coefficients of $\mathbf{m}_l$ epipolar line. We will then note that this coefficient vector is linearly expressed from $(\mathbf{m}_l^t \ \ 1)^t$ thanks to a matrix $\mathbf{F}$ which depends only on extrinsic and intrinsic camera parameters and thus can be precomputed only once for all pixels. This implies that homologous pixels have to verify a bi-linear equation built from $\mathbf{F}$. This shows that $\mathbf{F}$ contains the whole epipolar geometry. This matrix $\mathbf{F}$ is called the *fundamental matrix*.

Let us consider that the given left pixel $\mathbf{m}_l$ is the projection of a 3D point $\mathbf{M}$, which is also projected onto the right image the pixel $\mathbf{m}_r$. Based on the equation 2.11, these projections are expressed as follows:

$$\begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix} \sim \mathbf{P}_l \begin{pmatrix} \mathbf{M} \\ 1 \end{pmatrix} \quad \text{with:} \quad \mathbf{P}_l = \mathbf{K}_l \mathbf{R}_l (\mathbf{I}_3 \ -\mathbf{t}_l), \tag{2.15}$$

$$\begin{pmatrix} \mathbf{m}_r \\ 1 \end{pmatrix} \sim \mathbf{P}_r \begin{pmatrix} \mathbf{M} \\ 1 \end{pmatrix} \quad \text{with:} \quad \mathbf{P}_r = \mathbf{K}_r \mathbf{R}_r (\mathbf{I}_3 \ -\mathbf{t}_r). \tag{2.16}$$

The epipolar line equation is derived from the expression that the unknown 2D point $\mathbf{m}_r = (u_r, v_r)^t$ is aligned with two known points $\mathbf{e}_r$ and $\mathbf{v}_r^{\mathbf{m}_l}$. This is achieved by expressing the zero value of the determinant of the matrix composed of those three points in homogenous coordinates as follows:

$$Det\left(\begin{pmatrix} \mathbf{e}_r \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_r^{\mathbf{m}_l} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{m}_r \\ 1 \end{pmatrix}\right) = 0. \tag{2.17}$$

The first known point expressed as $(\mathbf{e}_r^t \;\; 1)^t$ can be easily defined since this right epipole represents the projection of the left optical center C_l into the right image:

$$\begin{pmatrix} \mathbf{e}_r \\ 1 \end{pmatrix} \sim \mathbf{P}_r \underbrace{\begin{pmatrix} \mathbf{t}_l \\ 1 \end{pmatrix}}_{C_l} = \mathbf{K}_r \mathbf{R}_r (\mathbf{t}_l - \mathbf{t}_r). \tag{2.18}$$

The second known point expressed as $(\mathbf{v}_r^{\mathbf{m}_l t} \;\; 1)^t$ is defined as the projection in the right image of the vanishing point $\mathbf{V}^{\mathbf{m}_l}$ on $\mathbf{m}_l$ ray, expressed as follows:

$$\begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix} \sim \mathbf{P}_l \begin{pmatrix} \mathbf{V}^{\mathbf{m}_l} \\ 0 \end{pmatrix} = \mathbf{K}_l \mathbf{R}_l (\mathbf{I}_3 \;\; -\mathbf{t}_l) \begin{pmatrix} \mathbf{V}^{\mathbf{m}_l} \\ 0 \end{pmatrix} = \mathbf{K}_l \mathbf{R}_l \mathbf{V}^{\mathbf{m}_l}, \tag{2.19}$$

$$\mathbf{V}^{\mathbf{m}_l} \sim \mathbf{R}_l^t \mathbf{K}_l^{-1} \begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix}. \tag{2.20}$$

Using 2.20, the point expressed as $(\mathbf{v}_r^{\mathbf{m}_l t} \;\; 1)^t$ is written as follows:

$$\begin{pmatrix} \mathbf{v}_r^{\mathbf{m}_l} \\ 1 \end{pmatrix} \sim \mathbf{P}_r \begin{pmatrix} \mathbf{V}^{\mathbf{m}_l} \\ 0 \end{pmatrix} = \mathbf{K}_r \mathbf{R}_r \mathbf{V}^{\mathbf{m}_l} = \mathbf{K}_r \mathbf{R}_r \mathbf{R}_l^t \mathbf{K}_l^{-1} \begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix}. \tag{2.21}$$

After defining two known points using 2.18 and 2.21, the determinant 2.17 which describes the epipolar line equation is written using the triple product:

$$(\mathbf{m}_r^t \;\; 1) \left(\begin{pmatrix} \mathbf{e}_r \\ 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{v}_r^{\mathbf{m}_l} \\ 1 \end{pmatrix} \right) = 0, \tag{2.22}$$

$$(\mathbf{m}_r^t \;\; 1) \left((\mathbf{K}_r \mathbf{R}_r (\mathbf{t}_l - \mathbf{t}_r)) \times (\mathbf{K}_r \mathbf{R}_r \mathbf{R}_l^t \mathbf{K}_l^{-1} \begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix}) \right) = 0. \tag{2.23}$$

Using the rule $(\mathbf{Aa}) \times (\mathbf{Ab}) \sim A^{-t}(\mathbf{a} \times \mathbf{b})$, we can write the equation which defines the epipolar line:

$$(\mathbf{m}_r^t \;\; 1) \underbrace{(\mathbf{K}_r \mathbf{R}_r)^{-t} \left((\mathbf{t}_l - \mathbf{t}_r) \times (\mathbf{R}_l^t \mathbf{K}_l^{-1} \begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix}) \right)}_{\mathbf{l}_r(\mathbf{m}_l) \equiv (a \;\; b \;\; c)^t \in \mathbb{R}^3} = 0, \tag{2.24}$$

$$(\mathbf{m}_r^t \;\; 1) \; \mathbf{l}_r(\mathbf{m}_l) = 0 \quad \Leftrightarrow \quad a u_r + b v_r + c = 0. \tag{2.25}$$

Using the notation of 3D cross product $\mathbf{a} \times \mathbf{b}$ with anti-symetrical matrix $[\mathbf{a}]_\times$ as $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b}$, we can express the vector of the line equation coefficients $\mathbf{l}_r(\mathbf{m}_l)$ linearily according to $(\mathbf{m}_l^t \ \ 1)^t$ as follows:

$$\mathbf{l}_r(\mathbf{m}_l) \equiv \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \underbrace{(\mathbf{K}_r\mathbf{R}_r)^{-t}[\mathbf{t}_l - \mathbf{t}_r]_\times \mathbf{R}_l^t \mathbf{K}_l^{-1}}_{\mathbf{F}} \begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix}. \tag{2.26}$$

In the equation 2.26, one can note that the matrix $\mathbf{F}$ only depends on the extrinsic and intrinsic parameters of both cameras. As such, it may be precomputed once for all left pixels $\mathbf{m}_l$. As we mentioned previously, the point $\mathbf{m}_r$ corresponding to $\mathbf{m}_l$ should be located on the line of coefficients $\mathbf{l}_r$. Therefore, the dot product between $\mathbf{m}_r$ and $\mathbf{l}_r$ should be zero $(\mathbf{m}_r^t \ \ 1).\mathbf{l}_r(\mathbf{m}_l) = 0$. The epipolar constraint mentioned in the section 2.1.2.1 is defined by the equation 2.25. Transposing this equation provides a symmetrical equation expressing that a left pixel $\mathbf{m}_l$ lies on the epipolar line of a right pixel $\mathbf{m}_r$, identified by its coefficients $\mathbf{l}_l(\mathbf{m}_r)$ (see equation 2.27). Thus the precomputed fundamental matrix $\mathbf{F}$ contains the whole epipolar geometry as it builds epipolar line equations for any pixel of both images as follows:

$$(\mathbf{m}_r^t \ \ 1)\underbrace{\mathbf{F}\begin{pmatrix} \mathbf{m}_l \\ 1 \end{pmatrix}}_{\mathbf{l}_r(\mathbf{m}_l)} = 0 \quad \Leftrightarrow \quad (\mathbf{m}_l^t \ \ 1)\underbrace{\mathbf{F}^t\begin{pmatrix} \mathbf{m}_r \\ 1 \end{pmatrix}}_{\mathbf{l}_l(\mathbf{m}_r)} = 0. \tag{2.27}$$

In the case of unavailable camera parameters, the fundamental matrix $\mathbf{F}$ can be computed using the equation 2.27 by identifying a set of corresponding points between different images using feature-based or intensity-based methods. One of important computer vision applications which need the fundamental matrix is structure-from-motion (SfM). SfM estimates three-dimensional structure from image sequences using the fundamental matrix as geometry constraint.

2.1.2.2 Simplified Epipolar Geometry

2.1.2.2.1 Concept

If image planes are both parallel to the baseline, the epipolar planes then intersect the images at epipolar lines which are also parallel to the baseline. Moreover, if the image rows are parallel to the baseline, the epipolar lines are the image rows. If, furthermore, the image planes are identical (parallel one to the other with the same focal length) and sensors have the same vertical pitch and centering, epipolar couples are composed of rows of same rank of both images in binocular geometry. The homologous point search in the second image is thus limited to a horizontal line of the second image located at the same ordinate.

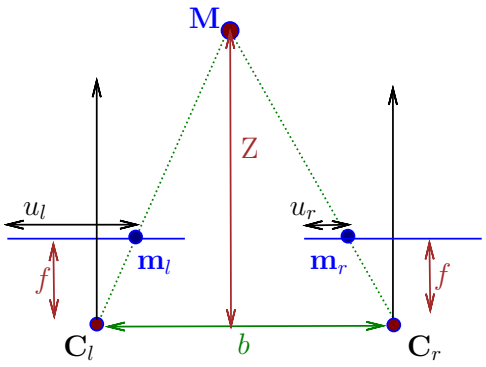


Fig. 2.4 Relationship between disparity $\delta = u_l - u_r$ and depth Z.

In this configuration, the epipolar geometry is called "simplified". No fundamental matrix is required to yield the epipolar equation ($v_r = v_l$) while the epipolar segment is easily identified in this row: $u_r \in [0, u_l]$ as $\mathbf{v}_r^{\mathbf{m}_l} = \mathbf{m}_l$. Moreover, a practical tool, the disparity, is usually used to index possible matches as we will show in the next section.

2.1.2.2.2 Disparity

In simplified epipolar geometry, the homologue $\mathbf{m}_r = (u_r, v_r)$ on the right image of a left pixel $\mathbf{m}_l = (u_l, v_l)$ is identified by its coordinates u_r and $v_r = v_l$ in image frame. It is then common to keep the *disparity* $\delta \equiv u_l - u_r$ as a relative identifier since this simple matching result directly relates to the depth Z of the projected 3D point $\mathbf{M}$ (see below) which thus greatly simplifies the triangulation step. Given a 3D point $\mathbf{M}$ and its projections $\mathbf{m}_l$ and $\mathbf{m}_r$ onto image planes (see figure 2.4), $(\mathbf{M}, \mathbf{C}_l, \mathbf{C}_r)$ and $(\mathbf{M}, \mathbf{m}_l, \mathbf{m}_r)$ are similar triangles. The depth Z is then defined as follows:

$$\frac{b}{Z} = \frac{b-\delta}{Z-f} \quad \Leftrightarrow \quad Z = \frac{f\,b}{\delta} \quad \Leftrightarrow \quad Z\,\delta = f\,b. \tag{2.28}$$

Thanks to this disparity-depth relationship, some authors express their stereovision results as disparity maps defined as gray scale images where the intensity of each pixel is its disparity δ related to the depth of the associated 3D point thanks to equation 2.28. Given a disparity map D_l computed from two images, the pixel intensity $D_l(u_l, v_l)$ of this map can be described by:

$$D_l(u_l, v_l) = \delta = u_l - u_r.$$

The disparity map defined above is computed considering the left image as reference image. Using the right image as reference, we should fulfill the coherence relationship between the two disparity maps which expresses that both homologues should have same disparity as

(a) (b)

Fig. 2.5 Evolution of epipolar line (blue line): a) in source images and b) after rectification process, in rectified images.

follows:

$$D_l(u_l, v_l) = D_r(u_l - D_l(u_l, v_l), v_l). \tag{2.29}$$

One should note that depth reconstruction is much easier thank to equation 2.28 in simplified geometry than in generic epipolar geometry which requires a triangulation process. Indeed the intersection of two optical rays implies some optimality issues tackled by multiple proposals such as Mid-point method or Direct Linear Transformation (DLT) [1]. This is the main reason why numerous authors propose to switch from generic actual shooting geometry to virtual simplified geometry thanks to a rectification process.

2.1.2.3 Rectification

In practice, it is difficult to have actual simplified camera geometry and the rectification helps providing views in simplified geometry. It is possible through rectification [24] to transfer from any geometry (figure 2.5.a) to simplified geometry (Figure 2.5.b) in order to simplify the problems of matching points between images and triangulating their associated 3D points. The rectification approach consists of projecting images (I_l, I_r) (see figure 2.6) from their optical center respectively ($\mathbf{C}_l$, $\mathbf{C}_r$) on a same plane $\prod_C$ parallel to the line of the optical centers on virtual sensors whose rows are parallel to the baseline, of same pitch and vertical alignment [24].

2.1.3 Multiocular geometry

2.1.3.1 Multiple epipolar geometries

2.1.3.1.1 Concept

Given $n > 2$ images and homologous pixels $\mathbf{m}$ and $\mathbf{m}'$ into images i and j provides a 3D point $\mathbf{M}$, the projection $\mathbf{m}''$ of $\mathbf{M}$ onto any other image plane k then corresponds to the matching pixels $\mathbf{m} \leftrightarrow \mathbf{m}' \leftrightarrow \mathbf{m}''$. This procedure only requires projection information to find $\mathbf{m}''$. An alternative method which expresses that $\mathbf{m}''$ lies on epipolar lines of $\mathbf{m}$ and $\mathbf{m}'$ in image k, is to apply the multifocal tensor (defined in the next section) to transfer the point directly without an explicit 3D reconstruction.

However, if the centers of projection are aligned, these epipolar lines are then identical and multifocal tensor is unpractical. In such case, multi-simplified epipolar geometry proposes

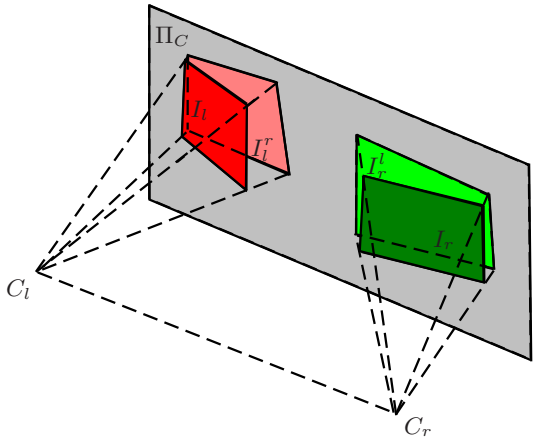


Fig. 2.6 The rectification process.

a convenient replacing tool. This section will briefly review some aspects of the multifocal tensor and multi-simplified epipolar geometry.

2.1.3.1.2 Multifocal tensor

As mentioned in the section 2.1.2.1.2, the fundamental matrix represents the relationship between matching points of two images of the same scene. Similar to the fundamental matrix for two images, Faugeras et al. [21] [20] introduce trifocal tensor and quadfocal tensor for three and four views respectively. The quadrilinear relations [21] are built as linear combinations of the bi-linear ones (expressed by fundamental matrices) and the trilinear ones (expressed by trifocal tensors), and any higher multilinear relation can be obtained from the bi-linear, trilinear and quadrilinear ones. This section will briefly review some aspects of the trifocal tensor.

Considering three views, it is possible to group them in pairs to get the two view relationships introduced in the section 2.1.2. Let us suppose, the 2D points $\mathbf{m}_1$ and $\mathbf{m}_2$ are matching pixels within first and second image. Using the fundamental matrix equation 2.26, we can compute the coefficients $\mathbf{l}_{13}(\mathbf{m}_1)$ and $\mathbf{l}_{23}(\mathbf{m}_2)$ of the epipolar lines in a third image corresponding to $\mathbf{m}_1$ and $\mathbf{m}_2$ respectively. The point in the third image corresponding to both pixels may be determined by the intersection of their epipolar lines (see figure 2.7). We note that finding the matching point in the third image fails if the two lines are identical ($\mathbf{l}_{13}(\mathbf{m}_1) \sim \mathbf{l}_{23}(\mathbf{m}_2)$). This case can occur if one of the pixels $\mathbf{m}_1$, $\mathbf{m}_2$ or $\mathbf{m}_3$ is coplanar with the three projection centers (called the trifocal plane) in which case the three homologous pixels lie in this trifocal plane. One should note that when the cameras are aligned every pixel of any image is coplanar with the three optical centers which implies that epipolar lines in other images are identical for every corresponding couple.

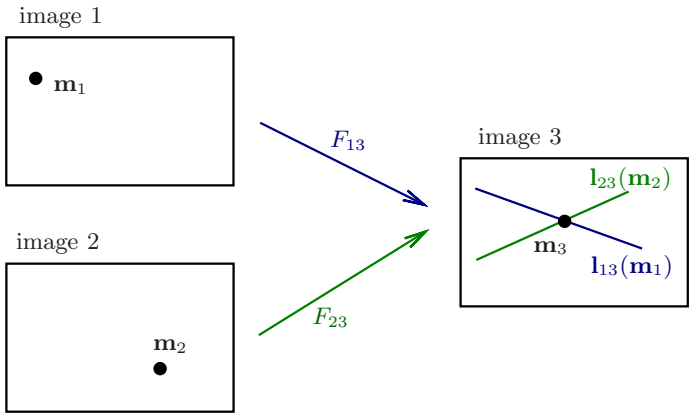


Fig. 2.7 Pixels matching through three images: the pixel $\mathbf{m}_3$ in the third image corresponding to the homologous $\mathbf{m}_1$ and $\mathbf{m}_2$ has to lie on the intersection $\mathbf{m}_1$ and $\mathbf{m}_2$ epipolar lines in $\mathbf{m}_3$ image.

Thanks to the trifocal tensor, the matching pixel in the third image can be directly computed even when matching pixels are coplanar with optical centers. We have worked with multiscopic units composed of multiple aligned cameras within the RECOVER3D project (see chapter 1). While the multiscopic units are closed to simplified epipolar geometry, we chose to rectify our views to benefit from this convenient geometry. The corresponding lines within multiple images are rows of same index and we do not need to identify the tensor.

2.1.3.2 Multi-simplified epipolar geometry

2.1.3.2.1 Concept

In binocular case, the simplified epipolar geometry is introduced to avoid using fundamental matrix and to simplify 3D reconstruction through disparity evaluation instead of rays triangulation. This geometry reduces the homologous pixel search to one dimension and facilitates the matching process. The same concept is applied for multiocular geometry in order to constrain matching search spaces in multiple images ($n-1>1$) without handling multifocal tensors. Being able to use simplified epipolar geometry on any image couple through $n>2$ images requires specific camera layout. Indeed, for any camera i, its sensor rows have to be parallel to the each baseline between cameras i and j with $j \neq i$. The centers of projection should thus be aligned on a *common baseline*. Furthermore, sensors need to be pairwise coplanar and parallel to the common baseline, which implies that they all lie on a single sensor plane parallel to this baseline. They also have to share same vertical pitches and alignments. Given two images i and j, thanks to the multi-simplified epipolar geometry, their corresponding epipolar lines are rows of same index $v_i = v_j = v$. Consequently, in

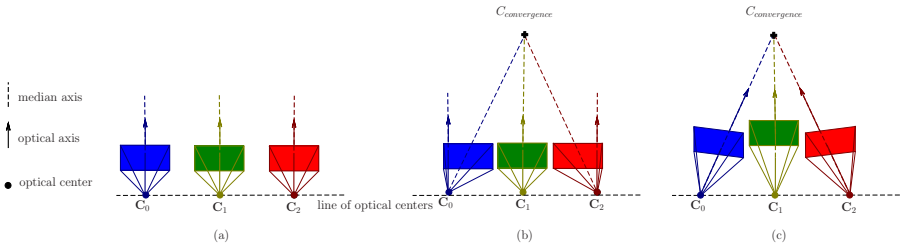


Fig. 2.8 Aligned geometry: a) parallel, b) decentered parallel geometry and c) aligned toed-in geometry.

such multi-simplified geometry, rows of same rank $v_i = v$ in each image $i \in \{0,\ldots,n-1\}$ define n-tuples of pairwise epipolar lines. Altogether, multi-simplified epipolar geometry is characterized by:

- aligned centers on a common baseline;
- a common sensor plane parallel to the baseline, this implies parallel sensor normals which are the camera optical axes;
- sensor rows parallel to the baseline;
- sensors of same vertical pitch and height with aligned top and bottom rows.

We propose to classify the camera layouts which verify such multi-simplified epipolar geometry into two groups, according to the camera frustums as follows:

- **Parallel geometry:** this usual, natural case implies frustums horizontally centered on the associated optical axis, $(u_0 = \frac{nc}{2})$ (see figure 2.8a).
- **Decentered parallel geometry:** In this more unusual setting, frustums are not anymore centered on their optical axes but laid with a median axis (passing through optical center and sensor's center) converging on a 3D point at finite distance called *point of convergence* (see figure 2.8b).

In the case of toed-in geometry with optical axes of the cameras converging at a same point in 3D space, if optical centers are aligned as illustrated in the figure (see figure 2.8c), multi-simplified geometry is achievable through a rectification process. This process adjusts the image planes into one single plane parallel to the baseline with conveniently laid virtual sensors. This will be shown in the next section. RECOVER3D uses this layout (see section 1.5b for justifications) and thus requires a rectification step.

2.1.3.2.2 Disparity

In binocular simplified geometry, we exposed in section 2.1.2.2.2 a convenient tool, the disparity δ of a pixel $(u_l, v_l)^t$, which easily yields both its homologue coordinates $(u_r = u_l - \delta, v_r = v_l)^t$ and its associated depth Z thanks to equation 2.28. This section studies the existence of such a tool in multi-simplified geometry. Obviously, as multi-simplified

geometry implies binocular simplified geometry for each image couple, we can start the investigation from the multiple available binocular disparities as illustrated the figure 2.4. Considering that a pixel (u_i, v_i) of image i has homologous $(u_j, v_j = v_i)^t$ in the other images j, it may be given multiple disparities $\delta_{i,j} = u_i - u_j$. Moreover, the global disparity related to depth Z of associated 3D point according to equation 2.28 is measured from image domains centered on the optical axes. In case of parallel decentered geometry (see figure 2.10), the global parallel disparity $\delta'_{i,j} = u'_i - u'_j$ is derived from the decentered views disparity $\delta_{i,j} = u_i - u_j$ from $\delta'_{i,j} = \delta_{i,j} + (a_i - a_j)$ where a_k stands for the horizontal distance from optical axis of the center of image k. This yields the depth to views disparity relation:

$$Z\ \delta'_{i,j} = f\ b_{i,j} \Leftrightarrow Z\ (\delta_{i,j} + \bar{\delta}_{i,j}) = f\ b_{i,j}. \tag{2.30}$$

Where $\bar{\delta}_{i,j}$, the disparity correction for convergence is defined as the global disparity of image centers, $\bar{\delta}_{i,j} = a_i - a_j$ of the converging lines of sight passing through the centers of views. In non decentered setting $a_i = a_j = 0$ and $\bar{\delta}_{i,j} = 0$ which returns to previous equation 2.28 defined in the section 2.1.2.2.2. Furthermore, disparity associated $\bar{\delta}_{i,j}$ with the convergence point $\mathbf{M}_c$ is related to the depth Z_c of $\mathbf{M}_c$ by:

$$Z_c\ \bar{\delta}_{i,j} = f\ b_{i,j} \Leftrightarrow \bar{\delta}_{i,j} = f\ \frac{b_{i,j}}{Z_c}. \tag{2.31}$$

Each of these views $\delta_{i,j}$ disparities relates to the same 3D point which is associated to pixel $(u_i, v_i)^t$. They are thus tied to the depth Z of this point through equations 2.32 where $b_{i,j}$ stands for the baseline signed distance between cameras i and j:

$$\begin{aligned} \forall (i,j) \in \{0,\ldots,n-1\}^2,\ i \neq j \qquad Z\ \delta_{i,j} &= f\ b_{i,j} - Z\bar{\delta}_{i,j} \\ &= f\ b_{i,j}(1 - \frac{Z}{Z_c}). \end{aligned} \tag{2.32}$$

Equations 2.28 and 2.32 clearly show that for any single pixel $(u_i, v_i)^t$, its compatible disparities $\delta_{i,j}$ are proportional to the implied baseline distances $b_{i,j}$. This fact induces that one of those disparities may be chosen as reference from which any other may be re-computed thanks to equation 2.33, where we choose for convenience $\delta_{0,1}$ as the reference:

$$\forall (i,j) \in \{0,\ldots,n-1\}^2,\ i \neq j \qquad \delta_{i,j} = \delta_{0,1}\ \frac{b_{i,j}}{b_{0,1}}. \tag{2.33}$$

Let us now express the baseline signed distances between any image couple from the successive baselines between adjacent cameras:

$$\forall (i,j) \in \{0,\dots,n-1\}^2,\ \ i \neq j \qquad b_{i,j} = sgn(j-i) \sum_{k=min(i,j)}^{max(i,j)-1} b_{k,k+1}. \tag{2.34}$$

If optical centers are equidistant those baselines between adjacent cameras are identical ($\forall i,\ b_{i,i+1} = b_{0,1}$) and the equation 2.34 is simplified:

$$\forall (i,j) \in \{0,\dots,n-1\}^2,\ \ i \neq j \qquad b_{i,j} = (j-i)\ b_{0,1}. \tag{2.35}$$

Finally binocular disparities may be expressed from reference disparity thanks to equations 2.33, 2.34, and 2.35 as :

$$\forall (i,j),\ \ i \neq j \qquad \delta_{i,j} = \delta_{0,1}\ \beta_{i,j}, \tag{2.36}$$

with

$$\forall (i,j),\ \ i \neq j \quad \beta_{i,j} = \begin{cases} \frac{sgn(j-i)\ \sum_{k=min(i,j)}^{max(i,j)-1} b_{k,k+1}}{b_{0,1}} & \text{generic case} \\ (j-i) & \text{equidistant centers} \end{cases}. \tag{2.37}$$

These results clearly show that knowing that optical centers are equidistant and their common baseline $b_{0,1}$ or knowing their successive baselines $b_{k,k+1}$ is enough to use for any pixel $\mathbf{m}_i = (u_i, v_i)^t$ a single binocular disparity δ chosen as reference ($\delta \equiv \delta_{0,1}$ for instance) to express each of its disparities $\delta_{i,j}$ through equation 2.36, each of its corresponding pixels $\mathbf{m}_j = (u_j, v_j = v_i)^t$ thanks to equation 2.38 and its depth $Z = f\ b_{0,1}\ \delta^{-1}$:

$$\forall (i,j),\quad i \neq j \qquad u_j = u_i - \delta\ \beta_{i,j}. \tag{2.38}$$

This section has thus proven that a reference disparity δ may be defined for any pixel $\mathbf{m}_i = (u_i, v_i)^t$. This reference disparity may then identify each of its matching pixels $\mathbf{m}_j = (u_j, v_j = v_i)^t$ thanks to equation 2.38 expressing their abscissa difference. This single scalar disparity value actually expresses a multiple pixel matching across the whole set of images. Indeed assigning reference disparity δ to pixel $\mathbf{m}_i$ implies matching together the whole set of pixels $\{\mathbf{m}_j | \mathbf{m}_j = \mathbf{m}_i - \beta_{i,j}\ \delta\ (1\ 0)^t\}$.
Furthermore this reference disparity is even more straightforward and helpful in case of equidistant centers as the binocular disparities $\delta_{i,j}$ are proportional to the index difference of the two images $j-i$ as $\beta_{i,j} = j-i$.
We notice that the disparity computing using the equation 2.36 ensures that the homologous

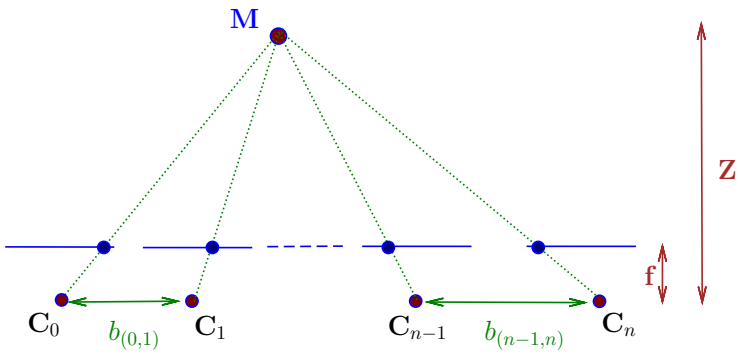


Fig. 2.9 Multi-simplified epipolar geometry with different baseline(s)

pixels of $\mathbf{m}_i$ along all images represent the same 3D point $\mathbf{M}$. We will show in chapter 3 how the disparity computing is integrated in our multi-baseline stereo framework.

2.1.3.3 Rectification

The multiple image rectification consists in fitting all image planes into one common plane. This yields multi simplified geometry. Ayache and Hansen [3], Sun [69], and also An et al. [2] present some methods to perform an image rectification over three views acquired with cameras laid in right-angled triangle. They combine a horizontal image rectification between the central image and the left image and a vertical image rectification between the central image and bottom image. This approach is designed to extend depth from stereo methods to three views. However this technique cannot be used for three (or more) aligned cameras. Kang et al. [34] present an image rectification from multiple calibrated images. They adapt the images orientation and focal such that the cameras share a common image plane. The error derived from the rectification process of multiple images captured from multiple cameras located on semi-circular cannot be ignored in 3D reconstruction purpose. Therefore, the best manner to reduce the geometrical errors is to initially align the multi cameras and to set their optical axis parallel one to the other. Within the RECOVER3D project, we worked with multiscopic units (see chapter 1) composed of multiple cameras defined in aligned toed-in geometry. This layout facilitates the process of finding the matching pixels, as we will see in the next chapter.

2.2 Multi-view methods

This thesis develops two 3D reconstruction methods: multi-baseline stereovision and fusion of shape from silhouettes with multi-baseline stereovision. In this section, we introduce a state of the art for multi-stereovision and shape from silhouettes. Before discussing

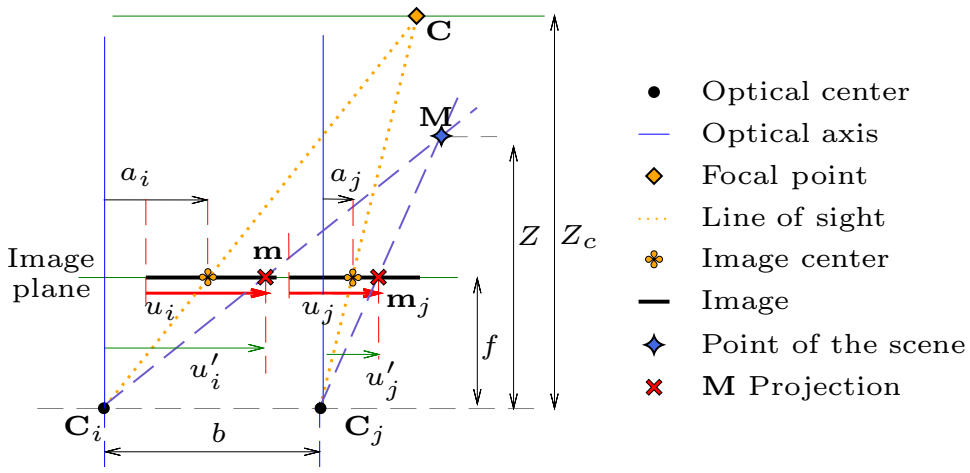


Fig. 2.10 Parallel decentered geometry: relationship between global disparity $\delta'_{i,j} = u'_i - u'_j$ and depth Z.

the multi-view methods, it is important to give an overview of the compatible acquisition systems with the RECOVER3D project constraints as described in the following section.

2.2.1 Multi-view aquistion systems

An important assumption of any multi-view method lies in its required, compatible or intended camera layout since various possibilities exist and may have an impact on the 3D reconstruction strategy.

Most of multi-view methods (notably among those listed on the Middlebury stereo site [61]) are designed for n cameras freely laid out in space. Some apply binocular stereovision on different couples (I_i, I_j) of views and then merge their separate binocular results [27].

The main difficulty in such approaches concerns regularizing the union of separate results, especially in scene areas where reconstructions overlap. Common problems to be solved in such areas are to reduce point density and to resolve ambiguities/inconsistencies.

Shape from silhouettes is one kind of multi-view 3D reconstruction methods which work with converging acquisition systems composed of multiples cameras located around the object(s) to reconstruct (see Figure 2.11). However this acquisition system may also be virtually achieved with just one fix camera shooting an object turning on itself.

Some other multi-view methods, sometimes called multi-baseline stereovision methods, are designed for the "parallel", "aligned toed-in" or "decentered parallel" camera layouts previously discussed (see Figure 2.8). The RECOVER3D studio is composed of multiscopic and monoscopic units as described in the chapter 1. Multiscopic units contain multiple cameras

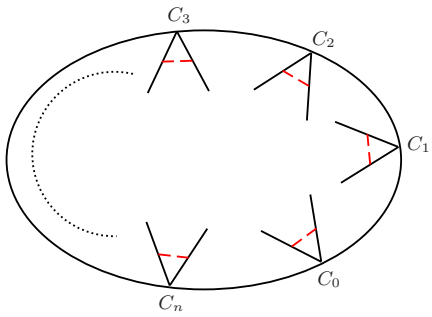


Fig. 2.11 Camera layout: converging acquisition system layout for multi-stereovision methods and shape from silhouettes.

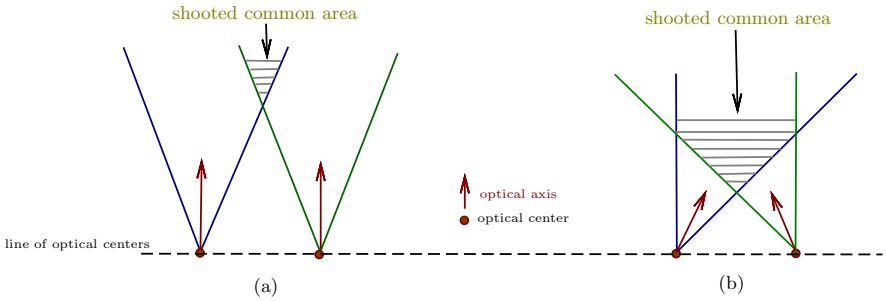


Fig. 2.12 Comparison of parralel (a) vs aligned toed-in (b) geometries according to common scene area.

in aligned toed-in geometry for two reasons: the first is to maximize as much as possible the intersection of the view frustums (see Figure 2.12) of the *n* cameras; the second is that the precision of the calibration for both multiscopic and monoscopic units is more robust when aligned toed-in geometry is chosen for multiscopic units.

2.2.2 Binocular stereovision approaches

During the last fifty years, binocular stereovision has been studied extensively and many matching methods have been proposed. They can be grouped into two different families: local methods (section 2.2.2.3) and global methods (section 2.2.2.4). As mentioned previously RECOVER3D chose to rely on simplified epipolar geometry for its multi-stereovision process. As such, we focus this state of the art study of binocular stereovision on the methods that rely on this assumption and thus usually use disparity. In order to get the 3D scene information, most of those stereo matching algorithms consist of the following steps [60]:

- matching cost computation;
- cost aggregation;

- disparity computation and optimization;
- disparity refinement.

The matching cost is described by the squared difference, absolute difference, cross correlation or any correlation criteria of intensity values at a given disparity, as we will present in the section 2.2.2.2. Whereas cost aggregation is done by summing matching cost over rectangle or square windows. The disparity computation and optimization step is represented for most of the local methods (section 2.2.2.3) by Winner-Talk-All (WTA) optimization. The disparity is computed by selection of the minimal cost aggregation for each pixel. Whereas the third step for stereo matching in global methods is often operated as the "dynamic programming", "graph cut", "belief propagation" or any other optimisation function. The literature gives far less attention to the last step. Ma et al. [44] proposed a method for this step which uses the weighted median filtering. This section will introduce some concepts that are common in stereovision algorithms. Moreover, local and global methods will be presented in detail.

2.2.2.1 Matching constraints

The matching in binocular stereovision is a very difficult search procedure. In order to minimize false matches, some matching constraints must be imposed. In the section 2.1.2.1, we described in details one of the important epipolar constraint. Here we present a list of the commonly used constraints:

- *The uniqueness constraint* can be defined as follows: a given pixel from one image can match no more than one pixel from the other image;
- *The ordering constraint* means that if a point in the scene appears to be to the right of another point in the left image, the relative positioning should be the same in the right image;
- *The symmetric constraint* assures the uniqueness constraint. This constraint means that the correspondence between any two corresponding points is bidirectional as long as there is no occlusion in one of the images.

These three constraints can help to limit the ambiguities generated by textureless areas, repetitive textures or lighting changes between views. However, none of the constraints discussed here can limit the effects of occluded regions. Different global and local methods are proposed to overcome such difficulty as we well see in the sections 2.2.2.4 and 2.2.2.3.

2.2.2.2 Windows matching

The matching in binocular stereovision, involves finding couples of homologous points. Figure 2.4 shows a 3D point $\mathbf{M}$ of the scene, visible in both images (acquired in simplified epipolar geometry) which is projected into $\mathbf{m}_l$ and $\mathbf{m}_r$ of the left and right image respectively. While the pixels $\mathbf{m}_l$ and $\mathbf{m}_r$ of two images represent the same point in the scene, they are

considered as homologous pixels. Binocular stereovision is often based on a window matching working with pixel similarity or dissimilarity measures $\mathcal{MS}$ in order to compute some aggregated matching score between two pixels. The dissimilarity and similarity measure respectively increases and decreases as the likeness between two compared pixels decreases. Given a pixel of left image $\mathbf{m}_l = (u_l, v_l)^t$ and a potential homologue pixel of the right image $\mathbf{m}_r = (u_r, v_r = v_l)^t$ (see Figure 2.13), the aggregated matching score $\mathcal{AM}(\mathbf{m}_l, \mathbf{m}_r)$ (e.g., Normalized Cross-Correlation (NCC), Sum of Squared Difference (SSD) or Normalized Sum of Squared Difference (NSSD)) can be computed between $\mathbf{m}_r$ and $\mathbf{m}_l$ considering individual matching scores through $\mathcal{MS}$ of their neighboring pixels according to some window $\mathcal{W}$.Figure 2.13 shows in red pixels a matching window around a blue reference pixel. We look then for the corresponding pixel in the right image by observing the neighborhood of each pixel located in the same horizontal line. The aggregated matching score for $(\mathbf{m}_l, \mathbf{m}_r)$ can be defined in a generic way by the equation 2.39.

$$\mathcal{AM}(\mathbf{m}_l, \mathbf{m}_r) = \frac{\sum\limits_{v \in \mathcal{W}_{\mathbf{m}_l, \mathbf{m}_r}} \mathcal{MS}(\mathcal{I}_l[\mathbf{m}_l + v], \mathcal{I}_r[\mathbf{m}_r + v])}{\mathcal{N}(\mathbf{m}_r, \mathbf{m}_l, \mathcal{W})}, \tag{2.39}$$

with:

$$\begin{array}{ll} \mathcal{W} & \text{Reference window of size } [\mathcal{W}idth, \mathcal{H}eight] \\ \mathcal{W}_{\mathbf{m}_l, \mathbf{m}_r} & \text{Window } \mathcal{W} \text{ truncated in order not to extend beyond} \\ & \text{image borders while applied around } \mathbf{m}_l \text{ and } \mathbf{m}_r \end{array} \tag{2.40}$$

$\mathcal{MS}(p_i, p_j)$ matching score between two color pixels p_i and p_j:

$$\begin{array}{ll} \sum\limits_{c=0}^{3} (p_i[c] - p_j[c])^2 & \text{Squared component Difference} \\ \sum\limits_{c=0}^{3} \left| p_i[c] - p_j[c] \right| & \text{Absolute component Difference} \\ \sum\limits_{c=0}^{3} \prod_{k=i,j} (p_k[c] - a_k[c]) & \text{Centered component} \\ & \text{Cross-Correlation} \\ & \text{with } a_k = \sum\limits_{v \in \mathcal{W}_{\mathbf{m}_i, \mathbf{m}_j}} \frac{\mathcal{I}_k[\mathbf{m}_k + v]}{card(\mathcal{W}_{\mathbf{m}_i, \mathbf{m}_j})} \end{array} \tag{2.41}$$

$\mathcal{N}(\mathbf{m}_l, \mathbf{m}_r, \mathcal{W})$ normalization function of the sum of neighbor scores

- number of neighbors used (SSD, SAD): $nv = card(\mathcal{W}_{\mathbf{m}_i, \mathbf{m}_j}) = \sum\limits_{v \in \mathcal{W}_{\mathbf{m}_i, \mathbf{m}_j}} 1$
- $nv\times$ Standard deviation (NCC): $\left(\Pi_k \in \{i, j\} \sum\limits_{v \in \mathcal{W}_{\mathbf{m}_i, \mathbf{m}_j}} (\mathcal{I}_k[\mathbf{m}_k + v] - a_k)^2 \right)^{\frac{1}{2}}$

(2.42)

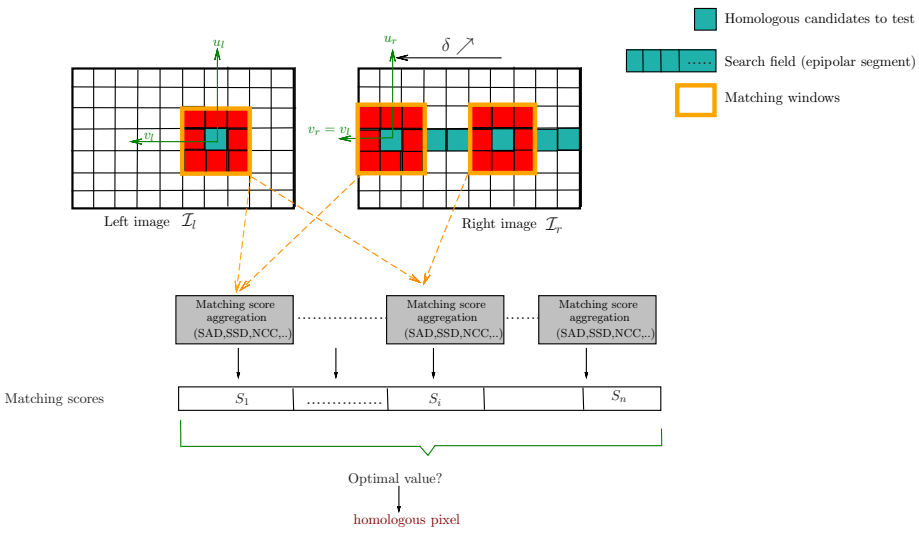


Fig. 2.13 Schematic illustration of the local search for homologue respecting simplified epipolar geometry with WTA process.

2.2.2.3 Local Methods

The disparity map is obtained using only the information located in closest neighbors of the studied pixels. Therefore the common approach for local methods is to assign independently for each pixel of the reference image the disparity value that optimizes its matching score. This is often referred as Winner-Talk-All (WTA) otpimization (see Figure 2.13). Figure 2.18d shows the disparity map obtained from images created by University of Tsukuba (tsukuba) applying a traditional local method. In recent years, local methods experienced tremendous progress. Yoon el al [84]. proposed to integrate the adaptive support weights. The idea is to control the impact of neighboring pixels on final matching score according to a similarity metric with respect to the studied pixel in reference view, most often based on color and spatial similarity. The method of adaptive support weights is based on Gaussian distribution considering similarity and proximity to the central pixel in the support window. Hosni et al. [30] proposed another method close to [84] using geodesic distance to replace the spatial proximity in order to overcome the problem of spatially close but distinct objects influencing each other. Using the methods of Yoon et al. [84] or Hosni et al. [30], the local methods provide better results than traditional local methods as described in Figure 2.18. However, the computation of adaptive support weights is costly. To speed up the aggregation step it can be converted to an image filtering procedure. It turns out that the bilateral weighting scheme of [84] is equivalent to applying a cross bilateral filter or derivations of it to the (u,v) slices of a score volume [57] [58] [36] [31].

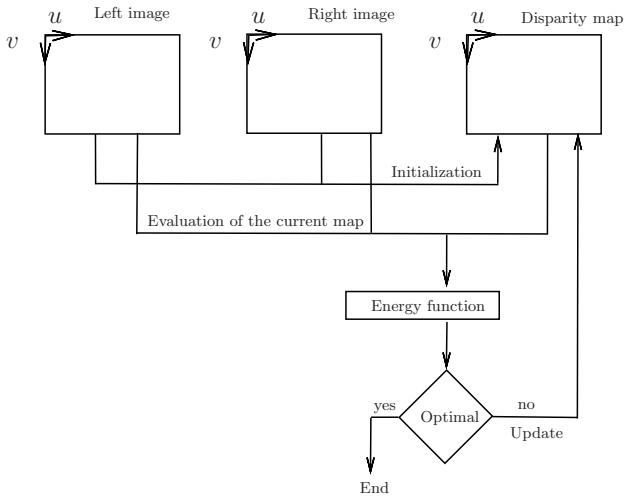


Fig. 2.14 Process of global methods for stereo vision

2.2.2.4 Global methods

Contrary to the local methods that take into account only the neighborhood of each pixel, the global methods optimize the global matching score over the whole image domain. As illustrated in the scheme of Figure 2.14, global methods usually start with similarity criteria mentioned in 2.2.2.2 to find the initial homologues. After calculating the disparity map, a energy function is evaluated on the observed data (right and left images) and the unknown data (the disparity map). The matching problem searches a disparity function $\delta(u,v)$ over the image domain that minimizes the following energy function:

$$E(\delta) = E_d(\delta) + \alpha E_s(\delta)$$

where E_d is the global matching score which sums aggregated matching scores $\mathcal{AM}(\mathbf{m}_l, \mathbf{m}_l - (\delta(\mathbf{m}_l), 0)^t)$ for each pixel $\mathbf{m}_l$ of the reference image, and where E_s is the stabilization function favoring continuity and smoothness properties in the solution. The regularization coefficient α controls the relative weight of smoothness and continuity with respect of global matching score. Once the energy function is defined, some optimization algorithm is used to find a solution close the global optimum. To this end, optimization methods such as dynamic programming, belief propagation, and graph cut are among the most usually used [73].

2.2.2.4.1 Dynamic programming

Dynamic programming (DP) was first used by Baker and Binford [4] for sparse stereovision matching. They proposed an edge-based dynamic programming stereo matching. The basic

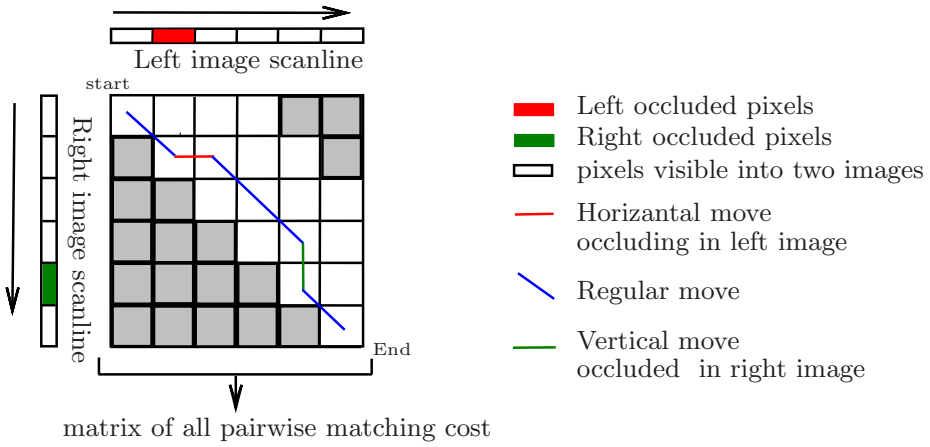


Fig. 2.15 Dynamic programming for two images yields the optimal path through matching cost grid.

idea is to match the corresponding edges of two images of a stereo pair rather than pixels. The disparities for the best edge matches are then interpolated over untextured regions. Unfortunately, the method presumes that the edges are accurately found in both stereo images. More recent approaches have performed dense matching based on pixel intensity or color values [73]. DP attempt to solve the shortest path problem through the matrix of all possible matches between two corresponding scanlines. It is usually done in two passes running, respectively, forward and backward [15]. The forward step constructs for each possible match the optimal path leading to this match from left hand side of the scanlines, and stores in matrix C its score computed as follows:

$$\begin{aligned} \forall u_l, u_r \quad C(u_l,u_r) = min(&C(u_l-1,u_r)+\beta, C(u_l,u_r-1)+\beta, \\ &C(u_l-1,u_r-1)+M(u_l,u_r)) \end{aligned} \tag{2.43}$$

Where $M(u_l,u_r) \equiv \mathcal{AM}(\mathbf{m}_l,\mathbf{m}_r)$ is the aggregated matching score for pixels $\mathbf{m}_l = (u_l,v)^t$, $\mathbf{m}_r = (u_r,v)^t$ in left and right scanlines respectively and $C(u_l,u_r)$ indicates the cumulative cost of the path from the match $(u_l = 0, u_r = 0)$ to the match $(u_l,u_r) \in \{0,..,nc-1\}^2$ where nc is the image width. The cost β of a horizontal/vertical move implying occlusions, is preset to a chosen rather high value if $\mathcal{AM}$ is expressed as a dissimilarity score or rather low value elsewise. Note that only three moves are permitted according to [73]: an horizontal occluding move, (new match occludes preceding one) a diagonal regular move and a vertical, occluded move (new match is occluded by preceding one). The backward pass extracts the optimal path from the matrix corresponding to the global minimum of the cost function

from $(u_l = nc-1, u_r = nc-1)$ to the left most one $(u_l = 0, u_r = 0)$.
Figure 2.15 demonstrates the search grid for two scanlines with six pixels and a disparity range [0,3]. Each (u_l,u_r) cell in this grid means a possible match between the pixels $\mathbf{m}_l$ and $\mathbf{m}_r$ in the left and right images respectively. The minimum cost path going down and right from top-left corner of the grid to its bottom-right corner is found. Thus, the dynamic programming method can be used to find the best possible match sequence between the start and the end matches. A lot of matches are excluded thanks to the order and maximum allowed disparity assumptions. For example, if $\mathbf{m}_{l1}$, $\mathbf{m}_{l2}$ of left image correspond respectively to $\mathbf{m}_{r1}$ and $\mathbf{m}_{r2}$ on the right image and $\mathbf{m}_{l1}$ is the left of $\mathbf{m}_{l2}$ $(u_{l1} < u_{l2})$ then $\mathbf{m}_{r1}$ should not be to the right of $\mathbf{m}_{r2}$ such that $(u_{r1} \leq u_{r2})$. This heuristic order is mandatory as the local best move is selected from $(-1,0),(-1,-1)$ and $(0,-1)$ neighbors. The cells of the matching array corresponding to other forbidden matches are below the diagonal (marked in gray in Figure 2.15) knowing that the two images respect the parallel geometry ($\delta \geqslant 0$).
The three major problems of dynamic programming are:

- selection of convenient cost β for occluded pixels;
- maintenance of the consistency between successive scanline(s);
- enforcement of the ordering constraint (see section 2.2.2.1), requiring that the relative ordering of pixels on a scanline must be the same between the two views, which may not be the case (for thin foreground objects containing images) in scenes containing narrow foreground objects;
- enforcement of the uniqueness constraint (see section 2.2.2.1) meaning that two distinct 3D points cannot be projected in the same pixel of an image.

An alternative to traditional DP, introduced by [60] use recursive algorithm through Disparity Image Space DIS indexed by $(\mathbf{m}_l, \delta)$ in image domain and disparity range. Scanline optimization is a simple approach designed to assess different smoothness terms. The method is asymmetric and does not utilize ordering constraints. This approach fills DIS as follows:

$$C(\mathbf{m}_l, \delta) = \mathcal{AM}(\mathbf{m}_l, \mathbf{m}_l - (\delta \;\; 0)^t) + \underset{\delta'}{opt}\left(C(\mathbf{m}_l - (1 \;\; 0)^t, \delta')\phi(\delta - \delta')\right), \tag{2.44}$$

where ϕ is some monotonical function of disparity difference. According to $\mathcal{AM}$ definition (see equation 2.39) as a dissimilarity (similarity) score, ϕ is chosen increasing (respectively decreasing) and the optimal search is a minimum (respectively maximum) search. The global optimum can again be computed using DP; however, unlike in traditional (symmetric) DP algorithms, the ordering constraint does not need to be enforced, and no occlusion cost parameter is necessary. Scanline optimization can be done in several ways depending on chosen the energy function. Some favor small movements between disparities while oth-

ers encourage wide disparity steps. The latter have the advantage of allowing all possible displacements but tend to smooth the depth map in the case of locally small differences in disparity. Both DP and Scanline optimization algorithms suffer from the well-known difficulty of enforcing inter-scanline consistency, resulting in horizontal "streaks" in the computed disparity map. On the other side, both algorithms require enforcing the uniqueness constraint. However among different methods developped in DP, Veksler [76] impose smoothness in both horizontal and vertical directions to obtain a disparity maps close to ground truth.

2.2.2.4.2 Belief Propagation

The belief propagation algorithm was first formulated by Judea Pearl in 1982 [55], who formulated this algorithm on trees. Trees are graphs that contain no loops. After that, it has since been shown to be a useful approximate algorithm on general graphs like as Markov Random Fields (MRF) models [79]. This latter is called "Loopy Belief Propagation" (LBP) which is an iterative message passing algorithm. At each pass, for every pixel in the image, the method computes a message for all 4-neighbors of that pixel at given disparity δ.
The messages from each pixel $\mathbf{m}_i$ in each direction (left, right, up, down : $d \in D_4 \equiv \{(-1\ 0)^t, (0\ -1)^t, (1\ 0)^t, (0\ 1)^t\}$) are stored in four local vectors $msg^d_{\mathbf{m}_i}$ indexed by disparity range $[\delta_{min}, ..., \delta_{max}]$. The message emitted from pixel $\mathbf{m}_i$ to its neighbor $\mathbf{m}_i^d \equiv \mathbf{m}_i + d$ at given disparity δ is the estimated penalty for the neighbor taking the disparity δ. LBP process described in algorithm 1 starts by an initalisation of messages to 0 or 1 depending on the chosen energy formulation. The choice of the message passing order (right, left, up, down in the algorithm 1) is arbitrary. Let us formally define the message using **sum-product**:

$$msg^d_{\mathbf{m}_i}(\delta) = \sum_{\delta'=\delta_{min}}^{\delta_{max}} \left(e^{-E_d(\mathbf{m}_i,\delta')} e^{-E_s(\delta,\delta')} \times \prod_{d' \neq d \in D_4} msg^{-d'}_{\mathbf{m}_i+d'}(\delta') \right), \tag{2.45}$$

where $E_d(\mathbf{m}_i, \delta')$ and $E_s(\delta, \delta')$ are data and smoothness function. The equation 2.45 represents message passing from $\mathbf{m}_i$ to $\mathbf{m}_i^d = \mathbf{m}_i + d$ pixels about the disparity δ. These messages then work to compute later messages at subsequent time steps [82]. The algorithm runs as many steps as required, sometime for hundreds of iterations, and computes disparity values for each pixel according to their belief. The belief for pixel $\mathbf{m}_i$ about the disparity δ can be written using **Sum-product** function as follows:

$$Belief_{\mathbf{m}_i}(\delta) = e^{-E_d(\mathbf{m}_i,\delta)} \prod_{d \in D_4} msg^{-d}_{\mathbf{m}_i+d}(\delta) \tag{2.46}$$

Algorithm 1: Loopy Belief Propagation algorithm

Data: Stereo images
Result: Disparity map
Initialize all messages
for *t iterations* **do**
    at every pixel pass messages right through all disparities δ
    at every pixel pass messages left through all disparities δ
    at every pixel pass messages up through all disparities δ
    at every pixel pass messages down through all disparities δ
find the best disparity at every pixel $\mathbf{m}_i$ by WTA on belief scores.

In the state of art, other fonctions like as **Max-Product** or **Min-Sum** are applied to compute the messages and the belief [46]. The Figure 2.18g shows the results derived from LBP described in algorithm 1 on Tsukuba images. The results show the ability of LBP to overcome the inter-scanline consistency problem. However, LPB still lacks ability to preserve the object edges in disparity maps and thin objects disparity as the lamp arms in the Figure 2.18.

2.2.2.4.3 Graph Cut

The local stereovision methods try to match pixels in the left image with their corresponding pixels in the right image without considering the disparity values attributed to other pixels. Although these methods are fast, they do not deal with neighboring inter-scanline consistency. Pixels near each other usually should have close disparities, unless they lie on different sides of an edge. For this reason, graph cut is used to formulate the stereovision problem in term of energy minimization [9] using Markov Random Fields [79].
The generic graph cut method can be described in two steps. The first one builds a weighted graph $G = (E,V)$ consisting of a set of nodes $T \subseteq V$ (usually those nodes correspond to pixels), a set of terminal nodes $S = \{s_1, s_2, ..., s_k\} \subseteq V$, and a set of edges $E' \subseteq E$ that connect pairs of nodes and are assigned some weight. The second step cuts the graph in order to find the optimum classification of T nodes in k classes identified by each $s_k \in S$. This step is applied by using Multi-way cut (also known as k-way cut) which computes a set $C \subset E$ called "cut" of edges such that in $G'(E-C,V)$, no node in T is backward connected to more than one node in S. The multi-way cut problem is defined as finding the cut where $\mid C \mid = \sum_{e \in C} weight(e)$ is minimal. For $k = 2$, the problem is reduced to the $s-t$ min-cut method introduced by Ford and Fulkerson [22] and known to have a polynomial time solution. The $s-t$ min-cut method finds the edges which separate the source node from the sink node and satisfy the two following conditions:

- The sum of the weights of these cut edges is as small as possible which implies a maximum flow between the source and sink (S nodes) according the theorem of Ford and Fulkerson[22].
- It no longer exists in the graph any path linking the source to the sink nodes.

In contrast, for $k \geq 3$, multi-way cut terminal problem is known to be NP-hard. Dahlhaus et al [17] propose approximate multi-way cut for $k \geq 3$: for each terminal, a minimum cut that separates that terminal from every other terminal and union of these cuts yielding the approximation of the multi-cut. The latter is considered as forward process and does not undo the decision-making separating a previous terminal from others.
In the stereovision context, T is usually considered as the set of pixels whereas S is represented by the set of disparities (see Figure 2.16). We can customize the objectives of multi-way cut for stereovision purpose as follows:

- Every pixel (node in T) remains connected to one disparity node (in S).
- Edges between neighboring nodes in T exist in the final graph only if those pixels are connected to same disparity node in S.
- Approximation of the multi-cut is run.

Roy [59] proposed to represent the stereovision problem using only two terminals as illustrated in the figure 2.17. The T nodes are built from couples (pixel, disparity) arranged in Disparity Image Space (DIS) grid. The source node, s, (see Figure 2.17) is located at the beginning side of the graph and connected to all nodes in the plane of minimum disparity, and the sink node, t, is located at the end side of the graph and connected to all nodes in the plane of maximum disparity. There are two types of edges between the nodes: $t-$links and $n-$links. $t-$links connect the neighbor nodes at same pixel and different disparity planes. The weight of $t-$links [59] is equal to the mean value of matching costs of two nodes. Whereas $n-$links connect the neighbor nodes in the same disparity plane, and their weights hold the smoothness energy. The graph cut will then separate the nodes in two sub-sets to obtain the optimal disparity map. This map is constructed by the assignment of each pixel with the maximum value of disparity for which the corresponding node is still connected to the source.

When comparing the figures 2.18 e, f, and g, we observe that graph cut provides better results than DP and belief propagation. However, the graph cut method suffers from a big disadvantage. Unlike other inference algorithm (like as loopy belief propagation), it does not provide any uncertainty measure associated with the produced solution. This is a serious drawback since researchers do not obtain any information regarding the credibility of a particular disparity assignment in a graph cut solution.

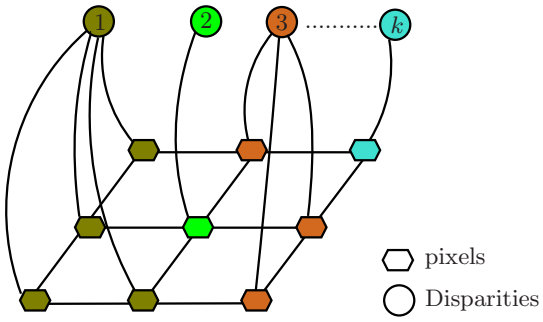


Fig. 2.16 Representative scheme for stereovision problem within graph for $K \geq 3$ terminal nodes.

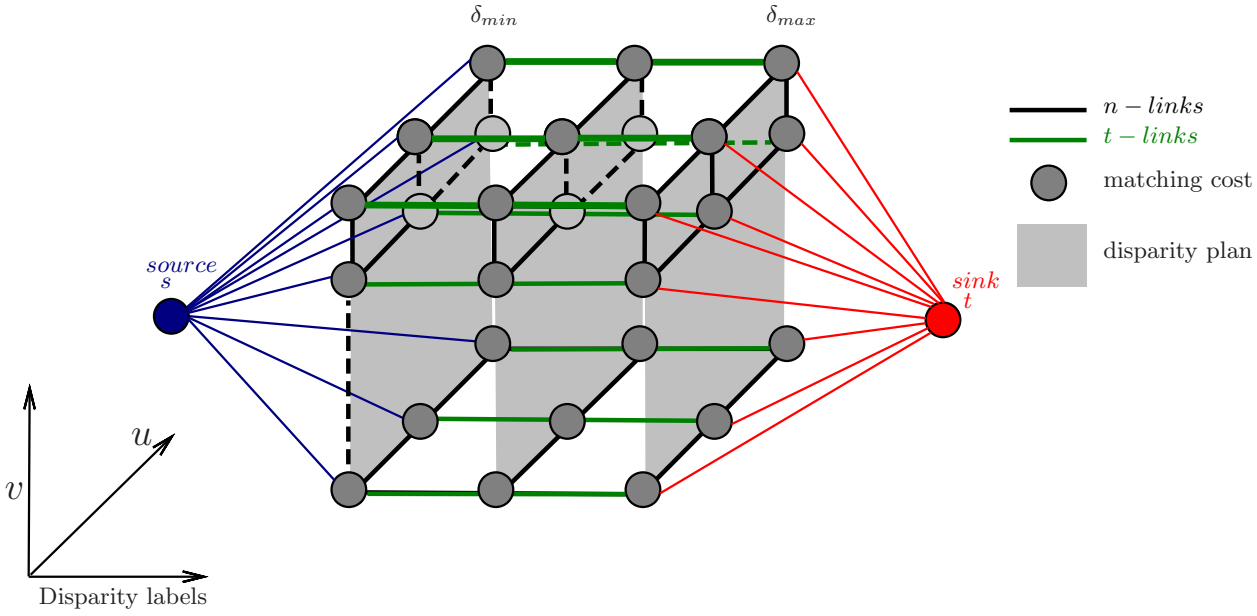
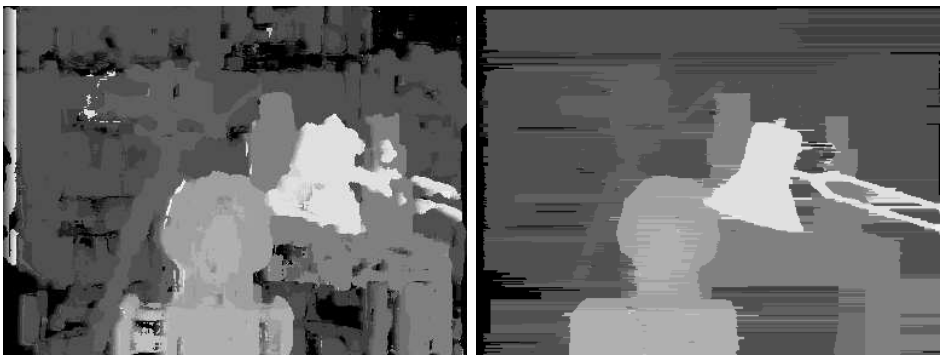


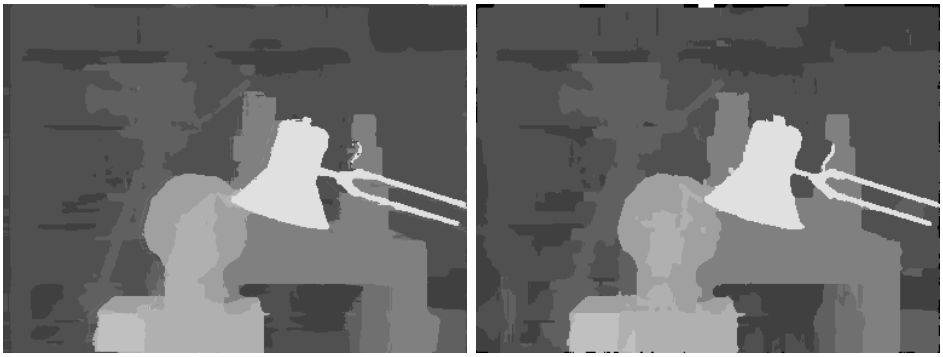
Fig. 2.17 Representative scheme for stereovision problem within graph for $K = 2$ terminal nodes.

(a) (b) (c)



(d) Local (SSD). (e) DP, source: [6].

(f) Graph cut, source: [9]. (g) Loopy Belief propagation.



(h) Adaptive support-weight, source: [84]. (i) Geodesic support weight, source:[30].

Fig. 2.18 f,b) Tsukuba synthetic stereo image of middlebury site, c) Ground truth disparity map , d) to i) results of local method (SSD), Dynamic programming [6], Graph cut [9], Loopy belief propagation, Adaptive support-weight approach [84] and Geodesic support weights [30].

2.2.3 Multi-view stereovision

The following section gives an overview of multi-view stereovision. The term multi-view stereovision (MVS) refers to stereovision-based reconstruction from $n > 2$ views with currently no a priori on the aligned of cameras.
While binocular stereovision [60, 37] enables to estimate depth, adding more images leads to more robust and accurate 3D reconstruction thanks to information redundancy [52, 14, 74]. Unfortunately, the matching process becomes more complex and still lacks robustness in regions either untextured, regularly textured, and/or totally occluded. Thus, the main difficulties are occlusions, changes in appearance, and ambiguities.
MVS has been an active field of research for several decades and at this moment more than seventy algorithms are listed on the Middlebury Multi-View Benchmark website [63]. This benchmark provides a commonly accepted test suite to evaluate the quality of multi-view stereo algorithms. MVS methods may also be categorized into three main groups :

- *Scene-based methods* attempt to recover photoconsistent models that minimize some measure of the discrepancy between the different image projections of their surface points. Space carving [40] algorithms represent the volume of space around the modeled object by a grid of voxels, and erode this volume by carving away successive layers of surface voxels with high photometric discrepancy.
- *Image-based methods* compute a set of depth or disparity maps which are merged later [48],[26] or to which they apply constraints [25], [72] to ensure a consistent 3D scene reconstruction. Two major classes are distinguished, the first contains the methods that expect a more restrictive camera layout, typically multi-baseline (a synonym for multi-simplified epipolar geometry with possibly non equidistant centers), directly match n-tuples as multiscopic pixel sets [51], [33]. The second class composed of the methods intended for a free camera layout. Some are more computationally intensive techniques are dedicated to MVS from community photo collections (CPC) [26] and have gained an increasing interest. They have to handle a large number of uncalibrated views of a scene [26]. New difficulties then arise as such views are typically shot at different times, with differing acquisition geometries (viewpoints, angles, focal lengths, resolutions), and usually differing environmental conditions (weather, exposure, occluding objects). This makes it necessary to restrict the matching to subsets of views sharing similar exposure, and empower the methods to deal with significant baselines (distances between the cameras).
- *Feature-based methods* compute sparse correspondences by first matching feature points which can be powerfully estimated and more robustly matched than regular pixels. In a second step a surface model is fitted to the reconstructed features [75].

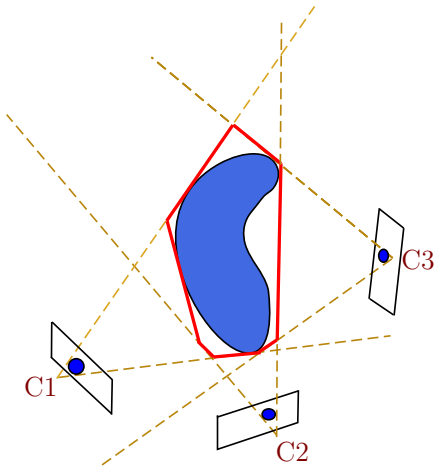


Fig. 2.19 Silhouette-based reconstruction.

Classical solutions for 3D reconstruction from multi-baseline stereovision are usually image-based (second group). They consist in matching algorithms that aim at finding homologous pixels in different images, which represent the same 3D point in the real scene. The most efficient of these methods match multiscopic pixel sets [51, 33] composed of one pixel per image, pair-wise verifying epipolar constraints. However, these methods still suffer from traditional binocular stereovision problems like occlusion and textureless zones.
Within the RECOVER3D project, we proposed a novel framework lying on scene-based multi-baseline stereovision. Instead of searching matches for image pixels like most multi-view stereovision methods, our method works completely in the disparity space exploiting the geometry, similarity and other informations to yield precise reconstruction even in low texture and semi occluded regions (as we will see in the next chapter).

2.2.4 Shape from silhouette

A silhouette is a binary mask associated with a given view that includes all pixels corresponding to the projection of a point of the 3D object to be reconstructed. In Figure 2.19 the colored pixels in the images taken by cameras C_1, C_2, and C_3 correspond to silhouettes of the 3D object in each view. Shape from silhouette [64] therefore involves estimating the visual hull of the 3D object defined as the intersection of generalized cones built from each optical center and associated silhouette. One slice of the polyhedral visual hull (described in detail in the section 3) is represented by the red polygon in Figure 2.19.

2.2.4.1 Silhouette extraction methods

The extraction of a silhouette involves isolating in an image the region of object projection to be reconstructed from the scene background. Several methods are proposed to achieve this result, grouped according to the following categories:

- *Color difference-based methods* exploit the knowledge of the background color. To extract an object from the background, the technique uses colorimetric image differences. To overcome the problem of variations in lighting in the background, the "chroma keying" technique is often favored. "Chroma keying" is one of the most common and most frequently used semantic segmentation techniques within audiovisual context. Video acquisition takes place against a "key color" background, generally blue or green. The problem of shadowing in the background is solved using learning techniques such as Gaussian mixture model or "k-means" [67, 86].
- *Region based methods* aggregate, step-by-step, pixels with shared colorimetric properties. They establish region filling heuristics within an image by propagating local criteria, often based on the image's gradient (higher at the edges and lower in the middle of the area). The most commonly used methods in this category include histogram segmentation, region growing and region merging. For a more detailed presentation of region-based segmentation methods, Caillet's doctoral thesis [11] is a clear and useful resource.
- *Contour-based methods* involve extracting the connected components using a threshold of the image gradient. Using these methods [35], the silhouette is characterized by its edge with the background of the scene.

2.2.4.2 Visual Hull (VH) reconstruction methods

VH reconstruction methods are classified into two major groups: i) polyhedral approach ii) volumetric approach. Polyhedral approaches deduce the object's visual hull as the surface of the intersection of silhouette cones from each camera. The silhouette cone associated with a camera (dotted line in Figure 2.19) is defined by the set of infinite triangles delimited by half-lines connecting the optical center with two neighboring pixels in the contour of the silhouette. These triangles are then segmented as polygones lying inside each other silhouette cone. The reconstructed object is therefore described by its surface, usually represented in the form of a triangular mesh [41]. Volumetric approaches subdivide the scene space according to a regular grid of cells, known as voxels (volume elements) and labelled "in" or "out". In these approaches, voxels are labelled "in" when they project into the silhouette for each camera. The VH is then described by the set of "in" cells within the discrete grid [71]. In Figure 2.21, all voxels inside of the red polygon cells are labeled "in".

2.3 Hybrid methods from stereo and silhouette

The objective of this this thesis is to propose to merge two 3D reconstruction methods: silhouette-based and stereovision-based reconstruction. Modeling objects using these methods has been addressed by many different algorithms during the last decades. In this section, we present a state of the art of methods merging these two techniques and propose to classify them into three major groups:

- Stereovision methods guided by visual hull (section 2.3.1).
- Collaborative methods applying simultaneously criteria borrowed from both techniques (section 2.3.2).
- Separate application of both methods with further merging of their results (section 2.3.3).

2.3.1 Stereovision methods guided by visual hull

In this section, we focus on the guidance of stereovision by the results derived from silhouettes based reconstruction. Space carving initialized by visual hull [16] is considered as one of the efficient method for 3D reconstruction from multiple views which can be classify in this category. Other methods are proposed to exploit the broad localisation results (like the disparity range [42]) from shape from silhouettes in stereovision computing as we will show in the next section.

2.3.1.1 Space carving initialized by visual hull

We mentioned in section 2.2.3 that space carving is one of scene-based methods of the multi-view stereovision. However, the visual hull can be used to improve space carving. In this section, we will explain in detail the space carving method and show the advantage of working with visual hull. The space carving proposed by Kutulakos et al [40] is based on the idea of voxel coloring as proposed by Seitz et al [62]. This latter assumes that cameras are laid on a same side of the scene. Voxel coloring involves subdividing the regular grid of voxels into successive layers, from the nearest to the furthest of this "camera side". The serious drawback of this method is that the cameras should be placed on one side of the object to be reconstructed. Voxel coloring is based on the hypothesis that a voxel on the surface of an object must have the same color in each view in which it is not occluded, called as a photo-consistent voxel. On the basis of this statement, the voxel coloring process is written as described in Algorithm 2 and can effectively handles the occlusion problem. For example, two voxels, taken from different layers, can be projected onto the same pixel in a given view. The voxel from the nearest layer occludes the other. To solve this problem, the method takes into account the fact that a voxel from a layer i cannot occlude a validated voxel from a layer j when $j < i$, as illustrated in Figure 2.20. Therefore, the method com-

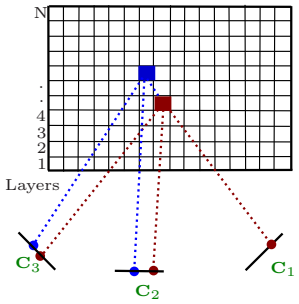


Fig. 2.20 Identifying explicitly the occluded voxel thanks to plane-sweeping from the nearest to the furthest from the cameras: in $\mathbf{C}_1$, the blue voxel is occluded by a previously validated red voxel [62].

Algorithm 2: Voxel coloring

Data: the sequence of calibrated images
Result: the volume of voxels *S* representing the reconstructed object
Initialize a bounding box box containing the object and divide it into layers
Initial set of validated voxels is empty $S = \{\}$
for *each layer* ***ly*** *from the nearest to the farthest from the cameras* **do**
  for *each voxel* ***g*** *in* ***ly*** **do**
    projection of **g** on all the images where it is visible according to *S*
    if ***g*** *is photo-consistent* **then**
      add **g** to *S*

putes the photo-consistency to voxel *g* according only within the set of images where it is visible and not occluded by a previously validated voxel.

The disadvantage of voxel coloring method lies in its inability to completely reconstruct the object due to the sidewise layout of cameras. The space carving algorithm, introduced by Kutulakos et al. [40], can be seen as an extension of the previous method adapted to an arbitrary camera arrangement. This is based like voxel coloring on the photo-consistency of surface voxels. For example, the object in figure 2.21 shows a concavity ignored by the silhouette-based reconstruction technique illustrated by the red polygon. The voxel $\mathbf{g}_1$ found on the visual hull, is projected on differently colored pixels in views taken from cameras 1 and 2.

The space carving relies on sweep planes normal to the three principal axes *x*, *y* and *z*. Only cameras behind the sweep plane are used to manage the occlusions and photo-consistency measures. For example, in figure 2.22a, the voxels in the highlighted increasing plane according to *x* axis are checked for visibility in the cameras 1 and 2, while for the decreasing plane according to *y* axis in the figure 2.22b, voxels are checked in the cameras 3 and 4.

Algorithm 3: Space carving guided by visual hull

```
Data: the sequence of calibrated images
Result: a volume of voxels representing the modeling object
Initialize the volume with visual hull
repeat
    foreach sweep plane in the 6 main directions do
        foreach voxel g labelled "in" in the current plane do
            project g onto the cameras in the sweep plane background (where it is
            visible)
            if g is not photo-consistent then
                g is labeled "out" in the volume
until no more voxels have been eliminated in last step;
```

According to Kutulakos et al. [40], a voxel is not visible by a camera if it is out of its view frustum or if it is occluded. Therefore, they consider an x-increasing sweep plane, if the voxel $\mathbf{g}_1 = (x_1, y_1, z_1)$ occludes the voxel $\mathbf{g}_2 = (x_2, y_2, z_2)$ then $\mathbf{x}_1 < \mathbf{x}_2$. As $\mathbf{g}_1$ is evaluated before $\mathbf{g}_2$, the occluder is always validated before checking the occluded. The main drawback of the original implementation described in [40], is the algorithm intialization. It is often necessary to initialize the algorithm with a very large volume in order to ensure that it completely encloses the surface. Each voxel must then be tested in turn for consistency in the images which results in a high computational load. Cross et al [16] proposed to use the visual hull as a starting point for the space carving algorithm (see algorithm 3). The visual hull completely encloses the object surface. Therefore, using the visual hull at the beginning of the space carving algorithm have many advantages i) easy to compute ii) tight outer boundary iii) parts of visual hull already lie on the surface and should be already photo-consistent.

2.3.1.2 Visual hull regularized stereo

Fan et al in 2008 [19] proposed another method to merge VH-based space carving and stereovision. The acquisition system includes a single fixed camera. The modeling object rotates on itself after each shot. The method consists of two steps:

- applying VH-based space carving
- applying binocular stereovision in global method for each couple of images and solve the problem of global optimization using the information from VH-based space carving

As we saw in section 2.2.2.4, the matching problem using global method consists in finding the disparity function which minimizes the following energy function:

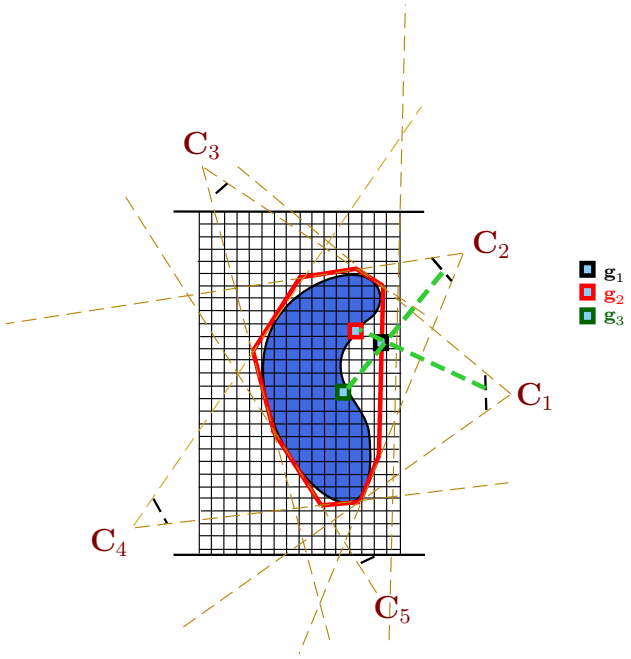
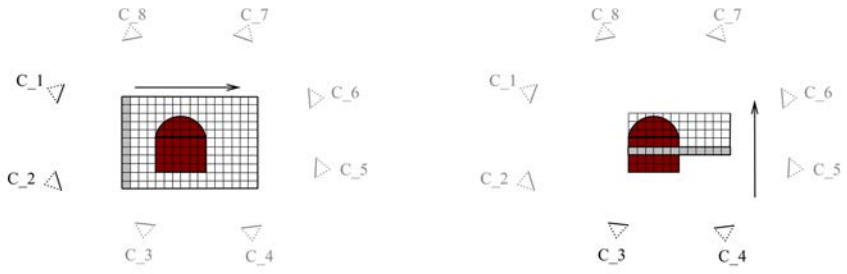


Fig. 2.21 Volumic VH : improvement by identifying concave zones from photo-consistency.



(a) (b)

Fig. 2.22 Configuration of cameras for space carving initialized by visual hull method.

$$E(\delta) = E_d(\delta) + \alpha E_s(\delta) \tag{2.47}$$

The equation 2.47 has been described in details in the section 2.2.2.4. Fan et al [19] have proposed to add a term to the previous formula, which expresses a new constraint by integrating the information derived from VH-based space carving. This term is called E_{sc} and defined as follows:

$$E_{sc}(\delta) = \sum_i \sigma(\delta_{\mathbf{m}_i}, \delta_{pred}), \tag{2.48}$$

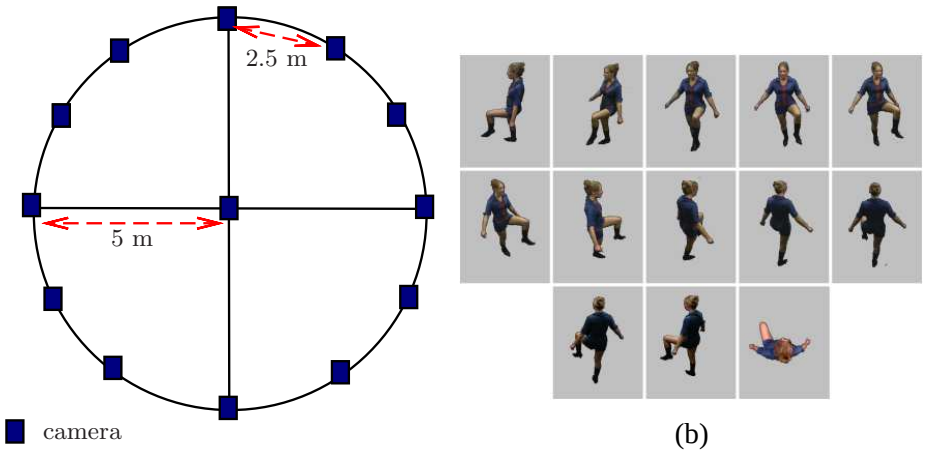
where $\sigma(\delta_{\mathbf{m}_i}, \delta_{pred})$ calculates the difference between the disparity of the studied pixel and the predicted disparity for the same pixel obtained from VH-based space carving. We can then describe the energy function as following:

$$E(\delta) = E_d(\delta) + \alpha E_s(\delta) + \lambda E_{sc}(\delta), \tag{2.49}$$

where λ is the regularization coefficient that controls the constraint E_{sc}. After defining the function of energy, Fan et al [19] used dynamic programming to find a solution close the global minimum. Their results show a more detailed depth variation than that obtained by the visual hull based space carving or stereovision with dynamic programming method.
Ming Li et al [42] proposed also a method for improving the results of stereovision using the visual hull obtained by polyhedral approach. Their acquisition system consists of six cameras arranged around a scene and grouped in pairs connected to computers (called clients), all those "client" computers being connected to a single server. Their method consists of three steps for modeling an object. The silhouettes of the object are estimated on each client. The server then computes the visual hull. Finally, these clients use the visual hull to guide depth maps computing. The visual hull can accelerate the calculation of depth maps by restraining the process to pixels belonging to silhouettes and not on the whole set of pixels. The disparity search range is also reduced by calculating the limited disparities $\{\delta_{min}, \delta_{max}\}$ of each pixel **m** of the matching window. Then this interval (segment $\{\delta_{min}, \delta_{max}\}$ of the ray associated with the pixel) is projected in the other image in order to restrict the search space on the associated epipolar segment. This stereovision algorithm is a local method and the quality of the reconstruction suffers particularly in textureless or uniform areas.

2.3.2 Collaborative methods applying simultaneously criteria borrowed from both techniques

In this section we present another group proposed of hybrid methods from stereo and silhouette. The methods classified in this group work with deformation methods (e.g., snake)



(a) (b)

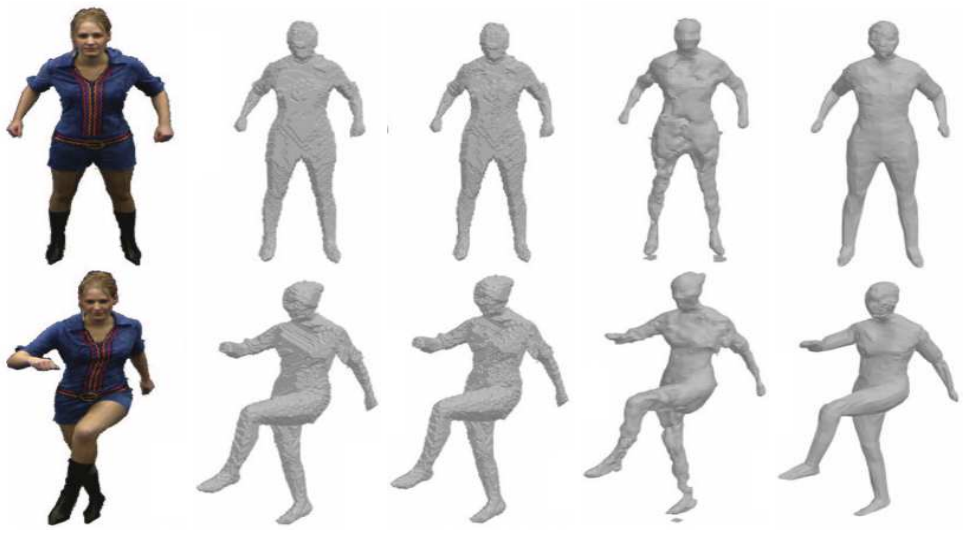
Fig. 2.23 Example of camera layout and associated views from Hilton and Starck method [29]

exploiting concurrently the information derived from silhouette-based reconstruction and stereovision. Hilton and Starck [29] propose a model-based reconstruction method relying on visual hull and stereovision information. The proposed acquisition system contains 13 cameras including 12 lateral cameras positioned around the object and one camera overhead to constrain the visual hull as illustrated in figure 2.23a. The method requires prior knowledge of the feature points of the modeled person. The proposed method optimizes a generic human model mesh to satisfy an energy function. This function consists of three terms E_v, E_s, and E_r expressing respectively the constraints of the visual hull, stereovision and feature points. The energy function is then written as follows:

$$E_{global} = E_v + E_s + E_r \tag{2.50}$$

The energy function E_{global} minimization is carried out using a gradient descent. Thanks to their assumptions about the scene, the results (see figure 2.23b) show a coherent structure for different frames of the sequence. Figure 2.24 shows that using two techniques produces more robust and efficient results. The main disadvantage of this method is the restriction on the scene content which is dedicated to human reconstruction.

Esteban et al. [28] proposed a 3D reconstruction method merging silhouette-based and stereovision-based reconstruction. This method can be classified in this group and is applied to 36 images. The images are taken by a single camera. The object to be reconstructed in 3D rotates on itself between each shot. The method reconstructs the object visual hull thanks to a space carving method. Stereovision based multi-resolution is then applied using



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

Fig. 2.24 Shape reconstruction, source:[29]: f) camera image b) visual-hull c) voxel-colouring d) merged stereo e) model-based.

this initial reconstructed object. The method divides the images into different resolutions (called layers). For the first layer, the method tries to find the depth for each pixel of each image using the depth interval defined by the initial reconstructed object. For the next layer, the depth interval defined in precedent layer works to constrain the depth search for current pixel. A detailed description of Hernandez et al's greedy depth map estimation approach is presented in [28]. After finding the best depth for a pixel **m**, the related 3D position **M** is calculated. The method then adds the score of correlation to the voxel corresponding to **M**. To merge all the information computed using visual hull and stereovision based multi-resolution, a classical method of deformation (*Snake*) is proposed to obtain the mesh as closest as possible to the actual surface of the object. The deformation method is considered as an energy minimization problem and expressed by the following formula using a step variable k for describing the evolution of the surface S of R^3 :

$$\forall i \in \text{mesh vertex} \quad S_i^{k+1} = S_i^k + \Delta t(F_{stereo}^{k\,i} + F_{sil}^{k\,i} + F_{int}^{k\,i})$$

A first estimate S^0 of the actual surface is found by space carving, the modeling object is then subjected to three types of forces:

- F_{int} aims at maintaining the surface adequately smooth and is the regularization term. F_{int} is defined as a force pushing vertex number i of the mesh towards the gravity center of its neighborhood :

(a) (b) (c)

Fig. 2.25 Shape reconstruction, source:[28]: f) visual hull b) final model c) texture mapping.

$$F_{int}^{k\,i} = \frac{1}{card(ng_i)} \sum_{j \in ng_i} \mathbf{v}_j^k - \mathbf{v}_i^k$$

where ng_i is the set of neighbor vertices of i^{th} vertex, and $\mathbf{v}_i^k$ is the 3D position of i^{th} vertex at step k.

- The force F_{stereo} deforms the model in order to minimize its distance to the mesh constructed by stereovision.
- F_{sil} deforms the model to minimize the distance between the projection of the model on each image and silhouettes of the images.

The result of the method [28] illustrated in figure 2.25 shows that the modeling object provides high quality reconstruction. However, their results are based on an acquisition system composed of only one camera. This system allows to obtain images in a control environment (light, background) provides the best results compared to the case of using more than one camera around the object.

Vogiatzis et al [78] propose another method formulating photo-consistency as a global energy minimization, using volumetric graph cuts. Graph cuts extract an optimal surface from a volumetric Markov Random Field. They first build a base surface (outer boundary) as visual hull S_{base} and an inner boundary surface S_{in} lying at a constant distance inside the outer boundary which defines a volume $\mathbf{V}l$ enclosed by S_{in} and S_{base}. Voxels of this volume $\mathbf{V}l$ become nodes in the flow graph. The photo-consistency measure determines the degree of consistency of a point identified as center of a $\mathbf{V}l$ voxel. Finally, optimal surface is obtained as minimum cut solution of the weighted graph. The algorithm proposed uses the visual hull of the scene to infer occlusions and as a constraint on the topology of the scene. Figure 2.27c shows that this method provides best result comparing to visual hull or space

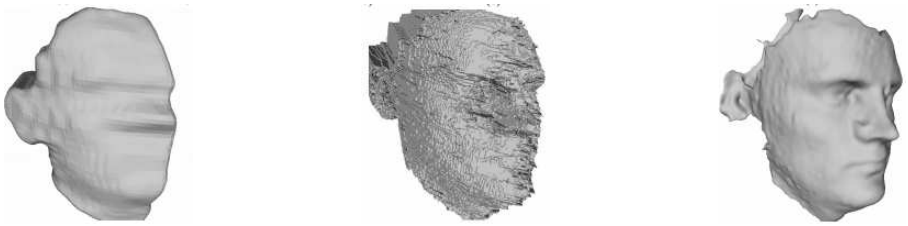
(a) (b) (c)

Fig. 2.26 Face reconstruction using Furukawa et al method [23] : a) one of the input image, b, c) two views of texture-mapped reconstructed patches.

carving methods. The main problem of this method based on graph cut [78] is that for high resolutions of voxel grid, the image footprints used for photo consistency measures become very small which often produces noisy reconstructions in textureless regions.
Furukawa et al. [23] propose a flexible patch-based algorithm for calibrated muti-stereovison using visual hull. The method starts by computing a dense set of small rectangular oriented patches covering the surfaces visible in the images. Then the algorithm converts the resulting patch model into an initial mesh deforming iteratively visual hull model towards reconstructed patches. The deformation is performed applying forces depending on three terms: 1) *a smoothness* term for regularization 2) *a photometric consistency* term derived from reconstructed patches, and finally 3) *a rim consistency* term pulling the rim of the deforming surface towards the corresponding visual cones. Figure 2.26 describes the face reconstruction result using Furukawa et al [23] method. According to the authors, their results are better than those of Esteban's method [28] especially at sharp concave structures (the results of these methods and other are available on middlebury site [61]). However, in the RECOVER3D project, our acquisition system and multi-baseline stereovision method described in the chapters 1 and 3 permit to exploit the information derived from silhouette-based and stereovision-based reconstruction in a novel and powerful way as we will see in chapter 4.

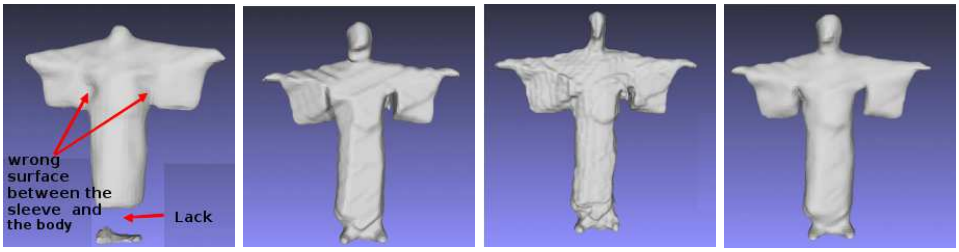
2.3.3 Separate application of both methods with further merging of their results

In this section, another group to merge silhouette-based and stereovision-based reconstructions is presented. Methods in this group start by applying the two methods independently and then merge their results. As we have seen for space carving (section 2.3.1.1), the visual hull is carved gradually until the photo consistency is satisfied. Matsuda et al [45] propose another method to carve the visual hull directly using 8 cameras which are posi-



(a) (b) (c)

Fig. 2.27 Face reconstruction by following methods, source:[78]: a) visual hull, b) space carving, c) method proposed by [78].



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

Fig. 2.28 Object reconstruction following several methods: f) space carving, b) stereovision, c) space carving + graph cut, d) method proposed by Matsuda et al. [45], source [45]

tioned around an object. They construct two point clouds from stereovision and silhouettes techniques. The proposed method maintains the constructed points from stereovision that meet the following conditions:

- They are not near to VH surface,
- their normals are not significantly different from the nearest VH surface normal.

After identifying the acceptable reconstructed points by stereovision, the method removes the voxels of VH volume that satisfy the condition of interrupting the lines between acceptable stereovision points and optical centers of images. Figure 2.28 shows that the proposed method by Mastuda et al [45] provides better results than those obtained by space carving. However, we can note that above conditions determining whether the points derived from stereovision are credible or not, are not always reliable. For example, if we want to reconstruct an object with strong concavities, we can find points that belong to the surface of the object far from the VH surface and whose normals are significantly different from the normal of the nearest surface of the VH. Another method proposed by Song et al. [65] merges both techniques by separate application of both methods. Depth maps and VH are generated. The point cloud is then extracted from all the depth maps containing the outliers and redundancy information. The outliers information are rejected in two steps. The first step is

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

Fig. 2.29 The reconstruction steps for the Captain sequence: f) visual hull, b) stereovision point cloud, c) points cloud extracted from third type voxels, From d) fusion b) and c), e) reconstructed model using poisson surface reconstruction, f) texturing mapping to e), source [65].

achieved by remove all the points of the cloud which are out of VH. A voting octree is built for the point cloud and each voxel of this octree contains the sum of the individual correlation scores derived from point cloud of stereovision. Therefore, the second step includes deleting the voxels which involve aggregated correlation scores under a specific threshold (see figure 2.29b). Afterwards, the authors classify the visual hull voxels into three groups: (1) Type 1: voxels containing a point cloud from stereo; (2) Type 2: the voxels intersecting the line between a point cloud from stereo and the optical center of an image (3) Type 3: all remaining voxels. According to the authors, most of these remaining voxels are located in textureless or occluded areas. Therefore, they extract the point cloud from the voxel of third type (see figure 2.29c. At the end, a fusion between point cloud generated by depth maps and those from the voxel of third type is performed to have a single point cloud like as illustrated in figure 2.29d. The result of the fusion of the two point clouds produced by stereovision and silhouettes is better than those obtained using the two techniques separately as shown in figure 2.29e. However, their acquisition system is the same as for Esteban et al's method [28] and this system provides control environment. One of the challenge of applying this method is to determine the best threshold which works to delete the voxels containing aggregated correlation scores under this threshold.

Recently, Narayan et al. [47] proposed to merge KinectFusion [49] and VH techniques to recover detailed models for challenging objects with major transparencies and/or concavities. Their method consists of the following steps: 1) computing the VH using RGB images; 2) fusing depth maps into a single mesh using a variant of the KinectFusion algorithm [49]; 3) refining depth maps using visual hull. At this step, they aim at constructing a dense cloud whose points lie on the surface of the object and deforming the visual hull towards this cloud. In particular, this dense point cloud will be a subset of the union of

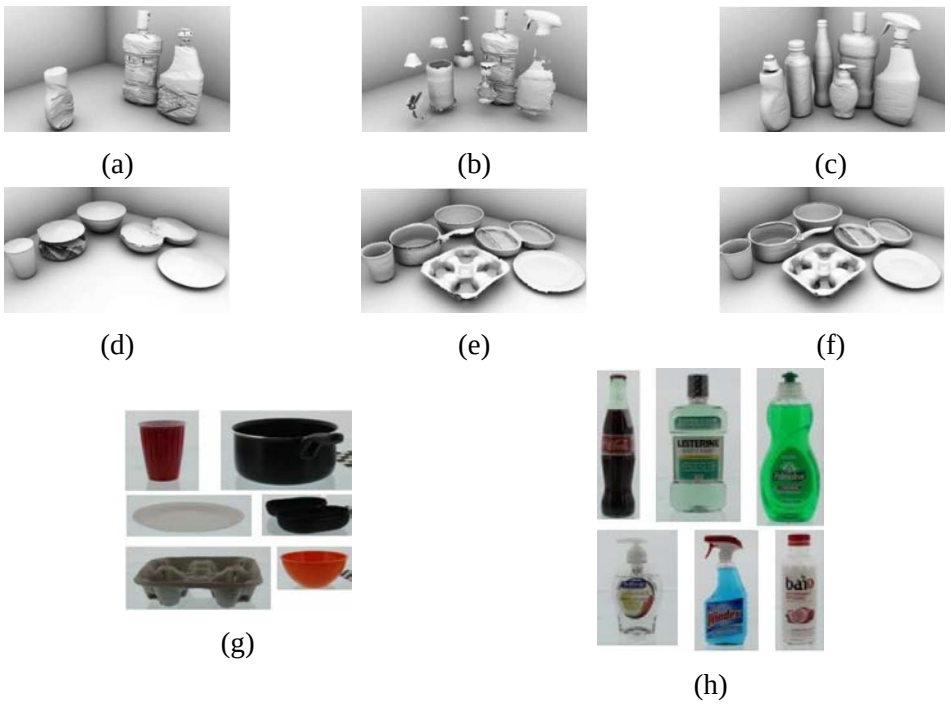


Fig. 2.30 Collections of scanned objects (with transparencies and concavities in first and second row respectively) constructed by visual hull method (f,d), KinectFusion method [49] (b,e), Narayan's method [47] (c,f), g) Color image, concave objects, h) Color image, translucent objects, source [47].

the visual hull and KinectFusion mesh vertices; 4) fusion of the dense refined point cloud with visual hull keeping voxels which verify the following condition: the distance between a voxel and its nearest neighbor of point cloud is less than a specified threshold. Figure 2.30 presents reconstructions of objects with major translucencies or transparencies in first row and objects with concavities in the second row, using VH method, KinectFusion [49], and their approach.

Individually, the KinectFusion algorithm does poorly in reconstructing objects with major transparencies but reconstructs concavities, while the visual hull does poorly in reconstructing concavities but reconstructs regions with major transparencies. Narayan's method recovers the majority of objects including concavities and translucencies zones. However, this method depends on Kinect sensor to apply KinectFusion algorithm and this sensor has a practical limiting range of $(1.2$ to 3.5 m$)$ distance. Within the RECOVER3D project, this sensor does not work efficiently to model actors which evolve at greater distance from the cameras.

2.4 Conclusion

In this chapter, we describe the monocular, binocular, and multiocular shooting geometries for 3D reconstruction purpose. Among these geometries, the multi-simplified epipolar geometry (described in section 2.1.3.2) provides an efficient and robust configuration for 3D object modeling thanks to disparity evaluation instead of rays triangulation. This geometry reduces the corresponding pixel search to one dimension and facilitates the multiple matching process. Our multi-baseline stereovision is based on this multi-simplified epipolar geometry as we will show in the next chapter.

Different techniques for silhouette-based and stereovision-based 3D reconstruction are presented in details. Following our review of the literature, we notice that the 3D reconstruction techniques based on silhouettes are used in multi-camera environments and in real time applications. The main advantages of silhouette-based techniques are robustness and simplicity of implementation. However, the quality of reconstruction from such techniques is limited. While stereovision approaches produce higher resolution, they are more complex and lack computation and robustness. Both stereovision and silhouettes approaches thus complement each other as shown in section 2.3, where different methods have been presented to merge them.

Moreover, we proposed to classify these methods into three majors groups i) Stereovision guided by visual hull methods, ii) Collaborative methods applying simultaneously criteria borrowed from both techniques, iii) Separate application of both methods with further merging of their results. The bibliographical study of scientific literature confirms the advantages and the benefits of hybridizing the two methods for 3D scene reconstruction from multiple views. Within RECOVER3D the project we propose an original scheme for such fusion of visual hull and multi-stereovision thanks to our multi-baseline stereovision framework described in chapter 1 and to the configuration of our acquisition system composed of multiscopic and monoscopic units.

2.5 Résumé: Reconstruction 3D à partir de multiples vues

Dans ce chapitre, nous présentons le concept de la modélisation de la scène 3D à partir de n vues et réalisons un bref état de l'art sur les principales approches existantes. Avant de discuter de la reconstruction 3D multi-view, il est important de savoir comment les images sont obtenues.

Dans la première partie de ce chapitre, nous revenons sur le modèle sténopé et la géométrie de prise de vue d'une seule caméra. À la suite de cette étude « monoscopique », nous abordons les contraintes géométriques existantes entre plusieurs vues d'une même scène sans apriori sur la disposition des caméras dans un contexte binoculaire puis multi-oculaire.

Dans la deuxième partie de ce chapitre, nous nous replaçons dans le contexte du projet RECOVER3D et étudions spécifiquement les approches de reconstruction basées stéréovision et silhouette en évoquant les différentes méthodes existantes.
La modélisation 3D à partir de deux images de point de vue différent est appelée stéréovision. En général, une méthode de stéréovision est constituée des étapes suivantes : i) calcul des coûts d'appariement, ii) agrégation des coûts, iii) optimisation et calcul des profondeurs, iv) amélioration des disparités. Toutefois, la stéréovision multi-vue est une généralisation de la stéréovision permettant une modélisation 3D à partir de plusieurs images chacune issue d'un vue différent. La stéréovision multi-vue peut être classée en trois groupes principaux : i) méthode basée scène, ii) méthode basée image, iii) méthode basée sur des points caractéristiques.
À la fin de cette partie, nous présentons aussi la reconstruction basées silhouettes. Cette méthode, déjà utilisée chez XDProduction, est l'une des techniques exploitées dans le projet RECOVER3D. Les silhouettes sont représentées par un masque binaire. Ce masque représente la projection des points des objets à reconstruire. La distinction des objets de l'arrière plan de la scène est obtenue par l'exploitation du fond uni du studio chromakey utilisé pour lors de l'acquisition des vues. L'ensemble des pixels blancs du masque forme la silhouette de ces objets. À partir de celles-ci , une estimation de l'enveloppe visuelle est réalisée pour reconstruire les objets. L'enveloppe visuelle d'un objet est l'ensemble des points 3D de l'espace scénique qui se projettent dans toutes les silhouettes représentant l'objet. Il existe deux approches pour réaliser cette estimation : l'approche « surfacique », qui consiste à construire la surface de l'intersection des cônes des silhouettes et l'approche « volumique », qui consiste à identifier par vérification les voxels qui se projettent dans toutes les silhouettes.

Dans le projet nous écartons la première au bénéfice de la deuxième afin de garder le même type de représentation volumique entre nos différentes méthodes de reconstruction. Pour conclure, nous proposons de regrouper ces approches en trois catégories : i) les méthodes de stéréovision guidée par l'enveloppe visuelle; ii) les méthodes collaboratives appliquant simultanément des critères issus de ces deux méthodes; iii) les techniques fusionnant uniquement les résultats après une application séparée de ces deux méthodes. Les méthodes de la première catégorie exploitent les informations issues de l'enveloppe visuelle afin d'améliorer les résultats obtenus par une technique de stéréovision; Tandis que celles de la 2ème catégorie exploitent des méthodes de déformation comme les snakes, en les contraignants avec les informations issues de la reconstruction basée silhouettes et basée stéréovision. Les méthodes de la dernière catégorie reconstruisent indépendamment la scène avec chacune de ces deux approches et fusionnent ensuite uniquement leurs résultats.

Chapter 3

Multi-baseline stereovision framework

In this chapter, we present the multi-baseline stereovision framework that we developed for RECOVER3D. In section 3.1, we describe our proposition and contributions. In section 3.2, we give an overview of the method algorithm and introduce the basic concept of *materiality map* that provides the probability of disparity space 3D samples of lying on actual surface(s). We present in section 3.3 our scene space sampling scheme based on disparity space [60] and its 3D samples that we call *target points*. In sections 3.4, 3.5, and 3.6, we explain the attributes provided for each target point: similarity, confidence, and visibility. In section 3.7, we define the energy function using these attributes. In section 3.8 we describe how to optimize the energy function while in section 3.9 we develop the basic algorithm of the optimization engine. The last step of the method, *i.e.* extraction of the reconstructed surface from the scene space, is presented in section 3.10. Finally, we show and discuss experimental results.

3.1 Introduction

While binocular stereovision enables to estimate depth [60] [37], adding more images leads to more robust and accurate 3D reconstruction thanks to information redundancy like described by Niquin et al. [51]. Unfortunately, the matching process becomes more complex and still lacks some robustness in regions either untextured, regularly textured, and/or totally occluded. Thus, the main difficulties are occlusions, changes in appearance, and ambiguities. The trade-off between easily finding correspondences (which favors camera layouts with narrow baselines) and accuracy (which is more robust in case of a wide baseline) has been alleviated using multi-baseline camera settings as illustrated by Okutomi et al. [52]. Classical solutions for 3D reconstruction from multi-baseline stereovision are image-based methods as we mentioned in the section 2.2.3. They consist in matching algorithms

that aim at finding homologous pixels, in different images, which represent the same 3D point in the scene. While the matching process relies on photo-consistency evaluation, this often fails to handle untextured areas or repeated texture. In this chapter, we describe a novel framework for multi-baseline stereovision exploiting the information redundancy to deal with known problems related to occluded regions. Inputs are multiple images shot or rectified in simplified geometry, which allows a convenient sampling scheme of scene space: the disparity space as described by Scharstein et al. [60].

3.1.1 Contributions

In this chapter, we propose a novel framework for multi-baseline stereovision exploiting the visual information redundancy to deal with known problems related to occluded regions. Our main contributions are to propose and build a new materiality map on the Disparity Space (DS) laid as a 3D array; to optimize this map according to some proposed relevant energy function and finally to use the optimized map to decide where the reconstructed surfaces lie in DS. Instead of uniquely relying on image-space information like most multi-view stereovision methods, we work directly in this discretized scene space. We use visibility reasoning and pixel neighborhood similarity measures in order to optimize a 3D discrete map of materiality yielding precise reconstruction even in semi occluded regions. The materiality map holds, for each 3D sample point, its probability about belonging to the scene surface(s). Traditional multi-stereovision methods that depend only on RGB information have some difficulties to solve the problem of ambiguity occuring in occluded regions. This is the reason why the idea of using the materiality map is important, relying on geometrical information like visibility of 3D point and RGB information in order to optimize materiality map.

3.2 Overview of algorithm

Our framework aims at solving the problem of 3D reconstruction from multiple cameras in equidistant multi-baseline layout that implies fully simplified multi-epipolar geometry as shown in the section 2.1.3.2. Our algorithm works with $n > 2$ images $\mathcal{I}_i(nc, nr)$ as inputs where *nc* and *nr* are respectively the common height and width of the image. In the RECOVER3D project, we use four cameras ($n = 4$) but the proposed framework is designed for a more generic assumption $(n > 2)$. Our approach defines the useful natural scene space as the discrete Disparity Space DS, a set of 3D sample points that we call target points. In DS, a target point is defined by the intersection of plan π_δ with constant integral disparity δ with the ray that goes through a pixel $\mathbf{m}_i = (u_i, v_i)^t$ of any image i as illustrated in Figure 3.2. Hence, each target point may be indexed by a disparity space index $\mathbf{t} \equiv (\mathbf{m}^t, \delta)^t$ giving the index $\mathbf{m}$ of the pixel on which $\mathbf{t}$ projects in a chosen reference image i_{ref} (here, we have

chosen $i_{ref} = 0$) and the integer disparity δ associated to its constant depth plane. We want to mention that **T** refers to a target point in space whereas the **t** vector contains its coordinates in DS. A materiality map μ is defined on DS, as a 3D array containing the likelihood for every sample of its existence on the reconstructed surface. This materiality map allows deriving a visibility function, defined in detail in section 3.6, that answers two questions:

- "is a target point inside the frustum of every image?": this detects semi-occlusion that identifies target points certainly not to be seen in every view because they lie out of some view frustum(s).
- "do two target points lie on the same ray of an image?": this detects total occlusion. The visibility function checks materiality values of each potential occluder, looking for downstream (closer to the camera) material target points on the same ray.

Target points are then given some attributes: a materiality score $\mu[\mathbf{t}] \in [0,1]$, visibility scores $\mathcal{V}_i(\mathbf{t}) \in [0,1]$ for each image i derived from semi-occlusion and occluders materialities (see section 3.6), and pre-computed neighborhood similarity scores $\rho_{ij}(\mathbf{t}) \in [0,1]$ for each image couple (i, j) in a given set.
Figure 3.1 shows our framework pipeline that consists of the following principal different steps:

- **Initialization:** the scene is discretized to yield the effective scene space. The method then computes the similarities (see section 3.4) and confidence (see section 3.5) for each target point as described in section 3.4. It derives the initial materiality scores from those image based information.
- **Materiality map optimization:** After initializing the attributes of target points, optimization of the 3D discrete materiality map is driven by an iterative gradient descent algorithm that minimizes a global energy term (see section 3.7) thanks to an iteration of successive back and forth passes over disparity planes. The gradient of energy function E is computed from far δ_m to near δ_M where δ_m and δ_M refer to constant disparity planes respectively of minimum and maximum integer disparity values. The materiality map is then adjusted and visibilities, are updated for each target point from near to far. The energy function is composed of a "data" term built from similarities, visibilities, and materiality scores of each target point and a "smoothness" term promoting desirable geometrical properties of the solution. This will be developed in section 3.7.
- **Final materiality decision:** once the optimization process reaches a pre-defined criterion (number of passes, cost gain threshold,...), some proposed final materiality decision method is applied to binarize materiality values and thus extract object surfaces (see section 3.10).

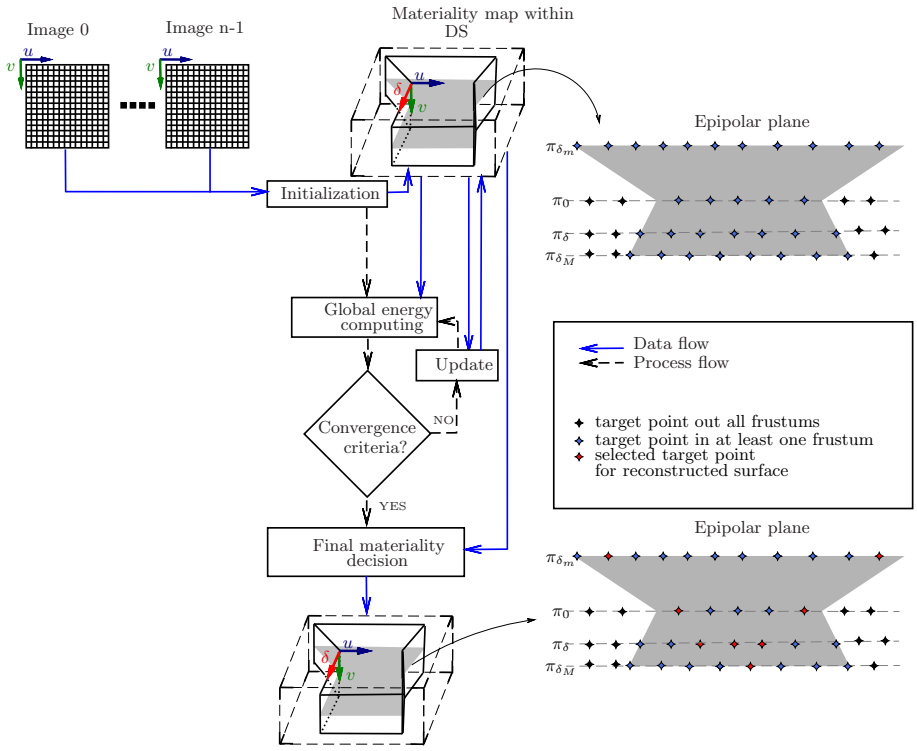


Fig. 3.1 Approach overview: pipeline of the proposed materiality map framework.

3.3 Scene space sampling scheme

Contrarily to image-based approaches, the useful space defined by target points expresses directly the solution domain where the scene can be reconstructed. Thanks to simplified multiscopic geometry, the target points, defined as the intersections between pixel rays from different cameras, lie on several planes of constant depths associated with integer disparities. The target points are inside the union of each camera frustum and are projected in every image frame on integer coordinates points (precisely on a pixel if inside this image frustum) (see figure 3.2). The idea in this chapter is inspired by the proposition of Niquin et al. [51] that aggregates homologous pixels over all images in a structure called *match* which is very closely related to our target points. Let's suppose n images taken from different equidistant viewpoints laid in simplified geometry as described in section 2.1.3.2(e.g., epipolar pairs are horizontal scanlines of same rank, and disparity factors $B_{i,j}$ introduced in the section 2.1.3.2 are of simplest expression $(j-i)$). The visible scene surfaces are supposed to be contained into a limited interval lying between two constant disparity planes of integer disparity values $[\delta_m, \delta_M]$. A target point is defined as the intersection of pixel rays of different images (see figure 3.2). The index **m** in reference image domain $\mathcal{I}_{i_{ref}}$ and the integer disparity δ describe the target point **t** lying at depth fb/δ on the optical ray of image

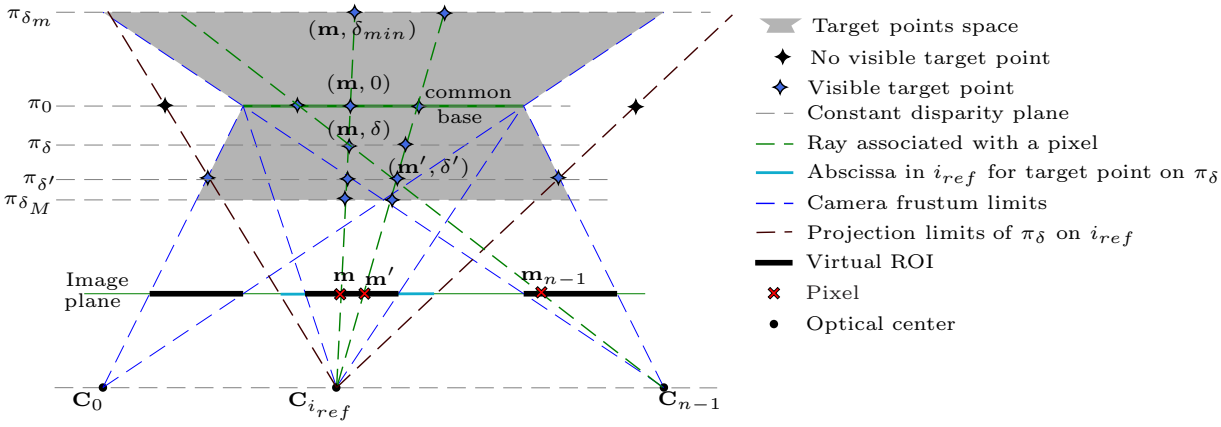


Fig. 3.2 Set of target points in frustum: an efficient discrete reconstruction space within DS.

i_{ref} associated to pixel $\mathbf{m}$. Each target point $\mathbf{t}$ is then identified thanks to the integer disparity δ indexing its constant depth plane and the integer coordinates $\mathbf{m}$ of its projection on the frame of a reference image $\mathcal{I}_{i_{ref}}$. A target point $\mathbf{t}$ projects on the images $\mathcal{I}_i$ and $\mathcal{I}_j$ respectively at $\mathbf{m}_i = (u_i, v_i)^t$ and $\mathbf{m}_j = (u_j, v_j = v_i)^t$. As we described in section 2.1.3.2, knowing that the optical centers are equidistant, the multi-simplifed epipolar geometry provides an efficient way to compute the disparity between two pixels as follows:

$$B_{i,j} = j - i \quad \Leftrightarrow \quad \mathbf{m}_j = \mathbf{m}_i + (i-j)\delta.\mathbf{u}. \tag{3.1}$$

Furthermore, given target points $\mathbf{t} = (\mathbf{m}^t, \delta)^t$ and $\mathbf{t}' = (\mathbf{m}'^t, \delta')^t$ located on the same ray (see figure 3.2) emitted through image i and the projections $\mathbf{m}_i$ of $\mathbf{t}'$, $\mathbf{t}$ in image i, the pixel $\mathbf{m}_i$ can be written as follows:

$$\mathbf{m}_i = \mathbf{m} + (i_{ref} - i)\delta.\mathbf{u} = \mathbf{m}' + (i_{ref} - i)\delta'.\mathbf{u} \Rightarrow \mathbf{m}' = \mathbf{m} + i(\delta' - \delta).\mathbf{u}, \text{ with: } i_{ref} = 0 \tag{3.2}$$

Thank to equation 3.2, we define $\mathbf{h}_i(\mathbf{t}, \delta')$ yielding the target point $\mathbf{t}'$ of disparity δ' using $\mathbf{t}$:

$$\mathbf{h}_i(\mathbf{t}, \delta') \equiv \mathbf{t}' = (\mathbf{m}^t + (\delta' - \delta)i.\mathbf{u}^t, \delta')^t = \mathbf{t} + (\delta' - \delta)\begin{pmatrix} i \\ 0 \\ 1 \end{pmatrix} \tag{3.3}$$

Let's now discuss of the choice of i_{ref} which is not obvious. As left part of equation 3.2 shows, target point $(\mathbf{m}^t, \delta)^t$ projects on image i at $\mathbf{m}_i = \mathbf{m} + (i_{ref} - i)\delta.\mathbf{u}$. In order to keep both target point and pixel coordinates as integer values $(\mathbf{m}^t, \delta)^t \in \mathbb{Z}^3$ and $\mathbf{m}_i \in \mathbb{Z}^2$,

i_{ref} has to remain in $\mathbb{Z}$. Considering actual geometrical properties of the disparity space (DS) in scene space, it should be useful to propose a symmetrical sampling scheme without lateral skew. As DS is built from optical rays of camera i_{ref}, this implies that i_{ref} should be the index of a camera set at the center of the multiscopic unit described in section 1.3.1 ($i_{ref} = (n-1)/2$). These two conditions ($i_{ref} \in \mathbb{Z}$, $i_{ref} = (n-1)/2$) are jointly fulfilled for odd numbers n of cameras. However, in the RECOVER3D project, this number n of cameras is even (four) and the "central camera" is virtual as $(n-1)/2 \notin \mathbb{Z}$. We thus choose $i_{ref} = 0$ for coding efficiency but remain aware of the necessity of switching to more central assumption if geometry becomes crucial. The efficiency of the proposed scene sampling scheme lies in its ability to strictly avoid partially occluded points as samples lie precisely on genuine optical rays associated to image pixels.

3.4 Similarity evaluation

Usually, in scene-based stereovision, the photo-consistency is defined as the similarity of the pixels which represent the projections of a 3D point in the images. Whereas in window matching-based stereovision, for each pixel $\mathbf{m}_i \equiv (u_i, v_i)$ of image $\mathcal{I}_i$, the method chooses its homologue $\mathbf{m}_j \equiv (u_j, v_j = v_i)$ in image $\mathcal{I}_j$ according to aggregated matching scores $\mathcal{AM}(\mathbf{m}_i, \mathbf{m}_j)$ within a same neighborhood $\mathcal{W}$ of both pixels as described in Equation 2.39. We distinguish two groups of matching measure between pixels in binocular stereovision. The first group is based on a similarity function as NCC described in the section 2.2.2.2. The best matching score corresponds to the maximum value of these functions. The second group is based on a dissimilarity function like SSD, SAD described in the section 2.2.2.2. In contrast to the first group, the minimum value of dissimilarity function represents the best matching score.

In this section, we will describe the methodology used to assign unit similarity scores $\rho_{ij}(\mathbf{t}) \in [0,1]$ to a target point $\mathbf{t}$ for various couples of images (i, j) from aggregated matching scores using either similarity function or dissimilarity function. In sections 3.4.1, 3.4.2, and 3.4.3, we will illustrate the general equation and concept to compute those scores. Improvement on similarity measurement is mentioned in sections 3.4.4, 3.4.5, and 3.4.7. At the end of this section, we evaluate different methods for similarity computing in order to integrate the best of them in our framework.

3.4.1 Set of similarity scores for each sample

In order to take into account the occlusion problem in a multi-baseline stereo context, much work has been proposed using similarity/dissimilarity measures based on RGB information. Okutomi et al. [52] propose to use both narrow and wide baselines from a set of cameras placed on a straight line with parallel optical axes. Their matching technique

is based on the idea that global mismatches can be reduced by adding the sum of squared difference (SSD) values from multiple stereo pairs, that is, SSD values are computed first for each pair of stereo images. The resulting SSD functions from all stereo pairs are added together to produce the sum of SSD, which they called SSSD. However, this method fails to deal with semi-occluded objects and it does not take into account the visibility reasoning described in section 3.2. Kang et al. [32] explicitly address occlusion in multi-baseline stereo. For each pixel of the reference view, a subset of the cameras with the best matching scores is selected under the assumption that the pixel may be occluded in the other images. Whereas Niquin et al. [51] proposed to aggregate corresponding pixels over all images in a structure called *match*. Thanks to the definition of these "matches", the similarities of mismatching pixels are not integrated into the energy function used by [51].
While these approaches depend only on photometric matching to handle the occlusion problem, they are sensitive to shooting and lighting settings. As mentioned previously, our framework uses both RGB and geometry information, therefore our method does not depend only on similarity computation to deal with occlusion zones.
Beside, within our framework, the similarity scores are computed for a set r of pairs of images that we propose to choose either as "every image couple" as in equation (3.5) or "consecutive images" as in equation (3.4):

$$r = \{(i, i+1) \mid i \in [0, n-1[\,\}, \tag{3.4}$$

or

$$r = \{(i, j) \mid i \in [0, n-1[, j \in [i+1, n[\,\}. \tag{3.5}$$

We found in experimental results that computing similarity scores $\rho_{ij}(\mathbf{t})$over every pair of images may emphasize ambiguities due to the usual fact that local illumination deviation between images grows with baseline width and, as such, with image indices difference $j-i$. Whereas computing $\rho_{ij}(\mathbf{t})$ over pairs of consecutive images yields more robust results and is thus usually chosen for our experiments.
The set r of image pairs from which similarity scores will be computed being known, the next section describes the similarity measure $\rho_{ij}(\mathbf{t})$ for a target point $\mathbf{t}$ over i and j images.

3.4.2 Generic equation for similarity score

The optimization process that will be described in the section 3.9, uses unit similarity scores $\rho_{ij}(\mathbf{t}) \in [0,1]$ increasing with the likeness of pixels associated to $\mathbf{t}$ in images i and j. Those similarity scores are computed from aggregated matching measures using either similarity or dissimilarity functions. We define a function $\mathcal{S}$ to scale aggregated matching score into unit similarity score. Two kinds of scaling function $\mathcal{S}$ can be exploited depending

on the chosen matching function described in section 2.2.2.2 in order to represent the best scores by the value 1.
The first kind concerns increasing functions (like as $\mathcal{IS}_{\lambda,k}(t) = (1 + tanh(\lambda t))/2$ shown in figure 3.3) used to scale aggregated similarity scores (SSD, SAD,..).
The second consists of decreasing functions scaling aggregated dissimilarity scores (SCC, NCC,...) (like as $\mathcal{DS}_{td,k}(t) = 2^{-(t/td)^k}$ shown in figure 3.4).
Alike for different matching methods, computing unit similarity scores $\rho_{ij}(\mathbf{t})$ implies several choices that reinforce the flexibility of our framework concerning its main components: matching function, aggregation support, normalizing factor, and scaling functions (see equation 3.6).

$$\rho_{ij}(\mathbf{t}) = \mathcal{S}\Big(ms_{ij}^{\mathcal{W}}(\mathbf{t})\Big), \tag{3.6}$$

with:

$$ms_{ij}^{\mathcal{W}}(\mathbf{t}) = \frac{\sum\limits_{v \in \mathcal{W}_{\mathbf{m}_i,\mathbf{m}_j}} \mathcal{MS}(\mathcal{I}_i[\mathbf{m}_i + v], \mathcal{I}_j[\mathbf{m}_j + v])\ w_{ij}(\mathbf{t}, v, \mathcal{W})}{\mathcal{N}(\mathbf{t}, \mathcal{W})},$$

with:

w_{ij}	weight function applied to each neighbor for (i, j) couple
$\mathcal{S}$	scaling function (see figures 3.4, 3.3)
$\mathcal{W}, \mathcal{N}, \mathcal{MS}$	other notations borrowed from equation 2.39
$\forall k \in \{i, j\}\ \ (\mathbf{m}_k, 0)$	$= \mathbf{h}_k(\mathbf{t}, 0) \Leftrightarrow \mathbf{m}_k = \mathbf{m} - k\delta\mathbf{u}$

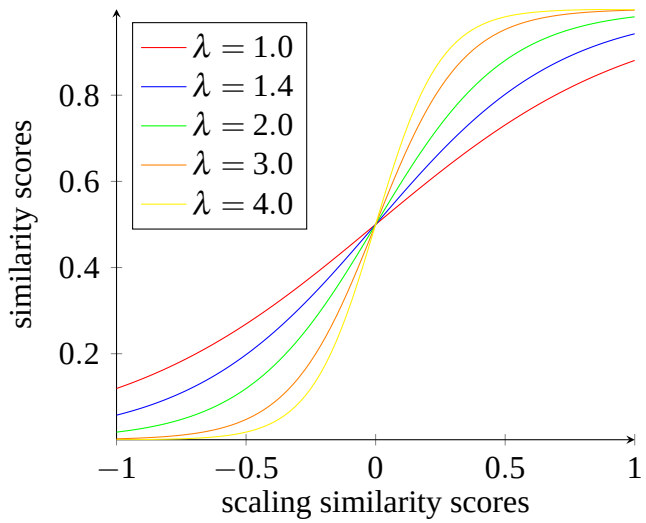


Fig. 3.3 Normalized increasing function $\mathcal{IS}_{\lambda,k}(t) = (1 + tanh(\lambda t))/2$ to scale similarity scores.

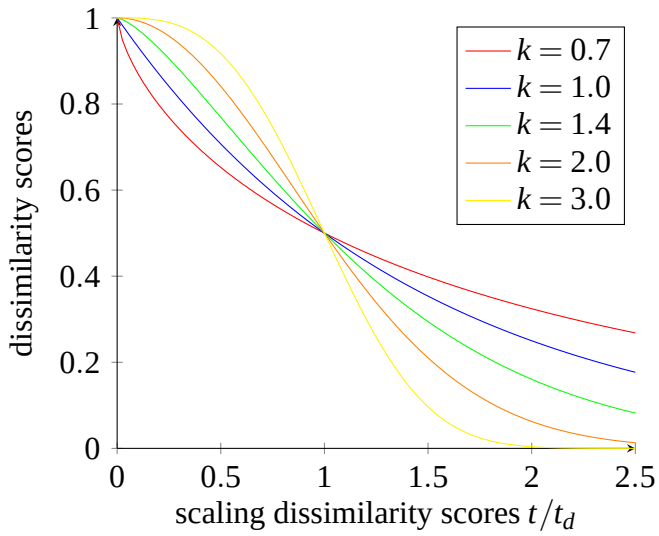


Fig. 3.4 Normalized decreasing function $\mathcal{DS}_{td,k}(t) = 2^{-(t/td)^k}$ to scale dissimilarity scores.

The generic equation 3.6 is used to compute similarity scores. In our method, the matching score $\mathcal{MS}$ can be the squared difference, absolute difference, or multiplication of centered intensity values. Aggregation consists of summing the matching scores over windows $\mathcal{W}$ of size $[\mathcal{W}idth, \mathcal{H}eight]$. The scaling function $\mathcal{S}$ shifts all the values into the range $[0,1]$. In order to use adaptive windows to enhance similarity scores, some weight function w_{ij} is applied on windows $\mathcal{W}$ as we will see in section 3.4.5. Computing unit similarity scores thanks to Equation 3.6 permits to choose various options different types of matching cost as described in section 3.4.3.

3.4.3 Non adaptive flat windows

We firstly propose to compute the similarity scores for a target point **T** in a simple way without considering any improvements to $\mathbf{m}_i$ neighbors selection (reminder: $\mathbf{m}_i$ is the projection of the target point **t** in the image i). As described in section 3.4.1, a target point is given similarity scores ρ_{ij} over pairs of images $(i, j) \in r$ using equation 3.4. Figure 3.5 shows an example of the behavior of similarity function, here applied on scanline 144 using four Tsukuba images (source: web site of Middlebury University to stereovision [61]). We developed our framework in a flexible way to permit to test different options for each of its main components including matching score functions. Using the equation 3.6, similarity scores $\rho_{ij}(\mathbf{t})$ are computed without adaptive support weight for matching score aggregation ($w_{ij}(\mathbf{t}, \mathcal{W}) = 1$). According to the choice of matching score $\mathcal{MS}$, a decreasing or increasing scaling function $\mathcal{S}$ is used as illustrated in table 3.1. The figure 3.5 shows one slice of disparity space containing similarity computation for target points over three pairs of images.

	$w_{ij}(\mathbf{t},\mathcal{W})$	$\mathcal{N}(\mathbf{t},\mathcal{W})$	$\mathcal{MS}(a,b)$	$\mathcal{S}$
SAD	1	$\sum_{v\in\mathcal{W}_{\mathbf{m}_i,\mathbf{m}_j}} 1$	$\lvert a-b\rvert$	(*) $\mathcal{DS}_{td,k}(t)$
SSD	1	$\sum_{v\in\mathcal{W}_{\mathbf{m}_i,\mathbf{m}_j}} 1$	$(a-b)^2$	(*) $\mathcal{DS}_{td,k}(t)$
NCC	1	$\left(\Pi_k\in\{i,j\} \sum_{v\in\mathcal{W}_{\mathbf{m}_i,\mathbf{m}_j}} (\mathcal{I}_k[\mathbf{m}_k+v]-a_k)^2\right)^{\frac{1}{2}}$ with $a_k = mean_{v\in W_{\mathbf{m}_i,\mathbf{m}_j}} \mathcal{I}_k[\mathbf{m}_k+v]$	$a*b$	(**) $\mathcal{IS}_{\lambda,k}(t)$

(*) $\mathcal{DS}_{td,k}(t) = 2^{-(t/td)^k} \equiv e^{(-\lambda t^k)}, k = 1,2,\ldots \lambda \equiv \frac{\log 2}{td^k}$
(**) $\mathcal{IS}_{\lambda,k}(t) = (1+tanh(\lambda t))/2, \lambda = 1,2,\ldots$
Table 3.1 SAD, SAD, or NCC definition by components of the function described in the equation 3.6.

We refer to ρ_{01}, ρ_{12}, and ρ_{23} by the red, green, and blue colors respectively. Therefore, the target point with white color means that it is seen by the three pairs of images with similarity scores higher than 0. We note that computing similarity using SSD and SAD provides an initial description of the reconstructed surface more robust and reliable than when using NCC (white zone in the figure 3.5). The final scores of similarity function will be between [0,1], where 1 represents the best score. We will call in the next sections the results of similarity evaluation on target point **t** as similarities attributes regardless of whether they have been calculated using similarity or dissimilarity functions. We can notice that the way of computing the similarity described in this section is the same as a traditional multi-baseline method. However, our framework uses these similarity attributes $\rho_{ij}(\mathbf{t})$ in a scene-based rather than image-based method, in order to improve and refine the materiality map as we will see in the next sections. In addition, we propose to optimize the similarity computation by using both the adapted window concept (see sections 3.4.4 and 3.4.5) and a post-processing correction approach (see section 3.4.7).

3.4.4 Separate windows

One common issue in dissimilarity/similarity evaluation consists of tackling image areas with large depth gaps. The origin of this problem is that the window matching considers the whole neighborhood into account while it may contain pixels of different depth. Kang et al. [32] propose to work with shiftable windows. The basic idea of shiftable windows is to keep the best matching score among several windows that include the pixel of interest instead of only the usual one centered at that pixel.This approach can improve the matching of foreground objects near depth discontinuities. In our framework, we use this idea and propose to separate the matching windows into two sub-windows (Left (L), Right (R)), or

(a)

(b) SSD

(c) SAD

(d) NCC

Fig. 3.5 Sample of slice through a 3D disparity space: a) Original Tsukuba image with highlight on scanline 144 (source: [61]). b,c,d) similarity scores for epipolar plane 144 using four Tsukuba images with disparity range [0,21] using SSD, SAD, and NCC respectively with centered window of size of 13x9. Red, green, and blue colors represent respectively similarities computing $\rho_{0,1}(\mathbf{t})$, $\rho_{1,2}(\mathbf{t})$, and $\rho_{2,3}(\mathbf{t})$ over pairs of images $(0,1)$, $(1,2)$, and $(2,3)$.

four sub-windows (Up-Right (*UR*), Up-Left (*UL*), Down-Right (*DR*), and Up-Left (*UL*)) according to the centered pixel (see figure 3.6) that we call (SP2) and (SP4) respectively. The maximum similarity computed from those sub-windows is then chosen as the target point similarity $\rho_{ij}(\mathbf{t})$.

$$\text{For SP2 method:} \quad \rho_{ij}(\mathbf{t}) = \max(\mathcal{S}(ms_{ij}^{\mathcal{W}_L}(\mathbf{t})), \mathcal{S}(ms_{ij}^{\mathcal{W}_R}(\mathbf{t}))) \tag{3.7}$$

with:

$$\begin{aligned} &\mathcal{W}_{\mathcal{L}}, \mathcal{W}_{\mathcal{R}} \quad \text{of size} \quad \frac{\mathcal{W}idth}{2} + 1 \times \mathcal{H}eight \\ &\text{For SP4 method:} \quad \rho_{ij}(\mathbf{t}) = \max(\mathcal{S}(ms^{\mathcal{W}_{UL}}(\mathbf{t})), \mathcal{S}(ms_{ij}^{\mathcal{W}_{DR}}(\mathbf{t})), \\ &\qquad\qquad \mathcal{S}(ms_{ij}^{\mathcal{W}_{UR}}(\mathbf{t})), \mathcal{S}(ms_{ij}^{\mathcal{W}_{DL}}(\mathbf{t}))), \end{aligned} \tag{3.8}$$

with:

$$\mathcal{W}_{LU}, \mathcal{W}_{LD}, W_{RU}, \mathcal{W}_{RD} \quad \text{rectangle windows of size} \quad \frac{\mathcal{W}idth}{2} + 1 \times \frac{\mathcal{H}eight}{2} + 1$$

Figure 3.7 shows the similarity behavior computation for the scanline 89 of a set of four Tsukuba images (source: web site of Middlebury University to stereovision [61]) with and without separate windows. In this figure, we have chosen the target point **T** indexed by $(\mathbf{m}(139,89)^t, \delta) : \delta \in [0,21]$ where the projection $\mathbf{m} = (u = 139, v = 89)^t$ with respect the reference image is located at the edge of blue box object as shown in figure 3.7a. The matching window $\mathcal{W}$ around **m** contains then pixels with different depths. The figure 3.7a shows that the best similarity target points is calculated at the disparity $\delta = 5$ for the pixel **m**. According to the ground truth, the actual 3D point corresponding to pixel **m** lies at disparity $\delta = 6$ instead of $\delta = 5$. Figure 3.7b shows the advantage of using separate windows. The best similarity for the pixel **m** is computed at the disparity $\delta = 6$. That is to say, **m** now has a correct disparity.

3.4.5 Weighted windows

In contrast to the idea that the global stereovision methods, such as graph cuts [39] or belief propagation [70], always provide better results than local methods, Yoon and Kweon proposed an alternative one called adaptive support weight [83]. Their approach is classified as a local method described in section 2.2.2.3 and yields results close to those of global methods as discussed in section 2.2.2. The effectiveness of their technique is due to the aggregation of large support-window sizes and to neighbors weight that adapts according to similarity and distance to the central pixel in the support window.

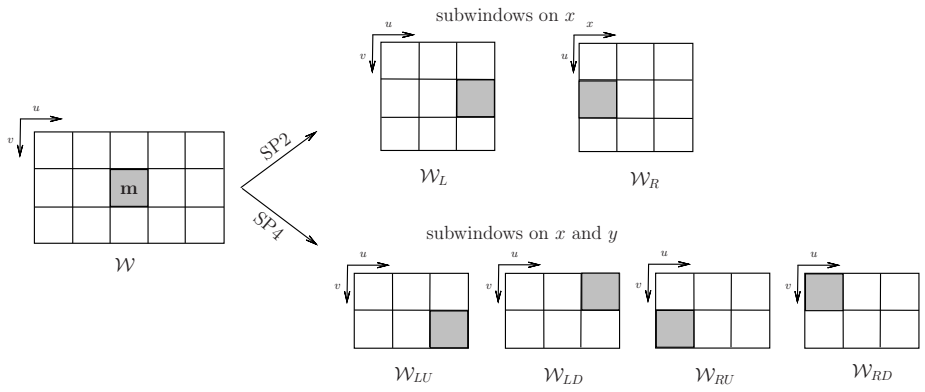


Fig. 3.6 Comparison of centered window and both SP2 and SP4 separate windows approaches.

We rewrite their technique using bilateral filtering described in [54] for an image processing technique smoothing images while preserving edges. It can be based on the Gaussian kernel. Contrary to Gaussian filtering, the bilateral filtering depends not only on the euclidean distance of pixels, but also on their color distance. This method handles depth differences through the assumption that close pixels of similar colors should have high probabilities to belong to the same object. It is thus to weight neighbor values according to the product of Gaussian of their spatial and colors distances to the studied pixel. One should note that, while spatial distance is common to all images, result of color distance computing may be modified over several images. We chose to compute the color weight using a Gaussian function applied to color distance of neighbor and central pixel. This color weight is computed for both implied images (i, j) and we keep the maximum value in order to penalize neighbors in matching window with low similarity to central pixel in one image and high similarity in the other image.

Like separate window method, the weighted window approach based on bilateral filtering is able to assign the correct disparity to the pixel $\mathbf{m} = (139, 89)^t$ as illustrated in figure 3.7. We can integrate the weight based on bilateral filtering into the generic equation for similarities measurement with the following weight using non normalized gaussian functions without

any overall error according to the global normalizing factor $\mathcal{N}$:

$$\begin{aligned} w(\mathbf{t},v,\mathcal{W}) &= \mathcal{G}_{\sigma_e}(v) * max_{k\in\{i,j\}}\mathcal{G}_{\sigma_s}(\|\mathcal{I}_k[\mathbf{m}_k+v]-\mathcal{I}_k[\mathbf{m}_k]\|) \\ \mathcal{N}(\mathbf{t},\mathcal{W}) &= \sum_v w(\mathbf{t},v,\mathcal{W}) \\ \text{With: } \mathcal{G}_\sigma(\mathrm{x}) &= exp(-\frac{x^2}{2\sigma^2}) \\ \sigma_e,\ \sigma_s &: \text{respectively, color and spatial standard deviations.} \end{aligned} \tag{3.9}$$

3.4.6 Evaluation and choice

To evaluate the different methods for similarity computing, the web site of Middlebury University to stereovision [61] provides different datasets (Tsukuba, Teddy, Cones,....) identifying the discontinuities regions where there is a sudden change in the depth between objects (see figure 3.10). We run different methods of similarity computing (non adaptive flat windows (NAFW), separate windows (SP2), separate windows (SP4), and weighted windows (WW)) over three datasets. We compare the results against the ground truth (see Table 3.2) considering only the discontinuity regions and the following measures proposed in [60]:

- The Root-Mean-Squared (RMS) error is computed between the disparity map $\mathcal{D}_i$ and its ground truth $\mathcal{D}_i^{gt}$ by the following formula:

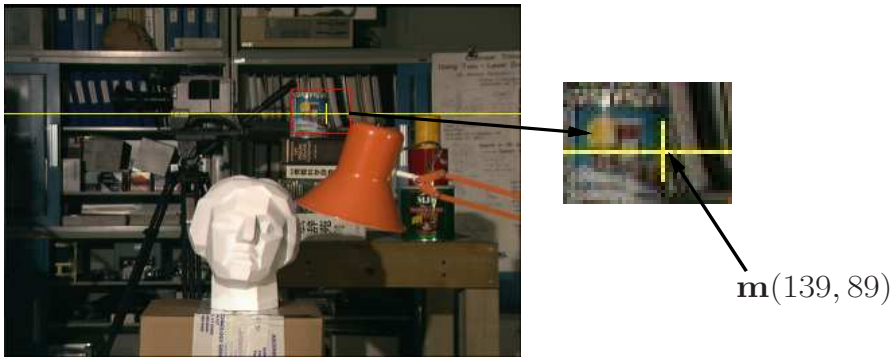
$$RMS = \left(\frac{1}{N}\sum_{\mathbf{m}}(|\mathcal{D}_i(\mathbf{m})-\mathcal{D}_i^{gt}(\mathbf{m})|)^2\right)^{\frac{1}{2}}, \tag{3.10}$$

  where N is the total number of pixels in the image.
- The Percentage of Bad Matching (PBM) pixels provides the percentage of mismatching pixels between two disparity maps using the following formula:

$$PBM = \frac{1}{N}card\{\mathbf{m} \ \big|\ |\mathcal{D}_i(\mathbf{m})-\mathcal{D}_i^{gt}(\mathbf{m})| > \delta_{threshold}\big|\}, \tag{3.11}$$

  where N is the total number of pixels in an image, and $\delta_{threshold}$ is the threshold for evaluating bad matched pixels (usually $\delta_{threshold}$ = 1.0).

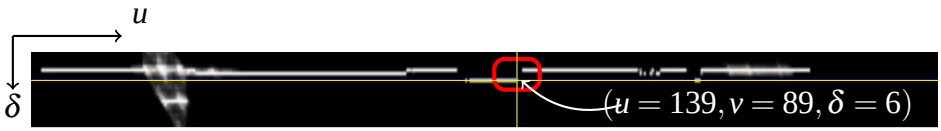
Table 3.2 compares each discussed approach for window matching using RMS error and PBM measures on three datasets: Tsukuba, Teddy, and Cones. In order to facilitate the interpretation, we propose in the table 3.3 to normalize the values derived from RMS and PBM measures for each dataset dividing on the minimum values through different windowing approaches. The figures 3.8 and 3.9 illustrate these normalized values. We can note



(a)

(b) SSD

(c) SP4



(d) *WW*

Fig. 3.7 Evaluation of different similarity methods using only one image pair (0,1) to facilitate the visual comparison: a) Original Tsukuba image with highlight on scanline 89 source:[61], b, c, d) similarity score $\rho_{0,1}$ for the image pair (0,1) using SSD, SP4, and *WW* methods with a disparity range of [0,21] and a matching windows size of 13x9.

	Tsukuba		Teddy		Cones	
	RMS	PBM	RMS	PBM	RMS	PBM
NAFW	44.9889	0.6025	17.9101	0.9128	21.8976	0.7766
SP2	43.8937	0.4273	16.8235	0.9056	19.4742	0.7687
SP4	38.1254	0.3124	15.6237	0.9032	18.8269	0.7719
WW^1	38.0951	0.3437	51.3073	0.9690	18.4480	0.7847
WW^2	44.3099	0.5528	17.9802	0.9146	22.7746	0.7778

Table 3.2 Comparing grounds truth against Tsukuba, Teddy, and Cones in discontinuity regions using different methods: Non Adaptive Flat Windows (NAFW), Separate Windows (L, R) (SP2), Separate Windows (LU, LD, RU, RD) (SP4), Weighted Windows(1) (WW^1) with $\sigma_s = 3$ and $\sigma_e = 0.05$, and Weighted Windows(2) (WW^2) with $\sigma_s = 6$ and $\sigma_e = 0.2$. The results are obtained using a matching window size of 13x9 for each dataset.

	Tsukuba		Teddy		Cones	
	RMS	PBM	RMS	PBM	RMS	PBM
NAFW	1,1809	1,928	1,1463	1,0106	1,1869	1,0102
SP2	1,1522	1,3677	1,0767	1,0026	1,0556	1.0
SP4	1,0007	1.0	1.0	1.0	1,0205	1.0004
WW^1	1	1.1001	3.2839	1.0728	1.0	1.0208
WW^2	1.1631	1.7695	1.1508	1.0126	1.2345	1.0118

Table 3.3 Normalized results computed from Table 3.2 for each dataset and for each measure dividing by the minimum value for the different methods on this dataset.

that the SP4 method provides better results than other approaches over the three datasets. Furthermore, by comparing the results of WW^1 and WW^2, we note that one of the problems of WW method is the choice of the best values of σ_s and σ_e over different datasets. For this reason, the SP4 method is used in our framework to compute the similarities for each target point.

3.4.7 Similarity correction

The goal of the similarity correction is to provide more reliable similarity scores. Figure 3.11 shows the similarity scores illustrated in figure 3.5 with and without correction for the scanline 144 of Tsukuba images. We notice that, using a scaling function $\mathcal{S}$ described in Table 3.1 with low value of td (first column), reduces similarities both in mismatching ambiguities areas (red rectangle in figure 3.11) and in some trust zones (green rectangle in second column of figure 3.11). Therefore, we propose the similarity correction concept in order to try and penalize mismatching similarities while maintaining the other zones as shown in yellow rectangle in the third column of figure 3.11.

To realize this correction, we propose to check out each similarity score ρ_{ij} for each target

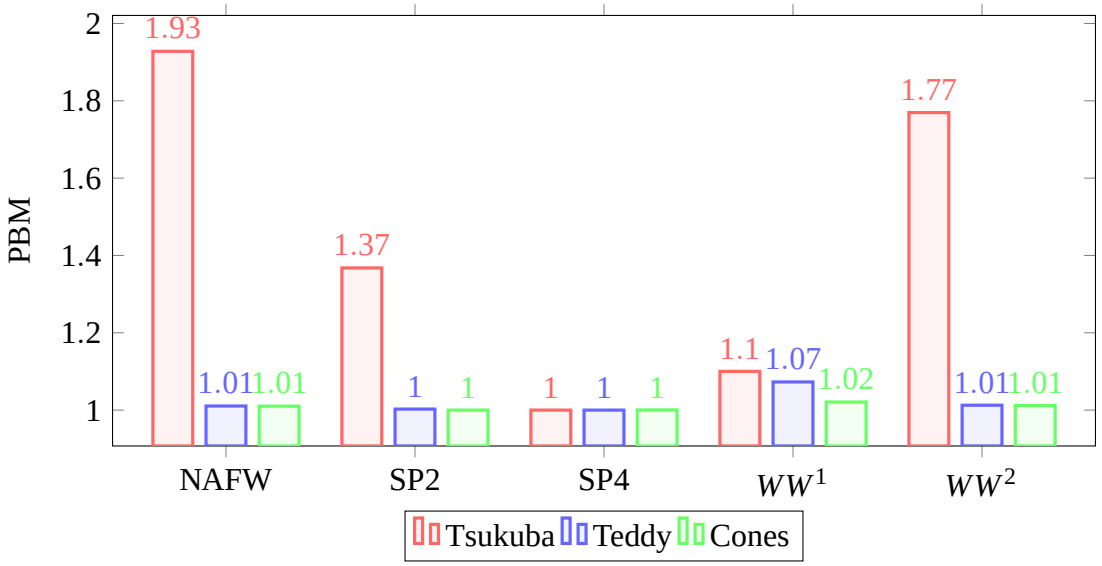


Fig. 3.8 Graph flowing normalized results derived from Table 3.3 applying PBM.

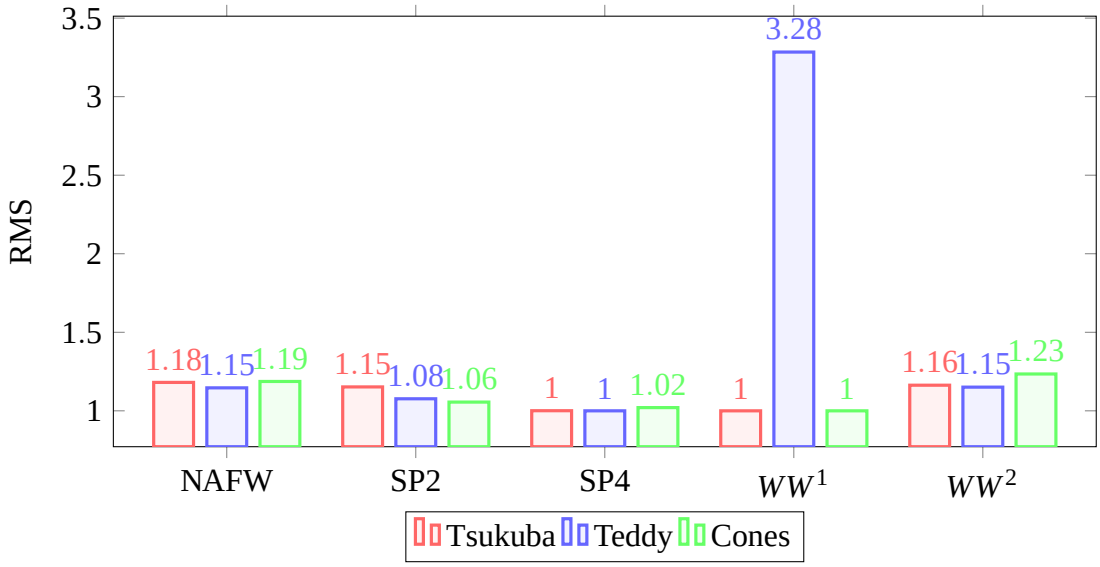
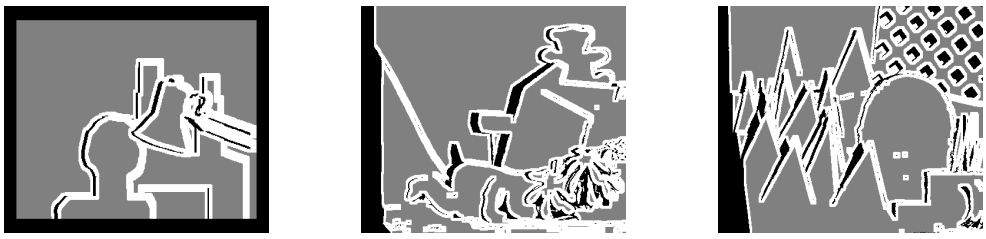


Fig. 3.9 Graph flowing normalized results derived from Table 3.3 applying RMS error.



(a) (b) (c)

Fig. 3.10 a, b, c) Regions near depth discontinuities, occluded and border, and other regions are indicated respectively by white, black, and gray color for Tsukuba, Teddy, and Cones datasets, source:[61].

point $\mathbf{t}$ according to best scores over rays of both i and j images passing through $\mathbf{t}$. If $\rho_{ij}(\mathbf{t})$ represents the maximum similarity score on one of these rays, the target point $\mathbf{t}$ then has high probability to be one of the reconstructed points. We thus normalize the similarity scores $\rho_{ij}(\mathbf{t})$ according the lower of the maximum of similarity scores $\rho_{ij}(\mathbf{t}')$ over respectively i an j rays passing through $\mathbf{t}$ ($\mathbf{t}' \in ray_k(\mathbf{t})$, $k \in i, j$). In order to carry out the similarity correction for the pair of images $\mathcal{I}_i$ and $\mathcal{I}_j$, we first check the maximum similarity values through $ray_i(\mathbf{t})$ and $ray_j(\mathbf{t})$. The minimum of the two maximum values is then used to normalize $\rho_{ij}(\mathbf{t})$. The whole correction yielding the normalized similarity score $\bar{\rho_{ij}}(\mathbf{t})$ is expressed as follows:

$$\bar{\rho_{ij}}(\mathbf{t}) = \frac{\rho_{ij}(\mathbf{t})}{min_{k\in\{i,j\}}\left(max_{\delta'}(\rho_{ij}\left(\mathbf{h}_k(\mathbf{t},\delta')\right))\right)} \tag{3.12}$$

The results in Figure 3.11 show the importance of this approach to refine the similarity scores and reduce the ambiguities similarities zones.

3.5 Confidence

One acknowledged problem in stereovision is associated to textureless regions. Pixels in a textureless zone are subject to bad matches in many stereo matching methods. This section will explain how we evaluate if the matching window is textured or not at similarity computation step and how we quantify this evaluation in the confidence score in order to reduce the influence of textureless zones in the materiality map optimization.
When the similarity score is computed through the matching windows, the variances $var^{\mathcal{W}_{\mathbf{m}_i,\mathbf{m}_j}}(\mathcal{I}_k,\mathbf{m}_k)$ of each of these windows are computed with:

$$\begin{aligned} var^{\mathcal{W}}(\mathcal{I},\mathbf{m}) &= mean_{v\in\mathcal{W}}\left(\mathcal{I}[\mathbf{m}+v] - moy^{\mathcal{W}}(\mathcal{I},\mathbf{m})\right)^2, \\ \text{with: } moy^{\mathcal{W}}(\mathcal{I},\mathbf{m}) &= mean_{v\in\mathcal{W}}\left(\mathcal{I}[\mathbf{m}+v]\right). \end{aligned} \tag{3.13}$$

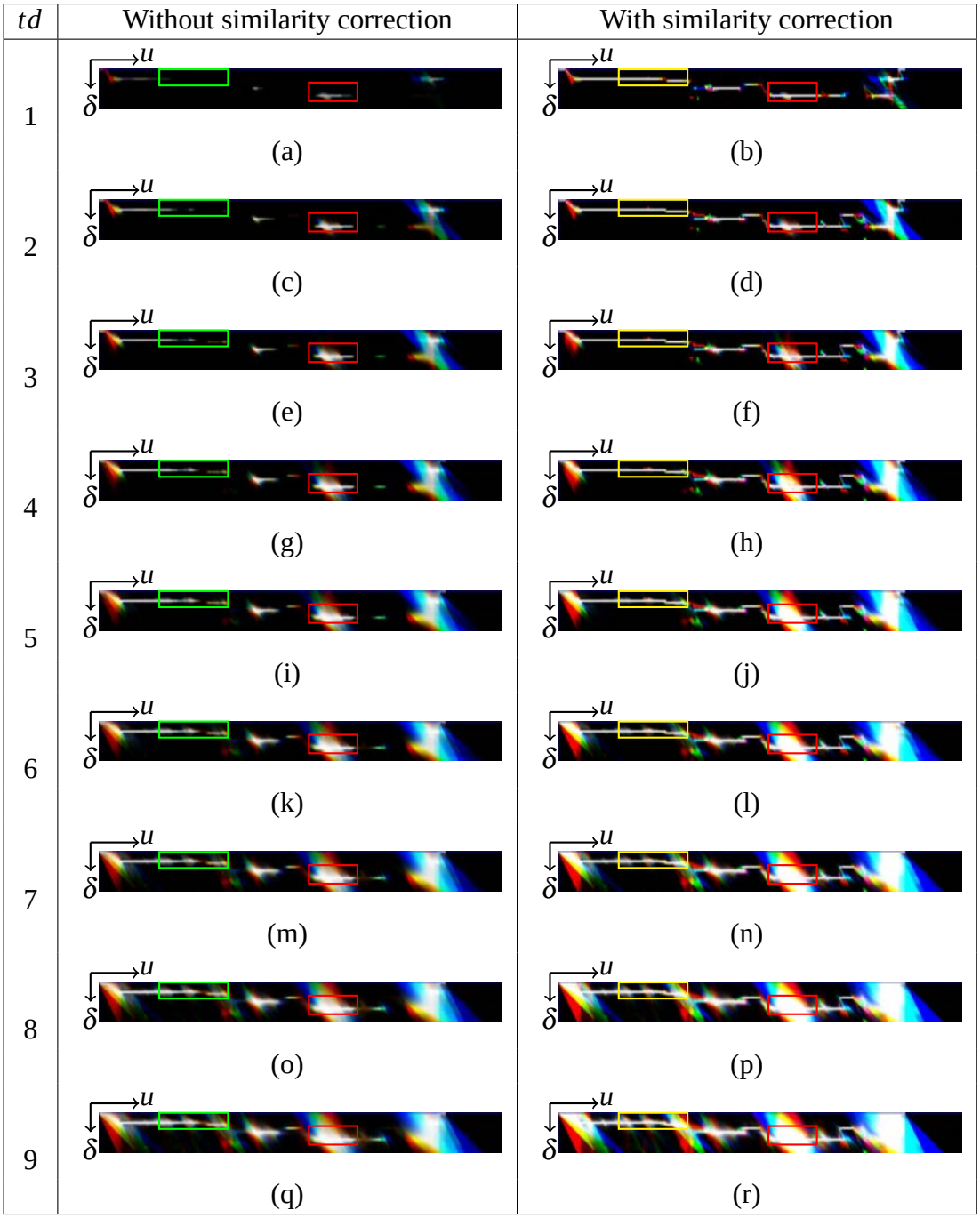
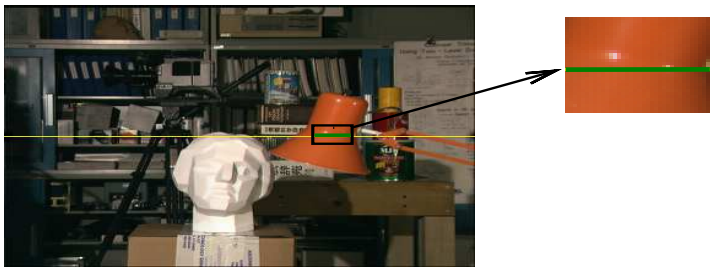


Fig. 3.11 Sample slice through a 3D disparity space: similarity scores with and without similarity correction for scanline 144 using four Tsukuba images with a disparity range [0,21] using the scaling function $\mathcal{DS}_{\lambda,k}(t) = 2^{-(t/td)^k}$, $k = 1$; where td is between 1 and 9 according to the first column. The red, green, and blue colors represent respectively ρ_{01}, ρ_{12}, and ρ_{23}.



(a)

(b) Similarity

(c) Confidence

Fig. 3.12 a) Original Tsukuba image with highlight on scanline 144, b) similarity scores for scanline 144 with disparity range [0,21], c) Confidence scores for scanline 144: white points refer to high confidences values.

Our method assigns to each target point a confidence score cnf_{ij} for each pair of images $(i,j) \in r$ (reminder : r is a set of pairs of images and is defined in section 3.4.1) using the variances as described in equation 3.14

$$\begin{aligned} cnf_{ij}(\mathbf{t}) &= \mathcal{S}(varM)\left(\frac{varM}{varm}\right), \\ \text{with } varM &= max_{k\in\{i,j\}} var^{\mathcal{W}}(\mathcal{I}_k, \mathbf{m}_k), \\ varm &= min_{k\in\{i,j\}} var^{\mathcal{W}}(\mathcal{I}_k, \mathbf{m}_k), \\ \mathcal{S}(t) &= (1+tanh(\lambda t))/2, \quad \lambda = 1,2,... \end{aligned} \tag{3.14}$$

Figure 3.12 shows the minimum confidence score ($min_{(i,j)\in r}\ cnf_{ij}(\mathbf{t})$) for the target points for Tsukuba image scanline 144. Green rectangles outline the areas with low confidence. In the zone which represents the lamp object, the confidence score indicates that the target points lie on a textureless zone.

3.6 Visibility function

In this section, we describe the visibility score for each target point thanks to the efficient definition of the scene space geometry described in section 3.3. Our visibility definition is close to the visibility function demonstrated by Szeliski et al. [74], which uses a recursive front-to-back algorithm to build a visibility map. This method targets natural matting, segmenting foreground from background without any special color screen. This visibility function proposed by [74] is re-used by Kang et al. [33] in order to handle occlusions. From a collection of images, this method computes multiple depth maps simultaneously and explicitly models the visibility map. This map is used by an energy function in order to weight the correlation scores. In this section, we adapt the visibility function proposed by [74] to our framework in order to evaluate, for each target point **t**, its visibility scores in each image. This evaluation permits to handle total occlusion and semi-occlusion illustrated in section 3.2. The proposed scene sampling scheme easily answers the two questions asked in section 3.2:

- The question "(a) is a target point inside the frustum of every image?" by verifying if the abscissa of the target point projection on the image plane lies in the scanline domain (its ordinate mandatorily lies in image column domain as target points are located on genuine epipolar planes).
- The question "(b) do two target points lie on the same ray of an image? " by taking into account the materiality of each downstream target point (with higher disparity) on the same ray. Downstream target points of **t** in image i (as introduced in section 3.3), are identified as $\mathbf{t}' = \mathbf{h}_i(\mathbf{t}, \delta') : \{\delta' > \delta, (\mathbf{m} = \mathbf{m}' + (\delta - \delta')i.\mathbf{u}, \delta')\}$.

The visibilities $\mathcal{V}_i(\mathbf{m}, \delta_M)$ on the nearest plane ($\delta = \delta_M$) are set to 1 for all the target points in frustum of camera i and 0 for all other target points.
For the visibility term $\mathcal{V}_i(\mathbf{m}, \delta)$ on other disparity planes ($\delta < \delta_M$) , a recursive formula is then defined considering the non-materiality of the downstream target points. That is to say, the visibility $\mathcal{V}_i(\mathbf{m}, \delta) = 0$, if there is a target point $\mathbf{t}' = (\mathbf{m}, \delta')$ located in the disparity plane $\delta' > \delta$ and passed through ray $ray_i(\mathbf{t})$ and its materiality $\mu[\mathbf{t}] = 1$. The visibility function of [74] can be expressed in our framework as follows:

$$\begin{aligned} \mathcal{V}_i(\mathbf{m}, \delta_M) =& Fr(\mathbf{m} - i\delta_M.\mathbf{u}) \\ \forall \delta < \delta_M \;\; \mathcal{V}_i(\mathbf{t}) =& \mathcal{V}_i(\mathbf{h}(\mathbf{t}, \delta + 1))(1 - \mu[\mathbf{h}_i(\mathbf{t}, \delta + 1)]) \end{aligned} \tag{3.15}$$

with:

$$Fr(x) = 1 \quad \text{if } x \in \mathbb{W} = [0, nc[\text{ or } 0 \text{ otherwise}$$

The recursive equation 3.15 may be derived as follows:

$$\forall \delta \in [\delta_m, \delta_M] \quad \mathcal{V}_i(\mathbf{t}) = Fr(\mathbf{h}(\mathbf{t},0)).\prod_{\delta'=\delta+1}^{\delta_M}\big(1-\mu[\mathbf{h}(\mathbf{t},\delta')]\big) \tag{3.16}$$

with

$$Fr(x) = \begin{cases} 1 & \text{if } x \in \mathbb{W} = [0, nc[\\ 0 & \text{otherwise} \end{cases}$$

Using this visibility definition, we propose a method to compute the target point visibility scores from far δ_m to near δ_M disparity plans using Equation 3.15 and as illustrated in the Algorithm 4. We call this method "Far-Near method".

Algorithm 4: Visibility computing for a target point $\mathbf{t} = ((u,v),\delta)^t$

if *disparity is max* $(\delta == \delta_M)$ **then**
  foreach *image i* **do**
    if $\boldsymbol{u} - i\delta \in [0, nc[$ **then** $\mathcal{V}_i(\mathbf{t}) = 1$;
    else $\mathcal{V}_i(\mathbf{t}) = 0$;
else
  foreach *image i* **do**
    $\mathcal{V}_i(\mathbf{t}) = (1-\mu[\mathbf{t}']) * \mathcal{V}_i(\mathbf{t}') \qquad \mathbf{t}' = \mathbf{t} + \begin{pmatrix} i \\ 0 \\ 1 \end{pmatrix} = \mathbf{h}_i(\mathbf{t}, \delta+1)$

3.7 Energy function

The optimization process relies upon an energy function defined on the materiality map μ using, classically, two terms illustrated in equation 3.17. The first term "data" penalizes inconsistencies between current solution and actual data (images). The second term "properties" penalizes undesirable properties of the current map. In our implementation, this term is composed of density and thickness energies. The energy function $\mathbf{E}_{global}(\mu)$ is written as follows:

$$\mathbf{E}_{global}(\mu) = \mathbf{E}_{data}(\mu) + \mathbf{E}_{prop}(\mu) \tag{3.17}$$

The data term $\mathbf{E}_{data}(\mu)$ compares image and scene geometry content of the current solution.

$$\mathbf{E}_{data}(\mu) = \sum_{\mathbf{t}} \mathcal{C}(\mu[\mathbf{t}], \mathcal{V}(\mathbf{t}), \rho(\mathbf{t})) \tag{3.18}$$

with:

$$\mathcal{C}(\mu, \mathcal{V}, \rho) = \sum_{(i,j)\in r} \mathcal{T}\left(\mathcal{V}_i \mathcal{V}_j \mu, \rho_{ij}\right) \tag{3.19}$$

The data term sums for each target point some cost function increasing for each pair of images $(i, j) \in r$ with the inconsistency between the target point materiality and visibilities on one side and its similarity scores on the other side. The underlying idea is that high similarity scores for a target point should be explained, at least in textured areas, by high materiality and high visibilities in the implied images. As every implied score is normalized, this term described in equation 3.18 penalizes the inconsistency between similarity scores and product of materiality by related visibilities using some penalty function $\mathcal{T}(a,b)$.

The penalty function $\mathcal{T}(a,b)$ penalizes the discrepancy between a and b. According to the equation 3.19, b is considered as the reference value indicating whether a should be high or low. This penalty can be expressed as the squared difference $\mathcal{L}_2(a,b)$ as follows:

$$\mathcal{L}_2(a,b) = (a-b)^2 \tag{3.20}$$

However, such function restrains the saturation of a if b is not saturated, as illustrated in Figure 3.14, which is not convenient. Another "anti-correlation" function like as $\mathcal{AC}(a,b)$ can be considered as the penalty function $\mathcal{T}(a,b)$ and is written as follows:

$$\mathcal{AC}(a,b) = -(a-\frac{1}{2})(b-\frac{1}{2}) \tag{3.21}$$

This function always pulls a towards saturation. However, $\mathcal{AC}(a,b)$ will also tend to saturate a even for close to average b values as illustrated in Figure 3.13. Therefore, we propose to apply a function combining the advantages of the precedent ones $\mathcal{AC}(a,b)$ and $\mathcal{L}_2(a,b)$. This function is considered as a weighted sum of $\mathcal{AC}(a,b)$ and $\mathcal{L}_2(a,b)$ as illustrated in Equation 3.22 and Figure 3.15, where ϕ and ψ are the weights.

$$\mathcal{T}_{\phi,\psi}(a,b) = \phi\mathcal{L}_2(a,b) + \psi\mathcal{AC}(a,b) = \phi(a-b)^2 - \psi(a-\frac{1}{2})(b-\frac{1}{2}) \tag{3.22}$$

Therefore, the equation 3.19 can be written using 3.22 as follows:

$$\mathcal{C}(\mu,\mathcal{V},\rho) = \sum_{(i,j)\in r} \phi.\left(\mathcal{V}_i\mathcal{V}_j\mu - \rho_{ij}\right)^2 - 2\psi.\left(\mathcal{V}_i\mathcal{V}_j\mu - \frac{1}{2}\right)\left(\rho_{ij} - \frac{1}{2}\right) \tag{3.23}$$

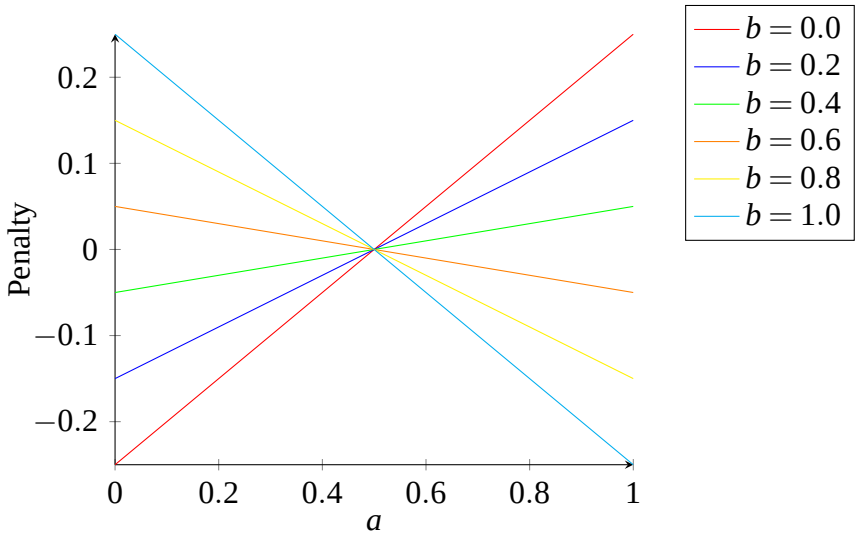


Fig. 3.13 Behavior of penalty function $\mathcal{AC}(a,b)$ according to variable a for various reference values b.

The properties term $\mathbf{E}_{prop}(\mu)$ described in the equation 3.17 is a weighted sum of two terms:

$$E_{prop}(\mu) = \alpha E_{density}(\mu) + \beta E_{thickness}(\mu), \quad \text{where } \alpha, \beta \text{ are weighting factors.} \tag{3.24}$$

- $E_{density}(\mu)$ tends to align the overall sum of materiality scores with the average number of target points in one constant disparity plane, approximately corresponding to the reconstruction of one coherent frontal surface of the scene in the cameras frustums (see Equation 3.25). This global cost is spread uniformly on each target point for gradient computation. The $E_{density}(\mu)$ is written as follows:

$$\mathbf{E}_{density}(\mu) = \frac{\left(\sum_{\mathbf{t}} \mu[\mathbf{t}] - \frac{card(DS)}{\delta_M - \delta_m + 1}\right)^2}{card(DS)}, \tag{3.25}$$

  where card(DS) is the total number of target points in DS.

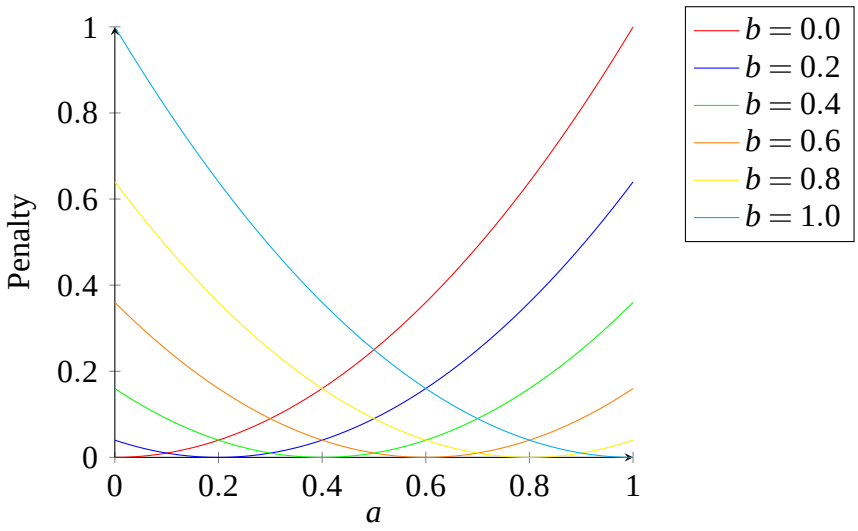


Fig. 3.14 Behavior of penalty function $\mathcal{L}_2(a,b)$ according to variable a for various reference values b.

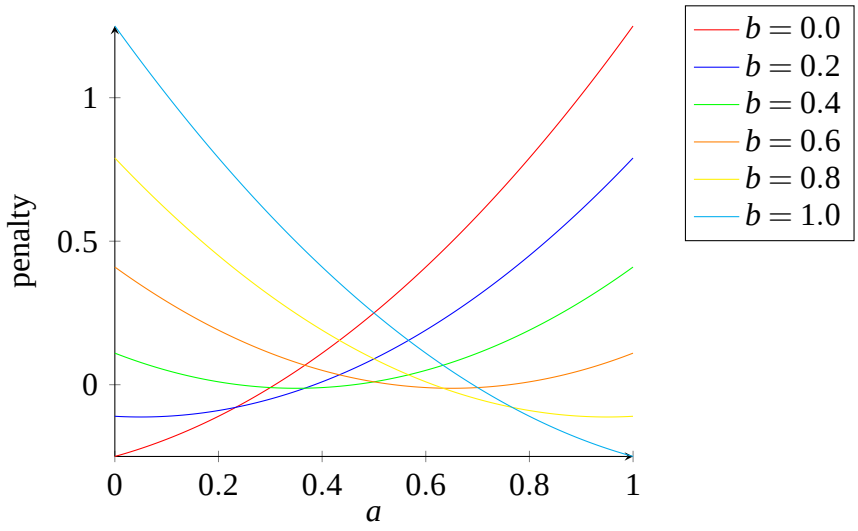


Fig. 3.15 Behavior of penalty function $\mathcal{T}(a,b) = \mathcal{L}_2(a,b) + \mathcal{A}C(a,b)$ according to variable a for various reference values b.

- $E_{thickness}(\mu)$ penalizes thick material areas as the expected solution is a frontal surface that has to remain as thin as possible.The thickness energy leads to the best distribution of materiality through the domain. This energy thus aims to facilitate the final step of our framework "Final materiality decision", which binarizes materiality values, as it helps reducing the number of target points with high materiality that doubtfully belong to the reconstructed surface. The function $E_{thick}(\mu)$ introduced in equation 3.26 can be expressed as:

$$\mathbf{E}_{thickness}(\mu) = \sum_{(\mathbf{t})} \mu[\mathbf{t}] * \underbrace{\|(\bigtriangledown \mu[\mathbf{t}])\|^2}_{\left(\frac{\partial \mu}{\partial \mathbf{u}}\right)^2 + \left(\frac{\partial \mu}{\partial \delta}\right)^2 + \left(\frac{\partial \mu}{\partial \mathbf{v}}\right)^2} \tag{3.26}$$

The set of target points on which the materiality map is defined is both discrete and bordered; thus the materiality values are available on a finite set of target points. Therefore, we choose to work with a discrete approximation of the gradient of the materiality map. The idea is typically to define the derivative components of the local materiality gradient as symmetric finite differences rather than the usual continuous derivatives. Therefore, while the set of target points is finite, the gradient computation of the $E_{thickness}(\mu)$ can be expressed as follows:

$$\bigtriangledown \mu[\mathbf{t}'] = \begin{cases} \frac{\partial \mu}{\partial \mathbf{u}}[\mathbf{t}'] = \frac{\mu[\mathbf{t}'+\mathbf{u}]-\mu[\mathbf{t}'-\mathbf{u}]}{2} \\ \frac{\partial \mu}{\partial \mathbf{v}}[\mathbf{t}'] = \frac{\mu[\mathbf{t}'+\mathbf{v}]-\mu[\mathbf{t}'-\mathbf{v}]}{2} \\ \frac{\partial \mu}{\partial \delta}[\mathbf{t}'] = \frac{\mu[\mathbf{t}'+\mathbf{d}_1]-\mu[\mathbf{t}'-\mathbf{d}_1]+\mu[\mathbf{t}'+\mathbf{d}_2]-\mu[\mathbf{t}'-\mathbf{d}_2]}{4} \end{cases} \tag{3.27}$$

$$\textbf{with: } \mathbf{u} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{v} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{d}_1 = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}, \mathbf{d}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \tag{3.28}$$

In equation 3.28, the derivative with respect to δ implies taking into account our proposition to solve the problem of lateral skew of geometrical properties of Disparity Space (DS) when $i_{ref} = 0$. This would lead to $\frac{\partial \mu}{\partial \delta}[\mathbf{t}'] = \frac{\mu[\mathbf{t}'+d_{1.5}]-\mu[\mathbf{t}'-d_{1.5}]}{2}$ with $d_{1.5} = (1.5 \quad 0 \quad 1)^t$. Unfortunately these two implied points $\mathbf{t}' \pm d_{1.5}$ do not belong to DS. We thus have to evaluate their materiality by interpolation of their nearest neighbors of same disparity. For this reason, we apply the symmetric finite difference over two couples of target points $\mathbf{t}' \pm \mathbf{d}_1$ and, $\mathbf{t}' \pm \mathbf{d}_2$ to compute the derivative with respect to δ according to a central point of view.

We can summarize our energy function as a sum of internal energy (density energy, thickness energy) and external energy (the difference between the information derived from

scene and that obtained from images) depending on unknowns or degrees of freedom which are the materiality scores of the target points. To minimize the total energy, we use the gradient descent method described in the following section.

3.8 Gradient descent based optimization

One of the simplest optimization algorithm for problems expressed as minimum energy of a system with numerous degrees of freedom (*dof*) is gradient descent. The principle is to start from a random point in *dof* domain and to then move a small step in the direction of the steepest slope of the energy according to the *dof*. This steepest slope of the energy is expressed by the energy gradient with respect to the *dof*. The norm of this gradient is the local steepest slope, while its direction in *dof* domain locally maximizes the energy increase rate. By applying a number of iterations, the algorithm converges to a solution which is a local minimum. Starting from a point close enough to optimum, this local minimum will be global one.

In order to apply the gradient descent, we have to compute the energy derivative respective to each *dof*. For the external energy E_{data} defined in section 3.7, we need to achieve the total derivatives of Equation 3.23 with respect to all of the unknowns (the materiality scores). In the algorithm, the materiality and visibilities will change during the process of optimization while the similarities are computed in the initial step and kept constant afterwards.

We will define the total derivatives independently of the choice to the penalty function described in Equation 3.22. This definition will be customized according to the chosen function.

We distinguish between two approaches to explain the derivation for E_{data}, local and global. In naive thinking and without taking into account the efficiency of the scene geometry definition, the local derivative $\partial_{\mu[\mathbf{t}]}E_{data}$ of E_{data} with respect to one materiality score $\mu[\mathbf{t}]$ considers only direct expression of this score in the energy function. As such, its involvement in visibility scores of upstream target points is ignored in local derivative. The global derivative $d_{\mu[\mathbf{t}]}E_{data}$ adds these indirect implications through visibility scores. The local and global derivatives of E_{data} are thus written as follows (in the following equations, we simplify $\mathcal{C}(\mu[\mathbf{t}],\mathcal{V}(\mathbf{t}),\rho(\mathbf{t})) \equiv \mathcal{C}(\mathbf{t})$):

$$\frac{\partial \mathbf{E}_{data}(\mu)}{\partial \mu[\mathbf{t}]} = \sum_{\mathbf{t}'} \frac{\partial \mathcal{C}(\mathbf{t}')}{\partial \mu} \frac{d\mu[\mathbf{t}']}{d\mu[\mathbf{t}]} = \frac{\partial \mathcal{C}(\mathbf{t})}{\partial \mu} \quad \begin{matrix}\text{as materiality scores} \\ \text{are independant unknowns,}\end{matrix} \tag{3.29}$$

$$\frac{d\mathbf{E}_{data}(\mu)}{d\mu[\mathbf{t}]} = \frac{\partial \mathbf{E}_{data}(\mu)}{\partial \mu[\mathbf{t}]} + \sum_i \sum_{\mathbf{t}'} \frac{\partial \mathcal{C}(\mathbf{t}')}{\partial \mathcal{V}_i} \frac{d\mathcal{V}_i(\mathbf{t}')}{d\mu[\mathbf{t}]}. \tag{3.30}$$

The materiality score of $\mathbf{t}$ is involved in visibility $\mathbf{V}_i$ computing for all their upstream target points for image i $\{\mathbf{h}_i(\mathbf{t},\delta')|\delta' < \delta\}$ (see equation 3.16). We propose then to compute the global derivative of E_{data} considering two facts:

- The materiality computation is independent :

$$\frac{d\mu[\mathbf{t}']}{d\mu[\mathbf{t}]} = \begin{cases} 0 & \text{if } \mathbf{t} \neq \mathbf{t}' \\ 1 & \text{if } \mathbf{t} = \mathbf{t}' \end{cases}$$

- $\mathcal{V}_i(\mathbf{t}')$ depends on $\mu[\mathbf{t}]$ only for upstream target points $\mathbf{t}' \in \{\mathbf{h}_i(\mathbf{t},\delta')|\delta' < \delta\}$ of $\mathbf{t}$ on $ray_i(\mathbf{t})$. As such, $\frac{d\mathcal{V}_i(\mathbf{t})}{d\mu[\mathbf{t}]}$ will be zero for all other target points $\mathbf{t}'$.

Using the two facts mentioned above, the global derivatives for E_{data} are described in the equation 3.31.

$$\frac{d\mathbf{E}_{data}(\mu)}{d\mu[\mathbf{t}]} = \frac{\partial \mathcal{C}(\mathbf{t})}{\partial \mu} + \sum_i \sum_{\delta'<\delta} \frac{\partial \mathcal{C}(\mathbf{h}_i(\mathbf{t},\delta'))}{\partial \mathcal{V}_i} \frac{d\mathcal{V}_i(\mathbf{h}_i(\mathbf{t},\delta'))}{d\mu[\mathbf{t}]}. \tag{3.31}$$

According to equations 3.3 and 3.16, the upstream target point visibility may be written as:

$$\mathcal{V}_i(\mathbf{h}_i(\mathbf{t},\delta')) = Fr(\mathbf{h}_i(\mathbf{t},0)) \prod_{\delta''=\delta'+1}^{\delta_M} (1-\mu[\mathbf{h}_i(\mathbf{t},\delta'')]) \tag{3.32}$$

As $\delta' < \delta$, it may be decomposed as:

$$\mathcal{V}_i(\mathbf{h}_i(\mathbf{t},\delta')) = \underbrace{Fr(\mathbf{h}_i(\mathbf{t},0)) \prod_{\delta''=\delta+1}^{\delta_M} (1-\mu[\mathbf{h}_i(\mathbf{t},\delta'')])}_{\mathcal{V}_i(\mathbf{h}_i(\mathbf{t},\delta))} (1-\mu[\mathbf{h}_i(\mathbf{t},\delta)]) \prod_{\delta''=\delta'+1}^{\delta-1} (1-\mu[\mathbf{h}_i(\mathbf{t},\delta'')]) \tag{3.33}$$

As $\mathbf{h}_i(\mathbf{t},\delta) = \mathbf{t}$, the equation 3.33 can be then simplified as follows:

$$\mathcal{V}_i(\mathbf{h}_i(\mathbf{t},\delta')) = \mathcal{V}_i(\mathbf{t})(1-\mu[\mathbf{t}]) \prod_{\delta''=\delta'+1}^{\delta-1} (1-\mu[\mathbf{h}_i(\mathbf{t},\delta'')]) \tag{3.34}$$

The only term depending on $\mu[\mathbf{t}]$ in the above equation is $(1-\mu[\mathbf{t}])$. As such

$$\frac{d\mathcal{V}_i(\mathbf{h}_i(\mathbf{t},\delta'))}{d\mu[\mathbf{t}]} = -\mathcal{V}_i(\mathbf{t}) \prod_{\delta''=\delta'+1}^{\delta-1} \left(1-\mu[\mathbf{h}_i(\mathbf{t},\delta'')]\right) \tag{3.35}$$

Using Equation 3.35, the global derivative of $\mathbf{E}_{data}$(see Equation 3.31) is then written as follows:

$$\frac{d\mathbf{E}_{data}(\mu)}{d\mu[\mathbf{t}]} = \quad \frac{\partial \mathcal{C}(\mathbf{t})}{\partial \mu} \quad - \sum_i \mathcal{V}_i(\mathbf{t}) \sum_{\delta' < \delta} \frac{\partial \mathcal{C}(\mathbf{h}_i(\mathbf{t}, \delta'))}{\partial \mathcal{V}_i} \quad \prod_{\delta''=\delta'+1}^{\delta-1} \big(1 - \mu[\mathbf{h}(\mathbf{t}, \delta'')]\big)$$

$$\frac{d\mathbf{E}_{data}(\mathbf{u})}{d\mu[\mathbf{t}]} = \frac{\partial \mathcal{C}(\mathbf{t})}{\partial \mu} - \sum_i \mathcal{V}_i(\mathbf{t})\ \mathcal{CJ}^{\mathbf{t},i} \tag{3.36}$$

with :

$$\mathcal{CJ}^{\mathbf{t},i} = \begin{cases} 0 & \delta = \delta_m \\ \sum_{\delta' < \delta} \frac{\partial \mathcal{C}(\mathbf{h}(\mathbf{t}, \delta'))}{\partial \mathcal{V}_i} \quad \prod_{\delta''=\delta'+1}^{\delta-1} (1 - \mu[\mathbf{h}_i(\mathbf{t}, \delta'')]) & \delta > \delta_m \end{cases} \tag{3.37}$$

This expression is still complex by the dependence of $\mathcal{CJ}^{\mathbf{t},i}$ on all upstream target points. To facilitate the computation $\mathcal{CJ}^{\mathbf{t},i}$, the relation between the term $\mathcal{CJ}^{\mathbf{t},i}$ and its nearest upstream target point $\mathcal{CJ}^{\mathbf{h}_i(\mathbf{t},\delta-1),i}$ identifies a recursive expression derived in Equation 3.38. To find this mathematical relation, instead of starting the sum operation in the equation 3.37 from $\delta' < \delta$, we begin with $\delta' < \delta - 1$ and extract the value for $\delta' = \delta - 1$. We can then get Equation 3.39:

$$\begin{aligned} \mathcal{CJ}^{\mathbf{t},i} = & \frac{\partial \mathcal{C}(\mathbf{h}_i(\mathbf{t}, \delta - 1))}{\partial \mathcal{V}_i} \\ & + (1 - \mu[\mathbf{h}_i(\mathbf{t}, \delta - 1)]) \underbrace{\sum_{\delta' < \delta - 1} \frac{\partial \mathcal{C}(\mathbf{h}_i(\mathbf{t}, \delta'))}{\partial \mathcal{V}_i} \prod_{\delta''=\delta'+1}^{\delta-2} \big(1 - \mu[\mathbf{h}_i(\mathbf{t}, \delta'')]\big)}_{\mathcal{CJ}^{\mathbf{h}_i(\mathbf{t},\delta-1),i}} \end{aligned} \tag{3.38}$$

$$\mathcal{CJ}^{\mathbf{t},i} = \frac{\partial \mathcal{C}(\mathbf{h}_i(\mathbf{t}, \delta - 1))}{\partial \mathcal{V}_i} + (1 - \mu[\mathbf{h}_i(\mathbf{t}, \delta - 1)]) . \mathcal{CJ}^{\mathbf{h}_i(\mathbf{t},\delta-1),i} \tag{3.39}$$

The equation 3.39 defines the recursive expression between $\mathcal{CJ}^{\mathbf{t},i}$ and its nearest upstream target point $\mathcal{CJ}^{\mathbf{h}_i(\mathbf{t},\delta-1),i}$. The recursive expression helps to find the global derivative efficiently.

We want to remind the reader that the global derivatives defined in Equation 3.36 is generic and does not depend on the choice of the penalty function. To customize this equation according to the penalty function described in Equation 3.23, we compute the partial derivatives with respect to the materiality and visibility as follows:

$$\mathcal{C}(\mu, \mathcal{V}, \rho) = \sum_{(i,j) \in r} \mathcal{T}(\mu \mathcal{V}_i \mathcal{V}_j, \rho_{ij}). \tag{3.40}$$

$$\frac{\partial \mathcal{C}(\mu,\mathcal{V},\rho)}{\partial \mu} = \sum_{(i,j)\in r} \underbrace{\frac{\partial \mathcal{T}}{\partial a}(\mu \mathcal{V}_i \mathcal{V}_j, \rho_{ij})}_{\equiv Da_{ij}(\mu,\mathcal{V},\rho)} \mathcal{V}_i \mathcal{V}_j. \tag{3.41}$$

$$\frac{\partial \mathcal{C}(\mu[t],\mathcal{V}(\mathbf{t}),\rho(\mathbf{t}))}{\partial \mathcal{V}_i} = \mu \sum_{ord(i,j)\in r} \mathcal{V}_j Da_{ij}(\mu,\mathcal{V},\rho), \tag{3.42}$$

with:

$$ord(i,j) = (min(i,j), max(i,j)).$$

Algorithm 5: Computing derivative algorithm of $\mathbf{E}_{data}$ for target point $\mathbf{t}$

Initialize the derivative result $dr = \partial\mu[\mathbf{t}]$ using eq.3.41
if *target point* $\mathbf{t}$ *is located in the farthest disparity plane* δ_m **then**
  foreach *image number i* **do**
    $\mathcal{CI}^{\mathbf{t},i} = 0$
else
  foreach *image number i* **do**
    Access to nearest upstream target point $\mathbf{t}' = \mathbf{h}_i(\mathbf{t}, \delta - 1)$ according to image i
    $\frac{\partial \mathcal{C}(\mathbf{t}')}{\partial \mathcal{V}_i}$
    $dr = dr - \mathcal{V}_i(\mathbf{t})\mathcal{CI}_i(\mathbf{h}_i(\mathbf{t},0))$
    $\mathcal{CI}_i(\mathbf{h}_i(\mathbf{t},0)) = \frac{\partial \mathcal{C}(\mathbf{t}')}{\partial \mathcal{V}_i} + (1-\mu[\mathbf{t}])$

As illustrated in section 3.7 the properties term $\mathbf{E}_{prop}(\mu)$ is composed of two terms $\mathbf{E}_{density}(\mu)$ and $\mathbf{E}_{thickness}(\mu)$. Therefore, to complete the computation of the derivative global energy function $\mathbf{E}_{global}$, we write firstly the derivative for density energy as follows:

$$\frac{d\mathbf{E}_{density}(\mu)}{d\mu[\mathbf{t}]} = 2\frac{1}{card(DS)}\left(\sum_{\mathbf{t}'}\mu[\mathbf{t}'] - \frac{card(DS)}{\delta_M - \delta_m + 1}\right) \tag{3.43}$$

Secondly, the derivative for thickness energy $\mathbf{E}_{thickness}(\mu)$ is written as follows

$$\frac{d\mathbf{E}_{thickness}(\mu)}{d\mu[\mathbf{t}]} = \sum_{\mathbf{t}'}\left(\frac{\partial\mu[\mathbf{t}']}{\partial\mu[\mathbf{t}]}\left(\bigtriangledown\mu[\mathbf{t}']\right)^2 + 2\mu[\mathbf{t}']\frac{\partial \bigtriangledown\mu[\mathbf{t}']}{\partial\mu[\mathbf{t}]}\bigtriangledown\mu[\mathbf{t}']\right) \tag{3.44}$$

$$\text{with } \mathbf{u} = \begin{pmatrix}1\\0\\0\end{pmatrix}, \mathbf{v} = \begin{pmatrix}0\\1\\0\end{pmatrix}, \mathbf{d}_1 = \begin{pmatrix}2\\0\\1\end{pmatrix}, \mathbf{d}_2 = \begin{pmatrix}1\\0\\1\end{pmatrix}.$$

$$\begin{array}{rll}
\frac{d\mathbf{E}_{thickness}(\mu)}{d\mu[\mathbf{t}]} = & (\bigtriangledown\mu[\mathbf{t}])^2 & // \ \mathbf{t}' = \mathbf{t} \\
& +\mu[\mathbf{t}-\mathbf{u}]\frac{\partial\mu[\mathbf{t}-\mathbf{u}]}{\partial\mathbf{u}} & // \ \mathbf{t}' + \mathbf{u} = \mathbf{t} \\
& -\mu[\mathbf{t}+\mathbf{u}]\frac{\partial\mu[\mathbf{t}+\mathbf{u}]}{\partial\mathbf{u}} & // \ \mathbf{t}' - \mathbf{u} = \mathbf{t} \\
& +\mu[\mathbf{t}-\mathbf{v}]\frac{\partial\mu[\mathbf{t}-\mathbf{v}]}{\partial\mathbf{v}} & // \ \mathbf{t}' + \mathbf{v} = \mathbf{t} \\
& -\mu[\mathbf{t}+\mathbf{v}]\frac{\partial\mu[\mathbf{t}+\mathbf{v}]}{\partial\mathbf{v}} & // \ \mathbf{t}' - \mathbf{v} = \mathbf{t} \\
& +\frac{1}{2}\mu[\mathbf{t}-\mathbf{d}_1]\frac{\partial\mu[\mathbf{t}-\mathbf{d}_1]}{\partial\delta} & // \ \mathbf{t}' + \mathbf{d}_1 = \mathbf{t} \\
& -\frac{1}{2}\mu[\mathbf{t}+\mathbf{d}_1]\frac{\partial\mu[\mathbf{t}+\mathbf{d}_1]}{\partial\delta} & // \ \mathbf{t}' - \mathbf{d}_1 = \mathbf{t} \\
& +\frac{1}{2}\mu[\mathbf{t}-\mathbf{d}_2]\frac{\partial\mu[\mathbf{t}-\mathbf{d}_2]}{\partial\delta} & // \ \mathbf{t}' + \mathbf{d}_2 = \mathbf{t} \\
& -\frac{1}{2}\mu[\mathbf{t}+\mathbf{d}_2]\frac{\partial\mu[\mathbf{t}+\mathbf{d}_2]}{\partial\delta} & // \ \mathbf{t}' - \mathbf{d}_2 = \mathbf{t}
\end{array} \tag{3.45}$$

Thanks to the global derivative of $\mathbf{E}_{data}$ which is described in Equation 3.36, we propose to compute the derivative cost for each target point from far to near like described in the algorithm 5. We call this proposition as Far-Near method which sweeps the disparity space from δ_m to δ_M. this method is convenient for $\mathcal{CI}^{\mathbf{t},i}$ computing using Equation 3.39.

Following the visibility reasoning and the global energy derivative, the basic algorithm for computing and updating materiality map using gradient descent is described in section 3.9.

3.9 Basic algorithm of optimization engine

In this section, we describe the core of our algorithm to initialize and optimize the materiality map using gradient descent. The Far-Near method to compute the visibility of the target points is proposed in section 3.6. By contrast, computation of the global derivation for E_{data} is based on Near-Far method as described in section 3.8. Thanks to these methods, we propose a recursive scheme to compute the materiality map independently for each epipolar plan $\mathbf{v}$. In order to apply our optimization method on only the target points that have the possibility to be reconstructed, we determine the limitation on $\mathbf{u}$-axis of DS at each disparity δ as described in Algorithm 6.

In Algorithm 7, the first phase is to compute the similarity and initializes visibility for all the target points. Furthermore we initialize the materiality map by the maximum similarity scores multiplied by the minimum confidence score over all image couples.
The second phase of the algorithm is to optimize the materiality maps using gradient descent by applying a specified number of iterations. We can divide the optimization process into two consecutive major blocks:

- Near-Far method compute the energy function derivative sweeping the disparity plans form δ_m to δ_M.

- Far-Near method to update the materiality and compute the visibilities by sweeping the disparity plans from δ_M to δ_m.

Algorithm 6: Computing $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ of disparity plane δ

```
if δ < 0 then
    u_min = (n − 1)δ
    u_max = nc
else
    u_min = 0
    u_max = (n − 1)δ + nc
```

Figure 3.16 shows the behavior of the energy function mentioned in the previous section and used to optimize the materiality map. Red rectangles outline thick or dense areas of high similarity scores. In these areas, the optimized materiality map illustrated in figure 3.16.d yields the right disparity, while the similarity map described in figure 3.16.c is ambiguous and does not induce the right decision about defining the best local disparity. Therefore the materiality map is more efficient than traditional similarity based stereo matching methods [52, 74].

3.10 Final materiality decision

After its optimization, the materiality map is composed of values in $[0,1]$. In this section, we propose a method to binarize the materiality map in order to extract the surface using two different approaches: i) Adaptive scanline optimization, ii) Graph cut for materiality map segmentation.

3.10.1 Adaptive scanline optimization

The last step of our method is to determine the target points which belong to the reconstructed surface. Our main contribution in this section is applying scanline optimization for multi-baseline stereovision. In section 2.2.2.4.1, we presented the dynamic programming methods implemented for two images. Scharstein et al. [60] propose a recursive algorithm through Disparity Space (DS) indexed by $(\mathbf{m}_I, \delta)$ using the left image domain and disparity range. Unlike traditional (symmetric) dynamic programming, the ordering constraint does not need to be enforced and no occlusion cost parameter is necessary using the scanline optimization method.

In our framework, the same concept of scanline optimization [60] is applied to search the optimal path for each scanline using two steps, forward and backward. In the forward step, the optimal path cost for each target point is defined as $\mathcal{PC}(\mathcal{P}ath_{\mathbf{t}})$, where $\mathcal{P}ath_{\mathbf{t}}$ refers to the set of connected target points starting from left side of disparity space until oncoming

Algorithm 7: Initialization and optimization of the materiality map

$\mathbf{t} \equiv (\mathbf{u}, \mathbf{v}, \delta)^t$
foreach *disparity plane* δ *from nearest to farthest* **do**
  Compute $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$
  foreach *epipolar plane* v **do**
    for $u = u_{min}$ *to to* u_{max} **do**
      for $(i, j) \in r$ **do**
        Compute $\rho_{ij}(\mathbf{t})$ (see eq.3.4)
      for $i \in [0, n[$ **do**
        Compute visibility $V_i(\mathbf{t})$ according i (see eq.3.16)
      $\mu[\mathbf{t}] = \max_{(i,j)\in r} \rho_{i,j}(\mathbf{t}) * \min_{(i,j)\in r} cnf_{i,j}(\mathbf{t})$
repeat
  foreach *disparity plane* δ *from farthest to nearest* **do**
    foreach *each epipolar plane* v **do**
      for $u = u_{min}$ *to* u_{max} **do**
        Compute E_{data}, $E_{density}$, and $E_{thickness}$
        Compute global total derivative for **t** using alg.5, eq.3.43 and eq.3.45
        Update derivative value for **t** neighbors using eq.3.45
  foreach *each disparity plane* δ *from nearest to farthest* **do**
    foreach *each epipolar plane* v **do**
      for $u = u_{min}$ *to* u_{max} **do**
        for $i \in [0, n[$ **do**
          Compute visibility $V_i(\mathbf{t})$
        Update materiality $\mu[\mathbf{m}, \delta]$
until *Convergence(number of iteration, cost gain threshold, ...)*;

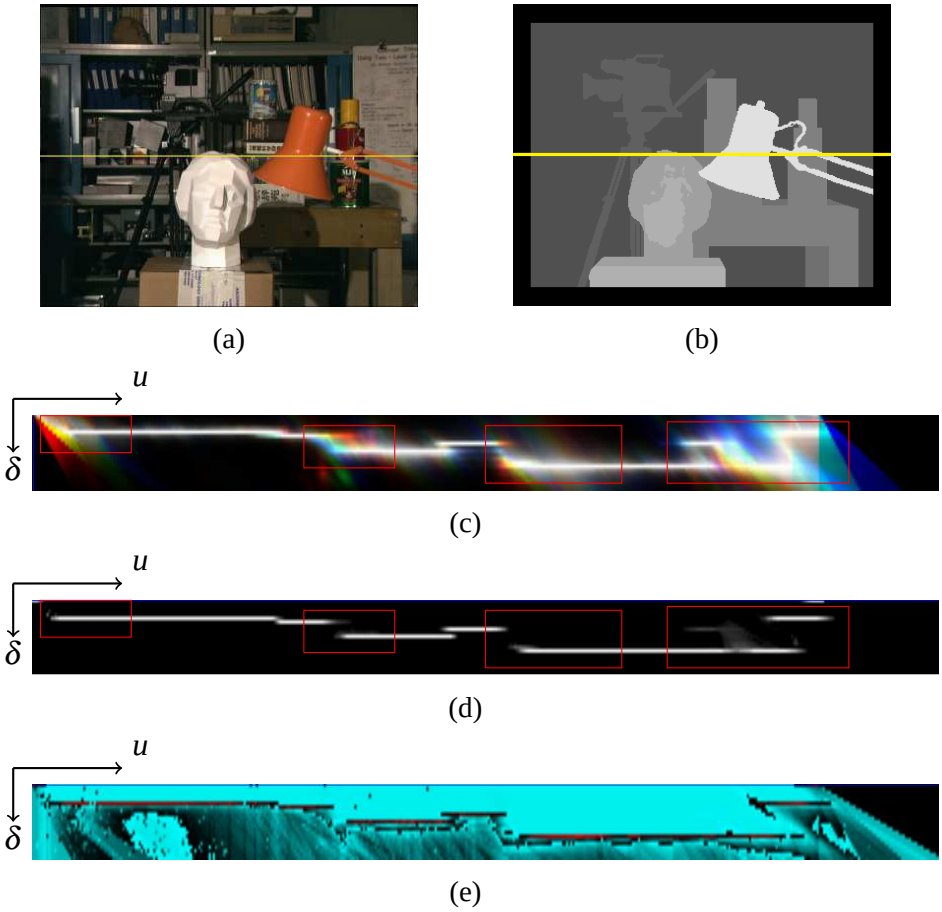


Fig. 3.16 Sample slice through a 3D disparity space: a,b) one original Tsukuba image and its ground truth disparity with highlight on scanline 144 drawn in yellow; c) similarity scores for epipolar plane 144 using four Tsukuba images with disparity range $\{0,\ldots,21\}$. Red, green, and blue colors represent respectively similarities for pairs of images ρ_{01}, ρ_{12}, and ρ_{23}; d) slice of optimized materiality map through epipolar plane 144: white points refer to high materiality values; d) total energy ($\mathbf{E}_{global}$) derivative according to local materiality for epipolar plane 144 with red, blue, and black points expressing respectively negative, positive, and zero values.

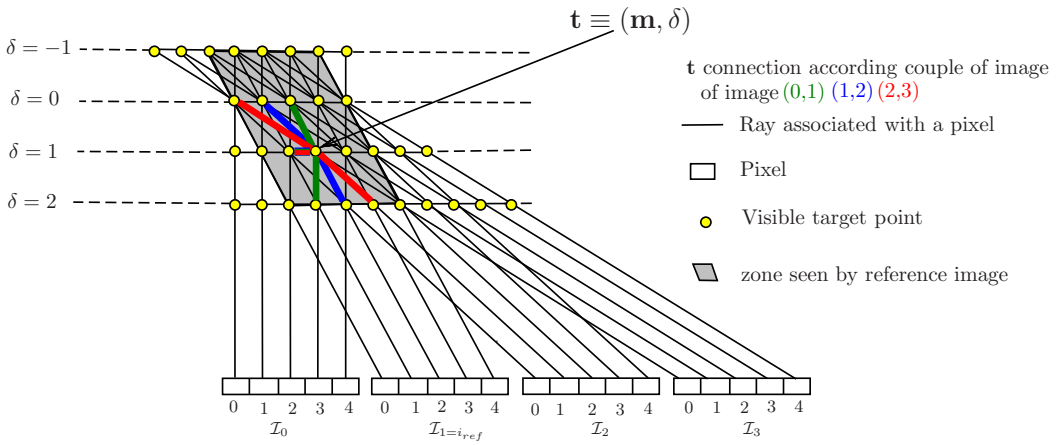


Fig. 3.17 Problem of scanline optimization in multi-stereovision. The figure is dedicated to RECOVER3D project (four cameras for each multiscopic unit) and illustrated with different possibilities to choose neighbors for a target point.

the target point **t**. Whereas in the backward step, the lowest cost path is extracted from the target points on the right side of DS and yields the target points of the reconstructed surface.

In the RECOVER3D project, we worked with multiscopic units composed of four cameras as we mentioned in section 1.3.1. Therefore, the method proposed here is customized for four input images. One of the major challenges to implement this method is represented by the following question: "which neighbors should be considered for each target point in the forward step?". Figure 3.17 shows all considered neighbors for a target point **t** according to the chosen reference image, taking into account the two following facts:

- Sweeping the disparity plane from δ_m to δ_M allows to introduce, in the forward step, a neighbor $\mathbf{t}' = (\mathbf{m}^t, \delta + 1)^t$ for a target point $\mathbf{t} = (\mathbf{m}^t, \delta)^t$. Exploiting this sweeping mechanism provides the possibility to build optimal paths $\mathcal{P}ath_{\mathbf{t}}$ containing two target points located on the same image ray and, thus reconstructing occluded surfaces.
- Favoring small disparity steps to smooth the final results , that is to say each target point $\mathbf{t} = (\mathbf{m}, \delta)^t$ seeks the best path from three neighbors located at disparities $\delta - 1$, δ and $\delta + 1$.

We mention that selecting the $\mathcal{I}_0, \mathcal{I}_2$, or $\mathcal{I}_3$ images as reference does not allow to access to the direct neighbors of the target point **t**. Direct neighbors for **t** do not generate holes in the reconstructed surface caused by ignoring target points located between **t** and its neighbors (see figure 3.17). However, using $\mathcal{I}_1$ as reference, **t** can be connected to its direct neighbors as illustrated in figure 3.17. For this reason, the image $\mathcal{I}_1$ is considered as reference in

our approach. The neighbors for target point $\mathbf{t}$ can be identified by $\{\mathbf{t}+dir : dir \in \{(1-i_{ref},0,1)^t,(-1,0,0)^t,(i_{ref}-2,0,-1)^t$ with: $i_{ref}=0\}\}$ knowing that we choose $i_{ref}=0$ for building the scene space sampling as described in section 3.3.

3.10.1.1 Cost function

As we mentioned previously, our proposition is composed of two main steps: forward and backward. In the forward step, the optimal path cost $\mathcal{PC}(\mathcal{P}ath)$ for epipolar plane $\mathbf{v}$ is built by sweeping from left $ray_1(\mathbf{t}_0)$ to right $ray_1(\mathbf{t}_{max})$ of reference image $\mathcal{I}_1$ and from nearest to farthest disparity planes.

Thus, we start from $\{\mathbf{t}_0=(\mathbf{u}_0,\mathbf{v},\delta)^t : \mathbf{u}_0=(1-i_{ref})\delta$ with: $i_{ref}=0\}$ and end with $\{\mathbf{t}_{max}=(\mathbf{u}_{max},\mathbf{v},\delta)^t : \mathbf{u}_{max}=nc+(1-i_{ref})\delta$ with: $i_{ref}=0$ and $nc=$ image width$\}$ to be certain to treat all target points of this epipolar plane. The path cost $\mathcal{PC}(\mathcal{P}ath_{\mathbf{t}})$ for any path of connected points is recursively defined in equation 3.46. The algorithm minimizes the global cost of path $\mathcal{PC}(\mathcal{P}ath)$, which is the sum of non materiality penalties $MC(\mathbf{t}_i)$ for each constituting target point $\mathbf{t}_i$ and the connection costs $\mathcal{CC}(\mathbf{t}_i,dir_i)$ between successive target point $\mathbf{t}_i$ and $\mathbf{t}_{i-1}=\mathbf{t}_i+dir_i$. The connection cost between two target points depends on the similarity scores over set of images pairs C_{dir} (see equation 3.49). When a target point $\mathbf{t}$ has the same cost CC derived from its neighbors, the target point which has the best similarity score is considered within the shortest path passing through $\mathbf{t}$. The set of image pairs $\mathcal{C}_{dir}$ implied in $\mathcal{CC}$ computation is identified depending on the neighboring direction dir in order to include every image couple for which a target point and its dir neighbor would not be occluded by a local surface passing through them. For connection between target points at same disparity $\mathbf{t}=(\mathbf{m},\delta)^t$ and $\mathbf{t}'=(\mathbf{m}',\delta')^t$ with $\delta=\delta'$, the local surface is locally frontal and the two target points are supposed to be seen in each image (see figure 3.17). Whereas, if $\delta>\delta'$ ($\mathbf{t}$ nearer than $\mathbf{t}'$), the couple of target points is seen by left most pair of cameras (only the pair of images $(0,1)$). On the other side, when $\delta<\delta'$ ($\mathbf{t}$ farther than $\mathbf{t}'$), the target point couple is viewed by right most pair of cameras (the pair $(2,3)$). This definition of per neighbor set of relevant image pairs $\mathcal{C}_{dir}$ permits to exploit the similarity information from all images considering the occlusion geometry. We can then get the path cost as follows:

$$\mathcal{PC}(\mathcal{P}ath)=\sum_{\mathbf{t}_i\in\mathcal{P}ath}MC(\mathbf{t}_i)+\sum_{\mathbf{t}_i\in\mathcal{P}ath}\mathcal{CC}(\mathbf{t}_i,dir_i) \tag{3.46}$$

$$dir_i\equiv\mathbf{t}_{i-1}-\mathbf{t}_i \qquad dir_i\in Dir$$

$$\mathcal{P}ath=\{\mathbf{t}_i(\mathbf{u}_i,\mathbf{v},\delta_i)|\ \ \mathbf{u}_i\in\mathbf{u}_0,....,\mathbf{u}_{max},\delta_i\in\delta_m,.....,\delta_M\}$$

$$MC(\mathbf{t})=1-\mu(\mathbf{t}) \tag{3.47}$$

$$\mathcal{CC}(\mathbf{t},dir) = \sum_{(i,j)\in\mathcal{C}_{dir}} \frac{\left|\rho_{ij}(\mathbf{t}+dir) - \rho_{ij}(\mathbf{t})\right|}{max(\rho_{ij}(\mathbf{t}+dir),\rho_{ij}(\mathbf{t}))} \tag{3.48}$$

$$\mathcal{C}_{(1-i_{ref},0,1)} = \{(2,3)\} \quad \mathcal{C}_{(-1,0,0)} = \{(0,1),(1,2),(2,3)\} \quad \mathcal{C}_{(i_{ref}-2,0,-1)} = \{(0,1)\} \tag{3.49}$$

The algorithm proceeds recursively and stores the best path $\mathcal{P}ath_{\mathbf{t}}$ leading to each target point $\mathbf{t}$ from the first ray $ray_1(\mathbf{t}_0)$ of images $\mathcal{I}_1$. To do this, the algorithm identifies and stores among evaluated neighbors the chosen left point $\mathcal{P}rec(\mathbf{t})$ that minimizes the cost of the optimal path to $\mathbf{t}$. This can be done minimizing $\mathcal{PC}(\mathcal{P}ath_{\mathbf{t}+dir}) + \mathcal{CC}(\mathbf{t},dir)$ as described in following equations:

$$\forall\delta \quad \mathbf{t}_0 = (\mathbf{u}_0, v, \delta) \quad \mathcal{P}rec(\mathbf{t}_0) = (-1,-1,-1)^t \quad \mathcal{PC}(\mathcal{P}ath_{\mathbf{t}_0}) = MC(\mathbf{t}_0)$$

$$\mathcal{P}rec(\mathbf{t}) = \mathbf{t} + argmin_{dir}\mathcal{PC}(\mathcal{P}ath_{\mathbf{t}+dir}) + \mathcal{CC}(\mathbf{t},dir)$$

$$\mathcal{PC}(\mathcal{P}ath_{\mathbf{t}}) = \mathcal{PC}\left(\mathcal{P}ath_{\mathcal{P}rec(\mathbf{t})}\right) + \mathcal{CC}(\mathcal{P}rec(\mathbf{t},dir) + MC(\mathbf{t})$$

The final decision is the minimum cost path within right ray of image 1 $ray_1(\mathbf{t}_{max})$: $\mathcal{P}ath_{\mathbf{tr}}$, with $\mathbf{tr} = argmin_{\mathbf{t}\in ray_1(\mathbf{t}_{max})}\mathcal{PC}(\mathcal{P}ath_{\mathbf{t}})$. This optimal path can be retrieved backwards from $\mathbf{tr}$ according to successive chosen left points $\mathcal{P}rec(\mathbf{t})$ up to the first point encountered on left ray of image 1 $ray_1(\mathbf{t}_0)$.

3.10.2 Graph cut for materiality map binarization

In spite of the efficient proposition of adaptive scanline optimization for multi-baseline stereovision, the binary disparity map still suffers from stroke lines due to independent handling of adjacent epipolar planes. In this section, we propose to use another approach to binarize the materiality map. Our main idea is to segment our materiality map into two classes thanks to a graph cut algorithm. The first class consists of the target points located on or behind the reconstructed surface, whereas the second represents the target points in front of the reconstructed surface. Our idea is inspired from one of the most famous use of "graph cut" for image segmentation [8]. In chapter 2, we explained different methods using graph cut as the matching method to solve the stereovision matching problem. Here, our proposition differs from those since the graph cut works as the segmentation method. Therefore, we start by building a weighted graph $G = (E,N)$ that is composed of edges E and nodes N as illustrated in figure 3.18. The nodes are the target points in addition to two other nodes (source and sink dedicated respectively to class 1 and 2) like the traditional graph cut method. Indeed, we can classify the edges E within the graph G into four majors groups E_{sink}, E_{source}, E_s, and E_δ:

- E_{sink} between each node and the sink node;

- E_{source} between each node (except the nodes representing the non visible target points that are considered out of reconstructed zones and connected only to the sink node) and the source node;
- E_s between each node and its four neighbors in the same disparity plane;
- E_δ between each target point $(\mathbf{u},\mathbf{v},\delta)$ and its neighbors located at disparity $\delta-1$, $\delta+1$ according to rays emitted from the second image. The second image choice will be explained in detail later in this section.

Thanks to the rich information available in each target point, we can write the edge capacity between each node and the source and sink nodes taking into account the following facts:

- The nodes representing the target points located in front of the surface should have low materiality and high visibility scores on at least one image.
- The nodes representing the target points on the surface should have high visibility scores (at least on one image) and high materiality scores.
- The nodes representing the target point behind the surface should have low visibility scores.

Equation 3.50 expresses the previous facts into capacity scores $(C_{E_{sink}},C_{E_{source}})$ for the connection edges between each target point $\mathbf{t}$ and sink or source nodes respectively as follows:

$$\begin{aligned} C_{E_{sink}}(\mathbf{t}) &= (1-\mu[\mathbf{t}])max_{i\in[0,n[}\mathcal{V}_i(\mathbf{t}). \\ C_{E_{source}}(\mathbf{t}) &= (1-max_{i\in[0,n[}\mathcal{V}_i(\mathbf{t}))\mu[\mathbf{t}]\ \ max_{i\in[0,n[}\mathcal{V}_i(\mathbf{t}). \\ &= 1-(1-\mu[\mathbf{t}])max_{i\in[0,n[}\mathcal{V}_i(\mathbf{t})). \\ &= 1-C_{E_{sink}}(\mathbf{t}). \end{aligned} \tag{3.50}$$

Moreover, the edge capacity C_{E_s} between a node and one of its four neighbors in the same disparity plane helps to maintain the connection between the target points that have the similar materiality values. The edge capacity C_{E_s} is described as follows:

$$\begin{aligned} C_{E_s}(\mathbf{t},\mathbf{t}') &= \beta\left(1-|\mu[\mathbf{t}]-\mu[\mathbf{t}']|\right) \\ &\quad \beta\text{: smoothing factor.} \end{aligned} \tag{3.51}$$

Finally, the E_δ edges are connections between each node and its neighbors in adjacent disparities $\delta-1$ and $\delta+1$. The main idea for computing their capacity is to keep high capacity for edges except those between, the nodes belonging to the reconstructed surface and direct neighbors of these nodes that are located in front of surface (at higher disparity).
In fact, our materiality map contains $\{2n$: n is number of image$\}$ connections for each target point with other target points that are located in disparities $\delta-1$ and $\delta+1$. Since we

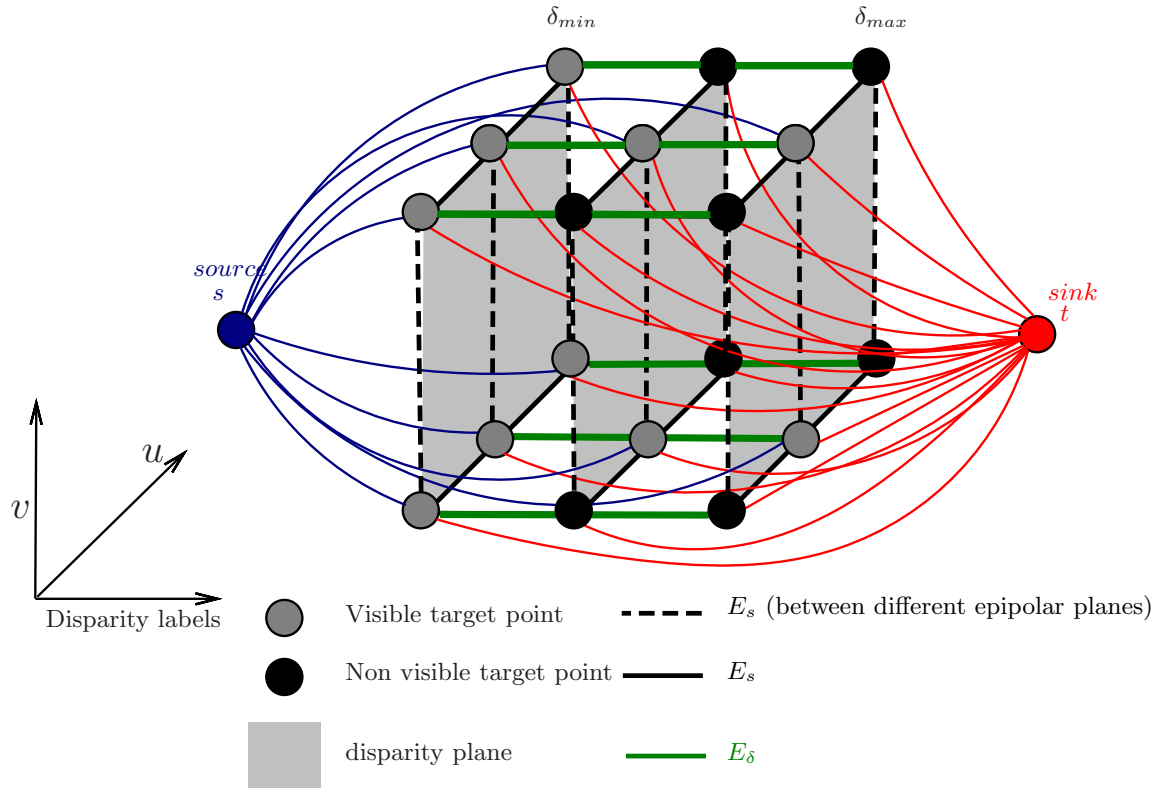


Fig. 3.18 Graph with two terminal nodes for materiality map binarization

can not implement all the connections introduced for a target point according to all images within the graph G, we choose one of the images as the reference. Between four available images, the second or third image could be chosen as reference since they provide the best description of the reconstructed scene and avoids the occluded zones. In our application, we choose the second image and the edge capacity is computed from opposite of visibility difference in image i behind each neighbor as follows:

$$C_{E_\delta}(\mathbf{t}) = 1 - (\mathcal{V}_{i=1}(\mathbf{t}) \; - \; \mathcal{V}_{i=1}(\mathbf{t}) \; (1 - \mu[\mathbf{t}])) = 1 - \mu[\mathbf{t}] \; \mathcal{V}_{i=1}(\mathbf{t}) \tag{3.52}$$

After building the graph G and assigning relevant capacities to the edges E, the $s-t$ min-cut method introduced by Ford and Fulkerson [22] is applied in order to find the optimum classification as described in section 2.2.2.4. In our application, we implement the method of Ford and Fulkerson [22] by using GridCut library[1].

[1] http://www.gridcut.com/

3.11 Experimental results

To study the properties of our multi-baseline stereovision method, we ran our program over a set of three image sequences created by Middlebury College (Cones, Teddy) and University of Tsukuba (Tsukuba) [61]. Figures 3.19, 3.20, and 3.21 show disparity maps derived from existing methods (TreeDP[77], MultiResGC[53], DoubleBP[81], GC+occ [38], AdaptAggrDP[80]) and those obtained from our materiality map results using adaptive scanline optimization and graph cut for materiality map binarization. The different comparison methods are chosen according to two features: i) The methods have good scores of evaluation in web site of Middlebury University to stereovision [61], ii) they contain most of widely known optimization processes for stereovision (bilateral filtering and dynamic programming: AdaptAggrDP[80], Dynamic programming: TreeDP[77], Belief propagation: DoubleBP[81], and Graph cut: GC+occ [38], MultiResGC[53]).
Figures 3.22, 3.23, and 3.24 show colored target point clouds derived from our materiality map results. These target points are extracted from binarized materiality maps using adaptive scanline optimization or graph cut using four images of Tsukuba, Cones, and Teddy datasets.
Our method fails to recover textureless regions as illustrated in Figures 3.21 and 3.23 using the Teddy dataset. However, our method is able to reconstruct the repeated texture which are one of the major problems in traditional stereovision, as illustrated in Figures 3.20 and 3.22 using the Cones dataset.
We intended to perform an online evaluation to compare ourselves with TreeDP[77], MultiResGC[53], DoubleBP[81], GC+occ [38], and AdaptAggrDP[80]. However, the new version (3) of available datasets in the web site of Middlebury University to stereovision [61] provides only two images in simplified epipolar geometry for each scene. For this reason, we compare and evaluate off-line our results with the second data set version using Root-Mean-Squared (RMS) and Percentage of Bad Matching (PBM) measures proposed by [60] and described in section 3.4.5. RMS expresses the mean square between produced disparity map and ground truth, whereas PBM produces the percentage of mismatching pixels between the two disparity maps.
The results show (see table 3.5) that our method with graph cut segmentation (MatGC) provides better results over the Tsukuba and Teddy datasets than the method with adaptive scanline optimization (MatAS). However, MatGC and MatAS produce equivalent results for the Cones dataset. The adaptive scanline optimization proposed in section 3.10.1 to find independently the optimal path for each epipolar plane fails to deal with the scenes containing objects with soft edges (e.g. the background cupboard in Tsukuba scene) which require a smoothing operation. Whereas the adaptive scanline optimization handles efficiently ob-

	Tsukuba		Teddy		Cones	
	RMS	PBM	RMS	PBM	RMS	PBM
TreeDP [77]	18.8413	0.2301	14.0155	0.4226	18.0412	0.4580
MultiResGC [53]	11.6705	0.1580	4.5579	0.3828	7.9205	0.3893
DoubleBP [81]	12.3372	0.1902	3.6295	0.4037	8.1742	0.3724
GC+occ [38]	13.3261	0.0711	13.2708	0.4412	16.1577	0.3846
AdaptAggrDP[80]	15.1570	0.2547	7.7827	0.4144	9.3184	0.4152
MatGC*	15.4442	0.1253	6.9411	1.026	6.9290	0.3247
MatAS**	19.8674	0.1629	8.2013	0.4327	5.7821	0.3211

(*) MatGC: our method using graph cut
(**) MatAS: our method with adaptive scanline optimization

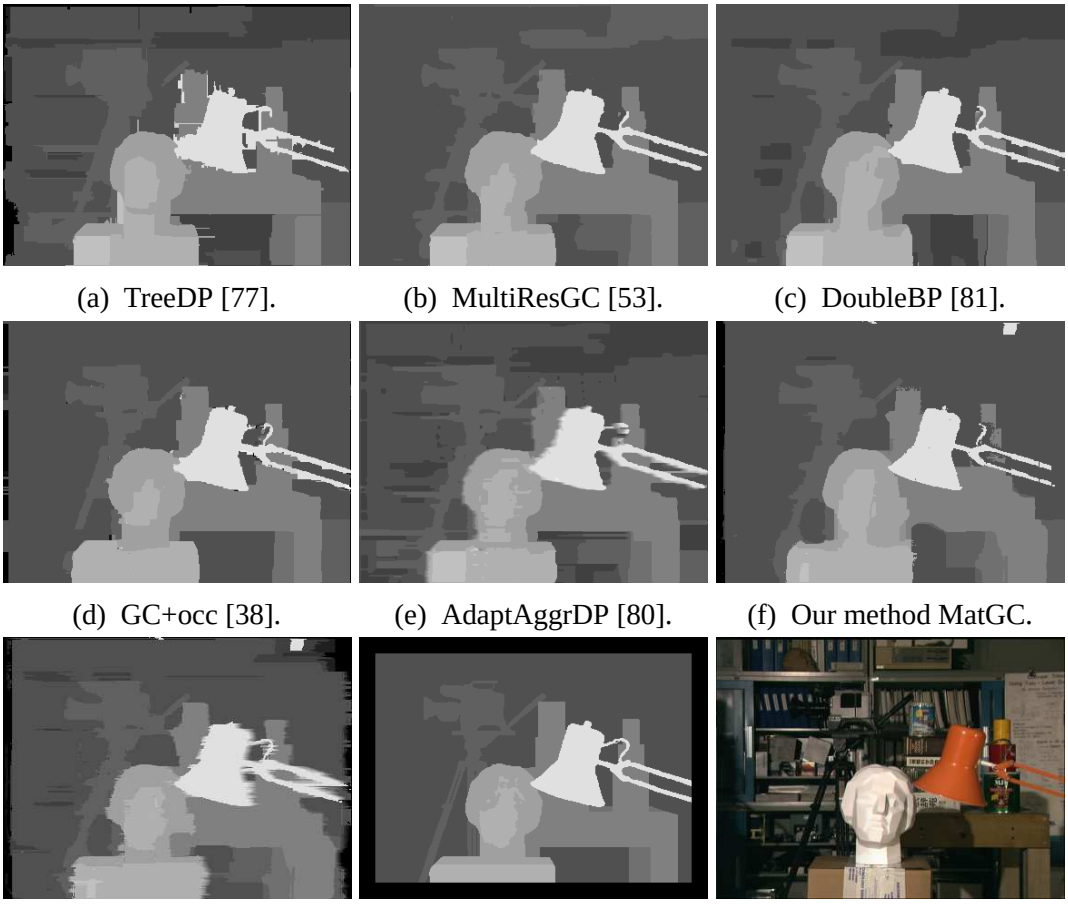
Table 3.4 RMS error and PBM measures over entire disparities maps for different methods.

	Tsukuba		Teddy		Cones	
	RMS	PBM	RMS	PBM	RMS	PBM
TreeDP [77]	1.614	3.236	3.861	1.103	3.120	1.426
MultiResGC [53]	1	2.222	1.255	1	1.369	1.212
DoubleBP [81]	1.057	2.675	1	1.054	1.413	1.159
GC+occ [38]	1.141	1	3.656	1.152	2.794	1.197
AdaptAggrDP[80]	1.298	3.582	2.144	1.082	1.611	1.293
MatGC	1.323	1.704	1.912	1.043	1.033	1.011
MatAS	1.702	2.291	2.259	1.130	1	1

(*) MatGC: our method using graph cut
(**) MatAS: our method with adaptive scanline optimization

Table 3.5 Normalized measures computed from table 3.4 for each datasets each measure is divided by the minimum one for the dataset.

jects including sharp edges (e.g. the cones in Cones image). Table 3.5 shows the results of the measures (RMS) and (PBM) using four images of the Tsukuba, Teddy, and Cones datasets. The graphs 3.25 and 3.26 show that our method is competitive with others especially for the Cones dataset where there are no textureless regions. Moreover, one of our main goals is to deal with semi-occluded regions. Therefore we focused our evaluation on occluded regions (see the figure 3.29) for three datasets Tsukuba, Teddy, and Cones. The graphs 3.27 and 3.28 show that our methods robustly deal with the occluded regions over the three data sets. As mentioned in this section, our method still not handles some known problems in stereovision like the textureless zones. Therefore, in chapter 4, we propose a novel framework to merge the approach described in this chapter with visual hull using the efficiency of our definition of scene geometry and the attributes of each target point (e.g. confidence).



(a) TreeDP [77]. (b) MultiResGC [53]. (c) DoubleBP [81].

(d) GC+occ [38]. (e) AdaptAggrDP [80]. (f) Our method MatGC.

(g) Our method MatAS. (h) Ground truth. (i) Tsukuba image.

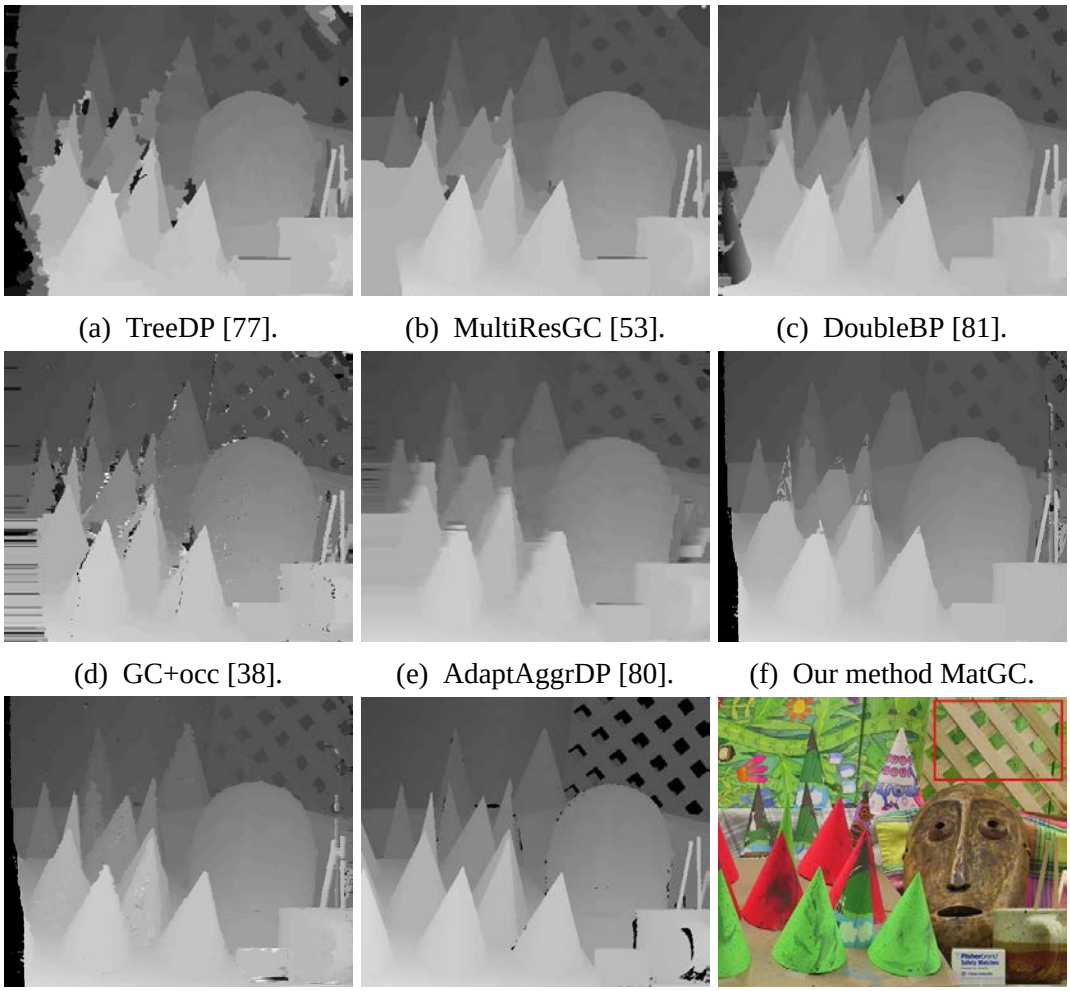
Fig. 3.19 Results (expressed as disparity maps) of several methods: a) [77], b) [53], c) [81], d) [38], e) [80]. f, g) Disparity maps extracted from our binarized materiality map using respectively graph cut and adaptive scanline optimization. h,i) Ground truth for disparity map and original image of Tsukuba dataset, source: [61].

	Tsukuba		Teddy		Cones	
	RMS	PBM	RMS	PBM	RMS	PBM
TreeDP[77]	67.8856	0.4703	49.4163	0.9932	55.4218	0.8451
MultiResGC[53]	33.1759	0.3246	14.9494	0.6397	19.4809	0.7700
DoubleBP[81]	36.6243	0.3542	11.4229	0.6042	25.8780	0.7656
GC+occ [38]	53.9264	0.4513	51.9460	0.7606	45.3130	0.8600
AdaptAggrDP[80]	61.1417	0.9247	30.2555	0.9478	23.8277	0.9438
MatGC	23.8020	0.3636	14.2685	0.5578	11.8073	0.5665
MatAS	60.4967	0.6098	24.1590	0.6613	11.8886	0.5673

(*) MatGC: our method using graph cut

(**) MatAS: our method with adaptive scanline optimization

Table 3.6 RMS error and PBM measures over occluded regions for different methods

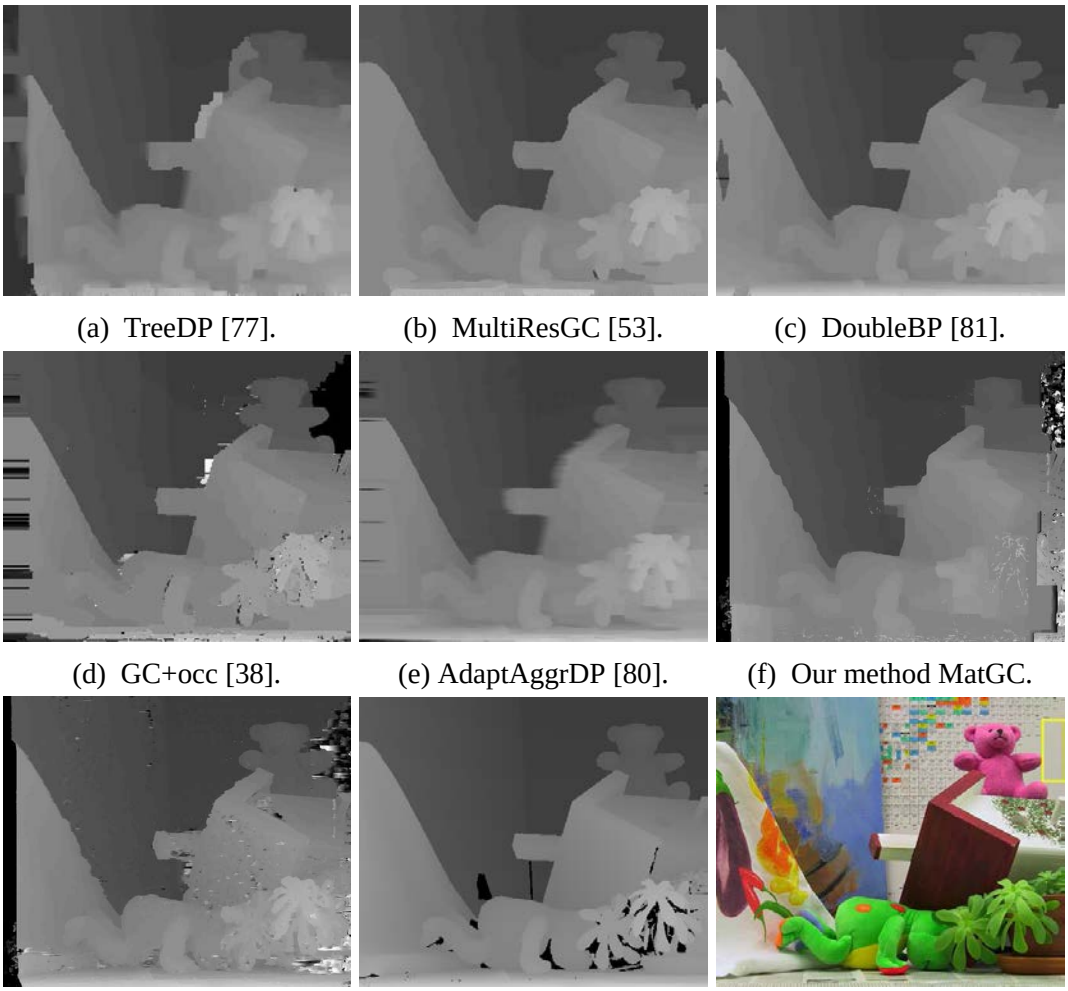


(a) TreeDP [77]. (b) MultiResGC [53]. (c) DoubleBP [81].

(d) GC+occ [38]. (e) AdaptAggrDP [80]. (f) Our method MatGC.

(g) Our method MatAS. (h) Ground truth. (i) Cones image.

Fig. 3.20 Results (expressed as disparity maps) of several methods: a) [77], b) [53], c) [81], d) [38], e) [80], source: [61]. f, g) Disparity maps extracted from our binarized materiality map using respectively graph cut and adaptive scanline optimization. h) Ground truth for disparity map. h) Original image with highlights on regions with repeated textures drawn in red for Cones dataset.

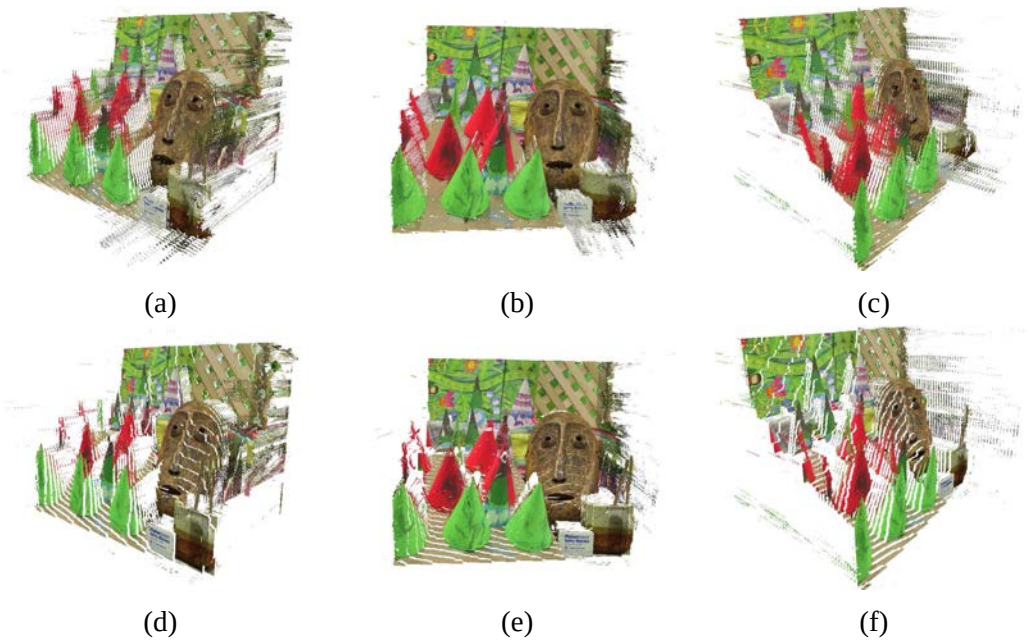


(a) TreeDP [77]. (b) MultiResGC [53]. (c) DoubleBP [81].

(d) GC+occ [38]. (e) AdaptAggrDP [80]. (f) Our method MatGC.

(g) Our method MatAS. (h) Ground truth. (i) Teddy image.

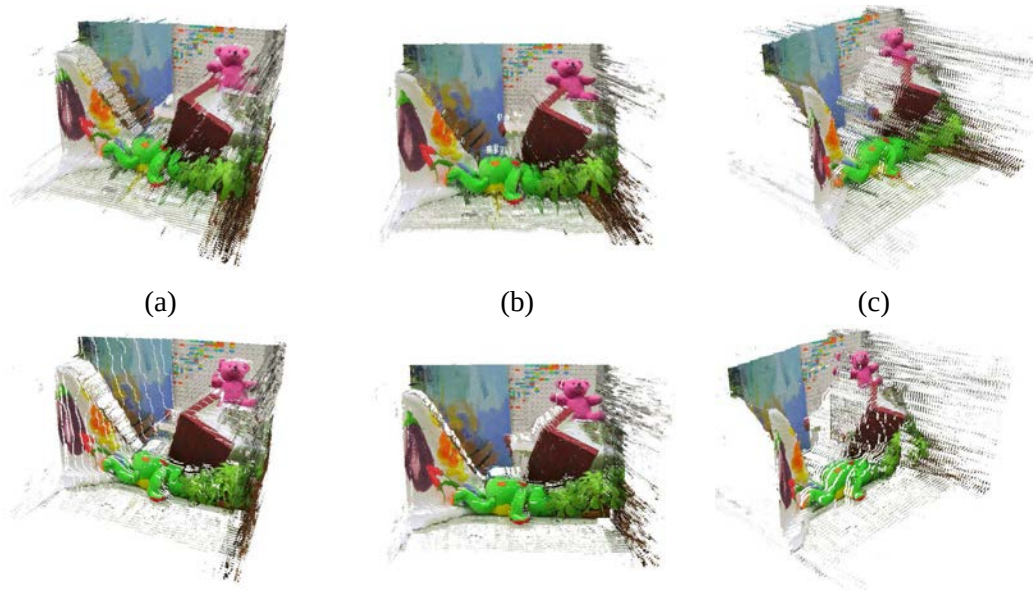
Fig. 3.21 Results (expressed as disparity maps) of several methods: a) [77], b) [53], c) [81], d) [38], e) [80], source: [61]. f, g) Disparity maps extracted from our binarized materiality map using respectively graph cut and adaptive scanline optimization. h) Ground truth for disparity map. h) Original image with highlights on textureless region drawn in yellow for Teddy dataset.



(a) (b) (c)

(d) (e) (f)

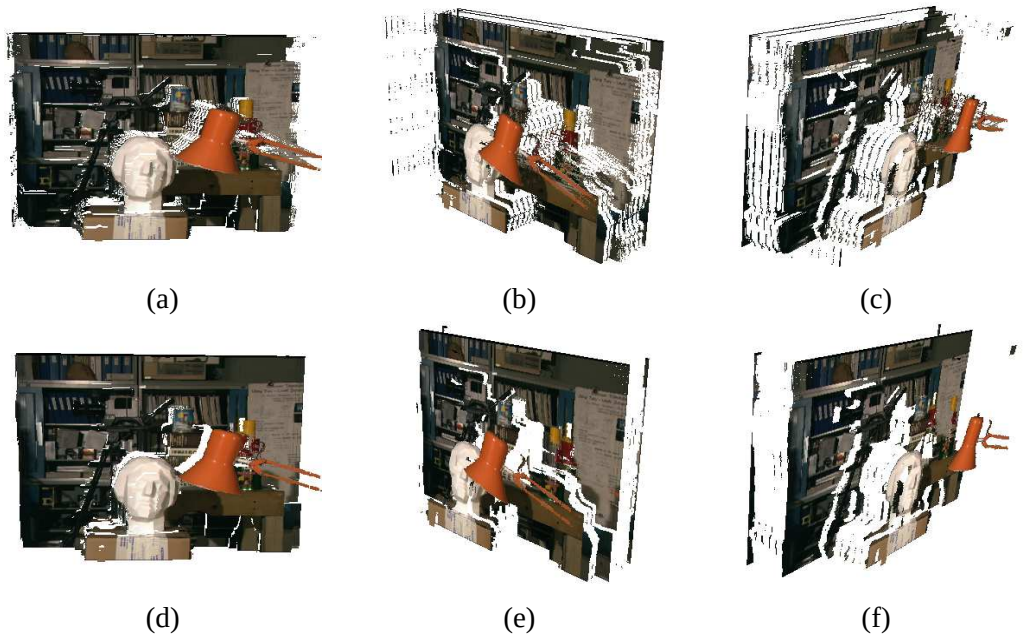
Fig. 3.22 Materiality map results: Three different views for target points extracted from adaptive scanline optimization (first row) and segmentation by graph cut (second row) using four images of Cones dataset.



(a) (b) (c)

(d) (e) (f)

Fig. 3.23 Materiality map results: Three different views for target points extracted from adaptive scanline optimization (first row) and segmentation by graph cut (second row) using four images of Teddy dataset.



(a) (b) (c)

(d) (e) (f)

Fig. 3.24 Materiality map results: Three different views for target points extracted from adaptive scanline optimization (first row) and segmentation by graph cut (second row) using four images of Tsukuba dataset.

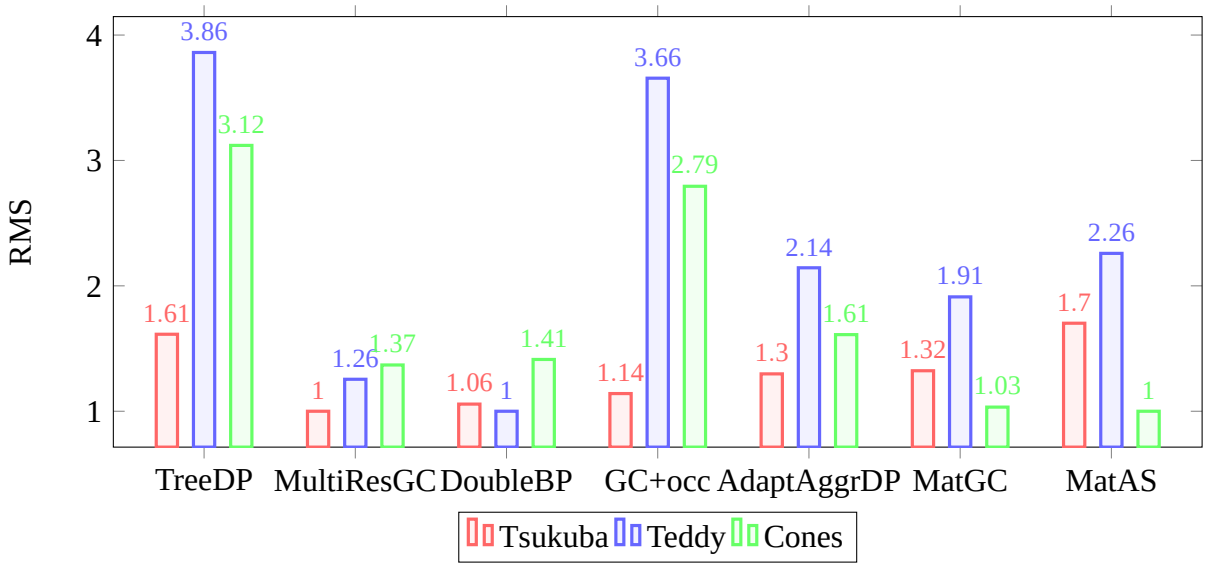


Fig. 3.25 Normalized RMS results derived from table 3.5.

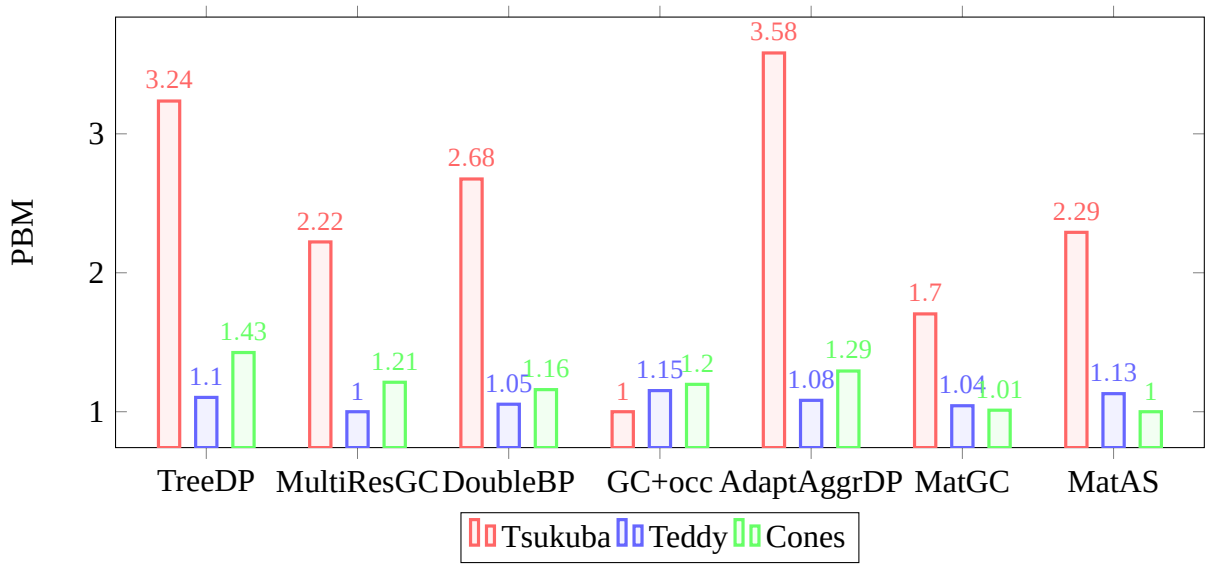


Fig. 3.26 Normalized PBM results derived from table 3.5.

	Tsukuba		Teddy		Cones	
	RMS	PBM	RMS	PBM	RMS	PBM
TreeDP[77]	2,852	1,448	4,326	1,780	4,693	1,491
MultiResGC[53]	1,393	1	1,308	1,146	1,649	1,359
DoubleBP[81]	1,538	1,091	1	1,083	2,191	1,351
GC+occ [38]	2,265	1,390	4,547	1,363	3,837	1,518
AdaptAggrDP[80]	2,568	2,848	2,648	1,699	2,018	1,666
MatGC	1	1,120	1,249	1	1	1
MatAS	2,541	1,878	2,114	1,185	1,006	1,001

(*) MatGC: our method using graph cut
(**) MatAS: our method with adaptive scanline optimization

Table 3.7 Normalized measures computed from table 3.6 for each dataset: each measure is divided by the minimum one for the dataset.

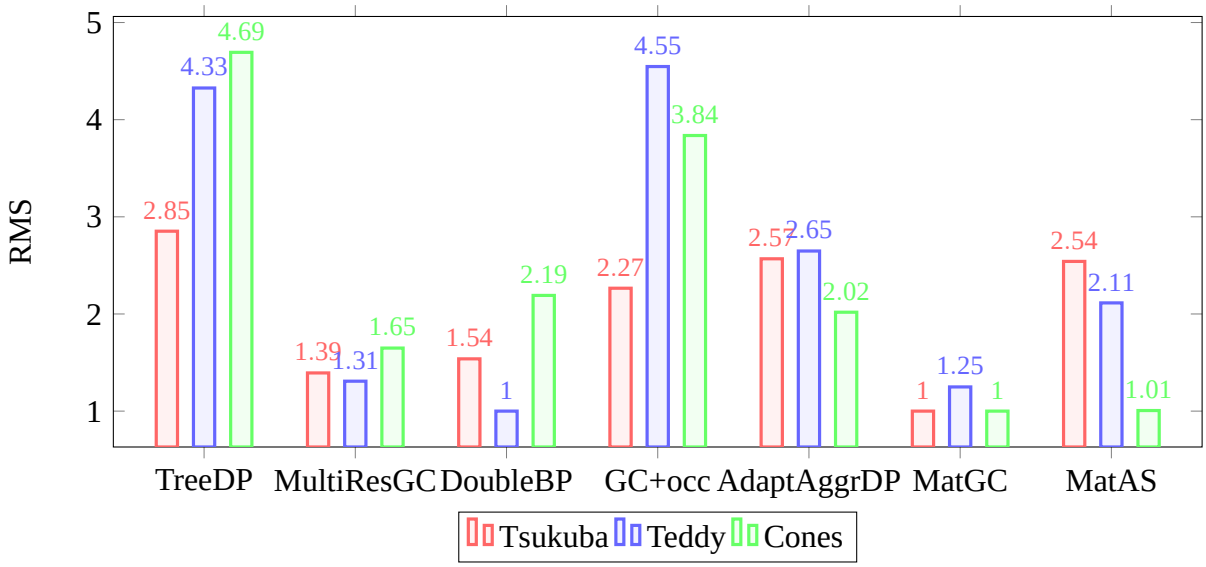


Fig. 3.27 Normalized RMS results derived from table 3.7.

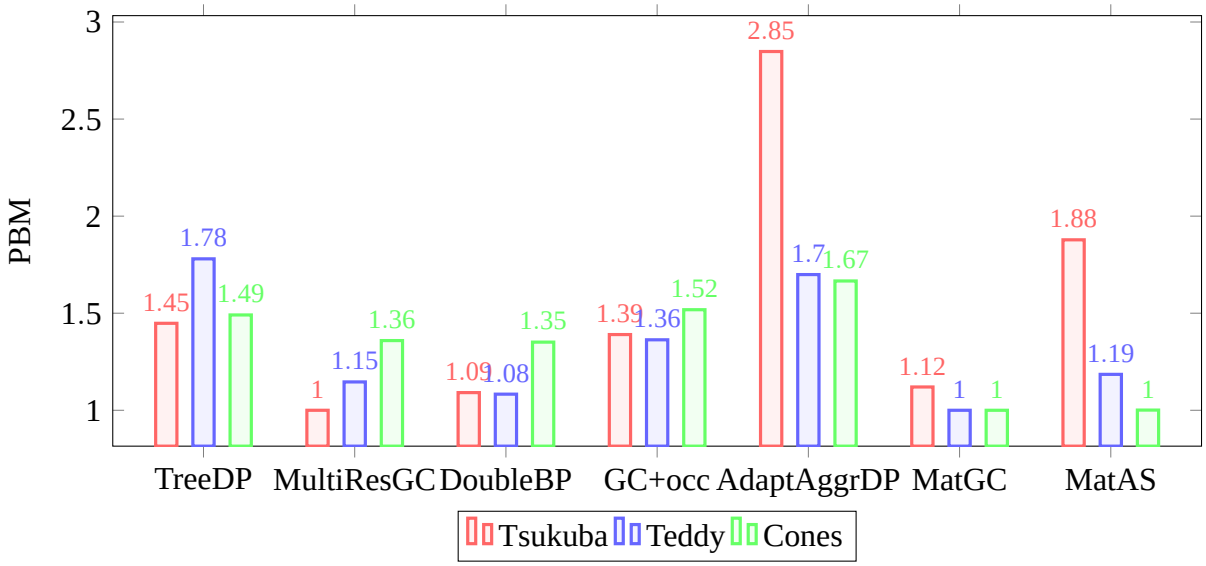


Fig. 3.28 Normalized PBM results derived from table 3.7.

(a) (b) (c)

Fig. 3.29 a, b, c) Non-occluded regions (white) and occluded and border regions (black) for Tsukuba, Teddy and Cones datasets.

3.12 Conclusion

This chapter proposes several new ideas to solve some of multi-baseline stereovision limitations. Using the disparity space as our sampling scheme for scene space domain, we focus on the useful 3D reconstruction space while strictly avoiding any semi-occlusion and simplifying handling of total occlusions. The proposed materiality map framework proves efficient at reconstructing the scene by integrating visibility reasoning. We can summarize the main framework concept as follows.

Visibility: the materiality map laid on Disparity Space (DS) delivers a direct and efficient support for visibility reasoning with the function proposed by [39] and used in [74, 51]. This function is in fact conveniently defined in the framework as the product of non-materialities of all potentially occluding samples. The visibility function results may be laid as visibility maps on DS and computed efficiently from near to far. DS ensures that each 3D sample point (target point) precisely lies on a genuine pixel ray in each image of the multiscopic unit for which it is inside the frustum. It thus intrinsically describes semi-occlusions (is $\mathbf{m}_i$ in camera i frustum?) and also totally avoids complex treatment of inter-sample partial occlusions because such occlusions often occurring in other scene-based methods do not in DS.

Similarity and confidence: the materiality and visibilities of target points are evaluated for input data according to pre-computed similarity scores of neighborhoods of their projections in some pairs of images. This similarity computation is rather classical but encompasses (i) confidence computation typically based on variances of the neighborhoods and (ii) a normalizing step of similarities along pixel rays that yields final similarity scores in the range $[0,1]$.

Optimization and binarization: the materiality map is shaped by an optimization process minimizing a dedicated energy penalizing any deviation from intended map properties (such as density, thickness) and inconsistencies between materialities, visibilities, and similarities. More precisely, for each sample, and each pair of images, a cost, weighted by

the relevant confidence score, penalizes any deviation from following assumptions: a good normalized similarity of a target point for a given couple of views should be explained by high materiality and visibilities on both images, whereas poor similarity should induce low materiality or an occlusion (low visibility). After the materiality map has been optimized, a binarization process delivers the final result, a binary materiality map standing as a volumetric direct model of the intended solution, whereas image-based methods usually deliver disparity/depth maps that have to be processed to yield the reconstructed scene.
The results show (see section 3.11) that our proposition deals efficiently with repeated textures and occluded regions as compared to other methods over several datasets. However, our approach does not handle textureless regions as shown in figure 3.21. We propose thus to enhance our proposition here using the information derived from the silhouette based reconstruction as we will describe in details in the chapter 4.

3.13 Résumé : Stéréovision multi-oculaire en géométrie parallèle décentrée

Ce chapitre présente une formulation originale de la multi-stéréovision, spécifiquement construite pour le contexte multi-oculaire en géométrie parallèle décentrée, qui induit une géométrie multi-épipolaire simplifiée et régulière. Son objectif est de résoudre globalement le problème de reconstruction 3D à partir des n vues disponibles en explicitant précisément les redondances d'informations entre ces images afin d'en tirer avantage. Cette redondance induite par la capture multi-oculaire est précieuse pour la robustesse de la reconstruction mais elle implique aussi des combinatoires de recherche plus importantes qu'en stéréovision binoculaire. Pour résoudre ce problème, une approche naturelle, basée image, consiste à calculer simultanément les n cartes de disparités (entières dans notre cas), tout en respectant certaines contraintes afin d'assurer la cohérence de la géométrie de la scène ainsi reconstruite. Cette approche peut aussi être vue comme une recherche de fonction de visibilité sur l'ensemble fini des points 3D de la scène atteignables depuis les pixels par la reconstruction en disparités entières que nous nommerons « points cibles ». Toutefois, cette même formalisation des points cibles peut être exploitée dans une approche globale basée scène, plus élégante, que nous proposons. Elle consiste à construire une carte discrète 3D de « matérialité » sur cet ensemble de points. La notion de matérialité proposée exprime un degré de croyance sur l'existence du point cible dans la scène en tant que source lumineuse ponctuelle (le plus souvent indirecte) captée par au moins l'une des caméras. Cette carte de matérialité contient naturellement toutes les informations de redondance (multi-projection d'un point matériel) et d'occultation (points matériels alignés sur un rayon de projection sur un pixel). Nous présentons en détails dans la première partie de ce chapitre l'échantillonnage de l'espace de la scène basée sur l'espace de disparité et de ses points cibles. Ensuite, nous introduisons et expliquons les attributs associés à chaque point cible : la similarité, la confiance, la visibilité et la matérialité. La similarité consiste à évaluer la ressemblance colorimétrique entre les pixels à apparier. Nous proposons trois formules pour le calcul de celle-ci. La première utilise une fenêtre de voisinages non adaptative, la deuxième une fenêtre séparée et la dernière une fenêtre pondérée. À la fin de cette section, après avoir évalué ces trois formulations, nous sélectionnons la deuxième proposition pour calculer l'attribut de similarité de nos points cibles dans notre framework.

Par la suite, un processus d'optimisation, basée sur la descente de gradient, est appliqué sur la carte de matérialité. La dernière étape de la méthode est l'extraction de la surface reconstruite à partir de l'espace de la scène. Pour étudier les propriétés de notre méthode de stéréovision multi-oculaire, nous avons appliqué notre méthode sur un ensemble de

trois séquences d'images créée par Middlebury College (Cones, Teddy) et l'Université de Tsukuba (Tsukuba) [61]. En comparant nos résultats avec ceux obtenus par d'autres méthodes (TreeDP[77], MultiResGC[53], DoubleBP[81], GC+occ [38], AdaptAggrDP[80]), nous constatons que notre approche traite efficacement les problèmes issus de l'occultation entre les objets et ceux liés à la présence de textures répétitives. Cependant, nous constastons aussi que notre approche ne gère pas les régions sans texture comme le montre la figure 3.21. Par conséquent, nous proposons d'améliorer la méthode exposée dans ce chapitre en utilisant les informations dŕivant de la reconstruction basée silhouette que nous allons décrire en détail dans le chapitre 4.

Chapter 4

Fusion of silhouette and multi-baseline stereovision for 3D object modeling

In the previous chapter, we presented our 3D reconstruction framework from multiple cameras in equidistant multi-baseline layout. However, the RECOVER3D project (described in chapter 3) is based on the exploitation of two 3D reconstruction approaches: multi-baseline stereovision and silhouette-based reconstruction. In this chapter, we explain our proposed framework for 3D reconstruction from monoscopic and multiscopic units described in chapter 1 using both approaches. In section 4.1, we introduce the different steps of the proposed method and the 3D reconstruction pipeline. Our multi-baseline stereovision method proposed in chapter 3 works in disparity space laid in front of a multiscopic unit whereas the result of silhouette-based reconstruction is a visual hull expressed in a regular 3D grid set in scene reference frame. However, the results of the two approaches should be expressed in the same coordinate frame in order to merge them. Therefore, in section 4.2, we introduce the geometrical transformations between disparity space and 3D grid index domain. In section 4.3, we explain the benefits of using the information derived from the visual hull within our proposed multi-baseline stereovision method. In order to merge all the results produced on each multiscopic unit by this multi-baseline stereovision process guided by visual hull, we propose a volumetric approach. The input data to this approach are carved volumes that are presented in section 4.4. We propose in section 4.5 a novel way to merge the carved volumes in order to obtain a single 3D model representing the 3D pose of the reconstructed object(s). Finally, we show the results of 3D reconstruction for virtual and real data sets using our proposal.

4.1 Introduction

In this chapter, we propose a novel framework for multi-view 3D reconstruction relying on both multi-baseline stereovision and visual hull introduced in section 2.2.4.2 in order to produce 3D object models with high precision. This method inputs are a visual hull (VH) and several sets of views derived from multiscopic units. For each such set of views, a multi-baseline stereovision method guided by VH yields a surface that is then used to carve the VH. Multiple carved VH from different sets of views are then iteratively fused to deliver the intended 3D model.

In chapter 2, we presented bibliographical study of different methods merging the stereovision and silhouette-based approaches. We classified them into three major classes: i) stereovision guided by VH methods, ii) collaborative methods applying simultaneously criteria borrowed from both techniques, iii) separate application of both methods with subsequent merging of their results.

The proposed framework in this chapter is summarized in figure 4.1 and borrows ideas from classes (i) and (iii). After VH computation, as in class (i), the VH guides each multi-baseline stereovision process. Then VH carving from stereovision is performed for each multiscopic unit similarily to class (iii) but relies on our multi-baseline stereovision result. Finally, multiple (one per multiscopic unit) VH/multi-stereovision results are merged in a single global 3D model.

Beyond its cross classification, our framework is innovative among each class as follows. For each multiscopic unit, a global scene-based multi-baseline stereovision process is run in DS which totally avoids partial occlusions and yields a robust stereovision result replacing more local and noisy photo-consistency usually used in class (i) carving. However, the proposed class (i) VH guidance is dedicated to our multi-baseline stereovision framework proposed in the chapter 3, which it enhances in terms of domain size, outliers avoidance and, more innovatively, robustness in multi-stereovision similarity. The class (iii) VH carving from stereovision relies on voxel classification usually based, for voxels occluding the stereovision solution (group 2 in [65]), on rays from surface to reference image. Replacing this image-based classification by a volumic one in disparity space brings more precision and robustness to our solution. Furthermore, merging at the final stage multiple carved VH involves to smartly handle reconstruction inconsistencies from separate multiscopic units, which may conveniently correct some residual stereovision mismatches.

4.1.1 Contributions

The contributions of this chapter are threefold: (i) improvement of our multi-baseline stereovision method (see chapter 3) thanks to visual hull guidance, (ii) carving of visual hull

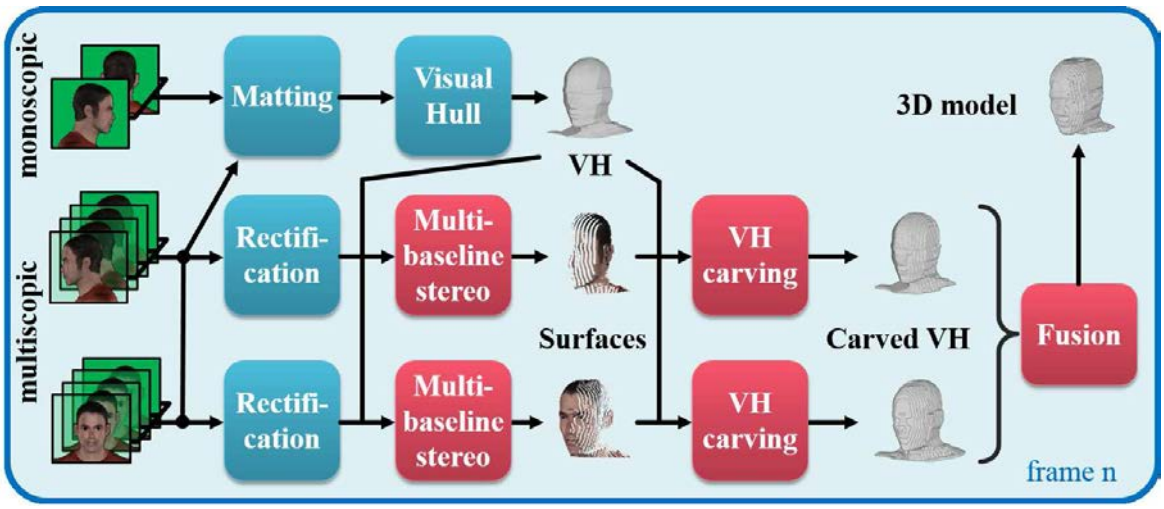


Fig. 4.1 Proposed 3D reconstruction pipeline. Red blocks involve more specific contributions of the chapter

from an interpolated and smooth stereovision surface and (iii) merging differently carved volumes in a suitable way in areas where they differ. This chapter shows that the proposed approach helps recovering a high quality carved volume (a 3D representation of objects such as humans) even for small details and in concave areas subjected to occlusion.

4.2 VH-DS geometrical mapping

Hybridizing VH and multi-baseline stereovision involves mapping results of both methods in a same coordinate frame. Natively, VH is expressed in a regular grid in the scene frame (attached to the capture studio), whereas multi-stereovision results are given in local DS irregular in actual 3D space as their samples are not evenly spaced on fan-spread pixel rays. The goal of this section is thus to produce, for any multiscopic unit, the mathematical relationships between three different coordinate systems in scene space: voxel grid index $\mathbf{g} = (w,h,d)^t$ in VH, cartesian spatial coordinates $\mathbf{M}_c = (x_c,y_c,z_c)^t$ in the frame of the rectified reference camera $\{i_{ref} = 0\}$, and index $\mathbf{t} = (u,v,\delta)^t$ in DS. This is described in Figure 4.2 in a five steps transformation from VH index to DS index. It involves using:

- the VH grid parameters (origin, size and orientation in scene frame as well as cell size or resolution) chosen at the VH extraction step.
  This grid is spatially situated in scene space using, for instance, its reference corner position $O_s = (X_v,Y_v,Z_v)^t$, volume size (W,H,D) or volume edges $\{\mathbf{W},\mathbf{H},\mathbf{D}\}$ and resolution (r_x,r_y,r_z). A 3D point's coordinates in scene frame $\mathbf{M}$ and associated volume index $\mathbf{g}$ are related by (4.1) using the transformation matrix $\mathbf{G}$ as illustrated with step

1 in figure 4.2 considering that the scene and volume frame have the same orientation parameters:

$$\begin{pmatrix} \mathbf{M}_s \\ 1 \end{pmatrix} \sim \mathbf{G} \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \text{ with } \mathbf{G} = \begin{pmatrix} \frac{\mathbf{W}}{r_x} & \frac{\mathbf{H}}{r_y} & \frac{\mathbf{D}}{r_z} & O_v \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

- calibration results for rectified cameras of the chosen multiscopic unit.
  More precisely, we use the extrinsic (we call here **E**) and intrinsic **K** matrices of the rectified reference camera $\{i_{ref} = 0\}$ (see steps 2,3,4 in figure 4.2). These matrices are described in detail in sections 2.1.1.1 and 2.1.1.2. We re-write these matrices and the geometrical relationship between 3D point's coordinates in camera frame $\mathbf{M}_c$ and its projection into image plane **m** as:

$$\mathbf{E} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ & 1 \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} \alpha_u & s & u_0 \\ & \alpha_v & v_0 \\ & & 1 \end{pmatrix} \tag{4.2}$$

$$\begin{pmatrix} \mathbf{M}_s \\ 1 \end{pmatrix} = \mathbf{E} \begin{pmatrix} \mathbf{M}_c \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} \mathbf{m} \\ 1 \end{pmatrix} \sim \mathbf{K} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{M}_c \\ 1 \end{pmatrix} \tag{4.3}$$

  Using Equations 4.3 and 4.1, the relation between $\mathbf{M}_c$ and **g** can be written as:

$$\begin{pmatrix} \mathbf{M}_c \\ 1 \end{pmatrix} \sim \mathbf{E}^{-1} \mathbf{G} \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \tag{4.4}$$

  Moreover, the relation between the projection **m** and **g** is written using the Equations 4.3and 4.4 as:

$$\begin{pmatrix} \mathbf{m} \\ 1 \end{pmatrix} \sim \mathbf{K} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \end{pmatrix} \mathbf{E}^{-1} \mathbf{G} \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \tag{4.5}$$

- conversion of local depth Z from reference camera $\{i_{ref} : \text{ with } ref = 0\}$ to disparity δ such that $Z(\delta + \bar{\delta}) = f\ b \Leftrightarrow Z\delta = -Z\bar{\delta} + f\ b$ described in section 2.1.2.2.2.
  The DS index **t** is thus obtained from equations 4.2, 4.3, and 4.4 by adding a convenient row (the red row in Equation 4.7) in $\mathbf{K} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \end{pmatrix}$ which adds δ, computed from equation 2.28, to its usual $\mathbf{t} = (\mathbf{m}^t = (u,v),1)^t$ output. We call this new matrix $\mathbf{K}'$ (see step 5 in figure 4.2). Therefore, the equation 4.4 with K' yields the intended equations and matrices **DSfV** and **VfDS** transforming respectively coordinates from VH to DS

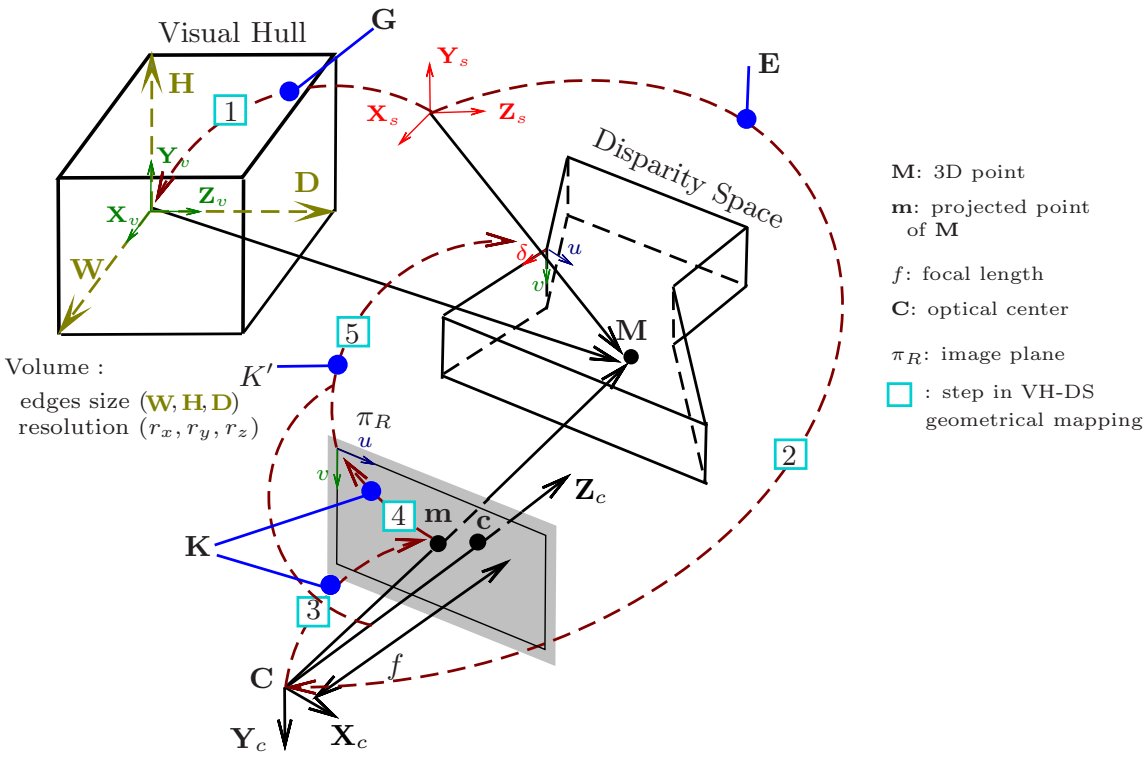


Fig. 4.2 Transformation from VH coordinates to DS coordinates: a five steps mapping.

in equation 4.6 and vice versa in equation 4.7.

$$\begin{pmatrix}\mathbf{t}\\1\end{pmatrix} \sim \underbrace{\underbrace{\begin{pmatrix}\alpha_u & s & u_0 & \\ & \alpha_v & v_0 & \\ & & -\delta & f\cdot b\\ & & 1 & \end{pmatrix}}_{\mathbf{K'}} \times \mathbf{E}^{-1} \times \mathbf{G}}_{\mathbf{DSfV}} \times \begin{pmatrix}\mathbf{g}\\1\end{pmatrix} \tag{4.6}$$

$$\begin{pmatrix}\mathbf{g}\\1\end{pmatrix} \sim \mathbf{VfDS} \times \begin{pmatrix}\mathbf{t}\\1\end{pmatrix} \quad \text{with} \quad \mathbf{VfDS} \equiv \mathbf{DSfV}^{-1} \tag{4.7}$$

4.3 Multi-baseline stereovision guidance by VH

Let us recall that our previous multi-baseline stereovision framework proposed in chapter 2 was developed without any VH usage. This section exposes how VH guidance is added to enhance its performances (see figure 4.3).

4.3.1 Core principle

The core classical idea behind the VH guidance is the fact that the reconstruction solution is necessarily included in the visual hull as any point out of VH was labelled as such because it was projected outside at least one silhouette. Handling this crucial information involves mapping VH and target point spaces using equations 4.6 and 4.7. Furthermore, these equations are likely to deliver homogeneous real coordinates in destination space which is populated on a bounded discrete 3D grid. Evaluating a map defined in one space for a sample of the other space is thus achieved for the intended sample via tri-linear interpolation at the resulting coordinates in map space. In order to keep notations simple we introduce three different bracketing schemes dedicated to direct 3D integer indexing, tri-linear interpolation and cross-space evaluation: angular bracketing $\langle\ \rangle$) expresses tri-linear interpolation at 3D real coordinates obtained using application of function $\mathcal{U}$ (see equation 4.10) on results of equation 4.6 or 4.7; round bracketing $(\ )$ is reserved for cross-space evaluation in the destination map; whereas direct map sample evaluation uses usual square bracketing $[\ ]$:

$$\mathbf{VH}(\mathbf{t}) \equiv \mathbf{VH}\left\langle \mathcal{U}\left(\mathbf{VfDS} \times \begin{pmatrix}\mathbf{t}\\1\end{pmatrix}\right)\right\rangle \tag{4.8}$$

$$\mathbf{DS}(\mathbf{g}) \equiv \mathbf{DS}\left\langle \mathcal{U}\left(\mathbf{DSfV} \times \begin{pmatrix}\mathbf{g}\\1\end{pmatrix}\right)\right\rangle \tag{4.9}$$

$$\mathcal{U}\left(\begin{pmatrix}\mathbf{v}\\a\end{pmatrix}\right) = \mathbf{v}/a \tag{4.10}$$

4.3.2 Bounding DS domain

The multi-baseline stereovision framework proposed in chapter 2 works on a 3D grid laid on disparity space DS and indexed by $\mathbf{t} = (u, v, \delta)^t$. As such, this grid has to be bounded as close as possible to useful areas where the solution is expected to lie. Without any such prior information, which is usual in "pure" multi-view stereovision (i.e. without VH), some lateral limits are easily set in u and v according to image frustums, but the disparity bounding is more of an issue as disparities could theoretically be spread over a wide range. In most methods, no information is available about the disparity limits and the disparity range is usually required as an input parameter providing the missing DS boundaries.

As stated above, VH is defined in a bounded 3D grid and may be seen as a superset of the actual solution. This information is crucial as it situates the solution (DS where a scene can be reconstructed) in a finite and closed area of scene space usually close to the actual solution. As such, this information yields opportunities to automize and optimize DS bounding.

Minimal and maximal DS coordinates of projections of the eight corners $\mathbf{g}_i$ of the VH grid, or even better, of the axis aligned bounding box (usually abbreviated AABB) of the VH solution yields a AABB in DS domain in which the solution is necessarily included. This AABB is identified by its min and max indices $\mathbf{t}_m$, $\mathbf{t}_M$ in DS as follows:

$$\left.\begin{array}{l} \mathbf{t}_m = floor\left(\min\limits_{i=0,\ldots,7} \mathbf{t}_i \equiv \begin{pmatrix} min_i\mathbf{u}_i \\ min_i\mathbf{v}_i \\ min_i\delta_i \end{pmatrix}\right) \\ \mathbf{t}_M = \ ceil\left(\max\limits_{i=0,\ldots,7} \mathbf{t}_i \equiv \begin{pmatrix} max_i\mathbf{u}_i \\ max_i\mathbf{v}_i \\ max_i\delta_i \end{pmatrix}\right) \end{array}\right\} \text{with } \mathbf{t}_i = \begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \\ 1 \end{pmatrix} = \mathcal{U}\left(\mathbf{DSfV} \times \begin{pmatrix} \mathbf{g}_i \\ 1 \end{pmatrix}\right) \tag{4.11}$$

This step indeed automatizes the DS bounding as no user input is required to set disparity limits. Furthermore, it even optimizes in lateral dimensions as the VH bounding box may appear thinner than the available views. Nevertheless, this first AABB is further optimized according to VH information. A sweeping process is run on each of its six faces, moving them inwards as long as they contain only target points whose interpolation in VH are considered *out*. This supposes (i) that the VH is defined on the grid as a numerical map **VH** with numerical values monotonically (let us suppose increasingly) associated to $in, surf, out$ semantic labels and (ii) that some interpolation threshold out_t is set. A target point indexed by $\mathbf{t}$ is thus considered out of the VH according to its interpolation in **VH** using the function $\mathcal{O}ut(\mathbf{t})$ defined as thresholding of cross-space evaluation in **VH** map as follows:

$$\mathcal{O}ut(\mathbf{t}) \equiv \mathbf{VH}(\mathbf{t}) \geq out_t \tag{4.12}$$

This double process reduces to optimal AABB the DS domain on which the different maps are laid (allocated), which thus optimizes memory and computational efficiency.

4.3.3 Filtering target points according to VH

The previous VH guidance for DS bounding has an actual but rather low impact on reconstruction quality as it eliminates some potential outliers outside the final AABB. Moreover, many more outliers are to be avoided if we remember that the target points have to lie inside VH volume.

A simple preprocessing step labels every target point in the optimized AABB as undoubtedly outside or possibly inside the solution according to its VH interpolation $\mathcal{O}ut(\mathbf{t})$ (equation 4.12). Target points labelled as outside (see figure 4.4 and details (1) in figure 4.4) will neither be given similarity scores, nor be considered for matching in the multi-baseline stereovision process. They will only be used as definitely non material points ($\mu[\mathbf{t}] = 0$)

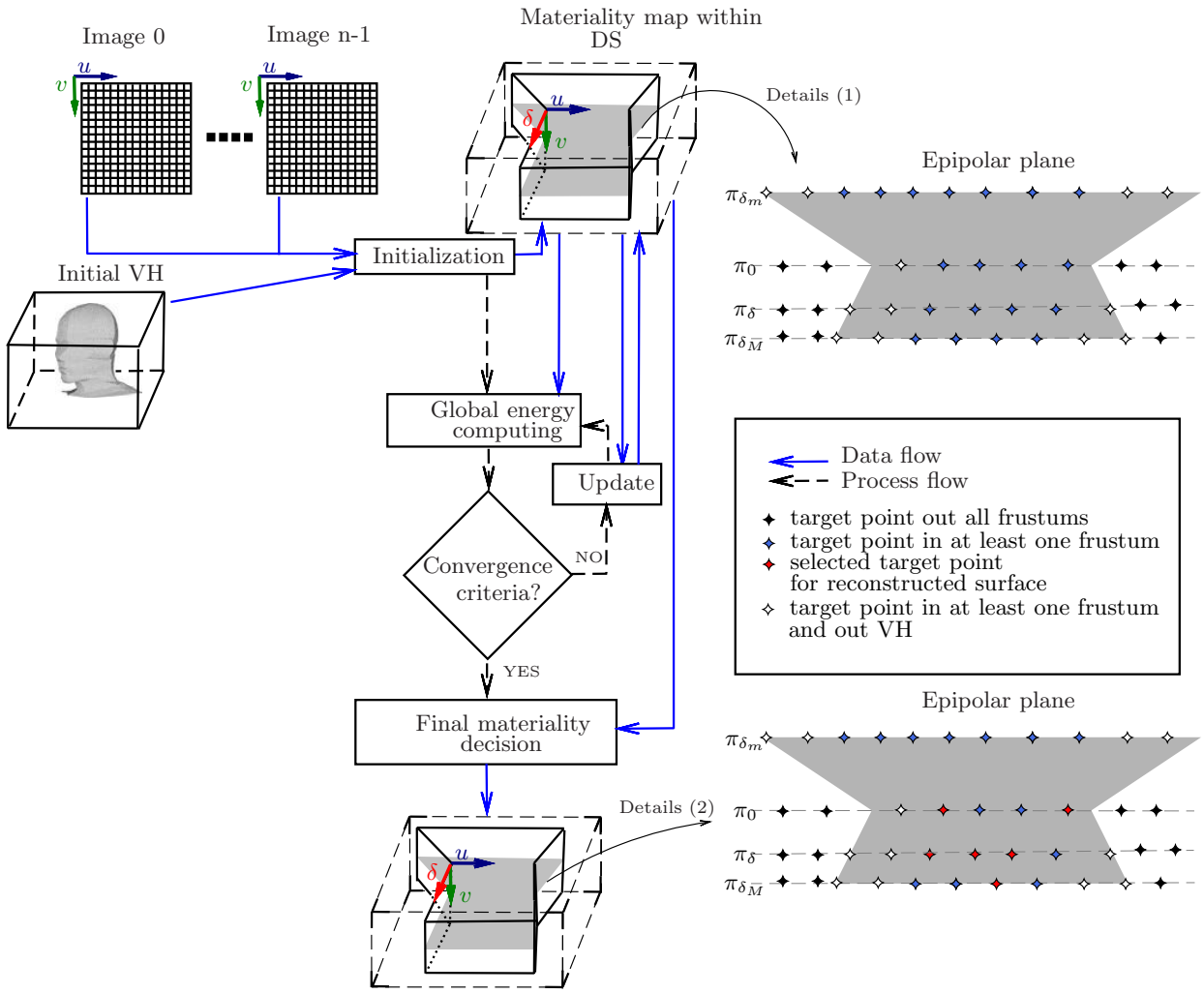


Fig. 4.3 Guidance by VH of materiality map method: framework pipeline.

for visibility reasoning purposes. These target points labelling enhances computational efficiency. Moreover, it restricts the solution domain and avoids evaluation of potential outliers lying in AABB but outside VH, which directly impacts reconstruction quality as illustrated in figure 4.11.

4.3.4 Enhancing similarity quality

In section 3.4, we presented different methods to compute the similarity scores for each target point describing the benefits of each of those methods: "non adaptive flat windows", "separate windows", and "weighted windows". However, the similarity computation for a target point can be enhanced using target point labelling: as this computation implies local constant disparity assumption, it is reasonable to exclude neighboring target points in the

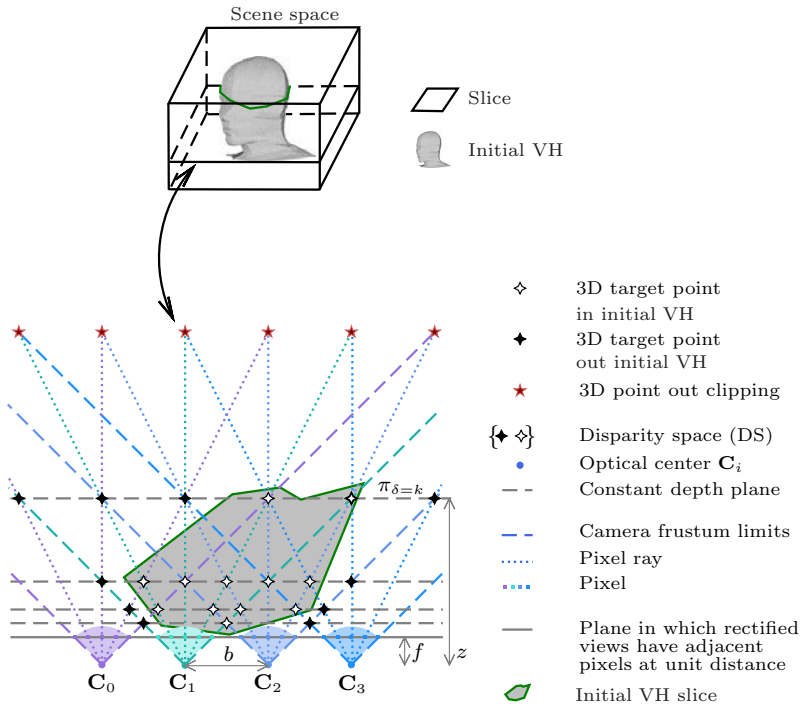


Fig. 4.4 Filtering target points: schematic representation for one epipolar plane slice of volume and its mapping in disparity space frame, the rhombuses with black and white color refer to the target points *"in"* and *"out"* of visual hull respectively.

constant disparity plane which are labelled outside the VH. Such neighboring samples are filtered out of the adaptative window before similarity computation. This ensures that neighbors known as irrelevant do not hinder the similarity scores computation. Those similarity scores are thus more relevant, enhancing the reconstruction quality and robustness.

4.4 Carving VH from stereovision

Our visual hull voxels are labelled as *in*, *out*, and $surf$. However, multi-baseline stereovision yields a surface composed of the 3D points valued 1 in the binary materiality map. Each such point also has a final confidence score related to its confidence scores (illustrated in section 3.5) associated to its similarities and possibly its comparison to other target points on its pixel rays. Therefore, merging both models results in the intersection between the VH and the complement of the space between the multiscopic unit and the reconstructed surface. This corresponds to the subtraction or carving from VH of the space between multiscopic unit and surface as illustrated in figure 4.5.

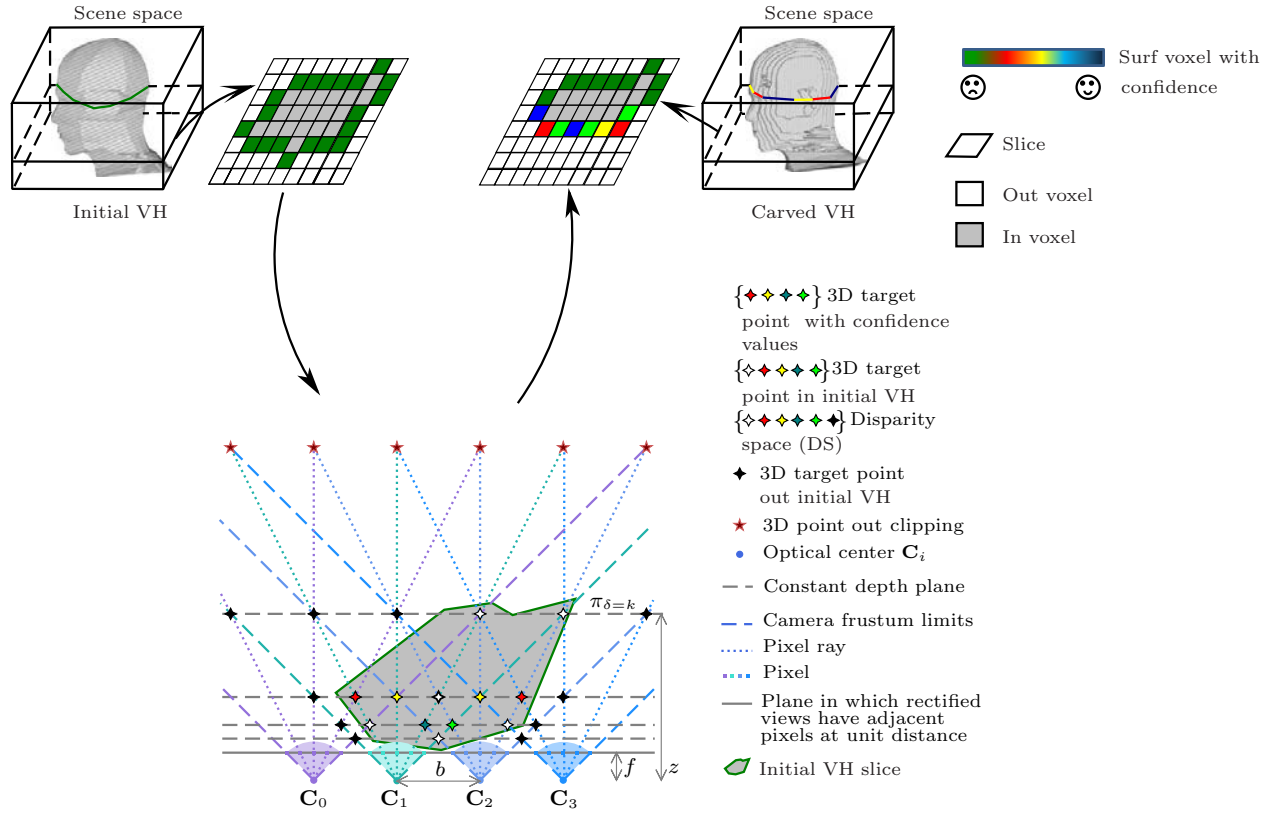


Fig. 4.5 Carving VH from stereovision: slice of initial VH represents *"in"*, *"out"* and *"surf"* voxels with gray, white, and green color respectively. The reconstructed target points are colored from green to blue, according to their confidence score (see color table on top right). Slice of carved VH consists of *"in"*, *"out"* and *"surf"* voxels with different confidence for the *"surf"* voxels derived from its corresponding target points in disparity space.

4.4.1 Stereovision surface coding

Precise definition of the space *"between"* the reconstructed surface and the multiscopic unit is not straightforward: this is a continuous space containing and interpolating, for every view of the unit, every part of the ray going from the optical center to any solution point which is not occluded in this view. Most of those rays are redundant across the different views and we chose, for the sake of simplicity, to replace all these view dependent segments by others, far less numerous and redundant, attached to the same solution points but coming from a single center located at the middle of the multiscopic unit. A drawback of this simplification may lie in a loss of solution points which could become occluded in this *virtual central view*. However, as a solution point has to be seen in at least a couple of successive views, this loss does not occur when $n < 5$ because the occluding rays of a solution point are limited to 0 to $n-2$ extreme views. As such, the central ray cannot

be flanked by two occluding rays ($n = 4$) or be itself occluding the solution point ($n = 3$). This remark enforces our chosen compromise to have $n = 4$. As shown in section 3.3, we choose the reference image $i_{ref} = 0$ for coding efficiency. In this chapter, we decide to build our surface representation according to a central and symmetrical sampling. This domain is called Central Disparity Space, abbreviated as CDS, and indexed in reference of the (virtual) central view (see figure 4.6). This central space is less biased in 3D space than any other, and thus interpolation in CDS will be more relevant. According to the multiscopic geometry (see chapter 2), this (virtual) central view corresponds to a camera indexed $i_c \equiv (n-1)/2$. Hence, a target point of index (u,v,δ) in DS would project in the central view at $(u_{i_c}, v_{i_c}) = (u + (i_0 - i_c)\delta, v)$ (see equation 3.1). In order to keep integer indices when n is even (as for our choice $n = 4$), we multiply the horizontal coordinate in CDS by $\gamma = 2 - n \; mod \; 2$. These remarks lead to new matrices managing the transformations between coordinates $\mathbf{t} = (u,v,\delta)^t$ in DS and $\mathbf{c} = (c,v,\delta)^t$ in CDS and between VH and CDS:

$$\begin{pmatrix} \mathbf{c} \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \gamma & & \gamma(i_0 - i_c) & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}}_{\mathbf{CfR}} \times \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix}, \quad \gamma = 2 - n \; mod \; 2 \tag{4.13}$$

$$\begin{pmatrix} \mathbf{c} \\ 1 \end{pmatrix} \sim \underbrace{\mathbf{CfR} \times \mathbf{DSfV}}_{\mathbf{CDSfV}} \times \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \quad , \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix} \sim \underbrace{\mathbf{CDSfV}^{-1}}_{\mathbf{VfCDS}} \times \begin{pmatrix} \mathbf{c} \\ 1 \end{pmatrix} \tag{4.14}$$

In this CDS, we decide to represent the solution surface as a disparity map **DM** tagged by a confidence map **CM** (see figure 4.6). This is achieved by assigning for each solution point in DS, from far to near, at its CDS *pixel coordinates* (c,v), its disparity δ to **DM** (initialized to $-\infty$) and its associated final confidence score to **CM**. When n is even (which in the case in RECOVER3D), gaps are induced between CDS neighbors by the horizontal stretching in CDS. To fill those gaps, if two successive target points on a row of CS are both solution, their middle point in CDS is assigned their common disparity in **DM** and mean confidence in **CM**. As the solution in CS is computed in a way to ensure that its intersection with any (u,δ) plane is a continuous suite of adjacent target points that are of same or adjacent disparities, no other gap may occur.

4.4.2 Carving VH from disparity map

Carving the VH according to the stereovision surface coded by **DM** and **CM** is described in Algorithm 8 and illustrated by Figure 4.7. Carving VH from disparity map **DM**

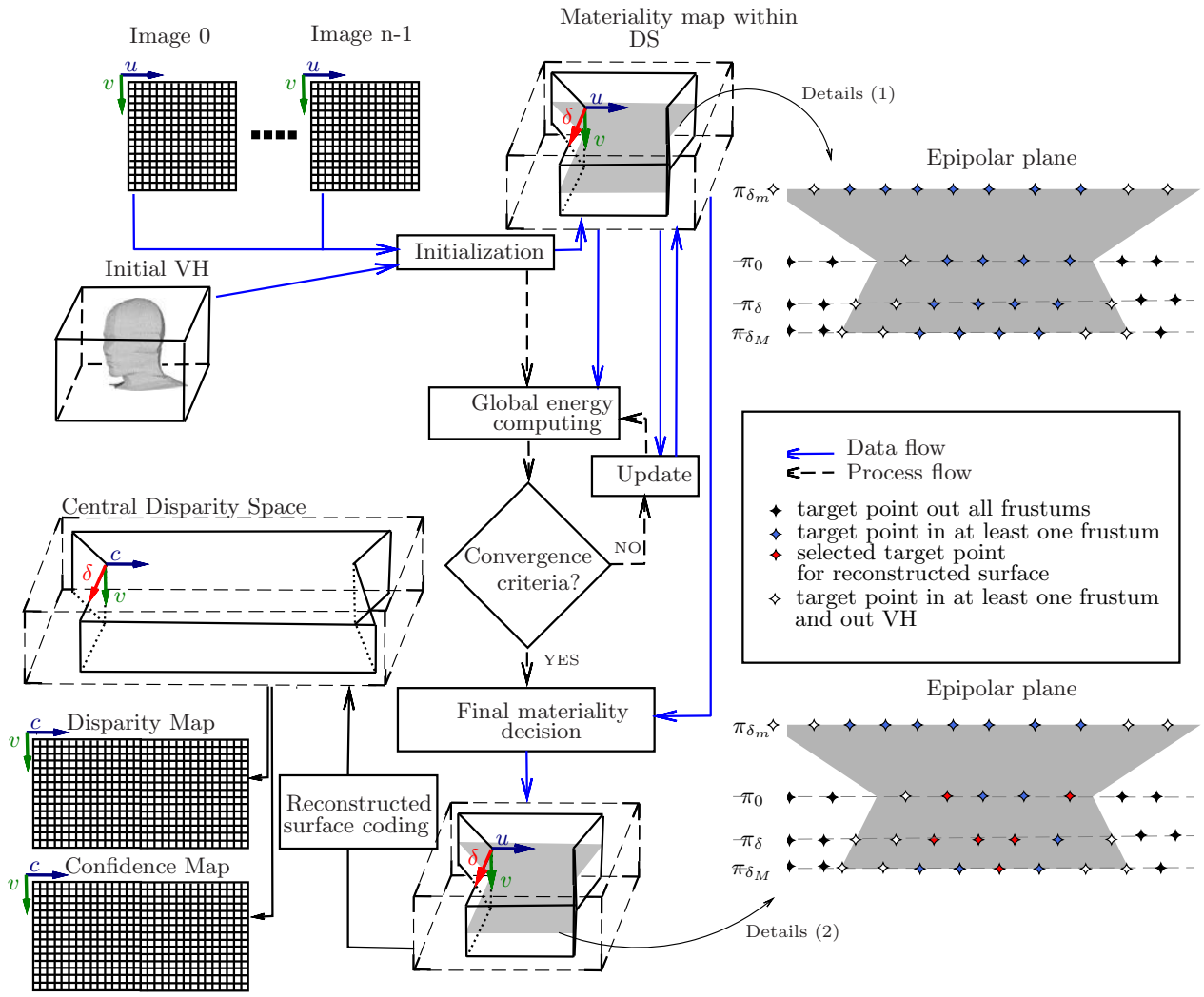


Fig. 4.6 Building the solution surface as a disparity map and confidence map according to a Central Disparity Space.

aims at filling a carved volume defined as a map **CV** laid over the VH grid and valued $in, surf_0..surf_q, out$. The different $surf_i$ values refer to increasing quantified confidence levels for surface voxels. The lowest confidence level $surf_0$ is reserved for $surf$ voxels of VH that are either occluded or out of frustum for the current solution. The other levels are associated with voxels identified as $surf$ in the stereovision solution: the effective level i is quantified according to the interpolated **CM** value of the voxel.

A key feature of this step for the latter fusion process is to yield a coherent topology to our carved volumes: *in* and *out* sets are considered in 6-connexity while $surf_{\{0...q\}}$ is considered in 27-connexity. With such topological evaluation, no direct 6-connexion should occur between *in* and *out* voxels.

Voxels in visual hull are looped over. Any voxel labelled *out* in **VH** is so labelled in **CV**. Each voxel $\mathbf{g} = (w,h,d)^t$ labelled *in* or $surf$ in **VH** is projected as $\mathbf{c} = (c,v,\delta)^t$ in CDS. Its disparity δ is then compared to the disparity of the solution on the same central ray $\delta_s = \mathbf{DM}\langle(c,v)^t\rangle$.

In order to handle the grid sampling while responding to the previous intended topological property, point comparison in CDS is related to actual axis aligned distance $\| \ \|_\infty$ in VH. Hence the interpolated solution point $\mathbf{c}_s = (c,v,\delta_s)^t$ is projected back in VH to measure its distance to initial voxel $\|\mathbf{g} - \mathbf{VfCDS} \times \mathbf{c}_s\|_\infty$. When this distance is no more than 1, $\mathbf{g}$ is labelled $surf$ in the carved volume with a confidence level quantified from tri-linear interpolation result $\mathbf{CM} < (c,v)^t >$. Otherwise, if the voxel $\mathbf{g}$ is in front the surface ($\delta > \delta_s$), it is labelled *out* in **CV**. In the remaining cases, the voxel is *a priori* labelled *in* or $surf_0$ according to VH label but could be labelled $\{surf_i : i > 0\}$ if it lies close enough of a steep slope of the surface. In order to check this possibility, we evaluate if any of its 4 neighbors in CDS c,v axes of same disparity δ, at unitary distance in VH, are to be considered *out* (with interpolated disparity lower than δ). This evaluation consists in measuring the distance lg in VH from the initial voxel to a neighbor $\mathbf{n}_0$ at unitary distance in CDS and then interpolating disparity δ_n in **DM** at a neighbor $\mathbf{n}_c$ in same direction but distance lg^{-1}. If $\delta_n < \delta$, this neighbor is considered *out* and the initial voxel is re-labelled $surf_i$ where the confidence level i is quantified from the disparity linear interpolation at δ of $\mathbf{CM}\langle\mathbf{n}_c\rangle$ at δ_n and $\mathbf{CM}\langle(c,v)^t\rangle$ at δ_s.

4.4.3 Improving surface smoothness

The result of the multi-stereovision method leads to a discontinuous surface divided into frontal planar patches with constant and integer disparity (see first and fourth rows of figure 4.12). Removing this effect is required for the visual quality of the 3D model result and for a more accurate management of reconstruction incoherencies between different multiscopic units. To deal with this problem coming from the restriction to integer disparities in reconstruction process, we propose to represent the solution surface previously saved in **DM** by a floating point derivated version $\mathbf{DM}_r$. The map $\mathbf{DM}_r$ is computed to ensure continuous transitions between adjacent horizontal segments of constant disparities with a disparity gap of 1. Computing $\mathbf{DM}_r$ consists in looping over rows v of **DM** that are thus scanned from one end to the other to identify disparity steps between adjacent pixels of finite disparity. When the disparity step is of magnitude 1 $(-1,+1)$, a contact point (black point in figure 4.8) is placed in CDS in the middle of the two pixels with the mean of their disparity values as illustrated in figure 4.8, and serves as end point of both segments. When a disparity gap is more than 1 (notably infinite) as well as for first and last pixels, a single end point is generated on the relevant pixel at its (finite) disparity. This process yields two end points

Algorithm 8: Carving VH by central disparity map

$\mathbf{c} \equiv (c, v, \delta)$ $\quad$ $\mathbf{N4} = \{(-1,0),(1,0),(0,-1),(0,1)\}$
foreach g in **VH** domain **do**
  if VH[**g**] is *in* or $surf$ **then**
    c = $\mathcal{U}(\mathbf{CDSfV} \times (\mathbf{g}^t, 1)^t)$
    if (c, v) in **DM** domain **then**
      δ_s= **DM** $\langle (c,v)^t \rangle$ $\quad$ $\mathbf{g}_s = \mathcal{U}(\mathbf{VfCDS} \times (c, v, \delta_s, 1)^t)$
      if $(\|\mathbf{g}_s - \mathbf{g}\|_\infty) \leq 1$ **then**
        CV[**g**] = $surf_{Quant(\mathbf{CM}\langle (c,v)^t \rangle)}$
      else
        if $\delta s < \delta$ **then**
          CV[**g**] = *out*
        else
          if VH[**g**] is *in* **then**
            CV[**g**] = *in*
          else
            CV[**g**]=$surf_0$
          foreach $n \in [0,4[$ **do**
            $lg = \|\mathcal{U}(\mathbf{VfDS} \times ((\mathbf{c}^t, 1) + (\mathbf{N4}[n], 0, 0))^t) - \mathbf{g}\|_\infty$
            $\mathbf{n}_c = (c, v)^t + \mathbf{N4}[n]/lg$
            if $\mathbf{n}_c$ in **DM** domain and $(\delta_n = \mathbf{DM}\langle \mathbf{n}_c \rangle) < \delta$ **then**
              $cnf = (\mathbf{CM}\langle (c,v)^t \rangle (\delta - \delta_n) +$
                $\mathbf{CM}\langle \mathbf{n}_c \rangle (\delta_s - \delta)) \;/\; (\delta_s - \delta_n)$
              CV[**g**] = $surf_{Quant(cnf)}$
    else
      if VH[**g**] is *in* **then**
        CV[**g**] = *in*
      else
        CV[**g**]=$surf_0$
  else
    CV[**g**] = *out*

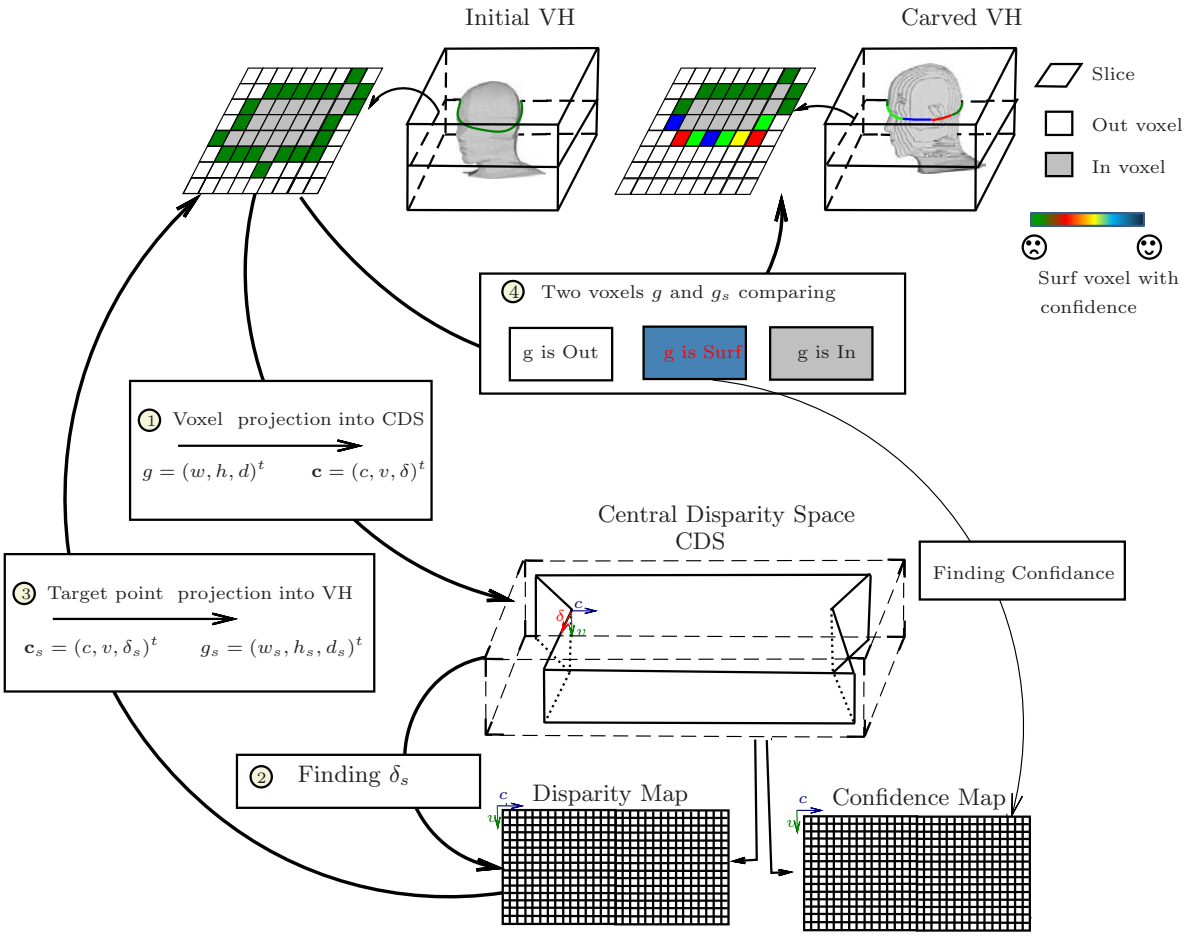


Fig. 4.7 pipeline of carving VH algorithm by disparity map.

per segment expressed in CDS (c_0, v, δ_0) and (c_1, v, δ_1). When a right end point (c_1, v, δ_1) is generated, the corresponding segment of initial constant disparity δ is filled in $\mathbf{DM}_r$ by a dedicated interpolation scheme between the end points

$$\begin{aligned}\mathbf{DM}_r[(c,v)^t] = \delta + (1-t)(2t-1)(\delta-\delta_0) \\ + \; t\cdot(2t-1)(\delta_1-\delta)\end{aligned} \quad , \quad t = \frac{c-c_0}{c_1-c_0}. \tag{4.15}$$

The interpolation function in equation 4.15 is ensured to pass through the central sample (t=$\frac{1}{2}$) and both end points $t \in \{0,1\}$ (see figure 4.8 where the black double lined curve expresses the interpolation function that yields the interpolated disparities in $\mathbf{DM}_r$). When δ_0 and δ_1 are both under or both above δ, or if one only of them equals δ (indicating large disparity gap or start/end point), this interpolation is parabolic and the equation 4.15

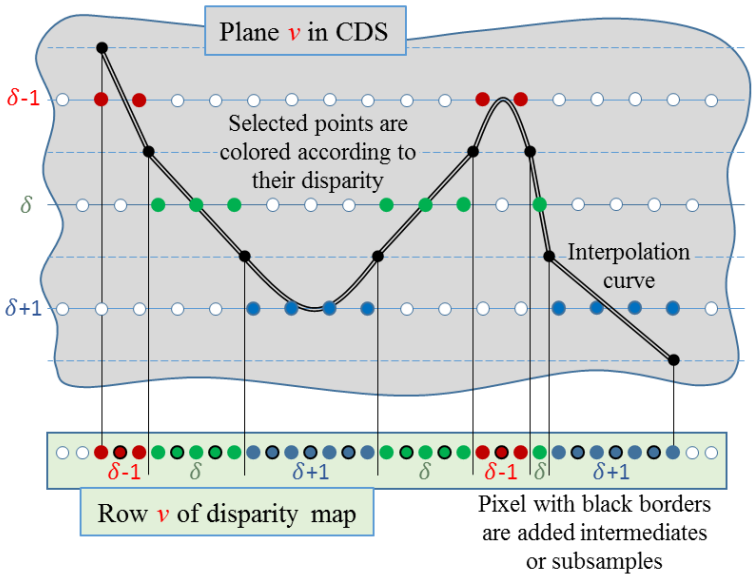


Fig. 4.8 Disparity interpolation: relation between disparity map **DM** (colored points) and interpolated disparity map $\mathbf{DM}_r$ illustrated in CDS by the interpolation function (black double line curve). The results of this process are illustrated in the second and fifth rows of figure 4.12.

is written as follows:

$$\mathbf{DM}_r[(c,v)^t] = \begin{cases} \delta - (2t-1)^2(\delta - \delta_0), & \delta_1 = \delta_0 \\ \delta + (1-t)(2t-1)(\delta - \delta_0), & \delta_1 = \delta \\ \delta + t(2t-1)(\delta_1 - \delta), & \delta_0 = \delta \end{cases} \tag{4.16}$$

When δ_0 or δ_1 is above and the other under, the interpolation is linear. Equation 4.15 is thus written:

$$\mathbf{DM}_r[(c,v)^t] = \delta + (2t-1)\varepsilon, \qquad \delta_1 - \delta = \delta - \delta_0 = \varepsilon \in \{-1/2, 1/2\}. \tag{4.17}$$

4.4.4 Smoothing using bilateral filter

The result of the disparity interpolation described in section 4.4.3 is a floating point disparity map more *continuous* or smooth on each row but still presenting numerous vertical depth steps. To handle this problem, a bilateral filter is applied on the disparity map $\mathbf{DM}_r$ to compute a smoothed disparity map $\mathbf{DM}_s$ as described in equation 4.18 and demonstrated in figure 4.12. The centered operating window is chosen rectangular as regulating transitions between segments implies a rather low width $2ww+1$ but reducing vertical depth steps

involves a much taller height $2wh+1$. Furthermore, as Equation 4.18 imply an overall normalizing factor $\sum_{\mathbf{n}\in W}\mathcal{W}(\mathbf{p},\mathbf{n})$, we chose not to normalize each individual gaussian function in individual weight computations.

$$\mathbf{DM}s[\mathbf{p}] = \frac{\sum_{\mathbf{n}\in W}\mathbf{DM}r[\mathbf{p}+\mathbf{n}]\ \mathcal{W}(\mathbf{p},\mathbf{n})}{\sum_{\mathbf{n}\in W}\mathcal{W}(\mathbf{p},\mathbf{n})} \tag{4.18}$$

with $\mathbf{n}=(dc,dv)^t$, $W=[-ww,ww]\times[-wh,wh]$ and

$$\mathcal{W}(\mathbf{p},\mathbf{n}) = \mathcal{G}_{\sigma_c}(dc)\ \mathcal{G}_{\sigma_v}(dv)\ wd(\mathbf{DM}_r[\mathbf{p}+\mathbf{n}]-\mathbf{DM}_r[\mathbf{p}])$$
$$\mathcal{G}_{\sigma}(t) = exp(-t^2/(2\sigma^2))$$
$$wd \text{ a function decreasing from 1, for example}$$
$$wd(\Delta\delta) = \sigma_\delta^2/(\sigma_\delta^2+\Delta\delta^2)$$

4.5 Omnidirectional 3D modeling

4.5.1 Merging difficulty

The final step of the 3D reconstruction consists in merging carved VH volumes $\mathbf{CV}_m$ from multi-baseline stereovision results for all multiscopic units m (see figure 4.9) in order to obtain a single 3D model representing the 3D pose of the reconstructed object(s).

Figure 4.12 illustrates that the result of each multiscopic unit provides information only on visible surfaces facing the unit while other surface areas are derived from VH result. Multiple carved VH from different multiscopic units spread around the scene thus yield stereovision details for almost every surface area of the model.

However, parts of the model surface are to be seen and reconstructed by multiple multiscopic units and these independant reconstructions are usually partially inconsistent one to another. Therefore, in such inconsistently reconstructed areas, we have to decide which reconstruction is locally kept in the final solution. This decision is based on the confidence attribute of surface voxels: as stated in section 4.4.2, surface voxels in $\mathbf{CV}_m$ bear different labels $surf_i$ indicating their quantified confidence level according to the stereovision process.

4.5.2 Merging process

The overall principle of this final step is to initialize the final merged volume **FV** to one of the carved VH ($\mathbf{FV}=\mathbf{CV}_{m_0}$) and then to iteratively merge each other carved VH $\mathbf{CV}_m$ into **FV** according, in inconsistently labelled areas, to decisions based on confidence scores of surface voxels.

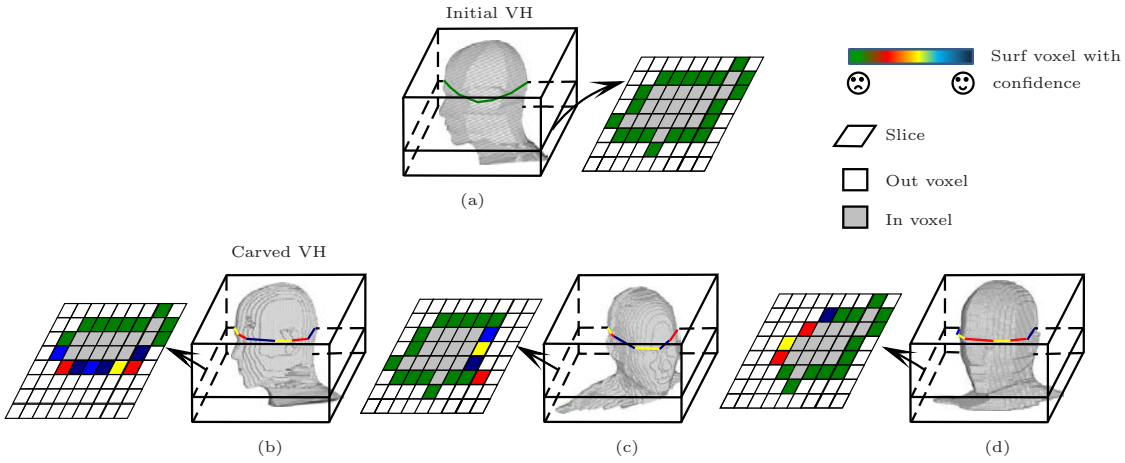


Fig. 4.9 Carved VH using multi-baseline stereovision method applying on 3 different multiscopic units. a) Initial VH and slice that represent the voxels with three label (*"in"*, *"out"* and *"surf"*). b,c,d) Three different carved VH and slices that refer to the voxels with three labels and the confidence level.

As VH is known to be a super-set of the solution, the process only evaluates voxels labelled *in* or $surf$ in **VH**. It thus loops over every voxel **g**, treating each one for which $\mathbf{VH}[\mathbf{g}]$ is not *out* according to its labels $\mathbf{FV}[\mathbf{g}]$ and $\mathbf{CV}_m[\mathbf{g}]$ (see figure 4.10):

- both *out*: voxel **g** is kept *out* in **FV**
- both *in*: voxel **g** is kept *in* in **FV**
- $surf_i$ and $surf_j$: voxel **g** is kept the label $surf$ with the highest confidence level $\mathbf{FV}[\mathbf{g}] = surf_{max(i,j)}$
- all other cases: voxel **g** has inconsistent labels, the global loop is suspended while an inconsistency resolution process is run from **g**.

In the last case, to decide which label is to keep, we propose a global evaluation of the 6-connected area implied in the detected inconsistency rather than a per voxel decision. Thus, when a voxel **g** is detected as inconsistent in the global loop, a two pass process starts in order to make a decision.

The first pass aims at collecting relevant information to help making the right decision. It goes from **g** through its inconsistent 6-connected area in order to compute each confidence level histograms of the encountered surfaces of both volumes. These confidence histograms for the two surfaces help making the decision on which volume **FV** or $\mathbf{CV}_m$ will transfer its labels to the final solution in this 6-connected area. We propose to choose the volume with the highest mean confidence level, but other competing scores could easily be proposed and tested based on the confidence histograms.

When the decision is made, a second pass is run. The same walk-through is performed over the area of 6-connected voxels with inconsistent labels in order to resolve the inconsistency by copying labels of the chosen volume into the other. One could have thought that, when the chosen volume is **FV**, nothing needs to be done, but the first pass and the decision making would then be repeated for every voxel of the area which is far from efficient. Moreover, during this second pass, when a voxel labelled $sur\,f_i$ and $sur\,f_j$ is encountered, its best confidence level $max(i, j)$ is kept in both volumes.

This process clearly relies on a consistent topology in both volumes. This point is ensured by the VH carving step described in section 4.4.2. This topological consistency further permits to keep our 6-connected area walk-through topologically consistent: the process starts from an inside position (*in* or $sur\,f_j$) in one of the volumes $\mathbf{V}_i$ and an outside position (*out* or $sur\,f_i$) in the other volume $\mathbf{V}_o$. This per volume topological consistency has to be ensured over the whole process by adding to the studied area only neighbor voxels with different labels topologically consistent with the starting condition. No shift from *in* label to *out* label should occur in each volume across a 6-connection. Thus, ensuring topological consistency consists in avoiding 6-connections transgressing initial inside/outside position in each volume. This could occur in $\mathbf{V}_i$ for voxels on surface connected to *out* voxels as in $\mathbf{V}_o$ for voxels on surface connected to *in* voxels.

4.5.3 Refinements

The rough application of the section 4.5.2 process appears not totally successful as the walked-through areas sometimes appear as several rather broad and distant *blobs* of non surface voxels connected by thin lines or surfaces. The decision is made once for the whole area while it should be differenciated for each blob and connection line or surface. This yields inconvenient decisions which need to be corrected.

In order to do so, we apply several times the merging process of section 4.5.2 (3 times in the present implementation) with less and less restrictive conditions on inconsistent voxels:

1. Considered voxels have to be labelled in/out or out/in. Furthermore a sufficient part of their 6-neighbors has to be labelled in the same way (at least 40% in our implementation). This step treats broad in/out blobs.
2. Considered voxels are the remaining in/out or out/in ones. This step treats rather thin areas.
3. Considered voxels are any other inconsistent ones. This steps finalizes the resolution and treats very thin areas with no more (in, out) or (out, in) voxel.

Results from this refinement are illustrated in figures 4.13 and 4.14.

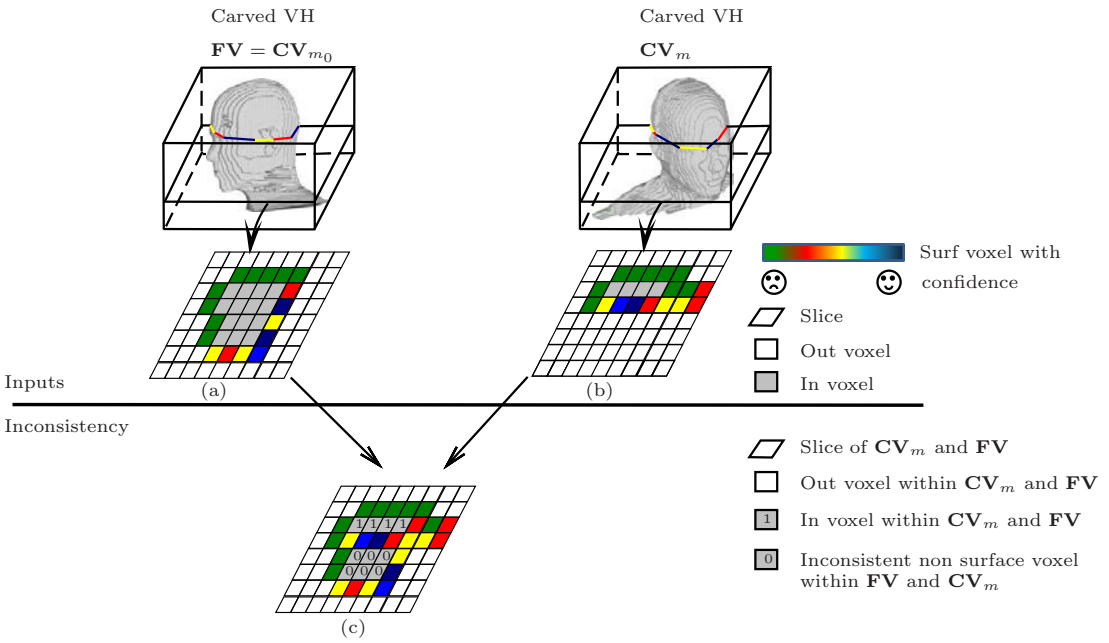


Fig. 4.10 Merging two carved VH volumes: a,b) two slices of two different carved volumes representing the confidence levels for the *"surf"* voxels and the *"in"*, *"out"* voxels. c) superposition of the two slices a,b exhibiting an area with inconsistent labels.

4.6 Results and discussion

To evaluate our framework described in figure 4.1, we used the studio layout scheme presented in section 1.3.1 both for real and virtual shooting and applied our software framework to the views they produced. These experimental conditions apply to each result discussed in this section.

Figure 4.11 illustrates that the VH guided stereovision method described in section 4.3 improves the materiality map derived from our previous multi-baseline stereovision method (see chapter 3) by ridding it of outliers outside the visual hull. Moreover, in non specular textured or concave areas, the materiality map solution proves to be more accurate than the visual hull as illustrated in figures 4.12 a, which clearly shows that concavities such as eye cavities are carved out by our stereovision method both for virtual and actual shootings. Figures 4.12 show results of the carving process described in section 4.4 on two view sets: the first one, of a virtual actor "Simon", shot under ideal calibration conditions by computer graphics software, and the second one, of a real actor "Philippe", captured in our dedicaced studio. Comparing the carved volume to the point cloud on each row of these figures qualitatively validates our carving method. The evolutions obtained on both figures from each row to the next, demonstrate the relevance of the disparity interpolation and smooth-

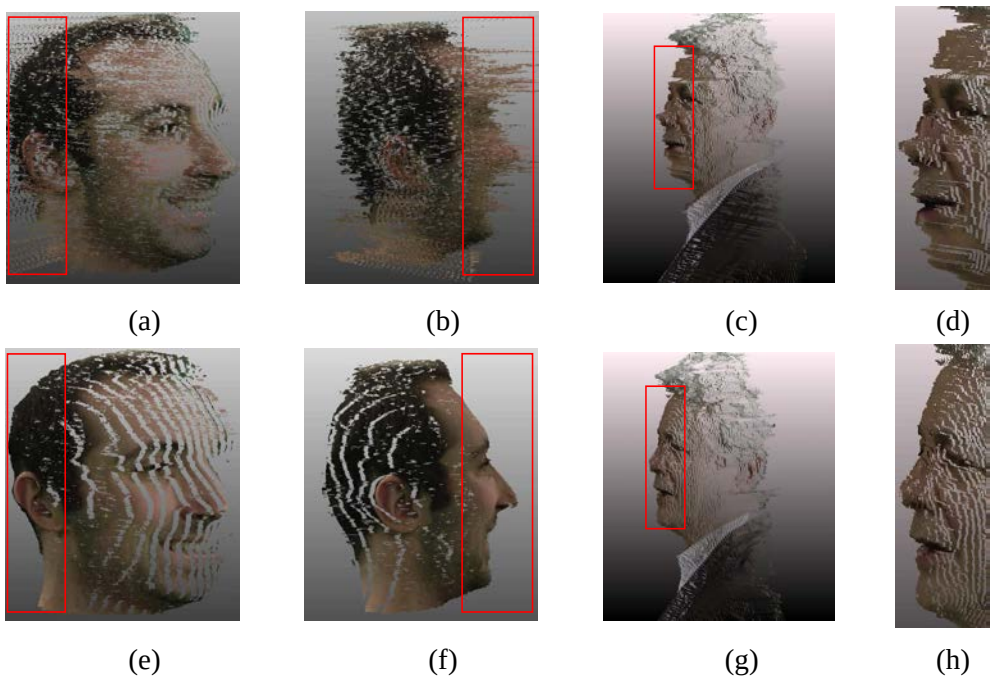


Fig. 4.11 First row: point clouds obtained with integer disparity values without any VH guidance for real actors. a, b) two different views for "Cédric". c, d) one view and its zoom in the red area respectively for "Jacques". Second row: similar values of point clouds obtained with integer disparity values with VH guided stereovision for the same data.

ing steps. The fusion of every multiscopic unit outcomes described in section 4.5 provides robust reconstruction especially in the areas where two or more multiscopic units compete. Figures 4.13 and 4.14 demonstrate this using results obtained respectively from a virtual dataset and a real dataset. One should notice the results quality despite the low number of used multiscopic units: 3 for actual shooting and 4 for the virtual one.

4.7 Conclusion

This chapter describes a new way of combining visual hull and multi-baseline stereovision in a fully automatic process. In section 4.3, we explained how to exploit information from the visual hull to guide the materiality map optimization process in order to increase its reconstruction accuracy, robustness and computational efficiency. It was demonstrated that our materiality map framework can integrate the visual hull guidance in a powerful way using its scene-based structure.

We also proposed in section 4.4 a new algorithm for VH carving from stereovision surface coded as a disparity map **DM**. This process yields a topologically consistent volume, which is crucial for many applications, including our further proposition of carved volumes

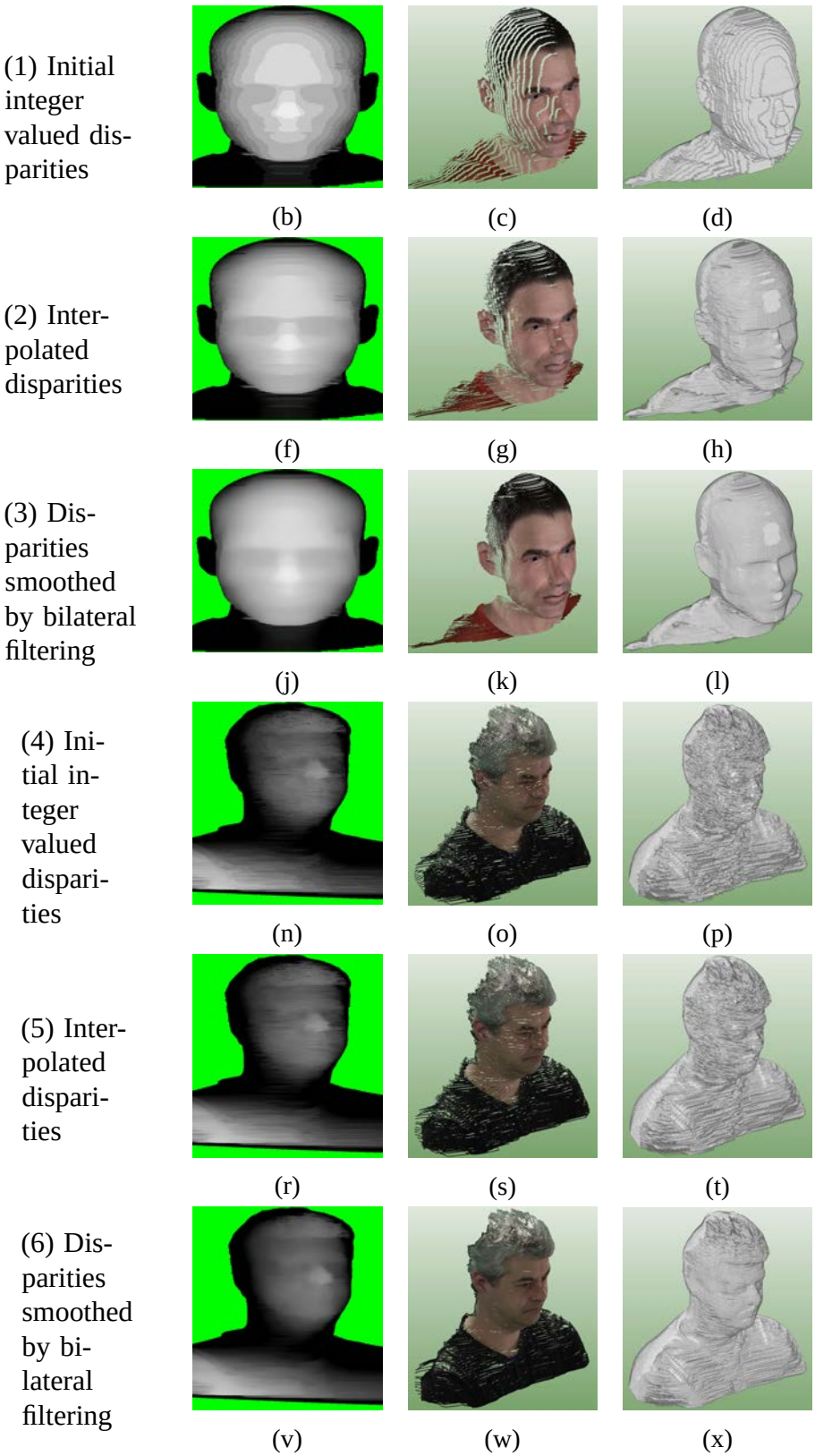
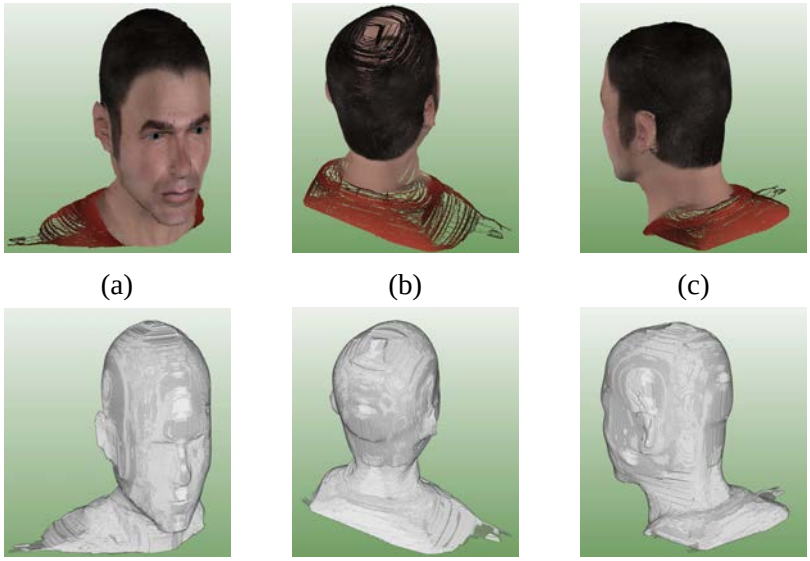


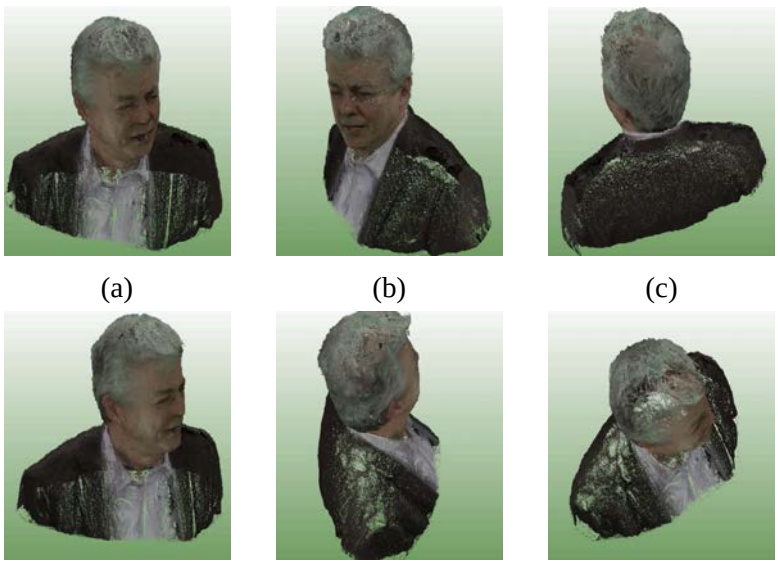
Fig. 4.12 Results from one multiscopic unit for virtual actor "Simon" (3 top rows) and real actor "Philippe" (3 bottom rows). From top to bottom, results with: initial integer valued disparities; interpolated disparities according to 4.4.3; disparities smoothed by bilateral filtering described in 4.4.4. On each row, from left to right: disparity map, point cloud, and carved volume.



(a) (b) (c)

(d) (e) (f)

Fig. 4.13 Results of the entire pipeline using VH and multiple multi-baseline stereovision reconstructions: several views of the point cloud and carved volume obtained from VH and four multiscopic units for virtual actor "Simon"



(a) (b) (c)

(d) (e) (f)

Fig. 4.14 Results of the entire pipeline: several views of the global point cloud obtained for real actor "Jacques" from final volume resulting from VH and three multiscopic units. It corresponds to the union of the projection, per multiscopic unit, of the initial point cloud on the final volume.

merging. We further showed on experimental examples both the algorithm results and the relevance of our disparity interpolation and smoothing methods.

Moreover, we proposed in section 4.5 a novel framework to merge multiple carved volumes obtained from different multiscopic units. We demonstrated the efficiency of the proposed inconsistency handling on both virtual shootings and actual shootings.

Altogether, these contributions, added to our previous stereovision framework proposed in chapter 3, yield a qualitative and robust omnidirectional 3D reconstruction tool to RECOVER3D project. The proposed solution proves the advantages of using both multiscopic and monoscopic cameras in a studio system as well as combining multi-baseline stereovision with VH approaches.

4.8 Résumé : Pipeline de fusion volumique des résultats issus des reconstructions multiscopiques avec l'enveloppe visuelle

Afin de reconstruire la globalité de la scène, nous proposons un pipeline de fusion à deux niveaux. Le premier se focalise sur la fusion des résultats issus d'une unité multiscopique avec l'enveloppe visuelle, tandis que le deuxième se charge de la fusion des résultats de toutes les unités multiscopiques. Au préalable, en nous replaçant dans le contexte du projet RECOVER3D, nous proposons une hybridation de notre méthode de stéréovision multi-oculaire tirant parti de la géométrie multi-épipolaire simplifiée et régulière afin d'en améliorer sa robustesse et son efficacité. En effet, notre méthode de stéréovision adaptée à de multiples caméras alignées ne délivre pas des résultats uniformément fiables, notamment dans les zones faiblement texturées ou avec un taux de redondance pauvre. La restriction de la zone de recherche des points cibles en utilisant l'enveloppe visuelle comme guide permet d'éliminer définitivement, en amont du processus de reconstruction, les points cibles candidats n'appartenant pas à cette enveloppe. Appliquée à une unité multiscopique du système de capture de RECOVER3D, nous obtenons une reconstruction 3D partielle de la scène. La surface obtenue est alors utilisée pour creuser l'enveloppe visuelle issue de la méthode basée silhouette. Seule la zone du volume visible par l'unité de capture a été modifiée, sur les parties arrière, l'enveloppe visuelle est conservée telle quelle. Ainsi, les différents résultats des unités multiscopiques représentent des zones d'influence sur le modèle 3D assez distinctes mais pouvant se chevaucher lorsque les unités sont situées l'une à coté de l'autre dans le système de capture. Dans ces zones de chevauchement, une des deux reconstructions hybrides proposées peut être plus pertinente que l'autre. Afin de quantifier cette pertinence, nous utilisons un des attributs de notre carte de matérialité calculée lors du processus de reconstruction multiscopique proposé dans le chapitre 3. Cet attribut est le score de confiance associé à chaque point cible. Afin d'obtenir à la fin du pipeline un modèle 3D unique de la meilleure qualité possible, les différences dans les zones de chevauchement des modèles partiels issus des unités multiscopiques sont identifiées et traitées. Pour éviter une complexité de résolution trop importante, nous avons opté pour un traitement incrémental des fusions volumiques. Le principe est d'initialiser la solution par l'enveloppe visuelle creusée par une première unité multiscopique, puis de fusionner itérativement la solution obtenue avec celle d'une autre unité multiscopique, avec un traitement approprié des zone où elles différent, . La dernière partie de ce chapitre, à travers quelques résultats, montre que l'approche proposée permet de récupérer, sous forme d'un volume creusé, une représentation 3D précise de la scène à modéliser. La qualité de ce volume permet de retrouver les petits détails et les

zones concaves sujettes à occultation. En conclusion, les contributions de ce chapitre sont triples : (i) une hybridation de notre méthode de stéréovision multi-vues grâce au guidage par l'enveloppe visuelle ; (ii) la sculpture de l'enveloppe visuelle à partir d'une surface de stéréovision interpolée et lissée; (iii) et enfin une fusion des volumes creusés d'une manière appropriée dans les zones où les informations diffèrent.

Conclusion générale

Notre travail s'inscrit dans le project RECOVER3D (Real-time Environment for COmputational Video Editing and Rendering in 3D). Le but de ce projet est de fournir un nouveau système virtuel de clonage d'acteurs, basé sur une capture multi-vidéo de leurs performances, et délivrant naturellement des modèles 4D texturés en haute résolution. Ce projet est basé principalement sur un partenariat entre des chercheurs académiques et des industriels. La principale caractéristique de ce système est de regrouper les caméras en deux types d'unité : les unités monoscopiques (une caméra) et les unités multiscopiques (quatre caméras). Ces unités sont placées autour de la scène sur deux niveaux afin de maximiser la zone de capture. Dans cette thèse, nous présentons la partie de ce projet dédiée à la reconstruction 3D de scènes dynamiques exploitant, pour chaque pas de temps, ce système d'acquisition illustrè dans le chapitre 1. La reconstruction 3D d'un objet à partir de silhouettes 2D est l'une des approches les plus répandues grâce à sa simplificité de mise en oeuvre. Les méthodes de reconstruction basée silhouettes sont généralement classées en deux groupes : i) approche volumétrique ; ii) approche polyédrique. L'un des principaux avantages de ces méthodes est leur capacité à reconstruire les zones sans texture, spéculaires et même transparentes. Toutefois, elles sont incapables de reconstruire les zones concaves, et le modèle 3D reconstruit est de faible précision comparé à ceux obtenus par d'autres approches comme la stéréovision. Beaucoup de travaux ont été proposés afin d'améliorer les reconstructions basées silhouettes en utilisant la stéréovision, car ces deux approches s'avèrent être complémentaires. En effet, l'approche stéréovision est capable de reconstruire les zones concaves et le modèle 3D résultant est plus précis. Notre travail s'inscrit dans cette dernière catégorie. Dans le chapitre 2, nous avons proposé de classer les méthodes existantes pour la fusion de ces deux approches en trois grands groupes clarifiant les avantages et les inconvénients de chacun d'eux : i) les méthodes de stéréovision guidée par l'enveloppe visuelle ; ii) les méthodes collaboratives appliquant simultanément des critères issus de ces deux techniques ; iii) les techniques fusionnant uniquement les résultats après une application séparée de ces deux méthodes. Nous avons également présenté dans chapitre 2 la

géométrie multi-épipolaire simplifiée et régulièrement espacée qui est celle utilisée pour exploiter les images acquises par les unités multiscopiques. Cette géométrie fournit une configuration efficace et robuste pour la modélisation d'objets 3D grâce à la réduction à une seule dimension de l'espace de recherche pour les pixels correspondants.

D'un autre côté, la reconstruction stéréovision multi-oculaire réalisée à partir de plus de deux points de vue est une généralisation naturelle de la reconstruction en stéréovision binoculaire. L'avantage d'utiliser un nombre d'images supérieur à deux est de pouvoir s'appuyer sur la redondance d'informations, laquelle aide à éviter les mauvaises mises en correspondance. Dans le chapitre 3, nous avons proposé une méthode de reconstruction multiscopique exploitant une capture multi-oculaire parallèle décentrée avec des centres optiques alignés et équidistants. Cette méthode propose une solution aux problèmes qui se posent couramment en stéréovision tels que les régions partiellement occultées. La méthode proposée est dite basée « scène », car elle s'appuie sur un nouvel échantillonnage de l'espace scénique adapté à la géométrie multi-épipolaire. Elle consiste à construire une carte discrète 3D de « matérialité » sur l'ensemble des points 3D que nous nommerons « points cibles » et définissons comme intersections des plans de disparités entières avec les rayons optiques des images. Une matérialité est codée entre 0 et 1 et exprime la vraisemblance de l'hypothèse que le point appartienne à la surface de la scène acquise. Cette approche définit aussi une fonction de visibilité sur l'ensemble fini des points cibles. En outre, notre proposition est bien adaptée pour le parallélisme. En effet, l'optimisation de la carte de matérialité est indépendante pour chaque plan épipolaire (voir le chapitre 3), cela nous permet ainsi de prévoir, dans une future implémentation, une mise en œuvre efficace sur GPU. Afin d'évaluer notre méthode, nous l'avons confrontée aux résultats issus d'approches existantes dans la domaine de stéréovision tel que TreeDP [77], MultiResGC [53], DoubleBP [81], GC+occ [38],et AdaptAggrDP [80] en utilisant deux mesures proposées par Scharstein et al. [60], la moyenne quadratique et le pourcentage de mauvaises mises en correspondance de pixels. Grâce à l'exploitation de la redondance des informations, de l'espace image et de notre nouvel espace géométrique de la scène, les résultats montrent que notre méthode est capable de traiter les régions occultées comme indiqué dans le tableau 3.7. Toutefois, ces résultats montrent aussi que notre méthode manque encore de robustesse dans les zones sans texture, comme nous l'expliquons dans la section 3.11. Par conséquent, dans le chapitre 4, nous avons proposé une hybridation par enveloppe visuelle de notre méthode de stéréovision multi-oculaire exploitant la géométrie multi-épipolaire simplifiée et régulière ainsi qu'une chaîne complète pour la reconstruction 3D adaptée au systéme de capture du projet. L'hybridation de la stéréovision et l'enveloppe visuelle tire parti de leur complémentarité afin de résoudre leurs problèmes individuels de reconstruction 3D. Cette hybridation consiste, pour notre méthode de stéréovision multi-

oculaire, à restreindre la zone de recherche des points cibles en utilisant l'enveloppe visuelle et ainsi à éliminer définitivement, en amont du processus de reconstruction, les points cibles candidats n'appartenant pas cette enveloppe. Dans notre chaîne de reconstruction, une fois l'enveloppe visuelle calculée, et puis sculptée par les résultats de stéréovision multi-vues pour chaque unité multiscopique, nous obtenons plusieurs volumes creusés pour la reconstruction d'un même objet 3D. Nous avons donc proposé de fusionner tous ces résultats en un modèle unique. Les grandes étapes de cette fusion sont : i) initialiser la solution avec la fusion de l'enveloppe visuelle et de la reconstruction issue d'une unité multiscopique ; ii) fusionner itérativement la solution courante avec la reconstruction obtenue pour une autre unité multiscopique. Ces fusions itératives nécessitent un traitement particulier sur les zones avec des informations contradictoires entre les reconstructions multiscopiques. Nous avons expérimenté notre chaîne sur des données réelles et sur des données virtuelles. Les résultats montrent que la fusion des deux techniques permet d'obtenir de meilleurs résultats que ceux obtenus séparément avec la stéréovision multi-oculaire ou l'enveloppe visuelle. Après avoir modélisé l'objet en 3D en chaque trame de la vidéo grâce à notre proposition, le projet RECOVER3D inclut un suivi temporel de modèle 3D qui évalue le champ de mouvements inter-trame par appariement de voxels. Ce champ est ensuite appliqué par déformation pseudo-rigide au maillage du modèle. Ce travail a été réalisé au sein d'une autre thèse [5] proposée par le laboratoire CReSTIC. Un perspective de notre travail pourrait être d'utiliser cette information de mouvement (e.g. les champs de vecteurs) afin d'affiner le modèle 3D reconstruit par notre chaîne. Dans de futurs travaux, nous proposons d'élargir le projet RECOVER3D pour être en mesure d'appliquer la reconstruction de la scène 3D dynamique dans des environnements extérieurs. Cela revient á abandonner la technologie du studio pour la remplacer par des méthodes de détourage adaptées aux environnements non contrôlés. Cette proposition nécessite un système synchronisé de plusieurs caméras portables et une méthode d'extraction du premier-plan en tenant compte des changements de dynamiques dans les vues au fil du temps. Un autre aspect important de la reconstruction au sein du projet RECOVER3D est la colorisation de l'objet 3D obtenu. En général, dans le domaine de la reconstruction de scène 3D, il existe deux grandes classes de colorisation de la géométrie. La première suppose que la résolution de la discrétisation de la scène en voxels est assez fine afin de pouvoir fournir une seule couleur pour chaque voxel, dérivée de sa projection sur les images. La seconde est le « texture-mapping » qui revient à projeter chaque vue disponible sur l'objet reconstruit. Dans cette dernière classe, un certain nombre d'approches ont été développées, comme la méthode proposée par Debevec et al.[18] qui applique, après la projection des images, sur chaque primitive de l'objet géométrique 3D (par exemple sur chaque sommet), un processus d'interpolation et de mélange utilisant les vues

les plus pertinentes et l'orientation locale de la surface. Par simplicité, nous avons choisi le « voxel coloring », une méthode de la première classe. Une des évolutions possibles serait de modifier ce choix et de mettre en œuvre une approche de la seconde classe, comme [18], afin d'obtenir des modèles texturés visuellement plus convaincants.

Chapter 5

Conclusions and perspectives

The 3D reconstruction of an object from multiple 2D silhouettes corresponding to different viewpoints has long been considered as to be a preferred approach. We distinguish two major approaches of silhouette-based 3D reconstruction : i) volumetric, ii) polyhedral. A major advantage is such approaches permit the reconstruction of textureless, specular, or even transparent objects. However, they fail to reconstruct the concave zones, and they lack precision in 3D object modeling compared with stereovision approaches. Recently, several approaches were proposed to improve the silhouette-based 3D reconstruction with stereovision. The stereovision and silhouette-based 3D reconstruction approaches complement approaches complement one another since stereovision is able to reconstruct the concave regions and produce highly detailed 3D reconstructions. In this thesis, we presented a part of the RECOVER3D project about the 3D reconstruction of an actor in multi-view studio, coupling video cameras laid in both monoscopic and multiscopic units, We propose a 3D reconstruction solution using both multi-baseline stereovision and silhouette-based 3D reconstruction.

In chapter 2, we proposed to classify existing methods that merge stereovision and silhouette-based 3D reconstruction into three major groups clarifying the advantages and disadvantages of each these methods i) Stereovision guided by visual hull methods, ii) Collaborative methods applying simultaneously criteria borrowed from both techniques, iii) Separate application of both methods with further merging of their results. We also presented in the chapter 2, the multi-simplified epipolar geometry which provides an efficient and robust configuration for 3D object modeling thanks to reduction to one dimension of the search space for matching pixels.

In chapter 3, we proposed a novel framework to deal with commonly occurring problems in multi-view stereovision such as semi or totally occluded regions. Furthermore, our framework uses multiple images shot or rectified in multi-simplified epipolar geometry (see section 2.1.3.2). Our approach relies on a new scene space sampling scheme fitted to this simplified geometry. Rather than dealing with the full 3D scene, our method estimates the effective 3D scene where the objects need to be reconstructed. However, the multi-view stereovision relies only on information in the image space and sometimes has difficulties to recover precise geometry, particularly in low texture regions. For this reason, we proposed to optimize scene geometry with respect to image information in order to obtain a high-accuracy 3D model of objects handling the semi and totally occluded regions. In addition to our scene geometry definition, the novelty of our approach lies in building 3D discrete materiality map with values ranging between 0 and 1. These values express the affiliation of target points in the useful scene to object surfaces. Compared with the results derived from other methods (TreeDP[77], MultiResGC[53], DoubleBP[81], GC+occ [38], AdaptAggrDP[80]), our results show that our method is able to deal with occluded regions thanks to exploitation of redundancy information and to rely on the image space and geometry space information, as shown in table 3.7. However, we showed that results of multi-baseline stereovision still lack robustness in low textured areas.

In chapter 4, we demonstrated the benefits of enhancing a multi-baseline approach with visual hull guidance. Applying our multi-baseline approach on each multiscopic unit, we obtain several carved volumes for a same 3D object. We proposed a novel framework to merge these volumes. The overall principle to get full 3D modeling is to initialize the computation by a reconstructed volume from a first multiscopic camera, then merge iteratively the current solution with those of following multiscopic unit taking into account the (in)consistency zones.

We applied our framework on a virtual scene composed of a virtual actor. The virtual scene permits to validate our method in some perfectly known setting (i.e. without any calibration error). This has significant impact on the final results. Afterwards, the framework proposed in this thesis was implemented and experimented with RECOVER3D real actors.

5.1 Perspectives

We identified several aspects of our work that could be improved in the near future as well as in a long term perspective. In the following, we discuss those different suggestions of future improvements:

- Using motion information: within the RECOVER3D project, after the reconstruction step described in this thesis, we obtain a sequence of discrete volumes that represent

the character's pose at each video frame. In traditional multi-view reconstruction pipelines, these volumes are transformed into a sequence of 3D textured meshes that are successively loaded to memory for the rendering of each frame. Another goal of the RECOVER3D project addressed within the CReSTIC laboratory [5] is to introduce a dynamic representation of the character to free ourselves from this static description of the scene in order to produce a single, temporally consistent, animated model according to the character's motion. Integration has not yet been performed and could be a natural immediate step. As a more long term goal, we could envisage using the motion information in order to refine the 3D shape reconstructed by silhouettes and stereovision methods. Silhouette, texture and motion information thus could be integrated to accurately fit the 3D mesh to the object surface.

- Texturing 3D object: another important aspect of 3D reconstruction within the RECOVER3D project is the coloring of the obtained 3D object. In general, there are two major classes for coloring 3D geometry for 3D reconstructed scene purposes. The first assumes the voxel's object to be fine enough in order to provide a single color for each voxel derived from their projections onto the images. The second is the texture mapping which expresses by the projection of each image onto the reconstructed object. A number of approaches were developed like Debevec et al. [18] who apply, after the image projection, an interpolation process on each primitive of the 3D geometric object using a subset of nearest views according to the orientation of the primitive surface. At the moment, in the RECOVER3D project, the first class of methods was chosen, for simplicity reasons, for surface coloring. In the future, the second class should be considered instead in order to obtain texture-mapped models more visually convincing.
- Handling scenes with multiple objects: Until now, our application is evaluated using the scene containing one object. Manipulating multiple objects yields to ambiguity in the 3D reconstruction scene using only VH. Therefore, we expect that the fusion between VH and stereovision will solve most of collisions problem between 3D reconstructed objects and refine the results of VH.
- Allowing outdoor capture: The RECOVER3D system is composed of cameras that are fixed and calibrated and has a chromakey background. The current assumptions would not allow to enable dynamic 3D scene reconstruction in outdoor environments. A perspective project would be to extend the developed approaches to uncontrolled environments. This proposition requires a synchronized portable multiple camera system and a specific method for foreground extraction taking into account the dynamic changes in appearance between views and over time.

- Addressing realtime and high resolution data: In order to market the RECOVER3D system and software solution as a product for TV and film producers, the XD production company would require an increase of resolution (4K) and realtime computations. In our application, we work with an image resolution of 1920×1080 which has a direct impact on the quality level of multi-baseline stereovision. One can easily figure that higher image resolutions would provide more detailed results since the number of the available target points to be reconstructed into the geometry scene is increased. The approach proposed in this thesis could be implemented on the GPU in order to address current market targeted images with high resolution images, like 4K resolution of 3840×2160. Since the materiality map optimization is independent (see chapter 3) for each epipolar plan, our proposition is well suited for parallelism, such that it can implemented efficiently on GPU. However, the Gradient Descent algorithm is an iterative process requiring many iterations in order to converge and find the minimum energy. This implies difficulties of our method to reach the real time. Therefore, an alternative to our materiality map using another optimization method should be envisaged in order to improve the computing time.

References

[1] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Proceedings of the Symposium on Close-Range photogrammetry*, 1:18, 1971.

[2] L. An, Y. Jia, J. Wang, X. Zhang, and M. Li. An efficient rectification method for trinocular stereovision. *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*, 4:56–59, 2004.

[3] N. Ayache and C. Hansen. Rectification of images for binocular and trinocular stereovision. *9th International Conference on Pattern Recognition*, 1:11–16, 1988.

[4] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 2:631–636, 1981.

[5] L. Blache, C. Loscos, and L. Lucas. Robust motion flow for mesh tracking of freely moving actors. *The Visual Computer journal*, 32(2):205–216, dec 2015.

[6] A. F. Bobick and S. S. Intille. Large occlusion stereo. *Int. J. Comput. Vision*, 33: 181–200, 1999.

[7] E. Boyer. INRIA Rhône Alpes Grenoble 4D modeling platform, 2015. URL http://kinovis.inrialpes.fr/.

[8] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[9] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[10] G. Bradski. Camera calibration and 3D reconstruction. *Dr. Dobb's Journal of Software Tools*, 2000.

[11] F. Caillette. *Real-time markerless 3D human body tracking*. PhD thesis, University of Manchester, 2006.

[12] G.K.M. Cheung, T. Kanade, J.Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:714–720, 2000.

[13] C.H. Chien and J.K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *CVGIP*, 36:100–113, 1986.

[14] R.T. Collins. A space-sweep approach to true multi-image matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 358, 1996.

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

[16] G. Cross and A. Zisserman. Surface reconstruction from multiple views using apparent contours and surface texture. *Conference of Computer Vision and Computer Graphics*, 84:25–47, 2000.

[17] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[18] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 11–20, 1996.

[19] S. Fan and F.P. Ferrie. Photo hull regularized stereo. *Canadian Conference on Computer and Robot Vision*, page 18, 2006.

[20] O. D. Faugeras. Three-dimensional computer vision: A geometric viewpoint. pages 475–475, 1993.

[21] O. D. Faugeras, Q. T Luong, and T. Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, Cambridge, MA, USA, 2001.

[22] D. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, USA, 2010.

[23] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

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

[25] P. Gargallo and P. Sturm. Bayesian 3D modeling from images using multiple depth maps. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:885–891, 2005.

[26] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. *Proceedings of IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[27] C. Hane, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys. Stereo depth map fusion for robot navigation. *IROS*, pages 1618–1625, 2011.

[28] C.E. Hernàndez and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding journal*, 96(3):367–392, 2004.

[29] A. Hilton and J. Starck. Multiple view reconstruction of people. *2nd International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT*, pages 357–364, 2004.

[30] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. *16th IEEE International Conference on Image Processing, ICIP*, pages 2093–2096, 2009.

[31] M. H. Ju and H. B. Kang. Constant time stereo matching. *International Machine vision and Image Processing conference, 2009. IMVIP '09. 13th International*, pages 13–17, 2009.

[32] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1: I–103–I–110, 2001.

[33] S.B. Kang and R. Szeliski. Extracting view-dependent depth maps from a collection of images. *Int. J. Comput. Vision*, 58:139–163, 2004.

[34] Y. Kang, C. Lee, and Y. Ho. An efficient rectification algorithm for multi-view images in parallel camera array. *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 61–64, 2008.

[35] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vision*, 1(4):321–331, 1988.

[36] Z. Ke, G. Lafruit, R. Lauwereins, and G. L. Van Gool. Joint integral histograms and its application in stereo matching. *17th IEEE International Conference on Image Processing, ICIP*, pages 817–820, 2010.

[37] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *Proceedings of the 18th International Conference on Pattern Recognition*, 03:15–18, 2006.

[38] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. *Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001*, 2:508–515, 2001.

[39] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 82–96, 2002.

[40] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, 2000.

[41] S. Lazebnik, Y. Furukawa, and J. Ponce. Projective visual hulls. *Int. J. Comput. Vision*, 74(2):137–165, 2007.

[42] M. Li, H. Schirmacher, M. Magnor, and H. P. Siedel. Combining stereo and visual hull information for on-line reconstruction and rendering of dynamic scenes. *Workshop on IEEE Multimedia Signal Processing*, pages 9–12, 2002.

[43] L. Lucas, P. Souchet, M. Ismael, O. Nocent, C. Niquin, C. Loscos, L. Blache, S. Prévost, and Y. Rémion. RECOVER3D: A hybrid multi-view system for 4d reconstruction of moving actors. *4th International Conference and Exhibition on 3D Body Scanning Technologies (3DBST)*, pages 219–230, November 2013.

[44] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu. Constant time weighted median filtering for stereo matching and beyond. *Proceedings of IEEE International Conference on Computer Vision*, pages 49–56, 2013.

[45] K. Matsuda and N. Ukita. Direct shape carving: Smooth 3D points and normals for surface reconstruction. *IEICE Trans. on Information and Systems*, pages 1811–1818, 2011.

[46] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

[47] K.S. Narayan, J. Sha, A. Singh, and P. Abbeel. Range sensor and silhouette fusion for high-quality 3D scanning. *IEEE International Conference on Robotics and Automation, ICRA*, pages 3617–3624, 2015.

[48] P.J. Narayanan, P.W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. *Proceedings of the Sixth International Conference on Computer Vision*, page 3, 1998.

[49] R. A. Newcombe., S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *Proceedings of 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[50] C. Niquin. *Reconstruction du relief et mixage réel virtuel par caméras relief multi-points de vues*. Phd thesis, Université de Reims Champagne-Ardenne, Mars 2011.

[51] C. Niquin, S. Prévost, and Y. Rémion. An occlusion approach with consistency constraint for multiscopic depth extraction. *Int. J. Digital Multimedia Broadcasting*, 2010, 2010.

[52] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(4):353–363, April 1993.

[53] N. Papadakis and V. Caselles. Multi-label depth estimation for graph cuts stereo problems. *Journal of Mathematical Imaging and Vision*, 38(1):70–82, 2010.

[54] S. Paris, P. Kornprobst, and J. Tumblin. *Bilateral Filtering*. Now Publishers Inc., Hanover, MA, USA, 2009.

[55] J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, 1982.

[56] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[57] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3017–3024, 2011.

[58] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. *Proceedings of the European Conference on Computer Vision, ECCV*, 6313, 2010.

[59] S. Roy. Stereo without epipolar lines: A maximum-flow formulation. *Int. J. Comput. Vision*, 34(2-3):147–161, 1999.

[60] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

[61] D. Scharstein, R. Szeliski, and H. Hirschmüller. Middlebury stereo vision site, 2002. URL http://vision.middlebury.edu/stereo/.

[62] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. Comput. Vision*, 35(2):151–173, 1999.

[63] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:519–528, 2006.

[64] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:345–352, 2000.

[65] P. Song, X. Wu, and M. Wang. Volumetric stereo and silhouette fusion for image-based modeling. *The Visual Computer journal*, 26(12):1435–1450, 2010.

[66] J. Starck, S. Nobuhara, A. Maki, A. Hilton, and T. Matsuyama. The multiple camera 3D production studio. *IEEE Trans. Circuits and Systems for Video Technology*, 19: 856–869, 2009.

[67] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 1999.

[68] E. Steinbach, B. Girod, P. Eisert, and A. Betz. 3D reconstruction of real-world objects using extended voxels. *International Conference on Image Processing Proceeding*, 1: 569–572, 2000.

[69] C. Sun. Uncalibrated three-view image rectification. *International Conference Image and Vision Computing*, 21(3):259–269, 2003.

[70] J. Sun, N. Zheng, and H. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, 2003.

[71] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Underst*, 58(1):23–32, 1993.

[72] R. Szeliski. A multi-view approach to motion and stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:163, 1999.

[73] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[74] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *Int. J. Comput. Vision*, 32(1):45–61, 1999.

[75] C.J. Taylor. Surface reconstruction from feature based stereo. *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2:184, 2003.

[76] O. Veksler. Stereo correspondence by dynamic programming on a tree. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:384–390, 2005.

[77] O. Veksler. Stereo correspondence by dynamic programming on a tree. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:384–390, 2005.

[78] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2: 391–398, 2005.

[79] C. Wang, N. Komodakis, and N. Paragios. Markov random field modeling, inference and learning in computer vision and image understanding: A survey. *Computer Vision and Image Understanding journal*, 117(11):1610–1627, 2013.

[80] L. Wang, R. Yang, M. Gong, and M. Liao. Real-time stereo using approximated joint bilateral filtering and dynamic programming. *J. Real-Time Image Process*, 9(3):447–461, 2014.

[81] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504, 2009.

[82] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. *Neural Information Processing Systems*, pages 689–695, 2000.

[83] K. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. PAMI*, 28:650–656, 2006.

[84] K. J. Yoon and I. S. Kweon. Locally adaptive support-weight approach for visual correspondence search. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:924–931, 2005.

[85] Z. Zhang. Chapter 2, camera calibration. *In emerging topics in computer vision*, pages 4–43, 2004.

[86] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*, 2: 28–31, 2004.

RECONSTRUCTION DE SCENE DYNAMIQUE A PARTIR DE PLUSIEURS VIDEOS MONO ET MULTISCOPIQUES PAR HYBRIDATION DE METHODES « SILHOUETTES » ET « MULTI-STEREOVISION »

La reconstruction précise d'une scène 3D à partir de plusieurs caméras offre un contenu synthétique 3D à destination de nombreuses applications telles que le divertissement, la télévision et la production cinématographique.
Cette thèse propose une nouvelle approche pour la reconstruction 3D multi-vues basée sur l'enveloppe visuelle et la stéréovision multi-oculaire. Cette approche nécessite en entré l'enveloppe visuelle et plusieurs jeux d'images rectifiées issues de différents *unités multiscopiques* constituées chacune de plusieurs caméras alignées et équidistantes. Nos contributions se situent à différents niveaux. Le premier est notre méthode de stéréovision multi-oculaire qui est fondé sur un nouvel échantillonnage de l'espace scénique et fournit une *carte de matérialité* exprimant la probabilité pour chaque point d'échantillonnage 3D d'appartenir à la surface visible par l'unité multiscopique. Le second est l'hybridation de cette méthode avec les informations issues de l'enveloppe visuelle et le troisième est la chaîne de reconstruction basée sur la fusion des différentes enveloppes creusées tout en gérant les informations contradictoires qui peuvent exister. Les résultats confirment :
I) l'efficacité de l'utilisation de la carte de matérialité pour traiter les problèmes qui se produisent souvent dans la stéréovision, en particulier pour les régions partiellement occultées ; II) l'avantage de la fusion des méthodes de l'enveloppe visuelle et de la stéréovision multi-oculaire pour générer un modèle 3D précis de la scène

Reconstruction 3D à partir de multiples vues, Stéréovision multi-vue, Enveloppe visuelle, Géométrie épipolaire parallèle décentrée, Reconstruction basée silhouette.

3D SCENE RECONSTRUCTION BY SILHOUETTE AND MULTI-BASELINE STEREOVISION

Accurate reconstruction of a 3D scene from multiple cameras offers 3D synthetic content to be used in many applications such as entertainment, TV, and cinema production. This thesis is placed in the context of the RECOVER3D collaborative project, which aims is to provide efficient and quality innovative solutions to 3D acquisition of actors. The RECOVER3D acquisition system is composed of several tens of synchronized cameras scattered around the observed scene within a chromakey studio in order to build the visual hull, with several groups laid as *multiscopic units* dedicated to multi-baseline stereovision. A multiscopic unit is defined as a set of aligned and evenly distributed cameras. This thesis proposes a novel framework for multi-view 3D reconstruction relying on both multi-baseline stereovision and visual hull. This method's inputs are a visual hull and several sets of multi-baseline views. For each such view set, a multi-baseline stereovision method yields a surface which is used to carve the visual hull. Carved visual hulls from different view sets are then fused iteratively to deliver the intended 3D model. Furthermore, we propose a framework for multi-baseline stereo-vision which provides upon the Disparity Space (DS), a *materiality map* expressing the probability for 3D sample points to lie on a visible surface. The results confirm i) the efficient of using the materiality map to deal with commonly occurring problems in multi- baseline stereovision in particular for semi or partially occluded regions, ii) the benefit of merging visual hull and multi-baseline stereovision methods to produce 3D objects models with high precision.

Multiview 3D reconstruction, Multi-baseline stereovision, Visual hull, Decentered parallel geometry, Shape from silhouette.

Discipline : INFORMATIQUE

UFR Sciences Exactes et Naturelles

EA 3804 CRESTIC

Moulin de la Housse – 51687 REIMS CEDEX 2