REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider - BISKRA
Faculté des Sciences Exactes et Sciences de la Nature et de la Vie
Département d'informatique
Laboratoire LESIA

N° d'ordre :.....
Série : .....

Thése

en vue d'obtenir le titre de

Docteur en Informatique 3 éme cycle LMD

Option : **Sciences et techniques de l'image**

Effective Multi-view Stereo 3-Dimensional Reconstruction for Virtual Reality

---

Présenté par
Abdelhak SAOULI
Soutenue le 16/09/2019

Devant le jury composé de :

Président	Noureddine Djedi	Professeur	Université de Biskra
Rapporteur	Mohamed Chaouki Babahenini	Professeur	Université de Biskra
Examinateur	Kadi Bouatouch	Professeur	Université de Rennes 1
Examinateur	Nadia BAHA	Professeur	Université ST Houari Boumediene
Examinateur	Abdelhamid DJEFFAL	MCA	Université de Biskra

2018 — 2019

Abstract

Given a set of photographs of real objects or a scene, estimate the closest Three-dimensional geometry that specifically explains those photographs. In Computer Vision literature, this problem is known as Image-based Modelling or Multi-view Stereo (MVS) reconstruction. It is considered a hot research topic due to the huge technological advances of digital cameras, where these last became a cheap and reliable high resolution sensors. In fact, its application range from 3D mapping and navigations in robotics to video games and film making industry. Only recently however has this technique matured enough to be used in a natural uncontrolled environment. Meanwhile, Virtual Reality (VR) is witnessing a huge revolution due to the advances in display, sensing, and computing technology. In fact, the VR head mounted displays are mass produced, and a wide variety of people have access to this technology, consequently, experiences like virtual society, virtual travelling and tele-presence flourish.

This class of applications however depends significantly on the visual fidelity of its contents. For instance, some applications capture a panoramic view of the remote environment, others build the virtual world inspired by real-life locations using classical modelling techniques, hence any false representation can render the experience inadequate due to the absence of immersion. Photogrammetry (also known as Multi-view Stereo) on the other hand seems to be a natural answer to this problem. Nevertheless, achieving a high degree of visual fidelity becomes a challenge because these algorithms suffer from multiple major failure modes.

This dissertation addresses the two major problem in multi-view stereo reconstruction related to virtual reality applications. First, we are focused on the interactivity. Such aspect puts the real-time as a high priority constraint. Thus, the reconstruction methods must be able to estimate the 3D shape of a static or dynamic object accurately in a matter of milliseconds. In fact, research proved that it is possible to use multiple cameras attached to cluster of networked computer, and model the 3D geometry of any rigid or static body in real-time. However, such setup is cost-effective. Hence, we study the possibility of building a simpler system that run on a single machine, and we present a GPU accelerated image-based modelling system, the algorithm estimate and render on the fly all visible parts of a visual hull from a novel viewpoint

without noticeable artifact. We carefully adapted our algorithm implementation to the recent off-the-shelf hardware.

In the second part, we investigate the accuracy of the offline reconstructed objects. We aim for highly immersive virtual reality experiences. Therefore, it is a necessity for MVS to preform on large cluster of images such as community photo collection. These datasets not only they are large but also contain numerous settings in which photogrammetry fails. We propose a robust shading aware multi-view stereo method based on meta-heuristic optimization namely the Particle Swarm Optimization (PSO) to faithfully reconstruct textureless areas without any explicit regularization. Furthermore, to handle the various shading and stereo mismatch problems caused by a non-Lambertian surfaces, we present our robust matching/energy function which is a combination of two similarity measurements. Finally, qualitative and quantitative experiments are performed for multiple benchmarks, proving the effectiveness of our approach.

Keywords: Multi-View Stereo (MVS), Depth Maps, Swarm Optimization, Virtual Reality, GPU.

المـلخص

نفرض أننـا قمنـا بإلتقاط عدة صورة لجسم معين . تكهـــن أو إستنتاج الشكل الحقيقي ذي الابعاد الثلاثية التي تطـــابق تلك الصور الملتقطة للجسم هو محور بحث معروف في مجال رؤية الحاسب, هذا المحور يسمى بالتجسيم من خلال الصور. يعتبر مجال البحث هذا هام جدا خلال السنوات الأخيرة وهذا راجع إلى التقدم التكنولوجي الكبير الذي شهدته صناعة آلات الكميرات الرقمية الجديدة والتي أصبحت تعتبر آلات إستشعار عالية الجودة أضف إلى ذالك الأسعار الزهيدة التي تتيح لأي شخص إقتنائها. في الحقيقة ، إن التطبيقات التي يمكن أن تستفيد من مجال النجسيم من خلال الصور عديدة ومتنوعة جدا نذكر على سبيل المثال إنشاء أالخرائط ثلاثية الأبعاد التي يمكن أن تستخدم في الهندسة المعمارية و الروبوتيكس

في الجانب الأخر من عالم التكنولوجيا, نشهد في هذه السنوات الأخيرة تقدما ملحوض في ما يعرف بإسم العالم الإفتراضي ، و هذا أيضا نتيجة التقدم التكنولوجي في تقنيات العرض ، والاستشعار ، وتكنولوجيا الحوسبة . في الواقع ، يتم إنتاج شاشات العرض التي يتم تركيبها على رأس بكميات كبيرة ، ممـا أدى إلى أن مجموعة واسعة من الأشخاص تلجأ لإستخدام هذه التقنية . بنـاءا على ذلك فإن تجارب مثل المجتمع الافتراضي والسفر الافتراضي والتواجد عن بعد تزدهر ونتطور نحو الأفضل. ولكن هذه الفئة من التطبيقات تعتمد بشكل كبير على الدقة البصرية وجودة الصورة التي تمثل محتوياتها. على سبيل المثال ، تلتقط بعض التطبيقات رؤية بانورامية للمشهد، في حين يبني آخرون العالم الافتراضي المستوحى من مواقع الحياة الحقيقية باستخدام تقنيات النمذجة الكلاسيكية ، وبالتالي فإن أي تمثيل زائف أو خطأ في التركيب يمكن أن يجعل التجربة غير فعالة بسبب غياب عنصر الإنسجام التام. من ناحية أخرى يبدو أن الجواب الطبيعي لهذه المشكلة يكمن في إستعمال تقنية التجسيم من خلال الصور. ومع ذلك ، فإن تحقيق درجة عالية من الإخلاص المرئي قد يصبح تحديًا لأن هذه التقنية تعاني من العديد من أنماط الفشل.

هذه الرسالة تتناول مشكلتين رئيسيتين في مجال التجسيم من خلال الصور، المشكلتان بطبيعة الحال متعلقتان بمجال العالم الإفتراضي. أولا ندرس جانب التفاعل الآني مع العالم الإفتراضي، إن هذا الجانب في الواقع يضع شرط مهما يجب أن يأخذ بعين الإعتبار أثنـاء التصميم، هذا الشرط هو الوقت. حيث انه يتوجب على تقنية التجسيم أن تركب الشكل الحقيقي ذي الابعاد الثلاثية لجسم متحرك أو ثابت في وقت قصير جدا يقدر بأجزاء من الثانية. إن أطروحتنا تدر س إمكانية بناء نظام بسيط يعمل على جهاز حاسب واحد مستخدما وحدة معالجة الرسومات .*GPU* تقوم الخوارزميةا المطروحة بتقدير وعرض جميع الأجزاء المرئية لما يعرض بالفيزيال هوول وذلك دون اي تشوهات رسمية.

في الجزء الثاني ، نتحقق من دقة الأجسام التي أُعيد بناؤها. إن الهدف الأساسي هو ضمان جودة عالية لتجارب الواقع الافتراضي . ولذلك ، فمن الضروري لتقنية التجسيم من خلال الصور أن تستعمل أنواع مخصصة من الصور بأعداد كبيرة. مثل هذا الوضع يسعب المهمة على الخورزميات وقد يتولد عنه فشل في `ادة تركيب الاجسام الحقيقة و إدماجها داخل الواقع الإفتراضي. من أجل خل هذه المشاكل فإننا نقترح طريقة جديد للتجسيم من خلال الصور ، حيث أن كريقتنا تأخذ بعين الإعتبار الأجسام المضللة و الأجسام تحتوي

على تويع في الألوان. تعتمد طرقيتنا على تقنية التحسين عن طريق السرب الطبيعي أخيرا ، تم إجراء تجارب نوعية وكميةبإستعمال عدة معايير ، النتائج المتحصل عليها تثبت فعالية نهجنا.

الكلمات المفتاحية: الواقع الإفتراضي، التجسيم من خلال الصور، وحدة المعالجة المرأية ، خراطة الأبعاد

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Problem Statement . . . 4
- 1.3 Contributions and Thesis Outline . . . 6
  - 1.3.1 Real-time and Interactivity . . . 7
  - 1.3.2 Multi-view Stereo Under General conditions . . . 7
  - 1.3.3 Thesis Outline . . . 8

2 The Digital World in Virtual Reality **9**
- 2.1 Defining The Reality of Virtual Reality . . . 10
  - 2.1.1 A Psychophysics Analysis . . . 12
  - 2.1.2 A Technological Analysis . . . 15
- 2.2 Modeling The Virtual World . . . 17
  - 2.2.1 Classical Polygonal Modeling . . . 18
  - 2.2.2 Procedural Generated Modeling . . . 19
  - 2.2.3 Image-Based Modeling . . . 20
- 2.3 Rendering The Virtual World . . . 21
  - 2.3.1 The Optical characteristic of surfaces . . . 22
  - 2.3.2 Rendering Algorithms . . . 25
- 2.4 Conclusion . . . 25

3 Multi-view Stereo Reconstruction **26**
- 3.1 Key concepts of the image-based reconstruction . . . 28
- 3.2 Multi-view Environment . . . 30
  - 3.2.1 Imagery Collection . . . 30
  - 3.2.2 Camera Models and Calibration . . . 33
  - 3.2.3 Multi-View Stereo . . . 36
- 3.3 Photometric Consistency in Multi-view Environment . . . 37
  - 3.3.1 Photo-consistency . . . 37
  - 3.3.2 Evaluating Visibility in literature . . . 38
  - 3.3.3 Consistency measurement tools . . . 40
- 3.4 State-of-art Algorithms . . . 42
  - 3.4.1 Visual Hull Reconstruction . . . 42
  - 3.4.2 Depth Maps Reconstruction . . . 43
- 3.5 Conclusion . . . 46

4 Interactive Multi-view Stereo Rendering **47**
- 4.1 The Proposed Visual Hull Reconstruction System . . . 48
- 4.2 Parallel Strategy For Image-based Visual Hull . . . 49

4.2.1 Image processing . . . 51
4.2.2 Visual Hull estimation . . . 52
4.2.3 Visual Hull rendering . . . 53
4.3 Experiment . . . 54
4.3.1 System setup . . . 54
4.3.2 Datasets . . . 57
4.3.3 GPU Implementation details . . . 58
4.3.4 Results . . . 63
4.4 Critical analysis . . . 70
4.4.1 Complexity . . . 72
4.4.2 Multi-GPU multi thread architecture . . . 73
4.5 Conclusion . . . 74

5 Accurate and Realistic World Reconstruction **76**
5.1 Problem Description . . . 77
5.1.1 Homogeneous Surfaces . . . 77
5.1.2 Non-Lambertian and Thin-details . . . 79
5.1.3 Motivations and Proposition . . . 80
5.2 Framework Description . . . 83
5.3 Key Elements of the Proposed Method . . . 86
5.3.1 Geometric Patch Model . . . 86
5.3.2 Photometric Model . . . 88
5.3.3 Similarity Measurement . . . 94
5.3.4 Estimating Light Direction . . . 97
5.4 Algorithm . . . 99
5.4.1 Global View Selection . . . 100
5.4.2 Particle Swarm Initialization . . . 102
5.4.3 Matching Process and Expansion . . . 105
5.4.4 Convergence Criteria . . . 107
5.5 Evaluation and Experimental Results . . . 108
5.5.1 Experimental Results . . . 109
5.5.2 Discussion . . . 117
5.6 Conclusion . . . 121

6 Conclusion **122**
6.1 Summary . . . 122
6.2 Perspectives . . . 124

References **135**

List of Figures

1.1 The main protagonist *Wade* from *Ready Player One* film. Artistic poster depicts the blurring of fantasy and reality. . . . 2
1.2 The Open-World concept in *The Witcher 3* (2015). A massive map (left image) of detailed virtual world (right image) . . . 3

2.1 Illustration of Virtual Reality System overview (a) and detailed presentation of that system (b). . . . 12
2.2 The dress has confused the world in 2014 where people couldn't agree on its color. . . . 13
2.3 Graphical illustration of the monocular and some binocular cues for depth perception (a) sight cue, (b) perspective cue, (c) stereo cue. . . . 14
2.4 Virtual objects modeled with great details to match their real world counter part using Blender. As it can be seen the 3D modeling is considred a new medium of art in our modern age . . . 18
2.5 Subdivision process, the left image (a) shows the original polygonal mesh, i.e, control surfaces, The following image (b) is subdivided three times. As can be seen, more and more vertices are generated and the surface is getting smoother . . . 19
2.6 The power of procedural generating algorithms in content creation for virtual world. This images from E3 2015 press conference presentation of *No man's sky* where (a) represent the whole universe and (b) represent a planet inside that universe and finally (c) is the world inside that planet all of it was procedurally generated (Image curtesy to Push Square youtube channel) . . . 20
2.7 "The Vanishing of Ethan Carter", one of the best looking game since 2015. This is largely due to its use of multi-view stereo reconstruction coupled with sophisticated post-processing (the picture is a screen from a gameplay) . . . 21
2.8 Real objects can have a complex reflection that can be modeled by a BRDF. Left image represent simple diffuse reflection surface modelled via a Lambertian law. On the right the microfacet BRDF model of reflection . . . 23

3.1 Photogrammetry. Given a set of photographs (a), the objective of Photogrammetry reconstruction algorithms is to infer the most likely $3D$ geometry that explains those photographs (b). . . . 27
3.2 Different MVS imagery types. From top to bottom: a controlled MVS capture in Laboratory setup, outdoor imagery of small environmental scenes, and finally a crowd-sourcing imagery from online photo-sharing websites. . . . 31
3.3 A point $\mathbf{x} = (X, Y, Z)$ is projected onto the image plane by the ray passing through the center of projection, and the resulting point on the image is $\mathbf{u} = (i, j, f)$; . . . 33
3.4 Main phases of a general SFM pipeline, following the arrows, the process start by feature detection and feature matching than track generation which lead to structure from-motion finally bundle adjustment and the result is point cloud and camera parameters. . . . 35

3.5 Multi-view stereo matching problem in nutshell. (a): The $3D$ shape of the scene gives the correspondence between pixels in different photographs. (b): If camera parameters are known, matching a pixel in one image with pixels in another image is converted to searching problem on $2D$ line. 36
3.6 Visibility problem, in order to estimate geometry using photo-consistency occlusion should not appears, in the same time occlusion is detected only using geometry 39

4.1 A single slice of an image-based visual hull which is a 2D representation of the full image-based visual hull. 48
4.2 System flowchart defining inputs, outputs and processes in reconstructing the visual hull using the modified images-based algorithm. Regions labelled 1–3 are data independent parallel regions 50
4.3 Rendering results of our GPU image-based visual hull reconstruction using the Dancer Dataset. One the left the original view while on the right The novel view of the visual hull is rendered by blending the textures from multiple viewpoints. The background scene is rendered as a textured sky box. 52
4.4 a 2D illustration for The process of reconstructing the visual hull using a ray casting approach. 53
4.5 High-Level Overview of Nvidia Geoforce graphic chip. 55
4.6 Image samples from the Datasets used in our experiments:(a) Human skull dataset, (b) Red Dinosaur Toy dataset, (c) Children Playing dataset, (d) Dancer dataset. 56
4.7 Pageable Data Transfer vs Pinned Data Transfer 59
4.8 A $2D$ illustration for (a) cuda $3D$ texture and (b) interpolation 61
4.9 Results on the Middlebury dataset. The images show examples from our GPU images-based visual hulls, many details were preserved with no voxalization artifacts. 64
4.10 Qualitative results of our GPU image-based visual hull reconstruction on *the skull Homo-Heidelbergensis*. The first row represent two images from the data set. While the second row shows the reconstructed views at output resolution equal to 800 x 600 66
4.11 Qualitative results of our GPU image-based visual hull reconstruction on *Dino Toy*. (a) Image of the object from the dataset. (b) A reconstructed visual hull using our approach with interpolated Texels. (c) A reconstructed visual hull using our approach with interpolated Texels. 67
4.12 The Kernel execution times on the down scaled *Children playing* dataset of the parallel image-based visual hull algorithm plotted as a function of the number of input images per frame. 68
4.13 Qualitative results of our GPU image-based visual hull reconstruction on *Children playing* integrated in virtual environment. (A) A reconstructed visual hull using our approach with interpolated Texels. (b) A reconstructed visual hull using our approach with interpolated Texels. 69
4.14 Computational performance comparison: image input size is 720x576 and the size of novel view is 800x600 approach with interpolated Texels. 70
4.15 comparison between the result qualities of our approach of each configuration using 36 input images from *Dinosaur toy* dataset and the size of novel view is 800x600s. 72
4.16 Theoretical architecture of our GPU image-based visual hull approach using CUDA Dynamic Parallelism. 74

5.1 Two images of a synthetic object which illustrates the multi-view stereo major problem namely reflective and homogeneous surfaces along with some thin details. 77

5.2 Example of two untextured images used as input to a photo-consistency algorithm matching a block of pixels (center of the left image presented as red point surrounded by black box) against a second image across the epipolar line. The second row illustrates the different photo-consistency measures namely the NCC and SSD computed for the above texturless images (images courtesy to Yasutaka Furukawa) 78
5.3 A non-Lambertian surface illustration using a synthetic model of bunny, the first row show three diffrent view position of the same object. due to the surface nature a given point may have different color in each image 80
5.4 Internet image collection of Timgad Roman Ruins and Masjid Sidi-Khaled. Different camera poses and focal length, also different lighting condition gives strong variation appearance. 81
5.5 Flow chart of the proposed method. 84
5.6 Stereo matching geometric configuration: Points $\mathbf{x}$ located on the planar Π at distance $h(\mathbf{u}_M)$ along viewing ray $\mathbf{r}(\mathbf{u}_M)$. $\mathbf{u}_M = (s,t)$ is central pixel and $\mathbf{u}_M = (s+i,t)$ or $\mathbf{u}_M = (s,t+j)$ are the rest. 87
5.7 Stereo matching photometric configuration : a rough surface represents a small area in the planar patch, this area is illuminated by directional light. 89
5.8 Directional light estimation. The left column represent a given view images of two diffrent scenes (Fountain and platar dino) which we want to estimate its lighting conditions. On the other hand, the right column represent the $3D$ ground truth geomtery illuminated using our estimated lighting direction 98
5.9 Large scale view clustering unstructed photographs of the founding stone of biskra university. Left the recovered camera view point along with all the cloud point extracted from SIFT feature. On the right a refrence view with its support view in green 100
5.10 The initialization process of the swarm through out the whole reconstruction process. (a) Initialization with only SIFT feature. (b) Dynamic Initialization with any correct solution. 103
5.11 Plot of a sophisticated energy function in term of a given pixel depth. The ground truth depth value is the green point, meanwhile the depth value of the initial features is illustrated as blue points 104
5.12 The MidleBury Benchmark sample images: (a) represents the 23th view in *templeRing* dataset, (b) represents the 23th view *dinoRing* dataset. 109
5.13 Visual results on *Dino* datasets with different approaches and decreasing number of input image. (a) The ground truth model. Top: reconstruction of full dataset with 363 images. (b) Our approach MVSPSO 2; (c) Goesele-MVE; (d) SMVS. Bottom : reconstruction of ring data set. (e) Our approach MVSPSO 2; (f) Our approach MVSPSO 1; (g) SMVS . . . 111
5.14 Histograms of signed errors 112
5.15 Visual results on *Temple full* dataset with different approaches (a) The ground truth model; (b) Our approach MVSPSO 2; (c) Our approach MVSPSO 1; (d) Goesele-MVE ; (e) SMVS. 112
5.16 Sample images from fountain-p11 dataset. (a) and their ground truth depth-maps (b). . . . 114
5.17 From left to right: depth error maps for the 8th image in Fountain-P11 using the our method, multi-view environment method and the shading aware multi-view stereo (SMVS) method, respectively. 115
5.18 The number of correct pixels in all images as a function of the error 116
5.19 *The angles* dataset from Wu et al. [132] reconstructed using low resolution of the original image (a) Original image ; (b) Our approach MVSPSO 2; (c) MVE (d) SMVS 116
5.20 Individual views from multiple datasets namelly the dinoRing, Achteck Turm, Fountain, and Foundation stone. Along with their corresponding depth maps and the shaded renderings of the reconstructed $3D$ model 118

To Father. Thank you for your support.
To Sabrine my dear sister. Thank you for your patience.
To Youcef my little brother. Whatever you ask may you receive, whatever you seek may you find.
To mother in the heavens.May ALLAH bless your soul.

Acknowledgments

First and foremost, I want to thank my supervisor, Prof. Babahenini Mohamed Chaouki, for giving me the opportunity to work on my PhD under his supervision and for guiding me into research. He inspired, guided and challenged me throughout the course of this thesis. Moreover, I learned several secondary skills from him, in particular, how to patient and ambitious.

The contents of this thesis have grown in collaboration with several people and this work would certainly not have been possible without my co-author. Therefore, I want to thank especially Sofiane Medjram for his time and effort.

My hearty thanks go to the members of my jury for their time necessary to read and understand the manuscript.

Special thanks goes to EX-director of the LESIA laboratory, Prof. Cherif Foudil for his support during all my years of study.

Finally, the most eminent thanks goes to my family who were unconditionally encouraging me the entirety of my life to achieve my goals and objects. also to all my relatives and friends. Thank you.

And the worldly life is not but amusement and diversion; but the home of the Hereafter is best for those who fear Allah, so will you not reason?

Al-An'aam-verse 32

1 Introduction

1.1 Background

The year is 2040, the world as we know it has gone and things look bleak. One day, everything went wrong. It was not expected but an energy crisis hit the world and the consequences of global warming coupled with overpopulation sweep through the nations. In only a few days, without resources society collapses into chaos. Therefore, people turn to a virtual reality simulator called OASIS in order to evade the decline their world is facing. Accessing this fantasy world require using Head Mounted Displays (HDM) and Haptic technology such as gloves. Immersed in such a world, people are willing to give up their real-life in favor of their virtual ones. Ernest Cline a science-fiction author, used this plot settings for his book *Ready Player One* [13] in which he gives an appealing depiction of a future very much on its way (*see*. Figure 1.1). In fact, the world we are living in has never been the same since social media and smart phones came to existence, thus alluring a near-absolute virtual presence.

Nowadays, The idea of a computer-generated virtual world has become a non-alien concept due to how society accepted video games as a new mean of entertainment, education, and even more. In fact, it is the gaming industry who is pushing forward the boundary of virtual environment, for instance the term **Open-World** was first used in video games to describe a virtual world in which the player can explore and approach objectives freely. Take *The Witcher Three* as an illustration, a typical modern video game with

Figure 1.1: The main protagonist *Wade* from *Ready Player One* film. Artistic poster depicts the blurring of fantasy and reality.

a huge **Open-World** map that has around 124 km^2 to explore with little to no restrictions, As shown in Figure 1.2, the progress in *Computer Graphics* techniques coupled with rapid hardware evolution allowed the creation of more and more realistic virtual worlds populated with high-quality avatars, thus assuring immersion. Furthermore, a wide range of applications can benefit from virtual reality. Neuroscience for instance, where scientists in this domain use VR technology to control and stimulate natural event while the subject has the freedom to navigate and interact with the virtual environment in real time. Researchers can then monitor a wide variety of neurone response made by the subject and find answers to problems that simply can not be solved by studying the subject performance in the wild [5]. Another interesting field is psychology, recently advanced virtual reality therapy techniques emerged as potentially an effective way to provide a healthy solutions for people with psychological disabilities [97].

In this modern era, the medium of VR is widely spreading due to the remarkable advances in hardware technologies. Starting from Palmer Luckey's *Oculus Rift* which represent the first modern design in 2012 to Valve and their *HTC vive* finale release in early 2016, virtual reality headsets are mass produced. Henceforth, more people will have access to the technology and the range of possible applications to explore is substantially broadens.

Figure 1.2: The Open-World concept in *The Witcher 3* (2015). A massive map (left image) of detailed virtual world (right image)

Immersion is an essential characteristic of state-of-the-art virtual reality experiences. According to Gilbert [33] this sensation of being present in a virtual environment must be derived especially through environmental realism. This raises the question of how should we generate the virtual world?. One strategy is to completely build a synthetic world triangle by triangle from scratch, however, this is not an easy task. The morphology of the three-dimensional objects is sometimes hard to capture due to its complexity. As a result the conformance between the modelled object and it original lessen. Moreover, the human brain and its visual system are effectively capable of detecting patterns and repetitions which are very present in synthetic world. Such effect can truly discards the virtual reality experience from its sense of realism. At the other end, the virtual environment might be digital version of our physical real world captured using modern Computer Vision techniques. In fact, in the area of artificial vision, we process the information captured via one or multiple digital cameras and transform the resulted data into either a decision or to a new representation such as the case of $3D$ reconstruction also known as photogrammetry. Image-based modelling approaches are powerful tools to create the most photo realistic looking virtual reality experience, as it can easily enforce immersion and remove the well-known problem of the *uncanny valley* via the massive variability of data captured in the photographs. However, it is important to comprehend that the performance of such techniques depends on the quality of the input photographs and camera parameters.

In this context, it becomes relevant to consider image-based reconstruction algorithms fused with Computer Graphics approaches to build a coherent and realistic virtual world. Nonetheless, we are well aware of the

fact that still much work remains to be done.

1.2 Problem Statement

It is in the context of virtual reality that we aim to present solutions for the *Multi-view Stereo* failure cases. In the ongoing search for the most photorealistic looking virtual environment, multi-view reconstruction shows great potential. However, the complexity of the real world and its properties make estimating a complete representation extremely hard. A popular alternative is the laser scanners also known as Light Detection and Ranging (LIDAR) technology [35, 65]. To recover the $3D$ geometry scanners use a physical real-world measurement such as the light pulse or signal phase, then directly acquire the distance between the scanner and the object scanned. However, such a tool is cost-effective. For instance, a high-end consumer laser scanner can capture at best 1.3 Megapixel with maximum working distance equal to 0.5 meter and almost $0.03mm$ depth accuracy, but their cost is in the $16000\$$. Moreover, scanning can be a rather slow process for complex shapes and structures even if these tools can measure one million points per second. On the other hand, digital cameras are capable of recording high-resolution photographs, therefore, a detailed $3D$ shapes could be recovered with accuracy almost equal to that of a scanner with the price of a few hundred dollars.

Image-based modelling is the process of automatically computing the $3D$ geometry of an object or a scene from a collection of photographs, in another term reconstructing a globally consistent model. In fact, similar to the human visual system, these techniques derive a variety of geometrical information from different visual cues such as texture, shading, contours, and stereo to name few. A good overview about of all these cues can be found in Heinrich paper [8] for more detailed information. Stereo correspondence, in particular, has been very successful in with regard to its performance. In essence, the term Multi-view stereo (MVS) is generally given to a group of algorithms that use stereo as their main cue to recover the three-dimensional shape from more than two images (*see. e.g.* [109, 121]). This dissertation provides solution to the two core problem in multi-view stereo reconstruction related to virtual reality, namely, interactivity, and the visual fidelity of reconstructed object.

Virtual reality or virtual tourism applications are the ultimate goal for image-based rendering and $3D$ reconstruction methods combined. Technically, the latency between capturing the real world and synthesising the virtual output needs to be at its lowest. Hence, the use of a graphical processing unit (GPU) became a necessity. One successful example in this domain is the *KinectFusion* by Izadi et al. [48] which exploit the GPU and utilize MVS technology with an RGB-D camera to reconstruct in real-time indoor environments.

However, the technique is not a fully passive reconstruction and recently *Microsoft* decided to cease production of the Kinect camera. Another popular solution is to reconstruct the convex hull of the object, algorithms such as *Space Carving* [57] can be implemented on the GPU due to the highly parallel problem formulation of the volumetric scene representation. However, researchers gave up on these classical approaches seeing that there is no room for improvements. In contrast, we think that with a careful design and good implementation we can squeeze more performance out of these classical approaches. A notable example is Matusik et al. [78] Image-Based Visual Hull (IBVH) approach. Unlike the original visual hull algorithm, this last is efficient in reconstructing and simultaneously rendering independent views of the scene in a short time window.

As we mentioned earlier, the technological evolution of digital camera gave multi-view stereopsis the ability to create impressive models, yet it still suffers from multiple limitations which are partially inherited from the binocular stereo methods [103]. It is important to note however, that the key idea for the stereo methods in general is matching blocks of pixels between two or more images. Undoubtedly, this process will be affected by the physical nature of the reconstructed object and its different interactions with the light.

In particular, homogeneous surfaces pose a stumbling block for these methods to achieve completeness. These poorly textured areas cause ambiguity since the informations contained in block of pixels are not sufficient to locate the match confidently in other images. One solution for such problem is to use the patch-based reconstruction approach proposed by Furukawa and Ponce [28]. Despite the algorithm's popularity, it still considered as a partial solution for untextured surface since it work only in the presence of a weak and delicate texture cues. In essence, the authors define a rectangular tangent plane centred around a point with a unit normal vector (patch) as the resulting output of the initial matching process which is based on a robust function that is applied to the photo consistency measurement. Then the algorithm alternate between two stages namely patch expansion and patch filtering respectively in order to intensify the initial patches. Thus, the algorithm can recover some geometry in areas that are poorly textured. Having considered the patch-based method and its local optimization approach, it is also reasonable to look at the globally optimal formulation approaches [54] as they show a strong resilience to untextured surfaces. Take for example Kostrikov et al. [55] volumetric approach, they propose a probabilistic model to the data term of a globally defined energy functional. The probability of each voxel being background or object is determined from an independently selected subset of all cameras, this help to determine outliers which do mainly occur in untextured regions.

Thin-details is another impediment for the standard MVS algorithms [34, 119], this is due to the consistency evaluation been done over a window of several pixels wide in order to assure robustness. One obvious solution is to use a pixel-wise photo consistency measurement, which has been proven to be not effective, instead a detail-preserving similarity measure can be more robust, this was sustained by the work of Li et al. [69]. The authors of these latter paper have proposed a novel inter-image similarity measure that is the result of guided image filtering [40] with image registration. The novel similarity measure is designed to capture fine-scale details of the reconstructed surface. Such approach however is very complex especially in a variational framework which is by nature tend to flatten the details. These small scale structures are often captured by shading changes across the surface, which is totally natural, since changes in surface orientation translate into corresponding variations on the image intensity. Evidence for in support of this position, can be found in recent work of Wu et al. [132]. Their approach is focused on applying shading-based refinement to the geometry computed via multi-view stereo. The workflow of Wu et al. [132] method consist of three stages, in order to assure a detail preserving reconstruction they first start by computing the $3D$ geometry of the object or scene in study using Furukawa and Ponce [28] MVS approach. Then the resulted model is exploited to estimate what is considered to be fixed and distant illumination. Finally, the geometry is refined so that the shading changes in the images are well explained by their multi-view shading gradient error. The downfall of refinement-based approaches lies in the initial model which is treated as fixed ground truth. Hence, any geometrical errors or ambiguity produced by the MVS methods are discarded.

Finally, most if not all MVS methods [108] describe the shapes that are been reconstructed as pure Lambertian surfaces. In other word, these surfaces reflect the same amount of energy in all direction according to Lambert in his book *Photometria* [60]. Hence, in a multi-view stereo setup any point in a Lambertian surface must appear with exact same color in all images. This assumption gave good reconstruction results across the years, However it is not realistic and multi-view stereo methods break apart in multiple cases.

1.3 Contributions and Thesis Outline

Throughout the course of this dissertation, we provide solutions to the above mentioned core problems of multi-view stereo that are related to virtual reality domain. Our contribution have been oriented along two axes, where the first axis is more experimental while the other is methodological driven work.

1.3.1 Real-time and Interactivity

Nowadays, when we talk about real-time in Computer Graphics, we are forced to consider the modern Graphical Processing Units also known as *The GPU's*. In fact, the head mounted displays used in VR experiences contain two screens which is essential for the stereoscopic effect. This made the effort of rendering the virtual environment doubles in term of computation. However, synthesizing a novel-view from multiple input images offers a good solution for such a problem. In fact, We present a massively parallel implementation for high-quality visual hull rendering on a single machine. Starting from multiple input images or video streams, we launch a *GPU* kernel every frame to extract the necessary silhouette for the reconstruction. As a matter of fact, if we only aimed to recover these $2D$ binary images in real-time, we would have to use the provided device texture memory since it is cached and optimized for $2D$ spatial locality. However, we desire to use the silhouette images as inputs for a second kernel, which estimate the visual part of the visual hull and synthesize it from a novel viewpoint. In this case, using the device texture memory will force the system to wait until the graphical processing unit computes the silhouette image for each input view, then build an array of textures with that data and send it back to the device. Such process lead to a huge bottleneck thus reducing significantly the performance in real-time scenarios. In order to minimize the data transfers between the device and the host (*CPU*). We consider using the device global memory as hub space for all the data. Consequently, every frame sent by each camera is uploaded only ones to the *GPU*, where it will be processed and cached until the end of the reconstruction.

For the reconstruction process, we propose a parallel implementation of the ray casting strategy [100]. Using a *GPU* kernel, each pixel in the desired novel view cast a ray to search for silhouette consistent point in a pre-defined volume, if such point is found a color for it is sampled from the input images and given to the pixel in the final render. Finally, the results of our implementation, gave us insights into the power of the shelf devices. Hence, we provide a theoretical design coupled with critical analysis for a system that harness the computation power of multiple *GPU* devices which are based on Kepler architecture, and multi-cores machine.

1.3.2 Multi-view Stereo Under General conditions

In this dissertation, we frame the classical dilemma of multi-view stereo reconstruction as a local search problem for the optimal depth, orientation and roughness on every input image [102, 101]. In our approach to solve the MVS problems namely, the untextured regions, thin details and quasi Lambertien surfaces, we focus on designing a robust photometric and geometric models for the optimization process. Before that,

our method start by matching initial features using a SIFT feature descriptors [73] which are often robust to large range of noise types. These features are used as relabel initialisation mechanism for our optimization algorithm. In particular, we introduce a new geometric model for patch-based reconstruction algorithm, along side an enhanced photometric model which balance between stereo and shading. The later is based on non-lambertian model. Then we prove that the introduced models can produce a sophisticated energy/objective function that can be optimized via a metaheuristic algorithm namely Particle Swarm Optimization. Our search for optimal depth and orientation of a given patch is robust to quasi lambertian surfaces and textureless area and can recover thin and small details.

1.3.3 Thesis Outline

The remainder of this dissertation is structured as follows: Chapter 2 opens a window through which one sees the reality of a virtual world and summarizes the fundamentals of virtual reality creation. We then present in detail the problem of creating three dimensional geometry from multiple input images followed by a survey of the state of the art relevant to the covered topic (Chapter 3). Our implementation of an interactive reconstruction system is introduced in Chapter 4 along with the description of the proposed parallelization strategy of the image-based visual hull. The complexity of the algorithm is also discussed in this chapter. Next, we describe a solution to the above-mentioned problems in multi-view stereo (Chapter 5). The manuscript ends with Chapter 6 which concludes the dissertation and gives an outlook on further possible future work and extensions.

In my dreams I found a little of the beauty I had vainly sought in life,
and wandered through old gardens and enchanted woods.

H. P. Lovecraft

2

The Digital World in Virtual Reality

Technology is what differentiate us from the ancient civilizations. In fact, the computing power is raising exponentially from one year to another and the latest technological components have enabled some amazing and compelling virtual reality experiences. However, We think that these pieces of technology only acts as a link between the experience and another core component. This last is called the human brain, a massively parallel organic structure which has about 100 billion neurons capable of perceiving all sorts of things. Evidently, all of our interactions within our real world or these different virtual worlds start with the perception as it is described by George Mather [77] in the first chapter of his book *Foundations of Sensation and Perception*. Our brains also capable of creating these wonderful experiences called dreams without the intervention of any external device. Almost all of us had that dream of falling from a high-ground to just wake up before hitting the bottom of the pit screaming in fear and scared for our life. Although it was not a pleasant experience, yet it shares the same characteristics as any other virtual experience namely, immersion and interaction. Perhaps virtual reality is the manifestation of the human desire to bend the dreams under his will ?.

In virtual reality the technology is evolving very fast lately, almost every year a new piece of hardware is presented to the consumer. Hence, if we want to stay ahead in terms of knowledge, we have to read relevant papers almost daily and look throughout the internet or keep track of what is happening on the media. In fact, corporations such as *Facebook* and *Sony* are racing to make low budget consumer products

and they will keep on improving very rapidly. In this chapter, however, we want to focus on fundamentals and essential concepts that should be more invariant with respect to time.

2.1 Defining The Reality of Virtual Reality

Linguistically, the term *Virtual Reality* represents what is known as an oxymoron in the English language. In fact, according to Merriam-Webster's [*] website, the word virtual is defined as:

" Being such in essence or effect though not formally recognized or admitted".

While the word reality means:

" The state or quality of being real, something that is neither derivative nor dependent but exists necessarily"

Putting these two words together to represent a new medium in the multimedia industry led to multiple and sometimes inconsistent ways to describe it. Take for example Burdea. et.al [7] in their book, they suggest that in order to be able to define virtual reality, one has to understand what is not VR. Doing so lead the virtual reality to be described in term of its functionality as follow :

Definition 1. Virtual reality is high-end user-computer interface that involves real time simulation and interactions through multiple sensorial channels. These sensorial modalities are visual, auditory, tactical, smell and taste.

Such definition depicts VR as an interactive human-machine interface which is very limiting, for example, would we consider watching a film via a head mounted displays as virtual reality experience ?. In fact, this medium is much more than interactive systems working with different modalities. Sherman and his co-authors [113] have another opinion, and they suggest in their book *Understanding Virtual Reality* that the description of the virtual reality is built upon four important pillars.

First, the Virtual World which a stage where all the action will happen. Such a world could be a copy of our real world or inspired by it, otherwise, it can be an imaginary space with its own set of rules and fundamentals. Often these worlds are manifested through some sort of a medium. Second, Immersion which is considered as the core aspect for virtual reality experiences. In particular, being immersed requires from the subject to be mentally engaged and involved in the virtual world. The third pillar is the Sensory

[*] www.merriam-webster.com

Feedback, a purely technical aspect of virtual reality. Moreover, The VR system via its tools must be able to guarantee direct feedback to the users based on their position and actions. Finally, it is preferable for a virtual reality system to provide some degree of interactivity like changing the user's viewpoint in virtual travel experience for example. Each of these key elements makes an important contribution to the definition of virtual reality, hence according to [113] this last is described as follow:

Definition 2. Virtual reality a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation (a virtual world)

Given the current high profile debate with regard to virtual reality, it is quite surprising that a fundamental definition was not given. Such description must captures the most essential aspect of VR while being narrow enough to discard any misunderstanding while staying technology independent. However, in 2015, Steve LaValle manage to put together a solid definition for VR after he finished his work on the *Occuls rift* project. In his book [63] titled *Virtual Reality* he describe this medium as :

Definition 3. Inducing targeted behaviour in an organism by using artificial sensory stimulation, while the organism has little or no awareness of the interference.

This definition has stretched the boundaries of what is VR, for example, take the word organism it can be human (most of the times) or maybe something else. Also the word artificial represents the technology used to take over the ordinary sensory input and replacing it with something that has been engineered thus leading the organism to feel fully immersed. Based on this, we gave a simpler definition that can be used by non-scholars without any confusions.

Definition 4. Virtual reality is an engineered dream like experience that can be controlled and designed to fulfil a specific task.

So in some sense, the participant is not consciously aware that he is been deceived at that moment, he might remember putting a virtual reality headset at some point, however, while in the experience it is very easy to forget that, and believe that he is somewhere else. All things considered, an ideal VR system is the combination of two parts (*see* Figure 2.1 (a)). There is the human part which is the source of all interactivity and also the receiving end of the system. And then there is the engineered part where the hardware and software reside. In fact, these two parts work in collaboration where both software and hardware provide intuitive pieces of information and replace the real world that is perceived by the human

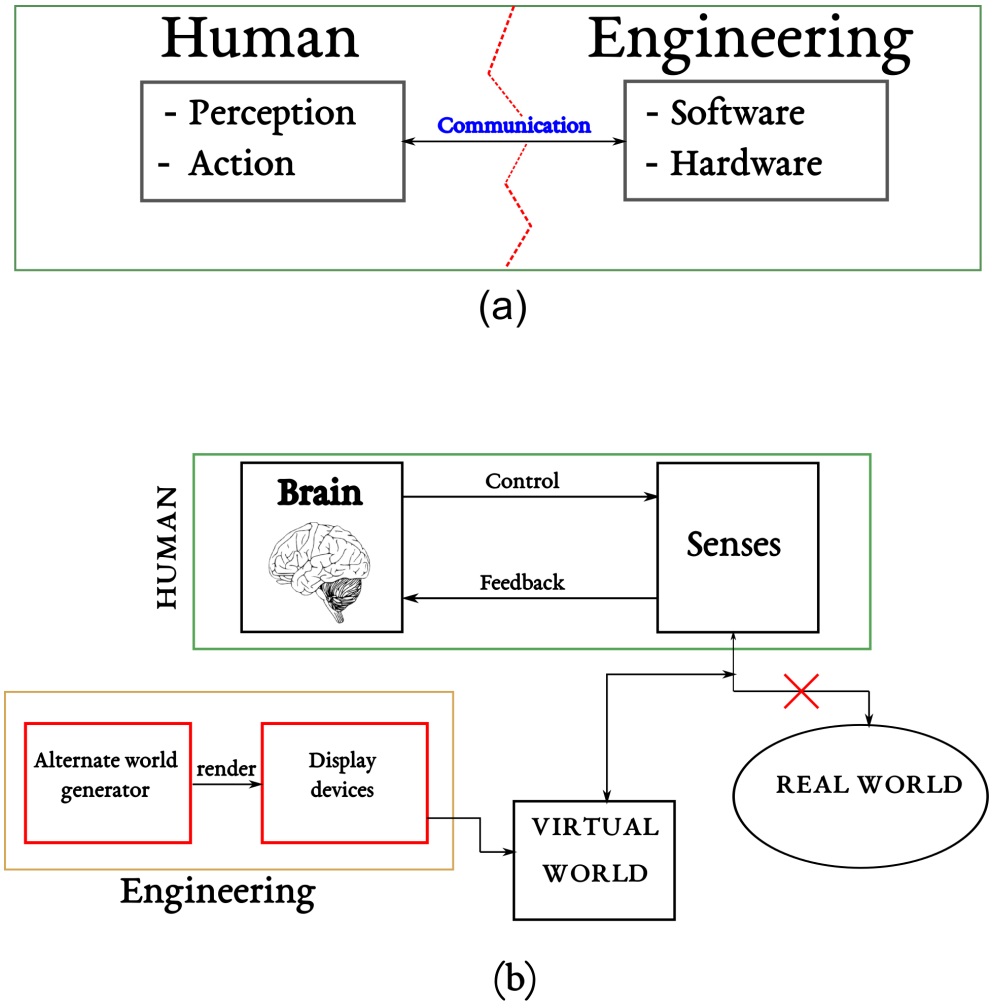


Figure 2.1: Illustration of Virtual Reality System overview (a) and detailed presentation of that system (b).

part in the VR system as shown in Figure 2.1 (b). In particular, for this dissertation, we focus our efforts on the engineered part of the system, more specifically the world generator process since it is the essence of the whole system. However, the following subsections will describe and analyze the two parts of the VR system in details.

2.1.1 A Psychophysics Analysis

As we mentioned earlier, the term virtual reality itself is contradictory, which by itself a philosophical dilemma. Indeed, from a scientific perspective, the reality we perceive is all but a mental reconstruction of what is really out there. Our brain combines the sensory signals with its prior expectations and believes to form its reality and when someone's reality matches the rest of the world we call that the true reality. One of the experiences made to support that claim is called "*The Rubber Hand Illusion*". The subjects were sat down in front of a table with their left hand hidden out of sight. However, a lifelike copy of that hand

Figure 2.2: The dress has confused the world in 2014 where people couldn't agree on its color.

made of rubber is positioned in front of them to mimic the original hand. The experimenters touched both the subjects hidden left hand and the visible rubber hand in the same place multiple times while subjects are looking. Then without any prior warning, the experimenters strike the rubber hand with a knife, the subjects immediately react as if its his real left hand that was attacked.

Evidently, if we really want to create a good VR experience, we have to understand some of the reverse engineering of the human brain and its sensors. As we mentioned at the beginning of this chapter, the human brain has a huge number of neurons. However, a large cluster of these neurons are devoted to the visual system and perception which make as visual based creators. Therefore most of the effort that is done to enhance VR experience is mainly focused on solving visual problems.

Colors

Color perception is the outcome of the human physiology and how the neurons are structured in the brain, thus, perceiving the natural color of a given surface is considered to be a subjective process. Take a look at Figure 2.2, It is a picture of dress[†] which was worn by some celebrity in 2014. However, the internet was divided into two major factions depending on what color they see, some see it as blue and black the others however perceive it as white and gold dress. This is known as constancy. In fact, colors are never really themselves. Depending on the light sources, white can look to our eyes as red, gray, or blue. Lighting

[†]www.independent.co.uk/

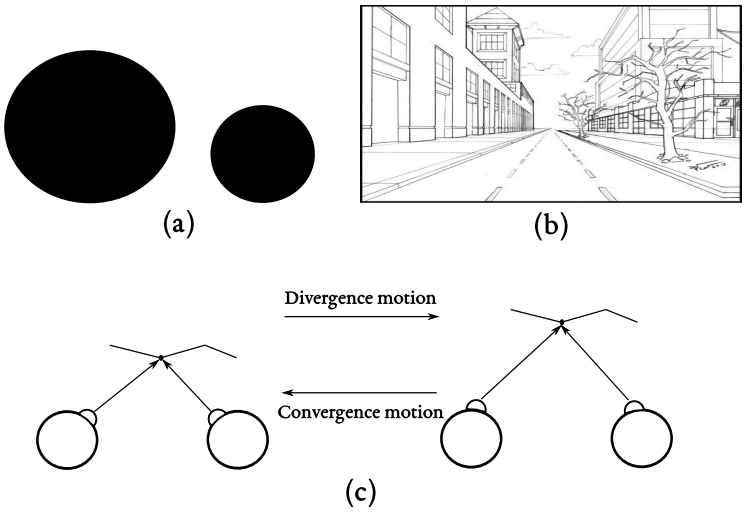


Figure 2.3: Graphical illustration of the monocular and some binocular cues for depth perception (a) sight cue, (b) perspective cue, (c) stereo cue.

change the color of things it illuminates, however, our brain is great at perceiving the original color rather than the color we actually see.

Depth

Depth is an important factor for almost all the mammals to perceive the world. However, how can we see $3D$ shapes and figures around us when the light trace and the image fall onto a two-dimensional retina. Over the years, a number of researches have been conducted and finally, it was understood that the depth perception is caused due to the presence of two types of sensory cues. First, the visual monocular depth cues which are clues generated by a vision from a single eye. On the other hand, a portion of the depth perception happens due to the stereo cues, which are the kind of visual experience we had when both eyes are used. It is important to note, however, that there are many monocular cues than the stereo, hence our ability to infer or perceive depth information from single a single photographs.

For example, in the monocular cues, there is the cue of sight, the larger is the projection of an object on the retina, the bigger we perceive it. Hence, in a scene if we perceive an object as larger, we feel that it is closer to us (*see* Figure 2.3 (a)). The second monocular cue is that of linear perceptive, It states, that as the distance of an object in the scene increases from our point of the view, parallel lines tend to converge to a singular point. This effect gives us the perception of distances, when things converge together we can feel that these objects are quite distant from us (*see* Figure 2.3 (b)[‡]). Having considered some examples for the monocular cues, it is also reasonable to look at stereo cue. Each eye receives a slightly different view.

[‡]www.johnsparrefors.wordpress.com

The two images are fused together via the eye movement which also known as vergence. In particular, the closer the object a convergence motion occurs this means that both eyes rotate to the inside. If the object is further away then a divergence motion occurs, which forces the eyes to rotate further apart as it is illustrated in Figure 2.3 (c).

2.1.2 A Technological Analysis

According to Moore's law , since 1965 the number of transistors per square inch on integrated circuits will be doubled every year. Hence, the computing power rises rapidly. In fact, at the GPU Technology Conference (GTC) that was held in Beijing China 2017 Nvidia CEO Jensen Huang declared that the trend of Moore's law is absolutely nullified due to the huge advancement in the graphical processing unit design. He also said that GPUs will soon replace CPUs. This declaration were the conclusion of an early observation made by Luebke et al. [74] where the performance of the GPU was increasing by factor of 3 every year. Such effect is considered to be good news for the VR community given that these huge computational power and parallelism will allow high-performance, graphically rich, immersive and interactive $3D$ virtual experiences.

It can be seen from the overall structure of the VR system that, the hardware modalities are grouped into three classes. First, the display devices, these are the output of the system which stimulate the human senses like for example TV screen or speakers. The second class of modalities is the sensors devices, they act as inputs for the system by capturing information from the real world such as the user position. Finally, the computers which are devices that receives the input data that generate the virtual world accordingly then send it to the display device. Nowadays, almost any home computer is capable of doing virtual reality as long as it equipped with a *GPU*. Hence, we are not covering such class of devices in this chapter.

Display

Visual display technology is one of the primary output devices used for VR. Although there are multiple way to display the rendered virtual world and generate a stimulus for the human visual system such CAVE-like displays, virtual tables, or panoramic screens. Time and experience proved that the most effective display devices that offers are the Head Mounted Display (HMD) also known as the headsets. These devices have undergone a massive evolution over the years and became the piece of technology that all people associate with immersive virtual reality. HMDs possess a variety of properties [113] such as *contrast, number of display channels, focal distance, field of view, graphics latency tolerance and user mobility*. The

contrast is the ratio of the luminance of all pixels in white to the luminance of all pixels in black. Hence, a higher contrast ranger will enhance the amount of detail in blacks areas.

For example, HTC vive headset uses OLED display screen with contrast value equal to $5000.0 : 1.0$. The number of display channels is preferred to be two for the stereopsis effect, the vive HMD contains two display channels each one has a resolution of 1080 x 1200 thus the displayed virtual world on such screens look much crisper. Focal distance on the other hand, also known as the eye relief is the distance between the user's eyes and the headset optical system. It defines the distance at which the user can see the full view, such property must be under the user's control for an easy adjustment. the field of view or simply *FOV* is the horizontal and vertical angular width of the user's optical system. HMD's that offer 100 to 120 degree of *FOV* such as the HTC vive headset with *FOV* equal to 110° are considered to be ideal for VR headset, since it covers a reasonable portion of the human viewing range. And finally, the user mobility which to this day poses a problem when designing a virtual reality experience due to the user been constrained by the cables tethering or tracking systems that have limited range.

Sensors

Sensors are a multitude of devices dedicated to provide real-world information namely position (three degrees of freedom) and orientation (three degrees of freedom) to the virtual reality system. The combination of these input devices with the software side of VR is called tracking systems. For head-based visual displays and sometimes the participant's hands, the positional cues are tracked by multiple sensors. Such trackers inform explicitly within small window of time the VR system about the user accurate location inside the virtual environment. Positional and orientation tracking can be expressed with a variety of technologies such as mechanical, electromagnetic and ultrasonic trackers (*see*. Chapter three of [113] or Chapter five of [127]).

The most widely adopted technologies in this modern era of virtual reality, however, are the inertial measurement unit also known as *IMU* coupled with advanced optical tracking. First in order to estimate the head orientation, three angular measurement namely Yaw, Pitch and Roll must be deduced from the inertial sensors [64]. The gyroscope measures angular velocity in radians per second for the three axes, the data then accumulated over time to obtain an estimate the orientation plus small drift of error that contain to grow up unless it is corrected using accelerometer which measures linear acceleration as the sum of gravity with other forces. Positional tracking on the other hand is implemented with optical technology.

For instance in Oculus Rift and Sony's Playstation VR headset, one or more camera are placed in the environment in order to look for a given marker and observe it. This markers are usually infrared LED points or actively illuminated retro-reflective small markers which are mounted on devices and calibrated by the manufacturer of the VR system. The position is then estimated from the set of $3D$ points (markers) in the world and their corresponding $2D$ projections on the image plane. In computer vision , such problem is known as the perspective-n-point (*PnP*) problem [68]. It is very easy to implement such method using a camera and image processing, However, it is more reliable to detect implicitly the line of sight between the marker and its observer.

The HTC Vive were the first to propose this idea and call their technology Lighthouse approach [96]. In contrast to the normal optical tracking, the Lighthouse approach replace the normal camera by projector and the LEDs are removed and replaced by photodiodes which mounted on the devices (Headset and controllers). The projectors sweep the room horizontally and vertically and emits light that illuminates the photodiodes thus recovering their $2D$ location in the virtual frame of the lighthouse.

2.2 Modeling The Virtual World

The most important thing in the virtual reality experience is the virtual environment it self. From the consumer perspective, it is a synthetic world that should feel immersive. However from an engineer point of the view, every object in this virtual world is but combination of multiple triangles planes situated in a three-dimensional Cartesian coordinate system. These triangles are considered to be the most atomic and primitive geometry that represent a planar which make it very memory efficient and can be sorted and rendered extremely fast using the current hardware. In fact, the process of creating $3D$ polygonal models is called geometric modeling which is essentially creating a virtual representation of anything. Such process needs an artistic and creative efforts thus it is time consuming.

Look to Figure 2.4 for example, simple scenes like these with few objects could take almost a day to build. There are different methods to generate $3D$ data. Moreover from a software perspective, all this methods can be performed by means of a special $3D$ modeling systems such as *Cinema 4D, Maya, 3ds Max or Blender*. One common thread of most of these systems is that they can represent their output models in polygonal form.

Figure 2.4: Virtual objects modeled with great details to match their real world counter part using Blender. As it can be seen the 3D modeling is considred a new medium of art in our modern age

2.2.1 Classical Polygonal Modeling

The modelers in this class describe a surface either as an explicit polygonal plane or implicate functions. A range of different actions and direct manipulation upon the surfaces are made to increase or lower the point count of the model, hence, controlling the amount of details needed for the application, in this case virtual reality. In particular, polygonal manipulation focuses on explicitly positioning the individual elements of the polygons namely the vertices, edges or the whole faces so they completely tile the created surfaces at a given resolution. One of more popular methods of creating polygonal meshes is subdivision a powerful tool in defining smoothness [1], such tool split the original edges and faces of the polygon into multiple pieces thus provide an infinite level of details (*see* Figure 2.5).

On the other the implicit surfaces are used to describe impressive and sophisticated organic effects or even landscapes by creating surfaces that are described mathematically via an implicit function $f(x,y,z) = 0$. Furthermore, such representation is more flexible to generate a complicated structure using lesser number

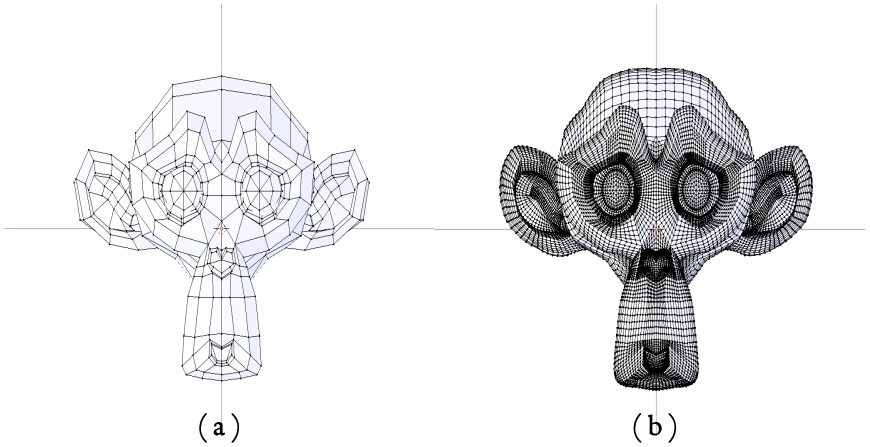


Figure 2.5: Subdivision process, the left image (a) shows the original polygonal mesh, i.e, control surfaces, The following image (b) is subdivided three times. As can be seen, more and more vertices are generated and the surface is getting smoother

of vertices while achieving a higher order of smoothness. Visualizing such type of representation can be achieved by the ray tracing algorithms [38], polygonalization techniques [15] can also be used to create a polygonal mesh for display.

2.2.2 Procedural Generated Modeling

While the classical modeling techniques prove to be effective, it is impossible however to over look the amount of effort required to generate large scale models such as buildings and cities or even planets. The creator often spend a lot of time performing the same routines and heavy tasks eventually, the resulted models are susceptible to errors and lack creativity. As a consequence, procedural generation approach was introduced to face these issues.

In fact, procedural modeling is the process of generating $3D$ models automatically or semi-automatically based on set of rules and coupled with some element of randomness. Merrell et a.l [80] for instance use a well known technique namely model synthesis which take as input an initial $3D$ mesh and generates a larger and more complex models which share the same local features of the initial inputs. Other well known algorithm are scripting languages, L-systems, fractals which are used for modeling landscapes and trees [81]. The most famous example that extensively used this approach of modeling is a video game created by *Hello games*[§] called *No man's sky* 2015. The game offers a procedurally generated deterministic open universe as it can bee seeing in Figure 2.6. For more information on this topic readers are encouraged to find more in a recent survey by Freiknecht et a.l. [23] for the sate-of-art algorithms that cover the automatic

[§]www.hellogames.org

Figure 2.6: The power of procedural generating algorithms in content creation for virtual world. This images from E3 2015 press conference presentation of *No man's sky* where (a) represent the whole universe and (b) represent a planet inside that universe and finally (c) is the world inside that planet all of it was procedurally generated (Image curtesy to Push Square youtube channel)

generation of content for virtual worlds.

2.2.3 Image-Based Modeling

Image-based modeling is the process of using more than one photograph to recover the three dimensional geometry and appearance of real objects. Consequently, This passive technology [30, 103, 108, 116, 121] offer a simple way for acquiring graphical models that are characterized with impressive details and realism by recording the physical world using modern digital cameras.

The idea behind image-based modeling is the extensive usage of depth cues which we discussed in Sect. 2.1.1. Provided that one or more cues is available in the input photographs, several systems try to exploit multiple cues in conjunction. The most successful example is the combination of the shading cue with stereo [132] as already hinted in the previous chapter. The initial shape produced from the multi-view stereo algorithm provides a depth range in which shading can refine the surfaces in ambiguous areas. It is worth mentioning however, that this type of modeling has to be coupled with the standard modeling and post-processing techniques in order to generate a believable and realistic virtual environment (*see*. Figure

Figure 2.7: "The Vanishing of Ethan Carter", one of the best looking game since 2015. This is largely due to its use of multi-view stereo reconstruction coupled with sophisticated post-processing (the picture is a screen from a gameplay)

2.7).

2.3 Rendering The Virtual World

Visual rendering is the craft of synthesising imagery by means of computer. Moreover, if the images were generated within short time window that allows action and interaction, than this process is called real-time rendering. Noting the compelling nature of virtual reality experience however, it is important for the models to have the correct three-dimensional shape and size, and also the desired visual appearances close to that of the real objects (photorealism) when rendering the virtual world and display it. Technically speaking, computer graphics algorithms attempt to simulate the ways in which light and surface materials behave in the reality.

In other words, given a sensor (human eyes or digital camera) positioned in the world, the amount light received by that sensor is equal to the contribution of all light sources in addition to that of the different interactions throughout the scene, where some part of these light rays is absorbed into the object surface, while the rest is scattered and propagates in new directions. This can be explain mathematically using the famous *Rendering Equation*.

Introduced by Kajiya [49] in 1986, this equation is considered as the foundation for all rendering algorithms. In it's most basic form, the equation states that the outgoing radiance $L_o(\mathbf{x}, \Theta_o)$ which is the amount of

energy leaving the surface from a point $\mathbf{x}$ in a given direction Θ_o is equal to the sum of self emitted radiance $L_e(\mathbf{x}, \Theta_o)$ and the reflected radiance from other sources of lights and objects $L_r(\mathbf{x}, \Theta_o)$:

$$L_o(\mathbf{x}, \Theta_o) = L_e(\mathbf{x}, \Theta_o) + L_r(\mathbf{x}, \Theta_o) \tag{2.1}$$

The equation Eq. 2.1 is built upon radiometric quantities which are physical measurements of electromagnetic energy also known in computer graphics as brightness. These measurements along with the optical characteristic of each surface in the $3D$ models are required to render the virtual world in the most physically plausible way:

Radiant Flux -Φ-: It also called the radiant power which the amount of energy per unit of time which measured in $Watt = Joules/sec$

Radiant Flux Area Density -u-: The infinitesimal amount of radiant flux for that infinitesimal surface area measured in $[Watt/m^2]$.

$$u = \frac{d\Phi}{dA}$$

The radiant flux area density represents two quantities according to direction of the direction of the radiant flux. If it is incident flux than this measurement represent the *Irradiance* and it is donated as E. However, if the flux is leaving the surface the entity is called *Radiosity* which notated by B

Radiance -L-: Radiant flux per unit area on the surface coming in per solid angle (incident or leaving from a point on surface area at an angle θ to the surface normal) measured in $[Watt/m^2.sr]$

$$L = \frac{d^2\Phi}{d\omega \cdot dA \cdot \cos\theta}$$

The radiance equation is useful since it tell us of how bright the surface appears from a given point of view. According to this we can re-write the *Irradiance* in term of incoming radiance as follow:

$$E(\mathbf{x}, \Theta_i) = L_i(\mathbf{x}, \Theta_i) \cdot \cos\theta_i \cdot d\omega_i \tag{2.2}$$

2.3.1 The Optical characteristic of surfaces

It is know that reflectance is a physical property that describes how surfaces reflect incident light. The visual appearance of several objects are determined by their reflectance properties, Thus, in order to achieve the highest degree of visual realism, these reflective properties of the surface material must be taken into

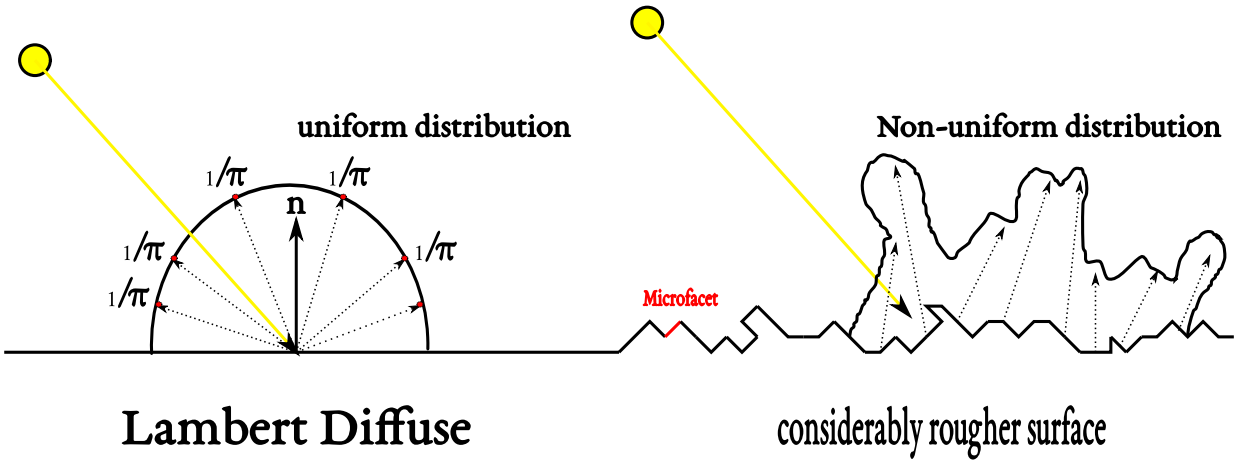


Figure 2.8: Real objects can have a complex reflection that can be modeled by a BRDF. Left image represent simple diffuse reflection surface modelled via a Lambertian law. On the right the microfacet BRDF model of reflection

consideration. In fact, there are three types of reflective materials in the nature. First, the specular reflectance surface, such surface reflect the incoming light in the mirror reflection direction. The second type of reflection is present in the diffuse surfaces, where the irradience hit the surface to be re-emitted again scattering in all directions.

Finally some surfaces combine between specularity and diffuse reflectance to form a glossy effect. We characterize these different material properties mathematically based on Bidirectional Reflectance Distribution Functions or simply BRDFs [87]. As its name implies, the function can determine how the reflected radiance is distributed in term of the distribution of the irradiance. In Particular, it is a function with multiple parameters that gives a measure for the amount of radiance coming from a given direction Θ_i and reflected in the Θ_o after hitting a point $\mathbf{x}$ on the surface. The BRDF function is in the form of :

$$f_r(\mathbf{x}, \Theta_i, \Theta_o) = \frac{dL_o(\mathbf{x}, \Theta_o)}{E(\mathbf{x}, \Theta_i)} \tag{2.3}$$

Shading Models

Suppose that we have the BRDF of a given surface, it is possible to compute the amount of the radiance reflected from a point $\mathbf{x}$ toward the eye or in particular the screen. According to equation Eq. 2.3, the reflectance equation is described as follow:

$$L_r(\mathbf{x}, \Theta_o) = \int_\Omega f_r(\mathbf{x}, \Theta_i, \Theta_o) \cdot L_i(\mathbf{x}, \Theta_i) \cdot \cos\theta_i \cdot d\omega_i \tag{2.4}$$

For many objects, the optical characteristic of a surface can be approximated by various reflection models. For instance, a perfectly diffuse surface does not absorb any fraction of the incident light. Hence, when illuminated the surface appears equally bright from all viewing direction. This effect is the result of distributing the illumination with same amount of energy in all directions. Mathematically this can be modeled by given an equal probability to every direction inside the hemisphere covering a point **x** on the surface (*see* Figure 2.8 left image)

$$f_r(\mathbf{x}, \Theta_i, \Theta_o) = \frac{1}{\pi} \tag{2.5}$$

From both equations Eq. 2.4 and Eq. 2.5 we deduce that the radiance of the so called Lambertian surface is totally independent of the viewing directions, however, it relay on the area foreshortening given by the incident polar angle θ_i.

Some BRDF's are modelled based on the microgeometry (*see* Figure 2.8 right image) and the Fresnel effect. In fact, surfaces like metal is but a collection of microfacets each one of these is a small planar Fresnel mirror. Hence, the distributions of the microfacets slopes and surface normals determine the macroscopic behaviour of light [124]. Such model account for geometrical effects like masking and self-shadowing, and also can predict off-specular reflection. The complete BRDF function is as follows:

$$f_r(\mathbf{x}, \Theta_i, \Theta_o) = \frac{1}{\pi} + \frac{1}{4\pi \cdot \cos\theta_i} \cdot D(h) \cdot F(\Theta_o) \cdot G(\Theta_i, \Theta_o) \tag{2.6}$$

The term $D(h)$ describe the distribution microfacets normals which are aligned relative to the halfway vector h. Usually it is defined using Gaussian distribution function. $F(\Theta_o)$ represent the Fresnel factor which describe the ratio of light that gets reflected over the light that gets refracted and it a value in the range of [0 - 1.0]. This factor is computed using Schlick approximation. Finally Shadowing and masking are accounted for via $G(\Theta_i, \Theta_o)$ the geometrical attenuation.

The microfacte model inspired many other works such as Oren and Nayar 1994 which will be described with details in Chapter (5), each model offers a new approach on the calculation of the functions D, F and G. There is also many physical-based BRDFs models described in the literature some of them can be found in a survey [56] and a detailed description with the mathematically theory behind these models is given in [1].

2.3.2 Rendering Algorithms

Many algorithms have been proposed across the years to render the virtual world, with many publications available, each approach search to achieve a photo realistic images in real-time. Nonetheless, the number of proposed methods has continued to expand, and we can distinguish two main methodologies. Some algorithm start rendering the scene by looping through the image pixel by pixel and determine the final color, these is known as the image-order rendering algorithm and the main tool used in the rendering process is the *Ray tracing* algorithm. Such algorithm was considered a dead end for real time graphics. However, recently in 2018 Nvidia released it new graphical hardware *The Geforce RTX*[¶] which is a GPU build to do ray tracing internally in real-time, undoubtedly this will offer researcher to reconsider ray tracing.

On the other hand, some algorithm render the virtual world by looping through each object and project it into the screen after estimating its color, this approach is called *Rasterization*. In fact, this is how real-time rendering is done for more than 20 years [50] and the reason is that such approach handle each point from the object individually, hence, the same operation can be executed for each vertex point in parallel. However, complex lighting produced by sophisticated BRDF's could not be achieved due to the sheer amount of calculation. As a consequence, advanced techniques are used to simulate the physical-based rendering in real-time, for interested reader please check the real-time rendering Third edition [1] for all state-of-art algorithms and optimization techniques.

2.4 Conclusion

The virtual reality is an interesting research area, this chapter was designed to give the reader a small overview on this new domain. We start by giving the reader a multiple definition of the virtual world and what virtual reality represents both in term of the used technology or the psychophysics side of things. we also described some of the state-of-art algorithm and techniques related to modelling and rendering the virtual world. In fact, the informations provided in this chapter were selected carefully to put this dissertation into context, the next chapter is going to be an overview about the multi-view stereo reconstruction well we will present the necessary knowledge for the reader to be able grasp our contributions later on.

[¶]https://www.youtube.com/watch?v=Mrixi27G9yM

Where words are restrained, the eyes often talk a great deal.

Samuel Richardson

3

Multi-view Stereo Reconstruction

More than three decades have passed, yet to this day reconstructing three-dimensional geometry from a set of photographs still occupies researchers in Computer Vision and it is considered a really active problem. In practice, a broad spectrum of applications could benefit from the results obtained after solving such vision problem. These applications range from computational photography and cultural heritage archival to 3*D* mapping and even virtual reality. However, it is only recently that these techniques have developed enough to be used outside of a laboratory controlled environment and provide a robust and accurate reconstruction at large scale for industrial purposes.

Digitizing real world objects or scenes is a challenging task. In fact a variety of tools and approaches were used to model the 3*D* geometry such as arm-mounted probes or active methods [2, 84, 133, 135] and passive image-based methods [70, 72, 98, 108, 121]. Among all and passive approaches more specifically, multi-view stereo methods which are the subject of this dissertation provide an acceptable 3*D* reconstructions with minimal cost compared to other active approaches. The reason behind such performance can be tracked to the technological evolution of consumer-grade digital camera. This last can be considered nowadays as high resolution sensor that can provides outstanding quality 3*D* content.

In general, the main objective of an image-based 3*D* reconstruction algorithm can be expressed as following *"Given a set of photographs of real objects or a scene, estimate the closest Three-dimensional*

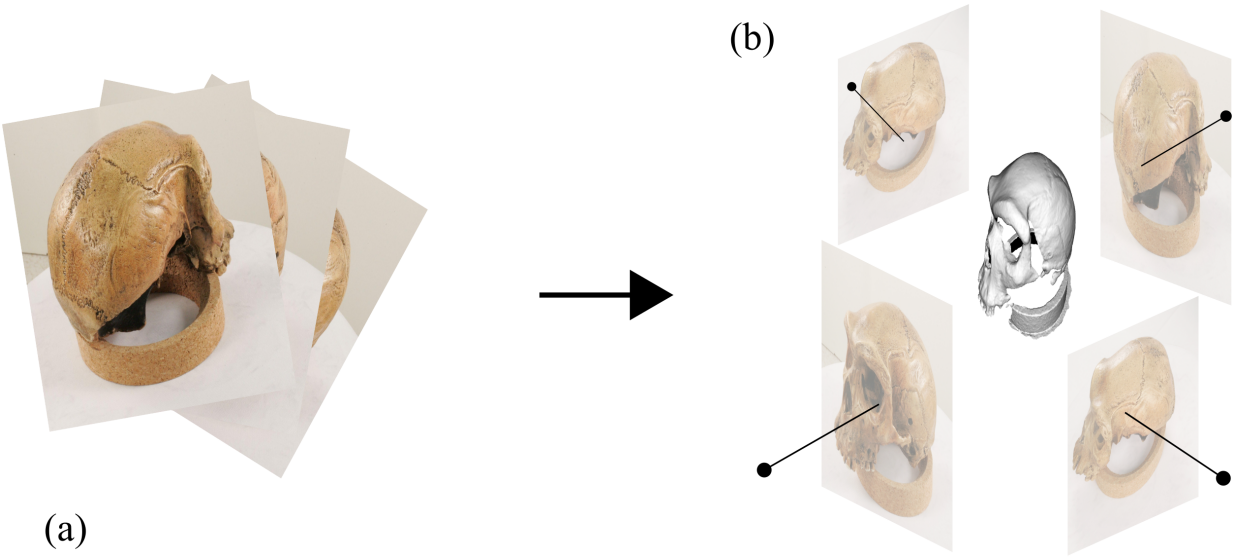


Figure 3.1: Photogrammetry. Given a set of photographs (a), the objective of Photogrammetry reconstruction algorithms is to infer the most likely $3D$ geometry that explains those photographs (b).

geometry that specifically explains those photographs while taken into consideration the surface materials, viewpoints, and lighting conditions" (*see* Figure 3.1). The definition highlights clearly the difficulty of the image-based reconstruction namely, the assumption that surface materials, camera position, and illumination are known. In fact, because the connection between the $3D$ geometry and the surface shading, it is quite impossible to correctly estimate one without having a prior knowledge of the other. As result the problem is generally ill-posed. Hence, if no further assumptions were taken, no single image-based reconstruction approach can legitimately infer the three-dimensional shapes from photographs alone. But, if a set of reasonable assumptions are taken, all state-of-art algorithms can reconstruct geometry even from a large number of photographs.

In the literature, and as we saw in previous chapter at Sect. 2.1.1 many visual cues can be used to recover depth from photographs such as texture, contours, defocus, shading, and stereo correspondence to name a few. The latter two however have been very successful, specially the stereo correspondence as it is considered the most robust cue and used in large number of applications. We emphasis that multi-view stereopsis or stereo for short (MVS) is the term agreed upon by researchers to cluster of methods that use more than two image to recover $3D$ shapes based on stereo correspondence cue [67, 108, 121, 128].

All the MVS methods and techniques described in this thesis, and more specifically in this chapter assume the same input namely a collection of images that captures the object of interest along with their

corresponding camera parameters. This chapter gives a general overview of an MVS pipeline that any approach would normally follow starting solely from input photographs. All things considered, this chapter emphasis on one conclusion which is that any MVS algorithm result in a good output geometry if and only if the quality of the input images and camera parameters is high enough. And with a good reconstruction results a good virtual reality experience will be guaranteed

In the following of this chapter, we describe first the common concepts related to image-based reconstruction. We will give then we will more insight into the first two main stages of MVS pipeline namely camera parameters estimation, and imagery collection. Section 3.3 establishes the notion of photo-consistency as the main signal being optimized by MVS algorithms. In the end, we give a brief survey on other $3D$s reconstruction techniques in computer vision besides multi-view stereo.

3.1 Key concepts of the image-based reconstruction

In this section, we will describe the most important key concepts for any multiple view reconstruction approaches. In fact, no matter what technique is used to recover the geometry from the input photographs, the same pipeline is always followed. This multi stage process is based on some basic vision geometry and can be sketched as follow:

- Collect a large set of photographs for a given scene or object.
- Extract camera parameters for each image.
- Reconstruct the scene geometry using the input photographs along with corresponding camera parameters.

Each step of this general pipeline contain one or more key concept that is used extensively. In the following we will address all of this concept.

Images: Basically this entity is the result of the interaction of $3D$ scene elements, lighting, and camera optics and sensors. An image I of a size equal to w x h is a discrete set of intensity values $I(\mathbf{u}) \in \mathbb{R}^d$ captured by the sensors. Where $\mathbf{u} = (x, y)$ is a pixel coordinate on a $2D$ grid with $x \in [0..w-1]$, $y \in [0..h-1]$. Note that if $d = 1$ than this images is considered to be a grayscale image however, if $d = 3$ than it is a color image.

Images Intensity: The proper name for image intensity is image irradiance. This entity is determined via the amount of energy radiated by the corresponding $3D$ point on the surface of a given scene captured in the image (*see* Chapter 2, in Sect. 2.3). Note that the value of the pixels intensity could contain some

noises due to many factors. Hence, most of computer vision algorithms start with image processing to pre-process the input photographs and change it into a suitable form for further analysis.

Projective Camera: In computer vision, many applications treat the camera as an ideal pinhole lens that simply project all the incoming rays through a common center of projection. This model is presented mathematically via $3D$ perspective transform matrix $\mathbf{H}$ with size of $3x4$. Such matrix convert a $3D$ point $\mathbf{x}$ to a $2D$ coordinate $\mathbf{u}$. More descriptions are found in [37].

$$\mathbf{u} = \mathbf{H} \cdot \mathbf{x} \tag{3.1}$$

Depth Map: Each $3D$ point $\mathbf{x}$ in the scene surface correspond to a pixel $\mathbf{u}$ on the image plane if it is visible and not occluded. We call the distance between the camera center of projection $\mathbf{c}$ and the point $\mathbf{x}$ in the viewing direction of that camera, by pixel depth. Hence, the set of depth values for all the image pixels is called depth map. This representation is one of the most popular description of $3D$ surface due to its flexibility and scalability. We like to inform the reader that the main goal of Chapter 5 is to reconstruct a dense depth maps for each input view.

Point Cloud / Patch: Sampling a continuous surface will result in $3D$ positions $\mathbf{x}_i$, optionally with associated normal vectors $\mathbf{n}8_i$ and other auxiliary surface properties. All this element together are called point cloud. Moreover, depending on the algorithm that created this set of points, the end result may contain some points that are far from the the true surface (called outliers). If we give the points a rectangle dimensions and fixed normal vector, we call than a patch.

Triangular Mesh: Is the final form of geometry used in computer graphics and multimedia industry. In practice, a $3D$ triangular mesh is defined as a list of points called vertices $V = (\mathbf{x}_1, ..., \mathbf{x}_V)$ and set of triangles called faces $F = (\mathbf{f}_1, ..., \mathbf{f}_F)$ that describe the $3D$ shape of an object. Moreover, The triangular mesh can be associated to a texture by sampling a specific color to each vertex, or by mapping each face to an $2D$ image.

Volume Gird: Generally a rectangular $3D$ grid which bound a given object, the volume split the space into small entities called **Voxels** which contain some sort of information that could be used to recover the shape. A common convention is to treat the voxels as a (signed) distance function field from a surface. This last can be extracted as a zero iso-surface of the volume.

Silhouette: Binary image that split the scene into two regions the background labelled in black and the object of interest in white, for more on how to obtain this images please refer to Chapter 4 at Sect. 4.2.1. A silhouette image seen from a given view can be defined by the projection (see Equation 3.1) of all the $3D$ points of a given surface on the $2D$ image plane. Let us define O as a set of all possible $3D$ point that belong to the object of interest, and $\mathcal{P}_i(\mathbf{x}, \mathbf{H}_i)$ the projection function of a given point on a given image plane π_i. The silhouette can be described mathematically as follow:

$$S(O,i) = \{\mathbf{u} \in \pi_i : \exists \mathbf{x} \in O \wedge \mathbf{u} = \mathcal{P}_i(\mathbf{x}, \mathbf{H}_i)\} \tag{3.2}$$

Visual Hull: Refer to the shape estimated using Shape-From-Silhouette techniques as a bounding volume that contain the true object O. Let $\mathbf{R}$ be the visualization space in $\mathbb{R}^3$ which is defined by all cameras C_i where $i = 1...N$. Know let $\mathbf{VH}(O, \mathbf{R})$ be the visual hull of the object of interest O relative to the visualization region $\mathbf{R}$. For all points $\mathbf{x} \in \mathbf{VH}(O, \mathbf{R})$, the ray that starts from any camera C_i center and pass through $\mathbf{x}$ must contain at least one point $\mathbf{x}'$ belonging to the true surface of the object O. We like to mention that the next chapter will be based on the visual hull concept.

After given they key concepts in this section that are used during the rest of our dissertations. The next section will dedicated to explain in details the first stage of the reconstruction pipeline.

3.2 Multi-view Environment

In this section the basic principles underlying the photographs used in multi-view environment along with camera models used for passive $3D$ reconstruction, are explained. Consequently, the main goal is to arrive at a $3D$ reconstruction from the uncalibrated image data alone. However, to fathom how is it possible for a three-dimensional object to be reconstructed from two-dimensional images, one first need to know the necessary information about the images used and how these last were formed.

3.2.1 Imagery Collection

Computer vision researchers, in particular image-based reconstruction experts agreed that multi-view stereo data sets can be clustered roughly into three groups (*see* Figure 3.2).

Controlled Settings Imagery

We refer to the first category as laboratory settings dataset, As it is shown by Figure 3.2 the first row, this type of imagery generally represent a single and compact object captured under controlled environment

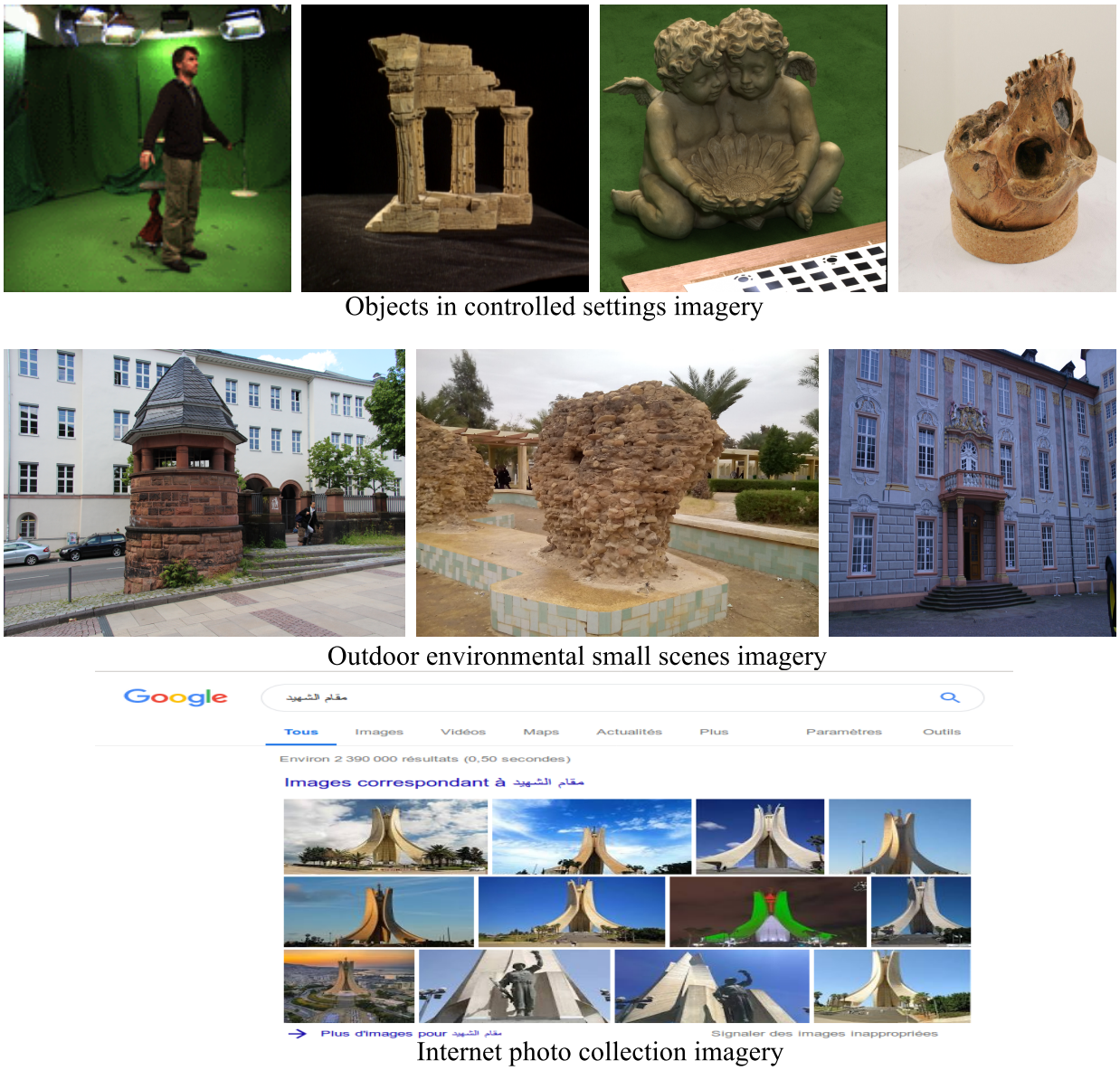


Figure 3.2: Different MVS imagery types. From top to bottom: a controlled MVS capture in Laboratory setup, outdoor imagery of small environmental scenes, and finally a crowd-sourcing imagery from online photo-sharing websites.

using robotic arm, turning table, or even capturing studios. As a result, these objects are fully visible in all the images which were taken from all around the object. Hence, the camera could easily be calibrated. What make such imagery interesting is that it is relatively straightforward to reconstruct probes and small objects that can be integrated directly into a virtual world. In fact, most multi-view stereo algorithms [34, 59, 93] commonly designed to recover geometry from such imagery collection. In this last decade multiple datasets has been proposed along with a trusted evaluation and comparison system which allow researchers to compare their founding with other published work [108].

Small Scene Imagery

Throughout the years multi-view stereo algorithms have evolved gradually. This major developments lead MVS to left the laboratory setting and deal with outdoor small scenes [52, 82, 114, 118]. In contrast to the previous imagery where uncluttered photographs are taken from all around the object of interest, in small scene dataset, occlusion may occur both partially or completely in some of the photographs. Moreover, objects can be embedded in clutter thus limiting the range of viewpoints making it impossible to determine the bounding volume of the scene, which is sometimes necessary for the reconstruction. In addition to all of that we, the lighting conditions in which these images were taken are not controlled, and could change slightly from one image to another. For these reasons, multi-stereo algorithms had to adapt by using new consistency measurement (*see* Sect.3.3) which count for such scenarios, some of the current research start even employing deep learning to MVS [46] as the next step toward better outdoor reconstruction.

Crowd-sourced Imagery

Nowadays, we witness a huge rising of Internet photo sharing websites, people are posting all sorts of images on *Facebook*, *Pinterest*, and even Google image. Crowd-sourced imagery became an interesting and powerful source of image datasets. For instance, a quick search for ”Khalifa Tower” on *Flickr* which is another website for image collection, yields more than 12,000 photographs illustrating the biggest tower in the world, from different viewpoints and appearance conditions. In the context of virtual reality and virtual world building, such type of imagery presents a singular opportunity to create a digital copy of the world’s geometry based on the largest and most diverse multi-view stereo dataset ever assembled for the last decades.

In fact, what makes this dataset unique do not only lie in its size, but the fact that it has been captured in uncontrolled environments under varying conditions. Thus, a new set of fundamental challenges in multi-view stereo research appears. In particular, when dealing with such type of imagery it is very important to account for various factors, like the tremendous variation in resolution and illumination, unstructured capturing geometry, and scene variability. One of the rare research that was done on such type of dataset is the work of Goesele et al.[36] in which they propose a method that explicitly targets crowd-sourced images, the authors reconstruct a depth map for each input view after selecting a camera clusters for it. Johannes et al. [107] present a multi-View Stereo system for robust and efficient dense modelling from unstructured image collections.

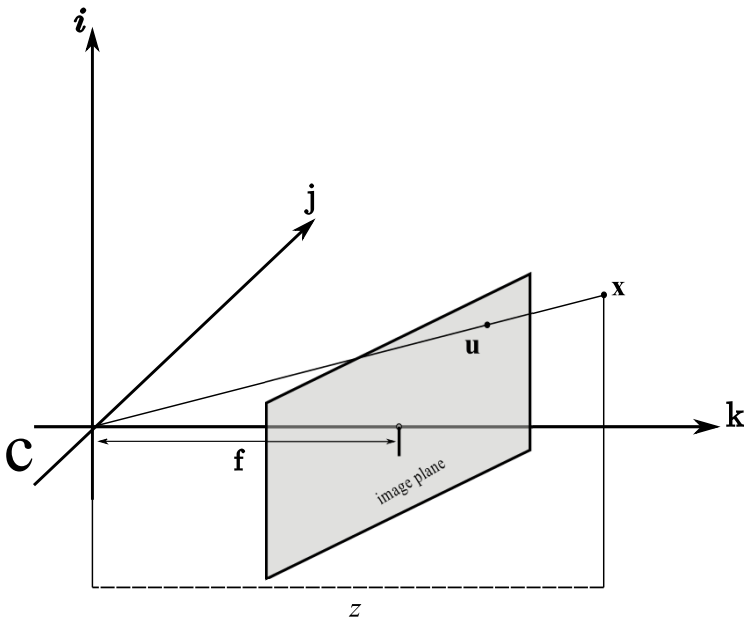


Figure 3.3: A point $\mathbf{x} = (X, Y, Z)$ is projected onto the image plane by the ray passing through the center of projection, and the resulting point on the image is $\mathbf{u} = (i, j, f)$;

3.2.2 Camera Models and Calibration

In this part of the chapter, we will discuses, in details the fundamental concept related to image-based reconstruction, namely the camera model used in the process and how this model is estimated from the input photographs.

Similar to the human eyes, where image is formed on the retina. Computer Vision in all the digital cameras in general replicate this by modeling the process of image formation, in particular, mapping $3D$ geometry to $2D$ image pixels using the so called pinhole camera model. As we discussed earlier in the beginning of this chapter, MVS methods require additional information in order for the reconstruction problem to be solvable. In detail, these algorithms require that every input view has to be jointed with its corresponding camera model with which it can express the projection process. Hence, pinhole camera model is the most commonly used camera model for MVS algorithms.

The geometry of the pinhole camera can fully explained by central projection [37], The image point $\mathbf{u}$ of a point $\mathbf{x}$ on the $3D$ surface is obtained by tracing a ray from the center of projection $\mathbf{c}$ to point $\mathbf{x}$ through the image plane. Figure 3.3 shows the process as $2D$ illustration, In the figure, we can see by similar triangles that $\mathbf{u}/\mathbf{f} = \mathbf{x}/\mathbf{z}$ hence, the point is projected as follow :

$$\mathbf{u} = \mathbf{f} . \frac{\mathbf{x}}{\mathbf{z}} \tag{3.3}$$

In an other world, equation Eq.3.3 means that the image is generated by intersecting the light rays with the image plane at a distance **f** from the center of projection. This projection can be described conveniently by matrix multiplication as we showed in Sect.3.1. In fact, the general projection **H** can be decomposed into the product of a 3×3 upper triangular matrix **K** and a pose matrix $[R|t]$ at the size 3×4.

$$\mathbf{H} = \underbrace{\begin{pmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \cdot \underbrace{\left(\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{array}\right)}_{\mathbf{R}|\mathbf{t}} \tag{3.4}$$

In details, the pose matrix $[R|t]$ is called extrinsics matrix where **R** represents a 3x3 rotation matrix which indicate the view direction of the used camera while **t** is the translation vector of the camera that is computed from the known camera center as $\mathbf{t} = -\mathbf{R} * \mathbf{c}$. To emphasis, The extrinsic parameters perform a transformation into the camera coordinate system. Note that the camera is looking along the positive z-axis, the x-axis goes to the left and the y-axis goes upwards.

On the other hand, the 3x3 matrix **K** is known as the calibration matrix which corresponds to the intrinsic of the camera. The role of **K** is to transform the point from camera coordinate to the image coordinate frame. As equation Eq.3.4 shows, the calibration matrix has five parameters. First, the vertical and horizontal focal lengths with the respect to x and y of image coordinate frame are described by (f_x, f_y) respectively. The third parenteral s represent the skew, in practice pixels are assumed to have no skew and have square shape. Finally, p_x and p_y define the principal point of the projection, also known as principal point. Using all this parameters we can describe the mapping of any surface point to an image pixel via a pinhole camera model.

A multitude of different methods and technique were used in the literature to estimate the above camera parameters. One of the most powerful technique is called Structure-From-Motions (SFM) and bundle adjustment [11, 14, 106]. The state-of-art for this technique is very vast and it is not in our intention to fully go through it and review it in this dissertation. However, we will discuss in this section some the key aspects of SfM.

Structure-From-Motion: We assume that we are given two view of a $3D$ scene, and observe the same point in the these two views thus having a pair of $2D$ points. Suppose now that we have such pairs of corresponding points in two images for a set number of $3D$ points on the observed scene. As a consequence,

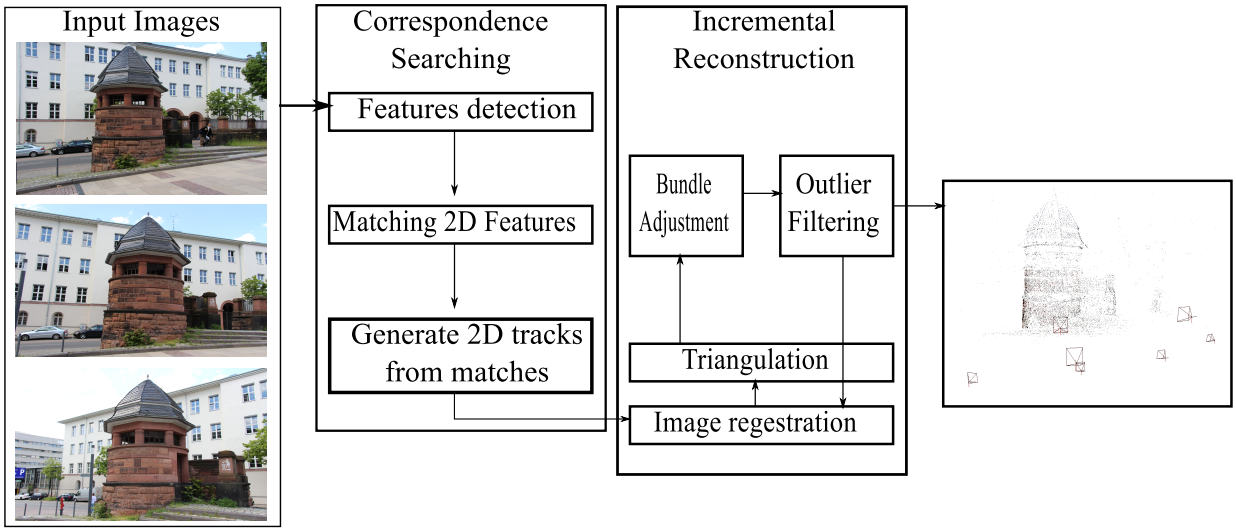


Figure 3.4: Main phases of a general SFM pipeline, following the arrows, the process start by feature detection and feature matching than track generation which lead to structure from-motion finally bundle adjustment and the result is point cloud and camera parameters.

we have two challenges that we are trying to resolve. One challenge is to estimate how did the camera move from one view to the next, that is the rotation and translation $[R|t]$ parameters, also known as the extrinsics matrix. The second thing we want to estimate is the three-dimensional structure along with camera intrinsic. To summarize, this algorithm take set of images as input, and produce the camera parameters for every image along with a cloud of $3D$ points which are visible in the images , this points are generally called tracks, in practices these tracks are coupled with a list of corresponding $2D$ points in a subset of the input images. Most of the current SFM algorithms have the same basics and follow almost the same processing pipeline (*see* Figure 3.4).

First for each input image, the algorithm starts looking for unique places in the image that can be unambiguously recognized if it is seen from another image. These small image regions are called features which as we said should be invariant under radiometric and geometric changes most popular feature descriptors is the scale-invariant features SIFT [73]. The next step in the process is to match the features between all possible image pairs using the feature descriptors. In another word, the algorithm make sure that the obtained feature in one image are approximately the same in other images. A variety of approaches tackle the problem of scalable and efficient matching are used such as [41, 111]. At this stage SFM algorithm checks the matches by estimating the geometrical transformation that maps feature points between images using projective geometry, in order to find this transformations a robust estimation techniques, such as RANSAC [20, 66] were used, which estimate the epipolar geometry between two or three views given a sample of noisy matches.

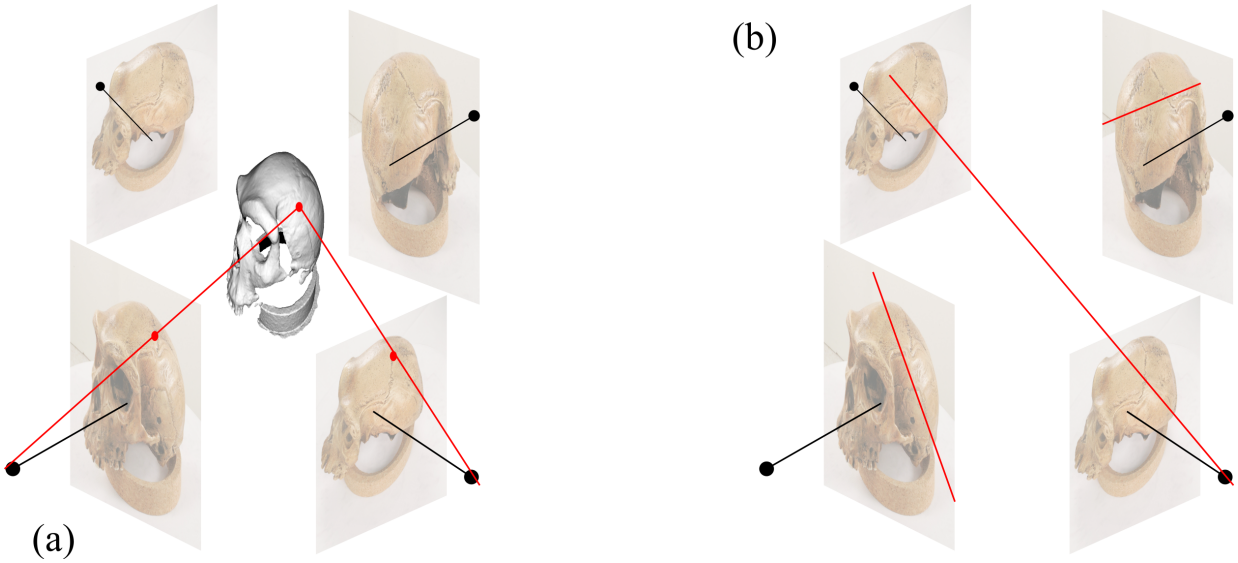


Figure 3.5: Multi-view stereo matching problem in nutshell. (a): The $3D$ shape of the scene gives the correspondence between pixels in different photographs. (b): If camera parameters are known, matching a pixel in one image with pixels in another image is converted to searching problem on $2D$ line.

Image Registration, comes after to resolve the $2D$ tracks and extract the true camera parameters by solving the Perspective-n-Point (PnP) problem, various solvers can be used such as [6]. Another important step of the SFM algorithm is Triangulation as it allows the registration of new images by providing new correspondences. Finally, uncertainties in the camera pose extend to triangulated points and vice versa. Hence, a refinement step must be included to enhance the accuracy. In fact, bundle adjustment is a common step used to refine the initial SFM $3D$ tracks hence more accurate cameras pose. Despite all the analysis above which toutch briefly on decades of work and development, creating a complete and reliable calibration pipeline remains a non-easy procedure which requires much of know-how, with multiple pitfalls and sources of errors.

3.2.3 Multi-View Stereo

The multi-view stereo is an old concept and its origins can be traced back to the earliest attempts of solving the stereoscopic matching problem [3, 75]. To this days, two-view stereo algorithms or disparity map computation techniques are still active and it is considered as promising research area. Despite these success stories, The multi-view stereo was proposed as a natural improvement to the two-view case. As the name suggest, MVS uses more than two photographs to increase robustness to all sorts of image noise [88].As we mentioned earlier in this chapter, the camera parameters should be known in order for the MVS problem to be solved. In fact, estimating the $3D$ geometry of a given scene is basically the same as solving the correspondence problem across the input images. For instance, let's take a $3D$ point which is situated

on a given rigid surface, as it is shown in Figure 3.5 (a) projecting this point into the image plane of the views creates a unique correspondence. Know suppose that we do not have an initial geometry, while also the camera parameters are known, the matching problem is simplified from searching the image pixel by pixel to search for match along the epipolar line on the other images (*see* Figure 3.5 (b)).

To summarize, in order to find true correspondence of any pixel in other images, the algorithm has to use the epipolar geometry to generate possible pixel candidates in all other images. And to confirm the matches a function that measures how likely a given candidate is acceptable. This last is called Photo-consistency measures which will be discussed in the context of MVS with more detail in the next section.

3.3 Photometric Consistency in Multi-view Environment

This section develops the heart of multi-view stereo reconstruction. Multi-view photometric consistency is considered the main concept used in all MVS algorithms. This concept is defined as a function that measures the amount of agreement or consistency between all the input photographs according to ingredients that take part in their image formation namely the camera parameters, illumination and the nature of the surface geometry of the scene being photographed. If all the conditions happened to be appropriate, state-of-art MVS methods can basically recover the $3D$ shape along with materials, and illumination from the input photographs. Hence this type of image-based reconstruction algorithms are considered as a constrained optimization problem. Where the photo-consistency measurement is maximized/minimized as a function of all the ingredients that take part in the image formation for that specific scene.

In this section we introduce the general concept of measurement, later, we discuss a crucial requirement for photo-consistency measurement namely visibility Sect. 3.3.2. Finally we provide, the most common consistency measurement used in the literature Sect. 3.3.3

3.3.1 Photo-consistency

Most photo-consistency measures are established to compute the matching cost of pixels between two or more images. This measurement is defined as follows :

Definition 5. Given a set of N photographs which capture the same rigid surface geometry. If a $3D$ point $\mathbf{x}$ is visible in each pair of images I_i and I_j of this set, The assumption used to define such a function is that the color of a pixel in the different images should be the same or close. we can define the photo-consistency of $\mathbf{x}$ using the following equation:

$$C_{ij}(\mathbf{x}) = \kappa(I_i(\Omega(\mathcal{P}_i(\mathbf{x}, \mathbf{H}_i))), I_j(\Omega(\mathcal{P}_j(\mathbf{x}, \mathbf{H}_j)))) \tag{3.5}$$

The function can be broken down into $\kappa(m, v)$ a similarity measure that compares two vectors $\Omega(.)$ (also known as the a support domain around point the projection of **x**) from the master image M and the corresponding support view V. each element of domain consists of intensity value. To summarize, every consistency measurement is described as a particular choice of κ and $\Omega(.)$.

In literature, some MVS methods compute photo-consistency on a level of one pixel. Hence, the usage of the support domain is negated. However, in most cases the small area surfaces of scene does not project on only one pixel. Thus, the support domain Ω defines a block of pixel where the appearance is expected to be unique and invariant. Such property however should be carefully chosen. In fact, if the size of support domain was selected to be very large, then the local appearance are guaranteed to be unique thus it is more easier to find matches in other images. On the other hand invariance is lost specially when there is varying capture conditions. For most of MVS algorithms the easiest way to define the domain is to use square of pixel with constant size across the process of matching. Generally, the size of $\Omega(.)$ relatively very small in comparison to the total image size, for example a 3x3 or 5x5 grid of image intensities was used in [28, 34]

Given the photo-consistency equation Eq. 3.5 between only two images, we can extrapolate for multiple views by simply averaging the similarity measurement $C_i(\mathbf{x})$ between the reference view and all its N support images. In the next part, we will talk about the visibility and how MVS take it into account when evaluating the photometric consistency across the images.

3.3.2 Evaluating Visibility in literature

To measure consistency of a point in the scene, it is important that this point has to be visible on the set of views used during the matching process. However, the visibility of any surface point can change dramatically from view to another. It is important for the MVS to account for occlusion, but there is not a unique and global visibility technique that could account for all scenarios.

In general, MVS algorithm start the process with out knowing which image see what surface, since the 3D geometry itself is unknown (*see*, Figure 3.6). This lead to a paradox effect where, in order to reconstruct three-dimensional geometry via photo consistency optimization, one needs to recover the correct visibility

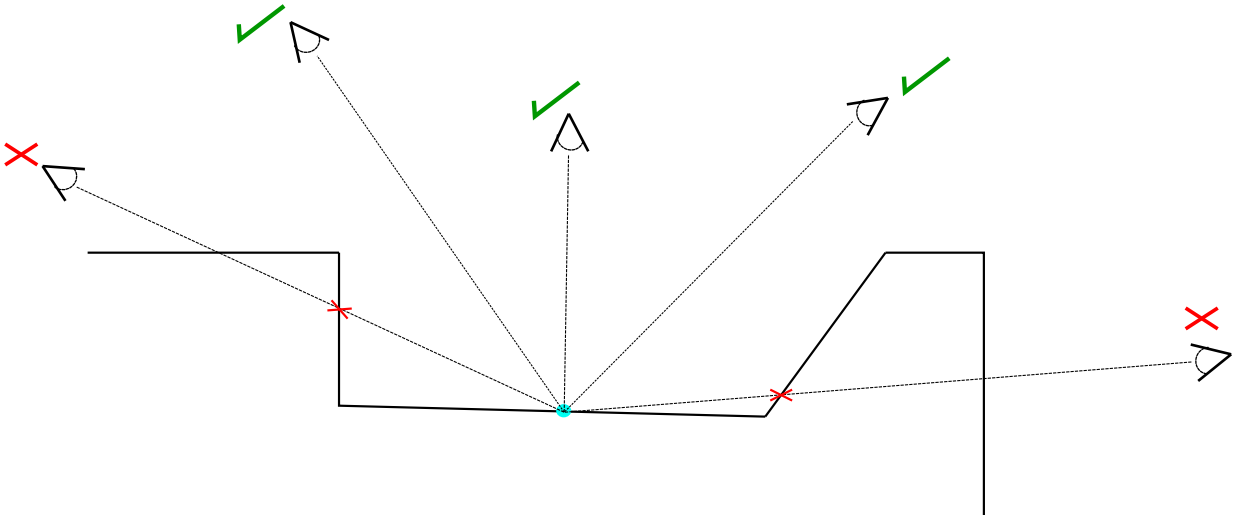


Figure 3.6: Visibility problem, in order to estimate geometry using photo-consistency occlusion should not appears, in the same time occlusion is detected only using geometry

information. This last it self needs the $3D$ geometry to be estimated. In the following we will touch briefly on common approaches to break this loop.

Fortunately, this does not mean that the visibility is an intractable problem, one of the earliest work that dealt with the visibility problem is the space carving approach [57, 109]. Given a $3D$ bounding volume divided into a grid of small voxels, the algorithm iteratively carve the volume and remove the voxels that are not photo-consistent. The key idea proposed by the authors of the space carving algorithm was imposing the camera position such that all the voxels of the volume has a systematic visibility constraint. and it can use some sort of depth order which guarantees that any potential occluded surface point is visited after the occlusion source. Such proposal adequately break the visibility paradox loop as it provides a voxel visiting order, hence when this last is tested and found to occluded in a given image the algorithm will automatically not use this image. However, the algorithm breaks when $3D$ point of the scene is located inside the convex hull of the camera centers.

In fact, such a constrain is limiting especially in modern State-of-the-art MVS methods which are designed to handle millions of images. Thus some of the multi-view stereo approaches estimate nearly ignore the visibility problem by estimating coarsely the occlusion informations [28, 32, 108, 112, 120, 137]. Most of this approaches divided the input set of images to small clusters making the large-scale MVS problem a sequence of small sub-problems. Note that in our work, we followed the same process for more details refer to Chapter 5. In fact, each of this small sub-problems is formulated as as a narrow-baseline stereo problem. Where for each depth map reconstruction, the reference image uses from 5 to 10 support images,

these last share almost the same view direction with small difference of 10 degree. Another idea was used by Furukawa et al. [26] uses the result point cloud of SFM algorithm as proxy to compute the parallax and formulate the visibility problem as constrain optimization search.

On the other side, other MVS methods prefer to enhance the reconstruction by revisiting the visibility after achieving an initial reconstruction or by iteratively updating depth maps and visibility information in some sort of a Bayesian framework [125]. In fact, this type of fine-scale visibility estimation needs a complete $3D$ model such as the visual hull to handle occlusions properly as it is illustrated in Figure 3.6. In fact, this kind of techniques used to determine visibility select which views see what parts of the scene, and iterate visibility estimation along with the reconstruction, this lead to more accurate reconstruction since the geometry is being refined iteratively. The only drawback for such technique is the quality of the initial geometry. In addition to these proposed solution in the literatures, others ignore all together the visibility problem and put their confidence on the photo-consistency measures. The logic behind this is wrong view will automatically yields a poor photo-consistency.

3.3.3 Consistency measurement tools

Our goal in this section is not to provide a full and extensive review or an evaluation of all the consistency measurement used in the literature, we refer the reader to this article [44] for more detailed informations on stereo correspondence methods. However, we introduce the different similarity measures used to compute photo-consistency and define each one.

Normalized Cross Correlation: Zero-mean normalized cross correlation (NCC) is the most commonly used and successful photo-consistency measurement in the multi-view stereo algorithms. The success of this measures can be associated with its invariant to changes in gain and bias. In fact, this photo-consistency is mainly used when lighting and material variance is presented in the photographs. However, the NCC can fail greatly in untextured surfaces, or on surface with repetitive pattern. This similarity measurement is defined mathematically as:

$$\kappa_{NCC}(m,v) = \frac{(m-m')\cdot(v-v')}{\sigma_m \cdot \sigma_v} \in [-1,1] \tag{3.6}$$

Where as we mentioned earlier, m and v are two intensity vectors defined on the support domain Ω in the reference image M and the support view V. Meanwhile, m' and v'. represent the mean value of m and v respectively. the values σ_m is the standard deviation of m, the same thing goes for other vector.

Sum of Squared Differences: The sum of squared differences (SSD) is the most simple and probably the earliest measurement function used in stereo matching. It is a pixel wise function that does not require any support vector Ω which make very sensitive to outliers. Generally such measurement is used when the baseline of the two view is very small. The SSD measure is defined as the L^2 squared distance between vectors:

$$\kappa_{SSD}(m,v) = ||m-v||^2 \tag{3.7}$$

Sum of Absolute Differences: The sum of absolute differences (SAD) is enhanced version of the SSD. This measurement uses the L^1 form defined on a given support domain Ω. such formulation make it very robust to outliers. However, it is very sensitive to bias and gain which are present in images with wide variability in illumination.

$$\kappa_{SAD}(m,v) = ||m-v||_1 \tag{3.8}$$

Census: One of the powerful matching costs function. Similarly to NCC, Census is insusceptible to changes in gain and bias, moreover, it requires a support domain to be computed. This measure differ however from previously mentioned photo-consistency function in the values used to compute the score, In fact, census uses a comparison operator (*see* Eq. 3.9) which compare the intensity values in the Ω domain to its center **u** resulting in a bit string (*see* Eq. 3.10)

$$\Delta(\mathbf{u},\mathbf{a}) = \begin{cases} 1 & \text{if } \mathbf{u} < \mathbf{a} \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

The bit string which tell us whether a pixel in the support domain is brighter or darker than the center of the domain in given view is computed simply by:

$$\mathit{census}(v) = \oplus_{a\in\Omega}\Delta(v(\mathbf{u}), v(\mathbf{a})) \tag{3.10}$$

The notation $\oplus$ simply represent the concatenation, Finally the consistency score is computed as a hamming distance between two strings which measures the minimum number of substitutions required to change one string into the other.

$$\kappa_{census} = \sqcap(\mathit{census}(m) \veebar \mathit{census}(v)) \in [0, N] \tag{3.11}$$

where $\sqcap(.)$ is a function that counts the number of 1 in a binary string, and N is the size of the string. Census preforms better is depth boundaries which is a problem for all the consistency functions that uses

a support domain. especially if the size is quite small. Finally this section presented a couple of the most used photo-consistency in order for the reader to be equipped by the necessary knowledge for the rest of this dissertation. In the next section we try to review the main methods closely related to this thesis.

3.4 State-of-art Algorithms

A large amount of literature on multi-view stereo reconstruction has accumulated during these 35 years. Evidence for in support of this statement, can be found in the new methods proposed and published every year in multiple international conferences and journals. In fact, a wide verity of properties and assumptions that differentiate each proposed method, which make classifying all the multi-view stereo and compiling a detailed survey a very hard and formidable task. In this section, we review the main methods closely related to this dissertation. Therefore, we refer interested readers to many excellent books of computer vision like [37, 92, 123].

3.4.1 Visual Hull Reconstruction

Recovering the $3D$ shape of a dynamic objects in a give scene at real-time is considered as the pillar of Tele-presence systems. According to this, the three-dimensions reconstruction techniques used in such systems must be able to resolve the exact geometrical form of any moving object several times per second. The *Shape From Silhouette* techniques are extensively used in such delicate scenarios. In fact, the *Shape From Silhouette* methods compute the visual hull of the object of interest via multiple silhouette images. The concept of visual hull appeared initially in 1994 by Laurentini [62], the latter presented a theoretical model which can be estimated and reconstructed from the intersection of an infinite number of silhouettes.

Such idea caught the researchers attention and multiple algorithms were proposed in the literature. In general, these algorithms can be grouped into two axes according to the data representation of the output model type. Surface-Based methods explicitly estimate the visual hull surface by directly intersecting the silhouette cones. On the other hand Voxel-Based methods, basically compute the maximal volume of the visual hull that is silhouette consistent with the captured object.

Surface-Based methods

This kind of methods focus on estimating an explicit representation of the visual hull surface, In fact, polyhedral methods usually start by approximating each silhouette with polygonal representation. The approach then re-project the edges of this polygon to the $3D$ space using the inverse camera projection

matrix resulting faceted conical shapes. The visual hull is then deduced via the intersection of those shapes as a polyhedral geometric form [4, 76]. Franco and Boyer [21] proposed *The Exact Polyhedral Visual Hull* (EPVH), which is considered as the state of the art of surface-based methods.

In contrast to previous works EPVH gives an exact description of the polyhedral visual hull, furthermore the method is fast and allows a real time recovery of both watertight and manifold polyhedral representation. An enhanced version of the algorithm came after that [22], where it addresses the problem of establishing a good topological proprieties of the polyhedral surface. Recently, Duckworth et al. [16] proposed a commodity tele-presence system based on a distributed version of EPVH, the authors changed how the problem was structured to reflect the different stages of the process rather than the need to minimize the data communication on the network.

Volume-Based Methods

In contrast to surface-based *Shape-From-Silhouette* methods, volume-based approaches estimate the volume of the visual envelope by a set of elemental primitives where the 3D space is divided into small regions called voxels. *Space Carving* [57, 109] methods for instance describe the $3D$ space around the object of interest by a grid of voxels, Each one of this element has to be processed and tested before it can be included in the final model, note that this step is critical in terms of computing time.

On the other hand the volume size makes those approaches suffer from memory problems and long execution time, numerous solution were proposed, usually the volume is stored as octrees [99] which speed up the process and make the model fit into less space [85, 94, 122]. The use of modern hardware GPU [39] offered a dramatic boost to volumetric visual hull computation.

3.4.2 Depth Maps Reconstruction

3D reconstruction via MVS has been around for decades. Eventually, several high-quality algorithms have been developed which led to an exponential growth in literature. Unfortunately, due to the wide variety of properties and assumptions that differentiate each method, it is quite challenging to propose a general taxonomy and put each method into a specific class. However, M. Seitz et al. [108] proposed multiple main proprieties which help to categorize these reconstruction methods. According to the authors, multi-view stereo approaches can be categorized based on the output scene representation; for instance via point cloud, voxels, a mesh or depth maps. The later is a popular choice due to its simplicity and scalability. Methods use this representation are also known as multi-view depth map estimation [130].

Multi-view depth map estimation can be further divided in term of the stereo algorithm used into two categories, *global variational approaches* and *local Winer-Take-All approaches*. In general, these algorithms, like the binocular stereo [103] evaluate the photo-consistency of the computed depth based on more sophisticated similarity metrics such as sum-of-squared-differences (SSD) or normalized cross-correlation (NCC). However due to the presence of non-Lambertian photometric effects, in addition to the different illumination changes and the lack of visibility information in some cases, it is not always guaranteed to find unique matches. As a result, it is crucial to design a robust cost measurement with high matching quality. Mei et al. [79] propose a robust matching metric, The main idea is to couple the Absolute Differences (AD) along with Census as initial matching cost volume.

While the absolute difference is sensitive to bias and gain and tends to give a good score under similar capture conditions or textureless region; The census is invariant to changes and can handle wide variability in illumination; hence combining the two consistency metric must improve the matching accuracy. Zhan et al. [134] go a step further, and they took into consideration the computational complexity. The proposed similarity measurement is a combination of the absolute differences, the census transform, and the double-RGB gradient model.

Local Winer-Take-All Approaches

Early Multi-view stereo models were local, meaning that the depth computed at a given pixel depends only on a local neighborhood. Local methods emphasis on computing the cost function and on the cost aggregation steps. Competing approaches range from simple window matching [34] to more sophisticated approaches that deal with the fundamental problem of the local method, such as the selection of support window [112]. Computing the final depth is trivia; simply perform a WTL strategy at each pixel.

The common patch-based approach such as [28, 36] reconstructs the depth maps by taking sparse feature points from SfM and grow iteratively to a full surface. The consistency of patch in a subset of neighbor images is defined as a function of its position and normal, thus by merely maximizing the photo-consistency function with respect to those parameters, we can reconstruct the patch. Occlusion in such approach was handled in different ways, for example, Goesele et al. [36] relay heavily on per-pixel view selection (Local view selection) and NCC to detect occlusion, this is because occluded pixels in some neighboring image may be visible in others. In aiming to handle occlusion Zhu et al. [137] further proposed a five steps methodology, in which depth, surface normals, and visibility are estimated for each image pixel instead of trying to utilize all neighbor images blindly without any knowledge of visibility, preprocessing steps are

added to extract minimum visibility and reject outliers.

Chen et al. [131] proposed a stochastic optimization, and replaced the derivative-based optimization step presented in Furukawa et al.[28]; their method relies on forcing a proper constraint on ranges of the patch parameters such as depth and normal, the results proved to be effective yet limited in terms of expanding into larger textureless regions. In contrast, our work introduces shading cues to the objective function which provide additional information and recover more details. Wu et al. [132] were considered the first to present an approach that uses pre-estimate lighting (spherical harmonics approximation) and a shading model to refine the stereo reconstruction. Their method was limited to single constant albedo.

Global Variational Approaches

The use of variational methods for multiple view 3D reconstruction is nothing new [19, 95], and most of the strongest works in the literature [69] follow that route. Such methods typically cast the reconstruction problem as one of searching for the optimal surface that minimizes a photometric energy functional defined over all pixels. This formulation combines a data fidelity criterion on the surface function with desired assumptions like smoothness.

Recently, a new variational approach was built from the ground up and proposed inside a multi-view framework by Semerjian [31]. This approach optimizes over a bicubic patch defined on a square grid using Gauss-Newton descent and the finite element method, and generate a surface per view that has continuous depth and normals. The author uses a new edge preserving smoothness term; in fact, it is based on the first derivatives of the normal against image coordinates. This is effective in producing curvatures, however, it cannot recover thin details in regions without strong gradients.

Langguth et al. [61] propose a variational MVS approach that recovers a smooth surface and can relate small gradients to surface details. The authors build on the previously mentioned work of Semerjian. However, the focus was on presenting an albedo-free and implicitly regularized optimization based on the *Retinax theory*, which says that small gradient is caused only by lighting hence, it is possible to separate surface albedo from shading. Accordingly, the reconstruction approach combines the geometric stereo error and the photometric shape-from-shading (sfs) error into one global optimization schema adequately. The approach has some limitations, the most important one as the authors motioned is the simple lighting model; a Lambertian model assumes that reflectance is constant over the hemisphere domain, which physically incorrect. This assumption led to weak and inadequate approximation thus it is impossible to

account for self-shadowing, indirect illumination, and specular reflection.

3.5 Conclusion

We have made a short survey in multi-view stereo literature. starting by introducing the common key concept used in computer vision mainly in the stereo reconstruction. After introducing the main knowledge, we moved to explain in details the multi-view environment, where we saw that, multiple types of imagery can be used to recover $3D$ geometry. However it is necessary to ensure good and accurate camera parameters via shape-from-structure algorithm. Later, we explain the core key of any multi-view stereo algorithm which is the photo-consistency function, we gave a detailed description of this measurement and presented some common example. Finally a state-of-art algorithms that are close to our contribution is presented.

In the next chapter we start by discussing the first contribution of this work which is related to interactivity and real time reconstruction. the proposed implementation showed impressive results which are shown later on.

They always say time changes things, but you actually have to change them yourself.

Andy Warhol

4

Interactive Multi-view Stereo Rendering

Most computer vision researchers focus their attention on 3D reconstruction methods, in particular, techniques that can infer a highly detailed 3D geometry from a collection of images alone. It is considered a hot research topic due to the technological evolution of digital camera, where the latter became a cheap and reliable scanner. However, such an inverse problem is generally not well posed since multiple shapes, surface materials, camera poses, and illumination can produce exactly the same input photographs. Moreover, the multimedia industry nowadays requires dynamic scene modeling for most of its applications, thus the quality of output of a 3D reconstruction technique is proportional to the time taken to compute it. However, under a set of reasonable assumptions and constraints it is possible to recover state-of-art geometry efficiently [31, 126].

In fact, it started around the 70's where Baugmat [4] tried to reconstruct a simple object from four different views, in the 90's Laurentini [62] introduced the concept of the visual hull (VH) and its features, since then a wide range of methods to reconstruct the scene from multiple images appeared. During this same time, and in the last several years, image-based rendering has become one of the most interesting application in computer vision. In fact, this technique depends heavily on real photographs and benefits from the fine detail and complex lighting effects existing in those input images, and hence is capable of rendering photo-realistic virtual views. Researchers were excited because there is a possibility that this new technique will allow virtual travelling to the world's most interesting places [117].

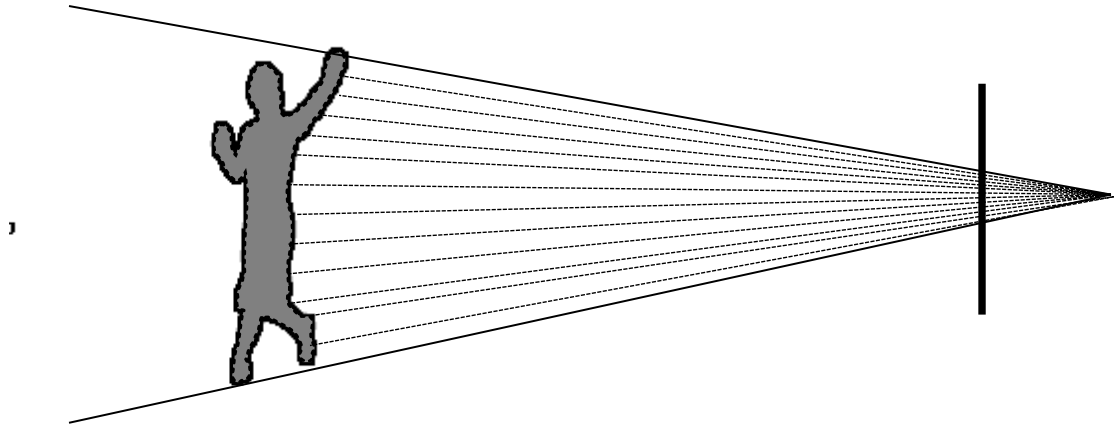


Figure 4.1: A single slice of an image-based visual hull which is a 2D representation of the full image-based visual hull.

4.1 The Proposed Visual Hull Reconstruction System

The basic idea behind our implementation is that the image-based visual hull [78] reconstructs and renders the objects on the image space. In fact, estimating the visual hull using the image-based representation is much simpler, where each pixel computes a small portion of the object surface individually. In other words, color only what is visible in front of the novel view as illustrated in Figure 4.1, where we present a two-dimensional slice of an image-based visual hull. The dotted lines represent viewing rays along one column of the image, each line is emitted from the center of the projection through a given pixel. Such a representation is more appropriate for real-time parallel implementation of the visual hull reconstruction using modern GPU's. In contrast to the classical volumetric approaches [39], image-based visual hull representation can be implemented on the current graphical hardware without using the projective-texture-mapping technique.

In particular, for real-time application the amount of volumetric data would limit the number of threads that the hardware uses to compute a dense volume of the visual hull. Ladikos et a.l.[58] use more advanced data structure to enhance the visual hull computation. The system has 16 cameras attached to a cluster of four machines where each computer has a GPU that is used to estimate a portion of the visual hull. Despite this, the algorithm was limited to small volume size equal to 128^3 unit. Moreover, the algorithm implementation is complicated and cost effective due to the use of multiple machines.

Furthermore, it is clear that the original image-based visual hull (IBVH) [78] reject the use of the volumetric data representation in order to avoid the resolution problems. In fact, this algorithm use the image-based strategy which alleviates some of the problems presented in the standard voxel approach. Then again, volumetric data embodiment is well suited for hardware implementation. Therefore, we think it possible to fuse image-based representation with the volumetric data structure which can be manipulated easily on the graphical hardware in order to enhance the performance of the original algorithm. While emphasising on real-time and accurate reconstruction, we presents an efficient GPU-accelerated methods to implicitly render a high quality novel view. In this chapter, we describe a new implementation for image-based interactive virtual rephotography on a single machine. Our objective is to propose a navigation system while fully exploiting the details presented in the original photographs, also we aim to achieve a high frame-rate while avoiding any polygonal representation. Our visualization system has the advantage of running entirely on GPU.

To this end, a detailed description of the proposed parallel strategy is provided in Section 4.2 and the implementation of the necessary auxiliary functions too this strategy are described in Sect. 4.3 Then, we give the complexity of our parallel algorithm under some proved assumptions while analysing the results on Sect. 4.4.

4.2 Parallel Strategy For Image-based Visual Hull

The parallel strategy proposed in our work is based on data dependencies. Such dependencies emerge where algorithms cannot be considered as a single data independent for each computing unit from starting point of the execution until the end. Figure 4.2 shows the parallelisation of the image-based visual hull algorithm according to our strategy. It can be seen from the figure that the framework consists of three phases, where the next stage of the algorithm depends upon the previous stage being completed. However, the parallelisation is achieved for each stage due to the nature of the image-based data representation. This last allow to execute independent data instruction per pixel, hence computing the novel view of the visual hull in real-time.

During this work however, we suppose that the objects of interest are positioned inside a bounding box and visible from all cameras. This volume is splitted into a regular grid of voxels with a given resolution, each voxel contains physical position of a point in the three-dimensional space. Note that the background and lighting are static.

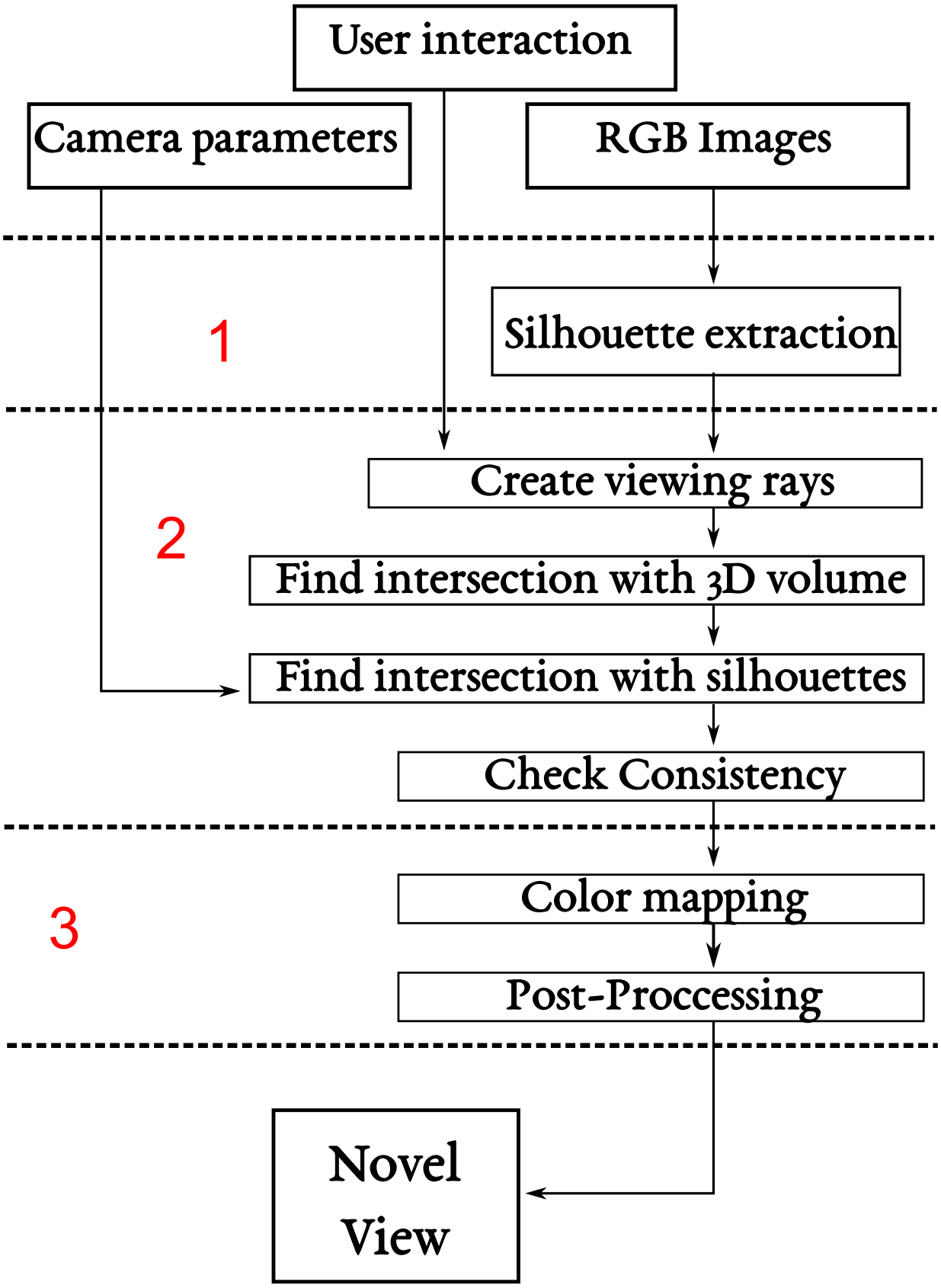


Figure 4.2: System flowchart defining inputs, outputs and processes in reconstructing the visual hull using the modified images-based algorithm. Regions labelled 1–3 are data independent parallel regions .

4.2.1 Image processing

Let us first recall that our goal is to reconstruct one or multiple objects which are part of a complete scene. Hence, it is necessary to detect and distinguish these objects in the input images. In other word, for each input image we want to pin down pixels that are correlated to objects of interest from the rest of the scene. Such a process results in segmenting the image into two classes namely silhouette and background. Extracting silhouette images is a well known problem in the computer vision.

Among the proposed methods in the literature, we focus on methods that extract the silhouette based on the pixel differences and they are also known as background subtraction techniques [42]. Such an approach can be easily implemented on modern hardware where each pixel is treated individually. Our approach to extract the silhouette for each frame can be sketched as follows:

- Initialise the system by the background images **G**
- Capture the current frame **F** of the ith camera.
- In parallel, compute the RGB difference between **F** and **G**.
- In parallel, apply an empirical threshold to extract the foreground pixels of the current frame.
- In parallel, label the background with a black colour and the foreground objects with a white colour.
- In parallel, apply morphological operation
- Merge the extracted silhouettes from all the cameras into one data block.
- Send the image big data block to the next stage.

This algorithm is implemented to run inside the main rendering loop in case we use multiple video streams. However, it is possible to execute it as pre-processing step for static object visualization (no video streams). The end result of this stage are multiple silhouette images i for an object O at a given frame t is defined by:

$$S(O,i,t) = \begin{cases} 1 & \text{if } pixel(x,y) \in O \\ 0 & \text{if } pixel(x,y) \notin O \end{cases} \tag{4.1}$$

Silhouette consistency: We like to remind the reader about the concept of consistency discussed in the previous chapter. An non-occluded point is said to be photo-consistent if its $2D$ projection share the same color in all cameras. However, in the case of using silhouette images the point is considered photo consistent if its projection is a pixel labelled silhouette for every input view. Hence the point is silhouette consistent.

Captured Image

Novel view in a virtual world

Figure 4.3: Rendering results of our GPU image-based visual hull reconstruction using the Dancer Dataset. One the left the original view while on the right The novel view of the visual hull is rendered by blending the textures from multiple viewpoints. The background scene is rendered as a textured sky box. .

4.2.2 Visual Hull estimation

After extracting the necessary silhouettes from the input RGB images, the the next stage is immediately launched to estimate the visual hull. To this end, all image-based visual hull reconstructions are capable of synthesizing a novel view of the visual hull from silhouette images without an explicit intermediate data representation. Unlike these approaches we used a GPU volume rendering approaches which fuse the image-based representation and volumetric data representation to carve the visual hull and create a novel view in short time window that allows interactions.

The interactivity in our system simply presented as the possibility to navigate the virtual scene (*see* Figure 4.3). Hence, the position and orientation of the virtual camera should be taken into consideration while reconstructing the novel view. Evidently, we adapted the ray-casting technique to reconstruct the visual hull from a given perspective. In fact, each pixel in the desired view of the visual hull, we cast a ray into the $3D$ space independently.

Consequently, every ray will be classified into one of three types according to what it intersects in the Cartesian space. First the null ray presented in Figure 4.4 with solid black line, these are rays that did not intersect the bounding box that surrounds the object of interest. However, if an intersection is detected the ray continue to march the volume and search for any point that is silhouette consistent. Thus, some rays

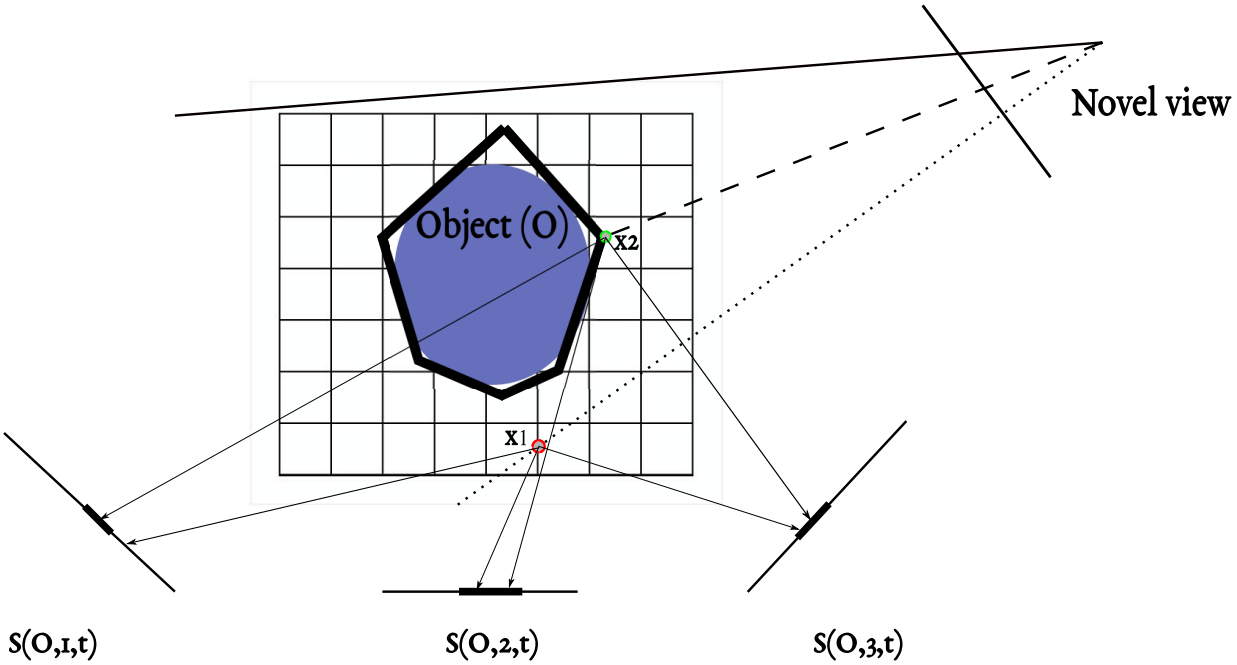


Figure 4.4: a 2D illustration for The process of reconstructing the visual hull using a ray casting approach. .

will be classified as non-consistent rays (dotted line in Figure 4.4) which are rays that did not found any silhouette consistent point while traversing the three-dimensional volume like for example point x_1 in the Figure.

On the other hand, rays which did found intersection with the volume and found a silhouette consistent point are classified as consistent rays which are shown in Figure 4.4 as dashed line. Algorithm 1 provides the detailed description of our parallel strategy for estimating the visual hull in real time.

It can be seen from the above analysis that, a novel view reconstruction based on this technique has several advantages. For instance, the resulted visual hull has no voxelization artifacts in the final render in contrast to the classical volumetric approach. Another important advantage is that such technique is fully hardware-accelerated and shows significant performance improvements and it can be used for real-time applications.

4.2.3 Visual Hull rendering

At this stage, the partial geometrical information of the visual hull that is visible from a novel view point is computed. In fact, each pixel of the output image knows exactly what part of the surface it covers. The natural next step in our pipeline is to render the novel view. Hence, assigning texture value to the estimated geometry. This done in parallel by projecting the point back to the input images and sample the colors than a blending operation is applied to fuse these samples into the final pixel color. Multiple post processing

Algorithm 1: Parallel image-based Visual Hull.

Input: User interaction, Camera parameters
Output: Visual Hull
Data: 3D GRID with Dimension (x, y, z), Silhouettes $S(O, i, t)$

```
1  For Each graphical processor i in {1,...,p} Do in Parallel
2      Launch a visual ray from the corresponding pixel to the graphical processor i.
3      if Intersection(ray, volume) = false then
4          QUIT
5      else
6          March in the direction of the ray throughout the volume
7          foreach S(O,i,t) where i in {1..k} do
8              Check silhouette consistency
9          if point is consistent then
10             Return point
```

algorithm then can be implemented using GPU kernels to enhance the visual fidelity of the final results.

4.3 Experiment

The current section deals with a concrete realization of the system. The proposed implementation is designed for off-the-shelf hardware as a proof-of-concept.The goal of our GPU-Image-Based Visual Hull (GIBVH) is to strengthen the performance of volumetric visual hull estimation via an image-based approach. To this end, we combine the volume rendering technique with a shape-from silhouette approach to reconstruct a novel view of the virtual model.

Multiple experiments are performed to evaluate our implementation of image based visual hull in term of quality and especially quantity. At the same time, we distinguish a different criterion that can affect the performance of the GPU implementation thus, the impact of parameter is examined and discussed. To demonstrate the superiority of the proposed imaged-based GPU-accelerated visual hull, multiple CPU and GPU visual hull implementations are developed and displayed to enable comparison.

4.3.1 System setup

Today, all modern computers include a graphical processing chips that support a large amount of global and shared memory in hardware. Therefore, parallel computing based on Single Instruction Multiple Data (SIMD) architecture is now very popular. With this in mind, a concrete system setup is realized to render in real-time the visual hull of an object on single machine. We decided to carry out experiments on Nvidia

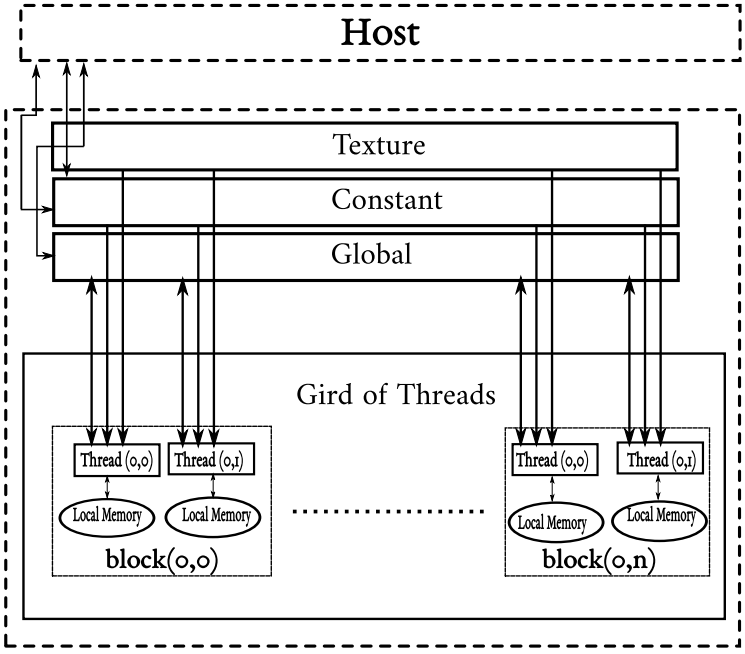


Figure 4.5: High-Level Overview of Nvidia Geoforce graphic chip. .

GeForce GTX 760 graphics card at 2GB of total dedicated memory. This parallel processor is connected to the mainboard viaa PCI Express (x16) Gen2, Its high level overview is provided in Figure 4.5.

With such a multi-threaded architecture, hundreds of thousands of threads can run in parallel, emulating the pixels of the final image. We have carefully adapted our algorithm implementation to the recent GPU-programming tool known as CUDA (NVIDIA Compute Unified Device. Therefore,the implementation has two different parts. A host section, which executes in a CPU side which is mainly responsible for initializations and setups, on the other side, a device part (kernel), which is invoked by the controlling CPU thread, but runs in parallel on the GPU device. Note that a huge amount of data will be moving between the host and the device. In our case , such a transfer is minimized to grantee the real time effect.

In fact, such a programming language offers an easy to use C-like programming interface which allows the developer to interact directly with a graphical device. Lindholm et al [71] gave a detailed overview on GPU programming using CUDA. Others like Fung and Mann [25] discuss the possibility of enhancing computer vision algorithms and image processing research using the power of GPU's. In the context of $3D$ reconstruction of the visual hull, Kim et al. [53] present a volumetric implementation on the modern graphics hardware and show a significant speed-up on a common machine.

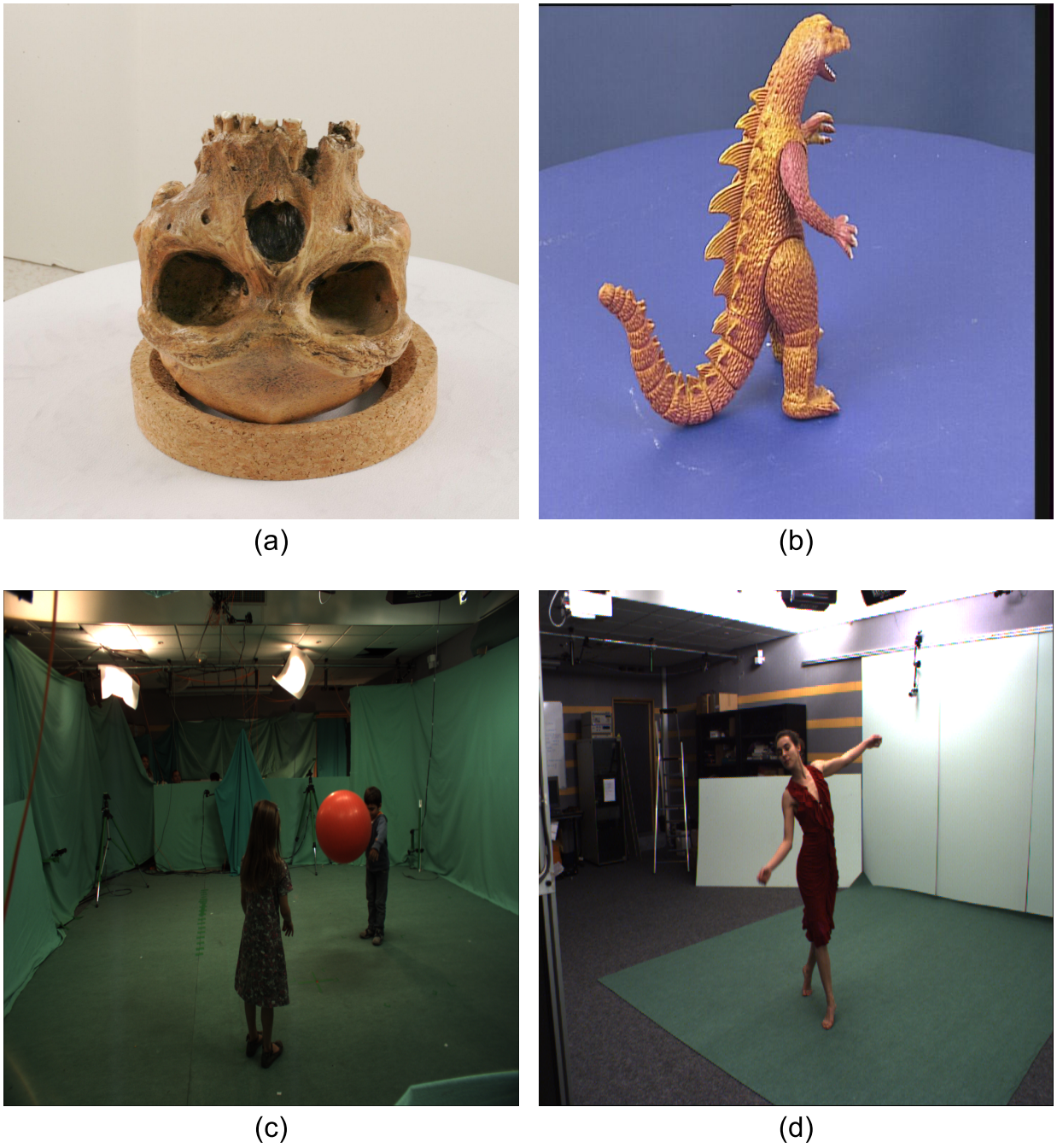


Figure 4.6: Image samples from the Datasets used in our experiments:(a) Human skull dataset, (b) Red Dinosaur Toy dataset, (c) Children Playing dataset, (d) Dancer dataset. .

4.3.2 Datasets

Multiple datasets are utilized to measure the system in term of performance and reconstruction quality.The first image dataset is the one of Visual Geometry Group (University of oxford) [*] multi view stereo datasets. It is a sequence of images of a static plaster Dinosaur toy. The images are captured with a signal fixed camera with resolution equal to 720 x 576. where the object of interest is set on a rotated table. Moreover, the background is set to be coloured with a uniform blue color, 36 pictures were taken where each image was captured after turning the table by 10 degrees. The dataset also offers the projection matrices of each view, the bounding volume can simply be inferred from the projection matrices (*see* Figure 4.6 (b)).

Another datasets for a static scene prepared by Yasutaka Furukawa and Jean Ponce [27] at the university of Illinois which can be downloaded on their web page [†]. From this datasets we chose *Human Skull Cast HomoHeidelbergensis* due to the complexity of object of interest (real life human skull) thus evaluating the robustness of our implementation to such scenario. The dataset consist of a set of 24 images and camera parameters of a single rigid object (human skull), the images has a high-resolution equal to 2000 x 1800 as shown in Figure 4.6 (a). In order to compare the obtained results to other state of art approaches namely the work of Ladikos et a.l [58], the work of Schick et a.l [104], and others, we exploit the Middlebury dataset[‡]. This dataset consists of a plaster object namely the "*Temple of the Dioskouroi*" which is recorded in three different camera setups (16, 47, and 312 views) at 640 x 480 pixels, more on this data will be given in Chapter 5. Lastly, all the images have been corrected to remove radial and tangential distortions using the intrinsic and the extrinsic camera parameters extracted in the calibration phases.

To evaluate the performance of our system in real-time setup, we decided to do our experiments on the dataset of Huang et a.l.[45] provided by the INRIA 4D repository[§]. This martial is designed to capture what the authors refer to as space-time models, which are sequences of $3D$ shape models that represent live and dynamic events, human activities in a given instance. In particular, *girl dancing* is dataset that capture a single rigid actor in the Lab environment. The sequence consist of 8 cameras with approximately 200 frame per camera at resolution of 780 x 582 pixel, the background images are also provided which allow as to recover the silhouette images. The dataset also contain the calibration information for the multi-camera setup. On the other side, *Children playing* sequences depict a dynamic scene from 16 different

[*] www.robots.ox.ac.uk/ vgg/data/data-mview.html

[†] www.cse.wustl.edu/ furukawa/research/mview/index.html

[‡] www.vision.middlebury.edu/mview/data/

[§] www.4drepository.inrialpes.fr/pages/home

view point simultaneously of two children playing with a red plastic ball. Each sequence offers 484 frame at the size of 1624 x 1224 pixel.(*see* Figure 4.6 (c) and (d))

4.3.3 GPU Implementation details

As we mentioned earlier, the original image-based visual hull (IBVH) [78] does not have a precise data representation. This critique, unfortunately, implies that it is complicated to benefit from the power of the modern GPU. On the other hand, GPU-based voxel carving like the work of Chang et a.l. [9] for instance relies on an adaptive volume grid representation and classified each voxel independently to either object or empty space. Subsequently, the reconstructed visual hull suffers from the visible artifacts. To remedy such a problem however, the authors propose a novel technique for creating a compact triangular mesh out of the volumetric representation using GPU implementation thus adding more work to the device leading to low performance.

Our implementation is based on a GPU volume rendering approaches to carve the visible part of the visual hull thus, synthesising a novel view in small time windows without any voxelization artifacts. The hardware we used as we mentioned above is a CUDA capable device with 1152 CUDA cores. Hence, our CPU code uses CUDA specific functions for allocating data on the device and to transfer data to the graphics memory and back to host. We took advantage of the global memory and the pinned allocation of memory space on the host to limit the data transfer from CPU to GPU each frame which is the main important step in our proposal. For the remainder of this chapter, we assume a scene contain one or more objects O that is observed by N calibrated cameras in a Lab environment under fixed illumination.

GPU Silhouette Extraction

At each frame t, images are captured in RGB color format from the source cameras (in the case of our experiments pre-captured videos) C_k where $k = 1...N$. For scene reconstruction, these images have to be uploaded into the GPU. Furthermore, $2D$ image segmentation has to be carried out, resulting in a creation of of silhouette image data. If the silhouette extraction is done on the CPU, the resulted data has to be uploaded as well. This can not be achieved in real-time for a set of $N > 2$ images. Thus, a PC cluster could be utilized and each image is processed on a single CPU in parallel. However, our goal is to run the entire system on a single machine to avoid any networking issues.

Our proposal is to use CUDA kernels to compute the silhouette as fast as possible, In other world, we propose to parallelize over the pixels and not over the input camera. This however, comes with a specific

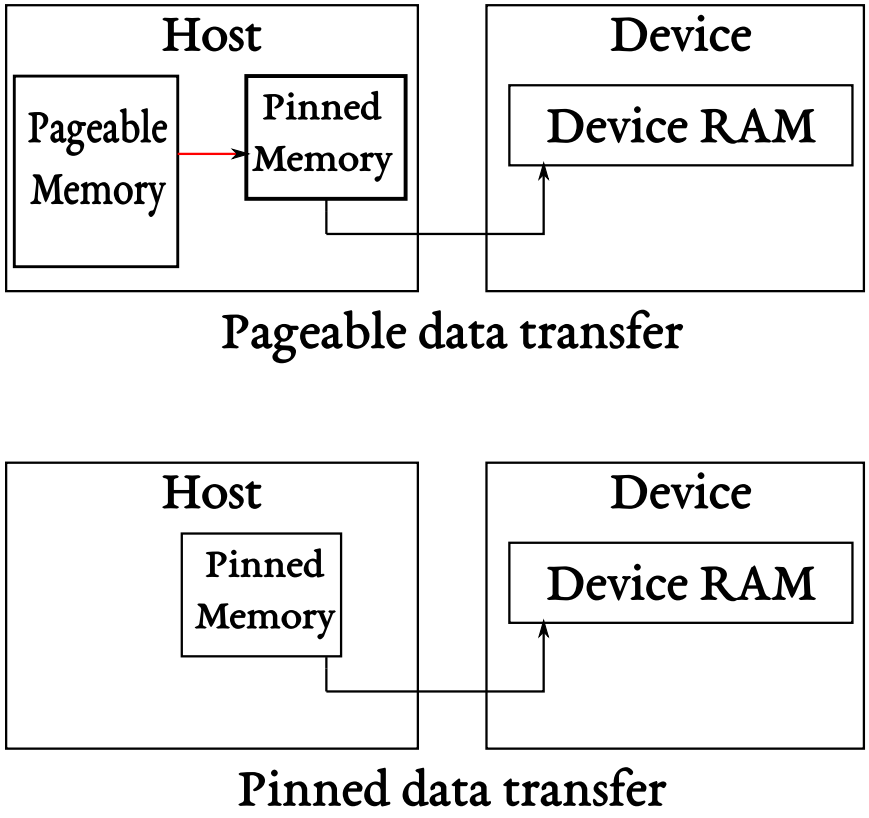


Figure 4.7: Pageable Data Transfer vs Pinned Data Transfer .

challenges namely the data transfer. The transfers between the host internal memory and device are the slowest link of data flow involved in GPU computing. So it is necessary to take care and minimize these transfers.

In order to resolve these issues, we first start by allocating non-pageable memory on the host side. In other word, a one dimensional array at image size with each element of it is a container for an RGB color allocated on the pinned memory space of the host RAM, this array will be filled by the data captured from each frame. Hence, we end up having N pinned arrays waiting to be filled with data and send to the GPU memory.

The logic behind this, is that the device cannot manage data from pageable host memory directly, Therefore, when a data transfer from pageable host memory to device memory is issued, an additional copy operation is initiated by the CUDA driver where it allocate a temporary page-locked host memory, copy the host data to the pinned space, and then spend the data from the pinned memory to device internal memory as illustrated in Figure 4.7.

Image size	Pinned	Pageable
406 x 306	3.975 *ms*	3.99 *ms*
812 x 612	15.74 *ms*	15.96 *ms*
1624 x 1224	64.098 *ms*	65.084 *ms*

Table 4.1: A comparison.between the pinned memory and the pageable memory in term of time consumed to transfer data using `cudaMemcpy`.

Algorithm 2: Data position.

```
int x = blockIdx.x * blockDim.x + threadIdx.x
int y = blockIdx.y * blockDim.y + threadIdx.y
  // the input and output data are stored in 1-D array
int thread1Dpos = y * width + x
int imageSize = high * width
int DataPos = i * imageSize + thread1Dpos
```

The effect will be noticeable especially if there are many transfers of huge chunks data like in our case as it is shown in Table 4.1 where we measured the total time consumed to copy 16 images with different sizes using `cudaMemcpy`, it can be seen that the pinned data is slightly faster. However; the obtained small difference is important since we aim to extract silhouette and render the visual hull multiple times in less than one second.

The next step is to upload the input frame from each camera along with the pre-captured background to the the the global memory of the GPU using the `cudaMemcpy` copy command with `cudaMemcpyHostToDevice`. A sequential CUDA kernels calls are then launched to segment the frame received from each input camera by subtracting each pixel of that frame from the background and thresholding it. As a result, a value of zero to the alpha channel of that pixel is given if it is a background or one if it is a silhouette.

To avoid multiple data exchanges between the host and the device, the silhouette extraction kernels write the final result into one big data array which allocated in the global memory of the device, the size of this one dimensional array is equal to the size of an input frame multiplied by the number of used cameras N. Note that the silhouette information is embedded with the color information of the pixel in the alpha channel. Each thread on the launched kernel has to know where to write the result on the final array of data which will used as input for the next stage. This can be done using the following psudo-code presented in Algorithm 2, The strategy presented here lowers the boundary of the total execution time.

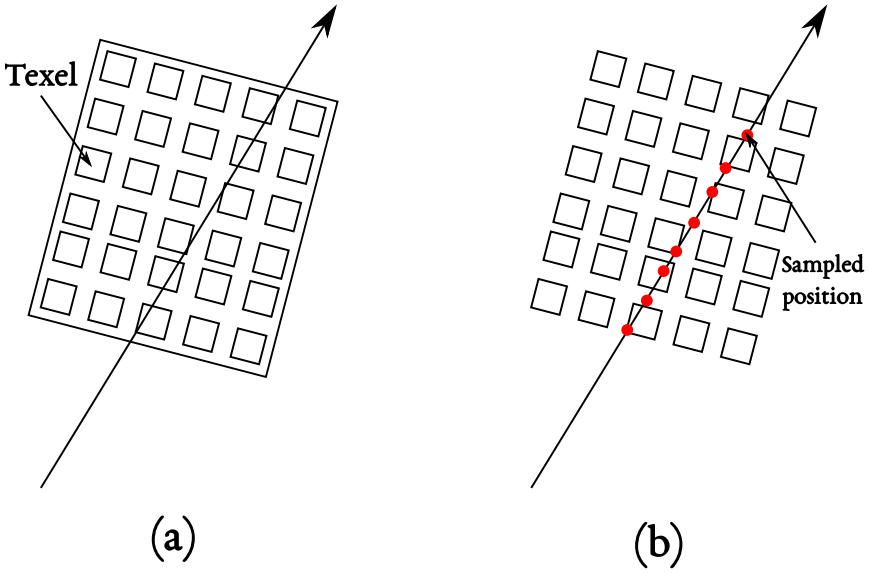


Figure 4.8: A $2D$ illustration for (a) cuda $3D$ texture and (b) interpolation .

GPU Visual Hull

To visualize the reconstruction results, our pipeline is based on three rendering passes namely, the visual hull detection pass, the visual hull texturing pass and finally the post processing pass. Each stage is fully parallelized on the recent GPU-programming tools such as CUDA and Modern OpenGL GLSL. The proposed strategy described in Sect. 4.2 carve out the desired parts of the 3D volume, then estimates the surface final colours.

The goal is to reconstruct the object of interest with no voxelization artifacts in final rendering results. Such objective can be achieved by using the ray casting technique on continues volume data. Voxels are volumetric structures that holds a specific data type. In our work, however, voxels are rectangular space that contain infinite number of points where these points project onto single pixels due to their small size. Therefore, CUDA $3D$ texture memory is suitable for such representation. In fact, the latter is a $3D$ array, each element in it is called a *Texel* as illustrated in Figure 4.8 (a). Each texel is sampled using a linear interpolation thus, making the volume a continues entity (*see* Figure 4.8 (b)). At the beginning of our experiments a $1D$ array which accommodates all the volume data will be initialized by calculating the exact physical position of each center voxel in the grid from a bounding box and a given resolution.

At this stage the color and silhouette images along with the cameras informations are stored in GPU global memory. The three-dimensional volume information is also stored on texture memory of the NVIDIA device. A first kernel is configured to compute the visual hull, the number of threads launched by this

kernel call is equal to the size of the output novel view.

Algorithm 3: Visual Hull Estimation Kernel in NVIDIA Cuda Pseudocode

```
1  Function VisualHull(perspective, volume, silhouettes, cameras):
2      x ⟵ blockIdx.x * blockDim.x + threadIdx.x
3      x ⟵ blockIdx.x * blockDim.x + threadIdx.x
4      Ray ⟵ CreateRay(x, y)
5      hit ⟵ intersectBox(Ray, volume, near, far)
6      if !hit then
7          terminate thread
8      while t < far do
9          position ⟵ tex3d(Ray.pos)
10         Bool consistent
11         forall the silhouettes i do
12             Coord2d ⟵ Project(cameras[i], position)
13             if Coord2d in Silhouette then
14                 Consistent ⟵ TRUE
15             else
16                 Consistent ⟵ False
17                 break
18         if Consistent = true then
19             VisualHull[y * Perspective.wdth + x] ⟵ position
20             terminate thread
21         t ⟵ t + step
22         Ray ⟵ MarchRay(t)
```

Algorithm 3 shows the pseudocode of our kernel that is executed by every thread on the GPU. Each pixel can be identified by a unique id due to the fixed grid size [10]. Each thread projects multiple texels into every camera image. Only if the texel projects into the silhouette for all camera views, than it's position is added into the visual hull result array.

Note that the `camera[i]` array hold the projection matrices of all input images. These matrices are constant per image (view) hence, CUDA language makes available another kind of memory which is known as constant memory (*see* Figure 4.5) . As the name may suggest, the constant memory is used for data that will not change during the course of a kernel execution. Using this space to cache data boost the kernels performance for two reasons [10] :

- First, a single read from constant memory can be broadcast to other neighbouring threads, thus, effectively saving up more than 15 reads.

- Second, constant memory is cached on the device, so successive reads from the same address will not provoke any additional memory traffic.

The constant memory is so effective. However, Nvidia hardware provides only 64KB of it [86], which is very little thus limiting our system to a small number of input cameras.

When a $3D$ point that is visible from the novel view is proved to be part of the visual hull, it's colour should be computed from the input images. A new kernel is launched for the visual hull texturing, such kernel can compute the colors using a bit of additional computation to project the visual hull back to the i camera images which are already stored on the GPU global memory and sample the colors. The final color is the weighted average of these sampled colors. The pixels color $Color(x,y)$ can be computed using the following formula:

$$Color(x,y) = \sum_{i=1}^{N} (1.0 - W_i) \cdot Color_i + W_i \cdot Color_{i+1} \tag{4.2}$$

We achieve view-dependent texturing by including a weighting factor W_i . The weight depends on the inverse distance of the surface from the reference view. Thus leading to colouring the object with the closet view to it.Such approximation leads to faster rendering kernel as well as less demanding graphics hardware resources. The result of this pass is $2D$ texture image which will be used for the next stage.

Finally, the last stage of our pipeline is the post processing pass. First the CUDA kernels finish computing and texturing the novel view on the OpenGL buffer or texture image, such process is called OpenGL interoperability with CUDA. Furthermore, a GLSL shader fragment programs is created to post process the attached texture then displaying it on the screen.

4.3.4 Results

In the previous sections, we have proposed several strategies aimed to make Shape-From-Silhouette more effective in the context of virtual reality. We will show throughout our experiments that the solutions provided allow the definition of simple and effective real time visual hull on single machine with a single consumer graphical device. Overall, the result presented in this sub-section shows an improvement in performance, while reducing the volume size compared to the original silhouette carving algorithm, and the image-based visual hull approach. We evaluated our algorithm in two different scenarios: static objects and multi-camera environment.

Data	Volume size	Our approach	Ladikos et a.l [58]	Schick et a.l [104]	Voxel Coloring	GPU Space carving
Temple ring	128^3	18.866*ms*	372.95*ms*	151.00*ms*	67890.00 *ms*	14.631 *ms*
Temple ring	256^3	19.892*ms*	3022.10*ms*	1036.00*ms*	578 x 10^3 *ms*	84.977 *ms*

Table 4.2: Runtimes of our approach on Middlebury dataset compared to results provided by Ladikos et a.l. and Schick et a.l

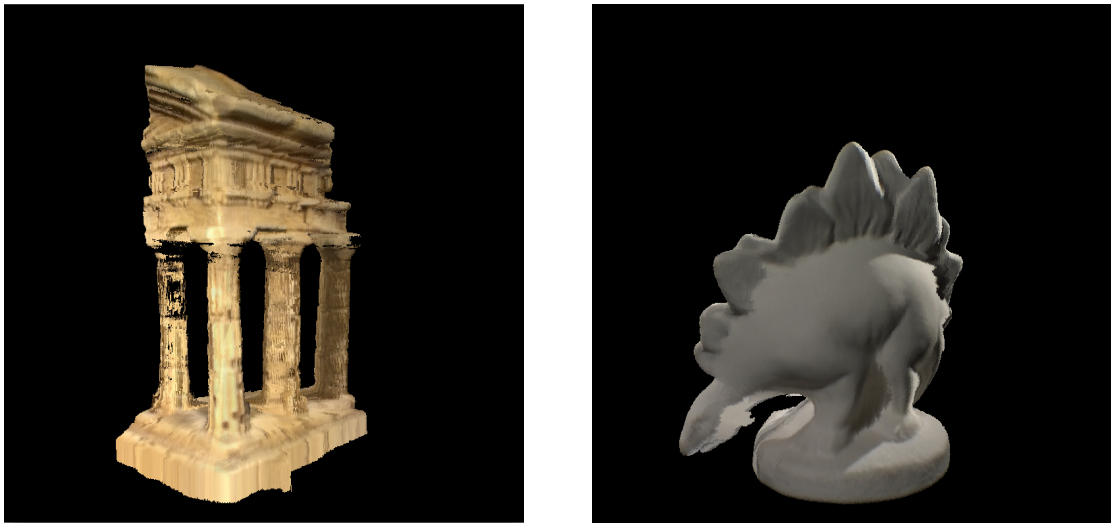


Figure 4.9: Results on the Middlebury dataset. The images show examples from our GPU images-based visual hulls, many details were preserved with no voxalization artifacts. .

Static Objects

First, static objects which are captured via multiple photographs. In the context of virtual reality, it is possible generate a novel view of the object of interest and blending the result with the virtual world. Hence, reducing the effort to model huge load of prefabs, and let the creator of the virtual world focus on more important things. Our parallel implementation uses off-the-shelf hardware and can be used for the previously mentioned scenario. Evidence for in support of this position, can be found in the above Table 4.2.

This table presents the computation times in milliseconds for *temple ring* object with two different volume configurations. We also compare our GPU image-based visual hull to the work of Ladikos et a.l. [58], which is similar to our approach in term of the objective, that is computing the visual hull in real time, it is also considered one of the fastest GPU-based voxel carving approaches in the literature. The authors use multiple machines with dedicated graphic cards simultaneously which make it difficult to preform a direct comparison. However, our approach is faster than their direct algorithm by factor of 20x for a volume with a size of 128^3 and by factor of 152x for the other case. Our work is also compared to that of Schick

et a.l [104].

The latter propose a voxel based visual hull reconstruction in real-time. Moreover, their approach can handle the presence of occlusions. Comparing the reported performance of that implementation to our GPU image-based rendering, we can see that our approach is more faster, note that the authors run their kernel using NVIDIA Geforce GTX 280 chip which is very old. On the other hand, we implemented a brute force version of Schick et a.l [104] work on the GPU, However, we did not consider any occlusions. The results are shown on the last column of Table 4.2 which tell us that in case of volume at the size of 128^3 our version of Schick et a.l [104] approach did slightly better than our proposed approach. While when doubling the volume size of the same dataset the performance became 6 times slower.

In contrast, the performance of our proposal stayed stable. It is however, important to note the limitations of this type of implementation in the context of virtual reality. As we mentioned earlier, using voxels alone will result in a non-realistic visual rendering which is not the case for our implementations as shown in Figure 4.9.

In fact, our visual results achieved an acceptable quality. However, image-based rendering and novel view synthesis can not be evaluated in term of visual fidelity. Michael et a.l. [129] propose a new virtual rephotography-based benchmark for image-based modeling and rendering systems. Unfortunately, the metric was not affective thus, the benchmark was not used and removed from the official website. In order to assert the quality of our visual rendering, we compare the original input images with the reconstructed novel views. Figure 4.10 shows such comparison, the top two images are the original input images used to reconstruct the image-based visual hull of *the skull HomoHeidelbergensis*.

On the other side, the two bottom images are the results of our implementation with a resolution of 800 x 600. In this experiment, we used only 8 views from the original dataset. The obtained results share almost the same view point of the original images. It is visible to the naked eyes that we were able to recover some important details which are highlighted with the red square on the figure. Note that the voxalization effect are absent which add more realism to the rendered image.

Figure 4.11 on the other hand shows the effect of voxalization (*see* image (c)). This effect is obtained by always fetching the center of the voxel when the visual ray traverses the volume. Note that The voxelsize defines the rendering performance along with visual quality due to the volumetric resolution. then again,

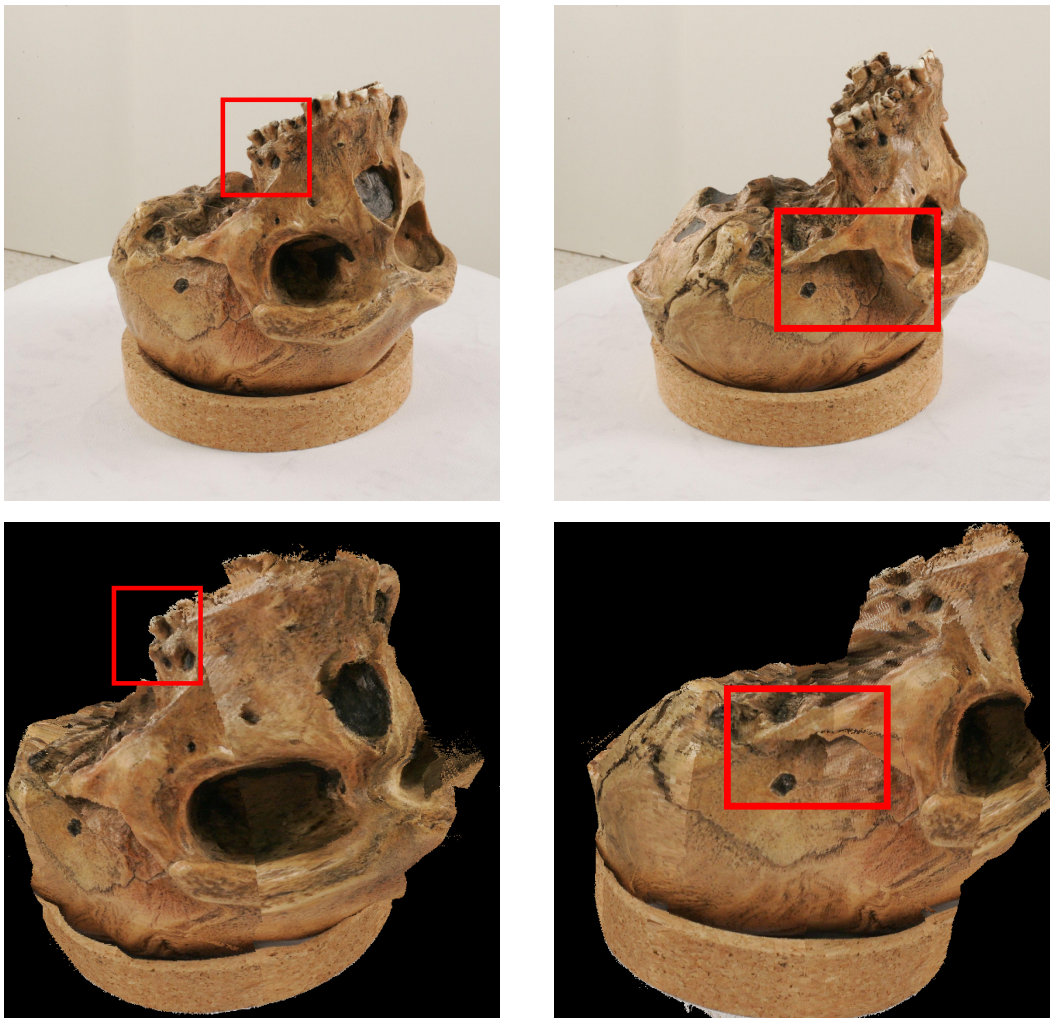


Figure 4.10: Qualitative results of our GPU image-based visual hull reconstruction on *the skull HomoHeidelbergensis*. The first row represent two images from the data set. While the second row shows the reconstructed views at output resolution equal to 800 x 600 .

using the three-dimensional linear interpolation offered by the CUDA API (Application Programming Interface) remove definitely the non desired effects of voxalization as it is shown on the same Figure 4.11 (b). Moreover, the visual details are automatically adapted according to the viewpoint by taking into account directional information, and the weighting function we introduced to compute the final color.

Multi-Camera Environment

In contrast to the typical scenario of multi-view 3D reconstruction of static scenes, a generalization to a multi-camera environmentl setup brings up several practical challenges mainly the high demands on memory and computation time. We also evaluated our GPU image-based visual hull implementation in a multi-camera environment. The results presented in this part are obtained by applying the proposed parallel algorithm 2 on the datasets presented in Sect. 4.3.2 namely the *Children playing* and *girl dancing*.

(a) (b) (c)

Figure 4.11: Qualitative results of our GPU image-based visual hull reconstruction on *Dino Toy*. (a) Image of the object from the dataset. (b) A reconstructed visual hull using our approach with interpolated Texels. (c) A reconstructed visual hull using our approach with interpolated Texels. .

Dataset	Image resolution	Phase I	Phase II	Phase II
Children playing	406 x 306	0,027 *ms*	1.736 *ms*	0.140 *ms*
	812 x 612	0,099 *ms*	1.874 *ms*	0.141 *ms*
	1624 x 1224	0,365 *ms*	2.141 *ms*	0.144 *ms*
Girl dancing	195 x 146	0,221 *ms*	1.323 *ms*	0.139 *ms*
	390 x 291	0,868 *ms*	1.340 *ms*	0.138 *ms*
	780 x 582	3,399 *ms*	1.395 *ms*	0.143 *ms*

Table 4.3: Performance statistics on our GPU accelerated visual hull construction method. Two different datasets were tested with respect to three images resolutions, Only 5 input cameras are used and the output novel view resolution is equal too 800 x 600

The former presents multiple actors captured via 16 camera, while the latter showcases a single person preforming in front of 8 cameras. Both of the datasets are captured in lab environment.

Table 4.3 shows statistics measured on an NVIDIA Geforce 760 GTX hardware with respect to three different volume resolutions. Where each row reveals the size of the input images used from both datasets along with the relative GPU execution time for the three computational phases. Here, phase one represents the silhouette extraction stage. In fact, the time showed in the table for phase one is the average execution time for the five input views. For the second phase where the visual hull is reconstructed, it can be seen that the running times remain almost constant while the size parameter increases whereas. The same can be also said for the final phase which consists of coloring the visual hull.

We observe that, in all the cases, a good speedup is reached thus allowing action and interaction during the virtual reality experiment. Furthermore, this experiment showed that the number of actors presented

	320 x 240	640 x 480	800 x 600	1024 x 768	1280 x 1024
Excursion time	0.771 *ms*	1.583 *ms*	2.675 *ms*	3.276 *ms*	5.061 *ms*

Table 4.4: The impact of the output novel view resolution on the excution time of our GPU implementation for the *Children playing* dataset

in the multi-camera environment does not affect the performance of the algorithm significantly. This is due to the absence of any geometric representation in the proposed strategy. On the other hand, if we take a deeper look at each dataset, we can see that the performance slowly decreases when the number of the input pixels doubles by factor of two, which was expected from the theoretical analysis of our parallel algorithms.

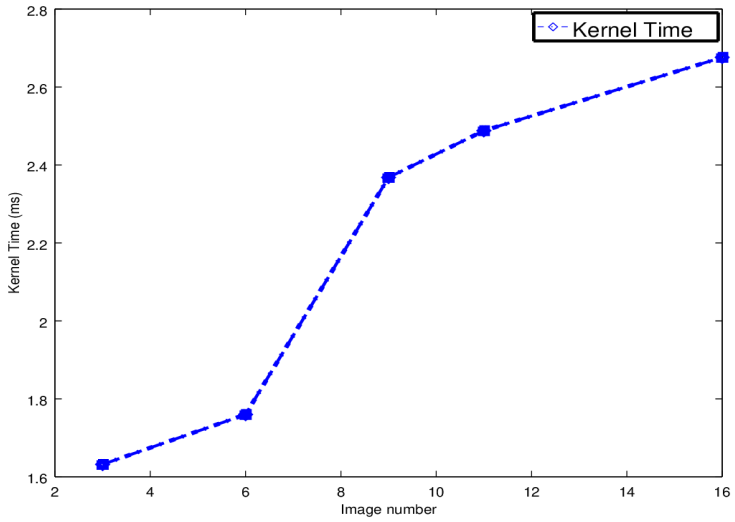


Figure 4.12: The Kernel execution times on the down scaled *Children playing* dataset of the parallel image-based visual hull algorithm plotted as a function of the number of input images per frame. .

Figure 4.12 on the other hand, shows another factor that slightly affect the execution time of the visual hull reconstruction GPU kernel. Clearly, and as expected, the number of input images will decrease the performance of the visual hull kernel. However, from the experiment it can be seen that with 16 input cameras the execution time was equal to 2.6 *ms*. Thus, if our goal is to achieve a real-time virtual reality experience with only 30 frame per second, it can be seen that we still have almost 15 extra milliseconds assuming the silhouette extraction take only one milliseconds. From this experiment, we can conclude that using 20 cameras with a small resolution equal to 406 x 306 results in a 30 frame per second reconstruction system with an acceptable visual quality using our GPU based implementation.

In another experiment we rendered a representative GPU image-based visual hull scene using the same dataset *Children playing* from a novel viewpoint that is close enough to have the actors appear fully on the screen. We captured the rendering times of the first frames at varying display resolutions. Table 4.4 shows performance measurements for our GPU implementation. While the resolutions doubled every step, the execution times maximally increases by one millisecond. This is promising for higher camera resolutions in the future.

Figure 4.13: Qualitative results of our GPU image-based visual hull reconstruction on *Children playing* integrated in virtual environment. (A) A reconstructed visual hull using our approach with interpolated Texels. (b) A reconstructed visual hull using our approach with interpolated Texels. .

The previous experiments was designed to asses the gain of our parallel image-based visual hull implementation. Let us now move to assess the visual results obtained. Figure 4.13 shows the visual hulls of multiple people in our multi-camera environment from a new viewpoint. The images show results obtained at a grid resolution of 64^3. Furthermore, the visual hulls in image (b) are detailed enough to provide an immersive

virtual experience in real-time which allow interactions. The results show a visual fidelity that emulates the state-of-art multi view stereo algorithms. Note that the quality increases the more input images are used.

4.4 Critical analysis

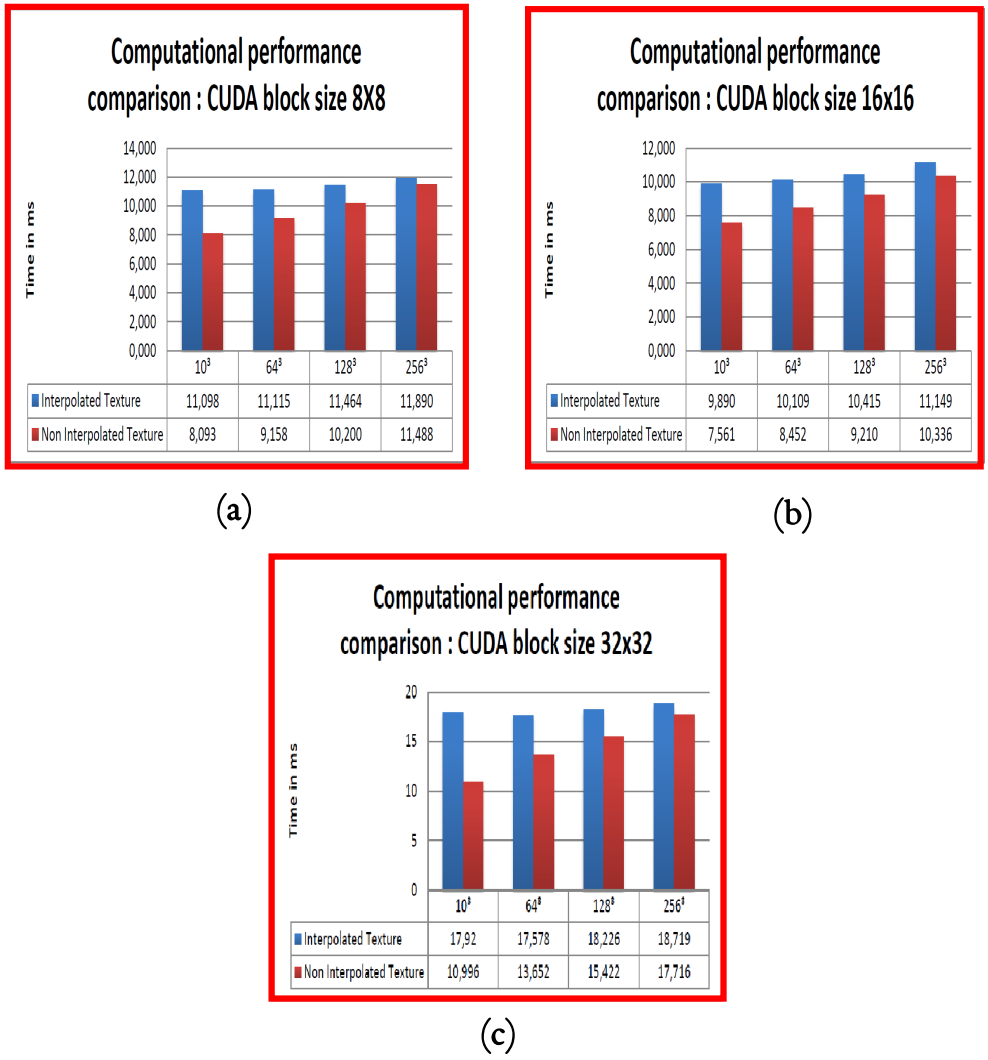


Figure 4.14: Computational performance comparison: image input size is 720x576 and the size of novel view is 800x600 approach with interpolated Texels. .

As we mentioned earlier, the first step to our GPU image-based visual hull reconstruction approach as well to every shape from silhouette reconstruction methods is detecting the object of interest. After that, the image data should be situated in the device global memory waiting to be used. The second phase starts by carving the visual hull and render the novel view pixel by pixel in parallel. However, there are a lot of criterion that can define the performance of our kernel. In this section we took those who dramatically impact the computation time.

It can be seen from the charts presented in Figure 4.14 that the volume size affects the performance of the GPU image-based visual hull implementation. The voxelsize influences reconstruction performance through volumetric resolution. In fact, the lesser the volume size is, the higher the performance will be in all the presented cases. However, when we look up the information from the three-dimensional volume stored in the GPU texture memory using the linear fetching mode also known as the interpolated texture access (see the blue bars in Figure 4.14 (a, b, c)) the performance of the image-based visual hull decreases slightly whenever the volume double in size. Take for example the first case illustrated in Figure 4.14 (a), the average change in time is equal to 0.2 milliseconds. However, for the point texture fetching mode (non-interpolated case) we see noticeable difference where the average change in time is equal to 1.13 milliseconds.

This results is due to the nature of three-dimensional volume and how it is stored on the device memory as we explained in Sect. 4.3.3. In fact, in the case of interpolated texels the rays always sample from the 4 neighbouring points, hence, the number of the voxels is relevant. This is not the case on the other hand, due to fetching the nearest texel from the $3D$ texture without the need of sampling which lead the performance being affected by the amount of the voxels traversed.

Furthermore, it is not possible to determine in a general way and validate the method of setting or maximizing the GPU occupancy rate. This last is defined according to CUDA programming guild as the ratio of active warps on an streaming multiprocessor (SM) to the maximum number of active warps supported by the SM. Evidently, a low occupancy results in poor instruction issue efficiency. One of the variables that affect the occupation is the partitioning of the threads, and the size of the block which is very difficult to determine. In our experiments, the results show that the best execution times were found using configurations that have a 16x16 threads per block (*see* Figure 4.14 (b)). This means that we divided the output novel view of a resolution equal to 800x600 into multiple block, and every block have 256 pixel. Hence, the configuration yield a good balanced workload among the warps in each block.

Figure 4.15 shows the visual differences between the previous configurations using the *Dinosaur toy* dataset. It can be seen from the image (b) that our image-based visual hull fails to reconstruct the novel view of the visual hull at volume size equal to 10^3 this is applicable when the CUDA filter mode is set to point fetching (no interpolation). In contrast, the full form is captured when fetching the $3D$ position from the volume using the linear mode as it is illustrated in the image (a). This results were expected, according

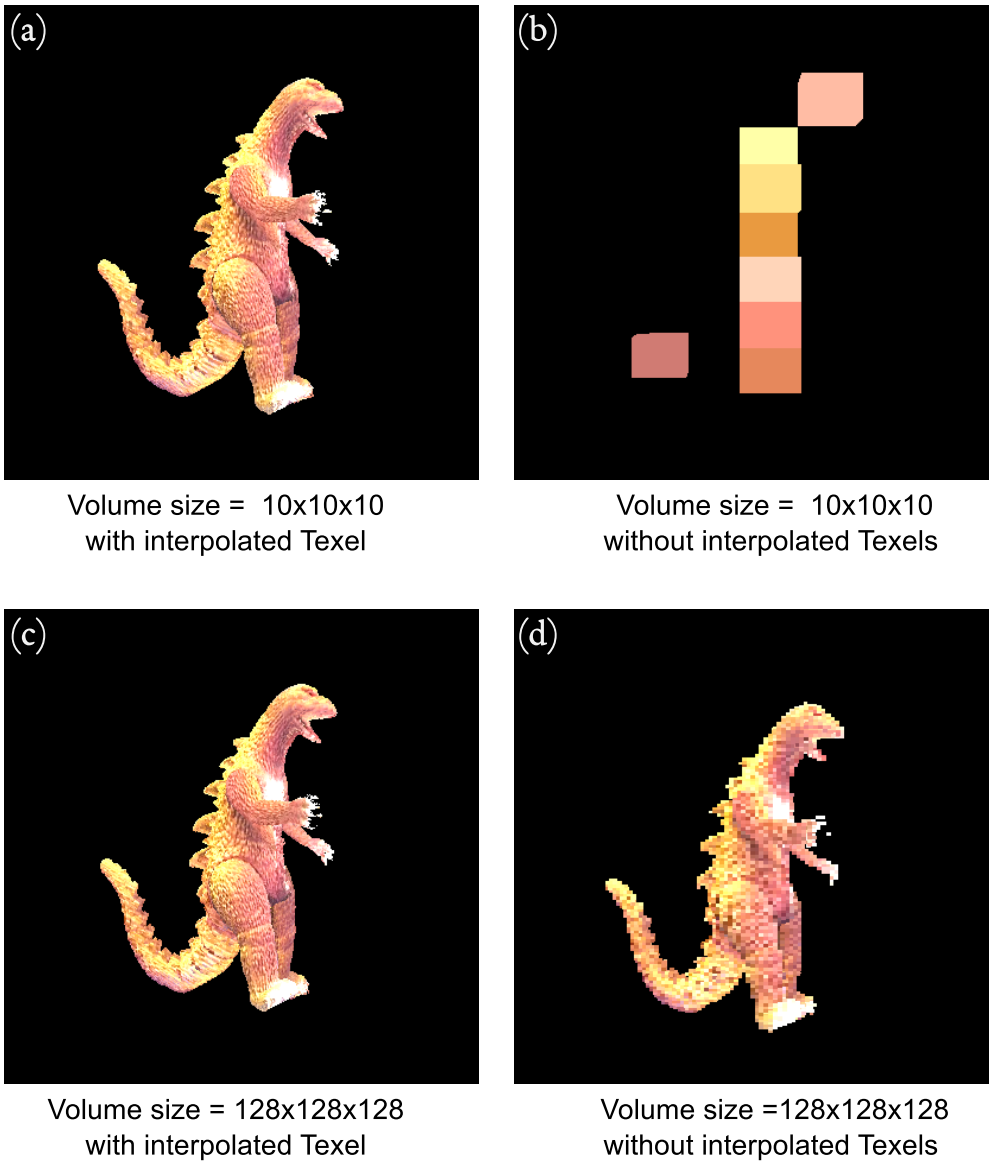


Figure 4.15: comparison between the result qualities of our approach of each configuration using 36 input images from *Dinosaur toy* dataset and the size of novel view is 800x600s. .

to our algorithm analyzes. Hence, this experiment proved that it is possible to represent the object of interest with minimal size of voxels without any impediments in regard to quality of the final render. Evidence for in support of this position, can be found in Figure 4.15 (a,c).

4.4.1 Complexity

The complexity of reconstruction kernel is affected by the novel view position of virtual camera, which is reasonable due to the implementation of our image-based visual hull is based on the ray-casting approach. Therefore, the complexity of the second phase of our system depends on the viewing rays and how far it travelled through the three-dimensional volume. Since, our implementation is on graphical hardware,

and each ray is represented by a single threads. The complexity is than given according to the worst case scenario. In another words, the execution time of the reconstruction kernel is equal to the time of the slowest launched ray from the novel view point. Imagine now that a given ray intersect the object bounding box, this ray start traversing the volume while checking for silhouette consistency.

The worst case is when the ray traverses a large distance to find the visual hull, or even traverse the whole volume without finding any consistent point. The running time is $O(tm)$ where t is the number of steps taken to traverse the ray, and m is the number of input images. Note that the algorithm can fail the reconstruction if a big step is chosen to walk through the volume, thus, a small step is required to minimize the chance of skipping $3D$ point that contribute to the final visual hull.

4.4.2 Multi-GPU multi thread architecture

Today we see that the hardware aspect of computer graphics evolved in a monstrous way compared to its earliest days. This chapter presented an effective way that utilise the power of these GPU's to create an interactive reconstruction system that can be used in virtual reality experience. However, there is still a room of improvement both in term of quality and performance. In fact, a modern GPU microarchitecture called *KEPLER* was introduced as the successor to the old *Fermi* microarchitecture. This kind of GPU architecture brought with a new concept that is still not well developed, this concept known as *Dynamic Parallelism*. As a consequence, developpers has now the ability to launch new kernel from the inside of the device dynamically ,simultaneously and independently.

One perspective of this thesis, on the computer architecture side, is about the exploitation of this new *KEPLER* architecture to enhance our GPU image-based visual hull. In fact, there are multiple ways to use multiple GPU's on the same machine along with a several CPU threads. Figure 4.16 present our envision for a higher quality and interactive reconstruction system implemented on single machine equipped with a modern CPU that has $N+1$ core and two NVIDIA devices which are build upon the *KEPLER* architecture.

In particular, each CPU core launches a GPU kernel on the first device to extract the image silhouette from the input frame, This kernels are asynchronous and streamed to achieve low latency, on the same time a master core on the CPU side launches the reconstruction phases, as soon as the first GPU finish extracting the silhouette the data is send to the second GPU global memory and the visual hull estimation phase start on this device. This kernel however, will also execute another kernel but on the first device hence the term dynamic Parallelism, this kernel will compute visibility which was not introduced in our algorithm do to

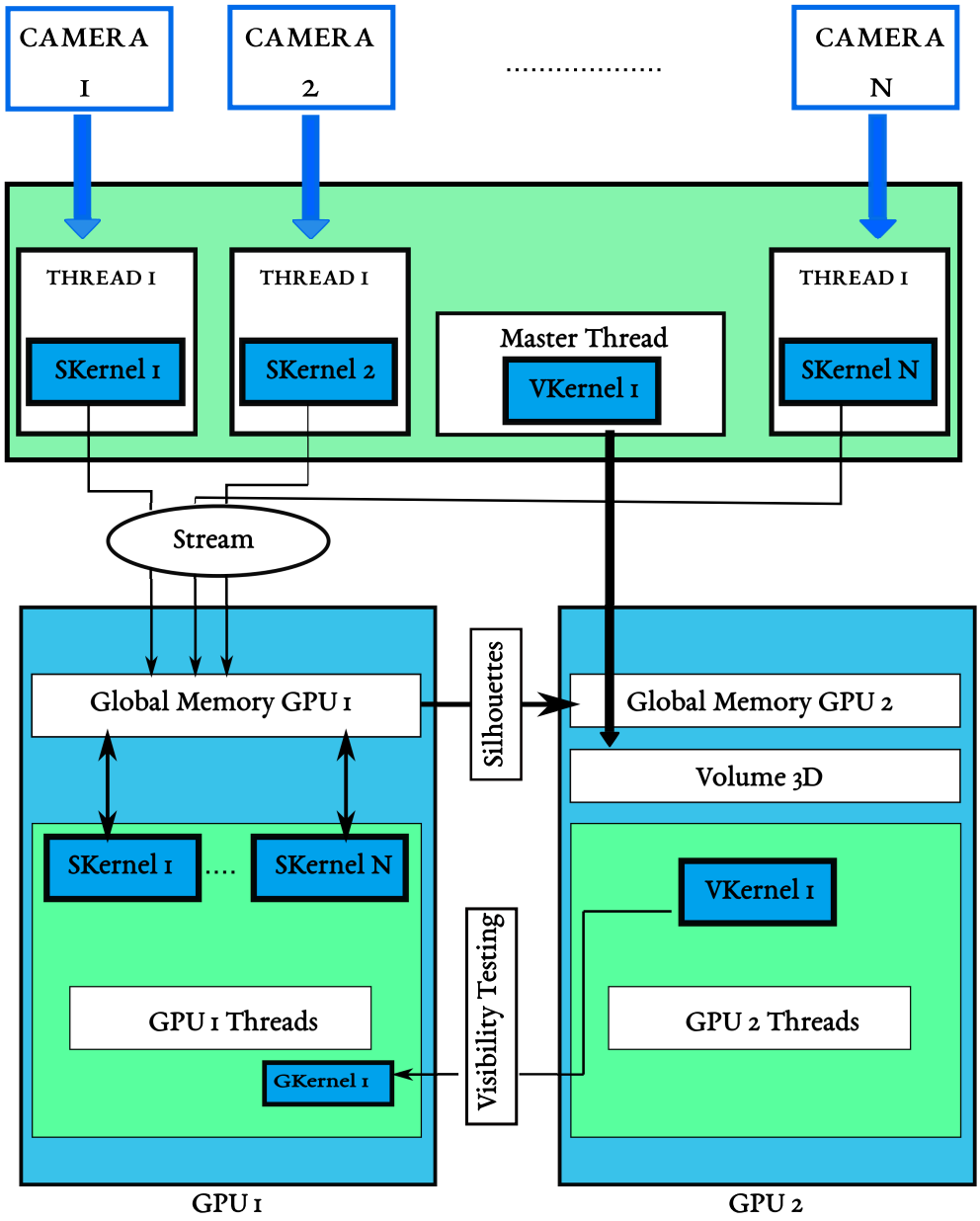


Figure 4.16: Theoretical architecture of our GPU image-based visual hull approach using CUDA Dynamic Parallelism. .

it's complexity. The resulted informations are then used to color the final visual hull.

4.5 Conclusion

In this chapter, we proposed a parallel strategy that leads to a fast computation of the image-based visual hull. Hence, such implementation could be used for virtual reality and tele-presence system. We have managed to enhance the performance of the visual hull algorithm using the GPU image-based approach. The careful usage of the CPU pinned memory and the GPU global and constant memory along with the

power of CUDA threads resulted in a parallel system capable on rendering more than 50 frames per second.Using the linear mode to fetch the information from the three-dimensional texture help to enhance the visual quality of the reconstruction.

The work presented in this chapter clearly has some limitations. Hence, a room for improvement is always available. Despite this, we think our system could be a starting point for photo-consistent or stereo approaches for a smoother 3D model. A device optimizations and multi-GPU usage is proposed as perspective for future works. The next chapter is devoted to propose a novel method which aim to enhance the accuracy of the reconstructed objects which are charachtrized by quasi-Lambertian surfaces, and small amount of textures.

Forty years ago, we had pong. Two rectangles and a dot, that was what games were. Now, we have photo realistic 3D simulation and it's getting better every year. If you assume any rate of improvement at all, then the games will become indistinguishable from reality.

Elon Musk

5

Accurate and Realistic World Reconstruction

In the early stages of our lifetime, we realized that the world is bigger than the house and the city we live in. Years later, we understand that it is impossible to see all the beautiful places and locations of this vast world. In this modern era however, a number of research teams in computer graphics and computer vision are starting to harnessing the power of *Virtual Reality*, and a technique known as *Photogrammetry*, to make the dreams became reality. In fact, there are some things that are highly intriguing about the combination of virtual reality and photogrammetry that we start to think about. Some of these intriguing things that we are excited about in the future, how VR is going to transform communications.

Basically you can put-up your head mounted display, and you can be communicating and talking to some body who is very far away. Moreover, that person feels like he is literally standing in front of you. In fact, this is what we tried to achieve throughout the previous Chapter 4. Know imagine if you had a collection of photographs or a video which capture the memory of a happy moments of your life, and you want to re-live these moments. Using photogrammetry to convert the $2D$ images to a $3D$ model that can be inserted in a computer simulated environment, then the whole experience can be visualized using the virtual reality tools.

Photogrammetry also known multi-view stereo reconstruction is the method of capturing, measuring, and fusing a multitude of two-dimensional photographs of real-world scenes for the purposes of generating

Figure 5.1: Two images of a synthetic object which illustrates the multi-view stereo major problem namely reflective and homogeneous surfaces along with some thin details. .

photorealistic and explorable $3D$ environments. As a consequence, the problem of MVS reconstruction consist of the fundamental task of establishing dense correspondence between images. However, multi-view stereo methods [108] still perform poorly in several cases. In fact, the world is not pure-Lambertian. Objects can have homogeneous surfaces or thin-structures. Furthermore, what we perceive in this world is determined by surface geometry, reflectance, and illumination. Hence, due to the high connection that relates these aspects, it is quite difficult to correctly estimate each one independently.

This chapter presents our method for reconstructing high-quality solid models of complex three-dimensional natural shapes from a cluster of calibrated photographs. We address two major limitations of the multi-view stereo algorithm namely the lack of texture and non-Lambertian surfaces. An example of such surfaces is presented in Figure 5.1

5.1 Problem Description

As we mentioned in Chapter 1, the process of reconstruction based on matching image features will undoubtedly be affected by the physical nature of the reconstructed object and its different interactions with the light. Before we present the possible solutions, let us restate the main problems with more details.

5.1.1 Homogeneous Surfaces

First textureless surfaces, also known as homogeneous regions considered as the worst case scenario for any multi-view stereo approach. The absence of texture cues leads to ambiguous results during the matching process. In fact, the color information contained in the different input images is highly identical which is not enough to locate the matches confidently.

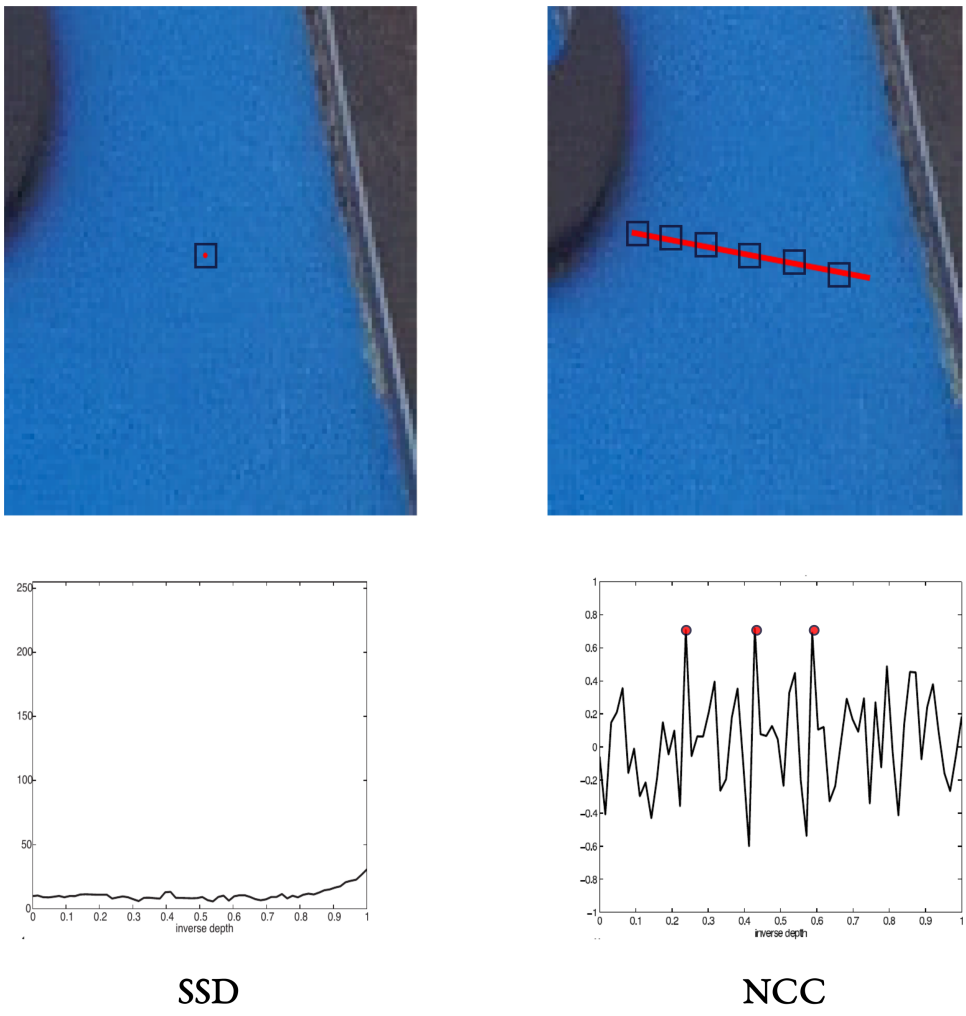


Figure 5.2: Example of two untextured images used as input to a photo-consistency algorithm matching a block of pixels (center of the left image presented as red point surrounded by black box) against a second image across the epipolar line. The second row illustrates the different photo-consistency measures namely the NCC and SSD computed for the above texturless images (images courtesy to Yasutaka Furukawa) .

In particular, the multi-view stereo model basically try to match pixels while assuming that they belong to a Lambertion surface. In fact we search among the possible depth and normal values for a solution that gives a correct stereo match. However the textureless surfaces presented in the input photographs throw this assumption out of the window. Basically, the photo consistency measurement in this case gives almost identical values across the depth range. Consequently, estimating depth values in such regions requires a certain amount of guesswork.

Take a look for example at Figure 5.2, The top two images represent an extreme case of a homogeneous region. In this experiment, we wish to estimate the depth information of the center pixel (red dot) of the left image, a block of pixels is selected around that specific point (which illustrated in the image with black rectangle) and matched against the right image along the epipolar line which is represented with red line.

The matching process is done by computing the photo-consistency score and choosing the best one.

The bottom two images represent the photo-consistency plots for the top untextured region. In particular, the left image shows the depth values in term of the Sum of Squared Differences (SSD) measurement. Clearly, it is impossible to find the correct depth that scores the best photo-consistency measurement. The same thing could be said on the Normalized Cross Correlation (NCC) which is shown on the right image. The ambiguity is represented on the plot by the three red dots. Evidently, it can bee seen that there is 3 different depth values with the same consistency score.

Unfortunately, such surfaces are very common in human made environments and objects such as walls, indoor building and furnitures. To remedy such a complex problem, researchers suggest to incorporate external projectors during the process of image acquisition as one of the easiest and practical solutions. One of these solutions is the first Microsoft Kinect Camera which has been used to capture indoor environments in real time [47, 81]. This camera uses an infra-red light projection along side the RGB camera to recover in real time the depth information directly. which is not possible using the MVS alone. On the other hand, passive approaches that does not benefit from any external support also exist. For instance, a photometric method would work perfectly with MVS [23, 18] in order to recover the necessary information from the untextured regions thus reconstructing complete $3D$ shapes with improved qualities.

5.1.2 Non-Lambertian and Thin-details

The main feature of multi-view stereo methods as we mentioned earlier in Chapter 3 is to evaluate small image patches and look for matches which have the same or similar appearances over multiple photographs. Consequently, almost all of the state-of-art algorithms [29, 108] assume Lambertian reflectance to retrieve the three-dimensional information. In fact, such surfaces practically reflect a constant amount of energy around a given point on that surface over the hemisphere domain. This assumption is of course not true in practice, and could lead to an inadequate approximation. Hence, it is impossible to account for self-shadowing, indirect illumination and specular reflection.

Figure 5.3 shows the effect of a non fully diffuse surface, It can be seen that the same $3D$ area has different color patterns, this is due to how the surface interacts with the light and reflects the incoming energy according to the view position. Such an effect cause holes in the reconstructed model. In particle, during the matching process a surface area with different visual appearance across multiple photographs is rejected.

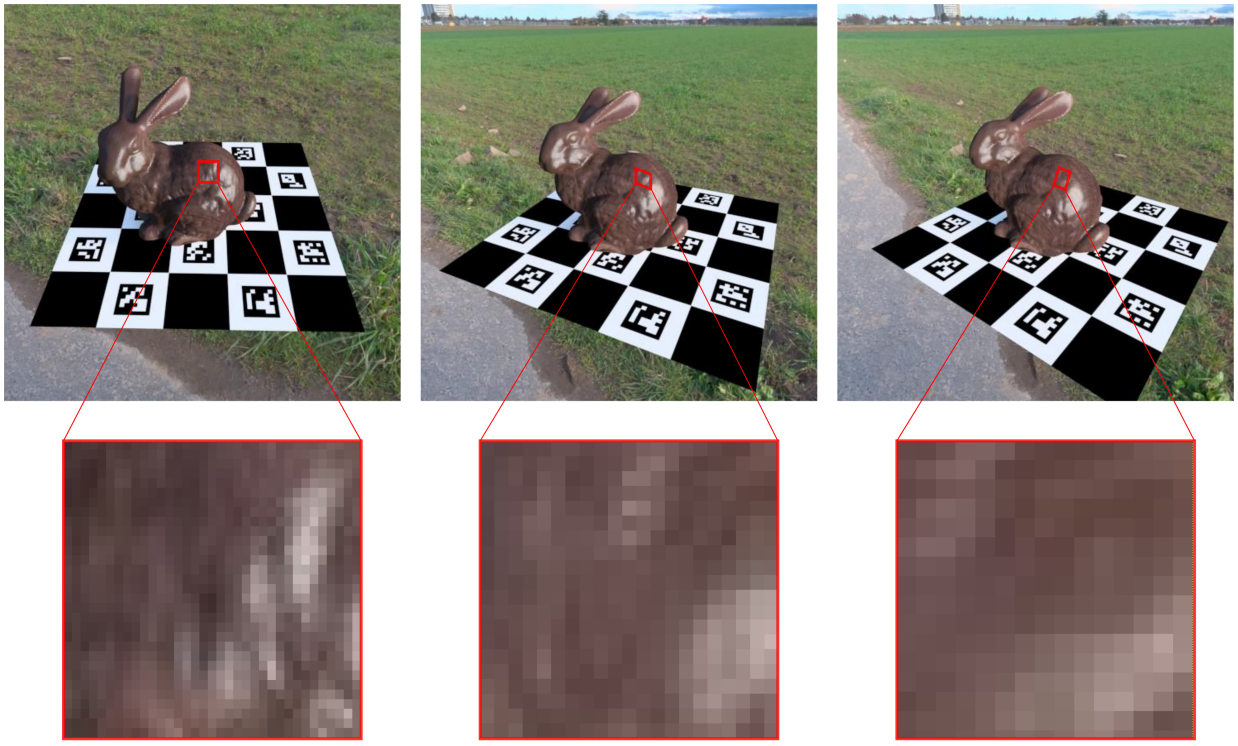


Figure 5.3: A non-Lambertian surface illustration using a synthetic model of bunny, the first row show three diffrent view position of the same object. due to the surface nature a given point may have different color in each image .

Despite all of these criticisms, it is verified that some MVS algorithms successfully generated non-Lambertian surfaces as long as the diffuse reflectance component is distinguishable. To solve such problem, recent approaches estimate the bidirectional reflectance distribution functions (BRDF) (*see.* Chapter 2 using complicated setups, and use it in combination with the standard multi view methods to recover and refine the three-dimensional shape [43, 91].

On the other hand, small objects and thin-details is considered as stumbling block for the standard MVS algorithms [34, 119]. The problem raises when choosing shape and size of the domain Ω used to evaluate the photo-consistency (*see* Chapter 3). In fact, most of the time this domain is represented as a $2D$ window of several pixel wide. In this case, small-details which are as wide as only a few pixels (lesser than the size of Ω) in images could be lost in the process of reconstruction.

5.1.3 Motivations and Proposition

According to the taxonomy of Seitz et al. [108], dense multi-view stereo (MVS) reconstruction methods can be categorized based on the output scene representation. For instance, one popular representation is a three-dimensional volume, where in this representation, the object can be described via gird of small units

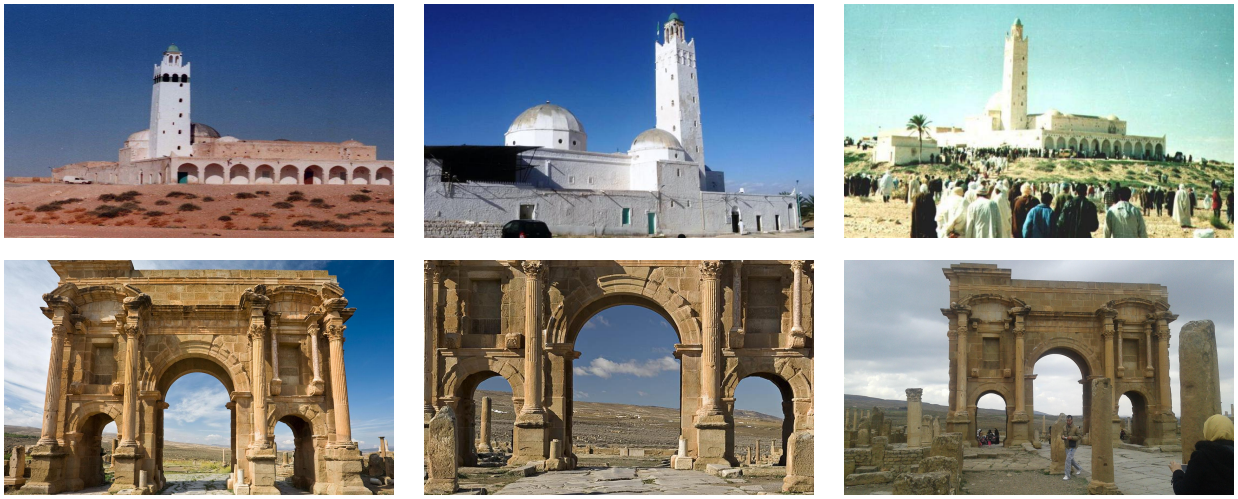


Figure 5.4: Internet image collection of Timgad Roman Ruins and Masjid Sidi-Khaled. Different camera poses and focal length, also different lighting condition gives strong variation appearance.

called voxels. Others use depth maps, where the scene is represented by the euclidean distance between a point on the object surface and the center of the camera projection. Moreover, multi-view methods that use this scene description to recover the $3D$ shape of any object are referred to as multi-view depth map estimation (MVDE) by [136] for more detail please refer to Chapter 3.

MVDE methods can be further categorized into two classes. First, suppose that we are living in a discrete space. Hence, we uses the patch-based reconstruction methods [28] which are equipped by the local optimization approach to refining the position and normals of each patch independently. Therefore, it is extremely successful in recovering thin details presented on object's surface. Meanwhile, homogeneous, occluded and non-Lambertian surfaces in general always force these approaches to achieve a low rate of completeness and accuracy.

However, if we suppose that we are living in a continuous space, we should use variational methods [61]. Such methods optimize over all the object surface areas while adding some sort of regularization term to address the textureless regions which the classical photo-consistency measurements could not deal with. However, such methods pay extras attention when designing the objective function. Since all the variational methods require a derivative based approach such as conjugate gradient method to recover the $3D$ shape. Sometimes such approach proved to be hard or even impossible according to the used energy function. Besides, any derivative optimization approach requires a proper initialization otherwise the solution found maybe trapped in a local optimum. Hence, such shortcomings motivated us to propose a derivative-free patch based matching approach.

In fact, our human brain is naturally equipped with a powerful and versatile meta-heuristic solver, add to that the widespread associative inputs from all other senses which act as the perfect initial guesses.The produced solutions to any visual struggles are guaranteed to be optimal. Inspired by the ability of our mind to solve vision problems with a perfect accuracy based solely on heuristic, we address the problem of computing the depth and normal, and surface roughness maps of a given scene under a natural but unknown illumination from multiple input photographs utilizing particle swarm optimization to maximize a sophisticated photo-consistency function.

This chapter presents a novel method that outputs a quasi dense depth maps covering the objects visible in the input photographs. In particular, we reconsider one of the classical multi-view stereo approach [36], where the stereo matching process is considered as local optimization problem, and argue that photometric regularization is a key aspect for a stereo approach that works on a dataset with variance in the capturing conditions. We think that the popular Winer Take ALL (WTL) optimization approach is too much relaxed, and it is mandatory to take into consideration a proper reflectance model which represent more complex surfaces to augment the chances of matching, hence achieving more complete reconstruction.

Theoretically speaking our idea sounds simple. Yet, its practical implementation requires a deep thinking at the pixel level in order to handle the previously mentioned problems. As a consequence, our method departs from the original works in multiple places. In particular, for each image in the photograph set, we need to select multiple image references for stereo computation. In contrast to the original work of Goesele et al.[36], we do not enforce any sort of a local selection at each pixel. Instead, we use the whole cluster selected globally from the input images to control the swarm during the optimization process and force it to find the correct matches. Hence, we enhance our method resilience to any appearance differences presented in the initial input photographs. Finally the general benefits of our method compared to the literature and the main contributions of this chapter can be listed as follow:

- Derivative free optimization.
- It can be easily implemented and integrated with the Multi-View Reconstruction Environment software [24].
- The method output dense depth which results in an accurate and complete reconstruction on par with best preforming state-of-art methods.
- Light direction per view and surface roughness along with its orientation per point can be provided if needed.

5.2 Framework Description

The fundamental new idea underlining this work is to attempt to compute geometry from a large set of online images while taking into consideration the non-Lambertian surface property. Certainly, there is no shortage of disagreement diversity in appearance within such type of photographs. This can be traced back to numerous reasons related to image acquisition conditions. For example Figure.5.4 shows two different scenes, with variable capturing conditions, for instance, the view point and camera parameters for each photographs, also different time of day and weather which lead to multiple illumination outcomes.

Given the advantages of the standard multi-view stereo approaches [17, 34, 115] outlined in the previous chapters, it is quite predictable that this type of dataset poses new problems that make computing correspondence during the matching process significantly harder.This part of our dissertation focuses on solving the following problems where we rely purely on robust similarity measurement and meta-heuristic optimization :

- Rough surfaces
- Shadows and inter-reflections
- homogeneous regions

The proposed solution to the previously mentioned problem is heavily inspired by Goesele's work [36], yet not fully identical. Our method consists of multiple stages assuming that we started with non-calibrated photographs as illustrated in the Figure 5.5. First, the pre-processing stage, in this phase using structure from Motion (SFM) algorithm [106] we compute the camera parameters along with the scale-invariant feature transform (SIFT)[73], the latter will be used on the later stages during the optimization process (Sect. 5.4).

It is possible also to compute the SIFT feature directly if the camera parameters are already given. Next and alwayse in the pre-processing stage we estimate the scene illumination parameters (*see* Sect. 5.3.4), in particular, the lighting direction for each view. In order to be able to compute such parameters we suppose that the scene is illuminated only by direct lighting in each photograph.

Next, The reconstruction stage. In here, we perform a simple images selection algorithm which is described in Sect. 5.4.1 for each reference view in order to ensure a subset of good support images that will be used during the stereo matching process. After that a depth map is reconstructed for every input images. Finally

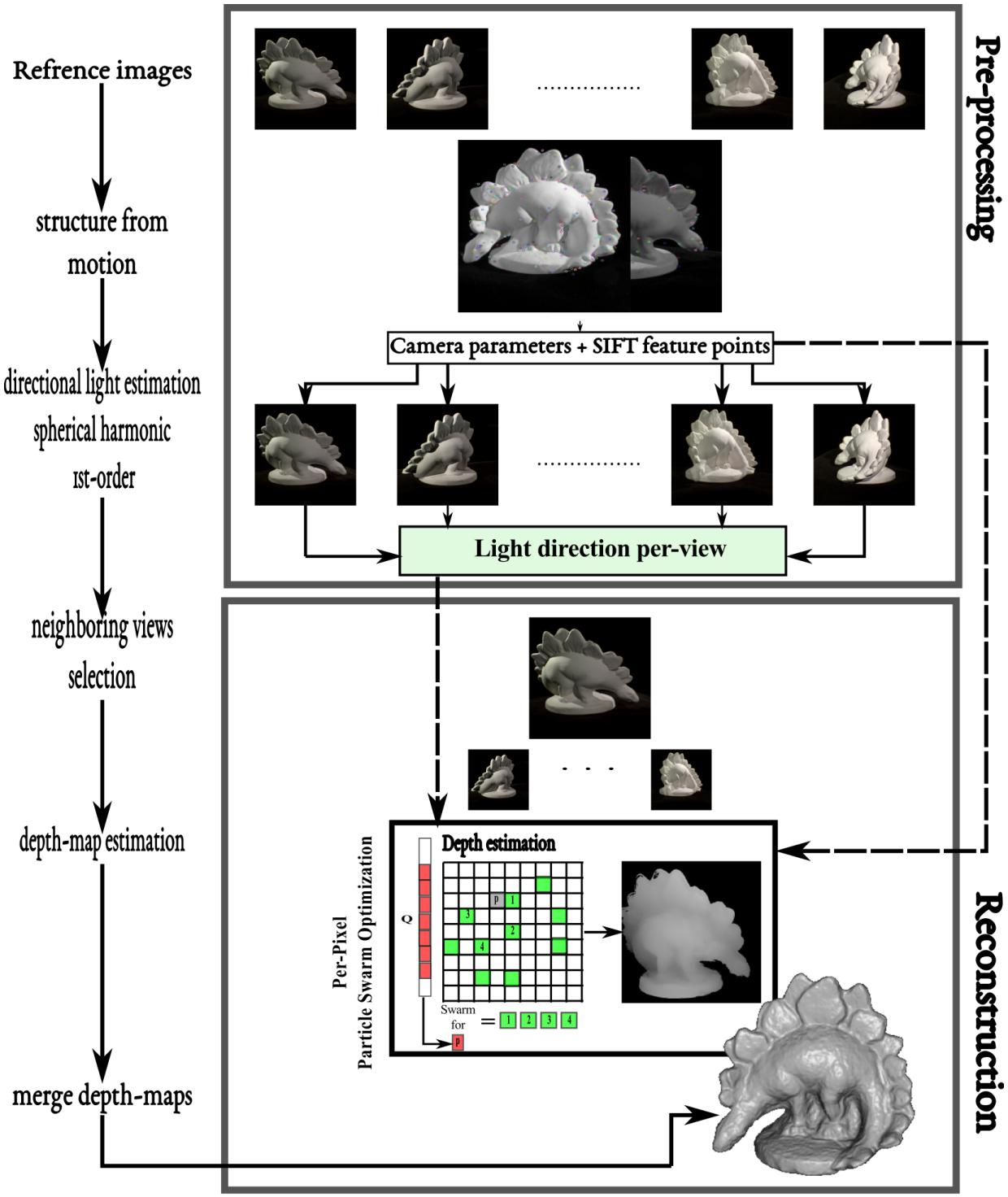


Figure 5.5: Flow chart of the proposed method.

these depth maps will be merged into a single $3D$ shape using the screened poisson surface reconstruction [51].

The most important process in the reconstruction stage is the matching process. It is here where we contribute the most in order to achieve a heigh quality $3D$ models that can be used efficiently in any virtual reality experience. As we mentioned in the above analysis (*see* Sect. 5.1.3), standard matching methods need expensive energy minimization techniques to create a dense reconstruction, moreover, it estimate depth maps with holes near silhouettes, highlights and in textureless or occluded regions.

To solve this optimization problem a derivative based approach must be used. For example, conjugate gradient method [28] which requires the partial derivatives [36] or the Jacobian matrix [61, 110] of the matching functions were exploited in order to recover the true depths value from the input photographs. In fact, it is hard and sometimes even impossible to derive for most of the designed matching functions due to their complexity. On top of that, all the derivative optimization requires a proper initialization in order to numerically solve the problem otherwise the solution maybe lost in the process.

The proposed stereo matching technique in this dissertation is built upon minimizing an energy function as it will be described in the Sect. 5.3.3. The minimization process was done via a well know optimization algorithm called the *Particle Swarm Optimization* (PSO) (*see.* Sect. 5.4). The reason behind such a choice, is the simplicity of the algorithm and it does not require the problem to be differentiable which is exactly what we wanted. Beside, the algorithm is an analogy of a biological entities searching for the best position in the euclidean space and this is exactly what the $3D$ reconstruction does, basically we are searching for the best position that is consistent with the input images.

We perform the stereo matching at each pixel in the reference view starting from the initial estimates provided by SIFT feature points (Sect. 5.4.2) and expand iteratively (Sect. 5.4.3). Note that it is not necessary to revisit an already treated pixel because we assume that the solution given by the matching process is definitive. To summarize, our method work on pixel level and has no explicit regularization unlike other approaches. In particular, our photometric model is designed using oren-nayar BRDF which is explained in the next section. This model basically tries to match pixels while assuming that they belong to a rough surface (nature surfaces are non-lambertion but they are rough). In fact, we are searching among the possible depth,normal and roughness values for a solution that give a correct stereo match.

5.3 Key Elements of the Proposed Method

Before detailing our algorithm in Sect. 5.4, we define here the patch models that will be used in our reconstructions process, as well as the photometric model used throughout to represent the surfaces captured by the input images. We also introduce a sophisticated similarity measurement that acts as the building block of our optimization approach, meanwhile, a method to estimate light direction from the input photographs is given in detail.

First of all, consider a scene captured by N photographs, whose intensity function is notated as $I_k(\mathbf{u}_k)$ where $\mathbf{u}_k = (i_k, j_k)$ is a pixel coordinate on image k . For the rest of this chapter, the problem of surface reconstruction is pictured as search problem for the optimal depth, orientation and roughness values on one of these images i.e. the Reference View M. Each one of those reference views is supported by cluster of H neighbouring views.

5.3.1 Geometric Patch Model

A small planar patch $\Pi(\mathbf{x}, \mathbf{n})$ is a rectangle represented by an $n \times n$ pixel window in a reference view.

$$\Pi(\mathbf{x}, \mathbf{n}) \Leftrightarrow \mathbf{n} \cdot \grave{\mathbf{x}} + d = 0 \tag{5.1}$$

where $\mathbf{n}$ is the planar orientation vector, while $\mathbf{x}$ describes the $3D$ position that is projecting into the image point $\mathbf{u}_M$ of the reference view M. Let us denote the entity called the position vector of the three-dimensional point $\mathbf{x}$ as the vector $\grave{\mathbf{x}} = (x, y, z)^T$ which starts from the origin of the coordinate system and ending at $\mathbf{x}$. Finally, we call d a scalar that regularize the planar depth *i.e* distance from the origin.

At this stage, in order to represent the $3D$ position of each $2D$ element $\mathbf{u}_M$ in $n \times n$ pixel window via a point $\mathbf{x}$ situated at distance $h(\mathbf{u}_M)$ from $\mathbf{c}_M$ the center of projection of the reference view, we use the following equation:

$$\mathbf{x} = \mathbf{c}_M + \mathbf{r}(\mathbf{u}_M) \centerdot h(\mathbf{u}_M) \tag{5.2}$$

This equation uses $\mathbf{r}(\mathbf{u}_M)$ as the normalized ray direction through a given pixel $\mathbf{u}_M$ starting from the center of projection $\mathbf{c}_M$. In fact, equation Eq. (5.2) is approximating a planar patch which is visible in pixel window at a given view. Despite the simplicity of this representation, yet it is still not compatible with our optimization process. The reason behind this conclusion is that each point that belongs to the planar technically is represented via its own depth value $h(\mathbf{u}_M)$. Such a case will certainly increases the number of

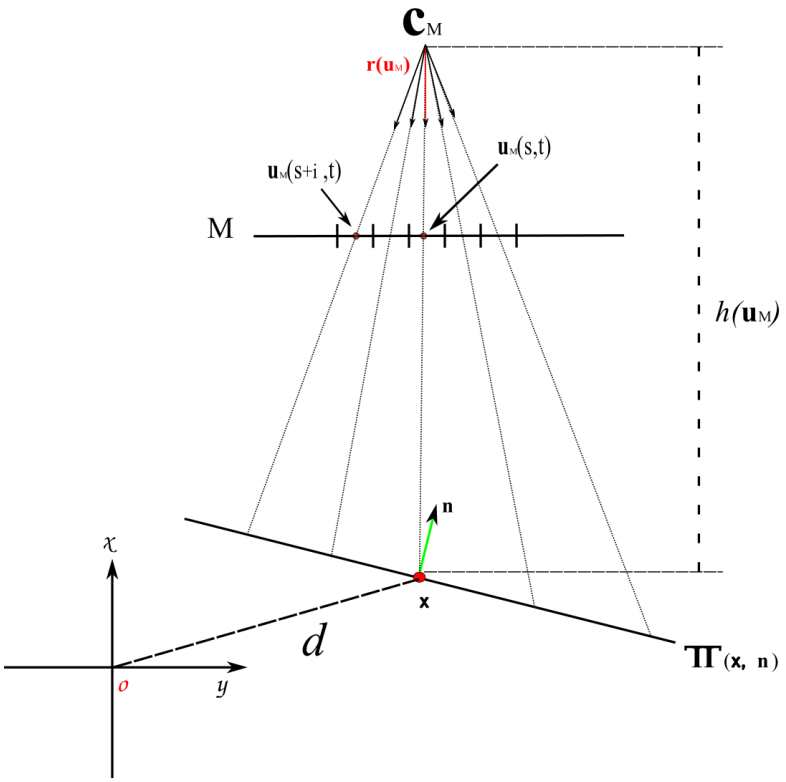


Figure 5.6: Stereo matching geometric configuration: Points $\mathbf{x}$ located on the planar Π at distance $h(\mathbf{u}_M)$ along viewing ray $\mathbf{r}(\mathbf{u}_M)$. $\mathbf{u}_M = (s,t)$ is central pixel and $\mathbf{u}_M = (s+i,t)$ or $\mathbf{u}_M = (s,t+j)$ are the rest.

parameters to estimate during the process of optimization. Consequently, this will lead to the computational effort increasing along with risking the stability of the optimization.

Now in order to fix such problem we tried to find an equivalent representation that suits our needs. This is easily done by using the equation Eq. (5.1) and rewrite the equation Eq. (5.2) in term of the planar patch orientation vector $\mathbf{n}$ and its depth regulator d. Our geometric patch model can be seen in Figure 5.6, where each entitle described in the previous analysis is clearly illustrated.

$$\mathbf{x} = \mathbf{c}_M + \mathbf{r}(\mathbf{u}_M) \cdot [\frac{\mathbf{n} \cdot \grave{\mathbf{c}}_M + d}{\mathbf{n} \cdot \mathbf{r}(\mathbf{u}_M)}] \tag{5.3}$$

We conclude that the given representation illustrated by equation Eq. (5.3) is advanced and accurate which makes it more suitable for PSO algorithm unlike what is presented in [36]. Moreover, it allows the usage of a bigger pixel window for more robust matching since all the point of that planar share the same orientation and the depth regulator d . We follow the standard definition of pinhole camera model [37] and establish the projecting function $\mathcal{P}_k(\mathbf{x}, \mathbf{K}_k, \mathbf{R}_k)$ in order to determine the corresponding location of a given patch in a neighboring view k according to the camera calibration matrix $\mathbf{K}_k$ and it rotation $\mathbf{R}_k$. This process simply

will create a matching candidate for the initial window of pixels situated in the reference view M.

5.3.2 Photometric Model

In this subsection we are going to discuss the image model used as a key element for the reconstruction. As we mentioned in the introductory chapter (*see* chapter. 1), we aim to reconstruct a realistic object that can be integrated easily in a virtual world. Hence, it is necessary for our photometric model to represent this world in an accurate way.

In fact, it is known that reflectance is a physical property which describes how surfaces reflect incident light Furthermore, the appearance of an object is determined by surface geometry, reflectance, and illumination. We characterize these different material properties mathematically based on BRDF which was detailed in Chapter 2. In general, with the absence of subsurface light transport, all incoming light at an object's surface is either reflected or absorbed at the point of incidence.

Looking at the computer graphics literature, a large range of BRDF models can be used [105] to simulate the light interaction with the natural surface of the objects. In fact, such models can be physically-based BRDF's, hence, these models accurately simulate light scattering by applying nature physics laws. On the other hand, some models are considered to be observational BRDF's, where these models aim is to provide an easy description specifically designed to imitate a given type of reflection.

For instance, the most simple one is Lambert model, the simplicity of this model coupled with its effectiveness, led it to be used extensively by multi-view stereo and shape from shading approaches [61, 108, 29] in order to recover the geometry from the observed scene. However, in reality objects are not Lambertian, which make this model some how inadequate approximation especially if we are seeking an accurate reconstruction of the environment which will be integrated in a virtual experience as we mentioned earlier (*see* Chapter 1). We took this observation into consideration and we built upon it.

In the proposed photometric model, we assumes that the surfaces are not fully Lambertian, hence another complex radiometric phenomena such as masking, shadowing, and interreflections could be simulated during the reconstruction. One of the physically-based BRDF function that caught our attention was Oren-Nayar BRDF model [89, 90]. To emphasize, we think that such a model is particularly fitting to reconstruct naturalistic real-world scenes and outdoor environment.

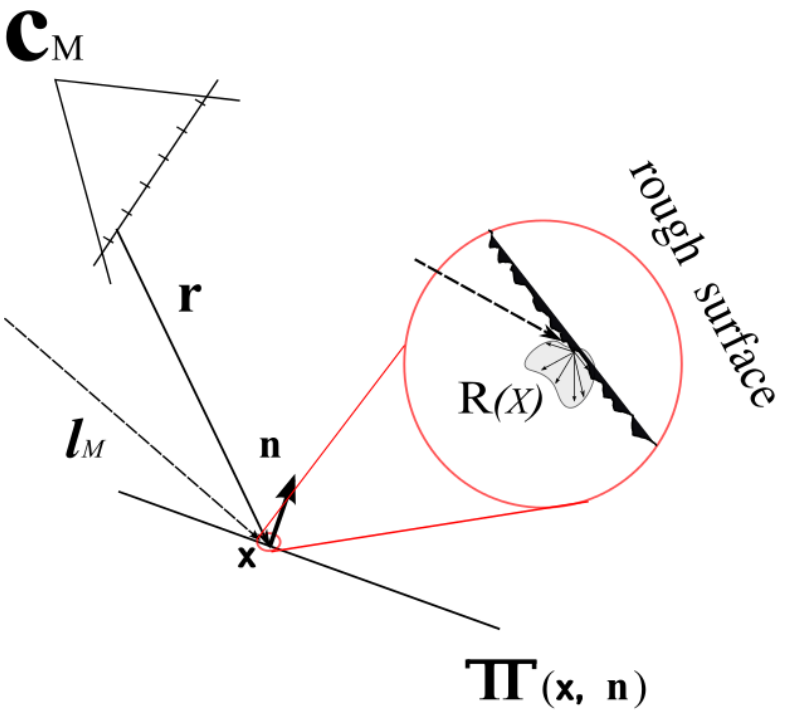


Figure 5.7: Stereo matching photometric configuration : a rough surface represents a small area in the planar patch, this area is illuminated by directional light.

To enforce this model into our method, we suppose that each view k is illuminated via *direct illumination* in order to reduce the amount of computations. In a scenario like that, the light source is assumed to be positioned far away i.e at infinity, hence, it is directional. Figure 5.7 shows a detailed illustration for our model, For instance, we denoted the light direction of a given view k with l_k, meanwhile the radiance of that light source is called E_k. Also in the figure we can see the out going energy reflected from a rough planar surface. We call the out going energy of any point $\mathbf{x}$ belong to the planar patch by radiance and it is denoted as $R(\mathbf{x})$.

Finally according to *energy conservation theory* (ECT) the irradiance at some point in the two-dimensional image plane $I_k(\mathbf{u}_k)$ is defined by the amount of energy radiated from the corresponding re-projected point on the scene $R(\mathbf{x})$ according to the direction of viewing ray. In other words the outgoing point radiance is equal to ingoing image irradiance, thus it is possible to write the following formula: $R(\mathbf{x}) = I_k(\mathbf{u}_k)$. This simple equality would be used later on to determine the final equation of our photometric model which will prove useful in the matching process of Sect. 5.4.3.

We emphasize that the proposed photometric model act as guess tool. In fact, this model does essentially predicts the color of a given pixel $\mathbf{u}_{ref} = (i,j)$ that belongs to a given patch centered around the pixel

$\mathbf{u}_{ref} = (s,t)$ when the depth and orientation of that patch changes.

After given the essential building blocks for our photometric model, we are now ready for providing a case study from which we deduce the final equation of the model. This is done by first studying the Lambertian case and then generalize the obtained knowledge for rough surfaces.

Lambert Planar Patch

We mentioned earlier in this chapter and in the second Chapter 2 that the Lambertian surface is an empirical BRDF model designed to be used as simple shading model. Moreover, multi-view stereo and shape from shading benefit from this model in order to recover a fair amount of $3D$ geometry. In this part of the dissertation we decide to study this model and build upon it.

Assuming that the planar patch is described by Lambertian surface. Basically means the radiance of each point fulfils the Lambert law. So, in this case the amount of the lighting energy that radiate from an area of the scene surface depends on the surface orientation in relation the light source, thus, the object finale shading results is proportional to the cosine of the angle between the surface normal and the light source direction. According to the information given in Chapter 2 namely equation Eq. (2.4) and the Lambertian BRDF model given by equation Eq. (2.5) we can write the mathematical description of this phenomena by the a general equation equation as follow:

$$radiance = (Energy \times \textbf{(}Direction \cdot Normal\textbf{)})/\pi \tag{5.4}$$

Notice that this equation suppose to be a general case which will be used later on to extract the photometric model. However, at this stage, it is important to know that we are working with multiple view scenario, hence, a set of photographs for the same scene. Therefore, we emphasis that the radiance entities $R(\mathbf{x})$ of the surface point $\mathbf{x}$ could have multiple value which is deduced according to the previously mentioned ECT.

$$R(\mathbf{x}) = \begin{cases} I_M(\mathbf{u}_M) & \text{if Image is refrence view .} \\ I_k(\mathbf{u}_k) & \text{if Image is support view .} \end{cases} \tag{5.5}$$

You can argue that the above formulation is incorrect since we already assumed that the surface is Lambertian. Yet, we confirm that the equation above holds true, the reason behind such confirmation is that the used photo collection impose on us to assume that every view has it's own lighting conditions even

in Lambert scenario. This fact lead to different radiance outputs R for the same surface point $\mathbf{x}$, and thus different image intensity I.

Now in our scenario where we have multiple photographs with multiple lighting condition (even small changes), the radiance of a $3D$ point belong to the planar patch will be described for each view mathematically using Lambert shading (equations Eq. 2.5 and Eq. 5.4):

$$R(\mathbf{x}) = \begin{cases} 1/\pi \,.\, E_M \,.\, (\mathbf{n} \cdot \mathfrak{l}_M) & \text{if Image is refrence view .} \\ 1/\pi \,.\, E_k \,.\, (\mathbf{n} \cdot \mathfrak{l}_k) & \text{if Image is support view .} \end{cases} \tag{5.6}$$

Finally, it can be seen from the above analysis, and the given equations that in a multi-view environment, the intensity of a pixel in a given view k should be rectified via a scalar entity which in this case we give the name of ε_k the *Lambertian correction factor*. In order to increase the ability to match images taken under different conditions, This constant scales the color of any pixel in a neighbour view so that it will be equal to its corresponding pixel in the reference view. By taking a closer look to this so called Lambertian factor we can see that it is view dependent, which means a constant illumination over the planar patch surface, while it changes from one input view to another. The mathematical description is given in the following equation:

$$\varepsilon_k = E_M \,.\, (\mathbf{n} \cdot \mathfrak{l}_M) / E_k \,.\, (\mathbf{n} \cdot \mathfrak{l}_k) \tag{5.7}$$

In fact, the proposed scalar shows some interesting properties interesting and can be used in the optimization process, These properties can be seen in both the numerator and denominator when one of them become equal to zero. The geometrical explanation for this is given as follow. First if the numerator is equal to value of 0, it indicates that the observed surface point is not even visible in the reference view thus it should ignored and not be considered in the reconstruction. On the other hand, when the denominator equal to value of 0, then we conclude that the geometry is not also visible however this time in the neighbouring view k. At the end the study case of a Lambertian surface gave the final *Lambertian photometric model* that can be used in the optimization stage. This model is expressed via the following equation:

$$I_M(\mathbf{u}_M) = \varepsilon_k \,.\, I_k(\mathcal{P}_k(\mathbf{x}, \mathbf{K}_k, \mathbf{R}_k)) \tag{5.8}$$

Such model provides some how a sufficient invariance to yield acceptable results on a some scenes. However, like we discussed earlier, this model decreases the ability to match images of an online collection resulting in depth maps that contains holes thus a non-complete reconstruction which effect the virtual

reality experience greatly. In particular, this model will fails for instance when the shading changes in unexpected way within the planar patch itself like at shadow boundaries. Even more, when the patch contains surface with a specular highlight. In the next subsection we propose an new model that solves this dilemma by taking into consideration the nature of the surface reconstructed and enhances equation Eq. (5.8). Combine this model with the usage of the particle swarm optimization (*see* Sect. 5.4) during the matching process and the obtained results will be satisfying despite the variation in appearance presented in photographs used.

Rough Planar Patch

In this part of the dissertation, we aim to generalize the above proposed concept of *Lambertian photometric model* via the simulation of one of the reflectance effects. We choose the *Oren & Nayar* model [89, 90] due to its nature, moreover, we think that such model is particularly fitting to reconstruct naturalistic real-world scenes and outdoor environment. In fact, this BRDF model is considered as faithful interpretation for quasi Lambertian diffuse martial also known as rough surfaces. Furthermore, the energy radiated from this matte surfaces increases gradually while the viewer position moves toward the illumination source. As result the shading of such rough surface change relative to the surface normal along with the viewing direction.

Additionally, we claim that such BRDF model is widely acceptable for community photo collection dataset especially for outdoor environment. We give the bidirectional reflectance distribution function of the *Oren & Nayar* model [89, 90] as follow: given 3D point $\mathbf{y}$ with its normal vector $\mathbf{n}_y$ at a given light and viewing direction vectors $\mathbf{l}_y$, $\mathbf{r}_y$ respectively the BRDF of a rough surface is equal to:

$$f(y,\mathbf{l}_y,\mathbf{r}_y) = \frac{\rho}{\pi} \cdot (A + B \cdot \cos(\gamma) \cdot \sin(\alpha) \cdot \tan(\beta)) \tag{5.9}$$

This formulation is an analogue to the general case of microgeometry presented in the second chapter (*see* Chapter 2, Sect. 2.3.1) by the equation Eq. 2.6. In fact, The object roughness is modelled by assuming that the surface consists of many micro-facets each tow facets form V-shaped symmetric cavities. The surfaces of these V-cavities, assumed to have Lambertian reflection. it is also assumed that each pixel in the final rendered image can see a surface patch which it self contain a large number of small facets that have infinity small area, As a result the roughness of the surface is described using a probability function which describe the distribution of facet slopes. To emphasize, the variable ρ is the scene albedo which does not change per view, also the Gaussian probability distribution of the rough surface microgeometry

is represented in this model via the two special factors described in the following equation:

$$A = 1 - 0.5 \cdot (\frac{\sigma^2}{\sigma^2 + 0.33}) \quad , \quad B = 0.45 \cdot (\frac{\sigma^2}{\sigma^2 + 0.09}) \tag{5.10}$$

As the equation Eq. 5.10 shows, the scalar σ represent the amount of roughness for a given surface area, meanwhile, the multiple angles α, β and γ presented in the BRDF equation Eq. 5.9 are the result of dot products between light and view direction vectors with the surface normal.

At this stage of our work, we would like to assume that each point **y** belong to our rough planar patch surface follows *Oren & Nayar* presented model. This indicate as we said earlier that given an incident light and view directions, the total radiance is the integral of micro-facet reflectance values and the radiance due to the bounces between them (BRDF model). This physical interaction between the light and a rough surface is described by the following equation:

$$R(\mathbf{y}) = f(y, \mathbf{l}_y, \mathbf{r}_y) \times Energy \times (\mathbf{n}_y \cdot \mathbf{l}_y). \tag{5.11}$$

To formulate our final photometric mathematical model, let us assume that each point **x** belong to the defined rough planar patch visible in *n*x*n* pixel window. and follow the exact same steps of the simple Lambertian case described in the previous entry. Hence, the ECT and equation Eq. (5.11) strongly suggest that the outgoing energy from the rough point in multiple view framework can have multiple representation.

First in case of the reference view M:

$$R(\mathbf{x}) = \frac{\rho_M}{\pi} \cdot E_M \cdot (\mathbf{n} \cdot \mathfrak{l}_M) \cdot [A + B \cdot \Theta_M(\mathbf{n}, \mathfrak{l}_M, \mathbf{r}(\mathbf{u}_M))] \tag{5.12}$$

and secondly in case of a given support view k from list of neighbour images:

$$R(\mathbf{x}) = \frac{\rho_k}{\pi} \cdot E_k \cdot (\mathbf{n} \cdot \mathfrak{l}_k) \cdot [A + B \cdot \Theta_k(\mathbf{n}, \mathfrak{l}_k, \mathbf{r}(\mathbf{u}_k))] \tag{5.13}$$

with

$$\Theta(\mathbf{n}, \mathfrak{l}, \mathbf{r}) = \cos(\gamma) \cdot \sin(\alpha) \cdot \tan(\beta) \tag{5.14}$$

Obviously, if scene captured by large cluster of photographs such as community photo collection, the capturing conditions will be different as we mentioned before. Hence, a point visible in multiple views yields varying values of its irradiance energy. Following the same principle which was applied in the

Lambertian in the previous analyse, our approach corrects the intensity of a pixel in a given view k to try to achieve a higher matching rate in the reference view. For this reason, the pixel value is scaled according to two different factors. Firstly, the *Lambertian correction factor* ε_k, the second factor is ξ_k the *non Lambertian correction factor*. The later is expressed using the equation Eq. 5.15 and it was deduced via equation Eq. 5.13 and equation Eq. 5.14. It can be seen from this formula that this factor is in fact view dependent and indeed can express the illumination change within the planar patch do to the presence of viewing rays in the formula.

$$\xi_k(\mathbf{r}(\mathbf{u}_M)),\mathbf{r}(\mathbf{u}_k))) = \frac{[A + B \cdot \Theta_M(\mathbf{n}, \mathfrak{l}_M, \mathbf{r}(\mathbf{u}_M))]}{[A + B \cdot \Theta_k(\mathbf{n}, \mathfrak{l}_k, \mathbf{r}(\mathbf{u}_k))]} \tag{5.15}$$

Note that the two-dimensional image point $\mathbf{u}_k$ is the after math of projecting the $3D$ point $\mathbf{x}$ into a support view k using the projection function $\mathcal{P}_k(\mathbf{x}, \mathbf{K}_k, \mathbf{R}_k))$. Finally we could express our *General photometric model* which will be used in the optimization (*see* Sect. 5.4) as follow:

$$I_M(\mathbf{u}_M) = I_k(\mathbf{u}_k) \cdot \varepsilon_k \cdot \xi_k(\mathbf{r}(\mathbf{u}_M), \mathbf{r}(\mathbf{u}_k)) \tag{5.16}$$

After establishing links between our proposed photometric and geometric patch models in this subsections. It is necessary to identify the unknowns presented in the final model (*see* equation Eq. 5.16) that will be used in the optimization process. Starting first by mains variables which are d the patch depth regulator and the normal vector $\mathbf{n}$, In fact, in order to deduce the patch depth by combining this two we can according to equation Eq. (5.3).

The next variable is the roughness factor σ which determines how rough the planar patch, In fact, all of this this five parameter are unknown only per patch. Finally, the *Lambertian correction factor*, ε_k which is unknown per view k. Furthermore, it represent implicitly the light source radiance along with the surface albedo, as a result we can safely declare that our approach is totally invariant to spatially changing albedo without explicitly modeling it in the optimization.

5.3.3 Similarity Measurement

In the third chapter of this thesis (*see* Chapter 3) we talked about the concept of multi-view photometric consistency, also known as photo-consistency in short. It is considered the main mechanism used in any multi-view stereo algorithm. We like to remind the reader that multi-view photo-consistency quantifies the degree of correspondence or consistency between a set of input photographs and all the ingredients

that take part in their image formation such as illumination, materials, and $3D$ geometry of the scene being captured.

Certainly, multiple mathematical model to represent this metrics were proposed in the literature of computer vision. In fact, each one of these models comes with the its trades. Take a look for instance to the Sum of Absolute Differences (SAD), such measurement is extensively used in almost all of the stereo reconstruction approaches. The reason to such thing can be traced to its beneficial trades namely its rapidity in term of computing time, and can be easily adapted for parallel implementation. Moreover, such metric proves it robustness when we expect to have no variation in the matched images [31, 83].

On the other side, some approaches are based on different metrics such as Census or Normalized Cross-Correlation (NCC) when a noticeable bias are present across the input set of photographs. In fact, there is no shortage of disagreement within the community that metrics like census and NCC will performs well when the matched photographs contain some diversity in the light condition and the surface martial type [34, 28]. We like to refer interested readers to [44] for a detailed review on the state-of-art of the photo consistency measurement.

In this subsection, we aim to make use of the above geometric and photometric models in order to create our photo-consistency measurement. Note that in this work we consider the multi-view stereo as a constrained search problem, where multi-view photo-consistency is optimized as a function of depth orientation and surface roughness. In the proposed work, we followed Mei et al. [121] suggestions. The authors claim that the combination of two well known photo-consistency metrics will lead to better matching result since one metric can hide the other weak point and vice versa.

Despite that it is tempting to use SAD since equation Eq. (5.16) predicts the pixel intensity and it counts for any small bias, hence, it is safe to assume that the matched images are theoretically not different. However, we think that the current setup is not assuring, thus to bypass the mismatches between photographs, we change our photo-consistency measurement by adding Census measurement which invariant to changes especially around depth boundary.

In the following, we will explain each part of our multi-view photo-consistent measurement formulation starting by $\mathfrak{E}_{sad}(\mathbf{n}, d)$ and then the control part $\mathfrak{E}_{cenus}(\mathbf{n}, d)$. In order to simplify the process we decide to

rewrite the equation Eq. (5.16) as the follow simple equality:

$$I_M(\mathbf{u}_M) = P_k(\mathbf{u}_k) \tag{5.17}$$

In here the entity $P_k(\mathbf{u}_k)$ simply represent the predicted intensity of a given pixel in neighbourhood view. At this stage , having considered all the necessary presupposition, it is also reasonable to look at how in the multi-view scenario the photo-consistency is computed. In fact, it is simply estimated by averaging the similarity measurement C between the reference view and all its H neighbors photographs, as it is shown in the following equation:

$$\mathfrak{E}(\mathbf{n}, d) = \frac{1}{H-1} \cdot \sum_{k=1}^{H-1} \rho(C, \lambda) \tag{5.18}$$

Now let us define the first part of our similarity measurement C_{sad} as the sum of absolute differences between each pixel $\mathbf{u}_M^i$ of an $n \times n$ pixel window in reference view and value of it's predicted correspondence $\mathbf{u}_k^i$ in the k view. The equation below show the simple mathematical formulation of this measurement.

$$C_{sad} = \sum_{i=1}^{n \times n} |I_M(\mathbf{u}_M^i) - P_k(\mathbf{u}_k^i)| \tag{5.19}$$

On the other hand the second part of our similarity measurement is C_{census} which refers to the Census measurement. To emphasis, this photo-consistency model is different from that of the above one, where it does not benefit from the usage of pixel intensity values directly, however, it uses the intensity relative order in the two-dimensional pixel window to construct what is known as the bit string.

$$C_{census} = |S(I_M(\mathbf{u}_M) - S(I_k(\mathbf{u}_k)| \tag{5.20}$$

In the equation Eq. 5.20 the function $S(.)$ computes the bit string. This process is done by comparing the central pixel u to its neighbors in pixel window which lead two have two cases First if the original value is lesser than its neighbor the resulted bit is equal to a value 0. However if the value is bigger or equal than the bit will receive a value of 1.

Both values of photo consistency computed using the presented formulas in this section are rarely used in multi-view stereo in later stages without normalization. Instead, they are transformed through a non-linear operation in order to normalize both values to the same space. This in fact already proved beneficial in [79] work, thus we follow the same process and we choose $\rho(C, \lambda)$ as exponential normalization function

Algorithm 4: Estimating normals for the initial point cloud.

Input: 3D point cloud **a**, Camera Position **c**
Output: Normals Vectors n

forall the *points* $\mathbf{a}_i$ *in the cloud* **do**
  $n_i \longleftarrow 0$
  $counter \longleftarrow 0$
  forall the *Camera* k *in the input photographs* N **do**
    if *Point* $\mathbf{a}_i$ *is visible in view* k **then**
      $n_i \longleftarrow normalized(\mathbf{a}_i\mathbf{c}_k)$
  $n_i \longleftarrow n_i/counter$

for SAD or Census. In particular, the value of λ act as a tuning parameter and it gives an easy control over outlier influence during the matching process.

$$\rho(C,\lambda) = 1 - \exp -\frac{C}{\lambda} \tag{5.21}$$

Finally, we combine all the previous equations, in order to formulate our photo-consistency measurement which will be used as the final objective function $O_{final}(\mathbf{n},d)$ to be evaluated and minimized in during optimization process The equation below summarizes all the previously mentioned information:

$$O_{final}(\mathbf{n},d) = \mathfrak{E}_{sad}(\mathbf{n},d) + \mathfrak{E}_{census}(\mathbf{n},d) \tag{5.22}$$

5.3.4 Estimating Light Direction

In order for us to compute the energy function presented in the equation Eq. 5.22, an important element must be computed first. Namely the directional vector of the light source. It is an important element used to compute the non-Lambertian factor as illustrated in the equation Eq. 5.15. Our formulation is developed based on the assumption of the scene been illuminated only by infinite light source, which gave us a directional light per view $\mathfrak{l}_k$. The shading is considered to follow direct illumination model to reduce complexity

Estimating these lighting parameters is done before the optimization process. In fact, in the post processing stage (*see* Figure 5.5), after extracting the SIFT features of each view, using the inverse projection matrix to re-project the $2D$ features into $3D$ points $\mathbf{a}_i$. The result of this process is a point cloud which we could use to estimate the light direction. As a matter of fact, it is also necessary to estimate initial normal vectors n_i for each of this point, the algorithm 4 shows how these normal were computed.

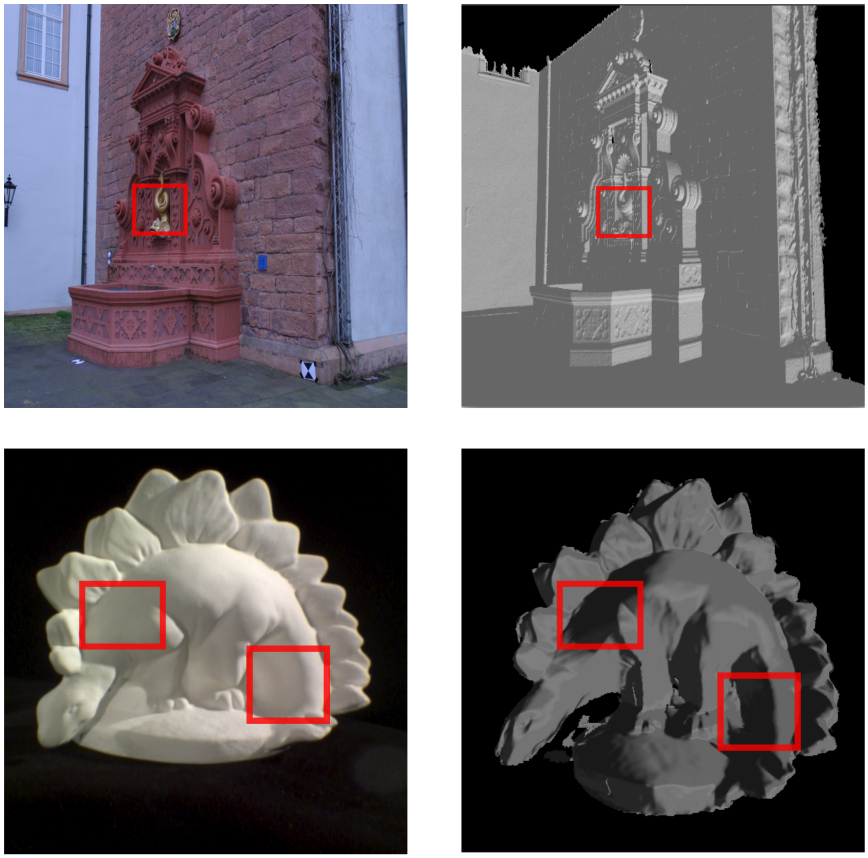


Figure 5.8: Directional light estimation. The left column represent a given view images of two diffrent scenes (Fountain and platar dino) which we want to estimate its lighting conditions. On the other hand, the right column represent the 3*D* ground truth geomtery illuminated using our estimated lighting direction

Lot of works in the literatures tries to integrate the inverse rendering techniques into the process of 3*D* reconstruction. For instance, the work [138] where they create an improved lighting model in each iterations during the process of reconstruction. In contrast we cannot improve on our lighting parameters during the optimization as these initial parameters are satisfactory and the lighting condition we supposed in the early stages does not need any complicated computations. At this stage, a surface reflectance is approximated using the 2*nd* order of Spherical Harmonics function (SH) [138], however as we mentioned earlier, the lighting conditions are supposed to be under local illumination. Hence, the first order spherical harmonics which based on four basis functions can successfully approximate the incoming radiance, the equation below show the mathematical formulation of the posed problem.

$$R(\mathbf{a}_i) = y_0(n_i) \cdot \mathfrak{f}_0 + y_1(n_i) \cdot \mathfrak{f}_1 + y_2(n_i) \cdot \mathfrak{f}_2 + y_3(n_i) \cdot \mathfrak{f}_3 \tag{5.23}$$

Where $y_j(.)$ are the SH basis functions and $\mathfrak{f}_j$ are the coefficients. From this equation Eq. 5.23 we construct

a simple linear least square system as follow

$$\mathfrak{l}_k = \sum_{i=1}^{Max} (R(\mathbf{a}_i) - I_k(\mathcal{P}_k(\mathbf{a}_i, \mathbf{K}_k, \mathbf{R}_k)))^2 \tag{5.24}$$

The light direction is embedded in the equation Eq. 5.24 as the vector $\mathfrak{l}_k = (f_1, f_2, f_3)$. In order for us to find the value of this vector $\mathfrak{l}_k$ we simply solve the linear least square system. We emphasis, that we followed the logic behind the work of Langguth et al. [61] which say that we should optimize the lighting condition using only a single image in order to be invariant to changing light conditions, note that unlike them the we optimize the lighting direction and not its intensity.

Figure 5.8 shows an illustration of the obtained results after following the above analysis, In fact, we applied the process of directional light estimation from a single photographs on two different scenes namely an outdoor fountain and indoor dinosaur toy. After estimating the lighting direction of the a given view for each scene, we take the resulted vector direction and plug it into a rendering engine that rephotograph the scenes using the ground truth geometry. It can bee seen that our lighting parameters mimic to an acceptable extent the original views, you can notice that we have the same shadow areas and the same illuminated spaces as it is illustrated by the red rectangles.

The next Sect. 5.4 is going to be dedicated to explain the detailed algorithm that we followed to reconstruct a dense depth maps from the input photographs. We will be using all the key element introduced in this Sect. 5.3. We emphasis, that the main idea is the smart usage of a meta-heuristic optimization technique to guaranty the robustness of our reconstruction to all the previously mentioned problem in Chapter 1.

5.4 Algorithm

Inspired by the ability of the human mind to solve vision problems with a perfect accuracy based solely on heuristic. In this section we detail a new multi-view depth map estimation (MVDE) algorithm based on the particle swarm optimization (PSO) approach. To mimic the inputs received from all senses, we use the scale-invariant feature transform (SIFT) obtained in early stages as well as the optimal solution (depth, normal and roughness) found during the optimization.

In particular, the main idea of our optimisation algorithm and how we propagate through the reference view M pixels is based on the initial SIFT features visible in that reference image. In order to estimate the depth and normal along with roughness of the surface captured by it, we create a list Q which contains

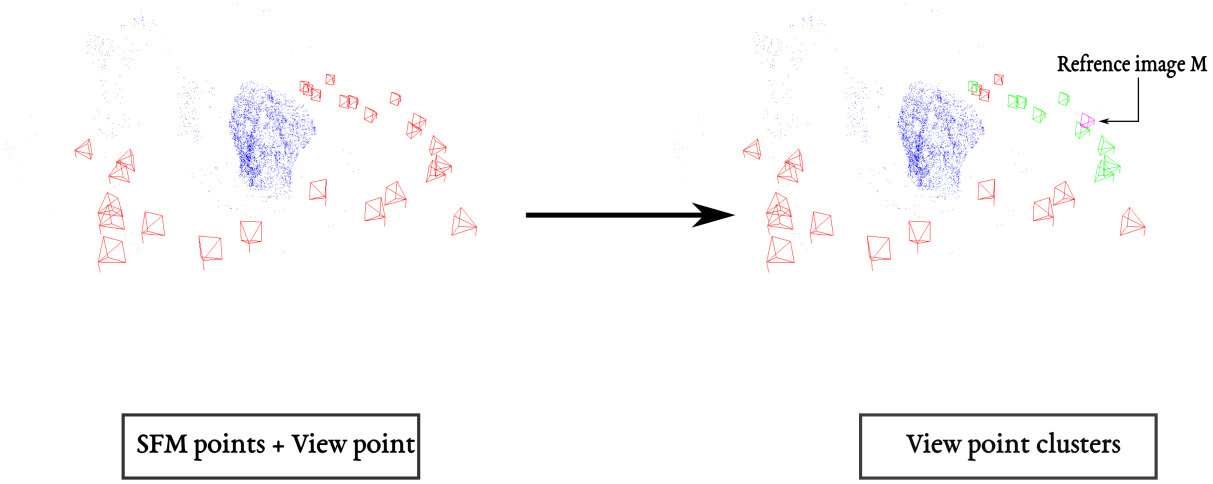


Figure 5.9: Large scale view clustering unstructed photographs of the founding stone of biskra university. Left the recovered camera view point along with all the cloud point extracted from SIFT feature. On the right a refrence view with its support view in green

candidate pixels that we want to compute their relative depth, normal and roughness. Therefore, if the candidate is successfully optimized, we add its four neighbors to the Q as new possible candidates. The algorithm used to find the optimal solution can be sketched as follows:

- Select the closet images to the reference image.
- Select the closet known solutions in a given range.
- Generate a particle swarm using these solutions.
- Search the solution space.
- Optimal solution = particle with best $O_{final}(\mathbf{n}, d)$ score.

The first step is done one time per input view. However, the rest of steps are repeated for each candidate pixel in the reference. The initial guess of the algorithm complexity is estimated to be proportional to the number of the pixel in this input image.

5.4.1 Global View Selection

For each image in the photograph set, it is fundamental to select a support images for multi view stereo computation. The selection of these stereo images is important not only for the accuracy of the stereo matching process but also for the final reconstruction colouring or texturing result. From an other perspective support view selection is considered as a coarse visibility estimation process. Stereo pair selection is a

relatively easy task for cameras in a controlled environment like the datasets used in chapter 4, but on the other hand this process needs to be carefully designed for unordered images such internet photo collection.

A good candidate support image must share the same viewing direction as the target image, while having an acceptable baseline. In another word, the baseline between the reference image M and a given image k should be neither too short to degenerate the reconstruction accuracy nor too long to have less common coverage of the scene.

Practically, there is no clear way to identify how good the selected support view is in compare to the rest of the large cluster of images. In this dissertation we adopted the global idea of Goesele et al. [36] to select eligible multiple stereo pairs. In fact, the authors work is considered the first to use view clustering for large collections of unstructured photographs. The core idea was to benefit from the SIFT features to infer a view classification. The process is done by simply counting the number of shared matches between any two views. Intuitively, the bigger the resulted number is the more confident we are of the overlap between the two views. Although the idea in essence correct, however, multi-view stereo algorithms are some how complicated and need additional information and other metrics such as baseline between the views and even the camera resolution. The authors of [36] were aware of such sensitivity thus they proposed a general ranking function to cluster a candidate view.

$$\mathfrak{g}_M(k) = \sum_{f \in F_M \bigcap F_k} w_n(f) \cdot w_s(f) \tag{5.25}$$

In particular, using equation Eq. 5.25 we aim to create a set of neighbouring images that are candidate for stereo matching process relatively to the reference view M. First, to encourage a good parallax between the candidate view k and the reference the weight function $w_n(f)$ down-weights neighbor views with a small baseline across the set of SIFT feature points observed in both the candidate and the master view. On the other hand the function $w_s(f)$ measure similarity in resolution of the previously mentioned images at the feature point f. The list of the support views is first initialized to the reference image then the best next view which maximizes the score $\mathfrak{g}_M(k)$ is iteratively added to the cluster until the maximum cluster size H is achieved.

Figure 5.9 shows the result of this process on a given dataset captured via a smart phone at the university of Mohammed Khider Biskra. after the pre-processing step which was explained in Sect. 5.2.The resulted

SIFT point along with camera parameters are used to determine the list of supporting view to given reference photographs using the above described method. Visually speaking it seems that the neighbouring views (green colour) were well chosen using this process. Hence, the quality of the reconstruction should be satisfactory (*see* Sect. 5.5)

5.4.2 Particle Swarm Initialization

We mentioned earlier that we are inspired by the human brain and how it is constantly working out enormous visual optimization problems. The reason for such organism to be able to do that is the formidable metaheuristics engine our brain equipped with along with the wide spread associative inputs from all other senses that act as the perfect initial guesses. Therefore, in this dissertation we addressed the multi-view stereo reconstruction of a given scene under a natural but unknown illumination via the usage of the particle swarm optimization [12] to minimize a sophisticated photo-consistency function. We emphasis that we adopt the constriction factor version of the original particle swarm optimization algorithm to build a dense depth map.

Let us first formulate the problem according to the well known PSO algorithm. Consequently, the dimensionality of our problem is equal to $D = 5 + (H - 1)$ i.e the number of unknowns for equation Eq. (5.16) which is considered as the objective function used in the optimisation. Consider now at this stage a swarm of particles called S. Each element of this swarm is called a particle i and it has multiple characteristics. These last are multiple vectors of size D at any given moment in time t.

The first vector is the position vector $x_i(t)$ in search space. Such vector represent the current position of a given particle in the space of solutions. Furthermore, the velocity vector $v_i(t)$ that determine how fast and at which direction the particle is moving, Finally, every particle remember its own best position in the solution space $b_i(t)$, such vector is considered as learning mechanism during the optimization problem, along with the global best position the swarm want to go to $g(t)$. This last is the final solution given by the swarm if it converge.

We are now ready for providing the process of initializing the swarm S which will act as a solver for a candidate pixel from the waiting list Q. At the beginning of the optimization of each candidate, we create a swarm with some number of particles. However, unlike any other PSO algorithm, the initialization of these particles is done via an already known solutions. For this process we followed two slightly different approaches.

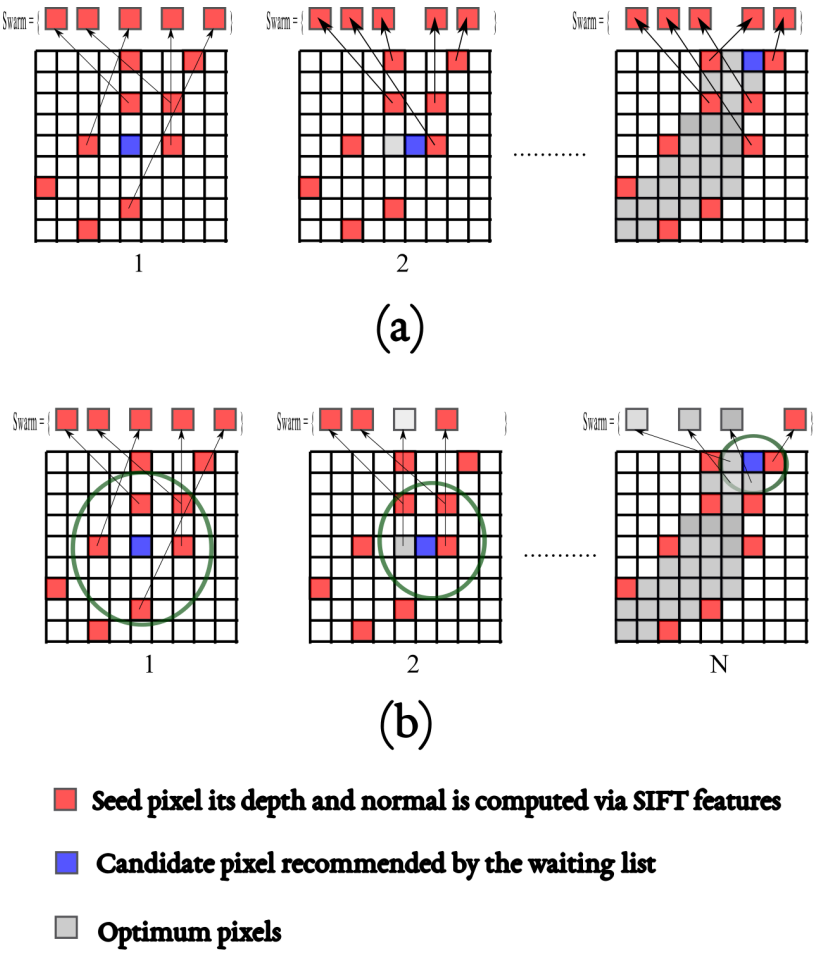


Figure 5.10: The initialization process of the swarm through out the whole reconstruction process. (a) Initialization with only SIFT feature. (b) Dynamic Initialization with any correct solution.

Features Only Approach

In this road, we create a swarm with a fixed number of particles equal to the P closest two-dimensional features to the candidate pixel as illustrated in the first image of the Figure 5.10 (a). Unlike standard PSO approaches the initial position of every particle in the swarm is initialized using existing information. First, among the SIFT feature extracted from the current reference view M and stored in $2D$ grid, we search for the closest feature to the candidate pixel on the two-dimensional image plane.

In fact, the distance between this pixel and any feature is simply computed using the euclidean distance between the pixels coordinate. Furthermore, after we found these closest feature, we search for its relative re-projected $3D$ point in the point cloud extracted earlier as we mentioned in the Sect. 5.3.4. From this we can easily infer the value d using the equation Eq. 5.3.

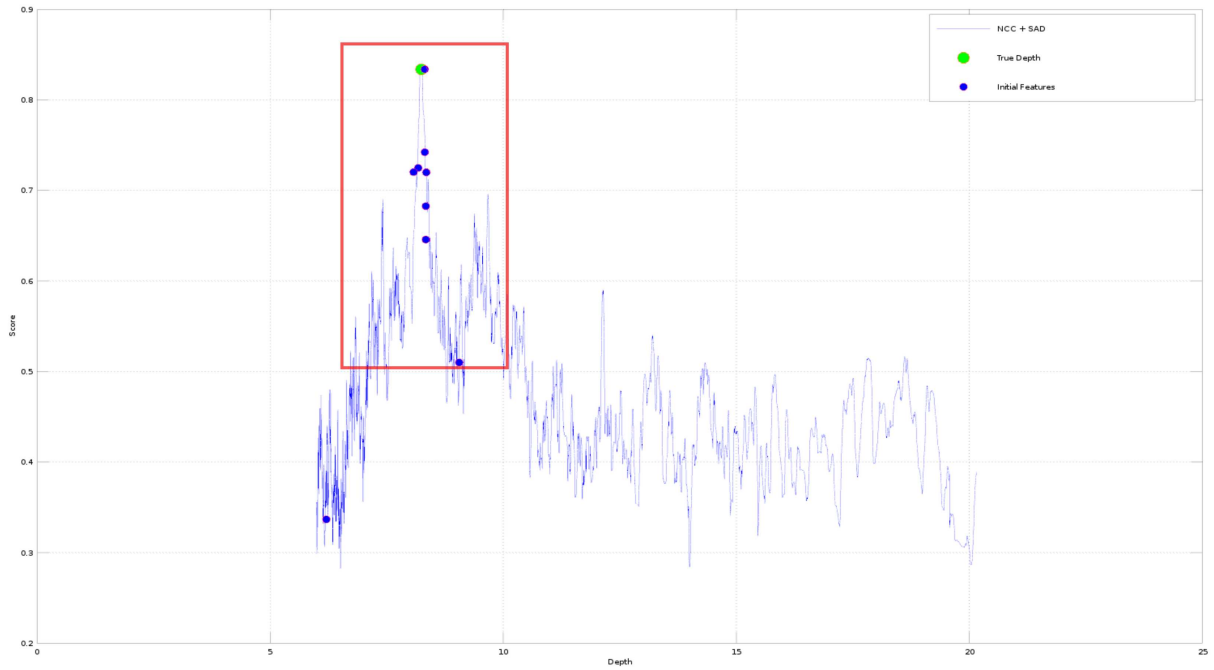


Figure 5.11: Plot of a sophisticated energy function in term of a given pixel depth. The ground truth depth value is the green point, meanwhile the depth value of the initial features is illustrated as blue points

At this stage, the initial position vector $x_i(t)$ of the particle has two element initialized namely the surface orientation and the depth regulator. The remaining unknowns are initialized as follow, first, the roughness factor σ is given the value of 0. In fact, We could have add extras step here and used the PSO to optimise first for only the roughness of input SIFT features, then use the resulted values to find the solution of the candidate pixel. However, we decided to avoid this extras step since the gain is very little to be considered. Second remaining unknown is the *Lambertion correction factor* ε_k. This unknown is a vector of scalars at the size of H. Each element of this vector is set to the ratio between the intensity of the feature at reference view, and its relative pixel at the support image k if visible. Otherwise, it is set to be equal to only the intensity at reference image. The velocity $v_i(t)$ of the particles is initialized also at 0.

As illustrated in Figure 5.10 (a), each candidate pixel has its swarm built with the same process. Hence, this approach is considered as static swarm initialization approach and it cannot count for any new information. To experiment with this approach, we used a ground truth depth map calculated from *the fountain* dataset than we picked the central pixel to a reference image that is the source of the used depth map and plotted its energy function as illustrated in Figure 5.11 and ping pointed the energy value of the ground true depth related to that specific pixel (green dot). Furthermore, we know the depth of the closest features to that pixel, thus we computed the energy value of each of these features and marked them on the plot. It can be

seen that the closest feature are good initial guess to start with the process of optimization.

Dynamic Approach

In this approach, we also assign a swarm S to each candidate element from the list. However, this time the size of the swarm is related to the radius of the search circle centred around the candidate pixel thus making the size change iteratively:

$$radius = \frac{max\_radius}{ln(number\_of\_optimum\_pixels + 1)} \tag{5.26}$$

The main idea is to construct a $2D$ selection two-dimensional pool of particles where the position of each particle in the grid corresponds to a pixel in the reference view. Therefore, we pick the closest particles to the candidate pixel inside a search space of a radius computed with equation Eq. 5.26. In fact, the pool of particles is gradually changing starting with particles initialized by SIFT features like the first approach. Later on, if a candidate pixel is considered optimum, it will be added as a new particle to the pool and will be used to optimise an other candidate pixel from the waiting list, The particle position vector is know initialized via the information extracted from this optimum pixel. The process of this approach is fully illustrated by Figure 5.10 (b).

The reasoning behind this approach is our desire to enforce smoothness on the whole surface implicitly. In particular, each new pixel candidate will receives a higher probability to choose an actual closest neighbour particle thus, boosting its chances for fast convergence. The main drawback of this approach of initialization is the complexity of this swarm creation process where it can be seen that the more converged pixels, the bigger the pool will be.

5.4.3 Matching Process and Expansion

As the second step of our algorithm, the goal in this phase is to optimize over the depth ,orientation and roughness of a given pixel that belong to a planar patch in order to minimize the proposed photometric consistency with its projections into the neighbouring views.

In particular, equation Eq. (5.16) is considered to be a crucial component for our energy function as we mentioned earlier in Sect. 5.3. In fact, in order to understand such equation you have to keep in mind that changing the depth or orientation of the patch in the reference view M moves merely each pixel in depth along the corresponding viewing ray. This ray forms an epipolar line when it is projected in the

neighbouring image k. Consequently, if we take steps along that line in the neighbouring view, the color will change in non obvious pattern when the line intersect different pixels.

Theoretically speaking, every pixel within the reference window, as well as those within the support windows of all the visible support images, should correspond to the same surface points. However, as we have mentioned in the beginning of this section, mapping depth in the reference view to color in the neighbouring view k is not well suited for derivative-based optimization. Thus, a meta-heuristic optimization algorithms is applied for solving such complex and intricate reconstruction problem. After initializing each element of the swarm responsible for finding the solution to a given pixel, an iterative matching process start by evaluating each particle using the complex photo-consistency measurement represented by equation Eq. 5.22.

Each particle uses the current position vector $x_i(t)$ to search for any similarity on the support views. In fact, each particle represent a variation of the planar patch $\Pi(\mathbf{x}, \mathbf{n})$ that is a rectangle represented by an $n \times n$ pixel window in a reference view M. The particles then chose the global best solution which has the best photo-consistency score, this process however is done according to a given convergence criteria as in Sect. 5.4.4.

Furthermore, we iteratively add new candidate pixels to the current waiting list Q. In fact, at the beginning of the reconstruction process we iterate through every $3D$ point of the initial point cloud that is relative to the current reference view. We project every point to a $2D$ image and we chose the four neighbouring pixels that surround the original pixel of the projected $3D$ point namely the top, bottom, left, and right. Each one of these pixel is then inserted to the waiting list Q. Such process assures the cover of the whole scene, especially if the initial features were well spread across the reference view M. To expand over these initial points and propagate throughout the image, we follow the same process however this time if the candidate is successfully optimized, we add its four neighbours to the Q as new possible candidates.

Basically, we are following the region growing approach, especially if we combine this with our dynamic swarm initialization approach proposed in Sect. 5.4.2. In fact, The main reason behind the region growing approach is that a successfully matched pixels provide a good initial estimates for depth and normal for the neighbouring pixel locations in the reference view M. Despite this, such approach may fail for non-smooth surfaces, for this reason we used our proposed photometric model which emphasis on rough surfaces thus reducing the chances of false expansions.

5.4.4 Convergence Criteria

Like any optimization method specially swarm based approach, a convergence condition is essential to ensure an heuristic solution. In this section, we propose two different criteria that assure convergence to an acceptable optimal solution. Both proposals were designed with different goals in mind. In fact, one criteria was specifically build to help the swarm find an acceptable solution as fast as possible, the quality of reconstructed surfaces was a secondary objective which lead to a faster optimization. On the other hand, we took into consideration the reconstruction quality while designing the convergence criteria. In the following we shine a light on these two proposed convergence criteria.

The concept of the first criteria which we will call it MVSPSO 1 during the evaluation section (*see* Sect. 5.5) was based on simple thresholding. In fact, the swarm keeps iteratively searching for an acceptable solution and stops only if the global best position has its similarity measurement function $O_{final}(\mathbf{n}, d)$ function value under a predefined threshold value. The solution provided by the swarm will be rejected if it does not satisfies this condition or a maximal number of iteration is reached. It can be seen that such convergence criteria is in fact unbalanced and would lead to inaccurate convergence hence noisy but dense depth maps. In particular, forcing a static threshold for every optimized pixel is unfair since the global optima in the searching space is not shared between all the reference view pixel, in an other word, the proposed framework is not a designed as global optimization problem.

In contrast, the other convergence condition named MVSPSO 2 in the rest of the chapter was designed to grantee a higher level of accuracy in the reconstructed depth map at the expense of computing time. In order for the optimization to converge and returns an optimal solution, this last has to undergo two tests. In particular, the swarm of particle which represent a variation of planar patch should agree on a given solution. However, before that we analyse the objective function and compute the swarm fitness variance $Variance_i$ value. Comparing the resulted value to a threshold general defined as $\varepsilon = 0.001$ will indicate if the solution should be taken into account. For instance, if the fitness variance is smaller than ε we continue immediately to the other test. Otherwise, we stop after a given number of iterations. The second convergence test is as we mentioned earlier to assure that the swarm is confident in its solution and all the element agree on it.

This basically is done by computing for each particle the average NCC score between the patch represented by the particle in reference view and all its support views, the mean average of all computed NCC scores is then computed and considered as the confidence value of swarm. Hence, if it value was higher than

Parameter	Description	value
H	number of neighbour images	4
n	window size	5
λ_{sad}	threshold that tune the consistency values of SAD	10
λ_{Census}	threshold that tune the consistency values of Census	8
$Conf$	the accepted NCC threshold	0.6
ε	fitness variance threshold	0.001
ε_{max}	the maximum fitness accepted	0.2 and 0.3
$Iter$	maximum iteration	20 to 101
P	swarm size	4
w	swarm constriction factor	0.2373270

Table 5.1: A choice of parameters used in our experiments.

$NCC = 0.6$ we accept the given depth, normal and roughness by the swarm as the optimal solution. At the end obtained depth map maybe less complete compared to the above approach (MVSPSO 1). Yet depth values are more accurate. The only drawback of this criteria is the amount of computing effort and time spend.

This section disused in detail all the necessary elements for our reconstruction algorithm starting from selecting the needed support view down to the convergence condition. The next section is devoted to show the obtained results using our proposal.

5.5 Evaluation and Experimental Results

In this section, we show our results obtained following our method. We have implemented the proposed approach in C++, For the rest of this section, We evaluate our method quantitatively and analyse the visual quality using a variety of well known datasets. Moreover, a choice of parameters for in our experiments are listed in Table 5.1. Note that all the parameters presented here are can not be considered as the standards settings for or method. However, this setting is the bare minimum which can grantee an acceptable accuracy in short amount of time. This parameters are also obtained via test and repeat process.

We mentioned earlier in Sect. 5.1.3 that our work, can be easily integrated with the proposed work of Fuhrmann et al. [24]. In fact, our implementation is based on this work and benefit from the available tools. In particular, We used *scene2pset* a tool provided by the authors of [24] in order to combine the resulted depth maps into single large dense point cloud. This points are then converted to a full solid mesh

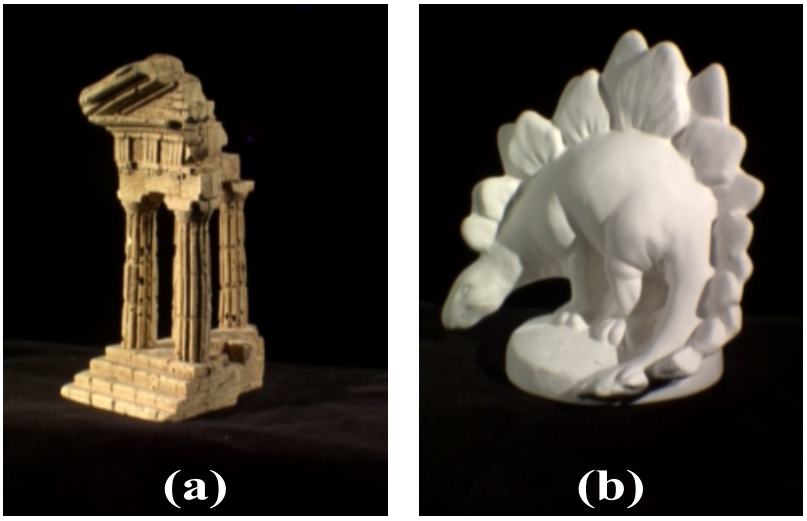


Figure 5.12: The MidleBury Benchmark sample images: (a) represents the 23th view in *templeRing* dataset, (b) represents the 23th view *dinoRing* dataset.

via Screened Poisson Surface Reconstruction [51], this algorithm uses an integer parameter which is the maximum depth of the tree that will be used for surface reconstruction, we set this parameter to a value equal to 8 for most of the experiments.

Before we dive into the experiments in the next subsections, we like to present the general key findings for the proposed experiments. this results will be discussed later on. In fact, we found that the use of the swarm optimization will mimics the variational approaches if it is used as we proposed in our method. In contrast to these variational methods, ours is less complicated, more efficient and derivative free. Moreover, our goal is to avoid explicit regularization. Thus, the usage of the swarm formulation coupled with the neighbouring strategy to initialize each particle, and the waiting list will assure to propagate in the image space and guarantee the smoothness of the surface implicitly while preserving thin details.

To summarize, our method found a middle ground between the variational and classical MVS approaches and it can be positioned in the literature as multi-view stereo depth map estimation method with local quasi-variational optimization.

5.5.1 Experimental Results

As most of the other research in Multi-view stereo, we tried to evaluate our method quantitatively and compare it to the other state of art methods. Thus we demonstrate our algorithm on standard multi-view stereo benchmark Middlebury Multi-view Stereo [108]. this benchmark contain 6 datasets. We used four of these datasets, namely the *temple* (312 images), *templeRing* (47 images), also the *dino* (363 images),

Datasets	Accuracy		Difference to best approach		completeness	
	MVSPSO-1	MVSPSO-2	MVSPSO-1	MVSPSO-2	MVSPSO-1	MVSPSO-2
Dino Full		1.3 mm		1.04 mm		99.1 %
Dino Ring	2.29 mm	1.05 mm	2.04 mm	0.8 mm	97.1 %	93.3 %
Temple Full	0.75 mm	0.54 mm	0.41 mm	0.20 mm	97.2 %	97.9 %
Temple Ring	1.07 mm	0.77 mm	0.67 mm	0.37 mm	93.5 %	95.8 %

Table 5.2: Quantitative evaluations using the Middlebury dataset regarding accuracy, completeness.The third and fourth columns list the difference between the accuracy of our approaches and the accuracy of the best performing state of the art approach

dinoRing (48 images) (*see* Figure.5.12). Such benchmark poses multiple challenges for any MVS reconstruction especially the *dino* dataset which has almost no textures and it is affected by inter-reflections and self shadowing.

The first experiments are designed to evaluates the proposed method quantitatively. The evaluation was done via a geometric comparison to the ground truth model offered by the Middlebury [108], the measurements used are accuracy which tell us how close our reconstructed model to that of the ground truth and completeness which indicate how much of the ground truth model did our method recovered. These lasts are given with thresholds being 90%, i.e an accuracy of 1 *mm* indicate that 90% of the point are within the 1 *mm* of the ground truth model. For completeness the used threshold is equal to the value of 1.25 *mm*.

The results of our quantitative evaluations is showed with details in the Table. 5.2, the result reconstruction of the *temple full* model achieved using the second convergence criteria MVSPSO 2 an accuracy of 0.54 *mm* with 97.9% completeness. Such statistics is considered as a prove for the capability of our method to handle interreflections effects presented in this dataset. On the other hand, the *temple ring* model achieved an accuracy of 0.77 *mm* at 95.8% completeness. Note that the accuracy of the reconstructed model has dropped by 0.22 *mm*. since we did not change any parameters for both experiments except for the number of used input photographs. We can safely conclude that this last has affected the process output. In fact, in multi-view scenario the more image is used the more information is extracted, hence, better accuracy.

The second dataset used to evaluate the accuracy and completeness of our method is *dino full*. As shown in Table 5.2, the reconstruction attained a low accuracy rate with 1.3 *mm* compared to the other results, however, we achieved a high ratio of completeness at 99.1% . Despite the lack of texture in this dataset, the resulted completeness we scored remain largely undiminished which indicate the capability of our

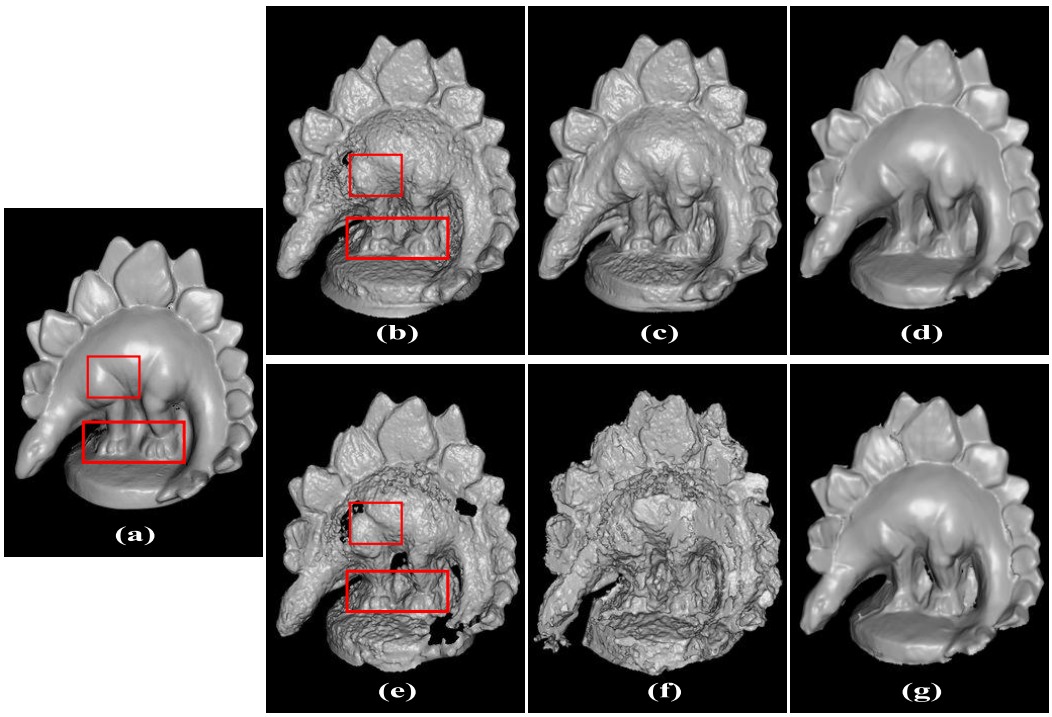
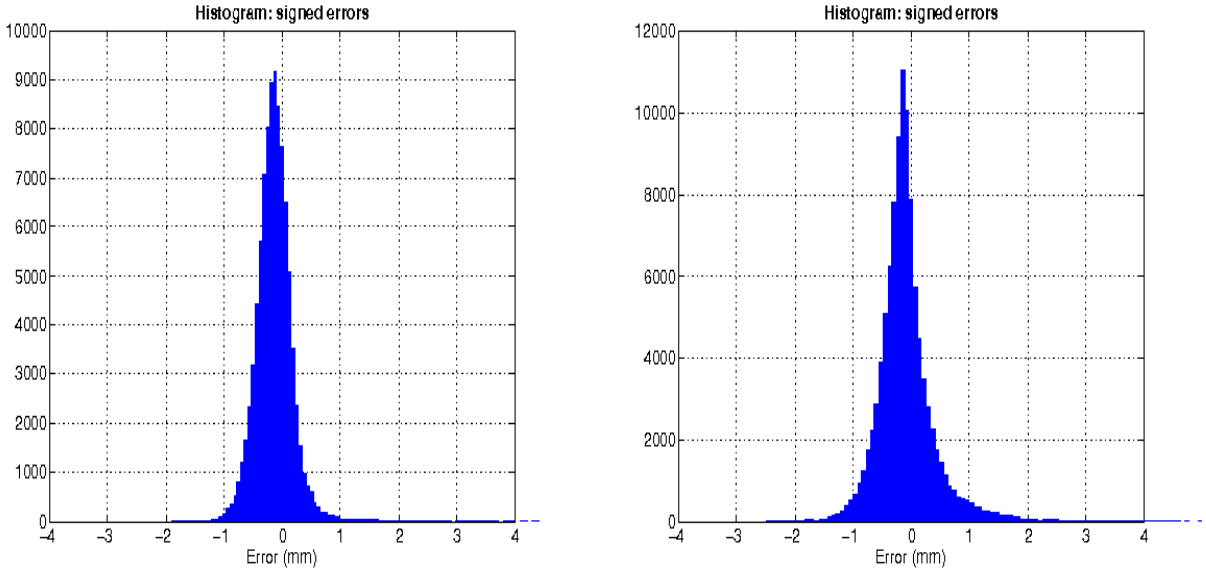


Figure 5.13: Visual results on *Dino* datasets with different approaches and decreasing number of input image. (a) The ground truth model. Top: reconstruction of full dataset with 363 images. (b) Our approach MVSPSO 2; (c) Goesele-MVE; (d) SMVS. Bottom : reconstruction of ring data set. (e) Our approach MVSPSO 2; (f) Our approach MVSPSO 1; (g) SMVS .

approach to deal with such surfaces as expected. Finally, we can see a small accuracy improvement in *dino ring* with 0.25 *mm* difference despite the lower amount of the used photographs. This due to the nature of the captured scene, in particular, using more photographs of untextured surface causes more confusing during the reconstruction where a notable differences between for a same surface visible in multiple depth maps. We emphasis, that we only used 4 images as subset for the matching process and the optimization stops after 60 iteration.

On the other side, the first convergence criteria MVSPSO 1 achieved the lowest of all scores under all datasets. As we mentioned in the above analysis at Sect 5.4.4 the usage of threshold approach as convergence criteria is quite dangerous and can lead to false matches. For instance, some of the solution given by the swarm can be under a given threshold yet they do not represent the true surface in the scene. another reason for the low accuracy in this setup is that we dawn sampled the input images to a half-scale, the goal was to benefit from the simplicity of this criteria and leverage the execution time. In fact, this condition achieved on the *dino ring* a small computing time almost equal to 53 second per view. The final visual results however were not satisfactory as shown in Figure.5.13 (f).



(a) templeFull [MVSPSO-2] (b) templeRing [MVSPSO-2]

Figure 5.14: Histograms of signed errors

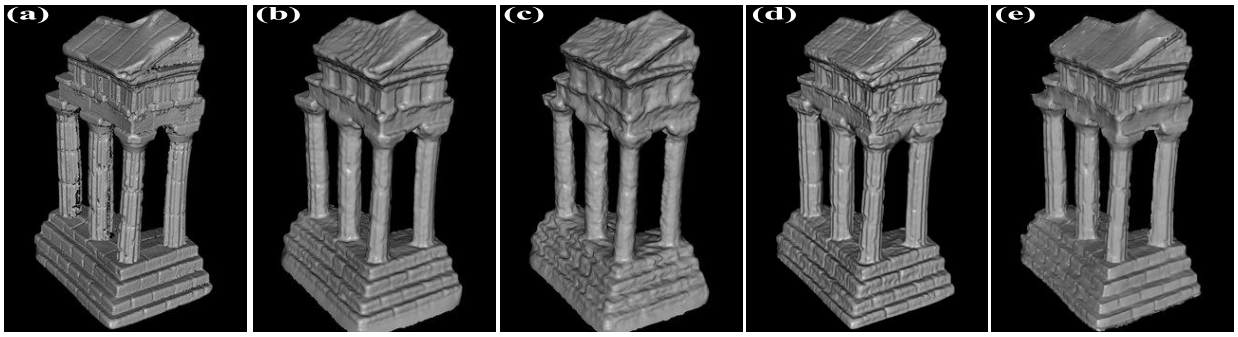


Figure 5.15: Visual results on *Temple full* dataset with different approaches (a) The ground truth model; (b) Our approach MVSPSO 2; (c) Our approach MVSPSO 1; (d) Goesele-MVE ; (e) SMVS.

It can be deduced from Figure.5.14 which represent the histograms of the signed errors for temple dataset that the error in approach is approximately Gaussian distributions, with approximately zero means. In fact, our reconstruction results showed an accuracy within a 0.20 to 2.00 millimetre difference from the best state-of-art reported results. For more in-depth information on the evaluation results and comparisons with other methods, interested readers are referred to the Middlebury MVS evaluation* web page.

Visual Evaluation

After asserting the quantitative quality of our results in term of accuracy and completeness. It is only fair to check the visual quality of 3*D* model reconstructed via our method. Let us for instance take a look and observe Figure.5.13. Such figure illustrates the visual quality differences between our proposals (b,e,f) and

*http://vision.middlebury.edu/mview/eval/

other state-of-art multi view stereo approaches. As we mentioned before this benchmark is considered as the worst case scenario for any multi-view stereo approach do to the lack of texture and strong shadowing. However, as it can be seen in the figure, the areas where our photometric model is correct, we were able to recover a good amount of details for example the dinosaur's foot area, especially in case of the full dataset.

A fully textureless and super smooth surfaces are in fact rarely found in outdoor environment and natural scenes. Hence, our photometric model designed by definition to favour rough surfaces. In fact, every pixel in the reference photograph represent a point on a rough surface and if by any chance one of this pixel neighbors happen to be projected on the same small surface, there will be no smooth depth transition instead the depth will be computed according to the surface roughness factor. Evidence for in support of this position, can be found in visual results presented here, where the presence of visual noise in some area is the direct result of our optimization estimating the wrong roughness thus affecting simultaneously the normal and depth estimation. To be more precise the proposed algorithm in Sect.5.4 in the rare case of super smooth textureless surfaces assume wrongly that some part of the surface is rough.

Visually speaking, Temple dataset Figure.5.15 (b,c) exhibits the effectiveness of our algorithm in recovering thin details, which shows another advantage of our approach, meanwhile the figure demonstrates the capability of our simple meta-heuristic method to contend with the current state-of-the-art approaches for images captured under lab conditions.

Miscellaneous Experiments

In the previous section, we demonstrated some interesting results in the case of controlled lab environment like the Middlebury benchmark data. Meanwhile, the realistic outdoor scenes for which multi-view stereo is not suitable are presented mainly in community photo collection. It is therefore a necessity to quantitatively evaluate our method in such a scenario. For this reason, we decided to use one the well known benchmark dataset of the outdoor environment namely *Fountain-P11* created by Strecha et al. [121] is used.

In particular, this dataset has 11 photographs at pixel resolution equal to 3072×2048, the authors also offers with the photographs the geometrical ground truth, which is a single high-resolution triangle mesh model obtained by laser scanning (LIDAR). Some sample images in the dataset are showed in Figure.5.16 (a). Using such dataset benchmark we can compare and evaluate the recovered depth-maps directly instead of evaluating the finale $3D$ model. The reason behind this is our fear that such approach in outdoor scenario

Figure 5.16: Sample images from fountain-p11 dataset. (a) and their ground truth depth-maps (b).

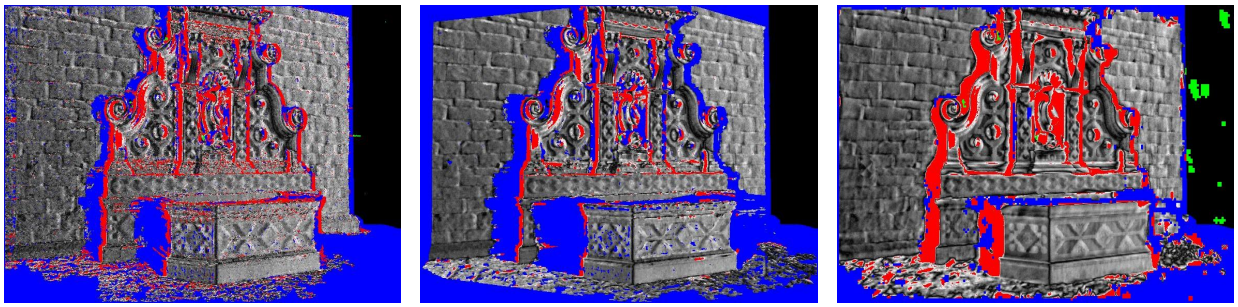
would lead to unfair evaluation and comparison, due to each method in the state-of-art has its own way of creating and merging the depth-maps.

We decided to compare our method to both variational and classical approaches namely the multi-view environment (MVE) approach by Goesele et al. [36] and the shading aware multi-view stereo (SMVS) proposed by Langguth et al. [61]. These two methods are the closest to our work.

In order to make this comparison appropriate, we first down scale the provided photographs to the scale of 768×512, therefore, which will reduce the amount of computation during the reconstruction phases. One problem we encounter during this evaluation is the absence of raw depth-map, since the ground truth model is in $3D$ triangulated mesh form. To resolve such minor problem, we resorted to rephotographing this model with precise camera calibration per view. Hence, we created a pixel-accurate depth map that we can work with as shown Figure.5.16 (b).

Let us define the ground truth depth as h_{gt} and the depth computed by the multi-view stereo method by h. For every pixel in the image, the estimated depth error between the ground truth and the computed depth can be computed using the following mathematical formula:

$$e = \frac{||h - h_{gt}||}{h_{gt}} \tag{5.27}$$



(a) (b) (c)

Figure 5.17: From left to right: depth error maps for the 8th image in Fountain-P11 using the our method, multi-view environment method and the shading aware multi-view stereo (SMVS) method, respectively.

We measure how accurate the reconstructed depth is for the outdoor scenes based on the above equation 5.27. In particular, we compare the depth error e to a threshold $\mathfrak{t}$. If we found that the error is below that threshold, then we conclude that the estimated depth by one of the used approach is considered as correct.

The result of this experiment is illustrated Figure.5.17 which represent the visual comparison as an error maps for a given input view image from the *Fountain-P11* dataset. Note that this depth map tested on are obtained via our method, multi-view environment method and the shading aware multi-view stereo (SMVS) method, respectively. The pixels in these images are classified into multiple groups. First, the blue pixels represent any missing depth values by the used MVS approach due to failure during the reconstruction process. As for the green pixels they are the incarnation of the missing ground truth data which basically rare case. Meanwhile the red pixels encode an error e larger than the threshold $\mathfrak{t} = 0.01$, however, pixels with depth errors less or equal to $\mathfrak{t}$ are represented in gray level between [255, 0].

The results show that our method along with SMVS method can reconstruct much more geometry than the MVE method, which prove that our method gain a important trade from the variational approaches. We can say also that our method is balanced. Evidence for in support of this position can be fount in the fact that we recovered more geometry with fewer errors as it is shown in Figure 5.17 (a) (more gray pixel and less red pixel). The meta-heuristic algorithm was a successful choice to solve our sophisticated objective function, this function was based on advanced shading model which is one of the reasons for capturing small details.

To further evaluate the reconstruction accuracy of our approach for outdoor scene and confirm the above experiment results, we compute the depth error map for all the dataset photographs and count the total

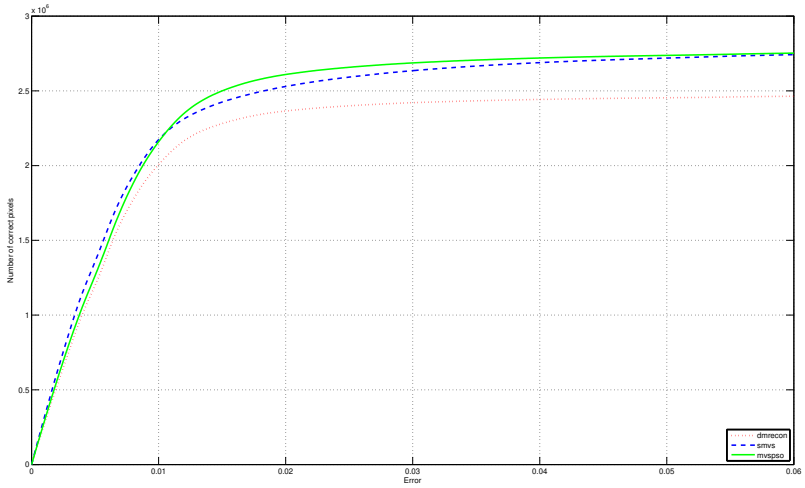


Figure 5.18: The number of correct pixels in all images as a function of the error

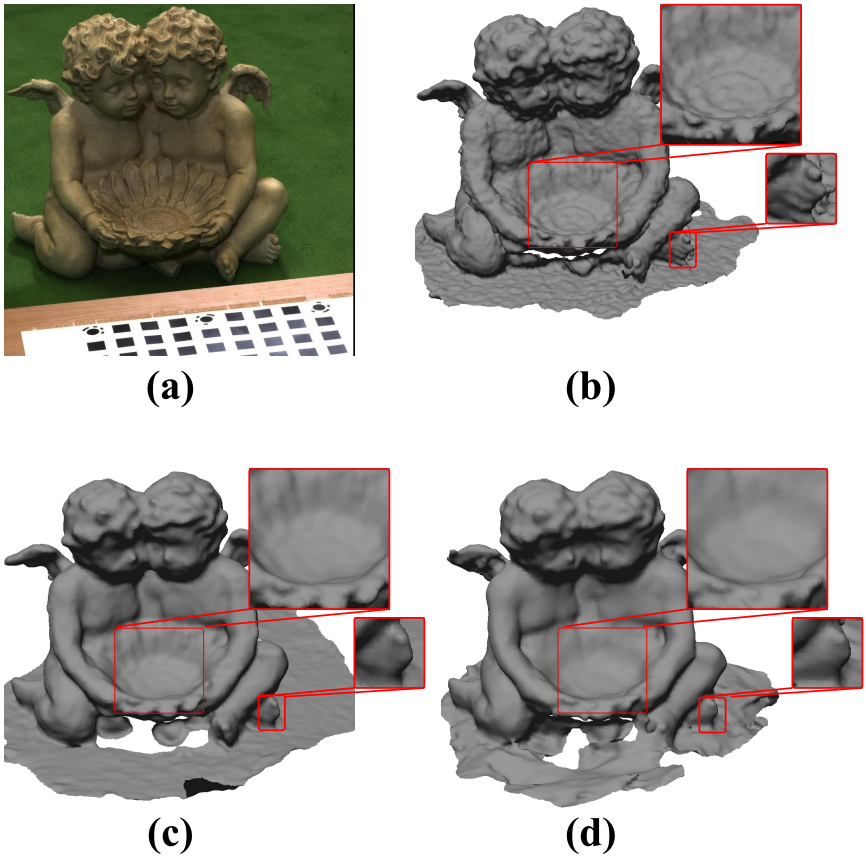


Figure 5.19: *The angles* dataset from Wu et al. [132] reconstructed using low resolution of the original image (a) Original image ; (b) Our approach MVSPSO 2; (c) MVE (d) SMVS .

number of correct pixels. This process is then repeated with multiple values for the error threshold in the range of $t = [0.0, 0.06]$, the results can be seen illustrated in the Figure.5.18. If you look at the number of correct pixels when the error threshold is in the range of $[0, 0.01]$, the proposed method (green line)

and SMVS method (blue dashed line) exceed the MVE method (red dotted line) and scored more correct pixels. Moreover, at $t = 0.01$ mark our method approximately has the same total of the correct pixels with that of the variational approach. This confirm that a simple meta-heuristic can rival a complex variational method.

Given these points, we concluded that our method can handle easily with an acceptable accuracy the outdoor scenes. Such conclusion come from the nature of our photometric model coupled with the proposed optimization which can deal with rough surfaces and other effects very well.

As a final experiment, we present *the angles* dataset which was used in Wu et.al paper [132]. We downscaled the input images used in reconstruction to a pixel size four times smaller than the original resolution that was equal to 840x872. The result of our method is shwon in Figure.5.19 (b). The figure prove that we were able to retrieve some of the small structural details (notice the angles fits, heads, and the vessel) compared to other approaches although the input photographs has small size. Moreover, the proposed method was able to recover more accurate geometry in shadowed regions, such as under the vessel.

Compared to other methods, it is clear that forcing smoothness on the surface will lead the reconstruction to recover a false geometry. In contrast, we do not impose any explicit regularization to force smoothness. To ensure the continuity of the surface reconstruction however we initialize the swarm with the candidate pixel neighbors as we mentioned in Sect. 5.4.2, yet we still capture a detailed geometry due to the nature of our energy function which is based on an advanced illumination model.

Finally, we computed depth maps for several outdoor datasets such as the *Fountain-P11* [121], and the first *Foundation stone* of our university dataset. Figure.5.20 shows for each site a sample view, the corresponding depth map, and a shaded rendering of the reconstructed model. the proposed method can recover easily a detailed surfaces in such scenarios.

5.5.2 Discussion

We have presented a method to reconstruct an accurate and complete three-dimensional geometry from unstructured collection of photographs, these lasts can be captured under natural yet uncontrolled illumination. As we mentioned in Chapter 1, the ultimate goal is to achieve a high quality models to be used in a virtual reality experience as an environmental probes or even a whole scene. According the results

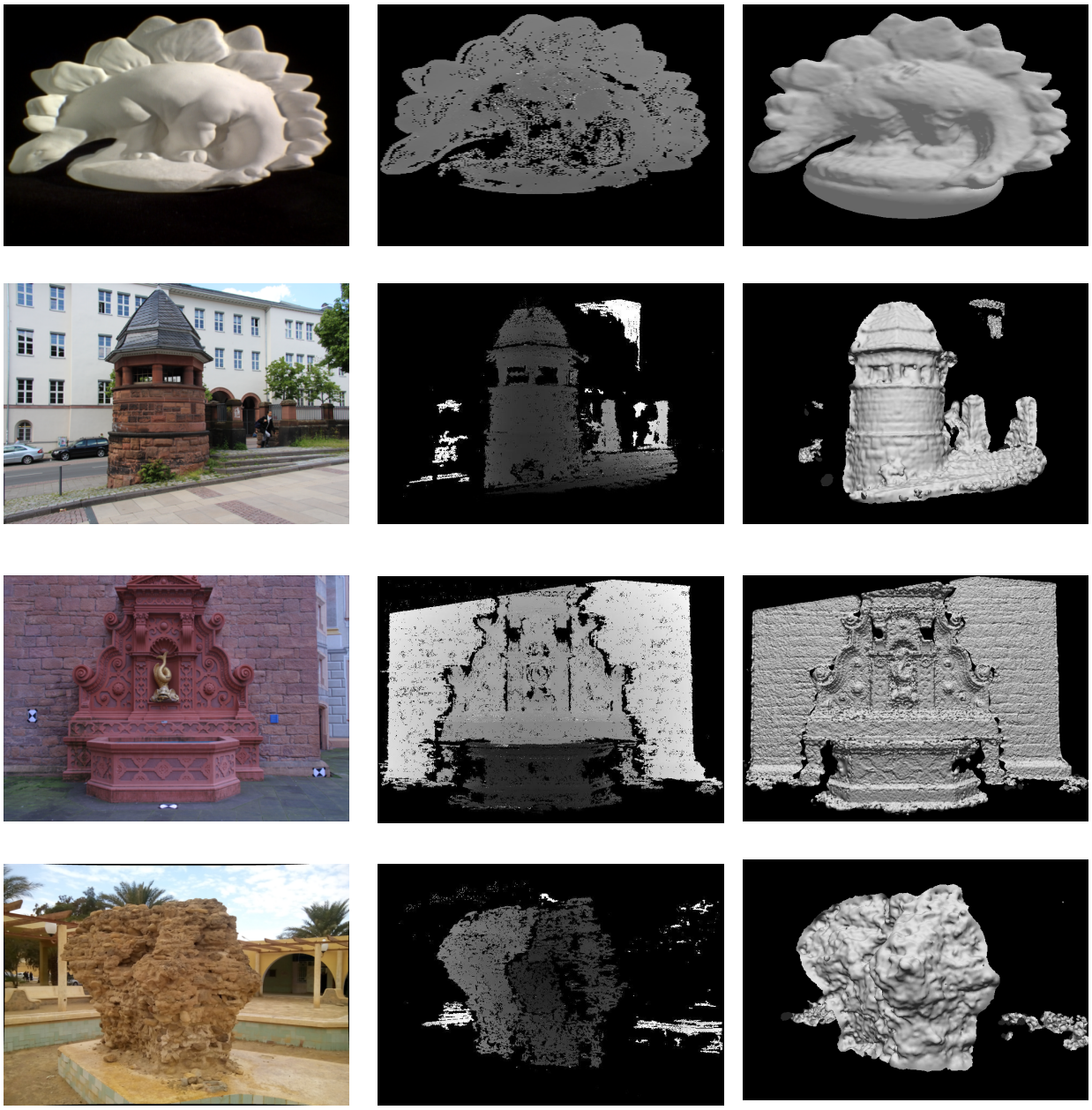


Figure 5.20: Individual views from multiple datasets namelly the dinoRing, Achteck Turm, Fountain, and Foundation stone. Along with their corresponding depth maps and the shaded renderings of the reconstructed $3D$ model

presented in Sect. 5.5.1, the proposed method achieved to some degree the underlined global objectives. In the rest of this section we will be discussing the obtained results.

In particular, approach runs under one main assumption that can lead to noise or errors in the final geometry if it is violated. We suppose that the reconstructed surfaces are non-smooth and rough. Also, the scene is under direct illumination. Despite this, the proposed method showed some interesting and expected results. However, for Dino datasets, the results is not as good as others. One could ask the reason behind that.

In fact, We support such assertion. However, this result was almost to be expected as we already mentioned, and can be explained as follow. First the Dino dataset is comprised of images of a plaster dinosaur. We have used the *dinoRing* dataset (48 image) and *dinoFull* dataset (363 image) the main difference between these two datasets aside from the number of used images is the baseline separation between camera poses. The *dinoRing* obviously has a larger separation. Moreover, the biggest challenge of the dinosaur images comes from their extremely low texture.

This benchmark is considered as the worst case scenario for any multi-view stereo approach. Our method is no different, not only the lack of texture but also the surface is almost perfectly diffuse which assume smoothly varying surface normal vector and depth value. Our method as mentioned in Sect.5.2 is focused on rough surfaces, despite that, we were successful on recovering geometry for this dataset.

In detail, first thing, our method work on pixel level and has no explicit regularization unlike MVE or SMVS approach. Furthermore, our photometric model is designed using oren-nayar [89, 90] BRDF which is explained in Sect. 5.3.2. This model basically try to match pixels while assuming that they belong to a rough surface (nature surfaces are non-lambertion but they are rough). Consequently, we are searching among the possible depth,normal and roughness values a solution that give a correct stereo match. However the dinosaur dataset throws this assumption out of the window. and with out any external regularization, obtaining results close to what is shown in Figure 5.13 (a) is quite hard.

The solution to such problem is one of two things. Either we enforce some sort of a smoothness term into our optimization (hard to do and counter-intuitive) or we play with our algorithm parameters such as number of the swarm particles and the iterations this will force the algorithm to find the true depth on surface that have a very small roughness factor. In fact, if we put the value of σ to be equal to zero, equation Eq. 5.9 will reduced to the Lambertian BRDF hence, the non-Lambertian factor ξ_k will be equal to a value

of one, thus the whole proposed model will collapse to a simple smooth surfaces and our method will be similar to the MVE approach.

Evidence for in support of this position, can be found in an experiment that we preformed using the *dinoRing*. In fact, during the optimization process we set $\sigma = 0.0$ excluding it from the optimization. The final result is shown in the third image of the first row in Figure.5.20. Our approach succeeded to recover a smooth surface similar to other multi-view methods.

At this stage, we like to emphasis that during the computation the depth maps, especially in the matching process which is presented in Sect. 5.4.3, we adapted gray scale color mode as the intensity function. The reason for that is using the full-color spectrum will enhance the results ever slightly, but at the cost of increasing the computational effort.

Speaking of execution time, we found that the most time-consuming step in our optimization method is the evaluation process. In fact, this evaluation is done for each particle in the swarm every iteration. We took advantage of the processing power available in a multi-core CPU and we assigned multiple particles to a single thread. Yet, In this unoptimized prototype implementation it took us *43566 ms* to compute a single depth map for the *achtek-turm* dataset presented in [24] with input images scaled down to four times less from the original resolution.

We think that the runtime per image is still unsatisfactory and it can be improved by adopting a sophisticated parallel implementation to some part of the proposed method. For instance, each pixel is computed independently, hence, we can implement our optimization as multi-stage process, and each stage will be executed on GPU as small shader programme.

Finally,the proposed method also output for each input photograph a normal map, confidence map which is NCC score for each pixel, and finally a roughness map. It is important to know that the depth map fusion and the surface reconstruction algorithms are not considred in this dissertation as the main contribution. However, in a future work we think it is possible to use other maps in conjunction with depth maps to reconstruct a cleaner and more accurate surface.

5.6 Conclusion

In this chapter, we presented a new robust multi-view stereo optimization algorithm that efficiently deals with non-smooth surfaces, shadowed and textureless regions. We also described the various key element considered for the proposed algorithm. This last, was also discussed with detail in this chapter. Multiple experiment were also presented to assess our initial claims and assumptions. Consequently, the analysis of the obtained results has confirmed that these assumptions are reasonable for the considered datasets.

In fact, after comparing our method in which depth, surface normal and roughness are estimated simultaneously to other state-of-art multi-view stereo methods we found that a high completeness score was achieved after reconstruction from set of photographs which lack the presence of textured regions or suffer from presence of shadows.

Moreover, a satisfactory accuracy and thin details retrieval were obtained for most evaluated datasets. Our method is a derivative-free patch-based optimization, and it is only limited by the simple lighting model and cannot account for very smooth surfaces and specular materials.

Overall we believe the idea of introducing a non-Lambertian surface property and using a meta-heuristic optimization approach can be compelling and will lead to more sophisticated approaches for community photo collection.

Its the not the Destination, It's the journey.

Ralph Waldo Emerson

6 Conclusion

6.1 Summary

In this dissertation, we have presented two contributions for two core problems in multi-view stereo reconstruction in the context of virtual reality applications: interactivity for real-time reconstruction, and multi-view stereo for objects captured under general conditions. The contribution of this dissertation are summarized as follow.

Image-based visual hull reconstruction: One of the earliest and most challenging problem in Computer Vision is reconstructing an accurate three-dimensional geometry from multiple photographs. In fact, the task to solve such an ill-posed problem can be more complicated if the desired object model needs to be reconstructed in real-time. For instance, in virtual reality applications and tele-presence systems, the reconstruction techniques must be able to resolve the $3D$ geometrical shape of any object in a matter of milliseconds. In order to deal with these issues, this dissertation present a GPU accelerated image-based modelling system, that aims to estimate and render on the fly all visible parts of a photo hull from a novel viewpoint without noticeable artifacts. We presented a parallel strategy that leads to a fast computation of the image-based visual hull [100]. Therefore, virtual reality and tele-presence system could benefit easily from such implementation which is available freely on Github [*].

[*] https://github.com/HelliceSaouli/GIBVH

Our goal was to upgrade the performance of the visual hull algorithm via a GPU image-based approach. This is not an easy task, since multiple considerations should be taken otherwise we can not beniift from the graphical processor power. Hence, the proposed solution first uses the CPU pinned memory and the GPU global and constant memory to overcome the a greate performance hit when transforming images from CPU to GPU. We also proposed a strategy to reduce the amount of data transformation. Basically each frame captured from every camera passes by pre-processing on the GPU and stay cached on the global memory until the system decide to use it for reconstruction. On the other side, we propose the usage of the linear-fetching mode to retrieve the position from the three-dimensional texture in order to guarantee the smoothness of the final visual quality of the reconstruction in another word removing the voxalization effect. At the end, we think that our system could be a starting point for photo-consistent or stereo approaches for a smoother 3D model. A device optimizations and multi-GPU usage is proposed as perspective for future works.

Towards a stochastic and accurate depth maps for quasi-lambertian surfaces: Application such as virtual travel, remote training, architectural walk troughs, needs to assure immersion for the users otherwise it is not beneficial. This can be solved if the virtual world was built based on real world locations and objects, to do that one has to rely on multi-view stereo reconstruction approaches. This fact, make our contribution [101, 102] beneficial for such a scenario. In fact, we frame the classical dilemma of multi-view stereo reconstruction as a local search problem for the optimal depth, orientation and roughness on every input image.

We decide to focus on the completeness and accuracy of the reconstructed model from multiple images. It is a challenging task a specially under variable acquisition conditions. This difficulty is due to the connection between the geometry and the shading, where it is impossible to estimate one without having a prior knowledge of the other. In our approach we try to solve MVS problems namely, the untextured regions, thin details and quasi-Lambertien surfaces. Generally multi-view stereo algorithms infer the 3D surface from image by observing the similarity between these images, most of the approaches assume the properties of the surface to be Lambertian which is physically not correct thus it can not deals with rough surfaces, shadows, inter reflections,and textureless or occluded surface regions. we presented a new robust multi-view stereo optimization algorithm that efficiently deals with non-smooth surfaces, shadowed and textureless regions. We also described the various key element considered for the proposed algorithm. namely, light

estimation a new geomtric patch model and a non lambertian photometric model. Our depth maps are reconstructed based on a meta-heuristic optimization technique. In fact, we were inspired by the ability of the human mind to solve vision problems with a perfect accuracy based solely on heuristic. We propose a new multi-view depth map and normal along with surface roughness estimation (MVDE) algorithm based on the particle swarm optimization (PSO) approach. Multiple experiment were also performed to prove the initial claims and assumptions. Consequently, the analysis of the obtained results has confirmed that these assumptions were reasonable and a detailed virtual world in fact could be reconstructed based on photographs sing our algorithm.

6.2 Perspectives

The contributions presented in this thesis are a partial solutions to the multi-view stereo problems in the context of virtual reality. The objective is to create an impressive virtual reality experience built from real world. Thus, multi-view stereo reconstruction method should accurately recovers the three-dimensional geometry. In fact, a big part of our future work will be focused on offline reconstruction where we aim for a higher accuracy. It is also our interest to use our contributions in gaming industry, where we hope for game engines such *Unreal Engine* to integrate image-based modelling approaches into the engine which will facilitate the work for game developers.

Out-of-shelf Tele-presence System: State-of-art reconstruction methods that aim for real-time performance often uses a very complicated hardware which is not affordable by any users. Sometimes a network of machine coupled with a complicated camera setup are required to recover the geometry in real-times. On the other hand, smart-phones processing power is rising to the roof, add to that the quality of digital camera integrated to it. We envision a tele-presence system build upon a couple of smart phones and laptop. For this, a very efficient reconstruction system should be designed.

Optimization problem: In this dissertation, we cast the reconstruction problem as optimization problem, where the objective is to search for optimal depth and orientation in the $3D$ space that minimize/maximize a given objective function. Such function could be sophisticated and contains a hundreds of variables which are involved in the reconstruction problem. Thus, it is possible for the optimization algorithm (in our case PSO) to avoid falling in a local minima although a good initialization were used (SIFT features). It is very interesting to explore the road of meta-heuristics furthermore, we think it is possible to use *Fractal-based* meta-heuristic to deal with large scale objective functions. It is also interesting to study the possibility of a parallel implementation to such scenario in order to reduce the time of the reconstruction process.

Photometric model: This dissertation proved that it is no longer necessary to assume the lambertian law for all the objects of the scene. However the proposed photometric model can be extended to enhance the accuracy of the reconstruction or even reconstruct other type of surfaces. It is very interesting also if we could include some elements from the global illumination theory to recover as much geometry as possible.

Bibliography

[1] Akenine-Möller, T., Haines, E., & Hoffman, N. (2008). *Real-Time Rendering*. CRC Press.

[2] Anwer, A., Ali, S. S. A., Khan, A., & Mériaudeau, F. (2017). Underwater 3-d scene reconstruction using kinect v2 based on physical models for refraction and time of flight correction. *IEEE Access*, 5, 15960–15970.

[3] Batsos, K., Cai, C., & Mordohai, P. (2018). CBMV: A coalesced bidirectional matching volume for disparity estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (pp. 2060–2069).

[4] Baumgart, B. G. (1974). *Geometric Modeling for Computer Vision*. PhD thesis, Stanford, CA, USA.

[5] Bohil, C. J., Alicea, B., & Biocca, F. A. (2011). Virtual reality in neuroscience research and therapy. *Nature reviews neuroscience*, 12(12), 752.

[6] Bujnak, M., Kukelova, Z., & Pajdla, T. (2008). A general solution to the P4P problem for camera with unknown focal length. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*.

[7] Burdea, G. & Coiffet, P. (2003). *Virtual Reality Technology*. Academic Search Complete. Wiley.

[8] Bülthoff, H. & L. Yuille, A. (1991). Shape-from-x: psychophysics and computation. *Proc. SPIE*, 1383.

[9] Chang, B., Woo, S., & Ihm, I. (2014). Gpu-based parallel construction of compact visual hull meshes. *The Visual Computer*, 30(2), 201–211.

[10] Cheng, J., Grossman, M., & McKercher, T. (2014a). *Professional CUDA C Programming*. EBL-Schweitzer. Wiley.

[11] Cheng, J., Leng, C., Wu, J., Cui, H., & Lu, H. (2014b). Fast and accurate image matching with cascade hashing for 3d reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014* (pp. 1–8).

[12] Clerc, M. & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Computation*, 6(1), 58–73.

[13] Cline, E. (2011). *Ready Player One*. Random House, first edition.

[14] Cui, Z. & Tan, P. (2015). Global structure-from-motion by similarity averaging. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015* (pp. 864–872).

[15] De Araújo, B. R., Lopes, D. S., Jepp, P., Jorge, J. A., & Wyvill, B. (2015). A survey on implicit surface polygonization. *ACM Computing Surveys (CSUR)*, 47(4), 60.

[16] Duckworth, T. & Roberts, D. J. (2014). Parallel processing for real-time 3d reconstruction from video streams. *J. Real-Time Image Processing*, 9(3), 427–445.

[17] Esteban, C. H. & Schmitt, F. (2004). Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3), 367–392.

[18] Esteban, C. H., Vogiatzis, G., & Cipolla, R. (2008). Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3), 548–554.

[19] Faugeras, O. D. & Keriven, R. (1998). Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE Trans. Image Processing*, 7(3), 336–344.

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

[21] Franco, J. & Boyer, E. (2003). Exact polyhedral visual hulls. In *British Machine Vision Conference, BMVC 2003, Norwich, UK, September, 2003. Proceedings* (pp. 1–10).

[22] Franco, J. & Boyer, E. (2009). Efficient polyhedral modeling from silhouettes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(3), 414–427.

[23] Freiknecht, J. & Effelsberg, W. (2017). A survey on the procedural generation of virtual worlds. *Multimodal Technologies and Interaction*, 1(4), 27.

[24] Fuhrmann, S., Langguth, F., & Goesele, M. (2014). MVE - A multi-view reconstruction environment. In *2014 Eurographics Workshop on Graphics and Cultural Heritage, 2014, Darmstadt, Germany, October 6-8, 2014* (pp. 11–18).

[25] Fung, J. & Mann, S. (2008). Using graphics devices in reverse: Gpu-based image processing and computer vision. In *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo, ICME 2008, June 23-26 2008, Hannover, Germany* (pp. 9–12).

[26] Furukawa, Y., Curless, B., Seitz, S. M., & Szeliski, R. (2010). Towards internet-scale multi-view stereo. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 1434–1441).: IEEE.

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

[28] Furukawa, Y. & Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8), 1362–1376.

[29] Galliani, S., Lasinger, K., & Schindler, K. (2015). Massively parallel multiview stereopsis by surface normal diffusion. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015* (pp. 873–881).

[30] Gallup, D., Frahm, J., Mordohai, P., & Pollefeys, M. (2008). Variable baseline/resolution stereo. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*.

[31] Gallup, D., Frahm, J., Mordohai, P., Yang, Q., & Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*.

[32] Gargallo, P. & Sturm, P. (2005). Bayesian 3d modeling from images using multiple depth maps. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2 (pp. 885–891).: IEEE.

[33] Gilbert, R. (2011). The prose project: A program of in-world behavioral research on the metaverse. *Journal of Virtual Worlds Research*, 4(1), 3–18.

[34] Goesele, M., Curless, B., & Seitz, S. M. (2006). Multi-view stereo revisited. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA* (pp. 2402–2409).

[35] Goesele, M., Fuchs, C., & Seidel, H. (2003). Accuracy of 3d range scanners by measurement of the slanted edge modulation transfer function. In *4th International Conference on 3D Digital Imaging and Modeling, 6-10 October 2003, Banff, Canada* (pp. 37–45).

[36] Goesele, M., Snavely, N., Curless, B., Hoppe, H., & Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007* (pp. 1–8).

[37] Harltey, R. & Zisserman, A. (2006). *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press.

[38] Hart, J. C. (1993). Ray tracing implicit surfaces. *Siggraph 93 Course Notes: Design, Visualization and Animation of Implicit Surfaces*, (pp. 1–16).

[39] Hasenfratz, J., Lapierre, M., & Sillion, F. X. (2004). A real-time system for full body interaction with virtual worlds. In *Proceedings of the 10th Eurographics Symposium on Virtual Environments, EGVE 2004, Grenoble, France, June 8-9, 2004* (pp. 147–156).

[40] He, K., Sun, J., & Tang, X. (2010). Guided image filtering. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I* (pp. 1–14).

[41] Heinly, J., Schönberger, J. L., Dunn, E., & Frahm, J. (2015). Reconstructing the world* in six days. In *IEEE Conference on Computer Vision and Pattern Recognition, 2015, Boston, MA, USA, June 7-12, 2015* (pp. 3287–3295).

[42] Herrero, S. & Bescós, J. (2009). Background subtraction techniques: Systematic evaluation and comparative analysis. In *Advanced Concepts for Intelligent Vision Systems, 11th International Conference, ACIVS 2009, Bordeaux, France, September 28 - October 2, 2009. Proceedings* (pp. 33–42).

[43] Hertzmann, A. & Seitz, S. M. (2005). Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8), 1254–1264.

[44] Hirschmüller, H. & Scharstein, D. (2009). Evaluation of stereo matching costs on images with radiometric differences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9), 1582–1599.

[45] Huang, C., Boyer, E., Navab, N., & Ilic, S. (2014). Human shape and pose tracking using keyframes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014* (pp. 3446–3453).

[46] Huang, P., Matzen, K., Kopf, J., Ahuja, N., & Huang, J. (2018). Deepmvs: Learning multi-view stereopsis. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (pp. 2821–2830).

[47] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R. A., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A. J., & Fitzgibbon, A. W. (2011a). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011* (pp. 559–568).

[48] Izadi, S., Newcombe, R. A., Kim, D., Hilliges, O., Molyneaux, D., Hodges, S., Kohli, P., Shotton, J., Davison, A. J., & Fitzgibbon, A. W. (2011b). Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2011, Vancouver, BC, Canada, August 7-11, 2011, Talks Proceedings* (pp.23).

[49] Kajiya, J. T. (1986). The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, Texas, USA, August 18-22, 1986* (pp. 143–150).

[50] Karis, B. & Games, E. (2013). Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, (pp. 621–635).

[51] Kazhdan, M. M. & Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), 29:1–29:13.

[52] Kim, H., Hunter, Q., Duchaineau, M. A., Joy, K. I., & Max, N. L. (2012). Gpu-friendly multi-view stereo for outdoor planar scene reconstruction. In *VISAPP 2012 - Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 2, Rome, Italy, 24-26 February, 2012.* (pp. 255–264).

[53] Kim, H., Sakamoto, R., Kitahara, I., Orman, N., Toriyama, T., & Kogure, K. (2007). Compensated visual hull for defective segmentation and occlusion. In *Advances in Artificial Reality and Tele-Existence, 17th International Conference on Artificial Reality and Telexistence, ICAT 2007, Esbjerg, Denmark, November 28-30, 2007, Proceedings* (pp. 210–217).

[54] Kolev, K., Klodt, M., Brox, T., & Cremers, D. (2009). Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1), 80–96.

[55] Kostrikov, I., Horbert, E., & Leibe, B. (2014). Probabilistic labeling cost for high-accuracy multi-view reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014* (pp. 1534–1541).

[56] Kurt, M. & Edwards, D. (2009). A survey of brdf models for computer graphics. *ACM SIGGRAPH Computer Graphics*, 43(2), 4.

[57] Kutulakos, K. N. & Seitz, S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision*, 38(3), 199–218.

[58] Ladikos, A., Benhimane, S., & Navab, N. (2008). Efficient visual hull computation for real-time 3d reconstruction using CUDA. In *IEEE Conference on Computer Vision and Pattern CVPR Workshops 2008, Anchorage, AK, USA, 23-28 June, 2008* (pp. 1–8).

[59] Lam, D. M., Hong, R. Z., & DeSouza, G. N. (2009). 3d human modeling using virtual multi-view stereopsis and object-camera motion estimation. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA* (pp. 4294–4299).

[60] Lambert, J. (1760). *Photometria*.

[61] Langguth, F., Sunkavalli, K., Hadap, S., & Goesele, M. (2016). Shading-aware multi-view stereo. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III* (pp. 469–485).

[62] Laurentini, A. (1994). The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2), 150–162.

[63] LaValle, S. M. (2016). *VIRTUAL REALITY*. Cambridge University Press.

[64] LaValle, S. M., Yershova, A., Katsev, M., & Antonov, M. (2014). Head tracking for the oculus rift. In *2014 IEEE International Conference on Robotics and Automation, 2014, Hong Kong, China, May 31 - June 7, 2014* (pp. 187–194).

[65] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S. E., Davis, J., Ginsberg, J., Shade, J., & Fulk, D. (2000). The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New Orleans, LA, USA, July 23-28, 2000* (pp. 131–144).

[66] Li, H., Qin, J., Xiang, X., Pan, L., Ma, W., & Xiong, N. N. (2018). An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE Access*, 6, 66963–66971.

[67] Li, S., Siu, S. Y., Fang, T., & Quan, L. (2016a). Efficient multi-view surface refinement with adaptive resolution control. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I* (pp. 349–364).

[68] Li, S., Xu, C., & Xie, M. (2012). A robust o(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7), 1444–1450.

[69] Li, Z., Wang, K., Zuo, W., Meng, D., & Zhang, L. (2016b). Detail-preserving and content-aware variational multi-view stereo reconstruction. *IEEE Trans. Image Processing*, 25(2), 864–877.

[70] Lim, H., Lim, J., & Kim, H. J. (2014). Real-time 6-dof monocular visual slam in a large-scale environment. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1532–1539).: IEEE.

[71] Lindholm, E., Nickolls, J., Oberman, S. F., & Montrym, J. (2008). NVIDIA tesla: A unified graphics and computing architecture. *IEEE Micro*, 28(2), 39–55.

[72] Locher, A., Perdoch, M., & Gool, L. V. (2016). Progressive prioritized multi-view stereo. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (pp. 3244–3252).

[73] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.

[74] Luebke, D. P. & Humphreys, G. (2007). How gpus work. *IEEE Computer*, 40(2), 96–100.

[75] Marr, D. & Poggio, T. (1979). A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156), 301–328.

[76] Martin, W. N. & Aggarwal, J. K. (1983). Volumetric descriptions of objects from multiple views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5(2), 150–158.

[77] Mather, G. (2009). *Foundations of Sensation and Perception*. Psychology Press.

[78] Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., & McMillan, L. (2000). Image-based visual hulls. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New Orleans, LA, USA, July 23-28, 2000* (pp. 369–374).

[79] Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., & Zhang, X. (2011). On building an accurate stereo matching system on graphics hardware. In *IEEE International Conference on Computer Vision Workshops, 2011 Workshops, Barcelona, Spain, November 6-13, 2011* (pp. 467–474).

[80] Merrell, P. & Manocha, D. (2011). Model synthesis: A general procedural modeling algorithm. *IEEE Trans. Vis. Comput. Graph.*, 17(6), 715–728.

[81] Musgrave, F. K., Kolb, C. E., & Mace, R. S. (1989). The synthesis and rendering of eroded fractal terrains. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1989, Boston, MA, USA, July 31 - August 4, 1989* (pp. 41–50).

[82] Muzzupappa, M., Gallo, A., Spadafora, F., Manfredi, F., Bruno, F., & Lamarca, A. (2013). 3d reconstruction of an outdoor archaeological site through a multi-view stereo technique. In *2013 Digital Heritage International Congress, Marseille, France, October 28 - November 1, 2013, Volume I* (pp. 169–176).

[83] Newcombe, R. A., Lovegrove, S., & Davison, A. J. (2011). DTAM: dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011* (pp. 2320–2327).

[84] Nguyen, T., Huynh, H., & Meunier, J. (2018). 3d reconstruction with time-of-flight depth camera and multiple mirrors. *IEEE Access*, 6, 38106–38114.

[85] Noborio, H., Fukuda, S., & Arimoto, S. (1988). Construction of the octree approximating a three-dimensional object by using multiple views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6), 769–782.

[86] Nvidia (2015). *CUDA C PROGRAMMING GUIDE*.

[87] of Standards, U. S. N. B. & Nicodemus, F. E. (1977). *Geometrical considerations and nomenclature for reflectance*, volume 160. US Department of Commerce, National Bureau of Standards.

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

[89] Oren, M. & Nayar, S. K. (1993). Diffuse reflectance from rough surfaces. In *Conference on Computer Vision and Pattern Recognition, CVPR 1993, 15-17 June, 1993, New York, NY, USA* (pp. 763–764).

[90] Oren, M. & Nayar, S. K. (1994). Generalization of lambert's reflectance model. In *Proceedings of the 21th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994, Orlando, FL, USA, July 24-29, 1994* (pp. 239–246).

[91] Oxholm, G. & Nishino, K. (2014). Multiview shape and reflectance from natural illumination. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014* (pp. 2163–2170).

[92] Paragios, N., Chen, Y., & Faugeras, O. D., Eds. (2006). *Handbook of Mathematical Models in Computer Vision*. Springer.

[93] Poms, A., Wu, C., Yu, S., & Sheikh, Y. (2018). Learning patch reconstructability for accelerating multi-view stereo. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (pp. 3041–3050).

[94] Potmesil, M. (1987). Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40(1), 1–29.

[95] Ranftl, R., Gehrig, S., Pock, T., & Bischof, H. (2012). Pushing the limits of stereo using variational stereo estimation. In *2012 IEEE Intelligent Vehicles Symposium, IV 2012, Alcal de Henares, Madrid, Spain, June 3-7, 2012* (pp. 401–407).

[96] Raskar, R., Nii, H., de Decker, B., Hashimoto, Y., Summet, J., Moore, D., Zhao, Y., Westhues, J., Dietz, P. H., Barnwell, J., Nayar, S. K., Inami, M., Bekaert, P., Noland, M., Branzoi, V., & Bruns, E. (2007). Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators. *ACM Trans. Graph.*, 26(3), 36.

[97] Riva, G. (2005). Virtual reality in psychotherapy. *Cyberpsychology & behavior*, 8(3), 220–230.

[98] Sadjadi, F. & Ribnick, E. (2010). Passive 3d sensing, and reconstruction using multi-view imaging. In *IEEE Conference on Computer Vision and Pattern Recognition, Workshops 2010, San Francisco, CA, USA, 13-18 June, 2010* (pp. 68–74).

[99] Samet, H. (1995). Spatial data structures. In *Modern Database Systems: The Object Model, Interoperability, and Beyond.* (pp. 361–385).

[100] Saouli, A. & Babahenini, C. M. (2016). High performance volumetric modelling from silhouette: Gpu-image-based visual hull. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)* (pp. 1–7).: IEEE.

[101] Saouli, A. & Babahenini, M. C. (2018). Towards a stochastic depth maps estimation for textureless and quite specular surfaces. In *ACM SIGGRAPH 2018 Posters* (pp.72).: ACM.

[102] Saouli, A., Babahenini, M. C., & Medjram, S. (2019). Accurate, dense and shading-aware multi-view stereo reconstruction using metaheuritic optimization. *Multimedia Tools Appl.*, 78(11), 15053–15077.

[103] Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3), 7–42.

[104] Schick, A. & Stiefelhagen, R. (2009). Real-time gpu-based voxel carving with systematic occlusion handling. In *Pattern Recognition, 31st DAGM Symposium, Jena, Germany, September 9-11, 2009. Proceedings* (pp. 372–381).

[105] Schlick, C. (1994). A survey of shading and reflectance models. *Comput. Graph. Forum*, 13(2), 121–131.

[106] Schönberger, J. L. & Frahm, J. (2016). Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (pp. 4104–4113).

[107] Schönberger, J. L., Zheng, E., Frahm, J., & Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III* (pp. 501–518).

[108] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA* (pp. 519–528).

[109] Seitz, S. M. & Dyer, C. R. (1999). Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2), 151–173.

[110] Semerjian, B. (2014). A new variational framework for multiview surface reconstruction. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI* (pp. 719–734).

[111] Shah, R., Srivastava, V., & Narayanan, P. J. (2015). Geometry-aware feature matching for structure from motion applications. In *2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015, Waikoloa, HI, USA, January 5-9, 2015* (pp. 278–285).

[112] Shen, S. (2013). Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE Trans. Image Processing*, 22(5), 1901–1914.

[113] Sherman, W., Sherman, W., & Craig, A. (2003). *Understanding Virtual Reality: Interface, Application, and Design*. Morgan Kaufmann series in computer graphics and geometric modeling. Elsevier Science.

[114] Sinha, S. N., Mordohai, P., & Pollefeys, M. (2007). Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007* (pp. 1–8).

[115] Sinha, S. N. & Pollefeys, M. (2005). Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China* (pp. 349–356).

[116] Sinha, S. N., Steedly, D., Szeliski, R., Agrawala, M., & Pollefeys, M. (2008). Interactive 3d architectural modeling from unordered photo collections. *ACM Trans. Graph.*, 27(5), 159:1–159:10.

[117] Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3), 835–846.

[118] Strecha, C., Fransens, R., & Gool, L. J. V. (2004). Wide-baseline stereo from multiple views: A probabilistic account. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), with CD-ROM, 27 June - 2 July 2004, Washington, DC, USA* (pp. 552–559).

[119] Strecha, C., Fransens, R., & Gool, L. J. V. (2006a). Combined depth and outlier estimation in multi-view stereo. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA* (pp. 2394–2401).

[120] Strecha, C., Fransens, R., & Gool, L. J. V. (2006b). Combined depth and outlier estimation in multi-view stereo. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA* (pp. 2394–2401).

[121] Strecha, C., von Hansen, W., Gool, L. J. V., Fua, P., & Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*.

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

[123] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London.

[124] Torrance, K. E. & Sparrow, E. M. (1967). Theory for off-specular reflection from roughened surfaces. *Josa*, 57(9), 1105–1114.

[125] Tylecek, R. & Sara, R. (2009). Depth map fusion with camera position refinement. In *Computer Vision Winter Workshop*, volume 70 (pp. 79–83).

[126] Ulusoy, A. O., Biris, O., & Mundy, J. L. (2013). Dynamic probabilistic volumetric models. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013* (pp. 505–512).

[127] Vince, J. (2004). *Introduction to Virtual Reality*. Springer London.

[128] Vu, H., Labatut, P., Pons, J., & Keriven, R. (2012). High accuracy and visibility-consistent dense multiview stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5), 889–901.

[129] Waechter, M., Beljan, M., Fuhrmann, S., Moehrle, N., Kopf, J., & Goesele, M. (2017). Virtual rephotography: Novel view prediction error for 3d reconstruction. *ACM Trans. Graph.*, 36(1), 8:1–8:11.

[130] Wei, J., Resch, B., & Lensch, H. P. A. (2014). Multi-view depth map estimation with cross-view consistency. In *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*.

[131] WEN-CHAO CHEN, ZEN CHEN, P.-Y. S. (2015). Stochastic optimization based 3d dense reconstruction from multiple views with high accuracy and completeness. *Journal of Information Science and Engineering*.

[132] Wu, C., Wilburn, B., Matsushita, Y., & Theobalt, C. (2011). High-quality shape from multi-view stereo and shading under general illumination. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011* (pp. 969–976).

[133] Yang, L., Sheng, Y., & Wang, B. (2016). 3d reconstruction of building facade with fused data of terrestrial lidar data and optical image. *Optik-International Journal for Light and Electron Optics*, 127(4), 2165–2168.

[134] Zhan, Y., Gu, Y., Huang, K., Zhang, C., & Hu, K. (2016). Accurate image-guided stereo matching with efficient matching cost and disparity refinement. *IEEE Trans. Circuits Syst. Video Techn.*, 26(9), 1632–1645.

[135] Zhang, J. & Singh, S. (2015). Visual-lidar odometry and mapping: low-drift, robust, and fast. In *IEEE International Conference on Robotics and Automation, 2015, Seattle, WA, USA, 26-30 May, 2015* (pp. 2174–2181).

[136] Zheng, E., Dunn, E., Jojic, V., & Frahm, J. (2014). Patchmatch based joint view selection and depthmap estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014* (pp. 1510–1517).

[137] Zhu, Z., Stamatopoulos, C., & Fraser, C. S. (2015). Accurate and occlusion-robust multi-view stereo. *ISPRS journal of photogrammetry and remote sensing*, 109, 47–61.

[138] Zollhöfer, M., Dai, A., Innmann, M., Wu, C., Stamminger, M., Theobalt, C., & Nießner, M. (2015). Shading-based refinement on volumetric signed distance functions. *ACM Trans. Graph.*, 34(4), 96:1–96:14.


LESIA
Laboratoire