DEPARTMENT OF
INFORMATION
ENGINEERING
UNIVERSITY OF PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI DI DOTTORATO

ACQUIRING SHAPE AND MOTION OF INTERACTING PEOPLE FROM VIDEOS

Luca Ballan

January 27, 2009

Abstract

Passive 3D reconstruction of dynamic scenes from multiple video sequences is a challenging problem in computer vision. The aim is to recover a mathematical time-varying description of the whole 3D scene using only videos recorded by some cameras.

In this thesis, we present a system for recovering the shape, the appearance and the motion of a scene where multiple people and objects interact with each other, using only passive and non-invasive techniques. The system, then, gives a user the possibility to navigate inside such a representation and to replay the action from any point of view. Moreover, the output can be easily used to animate virtual characters using any commercial animating software.

The acquisition takes place in two separate steps: the former consists in the acquisition of the shape and the appearance of each actor using a homemade passive body scanner, while, the latter captures their motions and the motions of all the objects they interact with, using a marker-less motion capture system.

We propose a new optimization framework for the pose estimation problem capable of handling, simultaneously and in a unified way, multiple entities in the same scene. This framework takes into account the non-rigid deformations of the actors' skin allowing an accurate pose estimation of also the small and high flexible parts of the body, like the spine and the clavicles. Moreover, multiple occlusions and possible lacks of information are avoided by the synergical use of two distinct type of motion cues, namely optical flow and silhouette information. We also avoid any kind of time consuming 3D reconstruction task during the estimation.

The validation of the entire system is made testing several sequences recorded by four synchronized video cameras and representing many different types of motions starting from the single person ones to the multiple people and multiple objects interactions sequences. The experiments show the validity of our approach and of our choices through both qualitative and quantitative measurements, even on sequences with more than 80 degrees of freedom. In particular, the average shape reconstruction error achieved was below $0.42mm$ and the average pose estimation error was below $2.5cm$.

Sommario

La ricostruzione passiva di scene dinamiche da più sequenze video è un problema fortemente discusso all'interno della comunità di computer vision. Lo scopo è quello di creare una descrizione matematica spaziale e temporale dell'intera scena utilizzando solamente video registrati da alcune telecamere.

In questa tesi, presenteremo un sistema per acquisire la forma, l'apparenza e il moto presenti all'interno di una scena nella quale più persone ed oggetti interagiscono tra loro, utilizzando solamente tecniche passive e non invasive. Il sistema, dunque, fornisce all'utente la possibilità di navigare in questo tipo di rappresentazione e di riprodurre l'azione da un qualsiasi punto di vista. Inoltre, il risultato può essere facilmente utilizzato per animare personaggi virtuali tramite un qualsiasi software di animazione.

L'acquisizione avviene in due fasi distinte: la prima consiste nell'acquisizione della forma e dell'apparenza di ciascun attore attraverso un body scanner passivo da noi realizzato, mentre, la seconda cattura i loro movimenti ed i movimenti di tutti gli oggetti con i quali interagiscono, per mezzo di un sistema marker-less di cattura del moto.

In questa tesi proporremo un nuovo modello di ottimizzazione per il problema della stima della posa capace di maneggiare, simultaneamente e allo stesso modo, molteplici entità nella medesima scena. Questo modello prende in considerazione le deformazioni non rigide delle superfici dei soggetti, permettendo anche una stima accurata della posa di parti del corpo piccole e flessibili, come la spina dorsale e le clavicole. Inoltre, occlusioni multiple e possibili carenze informative vengono evitate attraverso l'uso congiunto di due tipi distinti di indicatori di movimento, rispettivamente optical flow e le silouhettes. Viene inoltre evitato, durante la stima, una qualsiasi ricostruzione 3D della scena che richiederebbe molto tempo.

La convalida dell'intero sistema è ottenuta valutando il comportamento dell'algoritmo in diverse sequenze riprese da quattro videocamere sincronizzate tra loro e rappresentanti vari tipi di movimento da quelli che riguardanti singoli soggetti, fino a quelli concernenti sequenze di interazione tra più soggetti ed oggetti. Gli esperimenti dimostrano la validità del nostro approccio e delle scelte effettuate, attraverso misurazioni qualitative e quantitative, persino in sequenze riscontranti un numero di gradi di libertà superiore ad 80. Nello specifico, l'errore medio di ricostruzione ottenuto è risultato inferiore a $4.2mm$ e l'errore medio di stima della posizione inferiore a $2.5cm$.

Contents

Chapter 1

Introduction and Motivations

Passive 3D reconstruction of dynamic scenes from multiple video sequences is a challenging problem in computer vision that has attracted the interest of many researchers during the last decade and moreover, in the recent years, also the interest of the computer graphics community. The aim is to recover a mathematical time-varying description of the whole 3D scene using only the information extracted from video sequences recorded by some cameras arranged inside the environment.

The typical application of this data is the generation of 3D-video or free viewpoint video contents to be used as final result or as intermediate information for further analysis, like, for instance, an high level comprehension of the whole scene. Fields interested in these technologies are entertainment (movies and videogames), medicine, surveillance, ergonomics and biomechanics, just to name a few.

Three-dimensional video (3D-video) and free viewpoint video (FVV) are new types of media that expand the user experience beyond what is offered by traditional media. 3D-Video offers a 3D depth impression of the observed scene (the so called stereo parallax), while FVV allows for an interactive selection of viewpoint and direction within a certain operating range (movement parallax). Both are not mutually exclusive, on the contrary, they can be combined within a single system, since they are both based on a suitable 3D scene representation format.

FVV media is used in the so called virtualized reality consisting in an immersive experience of a real event where each spectator can dynamically change his point of view during the showing. In case of realtime events, this experience is also known as tele-immersion experience and easily find application in surveillance and in medicine.

3D-video and FVV media contents are instead well desired by the broadcasting and advertising companies which see in these technologies the future of the classical television, moving from the 2D image concept to the 3D interactive one, i.e., the so called 3DTV.

Indeed, the capability of choosing different point of view to watch sport events like soccer matches or car races, as well as the possibility to access instantaneously to the statistics or other data related to the event capture the attention and the wallets of a lot of consumers. Companies like LiberoVision[1] and 3DEverywhere[2] have currently developed two software to generate these kind of services.

In the videogames community, the next step is to give the possibility to personalize the main character of a story according to some real characteristics of the player and link the character movements directly to the player's one. The Nintendo Wii[3] system is a typical example of this trend.

Clearly, in order to generate these kind of information a partial or a complete model of the whole event has to be acquired and in this case, passive 3D reconstruction techniques play a crucial role. This justifies the constant interest of the computer vision community on this topic and the publication of more than 350 works, in this area, between 2000 and 2006.

Literature proposes various solutions classifiable basing on their final representation of the scene. In fact, the acquired scene can be either described by voxels or by time-varying and coherent surface/volume representation, i.e., the so called deformable models. In particular, in this last case, acquired scene data can be compressed and acquired information distinguished in shape, appearance and motion. Shape information describe the geometries of the objects inside the scene, appearance their reflectance proprieties, and finally motion information describe their dynamics.

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

[2] http://www.3deverywhere.com/

[3] http://www.nintendo.com/

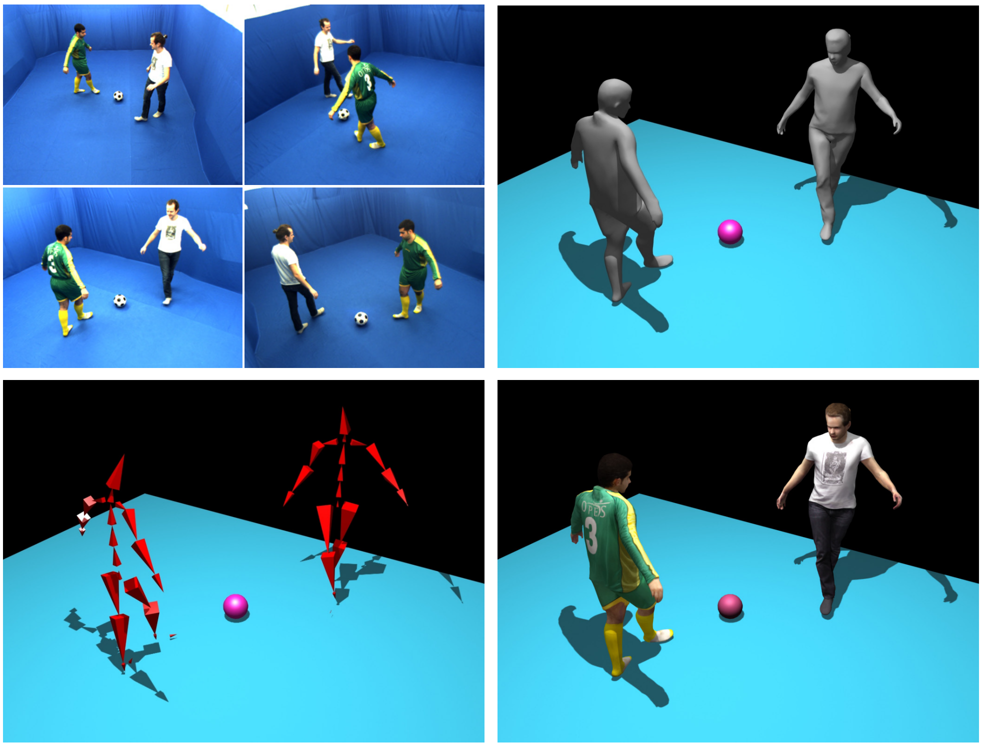


Figure 1.1: *Representation of the scene. In row major order: images of the scene captured by some cameras, mesh representation of the scene, skeletal structures of the articulated entities present in the scene, complete representation of the scene.*

1.1 Aim and contributions

In this thesis we present a system for capturing shape, appearance and motion of interacting people and objects using only passive and non-invasive techniques. Given a scene, where multiple people are interacting with each other and with some other objects, our system is able to provide a time-varying description of the whole 3D sequence considering both its geometry and its appearance. A user is therefore able to navigate inside this representation and look at the action from any point of view. In particular, the scene geometry is modeled by time consistent meshes and some skeletal structures connected to the articulated entities of the scene. Its appearance, instead, is modeled with simple Lambertian color information (see Fig. 1.1).

The acquisition is performed in two separate steps. First the shape and the appearance of each actor is acquired using a passive body scanner. Subsequently, the actors are invited inside a second location where the actual action will take place. A marker-less motion capture system is used to simultaneously capture their motions and the motion of all the objects which they interact with. More precisely, this last system estimates the pose assumed by each actor and each object in all the frames of the recorded action.

Three are the main contributions of this thesis. The first is the definition of an optimization framework for the pose estimation capable of handling, simultaneously and in a unified way, multiple entities interacting in the same scene. This framework also take into account the non-rigid deformations of the actors' skin allowing an accurate pose estimation of also the small and high flexible parts of the body, like the spine and the clavicles, which are often neglected by the classical approaches.

The second main contribution is due to the synergic use of two distinct sources of information extracted from the videos to overcome possible lacks of data given by the use of a few number of cameras recording the scene and the high number of degrees of freedom to estimate. Moreover, our formulation also avoids time consuming tasks like full 3D reconstructions and it is designed to be parallelized.

The last contribution consists in the designing and the making of an inexpensive passive body scanner capable to acquire both shape and appearance information of a human with an average accuracy of $0.42cm$.

A minor contribution is given by the introduction of a unified concept of deformable model which opens new prospective for future extension of this work.

To conclude, the system proposed by this thesis is able to recover scenes with more than 80 degrees of freedom using only four cameras recording the action.

Parts of this dissertation have been published in conference papers. The main algorithm for the pose estimation was proposed in [8]. [10] covers parts of passive modeling pipeline used inside the body scanner, while the texture reconstruction part was proposed in [21].

1.2 Outline of the thesis

Chapter 2 makes an overview of the most important approaches on this area classifying both the body shape acquisition systems and the motion capture systems. Our solution is then proposed in Sec. 2.2. The motivations of our choices are also described in this section.

Chapter 3 introduces our body scanning system and the subsequent Chapter 4 explains the theory behind the proposed motion capture system underlining its advantages and drawbacks with respect to the older approaches. In this chapter, it is also present a useful formalism to treat with articulated deformable models.

Chapter 5 explains all the problematics concerning the making of a multi-camera recording room for motion capture purposes. The adopted solution is described in details.

Chapter 6 describes the tests performed on our system and shows the obtained results. Finally, Chapter 7 draws the conclusions.

This thesis proposes also two appendix chapters describing respectively the state of the art of passive 3D reconstruction (Appendix A) and the state of the art of the digital keying techniques (Appendix B). In particular, Appendix A is a chapter extracted from the book "*3D ONLINE MULTIMEDIA AND GAMES: Processing, Visualization and Transmission*" [11].

Chapter 2

Related Works and Our Approach

2.1 Related works

This section describes and tries to classify the most important commercial and non-commercial solutions to the shape, appearance and motion acquisition problem. Shape and appearance capturing approaches are first described and, subsequently, a deep review on the existing motion capture systems is made.

2.1.1 Shape and appearance capturing systems

Shape and appearance capturing systems aim to mathematically model the shape and the reflectance proprieties of an existing object. In case of systems designed specifically to capture humans, the term *body scanner* is preferred.

Humans are particular examples of deformable models, i.e., objects having more than one single shape and more than one single appearance. These characteristics, in fact, depend on the poses they are assuming. A body scanner aims to acquire one specific shape and one specific appearance at a time, more precisely, those related to the pose assumed during the acquisition.

In general, a shape and appearance capturing system can be either with contact or contactless, reflective or transmissive, optical or non-optical, passive or active. A complete taxonomy of these systems can be found in the introductive chapter of [7].

Body scanners are typically active optical capturing systems based on laser or structured light. They usually acquire only the shape of the human without considering his appearance. This is partially due to the fact that, their optimal working conditions are not optimal for the acquisition of the human's appearance.

Commercial body scanner are proposed by several companies like Vitronic[1], Cyberware[2], Unique Patterns[3], Shape Analysis[4], Hometrica Consulting[5], RSI[6], 3D Metrology Solutions[7] and L3[8]. All their solutions consist in active systems.

Passive body scanners, instead, are not yet commercialized since their accuracy is usually lower than their active counterpart. However, they offer some advantages. First, they do not require any interaction with the object to acquire, neither by irradiation. This, for instance, allows the human, to be acquired, to keep his eyes open during the acquisition, action absolutely forbidden in all the laser based systems because the retina can be damaged by the beam.

Moreover, in passive systems, the acquisition consists in taking some pictures all around the whole object, which is usually faster than in other type of systems. This, unfortunately, leads to the drawback of a more complex data processing step aimed to infer the actual 3D shape from the acquired images. For a review on the most important passive optical techniques the reader should refer to Appendix A.

Concerning the body scanners, the cost of a passive one is relatively contained with respect to an active one. The commercial active products, previously mentioned, have a cost of about hundreds of thousands euros while we will demonstrate, in this thesis, that it is possible to build a passive system with less than two thousand euros.

Some important work concerning the design of a passive body scanner are made in [139], [126] and in [80].

2.1.2 Motion capture systems: an overview

Motion capture (mocap) refers to the process of estimating the movements of an actor and translating them onto a digital model. It usually aims to capture the movements of the skeletal structure but it can also be applied for acquiring skin deformations and facial expressions.

[1] http://www.vitronic.de/

[2] http://www.cyberware.com/

[3] http://www.uniquepatterns.com/

[4] http://www.shapeanalysis.com/

[5] http://www.hometrica.ch

[6] http://www.rsi.gmbh.de/

[7] http://3dmetrologysolutions.com/

[8] http://www.dsxray.com/

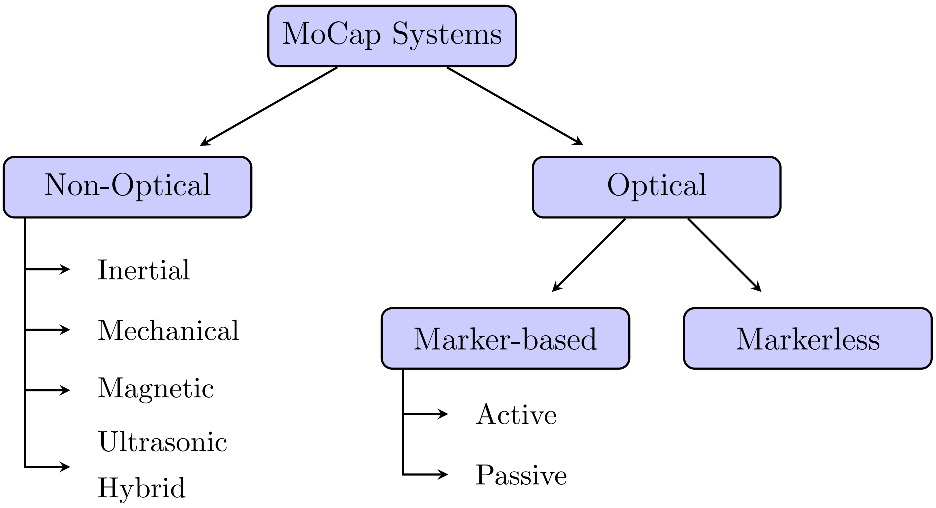


Figure 2.1: *Motion capture systems taxonomy.*

Mocap systems find application in a variety of fields like entertainment, sports, medical applications, ergonomics, bio-mechanical analysis, military and surveillance. In film-making they are used for recording the performance of human actors so that, the acquired motion can be used to animate digital characters. Commercial mocap solutions are currently developed by several companies like Animazoo[9], Measurand[10], Organic Motion[11], BioVision[12], META Motion[13] and Vicon[14].

The different motion capture systems can be classified according to the scheme depicted in Figure 2.1. A mocap system can be either optical or non-optical. The former infers the subject's pose using only the videos recorded by some cameras, while, the latter, uses complex devices, worn by the actor, capable to reveal their positions inside the capture environment. In both cases, the acquired data is processed by a central unit which actually estimates the motion.

Non-optical mocap systems are further classified according to the type of the used capture devices namely, inertial, mechanical, magnetic or ultrasonic. The most popular and cost-efficient devices are the inertial ones which consist in small systems capable to measure their instantaneous motion direction and

[9] http://www.animazoo.com/

[10] http://www.motion-capture-system.com/

[11] http://www.organicmotion.com/

[12] http://www.biovision.com/

[13] http://www.metamotion.com/

[14] http://www.vicon.com/

velocity. This data is then processed by the central unit which, using the past measurements, retrieves the current devices locations. The most popular commercial mocap system based on inertial devices is the Nintendo Wii[15] console. Mechanical mocap systems instead, make the use an exoskeleton to directly measure the angles of each body joint. These systems are real-time, free-of-occlusion and relatively low-cost but their rigid structures limit the actor movements and make them very uncomfortable. On the contrary, magnetic and ultrasonic hybrid mocap systems use small wireless devices capable to compute their relative position and orientation by measuring respectively the magnetic flux or the ultrasound intensities. The central unit then, retrieve the absolute position of each device.

Optical mocap systems can be subdivided in two main classes namely, the marker-based and the markerless one. In the former case, the actor is forced to wear special markers in specific location of his body. These markers are revealed by the image sensors and their positions triangulated to recover the actual pose of the actor. Markers ca be either active or passive typically made by respectively LEDs or reflective materials. Marker-based systems are less invasive than the non-optical ones but they still require to be used in very controlled environments.

The best non-invasive solution comes from the markerless motion capture systems (MMC) which tracking technologies are purely based on the videos recorded by some cameras without requiring any restrictions on the actor and sometimes neither on the capture environment. An exhaustive analysis of these last technologies is made in the following section.

2.1.3 Markerless motion capture systems

MMC systems offer a very attractive and non-invasive solution for motion capturing since they are not restricted to the motion information associated with the markers and relieve users from the inconvenience of wearing special garments or devices. Moreover, ideally, they can be applied in any type of environment, from the simple closed room to the more crowded urban area. These characteristics pave the way to new applications in fields like surveillance and medical analysis where the controlled subjects and the patients should be unaware of being observed.

Even if some commercial MMC solutions exist, these technologies are still at the research level as proved by the enormous number of publications in this area

[15] http://www.nintendo.com/

during the last decade. Two famous surveys, namely [105] and [104], list over 350 published works on this topic from 2000 to 2006 and about 130 from 1980 to 2000.

According to [105], MMC is related to four distinct problems which can be considered as the four separate steps involved in a MMC recording session namely, the initialization, the tracking, the pose estimation and, if needed, the recognition step, respectively defined as follows:

- **The initialization** consists in the definition of the humanoid model of the subject to be tracked, namely its shape, its kinematic structure, its appearance and its initial pose. This is the first step of a recording session and it is often performed semiautomatically. However, not all the MMC systems need to compute these information during the initialization step.
- **The tracking** consists in the detection and in the pursuit of the humans acting inside the scene. Currently, literature counts more than 2000 algorithms aiming to people tracking, nevertheless, this remains an open problem for the vision community, especially when the tracking is applied to crowded and uncontrolled urban areas.
- **The pose estimation** is the core of any MMC system and consists in the estimate of the configuration of the underlying kinematic at each frame of the recorded sequence.
- **The recognition** is an optional step which aims to give an interpretation of the recorded scene. Actions and activities are recognized and classified according to some hierarchies transforming the scene into a sentence of a language of actions. From an application point of view, these results could be useful for surveillance, medical studies, robotics, video indexing and HCI. In particular, in surveillance, the fact that each action can be classified as regular or not can be used to recognize potentially dangerous situations. Notable pioneering works in this area were developed by Nagel [107] in 1988 and by Neumann [109] in 1989.

The next section focuses on the pose estimation problem describing and trying to classify the most important approaches, present in literature, aimed to solve this problem.

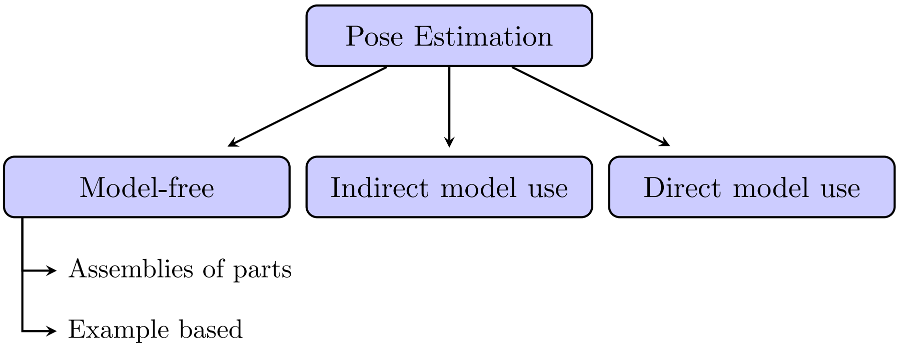


Figure 2.2: *Pose estimation algorithms taxonomy.*

2.1.4 Pose estimation

Several characteristics can be used to classify a pose estimation algorithm. These algorithms indeed, can be design specifically for capturing one single person at a time or for capturing multiple people simultaneously. Algorithms which require only one camera to perform the estimate are called *monoscopic algorithms* while all the others are called *multiview algorithms*. *2D pose estimation algorithms* estimate only the 2D pose of the human in the image space without giving any information about its depth. On the contrary, *3D pose estimation algorithms* estimate the full 3D pose. Pose estimation can be performed on each singular frame separatively without using any information about the previous or the subsequent frames. Algorithms adopting this strategy are called *pose detection algorithms*. On the contrary, algorithms using the information about the previous frames are called *pose tracking algorithms*. Obviously the pose at the first frame of a sequence, i.e., the initial one, cannot be estimate using a pose tracking algorithm. This pose is usually defined manually or by an eristic approach during the initialization step but it can be computed using a pose detection algorithm.

However, the most significant characteristic separating all these algorithms is the required a-priori information about the subjects to be captured. According to this, pose estimation algorithms can be classified in three main classes namely, the *Model-free* algorithms, the algorithms with *direct model use* and the algorithms with *indirect model use* (see Fig. 2.2).

Model-free algorithms do not use any explicit a-priori information about the actor. Their task is performed using one of following strategies: probabilistic assemblies of parts or example-based approach. In the first case, each body part is located inside the image and then the entire body structure is assembled to obtain the configuration which best matches the observations. The result is usually a 2D pose in the image space[16]. On the other side, example-based approaches use learning techniques to compute the map from the images depicting a human in a specific pose to the 3D pose itself. The training set is usually build by rendering a virtual model in several different poses. Hidden Markov models are then used to represent the temporal constraints. Model-free algorithms can be either tracking or detection algorithms and they are usually adopted for multiple people pose estimations. However, their accuracy is lower than the one achieved by the direct model use algorithms.

Recent works in this category aiming the capture the pose of multiple people are [2], [79], [125], [170] and [57]. In particular, the latter one uses two cameras to track and capture the motion of the people in the center of Zurich basing on an example-based approach. In [2] instead, the assemblies of parts strategy is used to detect and track humans in crowded scenes without focusing on the pose estimation problem. [79] uses the same strategy computing also the humans' pose for indoor video sequences with multiple occluding people. [125] attempts to track humans in very long sequences assuming that people tend to take on certain canonical poses, even when performing unusual activities. Like most of the works in this category, also these works do not focus on the accuracy of the pose estimate but on the accuracy of the tracking algorithm.

Direct model use algorithms make the use of an explicit model of the person kinematics, shape and appearance. According to [105], this is the class of approaches which received most attention in the literature since they are able to achieve accuracies comparable with both the marker-based and the non-optical mocap systems. The main drawback of these approaches is the loss of adaptability since the use of an explicit model adds an extra step during the initialization, where this information needs to be acquired. Whereas this might not be a limitation in some applications like, for instance, character animation and virtual reality, it would create some inconveniences in some other applications like surveillance and medicine.

It is worth to mention the two first pioneering works in this area namely, O'Rourke and Badler [112] in 1980 and Hogg [62] in 1983. In these two works and in most of the subsequent ones, the analysis-by-synthesis approach is adopted by optimizing a functional representing the similarity between observed and estimated data. This optimization is, in general, performed by gradient descent techniques. Direct model use algorithms usually perform the motion capture on

[16] The reader should refer to [124], [166], [123], [64] and [128] for an overview on this strategy.

a single person at a time, while only few of them, like [103], try to simultaneously estimate the pose of multiple people. They usually consist in tracking approaches and therefore they are sensible to abrupt pose changes. In fact, the failure probability of a tracking algorithm strongly depends on the estimate accuracy achieved in the previous frames and on the difference between current and the previous frame pose.

To overcome to this lack, some approaches adopt stochastic techniques, like particle filtering [89] which, however, has the drawback of a considerable increase of the computational complexity. In fact, particle filtering suffer of dimensionality problems which can be reduced only by resorting to some expedients like annealed particle filter [40] or hierarchal stochastic sampling schemes [103]. In [72], a stochastic search methods was proposed to avoid the local minima that may arise after an abrupt pose change with the added benefit of a considerable computational performance improvement. Other solutions come from approaches like [56] which propose a complex parallel system adopting both a direct model use tracking algorithm and a model-free detection algorithm. The former is used to achieve good estimate accuracies, while the latter controls the error drift and avoids loss of tracks.

Differently from the direct model use approaches, the *indirect model use* ones propose to recover the necessary a-priori information about the actor's body model during the motion capture session together with the motion estimation. Belonging to this class, works like Mikic et al. [102] present integrated systems for recovering both the human body model, in this case, represented by a set of cylinders, and its motion. Model acquisition is usually based on a hierarchical rule-based labeling of the voxels of the reconstructed visual-hull. An extended Kalman filter is used to recover the human motion between frames. On the other hand, Cheung et al. [28] first reconstructs a model of the kinematic structure, shape, and appearance of a person and then use this information to estimate its motion. [147] instead, uses an alternative approach based on full 3D-to-3D non-rigid surface matching by the use of spherical mapping. The recent work Balan et al. [6] uses a database of shapes to generate the model which best fits the subject to analyze and subsequently the tracking is performed using silhouette fitting.

The next section describes the details regarding both the direct model use approaches and the indirect ones.

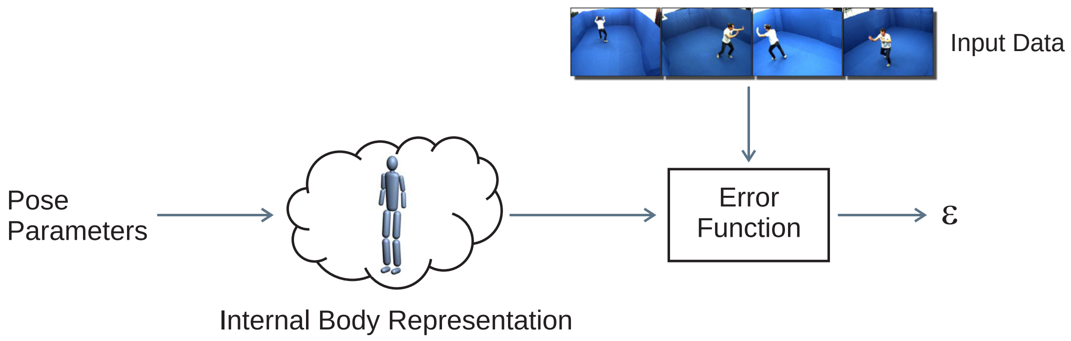


Figure 2.3: *Analysis-by-synthesis approach applied to the pose estimation problem.*

2.1.5 Direct/Indirect model use

As we mentioned before, most of the direct and the indirect model use algorithms adopt the analysis-by-synthesis approach. Figure 2.3 shows schematically this type of approach applied to the pose estimation problem. Given some parameters defining a human pose, an internal representation of the human body assuming that specific pose is generated and compared with the input data using an error functional. Clearly, the pose which minimizes this error is defined to be the best solution for the pose estimation problem.

The scheme in Fig. 2.3 suggests also a way to classify the algorithms using a direct or an indirect model, namely according to the used error function and the used internal body representation.

Concerning this latter one, the algorithms described in literature represent the human body either as shape primitives like sticks, cylinders or ellipsoids, or as complex 3D models. Figure 2.4 classifies graphically some of these representations. It is important to underline the fact that with the term internal body representation we refer to the one used during the estimation process and not the one produced by the algorithm as output. There exist, in fact, techniques using simple primitives as internal models with, instead, an output consisting in complex 3D meshes. Clearly, the more accurate is the internal body representation, the better tracking results are obtained.

A lot of works have been developed for each of the representations depicted in Fig. 2.4. Sticks, cylinders and 2D primitives were the first to be used and, nowadays, are the must suitable for monoscopic and real-time pose estimation.

On the other hand, complex 3D models are computationally expensive both to handle and to recover, but they allow to achieve better pose estimation ac-

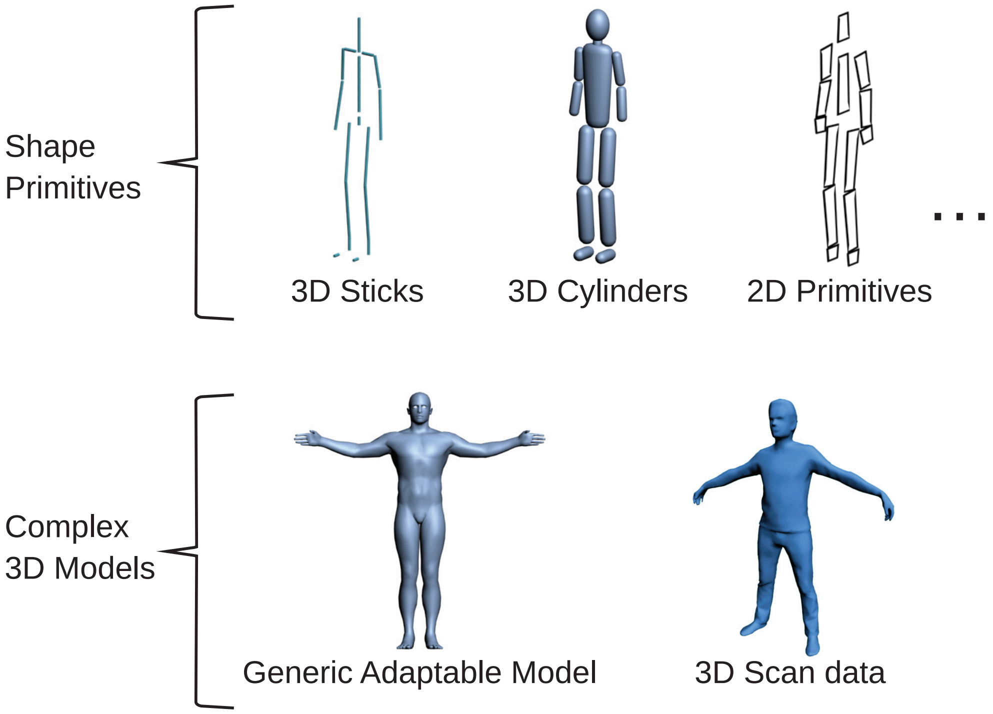


Figure 2.4: *Different internal body representations.*

curacies. In order to recover these models, two main approaches were typically used namely, the synthesis and the direct acquisition one. In the latter case, the model is acquired by a body scanner or a similar device. These hardware are rather expensive but they can achieve accuracies of the order of one millimeter. On the contrary, the synthesis approach recovers the model by adapting a generic human template to the actor characteristics, like the sizes and the lengths of each limb, or by interpolating some shapes retrieved from a database. For instance, Carranza et al. [22] adapts a generic human template during the estimation of the first frame of a recorded sequence. A database approach, instead, was successfully exploited by [6] using the so called SCAPE database [4]. The idea was to interpolate both the shape and the deformation contained in this database to generate a human, considered as a deformable model, which best match the actual actor characteristics. Unfortunately, these approaches are limited to represent only humans belonging to the space of shapes that they can span.

The most important peculiarity, characterizing each method using complex 3D models as internal representation, is how they deal with the skin deformations. Most of the works, like for instance [152] and [106], approximate these deformations splitting the human surface in small pieces, each rigidly attached to

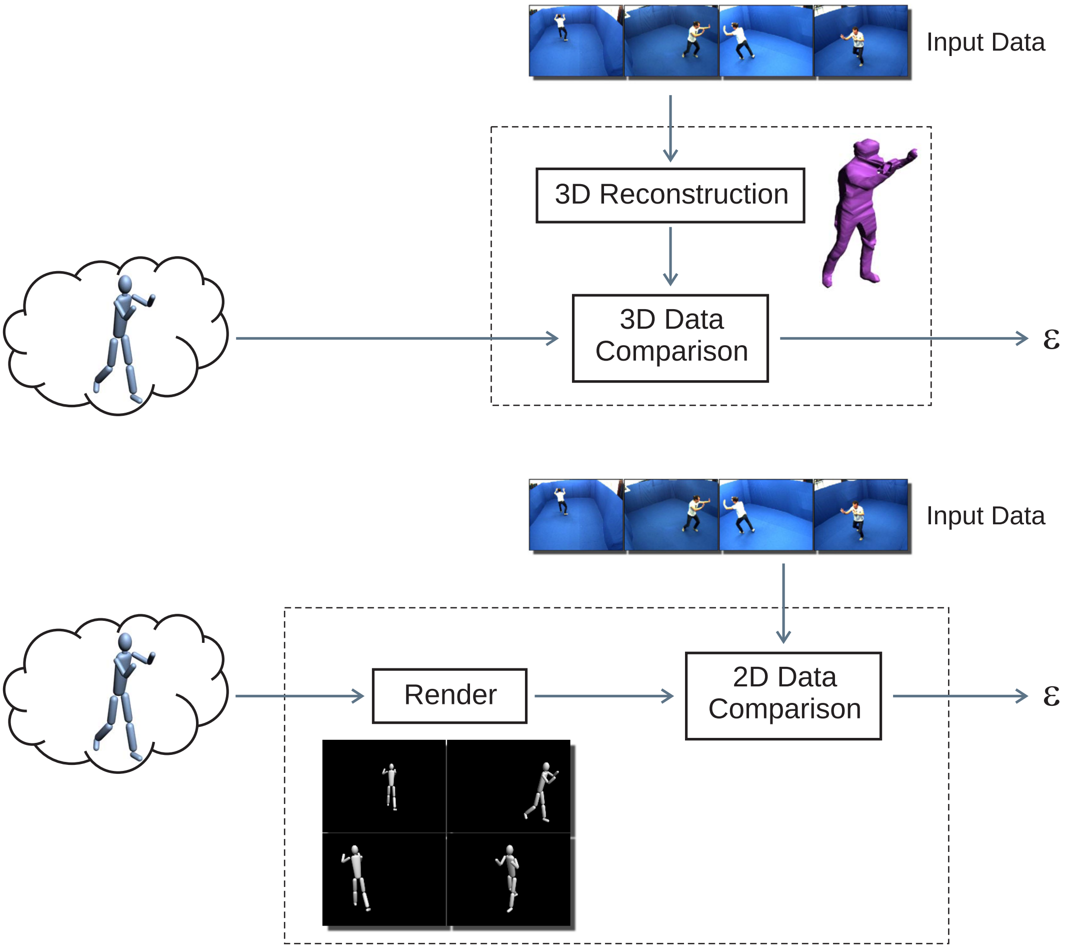


Figure 2.5: *Analysis-by-synthesis approach: (Top) 3D error functional scheme, (Bottom) 2D error functional scheme.*

only one bone of the skeleton. This solution, however, does not take into account what happen in between the junctions and it is absolutely not suitable for the motion estimating of small and high flexible parts of the body, like the spine and the clavicles. Indeed, the region of the actor's skin behaving rigidly with these bones is too small to be tracked.

Some recent works consider these deformations, but only in a subsequent step, after the pose estimation, with the only purpose of improving the output quality. For instance, [41] performs the pose estimation assuming rigid deformations but, in a further step, it uses the differential coordinates to animate a very accurate model of the actor captured using a laser scanner. [156] estimates the motion using a simple stick model, then, it captures a shape of the actor for each frame of the sequence.

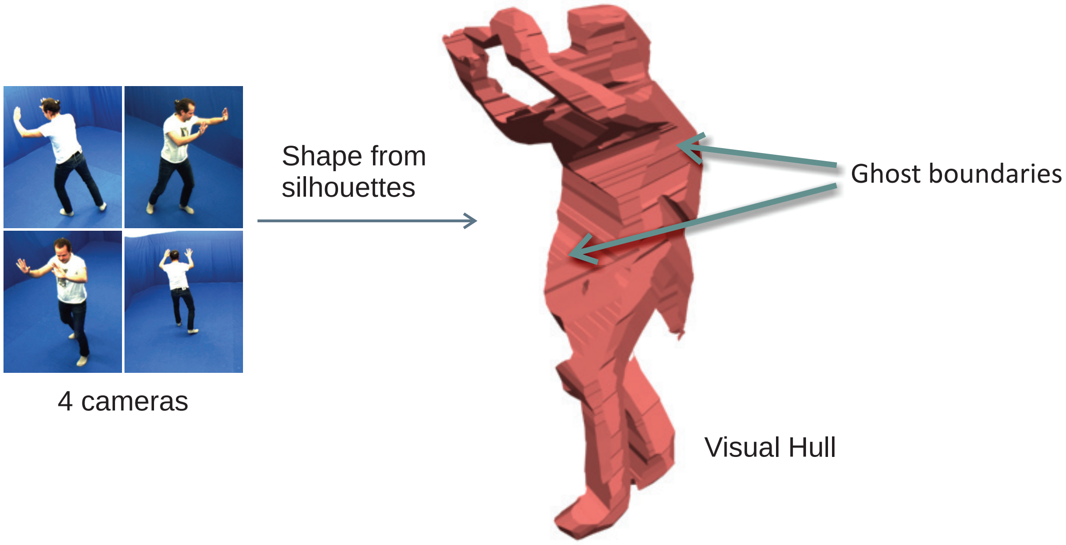


Figure 2.6: *Visual hull of the human reconstructed by four cameras: it presents a lot of ghost boundaries.*

All these approaches take into account the non-rigid deformations of the skin but in non of them, this information is used to increase the accuracy of the pose estimation process. Notable exceptions are [42], [5], [73] and [111]. [42] uses a variant of the Laplacian shape editing approach to model the deformation of both a volumetric and a mesh representation of the actor. The low frequency surface details are represented by the volumetric layer while the high frequency by the mesh layer. These latter are captured in a second refinement step using silhouettes and multi-view stereo. [5] uses the SCAPE deformation model which is in practice based on interpolation. [73] and [111] propose to use the linear blend skinning (LBS) for the deformations using however, 3D error functionals.

The error function can be classified according to the type and the domain of the features used to evaluate it. Features can be either in the 3D or in the 2D domain. Depending on the chosen features domain, the analysis-by-synthesis scheme assumes different forms depicted respectively in Figure 2.5 top and bottom. In the former case, the input images are first used to get a course 3D reconstruction of the scene and then, the human body model is fitted inside such a reconstruction minimizing a 3D matching error. In the 2D case, no explicit 3D reconstruction is generated, instead, the images of the internal body representation are synthesized during a rendering step and compared with the original ones. The error evaluation is therefore performed on the image domain.

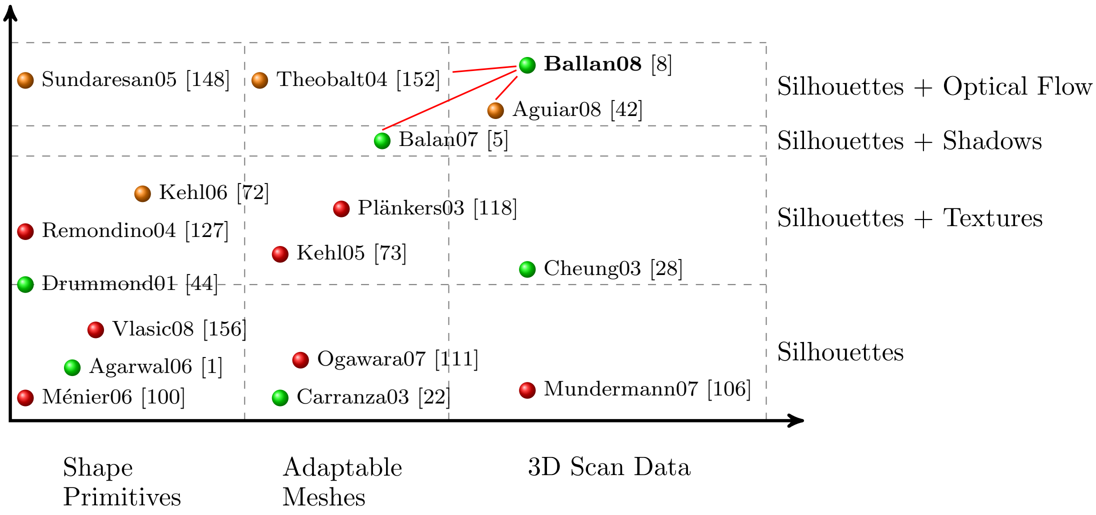


Figure 2.7: *Some of the past works organized according to their internal body representation (x-axis) and the used features type (y-axis). Red, green, yellow bullets indicate the use of respectively a 3D error function, a 2D error function and a mixed one.*

Differently from the 2D approach, the 3D one has two main drawbacks. First it needs an additional reconstruction step which is very time consuming with respect to the simply scene rendering. However, the situation in which a 3D approach clearly fails happens when silhouette information and a small number of cameras are used. In fact, since the visual hull represents only an upper-estimate of the real surface, it contains a lot of ghost boundaries which do not exists in the real actor model. Figure 2.6 shows the visual hull of a human obtained by only four cameras, the ghost boundaries are clearly visible on his back and on his chest. Clearly, if the algorithm tries to fit its own model inside such a reconstruction using a 3D matching error, it would also try to approximate all these ghost boundaries, resulting in a wrong estimation. This is cause by the fact that, in this case, the 3D matching error does not approximate the true pose estimation error.

The used feature types are typically silhouettes, textures, shadows or optical flow. Figure 2.7 and Figure 2.8 organize graphically some of the past works according to the used internal body representation, the used deformation model, the used feature types and their domains. The works visualized in these two graphs are either model-free approaches or direct/indirect model use approaches. As the reader can see from Fig. 2.7, most of the works focus on shape primitives and adaptable meshes since body scanner data is a very expensive technology.

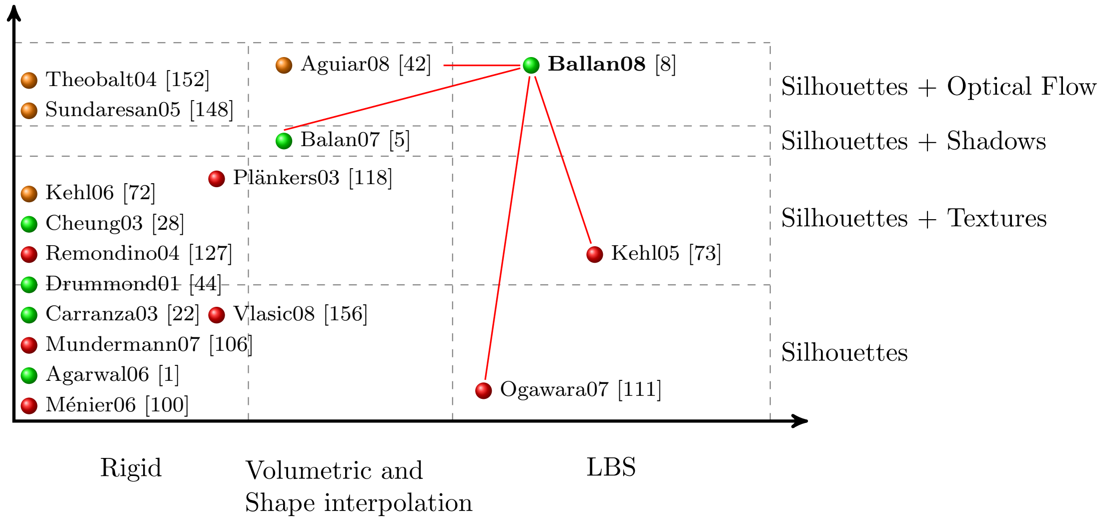


Figure 2.8: *Past works, shown in Fig. 2.7, reorganized according to how they deal with deformations (x-axis) and the used features type (y-axis).*

On the other hand, from Fig. 2.8, we can observe that most of the works assume only rigid deformations of the skin, while, only five of them, including ours, take into account the non-rigid deformations.

Concerning the number of used cameras, most of the works, performing an accurate full body pose estimation, use more than 8 cameras. Some works limit their scope to only upper body or to only approximate pose estimations. Notable works are [73] and [6], where only, respectively, 5 and 4 cameras are used. Concerning the number of degrees of freedom instead, most of the works keep it below 24, while, some works can handle also 37 DOF model.

2.2 Our approach

Our approach consists in a system able to capture shape, appearance and motion of interacting people and objects. Shape and the appearance is acquired by a passive and inexpensive body scanner in an off-line step. The motion instead, is recovered by a marker-less motion capture system. The whole scene is modeled by articulated deformable models so that the result of the entire process can be used in any commercial animating software.

According to the classification above, our mocap system belongs to the class of the *Multiple views 3D Pose estimation with Direct Model Use.* Figure 2.7 and

Figure 2.8 show the position of our approach with respect to the past works.

Given the considerations made in the previous section, our system is designed to try to incorporate all the good things of the previous works and to avoid the previously mentioned problems. In particular, a 2D error functional is used to avoid time-consuming 3D reconstruction tasks and the above mentioned problems about the visual hull (see Fig. 2.6). Both optical flow and silhouette information are exploited from the videos to overcome possible lacks of data given by the few number of cameras recording the scene (in our case, four) and the high number of degrees of freedom to estimate (in our case, more than 80). The internal body representation is a complex 3D mesh model which initial shape is acquired by the body scanner and which deformations are modeled using linear blend skinning.

As result, our formulation considers in a unified way both the two kinds of information and accounts for the non-rigid deformations of the actors' skin. In particular, non-rigid deformations are used during the pose estimation to improve its accuracy especially for the small and high flexible parts of the body. The overall accuracy of the system, instead, is maintained high by the choice of using of a complex and accurate internal body representation.

Multiple people and objects interactions are handled by our formulation in an elegant way, considering the occlusions that may arise between the characters. Differently from the previous approaches for the tracking of multiple people, our system is able to achieve accuracies comparable to the algorithms designed for a single people tracking.

The next chapter describes in details the realized body scanner while Chapter 4 deals with the pose estimation problem. In particular, Section 4.4.3 gives a more detailed comparison between our pose estimation approach and the past works.

Chapter 3

The Body Scanner

This chapter describes the body scanner developed to recover the shape and the appearance of the actors. Differently from most of the other body scanners, described either in literature or commercialized, our purpose was to design a system with a low cost and capable to capture also subjects' color.

Commercial body scanner are indeed based on active sensors which are very expensive and limit the acquisition to the only shape without considering its appearance. In their complex, these system cost hundreds of thousands of euros.

On the contrary, body scanner based, on passive sensors, are not commercialized yet but described only in literature. However, their accuracy is still not comparable with their active counterparts.

Here, we describe a passive body scanner which cost is less than two thousand euros. We start describing some of the problematics encountered during the design of this hardware and then, it is described in details. From Section 3.3 on, we describe each phase of the shape and appearance reconstruction pipeline, namely, the preprocessing, the shape reconstruction, the skeleton estimation and the appearance (texture) reconstruction. At the end, Figure 3.17 shows a results obtained with the developed body scanner.

3.1 Design of the hardware

The hardware of our body scanner consists of a mechanical system capable of taking several pictures of the actor, whose shape and appearance have to be

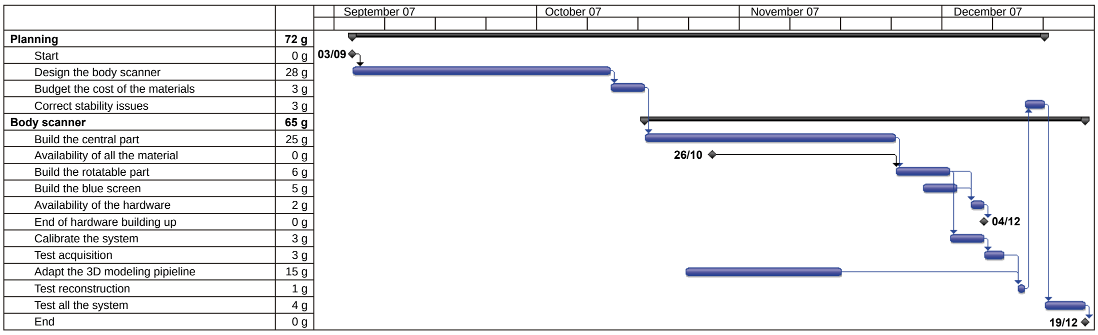


Figure 3.1: *Gantt chart describing the realization phases of our body scanner.*

acquired, in such a way that, the following requirements are satisfied:

1. *The pose of the actor is always the same in all the acquired pictures*: as previously said, a person cannot be modeled using a single geometry because his shape is not unique and depend on the pose he is assuming. The purpose of a body scanner is to recover the shape of the actor in the specific pose he had assumed during the acquisition. In our solution, the actor is forced to assume a pose similar to the Vitruvian man's one. This choice is justified by the fact that this particular pose reduces considerably the number of surface occlusions, allowing an accurate reconstruction of both shape and appearance with the few images.
2. *The silhouettes of the actor can be accurately extracted from the images*: This, in our solution, is accomplished by ensuring that a solid blue background, namely a blue screen, is present in every image and that the actor is not wearing blue clothes so that, a clear distinction between the foreground object and the background ones always exists.
3. *Each point on the surface of the actor's body is well illuminated and its color is consistent in each acquired picture*: this condition is essential to obtain reliable dense stereo results and moreover, to ensure that the recovered texture is free from illumination artifacts like low saturation regions and over or under exposure issues.
4. *The camera can be repositioned in the same place where each picture was taken*: this ensure that the system have be calibrated only once, just before the first acquisition, speeding up considerably the entire process.

The Gantt chart describing the realization phases of our body scanner is depicted in Figure 3.1. During its design, the hardest requirement to guarantee was the first one, because a common person is not able to stay still in the same pose for more than ten or twenty seconds. After this time small movements, due to external forces, attention distractions and losses of orientation, accumulate over time changing the original pose into a completely different one. In order to reduce these movements, the number of interactions between the system and the actor has to be reduced to the minimum.

In one of my first work namely [21], we used an hardware capable to acquire several pictures of small objects with the purpose of reconstructing their shapes. That system, as many other 3D modeling hardware developed by different research groups all over the world, consists of a fixed camera and a turntable. The object to acquire is placed on the turntable and while it rotates in front of the camera, some photos are taken.

Clearly, this approach is no longer suitable for our current purpose because an eventual forced rotation of the actor around his axis would involve such a big interaction which would lead to the impossibility for the actor to maintain the original pose. In fact, every person is subjected, in addition to his physical inertia, to a psychological inertia. More precisely, if a person is forced to rotate around an axis, he reacts subconsciously contrasting the rotation using, as motion information, both the visual and the equilibrium cues.

To overcome, at least partially, to the psychological inertia, a solution is to focalize the subject's attention on a object rotating according to him. However, the equilibrium cues inform his brain on what is really going on and therefore, some small corrections of the pose are still undertaken.

The best solution to overcome both the psychological and the physical inertia is to do not move the actor at all and, instead, to rotate the camera all around his body. In this way, no physical interaction is made on the subject during the acquisition process.

Unfortunately, this approach leads to a more complex solution to guarantee the second requirement, i.e., the one concerning the segmentation. Indeed, it is not trivial to ensure a solid blue background behind the actor in every image. Two solutions for this problem were considered. The former consists in taking the pictures inside a completely blue room while the latter consists in rotating an entire blue screen according to the camera rotation in such a way that the blue screen always appears behind the actor from the camera point of view. Dimensionality problematics and budget limitations led us to choose the latter solution since, at that time, no other room was available for our experiments and the one we had for building the system described in Chapter 5 was too small. Therefore, we designed our system with the intention to make it compact and

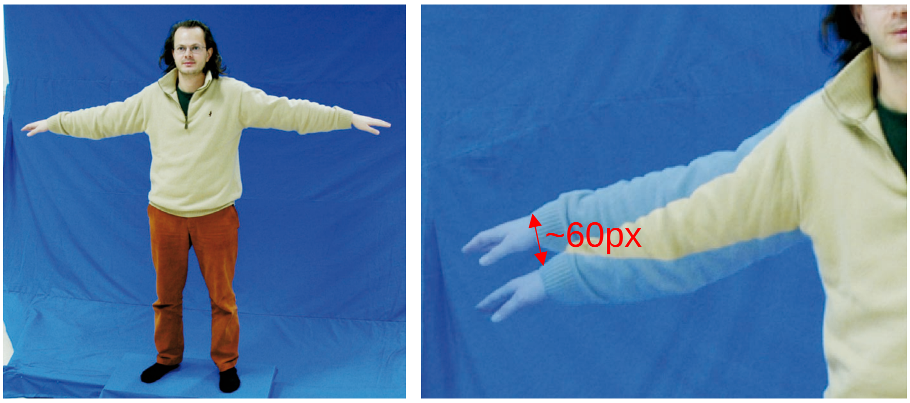


Figure 3.2: *(Left) Vitruvian man's pose, me in one of the first tests for evaluating the movements of the arms. (Right) Visual estimate of the lowering of the actor's arms after* 20 *second, the image of the arm in the original position is superimposed with the image of the same arm after* 20 *seconds.*

portable so that it could be mounted in any room inside our department booked for few days. The second solution was clearly the best one.

However, there still is an open issue to cover. Experiments, indeed, showed that the attention of the actor was continuously distracted by the movement of the system generating a visual feedback which his brain interpreted as a self rotation in a direction opposite to the system one. As before, this sensation, which felt like a loss of orientation, led to an automatism which corrected the actor's pose to compensate for the perceived rotation.

In order to mitigate the influence of these visual feedbacks, the adopted solution was to force the subject to focalize his attention on a reference object placed statically in front of him. The visual feedbacks generated by the reference object contrast the rotational ones generated by the system, leading to a reduction of the perceived sense of rotation.

To guarantee this last statement, the reference object has to be always visible from the actor point of view even when the blue screen is in front of him. This means that the reference object has to be placed between the subject and the blue screen and, in order to guarantee the requirement two, it has to be invisible from the camera point of view. The solution we adopted was to use a blue colored reference object attached to the ceiling by a transparent nylon thread and placed in such a way that it never crosses the visual rays starting from the camera center and ending to a point on the actor's body surface. Therefore, this object is always detected as background in every image and since it always

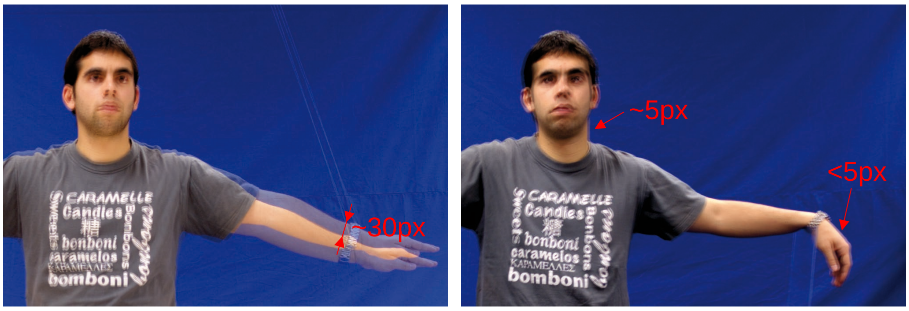


Figure 3.3: *Comparison between the two tested approaches to support the arms during the acquisition namely, the nylon threads approach (left) and the plexiglas plates approach (right). Each picture is obtained superimposing the image of an actor's arm in the original position with the image of the same arm after* 5 *minutes.*

lies on background visual rays, it does not compromise the silhouette extraction accuracy.

In spite of all these precautions, the Vitruvian man's pose is not easy to maintain precisely for a long time, especially the position of the arms. In fact, the Vitruvian man's pose requires straight arms out of the man's frontal silhouette and perpendicular to his main axis. This position is very tiring to maintain because all the arm muscles have to be kept tense. Experiments showed that within 20 seconds a normal person assuming this pose lower his arms of about $15cm$ without realizing it (see Figure 3.2). Clearly, such a big movement invalidates any possible approximation of the stationariness of the actor's pose.

To overcome this problem, we used two plexiglas plates to support the weight of the arms during the acquisition. The thickness of these supports (about $5mm$) is chosen both to guarantee a sufficient transparency in all the directions and to guarantee a rigidity sufficient to balance the weight of the arms. The residual deformation, instead, has to make these supports adaptable to the specific actor characteristics, assuming the most comfortable curvature.

Experiments showed that this solution considerably reduces the movement of the arms during the acquisition. More precisely, in all the three tested cases, the movement measured after 5 minutes was less than one centimeter. We made another experiment using some nylon threads in place of the plexiglas plates to support the arms. Figure 3.3 shows the comparison between these two approaches superimposing the image of an actor's arm in the original position with the image of the same arm after 5 minutes. Using the threads, the measured

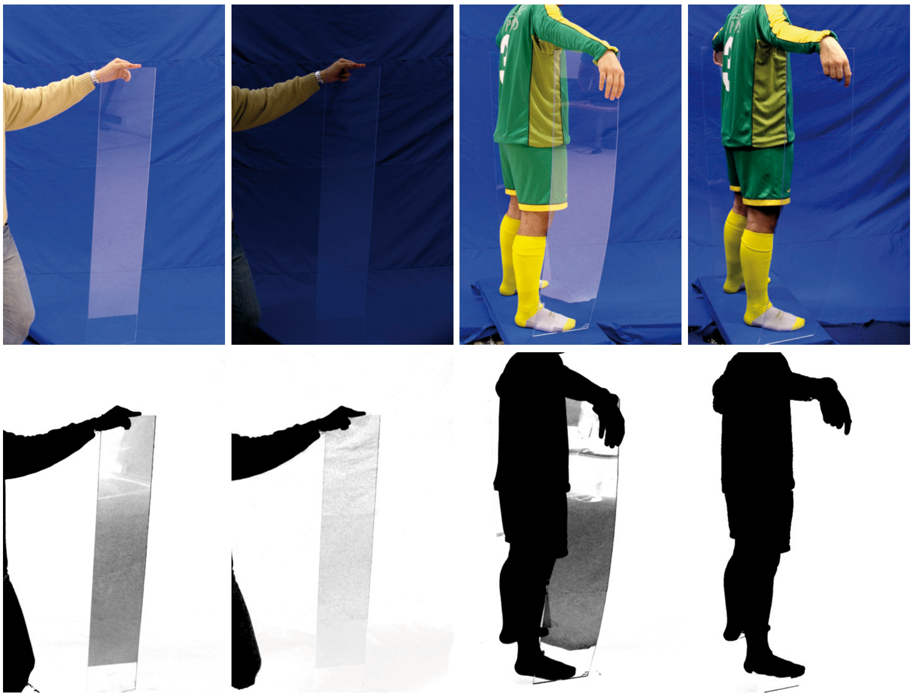


Figure 3.4: *Specular reflections generated by the plexiglas plates and their influence on the colors and the segmentation results.*

visual movement was about 30 pixels, on the contrary, using the plexiglas plates, this movement was reduced to less than 5 pixels. This last value can be easily neglected considering that the used image resolution is 6.1 MPixels.

Unfortunately, the use of the plexiglas undermines the requirements number two and three. In fact, the plexiglas plates are not, in general, fully transparent especially along their borders and, moreover, they present strong specular reflections. Therefore they can alter the colors of the actor body and also compromise the accuracy of the silhouette extraction procedure.

Even if the opacity of the borders can be neglected, since their thickness is very small, the specular reflections remain a problem. In order to reduce these artifacts, the lighting system can be adjusted in such a way that, no direct reflectional paths exist between the camera and the light sources. Instead, the indirect illumination of the supports can be reduced by blocking the light rays coming everything that is not a light source. An alternative solution to both these kind of illumination issues is the use of a polarized lens to filter the light reflected by the plexiglas plates.

Some experiments were made to evaluate these two approaches. Figure 3.4 shows the specular reflections generated by the plexiglas plates and their influence on the colors and the segmentation results. The first and the third columns show these artifacts while the second and the forth ones show the results obtained using respectively the polarized filter and the light path blocking approach. As can be seen from the figure, the specular reflections increase the brightness of the images and often lead to an increase of the ambiguities in the segmentation results. The polarized filter reduces these ambiguities but it less efficient than the light path blocking approach. In this last case, indeed, the obtained images are very close to the original ones and their segmentation is almost perfect.

3.2 The hardware

The bearing structure of our body scanner is depicted on the left of Figure 3.5. It is composed of two parts namely, a central part and a moving part. The former, located at the center of the system, is static while, the latter, composed mainly of aluminium profiles, can rotate around the former. The central part comprises the platform, where the subject to acquire has to be positioned, and the mechanism allowing the rotation of the latter part.

The platform is rectangular and made of wood. It is supported by a central hollow pin made of steel welded on a rectangular plate which, in its turn, is fixed with some screws on a cross-shaped structure made of aluminium profiles. This last structure lays directly on the floor and it is designed big enough to guarantee the stability of the entire system when the actor gets onto the platform.

The moving part includes both the structure supporting the camera and the

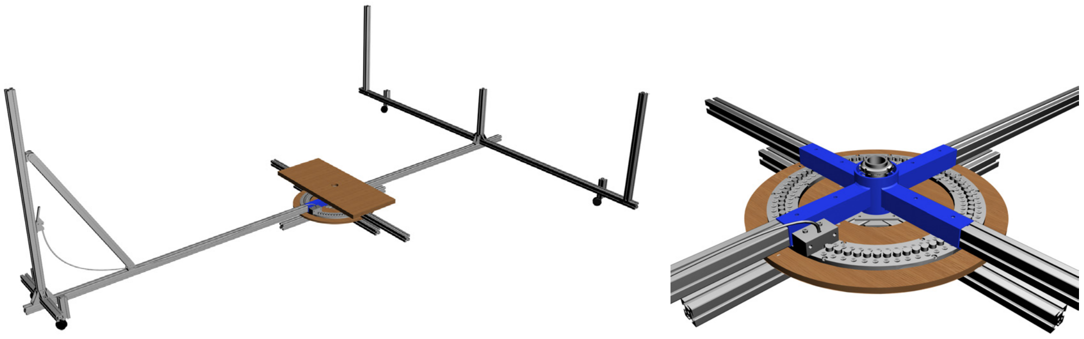


Figure 3.5: *(Left) Bearing structure of our body scanner. (Right) Particular of the rotational mechanism.*

structure supporting the blue screen. Both these structures are connected each other by four arms made of aluminium. These arms are connected, in their turn, with the central part by the rotational mechanism which allows the moving part to rotate around the central pin supporting the platform. The rotational mechanism consists in a hollow cylinder, coaxial with the central pin and connected to it by two angular ball bearings arranged in a circle (see Fig. 3.5(Right)). In order to reduce possible plays, the two bearings are fixed and slightly preloaded by a ring nut screwed on the central pin. Moreover, four housings are welded on the side of the hollow cylinder to lodge the arms of the moving part.

The camera and the blue screen are supported by the longest two arms of the moving part. Since that, the considerable mass of the blue screen and the significant length of these two arms undermine their rigidity, it is necessary to support them with some small wheels, fixed at their extremities.

In order to ensure the requirement number four, i.e., the one concerning the repeatability of the camera positioning, the rotational mechanism is designed in such a way that it can stop only in some specific angular positions. More precisely, a ring-shaped plate coaxial with the central pin is fixed on the top of the cross-shaped structure. 60 bolts are screwed near its external circumference spaced one to another by an angular step of 6 degrees (see Fig. 3.5(Right)). A blocking mechanism is fixed on one of the housings for the arms of the moving part. It is equipped with a V-shaped groove designed to slot exactly the head of one of the bolts. A lever controls the blocking mechanism. The activation of this lever opens the lock allowing the system to rotate. A release, instead, centers the lock in the nearest bolt.

The two profiles, placed perpendicularly and obliquely to the structure supporting the camera, have the only purpose to make this last structure more rigid so that, the plays of the camera, due mainly to bends and twists of the arms of the moving part, are reduced to the minimum.

Some experiments were made to evaluate the error on the repositioning of the camera. More precisely, each of the sixty allowed positions of the system was calibrated five times using a checkerboard and the calibration results compared. The measured average error was smaller than $0.071°$.

On top of its bearing structure, the system is equipped with some wood made sticks forming the infrastructure for the fabric covering the entire system (see Fig. 3.6 and Fig. 3.7). Part of this fabric forms the blue screen while the other namely, the darker one, is used to block the light rays generating the unwanted illumination on the subject and on the plexiglas plates.

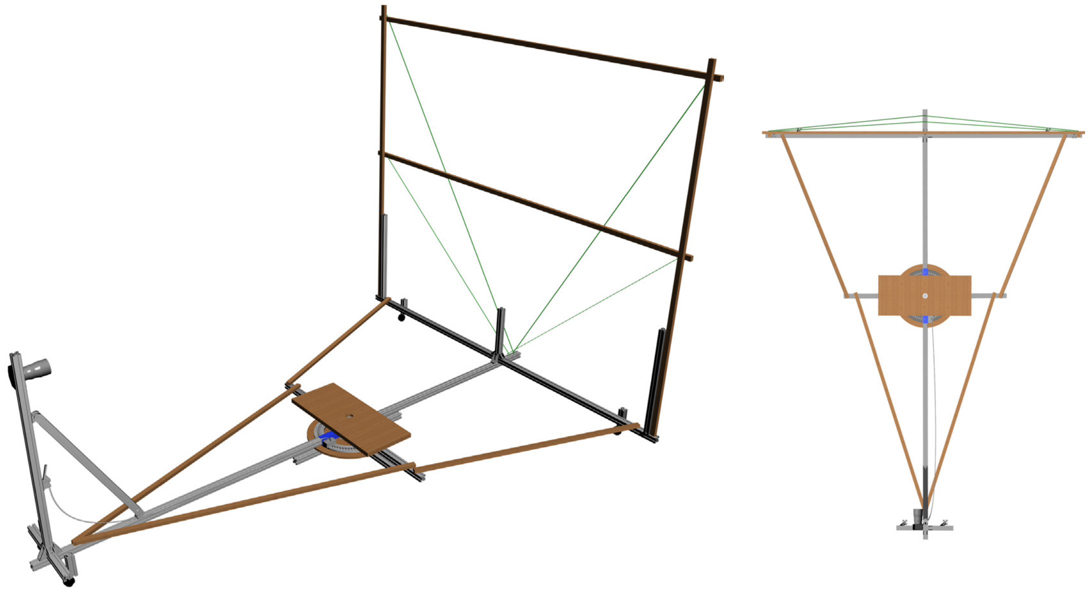


Figure 3.6: *Bearing structure completed with the infrastructure supporting the fabric. (Left) Perspective view. (Right) Top view.*

The size of the system, more specifically, the distance between the camera and the rotation axis and the one between the blue screen and the rotation axis, have to be decided considering the requirement number two. Since the system is rotating around the actor, this last one can be approximated using the sphere that circumscribes it. To ensure a solid blue background in all the acquired images, all the rays starting from the camera center and passing through this sphere have to intersect the blue screen. Clearly, the farther the screen is from the rotation axis the bigger it has to be to satisfy this last requirement. This would lead to the decision to place the blue screen as close as possible to the rotation axis. Unfortunately, the distance between the sphere and the blue screen has to be sufficiently big to avoid that the actor's hands generate shadows on the blue screen, decreasing the accuracy of the background subtraction procedure. Moreover, the camera have to be placed far enough from rotation axis to ensure that its field of view contains entirely the sphere.

Our body scanner mount a Nikon D70s[1] camera equipped with a 18mm lens. The image resolution is 6.1 MPixels and the CCD size are 23.7mm by 15.6mm. The camera is placed 2.3 meters away from the rotation axis at an height of 1.4 meters with respect to the platform. The blue screen is, instead, placed at 1.5

[1] `http://www.nikon.com/`

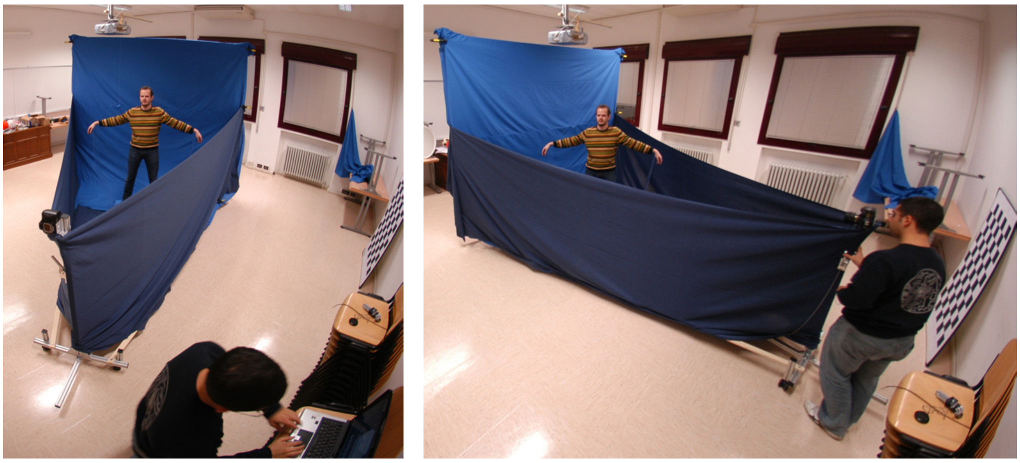


Figure 3.7: *The complete body scanner during an acquisition.*

meters away from the rotation axis and its size is 3 by 2.5 meters.

Concerning the lighting system, it consists in a set of neon lights static with respect to the floor and thus the actor. In this way each point of the actor's body is illuminated from the same direction in every picture and therefore, neglecting the non-Lambertian reflections, each point appears with a consistent color in every image. The non-Lambertian reflections are reduced displacing the lights in such a way to create a diffuse illumination. This moreover, reduces the shadows that could be generated by the actor onto the blue screen, increasing the accuracy of the background subtraction procedure. As for the plexiglas plates, the indirect illumination is blocked by the dark fabric surrounding the system.

Figure 3.7 shows two images of the final body scanner during an acquisition.

3.3 Acquisition and pre-processing

During an acquisition session, sixty photos of the actor are taken in each of the allowed configurations of our body scanner. An operator controls both the lever of the blocking mechanism and the shooting button of the camera. Once a photo is taken in one configuration, the system is rotated until the next configuration is reached. This procedure is repeated sixty times. Figure 3.8 shows some typical images acquired during an acquisition session.

The images are then processed to recover the silhouettes and the color infor-

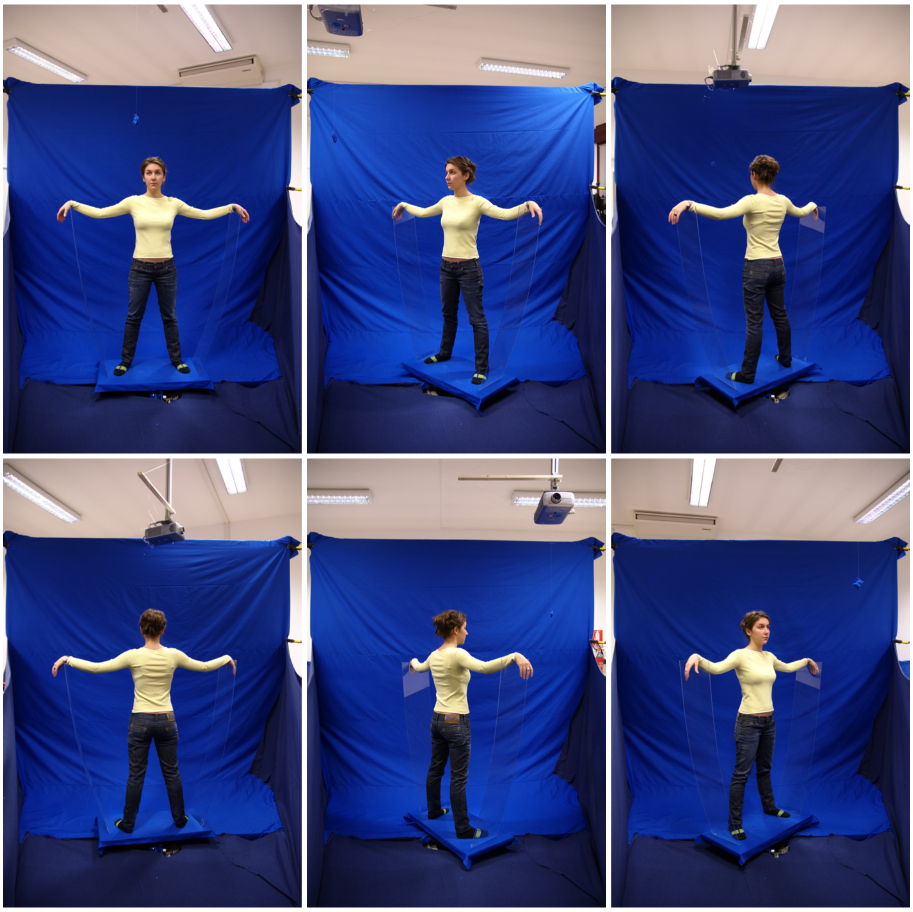


Figure 3.8: *Some typical images acquired during an acquisition session of our body scanner.*

mation. Since a blue screen is adopted, an HLS keyer is chosen to perform this critical task.

More precisely, the complete pipeline is shown in Fig. 3.9. Each input image is connected to a specific mask defining which pixels can be considered during the background subtraction procedure, i.e., the ones representing either a point on the blue screen or a point on the actor's body. All the other pixels represent points outside our system and therefore, are not considered.

In order to recover this mask for every input image, an off-line acquisition with no subject is performed and the HLS keyer is used to detect all the blue pixels in the images in the more conservative way as possible, i.e., a point is detected as blue only if it surely belongs to the blue screen. In particular, a green pattern was also used to cover the platform and the HLS keyer tuned to detect also the green pixels because, in this way, the detection of the point on the platform increases

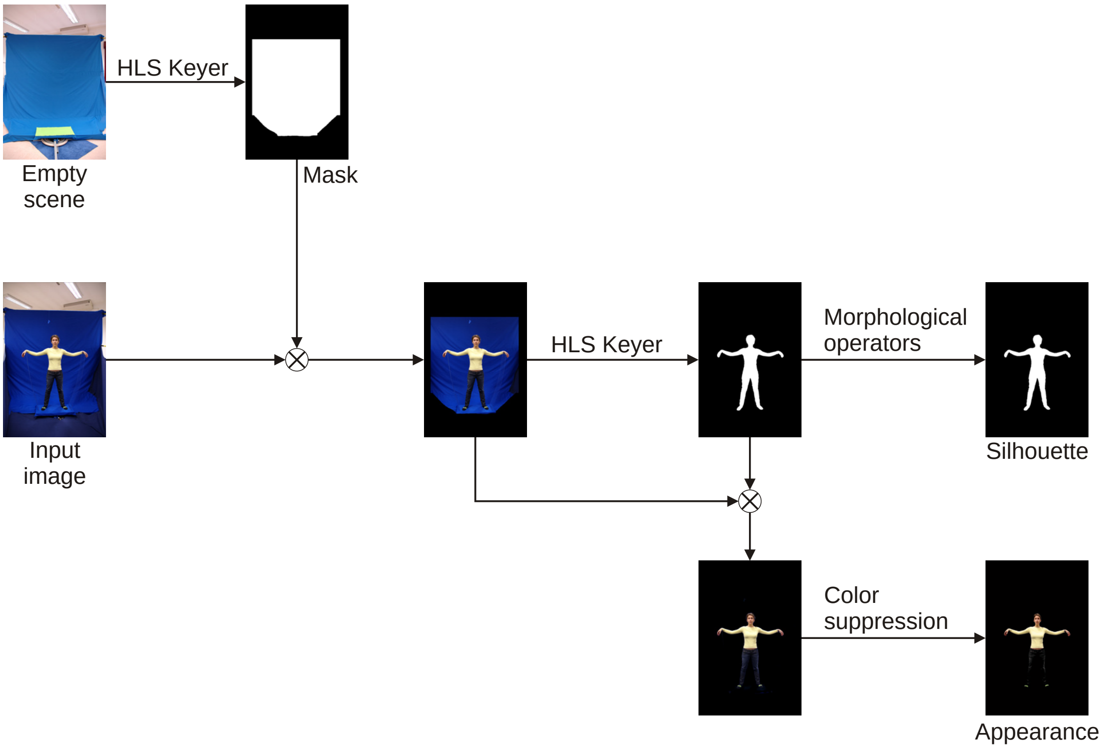


Figure 3.9: *Pre-processing pipeline aimed to recover the actor silhouettes and the color information.*

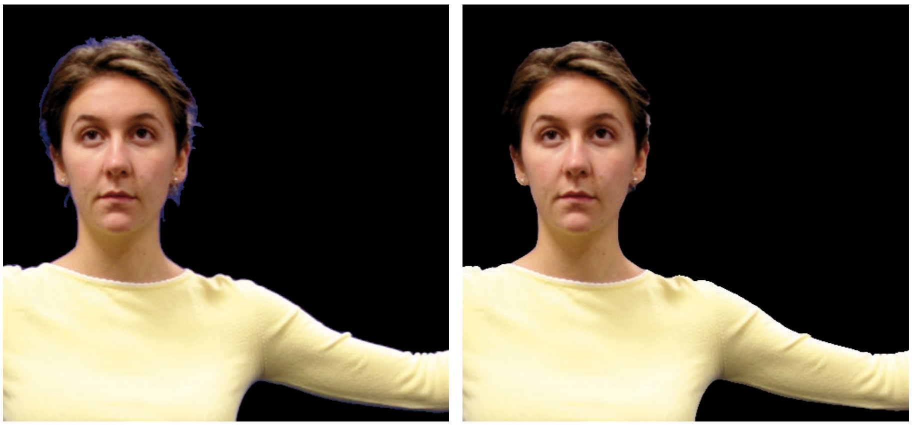


Figure 3.10: *Example result of the color suppression operator used to reduce the blue halos artifacts occurring near the silhouette border of an actor's image. (Left) Original image. (Right) Processed image.*

its accuracy.

Each input image is then multiplied by its related mask and the result processed by the keyer. Some morphological operators are then applied to improve the detection accuracy.

The HLS keyer was chosen because it works pretty well with this type of images and it is easy to tune. However other keying techniques can be used as well. The reader should refer to Appendix B for a review.

Once a silhouette is detected, it is multiplied by the original image to obtain the color information. A color suppression operator is then used to remove the blue halos typical of the use of a blue screen. These artifacts occur near the silhouette border where the screen color can spread inside the actor's image. As shown in Figure 3.10, the color suppression operator removes all the blue and violet shades generated by the blue screen.

3.4 Shape reconstruction

The shape reconstruction phase aims to recover, from the input images, a triangular mesh representing the actor's external surface. An overview about the state of the art of all the main techniques aimed to accomplish this task is made in Appendix A.

Here, instead, we briefly describe the solution adopted for our specific type of data. The reader should refer to the previously mentioned appendix for a deeper analysis or for an alternative solution.

Dense stereo information is first extracted from each pair of consecutive images and subsequently used together with the silhouettes, extracted during the previous phase, in order to recover the geometry of the actor.

More precisely, a first coarse estimation of this surface is obtained using the only silhouette information. This is accomplished by a volume based shape from silhouette algorithm. Afterwards, parametric deformable models are used to fuse together all these information as we did in our past work [10], which is a variant of [46]. Their deformations are iterated until a functional, considering all these of information and the smoothness of the entire surface, reaches its minimum. More details are provided in the last section of Appendix A regarding the multimodal methods, or in our paper [10].

The main difficulties during this task are due to the particular shapes we are

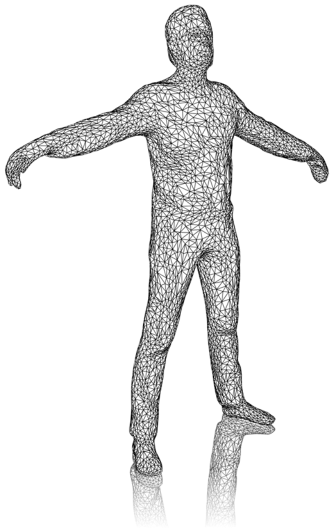


Figure 3.11: *A typical wireframe model reconstructed by our body scanner.*

reconstructing. The external surface of the human body presents, in fact, a lot of small high detailed regions separated by big and almost uniform ones. This leads to a concept of smoothness relative to each specific region of these kind of surfaces. The stereo algorithm has to be designed to take into account for this type of smoothness and to estimate depth maps which discontinuities occurs only along the border of the actor's silhouettes.

The accuracy of our approach decreases considerably in body parts like the face and the hands. These parts are too small with respect to the entire body and moreover, they present strongly non-Lambertian reflections. For these reasons they have to be reconstructed using a specific technique different from the one used for the rest of the body.

The obtained models count more than 500 thousands faces but we downsample them, in an non-uniform way, obtaining models with 13 thousand faces. This downsampling is necessary to limit the used computer resources maintaining a perfect description of the actor for our purposes. Figure 3.11 shows a typical wireframe, i.e., a triangular mesh, obtained at the end of this phase.

3.5 Skeleton estimation

This phase aims to recover the skeletal structure underlying the acquired actor.

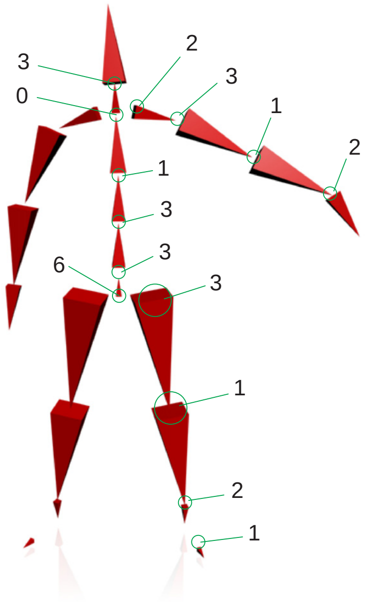


Figure 3.12: *Bones arrangement in the used skeletal structure with their related degrees of freedom.*

	X-Axis		Y-Axis		Z-Axis	
	min	max	min	max	min	max
Pelvis	-180	180	-180	180	-180	180
Spine lv. 0	-15	15	-10	10	-10	60
Spine lv. 1	-20	20	-18	18	-10	60
Spine lv. 2	-180	180	-180	180	-180	180
Neck	-180	180	-180	180	-180	180
Head	-70	70	-50	50	-60	30
Clavicles	-30	30	-10	10	0	0
L Upper arm	-90	40	-40	85	-90	30
R Upper arm	-40	90	-85	40	-90	30
Forearms	0	0	0	0	-140	0
L Hand	-90	30	-80	80	0	0
R Hand	-30	90	-80	80	0	0
L Thigh	-90	90	-40	90	-160	90
R Thigh	-90	90	-90	40	-160	90
Lower legs	0	0	0	0	-140	0
Feet	-90	90	0	0	-90	50
Toes	0	0	0	0	-30	30

Table 3.1: *Constraints of the skeleton.*

Our system models the human's skeleton using 22 bones arranged as depicted in Figure 3.12. The pose of this skeleton counts 46 degrees of freedom arranged as follows: head (3), neck (0), clavicles (2+2), upper arms (3+3), forearms (1+1), hands (2+2), spinal bones (3+3+1), pelvis (6), thighs (3+3), lower legs (1+1), foots (2+2) and toes (1+1). Not all the possible configurations of these degrees of freedom are plausible poses for the human. Table 3.1 reports the constraints used by our system, to determine the the allowed configurations of the skeleton.

The purpose of this phase is to estimate the length and the pose of each of these bones from the previously reconstructed triangular mesh. In our realization, this task is accomplished manually. However, a fully automatic procedure can be found in Baran and Popovic [12].

3.6 Texture reconstruction

Texture mapping is a method for adding detail, surface texture or color to a 3D model. Its application to the computer graphics was pioneered by Catmull[2] in his Ph.D. thesis of 1974 [25].

Ideally, given a 2-manifold Φ, the aim of the texture mapping is to assign to each point of Φ, a particular reflectance property. Formally, this assignment is a map between Φ and the space of the reflectance properties. Let's denote this latter space with the symbol Υ and denote this map with ξ. $\xi : \Phi \to \Upsilon$ is called *texture map*.

Commonly, instead of directly defining ξ, two intermediate functions are defined in the following way,

$$\begin{tikzcd}
\Phi \arrow[rr, "\xi"] \arrow[dr, "M"'] & & \Upsilon \\
& {[0,1]^2} \arrow[ur, "T"'] &
\end{tikzcd} \tag{3.1}$$

where $T : [0,1]^2 \to \Upsilon$ is a generic image with codomain Υ and $M : \Phi \to [0,1]^2$ is a injective and piecewise homeomorphic function. T is called *texture* and M is called *uv-map*[3]. Since M represents a map from a 3D manifold to a 2D plane, it

[2] Catmull is currently the president of the Walt Disney Animation Studios and the Pixar Animation Studios.

[3] The term uv-map comes from the fact that the coordinates in the texture domain are usually expressed with the symbols u and v.

is often called *unwrap map*.

The function M locally, is a chart of Φ and so its local inverse is a local parameterization of Φ. This means that Φ can be partitioned, neglecting some points, into open subsets $\{\phi_i\}_i$ such that, for each ϕ_i, the function $M_{|\phi_i}$ is a global chart of ϕ_i.

Given a manifold Φ and a set of images representing Φ, the texture reconstruction task consists in finding a texture map ξ of Φ such that the resultant model is photo-consistent with the given images. If an uv-map M of Φ can be recovered, the problem is reduced to the only synthesis of the texture T. We first address the problem of recovering an uv-map M of Φ and then we address the problem of synthesizing T.

In spite of the clear fact that, a lot of uv-maps can be defined for each single manifold, the problem of defining one is computationally hard. Common approaches cut the surface into pieces and then unfold each piece respecting both the continuity and the injectivity constraints. This unfolding procedure is usually performed by the so called *pelt map* approach, consisting in the stretching of the surface inside the 2D domain until all the overlapping regions disappear. Other obsolete techniques are the spherical mapping, the cylindrical mapping, the box mapping and the plane mapping.

All these techniques do not, in general, satisfy a very desirable property of the uv-maps, namely the isometric one. Maps satisfying this property preserve both angles and distances and thus, also areas. Regular sampling grids with uniform spacing in the parameter domain are undistorted by these maps onto the surface. Moreover, proportions are preserved, i.e., big regions of the manifold map into big regions of the texture, and viceversa. However, sometimes some of these regions are more important than others even if they are smaller. For instance, the face of a human is more important than his pants even if it is smaller. In these cases, it is preferred to weight the isometric property to correctly match these characteristics. The resultant maps are no longer isometries, but only conformal maps, i.e., maps preserving angles but not distances.

In all the other cases, isometries are preferred to conformal maps. Unfortunately, with the exception of the developable surfaces, such as the cylinders, general manifolds cannot be flattened by isometries, neither by a piecewise isometries. Therefore, a minimum distortion criteria, considering the distance between a generic map and the nearest isometry or conformal map, must be used to flatten general manifolds.

We started implementing the approach suggested in [146]. This method, first

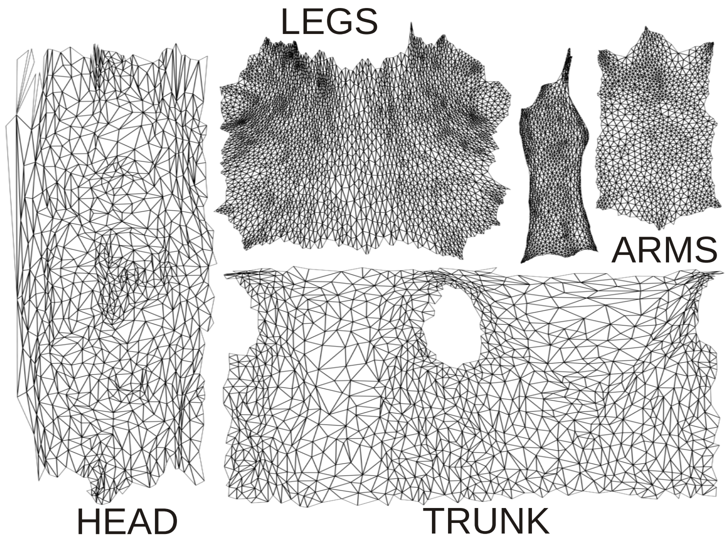


Figure 3.13: *Codomain of an uv-map generated by our system.*

defines a distortion metric penalizing maps not satisfying the isometric condition, then it produces an uv-map which distortion is below a certain threshold. The algorithm incrementally flatten the mesh surface by growing patches, maintaining the distortion metric below the preselected threshold. When this is no longer possible, it stops the flattening procedure and starts a new patch.

Unfortunately, the number of generated patches is inversely proportional to the preselected distortion threshold. As we will see next in this section, this becomes a very serious problem during the texture synthesization because it undermines the assumption that the effective texture domain, i.e., the uv-map codomain, could be considered similar to $\mathbb{R}^2$. This last property is absolutely necessary for treating the texture as a bidimensional signal.

In order to satisfy this property, our system adopts a strategy designed specifically for the type of objects to acquire, i.e., for humans. Given the 3D model of an actor, our system first defines specific cuts along the mesh surface, in such a way that it results subdivided in five main pieces namely, the head, the left arm, the right arm, the trunk and the legs. Then a pelt mapping algorithm is applied to stretch each patch in order to remove the overlapping regions.

A result can be seen in Figure 3.13.

Concerning the texture synthesization task, our system adopts the same blending approach we had proposed in [21]. The space Υ is restricted to the RGB color space so that, the resultant texture T is a common RGB image. Let's denote with I_i a generic input image and with Π_i its related projection map. Since the manifold Φ is known, Π_i can be inverted in a specific region R of its codomain, more precisely in $R = \Pi_i(\Phi)$. The combination of this latter function with M results in a map from R, subset of the domain of the input image I_i, to the domain of the texture image. Let's call this latter function, π_i. Summarizing,

$$\begin{tikzcd}
[0,1]^2 \arrow[r, "I_i"] & RGB \\
\Phi \arrow[u, "\Pi_i"] \arrow[r, "M"] & {[0,1]^2} \arrow[u, "T"']
\end{tikzcd} \tag{3.2}$$

$$\pi_i : \left(R \subseteq [0,1]^2\right) \to [0,1]^2 \tag{3.3}$$

In general, each image I_i gives only a partial reconstruction of the actual texture T. Let's denote with T_i the partial reconstruction obtained using only the image I_i. In order to recover T, one has to fuse together all the obtained T_i.

Let's note that the quality and the quantity of "useful information" about T are not, in general, uniformly distributed over all the partial reconstruction T_i. In fact, some of them describe well specific regions of T neglecting the others, and viceversa.

Therefore, it is obvious that, the synthesization task has to take into account of the information distribution over all these partial reconstructions of T. More precisely, it has to weight more data coming from a T_i with higher quality and quantity of information than the others.

In order to formalize the previous statements, we introduce the concept of quantity of information stored in a specific region of a T_i and the concept of distortion that this information is subjected in that specific region, that is, the quality of this information.

The distortion of a function, a signal or an image, is a very generic concept describing how much the original function/signal/image is being altered by some external factors. In fact, the distortion measures the similarity between the original signal and the distorted one.

In our case, the distortion of a source channel, i.e., a T_i, is mainly due to the distortion of the map π_i and to the distortion of the image I_i. In its turn, the distortions of π_i is due to calibration errors and to surface reconstruction errors. Instead, the distortion of I_i is due to all the possible distortions that may arise during the image formation process, such as, for instance, shading, shadows and highlights generated by non-Lambertian surfaces or illumination changes, or different image white balance settings, overexposure, vignetting and so on.

Finally, once the distortion of both I_i and π_i is known, the distortion of each pixel of T_i can be computed multiplying together the previous ones.

Concerning the quantity of information stored in a specific region ζ of a partial reconstruction T_i, it can be seen, intuitively, as the area of the region of the original image I_i representing ζ. This means that, the bigger this area is, the more information about ζ is provided by T_i. The previous definition make sense only considering both I_i and T_i as discrete domain images, i.e., made by pixels. In this case, indeed, the finite sampling of the image causes a loss of information during a shrink of the region. This does not happen in the continuous case except when this region collapse into a single point or a line.

In order to mathematically describe the previous concept, we define, where it is possible, the function $g_i(p)$ as the absolute value of the determinant of the jacobian of an inverse of π_i, i.e.,

$$g_i(p) = \left|\det\left(J\left(\pi_i^{-1}\right)(p)\right)\right| \tag{3.4}$$

$g_i(p)$ represents the transformation ratio between the area of the infinitesimal texture region around p and the area of the corresponding region in I_i with respect to the map π_i. Clearly, for a given point p of the texture, the fact that $g_i(p)$ is less than 1 means that the image I_i doesn't have enough texture information concerning p. Viceversa, $g_i(p)$ greater than 1 means that the texture cannot store all the details offered by the input image I_i, since they all collapse inside a same pixel. Therefore, for each point p, information coming from the partial reconstruction having the highest value of $g_i(p)$ is preferred, since this source gives more information about p than others.

At this point, it is important to define another useful map which characteristics are similar to ones of $g_i(p)$. Given a point p of a partial reconstruction T_i, the surface normal at the 3D point $M^{-1}(p)$ can be determined. Let's call it $n(p)$ and call $\widetilde{n}_i$ the normal of the image plane of I_i. The function

$$h_i(p) = -n(p) \cdot \widetilde{n}_i \tag{3.5}$$

describes the level of parallelism between the local surface and the image plane of I_i. Low values of $h_i(p)$ means that p is not well represented in I_i since its surrounding area is too small and it could suffer of shading effects. This means that the quantity of information is also low while the distortion could be high. Therefore, information coming from the partial reconstruction having the highest value of $h_i(p)$, is preferred.

The quantities $g_i(p)$ and $h_i(p)$ are, in general, strongly correlated, and while $g_i(p)$ is related to only the quantity of information coming from a specific point

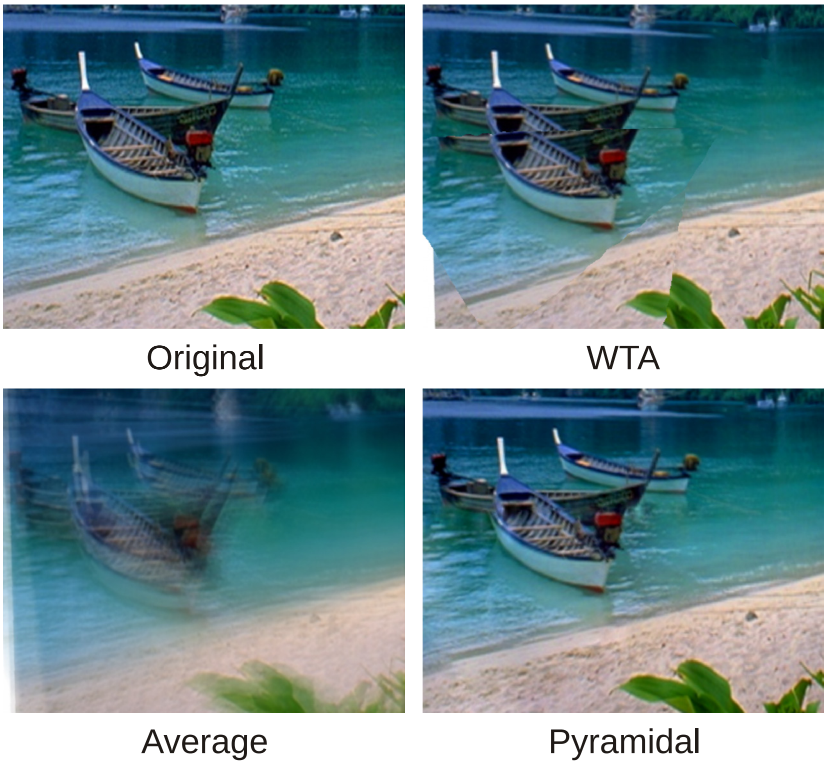


Figure 3.14: *Textures reconstructed using three different fusion techniques namely, WTA, average blending and pyramidal blending based on wavelet [21].*

of a T_i, $h_i\left(p\right)$ is related also to its distortion. However, differently from $g_i\left(p\right)$, $h_i\left(p\right)$ does not consider scaling. More precisely, if p is seen by two images, one a little bit far away from the model than the other, the value of $h_i\left(p\right)$ is the same in both images, while the value of $g_i\left(p\right)$ decreases in the second image because the area of the surround region around p is smaller than in the first image. Moreover, let's note that, $h_i\left(p\right)$ does not consider any other distortions other than shading.

Even so, the combination of $h_i\left(p\right)$ and $g_i\left(p\right)$ forms a meaningful functional considering both the quantity and the quality of incoming information from a given source. Therefore, it should be used as weight function during the process that fuses together all the T_i to recover the texture T.

Traditional fusion approaches are the *Winner Take All* (WTA) approach and the *Average blending* approach. For each point of T, the former selects information coming from the best source while the latter makes a weighted average over all the sources of information.

Unfortunately, none of these two approaches is artefact free. WTA generates non-continuous textures like the one on the left of Figure 3.14. The discontinuities

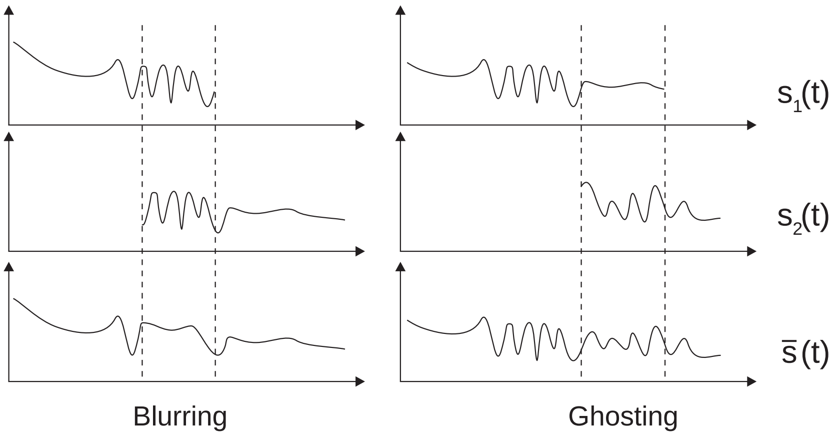


Figure 3.15: *One dimensional representation of blur and ghosting artefact generated by averaging two signals $s_1(t)$ and $s_2(t)$ in their common region.*

are located along the lines where the preferred source of information changes and they become visible for high distorted sources of information.

On the other hand, average blending generates blur and ghosting artifacts (see Fig. 3.14) when the input signals are respectively correlated and uncorrelated. A graphical explanation for the generation of these two artifacts is given by Figure 3.15. In the first case, on the left, $s_1(t)$ and $s_2(t)$ are correlated, more precisely, $s_2(t)$ is a shifted version of $s_1(t)$. The average of these two signals in their common region is equivalent to a spatial low pass filter which, in practice, causes the blur. In the second case, instead, the two signals are completely uncorrelated. The average is a signal having, in the common region, the ghost of $s_2(t)$.

A way to avoid the above described artifacts is to use a multi-resolution fusion approach. This type of technique, also known as pyramidal blending approach, is often described in the literature to fuse multiple images and it is usually based on Laplacian kernels. However, pyramidal blending had not been applied to the texture reconstruction problem before our work in 2005. In this work, all the sources of information are evaluated at each resolution and the multi-resolution analysis is performed using the discrete wavelet decomposition (DWT) [91].

It is worth to recall in what consists the wavelet decomposition of a bidimensional signal, i.e., an image. This decomposition splits the original image into a set of bidimensional signals organized in a tree structure. Each node of this tree has four children representing respectively the low band (LL), the high vertical band (LH), the high horizontal band (HL) and the high corners band (HH)

of the signal represented by the parent node. Figure 3.16 depicts the DWT decomposition of a bidimensional signal, showing on the left, its frequency domain representation and, on the right, its tree structure.

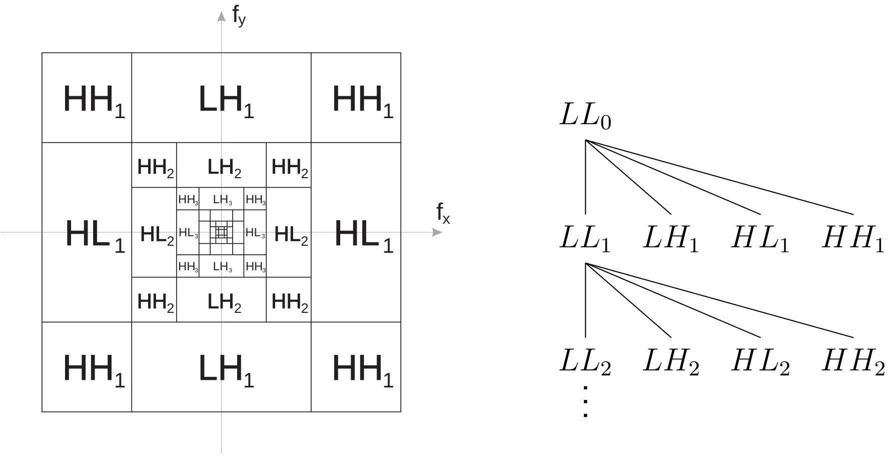


Figure 3.16: *Discrete wavelet decomposition of a bidimensional signal: (Left) frequency domain representation, (Right) tree structure of the decomposed bands.*

The basic idea behind our approach is to evaluate the quality and the quantity of information at each node of the wavelet tree and then fuse each node separatively using the WTA approach.

What can be easily observed is that the quality of information (i.e., its distortion) remains unchanged in every node of the tree, while, its quantity varies depending on both the type and the level of the node.

Let's denote with the symbol i^l_x the quantity of information stored in the unique node of type $x \in \{LL, LH, HL, HH\}$ presents at level l of the wavelet tree, and denote with i^l the quantity of information stored inside the entire level l. A plausible assumption is that the information is preserved inside the pyramidal structure. Therefore, the following statement holds

$$i^l = i^l_{LL} + i^l_{LH} + i^l_{HL} + i^l_{HH} \tag{3.6}$$

i.e., the quantity of information of an entire level is equal to the sum of all the quantities of information stored in each singular node belonging to that level. Moreover, the quantity of information of a level is equal to the one stored in its parent node, i.e.,

$$i^{l-1}_{LL} = i^l \tag{3.7}$$

For each point p of the texture T, let's define, the quantity of information at level 0 to be equal to the absolute value of the determinant of the jacobian $J\left(\pi_i^{-1}\right)$ evaluated in p, i.e.,

$$i^0 = i_{LL}^0 = \left|\det\left(J\left(\pi_i^{-1}\right)(p)\right)\right| \tag{3.8}$$

Inside each level l, the information i^l is distributed in all its nodes according to $J\left(\pi_i^{-1}\right)(p)$. In particular, if $\left|\det\left(J\left(\pi_i^{-1}\right)(p)\right)\right|$ is less or equal than $(1/4)^l$, all the information is stored in the LL node.

On the contrary, if $\left|\det\left(J\left(\pi_i^{-1}\right)(p)\right)\right|$ is greater than $(1/4)^l$, the norm of the two column vectors of $J_i^l(p)$ and their dot product tell how the information is partitioned respectively in the high horizontal band, the high vertical band and the high corners band. The remain information,

$$i^l - \left(i_{LH}^l + i_{HL}^l + i_{HH}^l\right) \tag{3.9}$$

is stored in the LL node.

From the above considerations, one can easily estimate the quantity of information stored in each node of the tree for any given point p. Then, an indicative value about the quality and the quantity of information stored in a node at point p can be retrieved multiplying the previous value with the distortion coefficient.

Our pyramidal blending approach blends separatively each node using the WTA approach based on the above computed indicative values.

The resultant reconstruction is free from artifacts like ghostings, blurs and texture discontinuities as the reader can see from Fig. 3.14. More precisely, blurs are avoided since no average is applied at all. Texture discontinuities and ghostings, instead, are avoided by the proprieties of the inverse wavelet transform. These can be explained observing that the low frequency components have a support bigger than the high frequency ones, i.e., a variation of a low frequency influences a bigger region of the final image. Therefore, a smoothed transition between distorted information is automatically generated.

Figure 3.17 shows a human model which texture was reconstructed using our pyramidal blending approach.

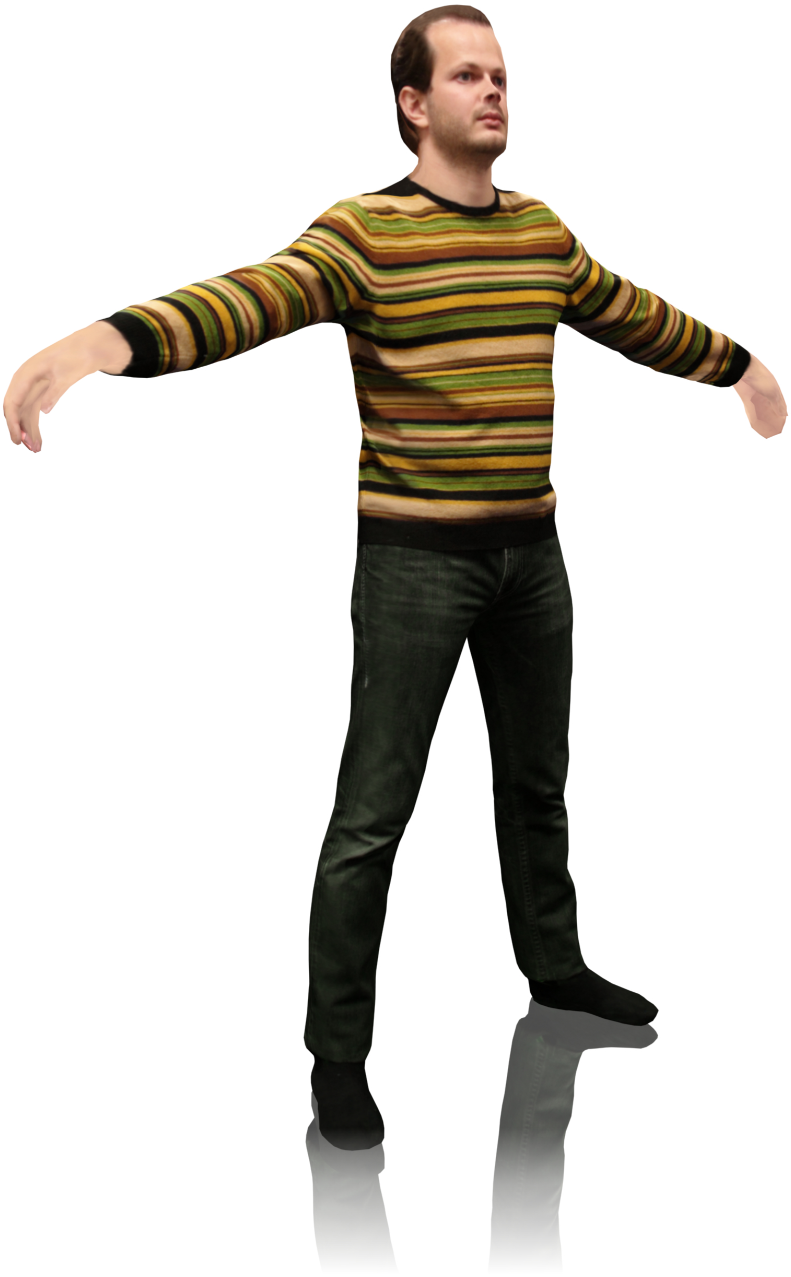


Figure 3.17: *Textured model recovered using the described body scanner.*

Chapter 4

Motion Acquisition

This chapter describes in detail our approach to the pose estimation problem. The first three sections define the notation and the concepts used in this chapter. In particular, Section 4.3 attempts to provide a formal definition of deformable model concept, in such a way that, the reader has a clear viewpoint on the main object the be estimated. Section 4.3.3 recalls some computer graphics concepts used in a standard animation production pipeline underlining the critical role of the deformable models. In the end, Section 4.4 describes our algorithm and makes theoretical comparisons with the current state of the art.

4.1 Notations and formalism

Given a field $\mathbb{k}$, the *general linear group* of order n, is defined to be the group of all the invertible matrices of $\mathbb{k}^{nxn}$ with the inherited matrix multiplication × as internal operation

$$GL(n, \mathbb{k}) = (\{A \in \mathbb{k}^{nxn} \mid A \text{ invertible}\}, \times) \tag{4.1}$$

Given the topology T inherited from $\mathbb{k}^{nxn}$, $(GL(n, \mathbb{k}), T)$ is also a *topological space* and thus a *topological group.* According to the topology T, $GL(n, \mathbb{k})$ is an open subset of $\mathbb{k}^{nxn}$, and, with the appropriate atlas, it is also a manifold in $\mathbb{k}^{nxn}$ of type C^∞, i.e., a *smooth manifold.* Let's, for simplicity, denote this manifold, again $GL(n, \mathbb{k})$, with an abuse of notation. From the above considerations and since both the operations $\times$ and $^{-1}$ in $GL(n, \mathbb{k})$ are smooth, $GL(n, \mathbb{k})$ is also a

Lie group.

The tangent space at the identity $T_I GL(n, \Bbbk)$ and the operation $[A, B] = AB - BA$ form a *Lie algebra* denoted as $gl(n, \Bbbk)$, where the operation $[\cdot, \cdot]$ represents the related *Lie bracket.* From the manifold theory, both the *exponential map* and the derivation concepts are inherited to both $GL(n, \Bbbk)$ and $gl(n, \Bbbk)$.

For the specific real case $\Bbbk = \mathbb{R}$, $gl(n, \mathbb{R})$ contains all the n-by-n real matrices and the exponential map is defined as

$$\begin{array}{rccc} \exp : & gl(n, \mathbb{R}) & \longrightarrow & GL(n, \mathbb{R}) \\ & A & \longrightarrow & \sum\limits_{k \geqslant 0} \frac{1}{k!} A^k \end{array} \tag{4.2}$$

or, in its extensive form

$$\exp(A) = I + A + \frac{1}{2} A^2 + \frac{1}{6} A^3 + \dots \tag{4.3}$$

It is worth noting that this series is not always convergent thus, for some matrices the function exp is not defined.

Same considerations can be made for some particular subgroups of $GL(n, \mathbb{R})$ namely, the *special linear group* of order n in $\mathbb{R}$ denoted as $SL(n, \mathbb{R})$, the *orthogonal group* $O(n, \mathbb{R})$ and the *special orthogonal group* $SO(n, \mathbb{R})$ also known as the *group of rotations* in $\mathbb{R}^n$. They are respectively defined as

$$SL(n, \mathbb{R}) = (\{A \in GL(n, \mathbb{R}) \mid \det(A) = +1\}, \times) \tag{4.4}$$

$$O(n, \mathbb{R}) = \left(\left\{A \in GL(n, \mathbb{R}) \mid A^{-1} = A^T\right\}, \times\right) \tag{4.5}$$

$$SO(n, \mathbb{R}) = (\{A \in SL(n, \mathbb{R}) \cap O(n, \mathbb{R})\}, \times) \tag{4.6}$$

where $A^{-1} = A^T$ is known as the orthogonality property (equivalent to $A^T A = AA^T = I$).

Let's, for simplicity, denote the previously defined group omitting the symbol $\mathbb{R}$, i.e., as $GL(n)$, $SL(n)$, $O(n)$ and $SO(n)$. All of them are topological groups, smooth manifolds and Lie groups, and their related Lie algebras are respectively denoted as $gl(n)$, $sl(n)$, $o(n)$ and $so(n)$.

In particular, $sl(n)$ is the space of all the n-by-n real matrices with null trace while both $o(n)$ and $so(n)$ refer to the space of all the n-by-n skew symmetric real matrices, i.e., the real matrices satisfying the property $A^T = -A$.

The exponential maps related to $GL(n)$, $SL(n)$ and $O(n)$, namely

$$\exp : gl(n) \longrightarrow GL(n) \tag{4.7}$$

$$\exp : sl(n) \longrightarrow SL(n) \tag{4.8}$$

$$\exp : o(n) \longrightarrow O(n), \tag{4.9}$$

are not surjective. The only one surjective exponential map is the one associated to $SO\,(n)$, i.e.,

$$\exp : so\,(n) \longrightarrow SO\,(n) \tag{4.10}$$

for which, there exists at least one inverse called

$$\log : SO\,(n) \longrightarrow so\,(n) \tag{4.11}$$

which, however, is not unique since exp is not injective.

The vector space $so\,(n)$ is isomorphic (with respect to $+$, $\cdot_e$) to the vector space $\mathbb{R}^{n(n-1)/2}$ but no simple considerations can be made for the manifold $SO\,(n)$. However, restricting to the $\mathbb{R}^3$ case, $SO\,(3)$ is known to be diffeomorphic to the *projective space* $\mathbb{RP}^3$, i.e., the Lie group of all lines in $\mathbb{R}^4$ passing through the origin. $\mathbb{RP}^3$, in its turn, is diffeomorphic to the quotient of the *unit sphere* by identification of the antipodal points, $\mathbb{S}^3/\{I, -I\}$. This connection between rotations in $\mathbb{R}^3$ and points on a hyper-sphere in $\mathbb{R}^4$ gives us the possibility to define a metric in $SO\,(3)$ based on the geodesic of $\mathbb{S}^3$. This metric is often used for interpolation purposes between two or more rotations since it offers a constant speed motion along a unit circle. This interpolation technique is known as *spherical linear interpolation* (SLERP) and it is usually performed inside the space of the unit quaternions $\mathbb{H}$ which is isomorphic to $\mathbb{S}^3$ (see [37] for details).

Here, we define the diffeomorphism $\widehat{\exp}$ between $\mathbb{RP}^3$ and $SO\,(3)$ that it is going to be used later in this chapter. Let $\widehat{\cdot}$ be the operator

$$\begin{array}{rccc} \widehat{\cdot}: & \mathbb{RP}^3 & \longrightarrow & so\,(3) \\ & x & \longrightarrow & \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \end{array} \tag{4.12}$$

$\widehat{\exp}$ is defined as

$$\widehat{\exp} = \exp \circ \widehat{\cdot} \; : \mathbb{RP}^3 \leftrightarrows SO\,(3) \tag{4.13}$$

$\widehat{\exp}$ covers injectively the entire manifold $SO\,(3)$ but it is not a local chart of $SO\,(3)$, indeed it doesn't refer to any real space of type $\mathbb{R}^m$, but only to $\mathbb{RP}^3$.

Important local charts of $SO\,(3)$ are the Euler angles, the Tait-Bryan angles, the angle-axis pairs, the unit quaternions, the Cayley rational parameters and also the exponential map. Each of these charts offers a compact version of $SO\,(3)$ but, they are not suitable for many applications involving rotations in $\mathbb{R}^3$. For this reason, it is often preferred to use $\widehat{\exp}$ as internal representation of $SO\,(3)$ and leave the previous local charts for visualization purposes.

The exponential map related to $SO\left(3\right)$ can be written in a closed and invertible form called the *Rodrigues' formula*:

$$\exp\left(A\right) = I + \frac{\sin\left(\theta\right)}{\theta}A + \frac{\left(1-\cos\left(\theta\right)\right)}{\theta^2}A^2 \tag{4.14}$$

where $\theta = \left\|x\right\|_2$ when $A = \widehat{x}$. Its inverse is defined as

$$\log\left(A\right) = \frac{\theta}{\sin\left(\theta\right)}\left(A - A^T\right) \tag{4.15}$$

where θ is a real value satisfying both $Tr\left(A\right) = 1 - \cos\left(\theta\right)$ and $\left|\theta\right| < \pi$.

The *special Euclidean group*, i.e., the group of the *rigid motions*, denoted as $SE\left(n\right)$, is defined to be the group of all the affine maps of $\mathbb{R}^n$ under the operation of composition. In other words, it contains all the maps $\rho : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$\rho\left(x\right) = Rx + U \tag{4.16}$$

where $R \in SO\left(n\right)$ and $U \in \mathbb{R}^n$. As before $SE\left(n\right)$ is a topological group, a smooth manifold and a Lie group and its related Lie algebra $se\left(n\right)$ is defined to be the set of all the $(n+1)$-by-$(n+1)$ matrices of the form

$$A = \begin{pmatrix} \Omega & U \\ 0 & 0 \end{pmatrix} \tag{4.17}$$

where $\Omega \in SO\left(n\right)$ and $U \in \mathbb{R}^n$.

As for $SO\left(n\right)$, the exponential map of $SE\left(n\right)$,

$$\exp : se\left(n\right) \longrightarrow SE\left(n\right) \tag{4.18}$$

can be expressed like in Eq. (4.2) and it is surjective but not injective. One of its inverse is denoted as $\log : SE\left(n\right) \rightarrow se\left(n\right)$.

Restricting to the $\mathbb{R}^3$ case, it can be proven that the manifold $SE\left(3\right)$ is diffeomorphic to $SO\left(3\right) \times \mathbb{R}^3$ which, in its turn, is diffeomorphic to $\mathbb{RP}^3 \times \mathbb{R}^3$. Since $\mathbb{RP}^3$ has the shape of an hyper-sphere in $\mathbb{R}^4$, $\mathbb{RP}^3 \times \mathbb{R}^3$ is shaped like an hyper-cylinder of dimension 6 immersed in $\mathbb{R}^7$.

As for the $SO\left(3\right)$ case, we define a diffeomorphism $\widehat{\exp}$ between $\mathbb{RP}^3 \times \mathbb{R}^3$ and $SE\left(3\right)$. Let $\widehat{\cdot}$ be the operator

$$\begin{array}{cccc} \widehat{\cdot}: & \mathbb{RP}^3 \times \mathbb{R}^3 & \longrightarrow & se\left(3\right) \\ & \left(r|u\right) & \longrightarrow & \begin{pmatrix} \widehat{r} & Fu^T \\ 0 & 0 \end{pmatrix} \end{array} \tag{4.19}$$

where

$$F = I - \frac{1}{2}\widehat{r} + \frac{2\sin\|r\| - \|r\|(1+\cos\|r\|)}{2\|r\|^2\sin\|r\|}\widehat{r}^2 \tag{4.20}$$

$\widehat{\exp} : \mathbb{RP}^3 \times \mathbb{R}^3 \leftrightarrows SE(3)$ is defined as the composition between exp and $\widehat{\cdot}$. It can be proved that $\widehat{\exp}$ is a diffeomorphism and that the following holds

$$\widehat{\exp}(r|u) = \exp\left(\widehat{r|u}\right) = \begin{pmatrix} \exp(\widehat{r}) & u^T \\ 0 & 1 \end{pmatrix} \tag{4.21}$$

4.2 Kinematic tree

Informally speaking, a *kinematic tree* is a set of hierarchically organized objects, called *bones*, which can move and rotate in a 3D space under certain constraints.

More formally, a kinematic tree K of order m is defined as a 4-tuple $K = (G, T, \rho, \Theta, \zeta)$ where $G = ([1,m], E)$ is a tree, T and ρ are particular diffeomorphism of type $\left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m \leftrightarrows SE(3)^m$ which are described later, Θ is a subset of $\left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m$ and ζ is an element of Θ.

Each element i belonging to $[1,m]$ represents a bone of the kinematic tree. From the graph theory, $(i,j) \in E$ means that j is the parent of i. Since G is a tree, there exists only one element, called root, which has no parent.

The manifold $C_K = \left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m$ is called *configuration space* of the kinematic tree K. Each element θ in C_K is called *configuration*, and it is defined to be valid if it belongs to $\Theta \subseteq C_K$, invalid otherwise. As a consequence, the set Θ is called the *space of the allowed configurations*. In particular, $\theta_i \in \mathbb{RP}^3 \times \mathbb{R}^3$ is called *configuration of the bone* i related to the configuration θ. Moreover, $\zeta \in \Theta$ is called *initial configuration* of K.

Let's note that C_K is shaped like the cartesian product of m hyper-cylinder of dimension 6 in $\mathbb{R}^7$, resulting in a manifold of dimension $6n$ immersed in $\mathbb{R}^{7n}$. The space of the allowed configurations Θ is a subset of this very particular manifold.

For a given configuration $\theta \in C_K$, $T(\theta) \in SE(3)^m$ is called *pose* of the kinematic tree K at configuration θ. For each bone i, $T_i(\theta) \in SE(3)$ is called *absolute pose* of the bone i at configuration θ and it represents the position and the orientation of the bone i at that configuration. $T(\zeta)$, i.e., the pose at the initial configuration, is called *initial pose* of the kinematic tree K.

$T(\Theta) \subseteq SE(3)^m$ is called *pose space* and represents the set of all the possible pose that the kinematic tree K can assume.

For a given configuration θ, $\rho_i(\theta) \in SE(3)$ is called *relative pose* of the bone i at configuration θ and, as we will see later, it represents the relative position and orientation of the bone i at configuration θ with respect to its parent bone.

In a kinematic tree T and ρ have to respect the following conditions. For each bone i and configuration θ,

$$\begin{cases} T_i(\theta) = \rho_i(\theta) & i \text{ is the root} \\ T_i(\theta) = T_j(\theta) \times \rho_i(\theta) & (i,j) \in E \end{cases} \tag{4.22}$$

i.e., the absolute pose $T_i(\theta)$ of each bone is defined to be equal to the relative one if the bone is the root, otherwise, it is defined to be equal to the absolute pose of its parent rotated by the its relative pose. By construction $T_i(\theta)$ is uniquely defined given all the relative poses $\rho_i(\theta)$, and moreover, each relative pose $\rho_i(\theta)$ is uniquely defined given $T_i(\theta)$ and $T_j(\theta)$.

The map ρ instead, is defined as

$$\begin{array}{rccc} \rho_i: & \left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m & \longrightarrow & SE(3) \\ & \theta & \longrightarrow & \widehat{\exp}(\theta_i) \end{array} \tag{4.23}$$

for every bone i.

From the above consideration, we can observe that the maps T and ρ are uniquely defined by the tree G. Therefore, a kinematic tree can be uniquely identified by its own hierarchical organization, its constraints and its initial configuration.

Let's note that, the configuration 0 is pretty useless in practical situations since it represents a pose where all the bones are placed at the origin aligned to the canonical reference system. For this reason, in place of configuration 0, the initial configuration ζ is used.

Typically, it is preferred to express a skeleton configuration with respect to the initial configuration ζ using, in place of $T(\cdot)$, the map $T(\cdot - \zeta)$. For instance, the constraints defining the set Θ are usually declared with respect to ζ.

4.3 Deformable models

Define a set Γ containing only ordered pairs of type $\gamma = (\Phi, \Psi)$ where Φ is a k-manifold in $\mathbb{R}^n$ and Ψ is a diffeomorphism. Call each element of this set, *shape* of Γ and assume that there exists one and only one singular element $\gamma_0 = (\Phi_0, \Psi_0) \in \Gamma$, called *reference shape*, such as for every other shapes (Φ, Ψ) of Γ, the related diffeomorphism Ψ has the form

$$\Psi : \Phi \longrightarrow \Phi_0 \tag{4.24}$$

and $\Psi_0 : \Phi_0 \rightarrow \Phi_0$ is equal to the identity map $\mathbb{1}$. In addition, call *reference shape*, any shape of type $(\Phi, \mathbb{1})$ and say that a shape $\gamma_1 = (\Phi, \Psi)$ *is referred by* γ_0 if and only if γ_0 is a reference shape $(\Phi_0, \mathbb{1})$ and $\Psi : \Phi \rightarrow \Phi_0$.

By the previous assumptions, for each pair of shapes (Φ_1, Ψ_1), (Φ_2, Ψ_2) there exists a diffeomorphism $\Psi_{12} : \Phi_1 \rightarrow \Phi_2$ defined as $\Psi_{12} = \Psi_2^{-1} \circ \Psi_1$ which maps points of Φ_1 into points of Φ_2.

$$\begin{tikzcd}
\Phi_0 & \Phi_2 \arrow[l, "\Psi_2"'] \\
\Phi_1 \arrow[u, "\Psi_1"] \arrow[ur, "\Psi_{12}"'] &
\end{tikzcd} \tag{4.25}$$

Let's define the metric d as

$$d\left(\left(\Phi_1, \Psi_1\right), \left(\Phi_2, \Psi_2\right)\right) = \int_{\Phi_1} \left\|p - \Psi_{12}\left(p\right)\right\|_2 ds + \int_{\Phi_2} \left\|q - \Psi_{21}\left(q\right)\right\|_2 ds \tag{4.26}$$

where $\left\|\cdot\right\|_2$ is the 2-norm of $\mathbb{R}^n$. It can be proven that d is a metric for Γ since it is non-negative, symmetric and the property $d\left(\gamma_1, \gamma_2\right) = 0 \Leftrightarrow \gamma_1 = \gamma_2$ and the triangle inequality hold.

If Γ is dense and connected with respect to the topology induced by the metric d and it admits an atlas such as Γ is a differential manifold of dimension k, then Γ is called *referenced deformable model* of dimension k in $\mathbb{R}^n$.

Informally speaking, Γ represents the set of all the shapes that the deformable model can assume. A smooth curve in Γ between two shapes γ_1 and γ_2 represents a smooth deformation of the model from the former shape to the latter one. This can also be seen as a natural time deformation of the model. The deformation smoothness is described by the metric d. Since Γ is dense and connected, given two shapes γ_1 and γ_2, there must exists at least one curve in Γ connecting γ_1 and γ_2. This means that the model must be able to assume all the shapes contained in at least one smooth deformation from γ_1 to γ_2.

Each allowed shape γ is connected by a diffeomorphism to a reference shape γ_0, thus, each point of γ_0 corresponds to one single point on γ, and viceversa. This allows us to extend the concept of simple manifold to the more natural one of deformable object. Let's, for instance, imagine a sphere that rotates around its own center. The sphere is always represented by the same identical manifold but, using the deformable model concept, the map Ψ is always different and thus the shape is too. Therefore, differently from the manifold concept, the concept of deformable model can represent a rotating sphere.

For practical reasons, in future, we will say that a point $p \in \mathbb{R}^n$ belongs to $\gamma = (\Phi, \Psi)$ and we write $p \in \gamma$, if it belongs to Φ. Moreover, we call $\Psi(p) \in \Phi_0$ *the point on the reference shape associated to* p and, viceversa, we call p the coordinates of the point $\Psi(p) \in \Phi_0$ on the shape γ.

Given a curve $\gamma(t)$ in Γ starting from γ_0, one can observe the trajectories of each point p of the reference shape γ_0, as the function

$$c(p)(t) = \Psi^{-1}_{\gamma(t)}(p) \tag{4.27}$$

where $\Psi_{\gamma(t)}$ is the diffeomorphism associated to the shape $\gamma(t)$. It can be proven that $c(p)$ is a smooth curve in $\mathbb{R}^n$. Note that $c(p)$ is a trajectory while, as defined above, $c(p)(t)$ are the coordinates of the point p on the shape $\gamma(t)$.

Let's note that, all the shapes that a deformable model can assume are constrained to have all the same genus because they must be always connected by a diffeomorphism of type (4.25).

Given a reference shape $\gamma_0 = (\Phi_0, \mathbb{1})$, the referenced deformable model having γ_0 as reference shape and containing all the possible shapes referred by γ_0, is called the *space of the deformations* of γ_0 and denoted with the symbol $\Gamma^*_{\gamma_0}$.

Let's now define the equivalence relation $\sim$ between referenced deformable models. Γ_1, Γ_2 are said equivalent if and only if they are equal to each other up to the choice of the reference shape. More formally, let $\gamma^1_0 = (\Phi^1_0, \mathbb{1})$ and $\gamma^2_0 = (\Phi^2_0, \mathbb{1})$ be respectively the reference shapes of Γ_1 and Γ_2. There must exists in Γ_2 a shape $\overline{\gamma^2}$ having as manifold Φ^1_0. The related diffeomorphism, call it $\overline{\Psi^2}$, has the form $\Phi^1_0 \to \Phi^2_0$. Now, for each element (Φ^1, Ψ^1) in Γ_1, there must exists an element (Φ^2, Ψ^2) in Γ_2 such that $\Phi^1 = \Phi^2$ and $\Psi^2 = \overline{\Psi^2} \circ \Psi^1$, i.e.,

$$\begin{tikzcd} \Phi^1 \arrow[r, "\Psi^1"] \arrow[rr, bend right, "\Psi^2=\overline{\Psi^2}\circ\Psi^1"'] & \Phi^1_0 \arrow[r, "\overline{\Psi^2}"] & \Phi^2_0 \end{tikzcd} \tag{4.28}$$

An (unreferenced) *deformable model* of dimension k in $\mathbb{R}^n$ is a class of equivalence of the relation $\sim$. We will identify a generic deformable model with the symbol Γ, the same one used for a generic referenced deformable model. In the future we will consider Γ either as a deformable model or as one of its referenced deformable model depending on case by case. So, we will use the term reference shape, the symbol γ_0 and its proprieties also for the unreferenced deformable models referring, in this case, to the specific referenced deformable model based on γ_0.

A deformable model inherits all the properties of the referenced one, moreover, it is not constrained to have a fixed reference shape since it contains all the possible referenced versions. Each pair of shapes that the model can assume is still connected by a diffeomorphism of type (4.25) and the curves in Γ can still be analyzed by looking at the trajectories of each point belonging to a chosen reference shape. Moreover, with this definition, the equality relation between two deformable model (inherited by the set theory) is equivalent to assert that one object is able to assume all the shapes of the other and viceversa.

Since each referenced deformable model is a differential manifold, the related deformable model is too and moreover, the metric d is still a metric for it. Local charts on Γ as well as diffeomorphism to other more simple manifolds can be built. In particular, in this latter case, the metric d can be pulled-back to the new manifold.

The concept of space of deformation $\Gamma^*_{\gamma_0}$ can be extended also to the unreferenced deformable models. We use the same symbol $\Gamma^*_{\gamma_0}$ to identify the class of equivalence containing the referenced deformable model $\Gamma^*_{\gamma_0}$. It is important to note that, in this case, given two different reference shapes γ_1, γ_2, their relative space of deformation $\Gamma^*_{\gamma_1}$, $\Gamma^*_{\gamma_2}$ can refer to the same space.

$\Gamma^*_{\gamma_0}$ is a deformable model and any connected submanifold of $\Gamma^*_{\gamma_0}$ is a deformable model.

A *discrete deformable model* is a deformable model which shapes are all geometric representations of meshes and the functions Ψ map consistently vertices into vertices, edges into edges, triangles into triangles and so on.

A *parameterization of a deformable model* Γ is a diffeomorphism $\Xi : \Theta \rightarrow \Gamma$, where Θ is a Lie group. Each element of $\theta \in \Theta$ is called *configuration.* Typically there exists a particular configuration $\zeta \in \Theta$ which is called *initial configuration* which refers to a particular shape $\Xi(\zeta) = \gamma_0$ called *initial shape.* The characteristics of this configuration depends on case to case. Let's note that using the pullback metric on Θ we can control smoothly the shape of the model. Moreover, with respect to Ξ and to a chosen configuration θ_0, the coordinates of each point p on the shape $\Xi(\theta_0)$ can be analyzed by the function

$$c^{\Xi}(p)(\theta, \theta_0) = \Psi^{-1}_{\Xi(\theta)\Xi(\theta_0)}(p) \tag{4.29}$$

Keep in mind that the first parameter of c^{Ξ}, in this case p, must belong to the shape identified by the third parameter, in this case $\Xi(\theta_0)$, and the resultant point belongs to the shape identified by the second parameter, in this case $\Xi(\theta)$.

Each deformable model has a lot of parameterization. We can classify them into families and characterize a deformable model observing if it admits or not a certain type of parameterization. This will be the topic of the next subsection.

Let's observe that given a class of parameterization X and a reference shape γ_0, the space of all the shapes assumed by at least one deformable model which admits a parameterization of class X and γ_0 as initial shape, is a subset of $\Gamma^*_{\gamma_0}$, called *X-set of deformations* obtained from γ_0 and denoted as $\Gamma^*_{\gamma_0,X}$. Since it is an infinite union of dense and connected subsets sharing γ_0, i.e., of deformable models, it is probably a submanifold of $\Gamma^*_{\gamma_0}$ and so a deformable model.

4.3.1 Families of deformable models

This subsection defines some of the most important parameterizations families for a generic deformable model. It is common to say that a deformable model is of type X if it admits a parameterization of class X.

If a deformable model admits a parameterization related to a kinematic tree then, the model is called *articulated deformable model.* In this case, if the model in $\mathbb{R}^n$ has dimension $n-1$, this specific parameterization is called *skinning* and it defines how the hyper-surface, the *skin*, is altered by the underlying skeleton represented by the kinematic tree. In general, a skinning $\Xi : \Theta \to \Gamma$ has the form

$$\begin{tikzcd}
\Theta \arrow[rr, "\Xi"] \arrow[dr, dashed, "T"'] & & \Gamma \\
& SE\left(3\right)^m \arrow[ur, dashed, "skin"'] &
\end{tikzcd} \tag{4.30}$$

where Θ is equal to $\left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m$, $T : \left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m \to SE\left(3\right)^m$ is the kinematic tree map and $skin : SE\left(3\right)^m \to \Gamma$ is a function controlling the skeleton influence on the skin. The initial configuration ζ for this type of parameterization is assumed equal to the initial configuration of the kinematic tree.

Linearly skinned deformable model

The most famous parameterization related to a kinematic tree is the *Linear Blend Skinning* (LBS or simply *Linear Skinning*).

A *linearly skinned deformable model* Γ based on the kinematic tree $K = (G, T, \rho, \Theta, \zeta)$ rigged at the initial shape $\gamma_0 \in \Gamma$ with skin map $\alpha : [1,m] \times \gamma_0 \to \mathbb{R}$, is a deformable model admitting a parameterization $\Xi : \Theta \to \Gamma$ where, for each $p \in \gamma_0$ and each $\theta \in \Theta$, the coordinates of p at configuration θ can be expressed as follows

$$c^{\Xi}(p)(\theta,\zeta) = \sum_{i=1}^{m} \alpha(i,p)\, T_i(\theta) \times T_i^{-1}(\zeta)\, p \tag{4.31}$$

and, the skin map α satisfies the convex hull propriety, that is,

$$\sum_{i=1}^{m} \alpha(i,p) = 1 \qquad \forall p \in \gamma_0 \tag{4.32}$$

Note that, the coordinates of p at configuration ζ, i.e., at the initial configuration, are

$$c^{\Xi}(p)(\zeta,\zeta) = p \tag{4.33}$$

and thus, the initial shape is

$$\Xi(\zeta) = \gamma_0 \tag{4.34}$$

The idea behind Equation (4.31) is that the motion of each point is equal to the weighted sum of all the motions that the point would undergo if considered as rigidly attached to every bones, one at a time.

Given an initial shape γ_0 and a kinematic tree K, the LBS-set of deformations obtained from γ_0 restricted to the use of K is called *skeletal subspace deformation* (SSD) and denoted as $\Gamma^*_{\gamma_0,LBS(K)}$.

A particular type of linearly skinned deformable model is the *rigid-body model.* In this case the skin map α has codomain $\{0,1\}$. Therefore each point of the model is attached to one and only one bone of the kinematic tree generating a rigid deformation of the entire skin.

A *skeleton representation* of the kinematic tree T is rigid-body model based on K. This definition gives the possibility to represent a kinematic tree with a physical object in $\mathbb{R}^n$, i.e., the skeleton. Obviously these representations are not unique.

Another type of parameterization similar to LBS is the so called *non-linear blend skinning* (NLBS). It expresses the coordinates of each point $p \in \gamma_0$ as follows

$$c^{\Xi}(p)(\theta,\zeta) = \sum_{i=1}^{m} \left(\alpha(i,p) + \beta\left(i,p,\log\left(T_i(\theta) \times T_i^{-1}(\zeta)\right)\right)\right) T_i(\theta) \times T_i^{-1}(\zeta)\, p \tag{4.35}$$

where $\log\left(T_i(\theta) \times T_i^{-1}(\zeta)\right)$ represents the angles related to the rigid rotation performed by the bone i at configuration θ with respect to the initial configuration ζ. $\beta : [1,m] \times \gamma_0 \times se(3) \to \mathbb{R}$ are non-linear weights which, differently to α, are dependent to the rotation angles. The obtained deformation is so composed by a linear blend part controlled by α, and a non-linear one controlled by β.

FFD deformable model

Any articulated deformable model can be seen as a manifold deforming itself according to the deformations of a simpler deformable model, i.e., the skeleton. It is easy to imagine that this idea can be extended to deformable models different from a simple skeleton. This is the concept behind the *Free Form Deformation* (FFD) parameterizations that we are going to describe.

Given a deformable model Δ parameterized by $\Pi : \Theta \to \Delta$ with initial configuration ζ and initial shape $\delta_0 = \Pi(\zeta)$, let Γ be another deformable model and γ_0 one of its shape, we say that Γ is an FFD deformable model based on $(\Delta, \Pi, \zeta, \delta_0, \gamma_0)$ and weight map $\alpha : \delta_0 \times \gamma_0 \to \mathbb{R}$ if and only if it admits a parameterization $\Xi : \Theta \to \Gamma$ defined as

$$\Xi : \Theta \xrightarrow{\Pi} \Delta \xrightarrow{FFD} \Gamma \tag{4.36}$$

where $FFD : \Delta \to \Gamma$ has to satisfy the following statement: for each $p \in \gamma_0$ and each $\theta \in \Theta$ the coordinates of p at configuration θ are

$$c^{\Xi}(p)(\theta, \zeta) = \int\limits_{\delta_0} \alpha(q, p) \;\; R_{\Pi(\theta)}\left(c^{\Pi}(q)(\theta, \zeta)\right) \times R^{-1}_{\delta_0}(q) \;\; p \; dq \tag{4.37}$$

where $R_{\delta_0}(q)$ is the local coordinate system[1] of the manifold δ_0 at the point q. $R_{\Pi(\theta)}\left(c^{\Pi}(q)(\theta, \zeta)\right)$ is the local coordinate system for the manifold $\Pi(\theta)$ at the point $c^{\Pi}(q)(\theta, \zeta)$, i.e., the point associated to q at configuration θ. α has to satisfy the convex hull propriety which, in this case, is

$$\int\limits_{\delta_0} \alpha(q, p) = 1 \qquad \forall p \in \gamma_0 \tag{4.38}$$

Normally α is defined to be a function decreasing with the square of the distance between p and q. For instance, it could be defined as a gaussian centered in q having some free parameters that allow a slope control.

As one can note, Equation (4.37) is the continuous version of Eq. (4.31). Δ is called *driver model* or *control model* and, in practical situation, is a discrete deformable model. In this case, every vertex of the mesh is called *control point*. Note that the mesh inherits the concept of local coordinate system even if it is a discrete surface, in fact, these systems are defined in a piecewise continuous way.

[1] The local coordinates system for a given point on a manifold is a typical property of the manifolds in $\mathbb{R}^n$. It can be obtained starting from a basis of the tangent space and extending it to the entire space in such a way that the resultant reference system is continuous all along the manifold.

Shape interpolation deformable models

Another class of deformable models is the *simple morphing deformable models.* Each element of this family can be uniquely identified by a finite number of shapes. In fact, a deformable model of this type is an affine space of deformation $\Gamma^*_{\gamma_0}$ defined by a basis of shapes and by a translation shape γ_0.

Formally, a simple morphing deformable model Γ, based on m shapes γ_0, γ_1, …, γ_{m-1} where each γ_i is a shape referred by γ_0 and γ_0 is a reference shape, is a deformable model admitting a parameterization $\Xi : \mathbb{R}^m \rightarrow \Gamma$ where, for each $p \in \gamma_0$ and each $\theta \in \Theta = \mathbb{R}^m$, the coordinates of p at configuration θ are described by following equation

$$c^{\Xi}(p)(\theta, 0) = p + \sum_{i=1}^{m-1} \theta_i \left(\Psi_i^{-1}(p) - p \right) \tag{4.39}$$

θ represents the local coordinates of the shape $\Xi(\theta)$ inside the particular affine space of deformation $\Xi(\Theta)$ identified by the basis $\left(\gamma_1, \ldots, \gamma_{m-1}\right)$ and the origin γ_0. Considering both $\mathbb{R}^m$ and Γ as vector spaces, it results that the map $\Xi : \mathbb{R}^m \rightarrow \Gamma$ is an affine map.

However, a deformable model of this type presents natural deformations only near one of the basis shapes γ_0, γ_1, …, γ_{m-1}.

The above concept can be extended defining Ξ as a generic function between $\mathbb{R}^m$ and a set Γ which, in this case, is no longer, in general, an affine space of deformations but only a simple subset. The idea is to define a class of deformable models which can be used to represent natural objects avoiding the drawbacks of the simple morphing deformable models.

A *shape interpolation deformable model* Γ based on m configurations θ_0, θ_1, …, θ_{m-1} and m shapes γ_0, γ_1, …, γ_{m-1} where each γ_i is a shape referred by γ_0 and γ_0 is a reference shape, is a deformable model admitting a parameterization $\Xi : \mathbb{R}^m \rightarrow \Gamma$ such that

$$\Xi(\theta_i) \approx \gamma_i \qquad \forall i = 1, \ldots, m \tag{4.40}$$

i.e., such that Ξ approximates each shape γ_i at the respectively configuration θ_i. The concept of approximation depend upon the chosen interpolation method. The most used one is the *radial basis function interpolation* which defines the class of deformable models called *RBF shape interpolation deformable models.* More formally, the coordinates of each point $p \in \gamma_0$ at the configuration $\theta \in \mathbb{R}^m$ are

$$c^{\Xi}(p)(\theta, \theta_0) = p + \sum_{i=1}^{m-1} \psi\left(\|\theta - \theta_i\|\right)\ w_i \tag{4.41}$$

where $\psi : \mathbb{R} \to \mathbb{R}$ is the chosen radial basis function, which in most of the cases could be a multiquadric, a polyharmonic spline, a thin plate spline or a simple gaussian like the following

$$\psi(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{4.42}$$

where σ is fixed a priori. Weights $w_i \in \mathbb{R}^n$ are vectors chosen in order to satisfy the condition of Eq. (4.40). The optimum choice for these weights is the one which minimizes the following functional

$$\sum_{i=0}^{m-1} d\left(\Xi\left(\theta_i\right), \gamma_i\right) \tag{4.43}$$

In case of discrete deformable models, the optimal choice can be computed using standard matrix methods like the pseudoinverse.

The introduced parameterization is significant only inside the convex hull of the configurations θ_0, θ_1, ..., θ_{m-1}. Outside this set, all the shapes look like the initial one γ_0 since all the basis functions tend to zero.

Let's note that, a shape interpolation deformable model can be uniquely identified by its basis shapes γ_0, γ_1, ..., γ_{m-1} and by its interpolation method. Indeed, except for some singular cases, whatever the m-tupla of configurations $(\theta_0, \theta_1, \ldots \ldots, \theta_{m-1})$ is chosen, $\Xi\left(\mathbb{R}^m\right)$ identifies always the same deformable model.

Therefore, given a set of shapes γ_0, γ_1, ..., γ_{m-1} and an interpolation method ψ, the unique shape interpolation deformable model with these characteristics is obviously a subset of $\Gamma^*_{\gamma_0}$ but as we state before it is not in general neither an affine space nor a subspace. In spite of this, this set is often called *pose space deformation* obtained by γ_0, γ_1, ..., γ_{m-1} and ψ.

Other families of deformable models

In literature, there exist several other classes of deformable models and related parameterizations. Each one try to describes deformable models which can be used to represent physical objects. Some of them are based on *anatomical models* virtually located between the skeleton and the skin. Other classes, like the *surface oriented FFD deformable model*, are strongly based on the skin final appearance.

An important family of deformable models is the class of the *as rigid as possible deformable models*. These models allow a parameterization based only on few points belonging to their initial shape. More formally, an as rigid as possible deformable model Γ, based on m points $(p_1, \ldots, p_m)$ and on the initial shape γ_0, is a deformable model admitting a parameterization $\Xi : (\mathbb{R}^n)^m \to \Gamma$

with initial configuration $\zeta = (p_1, \ldots, p_m)$ where, for each $\theta \in \Theta = (\mathbb{R}^n)^m$, the following statement holds

$$c^{\Xi}(p_i)(\theta, \zeta) = \theta_i \qquad \forall i = 1, \ldots, m \tag{4.44}$$

and all the other points belonging to γ_0 deform minimizing the shell energy between γ_0 and $\Xi(\theta)$.

The idea is to move the model along the geodesics introduced by the shell energy which imposes a smooth and locally rigid deformation, i.e., preserving the local shape. Since the shell energy is invariant to rigid rotations and translations, at least n points are needed to have a unique solution. The first point indeed fixes the translation, the second point fixes a rotation axis and defines a stretch along this axis, and so on. After n points, the rigid motion is fixed and all the further points define a unique solution.

4.3.2 The analysis of a deformable model

Given a physical deformable model Γ, the analysis of Γ aims to determine the class of the model and to propose a parameterization of it.

The deformations present on a real object are much more complex than the ones propose in the previous subsection. For instance, the presence of cloths or fur, as well as internal or external interactions, deform the shape of an object in a very complex way which is hard to describe by a mathematical model.

Good approximations can be obtained by sampling the real object Γ and using interpolation to recover all the remaining shapes. The resultant model is a shape interpolation deformable model that approximates the real one. The more shapes of the real model are acquired the more accurate is the approximation. Clearly, to achieve a good results, a lot of effort is needed either in terms of computational time, storage size and human resource.

Observing a real object, we can note that its deformations can be subdivided into big deformations and small deformations. A formal way to say this recalls the concept of *multi-band analysis of a manifold*. The idea is that, basing on a priori defined surface kernel, two operators in the space of the manifolds can be built, namely an analysis operator $A : M \to M^b$ and a synthesis operator $S : M^b \to M$. The aim is to represent each manifold using b different bands. This concept can be extended to the space of the deformable models leading to an analysis and synthesis scheme of this type:

$$\begin{tikzcd}
 & & \Gamma_1 \arrow[ddr] & & \\
 & & \Gamma_2 \arrow[dr] & & \\
\Gamma \arrow[r] & \boxed{A} \arrow[uur] \arrow[ur] \arrow[dr] & \vdots & \boxed{S} \arrow[r] & \Gamma \\
 & & \Gamma_b \arrow[ur] & &
\end{tikzcd} \tag{4.45}$$

where Γ is a generic deformable model and Γ_1, Γ_2, ..., Γ_b are the deformable models in which it is decomposed by A. Each Γ_i is called the i-th band of Γ. The operator S recovers the original deformable model from its bands.

The concept of band is related only to the spatial characteristics of the manifold related to each shape without taking into account about any deformation. Bands are usually related to the maximum curvature of the surface, therefore, low bands represent the coarse shape of the object while high bands represent the details of the surface.

These two operators offer a layered approach to the analysis of a deformable models. These layers, in a real objects, behaves almost independently to each other and, typically, high band layers have a very low impact onto the final manifold. Therefore, a real object can be decomposed into layers which can be modeled independently by different type of deformable models. For instance low layers can be modeled by a linearly skinned deformable model, middle layers using an FFD deformable model and finally high layers using a shape interpolation deformable models acquired by a 3D scanner.

Let's note that now, it is clear how to formally represent a linearly skinned deformable model as a combination of a skeleton and an FFD model, respectively in the low layer and in the high layer. In this case, the FFD model generates the non-rigid deformations introduced by the LBS.

Similar considerations can be made for the parameterization of a deformable model, i.e., also the parameterizations can be analyzed in a layered approach. Formally, given a deformable model Γ and its decomposition $(\Gamma_1, \ldots, \Gamma_b)$ according to a specific scheme, given a set of parameterization $(\Xi_1, \ldots, \Xi_b)$ where each $\Xi_i : \Theta_i \to \Gamma_i$ is a valid parameterization for Γ_i, the compound parameterization is defined as

$$\begin{array}{rccc} \Xi : & \Theta_1 \times \ldots \times \Theta_b & \longrightarrow & \Gamma \\ & (\theta_1, \ldots, \theta_b) & \longrightarrow & S\left(\Xi_1\left(\theta_1\right), \ldots, \Xi_b\left(\theta_b\right)\right) \end{array} \tag{4.46}$$

and it is a valid parameterization for Γ.

4.3.3 The synthesis of a deformable model

Contrary to the analysis, the synthesis aims to define a new deformable model basing on the tools offered by the analysis, namely, the parameterization and the multi-band analysis.

Computer graphics world is the biggest demander for synthesis tools of deformable models since they are largely used in the making of both 2D and 3D *CGI footage*[2] and of *real-time animations*. Both photorealistic and non-photorealistic CGI movies as well as video games require tons of *computer animated characters* to represent imaginary actors or to be used as temporary or permanent replacements for the real ones. Therefore, the efficiency of these tools inside a character production pipeline is the most important requirement that they have to satisfy. More specifically, they must be easy and intuitive to use and do exactly what the animator wants in as few steps as possible.

A standard character production pipeline consists in two main phases namely, a *modeling phase* and an *animation phase*. In former one, the shape of the actor fixed in a static pose is created. Then, during the animation phase, some *animation controllers* are assigned to this shape, i.e., the previous defined synthesis tools. At the end, the *animation*, considered as a time-sequence of shapes, is finally generated. The idea is to transform a static object into a deformable one parameterized in such a way that it is easy for an operator to control its animation. The animation itself will result as a curve inside this deformable model.

With the previous defined formalism, the production pipeline can be viewed in this way: during the first step, a shape γ_0 is defined, then, in the second one, the animator looks inside the space of deformation $\Gamma^*_{\gamma_0}$ related to γ_0 for a specific deformable model Γ, i.e., a particular subset, which satisfies the given requirements and, in the end, a parametric curve inside Γ is traced representing the final animation. Complex production pipelines involve the definition of multiple shapes during the first stage. In this case, the search inside $\Gamma^*_{\gamma_0}$ has to be restricted to the only subsets containing all these shapes.

The modeling phase is usually made by hands or by scanning real objects like humans, animals or hand-made sculpture made for instance, of clay or plaster.

The animation phase instead, is the direct evolution of the *stop-motion* techniques largely used in the past movie production pipelines. The animation controllers can be viewed as the replacement for the old armatures inserted inside the clay models. In place of shooting one frame at a time, CGI technology offers three

[2] CGI means Computer-Generated Imagery or Imaging.

new types of solution namely the *key-frame animation*, the *physical simulation* and the *motion capture*.

All of them aim to generate a time-sequence of shapes, i.e., the animation. Key-frame animation defines the shapes[3] that the model has to assume at some given times, these are called the *key-frames*. The computer then, interpolates the missing frames. Formally, each key-frame represents a point in the manifold Γ and, since the presence of an animation controller, it represents also a point in the configuration space of the controller. The curve is interpolated in the configuration space and then transposed in the Γ domain.

On the contrary, physical simulation needs only to know the initial state of the model. The curve is then traced as a unique solution of a PDE equation describing the physical interactions between the object and the scene.

Motion capture techniques instead, make the use of a real actor to animate a virtual one. In some cases, the virtual actor is just the model of the real one but in some others, it's not. In these latter cases, a further step, known as *motion retargeting* step, is required to transfer the motion between the two models. In both cases, both the real and the virtual actors are considered as deformable models parameterized with the same controller and the result of the motion acquisition process is always a curve in the configuration space.

For human like models, or in general for articulated deformable models, the controller assignment phase is further divided in two stage namely the *rigging stage* and the *skinning stage*. The former aims to place a skeleton structure inside the so called rest shape of the model, i.e., to find the so called rest configuration of the skeleton. The *rest shape* of an articulated deformable model is the one created by the modeling phase. The *rest configuration* and the *rest pose* are respectively the configuration and the pose of the skeleton related to the rest shape of the model. The skinning stage instead, defines the skinning parameters, i.e., the parameters which define how the skin is deformed according to the underlying skeleton. Moreover, this phase usually involves the definition of further controllers for modeling muscles, tendons, facial expressions and, in general, all the internal organs. The model is so represented in a multi-layer structure.

Historically, the first animation controller was described in the Parke's work on the facial animations [115]. Before 1974, every key-frames were explicitly defined by hand inside the deformation space and the missing frames recovered using an

[3] Note that we consider that an object is assuming different shapes also if it is only rotated and/or translated in the space. Moreover, the entire scene can be considered as a big deformable object.

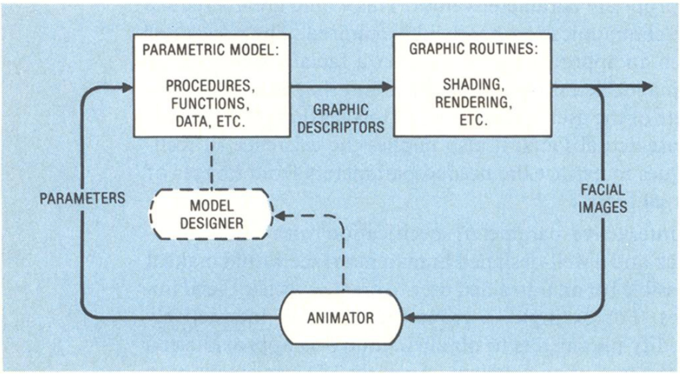


Figure 4.1: *Feedback scheme for the synthesis of facial expressions [115].*

early shape interpolation technique based on vertex coordinates interpolation in the 3D space. In his work, Parke described both the concept of parameterization and of animation controller suggesting an interesting feedback scheme for the synthesis of the facial expressions. We repropose that scheme in Figure 4.1 since it still represents the work of an animator.

Free-form deformations were introduced by Sederberg and Parry [136] in 1986 as a technique where the objects were deformed by warping a parallelepiped of control points. The definition we had proposed in the previous section is an extension of the work made by the Alias/Wavefront team in 2000 [142].

Articulated deformable models were first described in 1988 by [75] and [90] for modeling respectively an arm and a hand. Both methods adopted the skinning scheme described in Eq. (4.30), where each vertex is mapped to the bones and a function of the joint angles is used to deform the vertices. LBS comes out as a generalization of these two approaches even if, its exact definition have never been published.

Chadwick, Haumann, and Parent [26] introduced the first multi-layered and physically inspired approach to skin deformation in 1989. In their model an FFD volume abstractly represents the underlying body tissues such as the muscles and the fatty layer. The skin deformations, due to the skeleton movements, are mediated by the FFD interlayer. After this first work, other more complex approaches as well as new methodologies for defining and controlling these structures were developed.

Even if simple morphing is at the base of the early key-frame interpolation, it is not exactly clear when it started to be used as a controller. However, it is known for sure that it was largely used in the short “Tony de Peltrie” [15] in

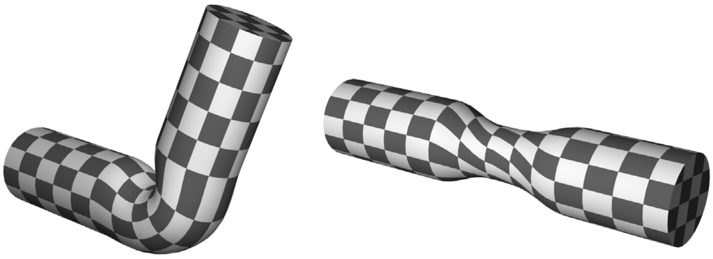


Figure 4.2: *LBS typical defects: joint collapsing artifacts (left) and candy-wrapping artifacts (right). Image taken from [66].*

1985. Later, several complex shape interpolation were introduced, in particular, in 2000, [84] defined the pose deformation space by treating the deformations as a RBF interpolation between key-shapes, exactly as we had defined in the previous section.

Despite of all these complex techniques, for years, LBS was considered the standard controller for animation. It was one of firsts to be implemented in commercial animating software and, for sure, the most used one. Now it is still the most adopted for real-time animations. LBS's fame is due to the fact that it is a very simple and versatile technique having a low computational cost. However, it has a lot of limitations, most of them due to the fact that its space of deformations do not include a lot of the natural deformations that occur in a real human. As a result, the model can only approximates, by projecting on its space, the natural deformations and, no amount of adjusting the α parameters will produce better results. These shortcomings are mostly visible near the shoulders and hips, which, in fact, are the most difficult parts to skin with LBS. The typical defects of LBS are the so called *joint-collapsing* and *candy-wrapping* artifacts showed in Figure 4.2.

In order to overcome these limitations, several authors propose LBS definitions which extend the space of deformation maintaining the linearity of the skinning operator. The main drawback is an increase of the number of the α parameters that has to be adjusted. The most important techniques, adopting this strategy, are the Animation Space [101] and the Multi-Weight Enveloping [160]. Other authors propose to replace matrix computation by more sophisticated tools to blend weights, such as log-matrices [36] or dual quaternions [71].

Assigning the weights α is a semi-automatic process that requires huge amount of human intervention. The basic technique is to use the Euclidean distance between each point and the closest bones which however, lead to a non-realistic deformation since the anatomy of the character is not taken into account. More complex solutions have been developed such, for instance, the heat equilibrium approach of [12] and the Kinodynamic skinning approach of [3]. In particular, in this last one, the weights are time-dependent and they are automatically computed using a physical model.

Nowadays, ordinary animating software implement the skinning using LBS as a base layer and some FFD layers to model specifically hips, shoulders, muscles, tendons and facial expressions. In particular, the FFD layers are usually controlled by the angles of a specific joint, i.e., their configuration space is mapped into $se\,(3)$. The resultant deformable model looks similar to an NLBS one but it is simpler to control and to define. Shape interpolation techniques are also implemented but, due to the enormous amount of needed data, they are rarely used. However, these last techniques can achieved more accurate and realistic deformations and recently a commercial database of humans has been built to perform shape interpolation. This database, called SCAPE [4], contains several models acquired in different poses that can be interpolated, including their deformations, to fit almost any human in almost any pose.

4.4 Pose estimation: our approach

In this work, we consider each trackable element of the real scene as a deformable model parameterized with an LBS controller. The rest shape, the skeleton and the rest configuration of each element of the scene are acquired using the system described in Chapter 3.

Skin parameters α are estimated using a normalized non-linear distance function between each the vertex of the mesh and each bone of the skeleton, i.e.,

$$\alpha_{i,k} = e^{-\frac{d(i,k)^2}{2\sigma^2}} \tag{4.47}$$

where $d\,(i,k)$ is the distance between the vertex k and the segment representing the bone i, and σ is a real value proportional to the bone length. Each $\alpha_{i,k}$ is then normalized to satisfy the convex property.

However, this type of estimation is not very accurate and needs some manual adjustments. A fully automatic estimation of both skeleton initial pose and skin parameters can be obtained using the method proposed by Baran and Popovic [12]. Instead of using a non-linear distance function, Baran and Popovic compute the weights α by solving an heat equilibrium equation. Source code and documentation is also provided in their web site.

The output of the pose estimation algorithm is a curve $\theta\left(t\right)$ in the configuration space of the LBS controller. Assuming a skeleton made of m bones and a surface mesh made of n vertices, the output curve is formally defined as

$$\theta : \mathbb{R} \to \left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m \tag{4.48}$$

In computational terms, this curve is represented by a finite list of elements of type $\left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m$, each represented by 6 real numbers. Therefore, the space of the allowed configurations $\Theta \subseteq \left(\mathbb{RP}^3 \times \mathbb{R}^3\right)^m$ is numerically represented by a subset of $\mathbb{R}^{6m}$. With this notation, the configuration of the bone i related to the configuration $\theta \in \Theta$ becomes

$$\left(\theta_{6i+1}, \theta_{6i+2}, \theta_{6i+3}, \theta_{6i+4}, \theta_{6i+5}, \theta_{6i+6}\right) \tag{4.49}$$

where the first part $\left(\theta_{6i+1}, \theta_{6i+2}, \theta_{6i+3}\right)$ represents the bone rotation and the second one, $\left(\theta_{6i+4}, \theta_{6i+5}, \theta_{6i+6}\right)$, its translation.

The deformable model is obviously discrete and we denote with $v_k\left(\theta\right)$ the homogeneous coordinates of the k-th vertex of model at the configuration θ, i.e.,

$$v_k\left(\theta\right) = c^{\Xi}\left(v_k\left(\theta_0\right)\right)\left(\theta, \theta_0\right) \tag{4.50}$$

where θ_0 is the rest configuration of the skeleton and each $v_k\left(\theta_0\right)$ is a vertex of the rest shape.

We define the space of the allowed configurations Θ as a $(6m)$-dimensional box aligned with the canonical basis. A particular element of this box is defined to be ζ, the initial configuration of the deformable model. This configuration is common to every model of the scene.

The choice of defining such a configuration is due to the necessity to be independent from the rest configuration θ_0, i.e., the configuration assumed by the actor during the shape acquisition, which is different from model to model. In this way, the constraints defining Θ can be declared only once and they can be applied for every model with similar skeletal structure. In particular, for human models, we chose to use a Da Vinci's Vitruvian man's configuration as ζ.

Therefore, each constraint defining Θ has this form

$$\left\{\theta_{j,min} \leqslant \theta_j \leqslant \theta_{j,max}\right\} \qquad j = 1, 2, \ldots, (6m) \tag{4.51}$$

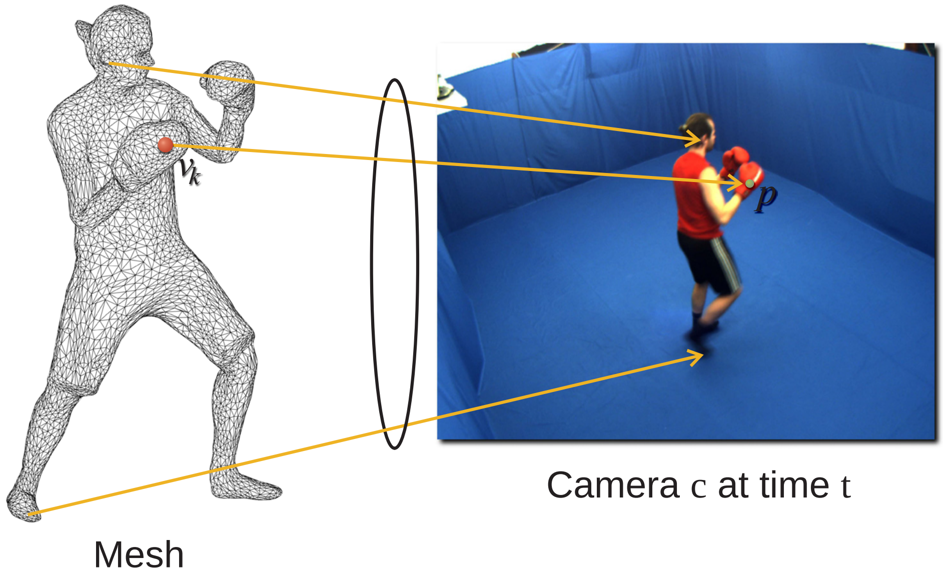


Figure 4.3: *Few vertex to image point correspondences extracted at time t for the camera c.*

where $\theta_{1,min}, \ldots, \theta_{6m,min}$ and $\theta_{1,max}, \ldots, \theta_{6m,max}$ are the same for every human model.

Equation (4.31) becomes

$$v_k(\theta) = LBS_k(\theta)\, v_k(\theta_0) \tag{4.52}$$

where we denote with $LBS_k(\theta)$, the *linear blend skinning operator* for the vertex k, defined as follows,

$$LBS_k(\theta) = \sum_{i=1}^{m} \alpha_{i,k} T_i(\theta) \times (T_i(\theta_0))^{-1} \tag{4.53}$$

The proposed motion tracking algorithm addresses the estimate of $\theta(t)$, given the previous estimate $\theta(t-1)$ and the set of the images taken at time t and at time $t-1$ from a set of q cameras $\{V_1, \ldots, V_q\}$.

These images are first preprocessed to get the motion cues, i.e., silhouette and optical flow. Then, from each channel of information a set of vertex to image point correspondences is extracted, each represented by a quadruple of type (w, k, p, c) where w is a real value weighting the correspondence, k is an index representing the 3D mesh vertex v_k, c is a camera view index and $p \in \mathbb{R}^2$ is the actual projection of v_k on the view V_c.

The meaning of such a correspondence is that, for a correct pose estimation, the vertex of index k has to be seen from camera c in the image point p. The weight w quantifies the importance of this statement (see Fig. 4.3). This formulation allows us to treat both type information in a similar way.

Given these correspondences, the algorithm defines an objective function $g\left(\theta\right)$ having a global minimum in the current configuration $\theta\left(t\right)$. Then, $g\left(\theta\right)$ is optimized using $\theta\left(t-1\right)$ as starting point in the minimization.

Objective function $g\left(\theta\right)$ is first examined and next the algorithm is described.

4.4.1 The objective function

Given a set of z correspondences of type $\left(w_s, k_s, p_s, c_s\right)$ where $s \in \{1, \dots, z\}$, $k \in \{1, \dots, n\}$ and $c \in \{1, \dots, q\}$. Denote with $\Pi_V\left(\cdot\right)$ the projection of a point in the 3D space to a 2D point in the image space of view V, i.e.,

$$\Pi_V\left(\cdot\right) = \frac{1}{\eta_V^z\left(\cdot\right)} \left[\begin{array}{c} \eta_V^x\left(\cdot\right) \\ \eta_V^y\left(\cdot\right) \end{array} \right] \tag{4.54}$$

where $\eta_V\left(\cdot\right) = \left(\eta_V^x\left(\cdot\right), \eta_V^y\left(\cdot\right), \eta_V^z\left(\cdot\right)\right)$ is the affine map transforming the world space coordinates to the camera space coordinates of the view V, defined as

$$\eta_V\left(x\right) = R_V x + T_V \tag{4.55}$$

where $\left[\begin{array}{cc} R_V & T_V \end{array} \right] = K_V E_V$ and K_V, E_V are respectively the intrinsic and the extrinsic matrices of the view V. Therefore,

$$\begin{array}{rcl} \eta_V^x\left(x\right) & = & R_V^1 x + T_V^1 \\ \eta_V^y\left(x\right) & = & R_V^2 x + T_V^2 \\ \eta_V^z\left(x\right) & = & R_V^3 x + T_V^3 \end{array} \tag{4.56}$$

where

$$R_V = \left[\begin{array}{c} \cdots R_V^1 \cdots \\ \cdots R_V^2 \cdots \\ \cdots R_V^3 \cdots \end{array} \right], \qquad T_V = \left[\begin{array}{c} T_V^1 \\ T_V^2 \\ T_V^3 \end{array} \right] \tag{4.57}$$

We say that the actual model configuration θ is the one which minimizes the following functional

$$g\left(\theta\right) = \sum_{s=1}^{z} \beta_{k_s} w_s \left\| \Pi_{V_{c_s}}\left(v_{k_s}\left(\theta\right)\right) - p_s \right\|^2 \tag{4.58}$$

and belongs to the space of the allowed configurations Θ.

Note that, $v_{k_s}(\theta)$ is the same function defined above in Eq. (4.52). β_{k_s} instead, is a real number inversely proportional to the sampling rate of the surface near the vertex v_{k_s}. We assume this value constant for every pose that the deformable model can assume and we initialize it as the area of all the faces incident on vertex v_{k_s}.

This term is very useful since it weights more vertices attached to larger pieces of surface and viceversa, giving the possibility of estimating the pose of also non uniformly sampled meshes.

Constrained optimization of functional (4.58) can be performed by classical gradient descent approaches with hard constraints since the gradient of $g(\theta)$ can be easily derived in closed form. Indeed for each j,

$$\frac{\partial g(\theta)}{\partial \theta_j} = 2 \sum_{s=1}^{z} \beta_{k_s} w_s \left(\Pi_{V_{c_s}} \left(v_{k_s}(\theta) \right) - p_s \right) \frac{\partial \left(\Pi_{V_{c_s}} \circ v_{k_s} \right)}{\partial \theta_j}(\theta) \tag{4.59}$$

where

$$\frac{\partial \left(\Pi_{V_{c_s}} \circ v_{k_s} \right)}{\partial \theta_j}(\theta) = \frac{1}{\eta^z_{V_{c_s}}\left(v_{k_s}(\theta) \right)^2} \left[\begin{array}{r} \eta^z_{V_{c_s}}\left(v_{k_s}(\theta) \right) \frac{\partial \eta^x_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) - \\ \eta^z_{V_{c_s}}\left(v_{k_s}(\theta) \right) \frac{\partial \eta^y_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) - \\ -\eta^x_{V_{c_s}}\left(v_{k_s}(\theta) \right) \frac{\partial \eta^z_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) \\ -\eta^y_{V_{c_s}}\left(v_{k_s}(\theta) \right) \frac{\partial \eta^z_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) \end{array} \right] \tag{4.60}$$

and

$$\begin{aligned} \frac{\partial \eta^x_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) &= R^1_{V_{c_s}} \frac{\partial v_{k_s}}{\partial \theta_j}(\theta) \\ \frac{\partial \eta^y_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) &= R^2_{V_{c_s}} \frac{\partial v_{k_s}}{\partial \theta_j}(\theta) \\ \frac{\partial \eta^z_{V_{c_s}} \circ v_{k_s}}{\partial \theta_j}(\theta) &= R^3_{V_{c_s}} \frac{\partial v_{k_s}}{\partial \theta_j}(\theta) \end{aligned} \tag{4.61}$$

From Eq. (4.52), the partial derivatives of $v_{k_s}(\theta)$ are

$$\frac{\partial v_{k_s}}{\partial \theta_j}(\theta) = \sum_{i=1}^{m} \alpha_{i,k_s} \frac{\partial T_i}{\partial \theta_j}(\theta) \times \left(T_i(\theta_0) \right)^{-1} v_{k_s}(\theta_0) \tag{4.62}$$

where $\partial T_i / \partial \theta_j$ can be computed exploring the kinematic chain associated to the bone i, i.e., its root path. More precisely, let this chain be $\Lambda = (h_1, \ldots, h_l)$, where each h_b is a bone index, h_1 is the index of the root and h_l is the equal to i. By the definition of kinematic chain, the following equation holds

$$T_i(\theta) = \prod_{b=1}^{l} \rho_{h_b}(\theta) \tag{4.63}$$

Let o be the index of the bone associated to the degree of freedom θ_j, i.e., $\lfloor (j-1)/6 \rfloor = o$. If o belongs to the chain Λ at index d, or in other words, if there exists a d such that $o = h_d$, then

$$\frac{\partial T_i}{\partial \theta_j}(\theta) = \rho_{h_1} \times \ldots \times \rho_{h_{d-1}} \times \frac{\partial \rho_o}{\partial \theta_j}(\theta) \times \rho_{h_{d+1}} \times \ldots \times \rho_{h_l} \tag{4.64}$$

otherwise,

$$\frac{\partial T_i}{\partial \theta_j}(\theta) = 0 \tag{4.65}$$

By the definition (4.23), the partial derivatives of $\rho_o(\theta)$ coincide with the partial derivatives of the map $\widehat{\exp}(\theta_{6o+1}, \ldots, \theta_{6o+6})$ which are straightforward.

The Levenberg-Marquardt method [82], [92] was found rather stable and reliable with respect to other methods especially when the configuration θ approaches the optimal solution. In this case, only the closed form of the jacobian $\partial\left(\Pi_{V_{c_s}} \circ v_{k_s}\right)/\partial\theta_j$ is needed.

However, for all these algorithms, a normalization of the configuration space is needed to account for the different valid intervals sizes of each degree of freedom.

A two levels pyramidal approach was also attempted. During the first level, a solution for the pose estimation problem is found keeping foots, forearms and head blocked to the previous frame configuration. During the second level, this solution is refined freeing all these bones. This procedure was found to be more robust to local minima.

4.4.2 Correspondences extraction

The use of the objective functional described in Eq. (4.58) requires to extract a set of valid correspondences from the given images. To this aim we used two types of motion cues namely, optical flow and silhouette information.

Since a blue screen is adopted, silhouettes can be extracted using a standard HLS keyer [164]. A KLT operator [87] is instead independently applied to each video stream to get the optical flow.

The result of this latter process is a large set of 2D correspondences on consecutive frames. Let's call $(y_{t-1}, y_t) \in (\mathbb{R}^2 \times \mathbb{R}^2)$ one of such correspondences viewed by the camera c. Since $\theta(t-1)$ is known, we can easily find which vertex

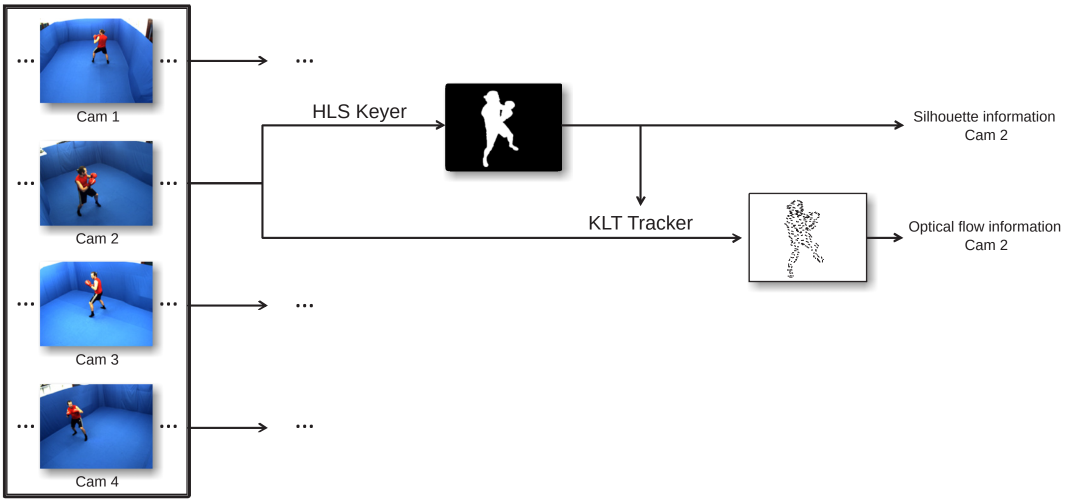


Figure 4.4: *Preprocessing Pipeline. Each video stream is treated separatively generating two information channels for stream. The correspondences extraction is performed separatively for each of these channels.*

of the deformable model is projected onto image point y_{t-1}. This can be achieved by computing the z-buffer and finding the visible mesh vertex with the nearest projection to y_{t-1}. If such a vertex has index k, then a valid correspondence for our algorithm is (w, k, y, c). The weight w is defined to be proportional to the confidence obtained in the related 2D correspondence.

On the other hand, silhouette information is used similarly to the ICP approach [16] with the difference that our method does not operate in the 3D space but in the image space. A shape matching metric is adopted to quantify the accuracy of each 2D silhouette correspondence. In particular, we measure the gradient similarity on the segmented images, i.e., we privilege edges with similar orientation.

Differently than optical flow, silhouette information is updated also during the optimization phase by the following method: assuming that we are in the process of estimating $\theta(t)$ with our tracking algorithm, call $\widetilde{\theta}$ the estimate of configuration $\theta(t)$ at the current algorithm iteration and call $I_c(t)$ the segmented image recorded by camera c at time t,

```
for each camera $c$ do
    Find all the vertices $v_k$ of the deformable model with configuration $\widetilde{\theta}$
    which belongs to the silhouette viewed from $c$;
    Project such vertices on view $c$;
    for each projected vertex $q$ do
        Find the point of $I_c(t)$ which is closest to $q$ and maximizes the
        gradient similarity metric with $q$;
        Call $y$ such a point;
        Call $w$ the obtained gradient similarity;
        Define $(w, k, y, c)$ to be a valid correspondence;
    end
end
```

The search for the closest and similar point y in the image $I_c(t)$ is restricted to the line passing through q and parallel to the 2D normal of the current surface estimate computed at point q. This restriction considerably reduces the amount of computational time needed for the correspondences extraction step, maintaining the convergence proprieties of the algorithm. Indeed, each 2D normal of the current surface estimate related to a point q represents the direction of the infinitesimal displacement of q obtained by an infinitesimal variation of the current pose $\widetilde{\theta}$. Finite variations are instead considered by updating the silhouette correspondences during the optimization phase, in our case, every 20 gradient descent iterations.

Outliers are detected by looking inside the set of founded correspondences for the ones which cannot be modeled with a 2D gaussian error model. More precisely, for each bone i and each camera c, the mean and the standard deviation of both the 2D rotation axis and the 2D translation vector of the bone i on camera c are first computed basing on the founded correspondences. Each correspondence on the same camera c which generates a 2D rotation or a 2D translation far away from the estimated mean is labeled as outlier and deleted.

Moreover, when $\widetilde{\theta}$ approaches to the optimal solution, a further outlier detection step is performed deleting all the correspondences that are too much far away from the current pose. In this way, the gaussian error model for the functional (4.58) can be assumed valid and thus, further algorithm iterations generate more accurate results.

At each frame the average number of optical flow correspondences is far less than the number of silhouette ones. Typically there are about one hundred of the

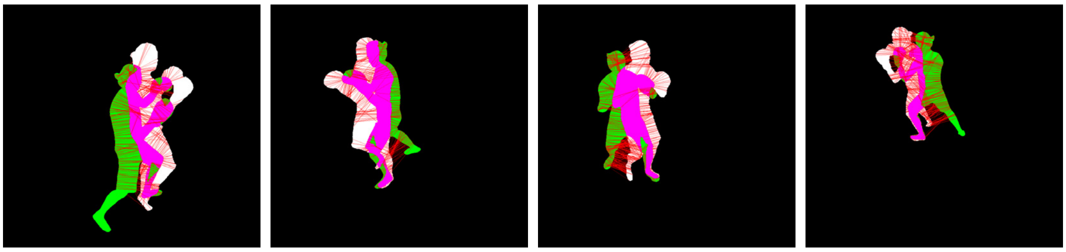


Figure 4.5: *Correspondences detected by our algorithm during the estimate of the pose assumed by the actor in a frame of a tested sequence The silhouettes of current pose estimate (green) are superimposed on the ones extracted from the video streams (white).*

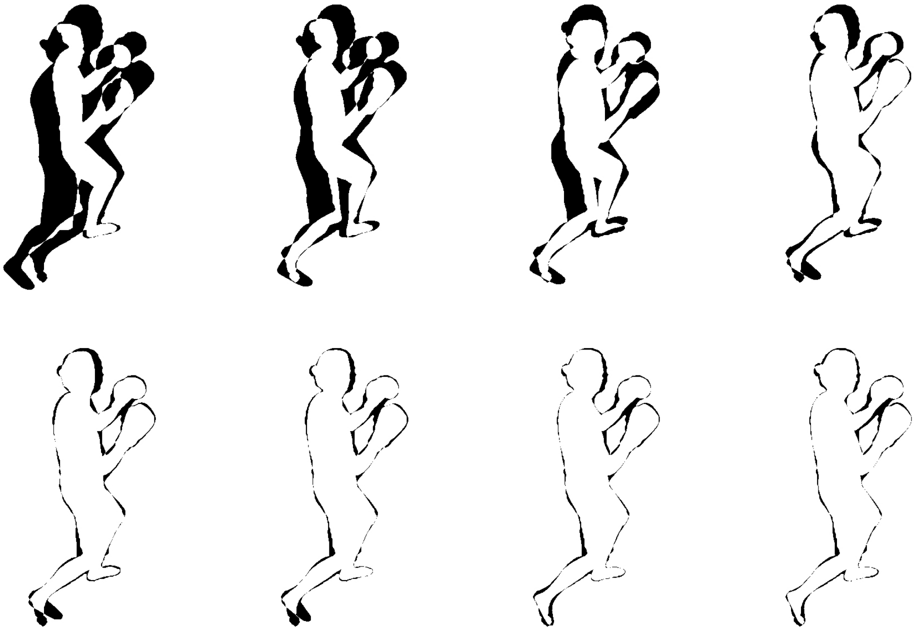


Figure 4.6: *Evolution of the pose estimate during the optimization phase of our algorithm.*

former versus one thousand of the latter. Therefore, in order to account for such proportions, the algorithm weights by a factor of 10 the contribution of optical flow versus that of silhouettes.

The entire correspondence extraction procedure is performed separatively on each video stream paving the way to a parallel version of the algorithm.

Figure 4.5 shows an example of correspondences detected by our algorithm during the estimate of the pose assumed by the actor in a frame of a tested sequence. Figure 4.6, instead, shows the evolution of the pose estimate during the optimization phase. In both cases, the silhouettes of the current pose estimate are superimposed, using a XNOR operator, on the actual silhouettes extracted from the video streams.

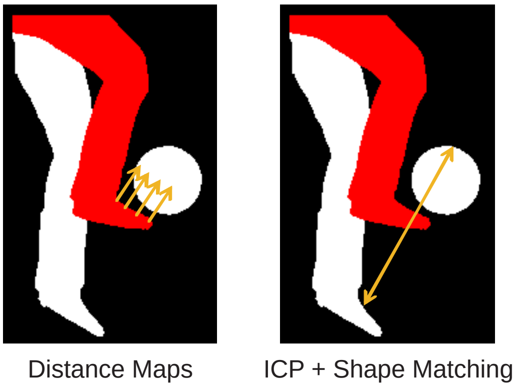
4.4.3 Comparisons

Differently from other approaches using distance maps to account for the silhouette information, like, for instance [6], the “ICP + Shape matching” approach generates less local minima inside the functional $g(\theta)$. Distance maps express only local information without giving the possibility to transmit an edge information over another edge and without giving any sort of edge similarity.

As an example, let’s assume the situation depicted in Fig. 4.7. There is a ball and a leg, the red shape represents the leg silhouette at the current algorithm iteration, while the white one represents the actual silhouette. Using distance maps, Fig. 4.7(left), the generated force field pushes the foot inside the ball causing a local minima. Instead ICP, Fig. 4.7(right), can solve the situation looking backward and forward for the most similar edge avoiding the local minima.

Differently from [111], [73] we do not use any 3D error function avoiding time consuming tasks like visual hull reconstruction or multi view stereo.

Differently from [152], we treat optical flow information as it is without inferring the actual 3D motion flow of the scene. [152], in fact, assumes that inter-cameras correspondences can be established and the optical flow triangulated obtaining an estimate of the actual 3D motion flow. However, finding these type of correspondences is not an easy task, indeed, it needs the use of complex and time consuming descriptors and it introduces further triangulation and correspondence error which is not negligible. Moreover, a lot of optical flow correspondences are discharged because only the ones forming a 3D motion correspondence are considered valid. On the contrary, in our approach, all the 2D correspondences are kept valid and no matching or triangulation tasks are needed.



Distance Maps ICP + Shape Matching

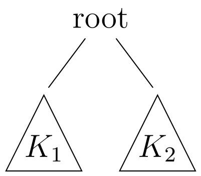
Figure 4.7: *Example situation: comparison of methods dealing with 2D silhouette information.*

The approach proposed by de Aguiar et al. [42], instead, uses a completely different deformation model. This model is complex and computational expensive but it allows to capture movement details all over the surface, also on skirts, pants ant t-shirts. However, the output of this procedure is not a motion capture data since no skeleton is used. The output is a only a time-varying mesh similar to the one captured by a dynamic 3D scanner.

4.4.4 Handling multiple people and objects

Using a property of the LBS controller, multiple deformable objects can be tracked assuming all parts of the same deformable object having as kinematic tree the union of all the kinematic trees related to the original objects.

More formally, given two deformable model with respectively initial shapes γ_1 and γ_2, kinematic trees K_1 and K_2, and skinning matrices α_1 and α_2, a deformable model considering both these two objects can be defined having initial shape γ, kinematic tree K and skinning matrix α defined as follows. γ is obtained by the union of γ_1 and γ_2 in such a way that the first n_1 vertices belong to γ_1 and the subsequent n_2 vertices belong to γ_2. The kinematic tree K is made by the two subtrees K_1 and K_2 both children of the root bone of K as:



(4.66)

Bones are enumerated starting from the root bone followed by the m_1 bones of K_1 and then by the m_2 bones of K_2. Finally, the skinning matrix α has the form

$$\alpha = \left[\begin{array}{c|c} \multicolumn{2}{c}{0,\ldots,0} \\ \hline \alpha_1 & \mathbf{0} \\ \hline \mathbf{0} & \alpha_2 \end{array}\right] \tag{4.67}$$

where the first row, i.e., the one representing the weights related to the root bone, is all zero.

This procedure can be iteratively extended to more than two deformable objects and therefore, the above described algorithm can be used to handle simultaneously multiple deformable objects. Rigid objects instead, like balls and sticks, can be considered as a articulated deformable models with only one bone and alpha matrix equals to an 1-by-n matrix of ones.

Occlusions between objects are automatically handled since they appear, to the algorithm, as self occlusions, i.e., the same object occludes itself. In these cases, the shape matching and outlier detector become essential since silhouettes belonging to different objects have to be distinguished.

Chapter 5

The Making of the Blue-Room

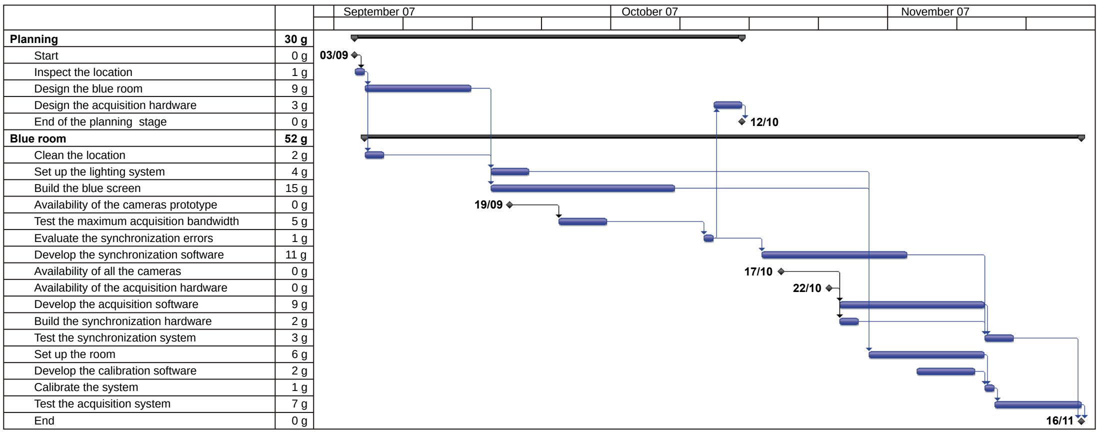
5.1 Introduction

The planning phase for the construction of the acquisition room started on the 3rd September 2007, as stated in the Gantt chart depicted in Figure 5.1. The only one room available for my experiments was located inside an old and small building, called “Casetta del custode” or simply “Casetta”, detached from other buildings of the Department of Information Engineering.

The moment I applied for the use of that room, almost the entire building was used as a store for obsolete hardware like computers, monitor and other broken stuff. At the top of the stairs, on the second floor, was located a small antechamber of about 8 square meters serving as a entryway for an unused bathroom and for a big 24 square meters room that soon would have become the setting for the Blue-Room (Fig. 5.2).

The electrical infrastructure was still working but neither the heating system nor the water one were active. The winter was coming and the prospect to do my experiments in a humid and cold environment didn’t make me so happy but it was the only chance I had. The people who have last worked there, about two years before, have moved in such a hurry that they left a lot of their stuff inside, so a first cleanup was needed before planning the work.

At that time, the Department had clear plans for the future of the entire building: the idea was to completely renovate it and create new offices. Fortunately, the lack of fund and bureaucratic delays had shifted all this plans years after years, leaving the entire building unchanged, as it had been before. However, at that time, nobody had a clear idea about the time when the renovation works would



Planning	**30 g**
Start	0 g
Inspect the location	1 g
Design the blue room	9 g
Design the acquisition hardware	3 g
End of the planning stage	0 g
Blue room	**52 g**
Clean the location	2 g
Set up the lighting system	4 g
Build the blue screen	15 g
Availability of the cameras prototype	0 g
Test the maximum acquisition bandwidth	5 g
Evaluate the synchronization errors	1 g
Develop the synchronization software	11 g
Availability of all the cameras	0 g
Availability of the acquisition hardware	0 g
Develop the acquisition software	9 g
Build the synchronization hardware	2 g
Test the synchronization system	3 g
Set up the room	6 g
Develop the calibration software	2 g
Calibrate the system	1 g
Test the acquisition system	7 g
End	0 g

Figure 5.1: *The Gantt chart for the Blue-Room project.*

have started and so the main risk for my plans was that I would have had to leave the room before I could effectively use it. In the end, the renovation works officially started on the first of March 2008, just fifteen days after I finished to record all the possible footage that, in that moment, I could imagine; about 120 video streams.

During the planning stage of this project, all the working steps were analyzed and simulated on a computer using a ray tracing software in order to decide which were the best and inexpensive solutions for achieving my purposes. The size of the blue screen, the resolution of the cameras, their optics and their positions and lastly the lighting system were decided during this first stage. Since time was a critical variable in the project, the order for all this materials had to be placed quickly taking into account of about 20 working days for delivery. Once the cameras arrived, several off-line tests were performed to ensure the correct synchronization of the streams, to choose the correct fabric to use for the blue screen and to decide the number of computer that had to be involved during the acquisition. The end of these tests declared the effectively end of the project planning stage.

The human resources assigned to this project were myself and another guy, named Marco, that was working on his master thesis about 3DVideo acquisition systems.

The next sections describe the elements and the problematics of a generic multi-camera recording room for motion capture purposes, namely, the blue screen, the lighting and the recording system. Then for each element, the choices made for our particular case, are described.

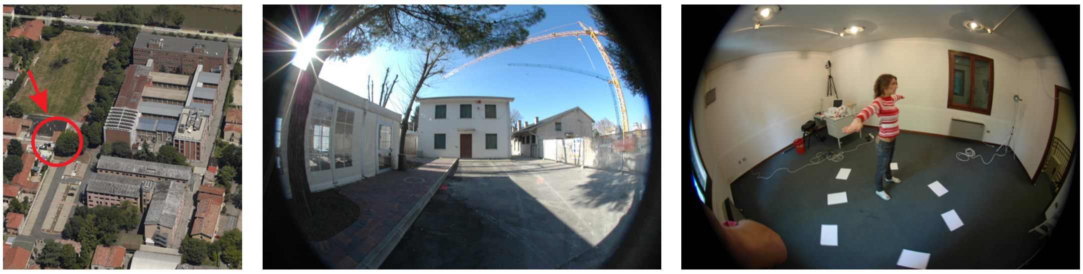


Figure 5.2: *(Left) The location of the "Casetta". (Middle) The outside. (Right) One of the first test picture taken inside the room that soon would have become the set for the Blue-Room.*

5.2 The blue screen

As for the body scanner, the chroma keying technique revealed to be the best solution to adopt in order to achieve a good background subtraction. Indeed, when the shadows can be neglected, the obtained segmentation has one pixel accuracy. The only one drawback is the requirement that all the background, falling inside the cameras fields of view, has to be solid colored (usually blue or green, to contrast the skin color). This condition can be satisfied by the use of a "screen" that can be made of fabric, wood or other sophisticated materials[1] which avoid specular reflections and strong Lambertian proprieties, or in other words, it reflects the light equally in each direction.

In our particular case, the room sized 6.35mx4.45mx2.8m and to cover almost all the cameras field of view, the height of the blue screen had to be at least 1.9 meters from the floor. The choice for the fabric fell onto the IKEA[2] DITTE (Blue), a very inexpensive cloth (about 2.14 euros per square meter) sold with a fixed width of 1.4 meters. The total area to cover was 69.3m^2 and since the number of seams has to be limited, to maintain the color uniformity, the needed cloth was about 75m^2 cut in the way shown by Figure 5.3. The pieces in line (b) were sewed together to make the floor cover while the others (a), (c), (d) and (e) were used to cover the walls. The light blue piece in line (c) instead, was discarded.

In order to stick the fabric onto the walls, a wood frame, all around the room at about 1.9 meters from the floor, had to be built up. The floor screen, instead, made a good friction with the moquette keeping it in place and avoiding

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

[2] http://www.ikea.com/

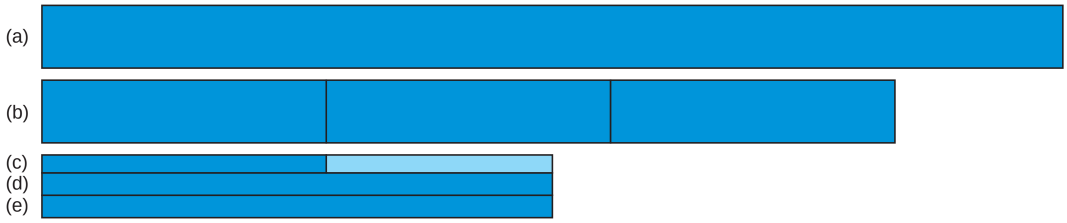


Figure 5.3: *Cut and sew diagram for the blue fabric used as blue screen.*

dangerous slides. However, due to the high humidity, the cotton fibers relaxed during the time and so, during the first twenty days, there had been the needed to tense the cloth up more than one time.

5.3 The lighting system

In order to obtain a good background subtraction, the blue screen has to be sufficiently and uniformly illuminated even if an object in present in the scene. Therefore, shadows and shading effects that could appear onto the screen have to be eliminated as much as possible. Moreover, in order to acquire good quality videos, the foreground people and the objects interacting inside the room have to be correctly illuminated. In other words, high gradients of illumination on foreground objects have to be avoided, because, with normal cameras (non-HDR cameras), this generates images with both overexposed and underexposed regions. This situation cannot be corrected adjusting the camera settings: the only way to do it, is to operate on the illuminants.

The design the lighting system for a blue room is not exactly equal to do it for a standard office room, because, a blue screen absorbs much more light than a normal office furniture, thus, the light reflections are lower. Moreover, if one plans to arrange the illuminants only at the ceiling, as in our case, the illuminance[3] of the wall screens is the first design constraint to take into account. These regions, indeed, shown a minimum of the illuminance function since they are far away to be perpendicular with respect to the incoming light rays.

The light spectrum is also important factor in the lighting system design. A full spectrum light is usually preferred since it avoids strong software white

[3] Illuminance is defined as the quantity of light, or luminous flux, falling on unit area of a surface.

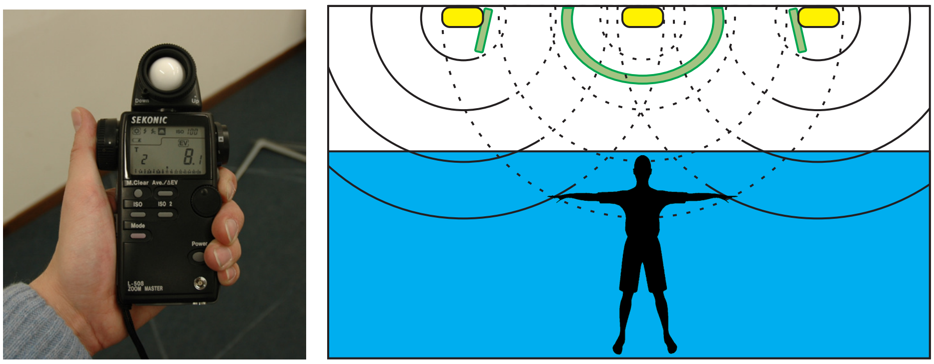


Figure 5.4: *(Left) A commercial light meter. (Right) Diagram of the filters arrangement and luminous flux attenuation.*

balance processes to get the actual colors.

In our case, in order to achieve 600lux on the wall screens we used six 3F Linda luminaires[4] with two 58$watt$ linear fluorescent-lamps[5] each, for a total electrical power of about 700$watt$. Each lamp generates 4000 lumen at the frequency of 100Hz and operated at full spectrum (5500°K). The luminaires were uniformly installed on the ceiling with the long side parallel to the short side of the room. Even if they were simple and inexpensive, they did their job, distributing smoothly the light in such a way to avoid shading effects, like bright and dark lines onto the blue screen.

Once the wall screens were correctly illuminated, the problem moved to the control of the illuminance on the floor and on the foreground objects. More specifically, the aim is to ensure a almost constant illuminance over all the actor surface, for every position and pose that he can be. Moreover, the same has to ensured for the floor surface. Clearly, it is impossible with only ceiling illuminants, but if we consider only the central part of the room, i.e., the part where the actions would effectively take place, then this can be archived by the use of some filters[6]. Indeed, in this way one can control the amount of light in every direction with small incremental steps. The arrangement of these filters cannot be planned on a computer simulation, in practice the right way to do it, is to equip ourselves with a light meter like the one in figure Figure 5.4(Left) and check in every point, the amount of incoming light from a specified direction, i.e., the illuminance.

Figure 5.4(Right) and Figure 5.12 show the filters arrangement inside the

[4] `http://www.3f-filippi.it/`

[5] `http://www.philips.it/`

[6] The simplest way to build up a light filter is to use thin polystyrene sheets.

blue-room. In particular, the central one stops the light coming from the top to the head of the actor while it still frees all the light directed to the wall screens. The lateral filters instead, avoid the overexposure of the upper body parts with respect to the lower body parts of the actor.

5.4 The recording system

A digital video recording system consists of several hardware components namely a digital camera, a transmission medium, a codec system and a storage system. The images captured by the camera are transformed into a digital signal transmitted through the transmission medium to a device which encodes it. The result is then stored in a non-volatile media which is usually an hard drive (see Figure 5.5).

All the hardware involved in the recording process have to be able to deal with the amount of data that they have to process. This is one of the main constraints in the design of a digital video recording system.

The amount of data produced by the camera depends upon the chosen spatial and time resolution and the chosen pixel format. In particular, this latter one describes how the color information is stored in the image, i.e., which color space and chroma subsampling are used. The color is usually represented using a RGB, YUV or a YCbCr space but the pixel itself does not always hold the entire information related to its color. This, is normally spread out to the neighboring pixels in order to reduce the bandwidth of the final stream with a simple and little lossy compression. This process is specified by the chroma subsampling parameter which can be 4:4:4 (no compression), 8:4:4, 4:2:2, 4:2:1, 4:1:1, 4:2:0, 4:1:0, 3:1:1 or specifically for digital cameras RAW BG, RAW GB, RAW RG or RAW GR.

From the previous considerations, B_{video}, the amount of data produced by the

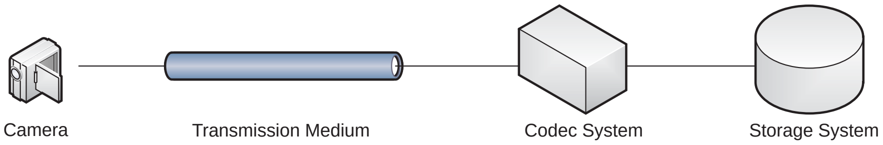


Figure 5.5: *The scheme of a generic digital video recording system.*

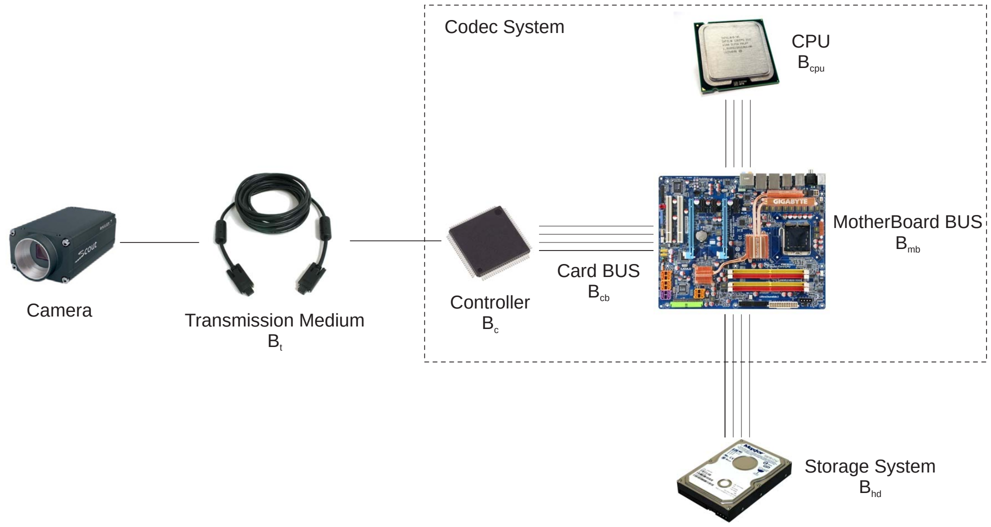


Figure 5.6: *Bandwidth analysis of a digital video recording system with a single camera.*

camera is

$$B_{video} = resX * resY * pixel\_size * fps \tag{5.1}$$

where $resX$, $resY$ and fps represent the spatial and time resolution of the video and $pixel\_size$ is the average size of a pixel. The maximum throughput B_t of the transmission medium instead, has to be greater than B_{video}

$$B_t \geqslant B_{video} \tag{5.2}$$

The codec hardware has to able to read the input stream and compress it at the same rate. This device is usually made up of a computer with a particular data acquisition card. In this case, the controller, which is the main component of the data acquisition card, has to read the input stream and transmit it through the card BUS to the motherboard. This latter component dispatch the stream first to the CPU, which encode it, and then from the CPU to the storage system. Referring to the scheme depicted in Figure 5.6, the bandwidth constrains are:

$$B_c \geqslant B_{video} \tag{5.3}$$
$$B_{cb} \geqslant B_{video} \tag{5.4}$$
$$B_{mb} \geqslant (1+\alpha)\,B_{video} \tag{5.5}$$
$$B_{cpu} \geqslant B_{video} \tag{5.6}$$
$$B_{hd} \geqslant \alpha B_{video} \tag{5.7}$$

where α is the compression ratio obtained by the codec hardware.

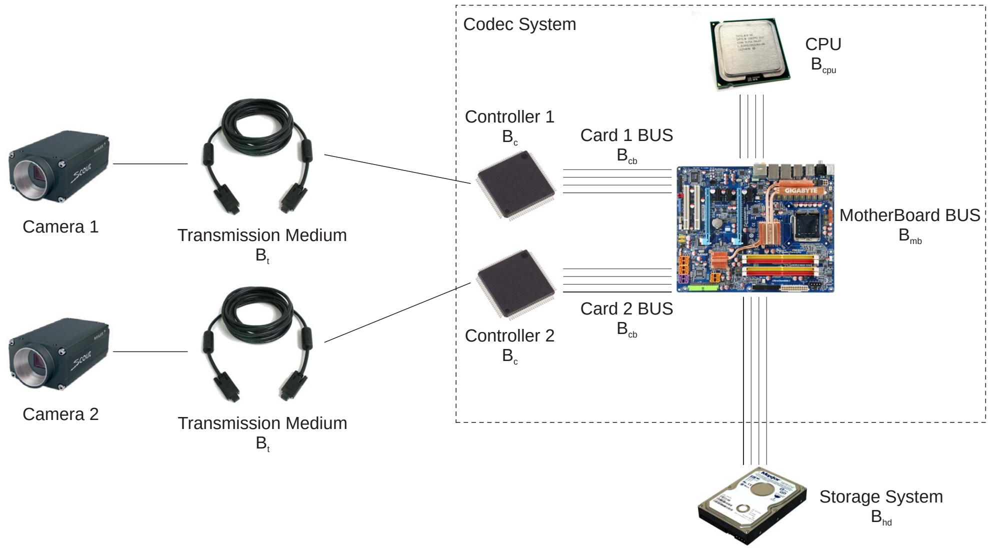


Figure 5.7: *Digital video recording system with two cameras and a single computer.*

In case of more than one camera, more controllers and separated card buses are needed, in order to avoid bottlenecks. Figure 5.7 describes the case of two cameras. However, the constraints regarding the codec system still increase linearly with the number of camera n

$$B_{mb} \geqslant n * (1+\alpha)\, B_{video} \tag{5.8}$$

$$B_{cpu} \geqslant n * B_{video} \tag{5.9}$$

$$B_{hd} \geqslant n * \alpha B_{video} \tag{5.10}$$

In fact, with the current technology and high video bandwidth, the solution with n controllers turns out to be very expensive. Thus, a solution with n separated computers, each one dealing with only one camera, is preferred.

The choice for the cameras model fell onto the Basler[7] Scout scA1000-30fc, a 30 fps firewire-b camera based on a 1/3″ Sony color CCD of resolution 1034x779 pixels. The chosen pixel format was the YUV 4:2:2 and the images were acquire at 21 fps for a total bandwidth of about 32 MB/s. The transmission medium was a 3 meters long shielded firewire-b cable (IEEE1394b 9-9pin) with a maximum transfer rate of about 750 Mbit/s (90 MB/s) in full-duplex. The used acquisition cards were four commercial 1394b to 32bit PCIe-1x host adapter interfaces each with a single 54 MB/s controller. Note that the maximum throughput for the

[7] `http://www.baslerweb.com/`

PCIe bus is 250 MB/s. For this specific configuration, also the size of the data packets has to be considered to ensure a safe transmission of all the frames from the camera to the controller. Big packets increase the cable throughput since no redundant header data is transmitted. However, if a packet is lost or corrupted, the retransmission time increases linearly with the packet size. Fortunately, the packet loss rate of a firewire cable is very low and thus, the best solution is use packets of the maximum allowed size, which is usually much smaller than an entire image data.

Four computers, one for each camera, were installed. Due to the limitation on the firewire cable length (10 meters cable is a very expensive technology), two of them had to be placed near the cameras, in the center of the room, just above the blue screen (as in Fig. 5.12). The other two computers were installed outside the room, in the antechamber, where an operator can easily control them. The available computers were four Dell Optiplex, each based on a 2.2GHz Intel Core2 Duo processor with 2GB of DDR2 RAM and a 250MB SATA/300 hard drive running at 7200 rpm. All the computers were linked together by a 100BASE-T LAN and a central hub (see Fig. 5.8(Right)). An ad hoc software was developed to control the recording state of each camera from the one of computers placed in the control room. This software is also, in part, responsible for the stream synchronization as it will be described in the subsection 5.4.3.

5.4.1 The design of the optics

In the design of a multi-camera video recording system for monitoring a generic area, two important aspects have to be taken into account, namely the optical properties and the arrangement of the cameras. Thus, field of views, resolutions, CCD size, distortions as well as the points of view have to be fixed in a planning stage considering all the requirements.

Specifically for a motion capture system, once the area to be monitored is defined, one has to ensure that each camera sees each point of this area. This is not a mandatory requirement but it is suggested for a good quality tracking. Computer simulations and on-site measurements help to achieve this step. The result is a set of constraints on the cameras positions and their optical proprieties.

Moreover, for a good motion tracking, the cameras have to be arranged all around the area to monitor in a non-symmetrical way. This is due to the fact that silhouette information extracted from a couple of symmetrical cameras, is redundant. More precisely, the worst case happens when the optical axes, belonging to

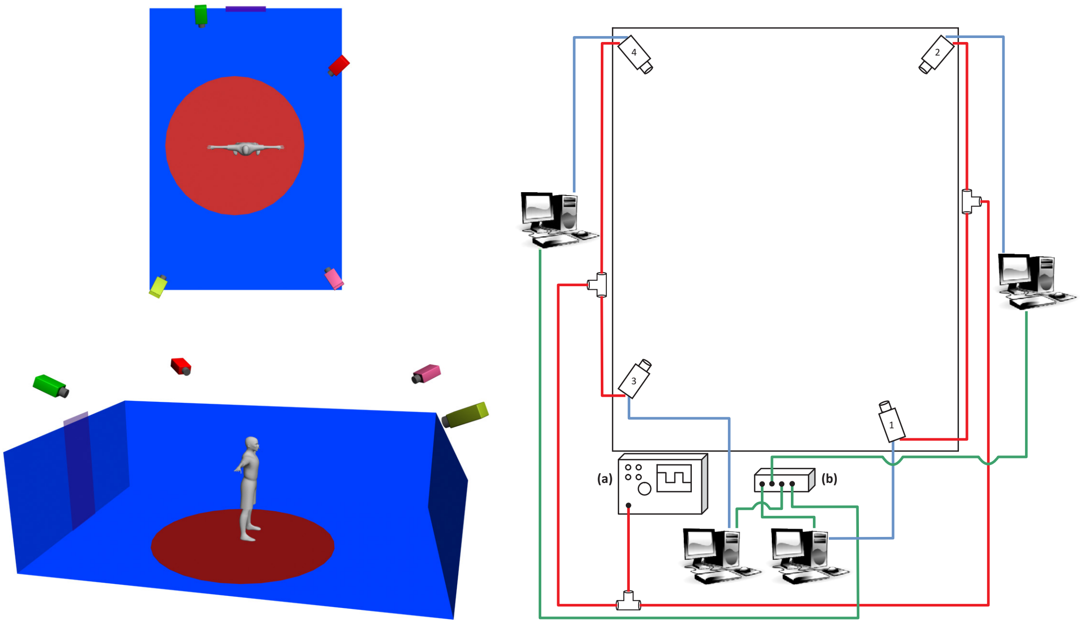


Figure 5.8: *(Left Top/Bottom) Arrangement of the cameras inside the blue-room. (Right) Schematic of the blue-room. Green, red and blue lines represent respectively LAN cables, synchronization cables and firewire cables. (a) represents the synchronization trigger while (b) represents the LAN hub.*

these two cameras, are parallel to the segment connecting the respective centers of projection and the object to analyze is placed in the center of this segment. Indeed, if the object is small enough with respect to the size of this segment, the silhouettes obtained from these two cameras are exactly the same[8].

The optical proprieties are strongly correlated to the arrangement of the cameras, since both determine the fields of view of the images. For instance, a camera can be placed close to the subject to record, but in this case, the focal length has to be short enough to maintain the subject inside its field of view. Unfortunately, short focal length, in commercial lenses, means also high image distortions, and these not always can be easily modeled by a calibration software, due to the elevate number of distortion coefficients.

A description on how to estimate the optical parameters for a scene can be found in Appendix A. The chosen camera arrangement for our blue room is depicted in Figure 5.8(Left). Camera 2, 3 and 4 mount a 4.5mm lens while camera 1 mount a 3.5mm lens. Each camera is also equipped with a lens hood

[8] This is, for instance, the case of the four cameras HumanEva dataset [141].

in order to prevent glare and lens flare.

5.4.2 The camera settings

In a final stage, once almost everything is ready, the camera settings can be adjusted in order to maximize the video quality. In other words, the optimal values for the aperture, the in-focus volume, the exposure time, the white balance, the sensor gain and the brightness of each camera, have to be found.

Although the concept of video quality does not have a mathematical formulation, some formal considerations can be made: the subject must fall inside the in-focus volume, overexposed as well as underexposed regions have to be avoided, colors must be similar to the actual ones and the histogram of the image have to cover all the possible intensities values (i.e., the image have to have a good contrast). Moreover, frame contrast and brightness must be kept constant during the time and motion blur artifacts have to be avoided. Normally, these conditions cannot be satisfy for the whole image, however, it is very important to respect them for at least the regions of interest.

The pixel intensity of an image is related to the camera settings by these formulas

$$\begin{aligned} pixel\_intensity &= brightness + gain * (signal + noise) \\ signal &= f\left(aperture\right) * g\left(exposure\_time\right) * pixel\_illuminance \end{aligned}$$

where both $f(\cdot)$ and $g(\cdot)$ are monotonically increasing function, *pixel_illuminance* [*lux*] represents illuminance received by the CCD cell and *signal* [*volt*] is the electrical quantity used to store internally the pixel intensity information. This latter one is always subjected to a thermal *noise* that increases with the working temperature.

In order to maintain a constant brightness and contrast of the images the exposure time has to be set as a multiple of the working frequency of the lighting system (in our case 100Hz). Moreover, this value has to be kept small to avoid motion blur artifacts. This last condition is in contrast with the fact that the exposure time determines the entire brightness of the image because in this way, it cannot be set below a certain threshold.

To avoid this problem, a solution could be the increase of the aperture size with the drawback of a smaller in-focus volume which is bounded to contain the central area of the room. Another solution could be the increase of the gain and the

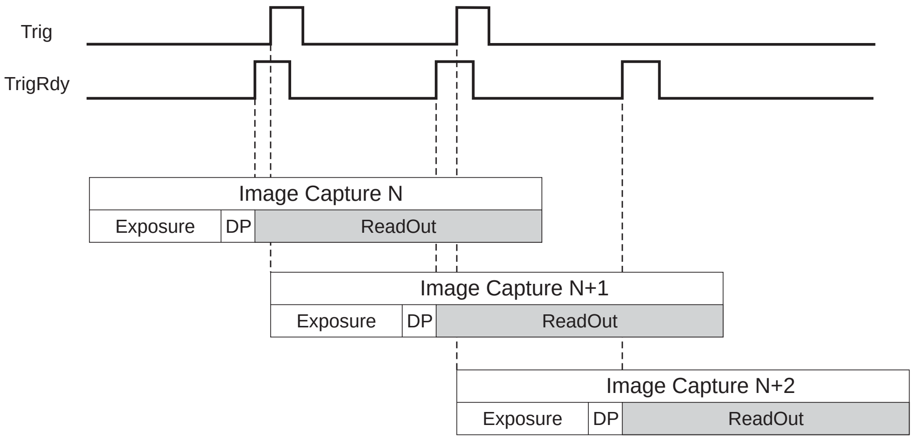


Figure 5.9: *Overlapped Readout and Exposure.*

brightness factors with the drawbacks that, in the first case, the sensor noise increases while in the second one, the image contrast decreases.

5.4.3 The time synchronization

A correct multi-camera recording system have to ensure that all the recorded images are synchronized and taken at fixed time intervals. Therefore for each time slot (fixed equal to $1/fps$, where fps is the number of frames per second captured by the system) all the cameras have to shoot all, at the same time.

The most common solution to the synchronization problem, is the use of an external centralized trigger which gives, to every cameras, the shoot command through a bus.

Most of the professional cameras have the possibility to be triggered either by the data-bus (the Firewire or the USB bus) or by a dedicated channel. The former case is also called software triggering while the latter one, hardware or external triggering. Once the trigger signal is recognized, the shutter is opened and the image acquired. After a time equal to the exposure time plus a data packing time, the camera is ready to accept another trigger signal. This event is often signaled by a trigger ready state sent to the synchronization device.

Even if the camera is ready to acquire another frame, the previous one still inside its memory for as long as it takes to transmit it on the other side of the data-bus. This process is called *readout* and it is often done in parallel with the subsequent frame acquisition (this situation is called *overlapped acquisition* and it is represented in Figure 5.9).

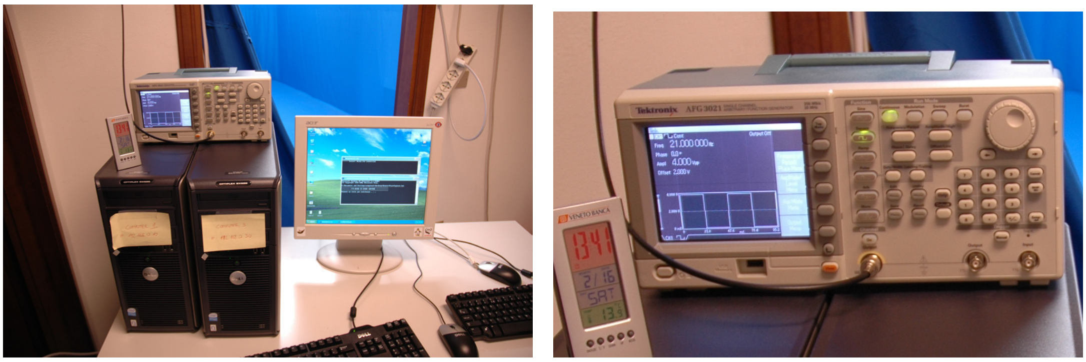


Figure 5.10: *(Left) Control room setup. (Right) Digital function generator used as trigger for the synchronization system.*

Figure 5.8(Right) shows schematically the used synchronization system. Each camera is connected to the trigger by a 50Ω coaxial cable ending up with a 12-pin Hirose connector. Even if the distances between the trigger and the cameras were not always the same, the length of each cable was kept constant in order to ensure a constant communication delay. A digital function generator (Fig. 5.10) were used as a trigger and was placed in the antechamber together with the other two computers to give the possibility to the operator to start the recording from there.

5.4.4 The space calibration

Space calibration can be done in several way. We chose to use a checkerboard based calibration with a modified fully-automatic version of the Matlab Camera Calibration Toolbox [18] extended to work with multi-camera systems.

In the design of a checkerboard, parameters like the number of the squares, their size as well as the material which are made, have to be chosen considering the environment to calibrate. The best accuracy can be obtained using a glass checkerboard since this material maintains its geometrical properties for a long time. Paper based checkerboards instead, are sensitive to humidity which deforms themselves in a non-isotropic way. These last checkerboards can be only used for few days after they were printed.

The numbers of the squares, in a checkerboard, have to be odd in one side and even in the other, so that a vision system can automatically recognize its orientation. The number of corners coincides with the number of information

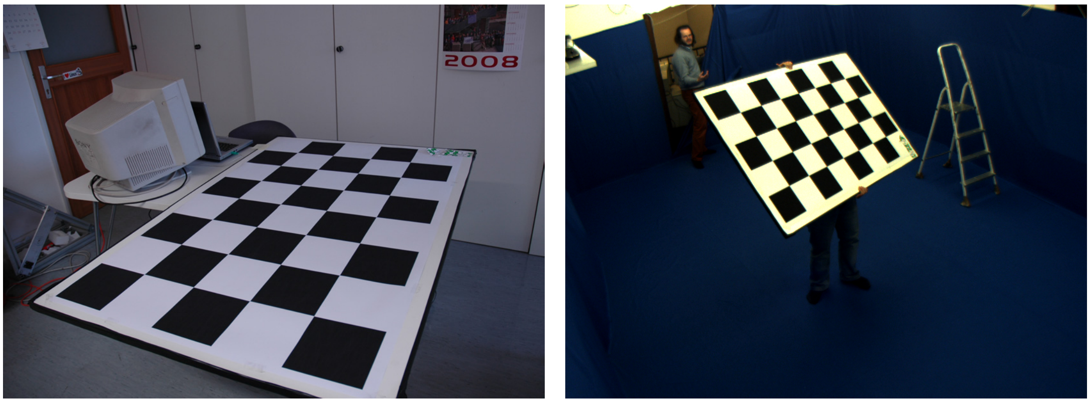


Figure 5.11: *(Left) Checkerboard used in the calibration stage. (Right) A calibration photo taken at low brightness to increase the corners detection accuracy.*

that the calibration software uses to calibrate a view, thus it has to be high. Moreover, the corner detectability increases with the size of the squares.

In our case, we built up a $0.8mx1.4m$ checkerboard with $5x8$ squares printed on a high quality paper (see Fig. 5.11(Left)). The support was made of thick wood to avoid the bends (about 20mm).

Clearly, the camera settings chosen to achieve good quality images in a recording stage are not, in general, valid to achieve good quality images of a checkerboard. Moreover, in the calibration case, the concept of good quality also changes. Indeed, low brightness images are preferred, since the white squares of the checkerboard, which are very sensitive to overexposure, cause blurring effects that increase the corners detection error (see Fig. 5.11(Right)).

To record low brightness images, the best technique is to lower the exposure time, since it is the only one hardware parameter which modifies the brightness without altering the geometric proprieties of the camera, i.e., the calibration results. Then the electrical parameters, i.e., the brightness and the gain, have to be adjusted to ensure a hight contrast image in the proximity of the corners. Note that, in this case, the exposure time can assume values that are not multiple of 100Hz, since the brightness coherence between frame is not required.

Photos taken with the checkerboard close to the camera and perpendicular to the viewing rays are useful to calibrate the camera distortions since they cover at the same time all its field of view. On the other hands, photos taken with high inclined checkerboard, in both directions, are useful for the estimate of the camera focal lengths.

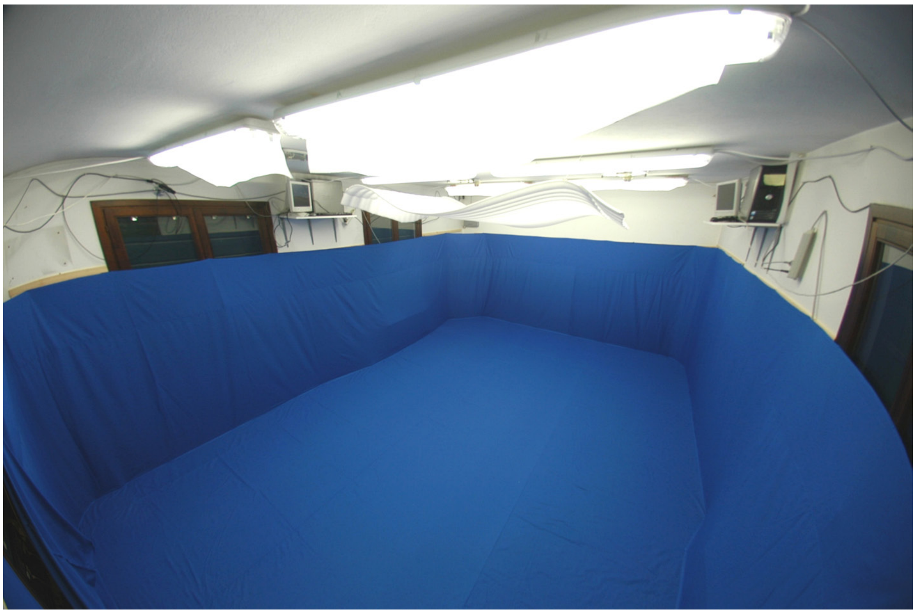


Figure 5.12: *A picture of the Blue-Room.*

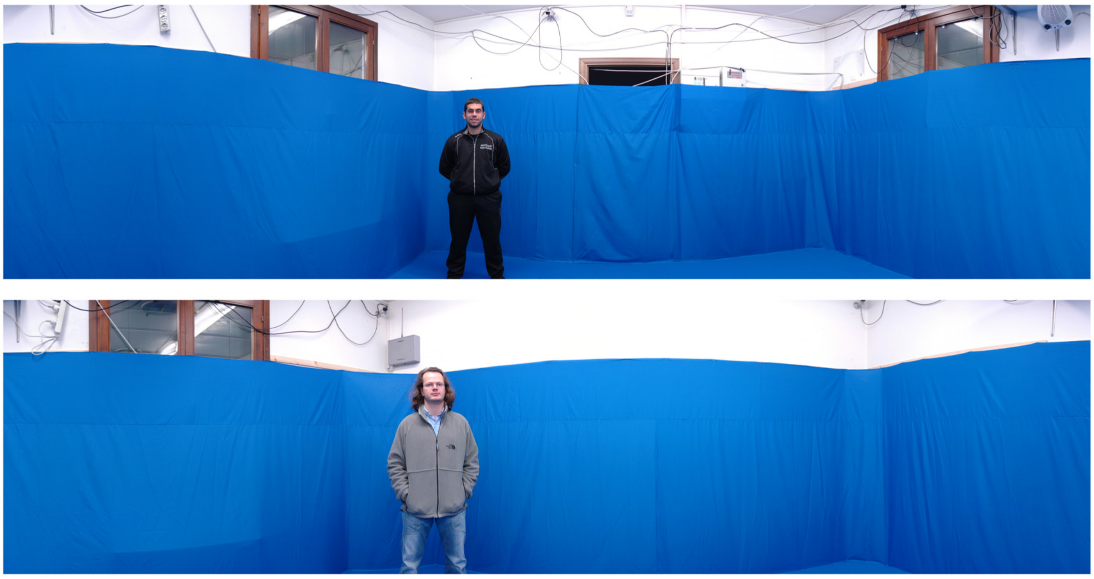


Figure 5.13: *A panoramic view of the Blue-Room cut in two slices.*

Chapter 6

System Evaluation

This chapter describes the experiments made to evaluate the entire system and discusses the obtained results. The body scanner results are analyzed in the first section, while, the results obtained with the motion capture system are described in the following sections.

6.1 Body scanner

The body scanner was tested on seven actors and three objects. The obtained meshes count, typically, more than 500 thousands faces, however, the related textures were reconstructed only for their simplified versions, namely counting only 13 thousand faces. This downsampling was necessary to limit the used computer resources, especially the RAM.

Texture resolution was limited to 21 MPixels (6000x3500 pixels). This can be considered the maximum resolution that can be obtained from 60 images at 6.1 MPixels each, where the person covers, in average, the 8.6% of the each image. Over this limit, the quantity of information carried by the texture remains the same, while, the added pixels are due to only an interpolation, implicit in the texture reconstruction procedure.

As expected, the adopted technique was able to generate, textures free from the illumination artifacts, ghostings and blurs. The possibility to recover this kind of information is a peculiarity of all the passive scanning systems differently from their active counterparts. Indeed, these latter ones require working conditions

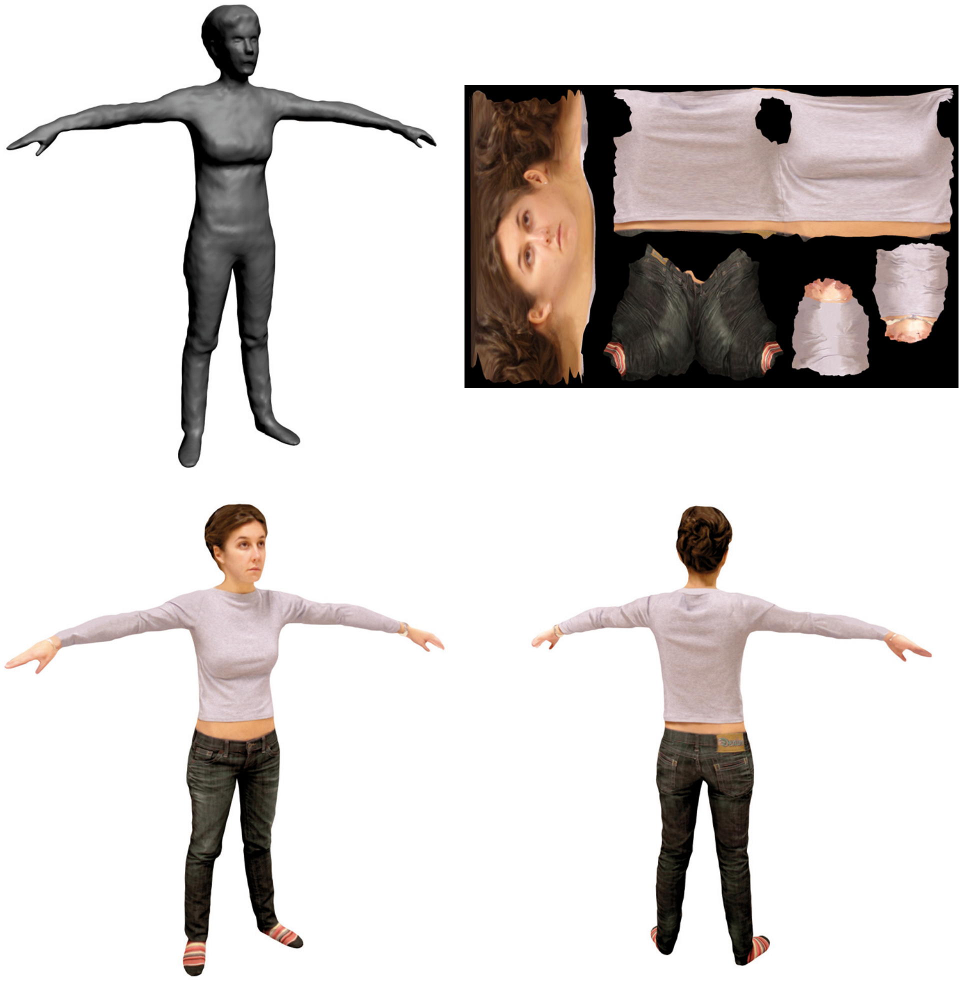


Figure 6.1: *Typical reconstruction obtainable with our body scanner. (Top Left) Rough mesh model counting* 902 *thousand faces. (Top Right) Related texture at* 21 *MPixels. (Bottom Left and Right) Low resolution textured model.*

which are not suitable for the acquisition of the texture, like, for instance, the low illumination.

Figure 6.1 shows the reconstruction of an actor with his related texture. As the reader can see, in the top left of this figure, our system have generated a good quality mesh without holes and spurious detached surfaces. This is mainly due to the use of the deformable models which imposes the just observed mesh characteristics. The quality of the mesh can be evaluated numerically using the parameters Q_{equ} and Q_{plan} defined in [53]. For the particular example of Fig. 6.1, their values are 0.81 for the Q_{equ} and 0.998 for Q_{plan}.

Concerning the geometry, the achieved reconstruction error is difficult to eval-

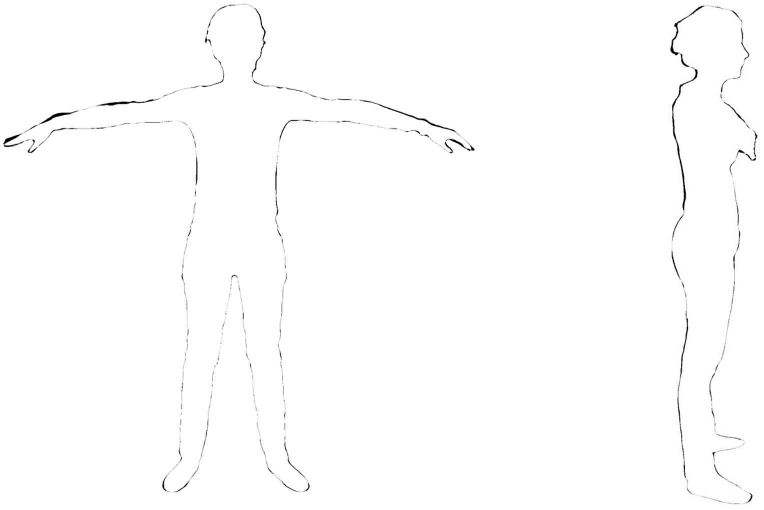


Figure 6.2: *Comparison between the silhouettes of a reconstructed actor and the silhouettes of the real one. Black pixels mean disagreement between the two.*

uate directly, because no ground truth, with the same characteristics of the human body, is available. Instead, an indirect measurement of this error is possible and can be performed by estimating the errors introduced by all the factors contributing for it. The reconstruction error is then, a combination of these factors.

The reconstruction error, in our system, is determined by three factors namely, the plays of the camera positioning system, i.e., the calibration errors, the actor's movements during the acquisition and the errors introduced by the passive reconstruction pipeline.

As mentioned in Chapter 3, our system achieves an angular error in repositioning the camera smaller than 0.071°. This value can be easily neglected, because the corresponding reprojection error on the surface of the actor, is less than 0.25 pixels.

Concerning the second factor, in spite of all the precautions undertaken to prevent movements of the actor during an acquisition session, these last ones, even in small parts, are still present, and cannot be neglected during the estimation of the reconstruction error. The actor's movements, in fact, generate conflicting stereo results, which, when fused together, lead to a global reduction of the final reconstruction details, i.e., to blurred surfaces. Concerning the silhouette information instead, these movements typically reduce the size of the limbs because the fusion process tends to satisfy all the extracted silhouettes, also the displaced ones. Figure 6.2 compares the silhouettes of a reconstructed actor with the silhouettes of the real one. In this case, the actor's movements are visible on his back and on his left arm. The other discrepancies instead, are due to errors in the reconstruction pipeline. As said in Chapter 3, the maximum movement of the actor is limited, by our expedients, to 5 pixels, which corresponds to an error smaller than $0.5cm$.

The error introduced by the reconstruction pipeline is estimated running these algorithms on the synthetic images generated by rendering a known geometry. This geometry provides the ground truth that can be compared with the output of the reconstruction. The differences between these two models are evaluated using both a volume discrepancy measure and a surface discrepancy measure.

The former is defined as follows. Let's denote with α the surface of the ground truth and with β the reconstructed one. Moreover, let's denote with A and B the two solids contained respectively inside the two surfaces α and β. The volume discrepancy $\varepsilon_V(A, B)$ is defined as follows

$$\varepsilon_V(A, B) = 2\frac{Vol\left((A\backslash B) \cup (B\backslash A)\right)}{Vol(A) + Vol(B)} \tag{6.1}$$

where $Vol(\cdot)$ is the operator computing the volume of a solid. In practice, the term $\varepsilon_V(A, B)$ indicates the percentage of volume in disagreement between A and B with respect to the average volume of these two solids.

The surface error $\varepsilon_S(\alpha, \beta)$ is, instead, defined as follows

$$\varepsilon_S(\alpha, \beta) = \frac{\int_\alpha d(P, \beta)\, ds}{2\int_\alpha ds} + \frac{\int_\beta d(P, \alpha)\, ds}{2\int_\beta ds} \tag{6.2}$$

where $d(P, \cdot)$ is the distance between the surface $\cdot$ and the point P. Roughly speaking, this term measures the average distance between the two surfaces.

Experiments showed that the volume discrepancy error is always less than the 0.6% and the average distance between the actual and the reconstructed surface is around $4.2mm$, value corresponding to the size of a pixel. The standard deviation is $3.2mm$, corresponding to three quarters of a pixels. This means that, the achieved reprojection error is, in average, one pixel with a standard deviation of 0.75.

In conclusion, we can state that, the entire system achieves a reconstruction error which, in the best case, has an average of $4.2mm$ and a standard deviation of $3.2mm$. This error inevitably increases in regions where the actor's movements, during the acquisition, become visible, i.e., where their size exceed one pixel. However, since the adopted precautions, these movements are limited to a maximum displacement of 5 pixels, leading to a final error always below the $2.1cm$.

Figure 6.3: *Computational resources used to process the acquired video streams. The most powerful used computer consists in a single core* 3.4*Ghz Intel P4 with* 2*Gb of RAM.*

6.2 Motion capture

The motion capture system was tested on about 120 video sequences. Since this large amount of data, the required computational power was considerable. Videos were acquired during the day while, their processings were performed by five computers during the night (see Fig. 6.3).

The acquired sequences cover many different types of motions and they are classified in three main groups namely, single person sequences, single person and multiple objects interactions sequences and multiple people and multiple objects interactions sequences.

The first group concerns scenes with only a single person performing motions like walks, jumps, break-dances, pirouettes, somersaults, hand stands, stretching, press-ups, sit-ups and lone kick-boxing. The second group, instead, concerns the interactions between a person and some objects present in the scene. In this group, the acquired motions consist in sword swings, tennis forehand and backhand shots, volleyball serves and bumps, golf shots, baseball hits, soccer kicks, and soccer juggles. At last, the third group contains sequences where more than a single person interact in the same scene either with other people or with objects. Concerning this group, we recorded sequences like two people walks, handshakes, boxing and soccer passes.

The algorithm performance was evaluated both qualitatively and quantitatively. The former evaluation consisted in a visual comparison between each reconstructed frame and the original one. The latter evaluation, instead, was based

	frames	mean [%]	st. dev. [%]
walk	390	8.84	0.93
break-dance	170	12.43	3.81
pirouette & jump	490	11.74	2.24
somersault	170	11.05	4.8
hand stand	200	12.03	3.9
press up	280	11.34	1.91
synthetic sequence	120	5.80	2.50
lone kick-boxing	790	11.84	1.75
soccer kicks	530	9.53	1.52
soccer juggles	590	11.32	1.70
golf shots	560	11.08	1.94
walk & handshake	190	10.77	1.32
boxing	300	9.32	1.25
soccer passes (1)	570	10.03	0.96
soccer passes (2)	860	9.83	1.17

Table 6.1: *Pixel discrepancy error statistics of some of the tested sequences.*

on the *Pixel Discrepancy Error* (PDE) computed as follows. The silhouettes, either obtained by segmenting the video streams or by rendering the reconstructed model, were represented as binary images, with the convention that the background pixels are white (1) and object pixels are black (0). For every frame, a XNOR operator was applied between the extracted silhouette and the silhouette of the reconstructed model. The percentage of black pixels, with respect to the total number of pixels forming the extracted silhouette, represents the PDE.

Table 6.1 reports the PDE statistics computed over the whole reconstruction, for some of the tested sequences. It is worth pointing out that, by definition, the PDE may be due either to an actual pose estimation error or to background subtraction errors and mismatches between the actual actor's shape and the used reconstruction acquired by the body scanner. Therefore, the values in Table 6.1 generally overestimate the actual pose estimation error especially in sequences like the somersault and the pirouette where, fast movements and ground interactions, make the silhouette extraction process rather critical. Furthermore, let's note that our actors wear casual clothes and two of them have long hair. This increases the mismatches between the actual actor's shape and the used reconstruction, indeed, for instance, it is very difficult to tie their hair in the same way as during the 3D scanning process. All the above factors contribute to an increase the PDE but, this increase is not related to an actual increment of the pose estimation error.

In order to exemplify the visual meaning of the PDE, let's consider the case

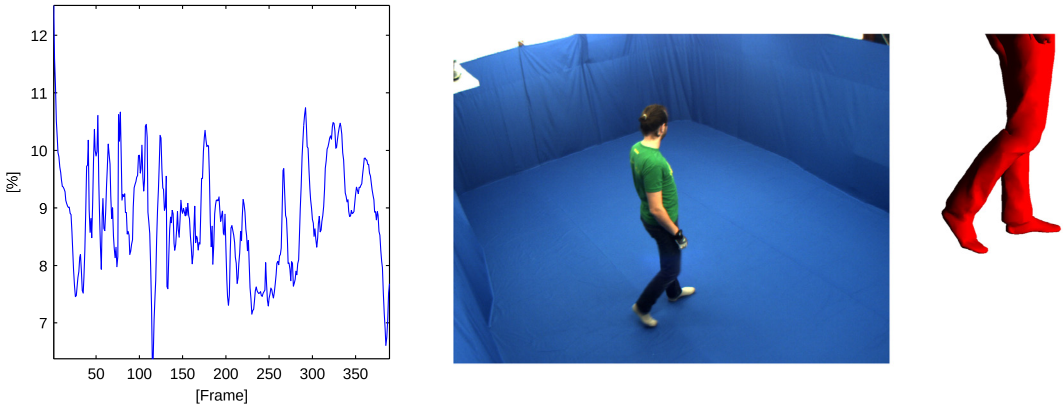


Figure 6.4: *Walk sequence. (Left) Pixel discrepancy error evaluated in each frame of the sequence. (Middle) A frame of the sequence. (Right) Detail of the reconstructed model.*

of frame 63 of the somersault sequence shown in Fig. 6.6 (row 2, column 2). This frame has a PDE of 10.2% which, in spite of its considerable value, has a visual impact rather contained.

Concerning simple sequences like, for instance, the walk (390 frames), the algorithm performance is rather accurate both in terms of PDE and in terms of visual quality. Indeed, as shown in Figure 6.4(Left), the PDE is rather low with an average of 8.84%. In the same figure it is also depicted, a reconstructed frame of the walk sequence where also small details, such as the right foot articulation, are accurately reproduced.

The sequences with fast movements pose a twofold problem. In this case, in fact, the implicit assumption, of the tracking algorithm, that $\theta(t-1)$ is a good starting point for the minimization, does not hold anymore. Moreover, fast movements cause motion blur which, consequently, deteriorates the quality of silhouette information typically altering the real object size. These artifacts are clearly shown in the top row of Figure 6.5 where the actor leg, during an hand stand, becomes smaller and some of its parts disappear. However, optical flow information, in these situations, effectively overcomes such problems since the visual quality of the reconstruction remains remarkably good, as shown in the bottom row of Figure 6.5.

The reconstruction power of the proposed method can be appreciated by the somersault sequence which combines fast movements, spine bends, clavicles rotations and a lot of self occlusions given by the fact that actor bends himself on

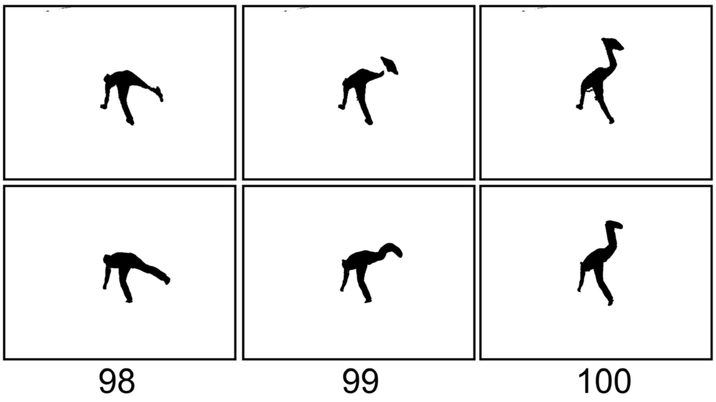


Figure 6.5: *Hand stand sequence. (Top row) Extracted silhouettes. (Bottom row) Silhouettes of the estimated 3D model.*

the ground.

Figure 6.6 shows some frames of such a sequence. The first rows represents the original video recorded by one of the cameras of the acquisition system. The third row shows the reconstructed actor while the second one shows a graphical representation of the PDE. More precisely, the black pixels represent disagreement between the two silhouettes while, the agreement is represented by both the white and the light grey pixels. In particular, a light grey pixel means that the reconstructed and the observed data represent both an object point while, a white pixel that they both represent a background point. At last, the forth row shows the skeletal structure.

It is worth noting that, thanks to the LBS model, the back of the actor is perfectly tracked. Indeed, the reconstructed silhouette lies exactly on the real one. Furthermore, this sequence could not be successfully processed without the use of the optical flow information because, the extracted silhouettes, alone, are not able to supply any information about body parts which do not belong to any silhouette of any point of view. Optical flow, in fact, overcomes the intrinsic limitation of the silhouettes, providing the missing information.

Finally, let's note that, in this sequence, hands are not correctly tracked, e.g. see frames 83, 106, 115 and 126. This is simply due to the fact that the used human model does not have an accurate description of the hands' shape and its skeleton does not contain finger bones. Therefore, it cannot model widely opened hands such the ones in the previously enumerated frames.

The current implementation of the algorithm is not time optimized. Its running time for processing a single frame depends on the actor's movement speed,

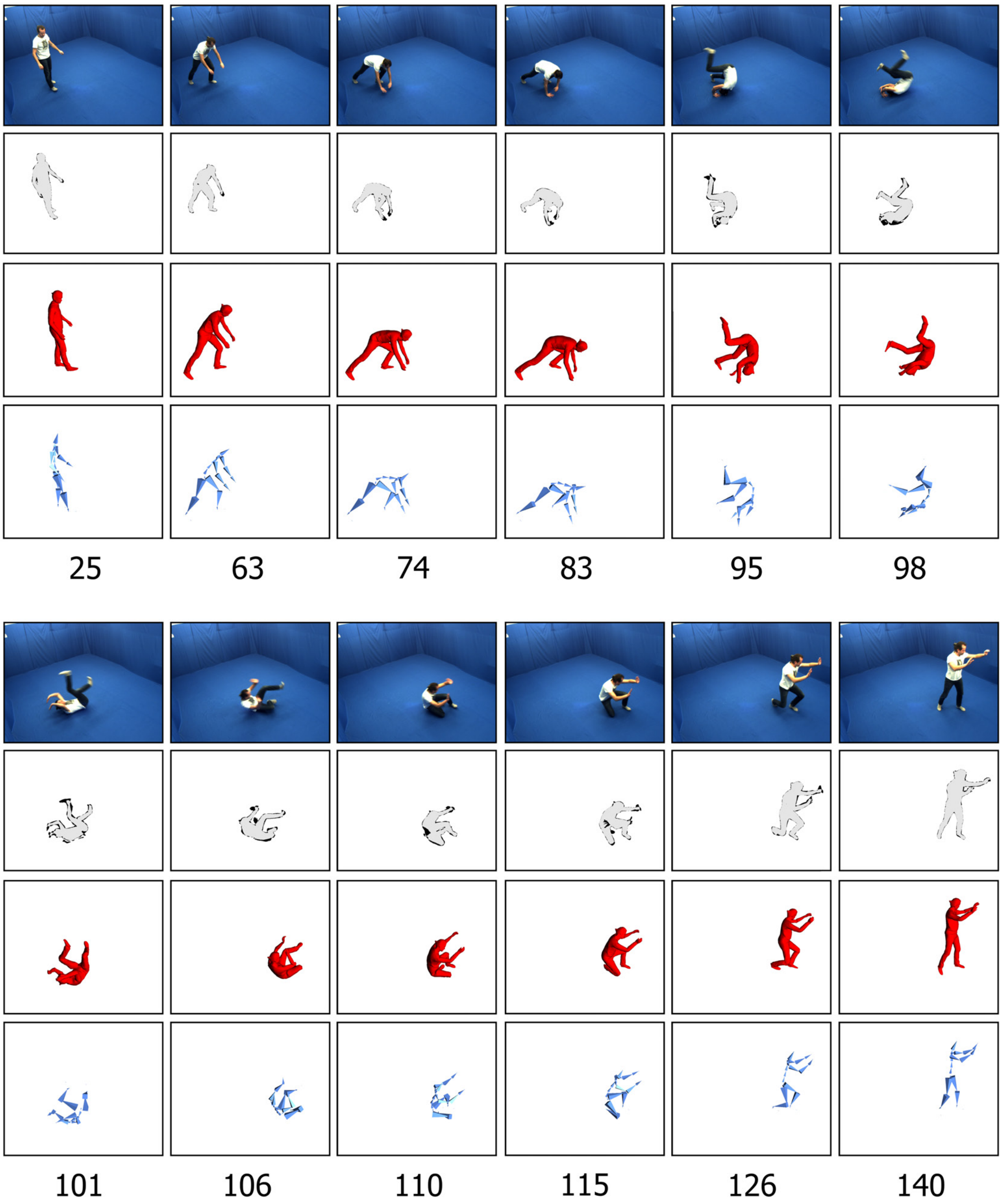


Figure 6.6: *Some frames of the somersault sequence. From top to bottom: real images acquired from one of the cameras of the acquisition system, PDE, reconstructed model and the reconstructed skeleton.*

i.e., how far the actual solution is from its initial guess. Typical running time on a single core 3.4Ghz Intel P4 with 2Gb of RAM is at most 20 seconds per frame.

6.2.1 Multiple entities results

This section describes the results obtained testing the sequences of the second and the third group. As mentioned before, these include both human to object interactions and human to human interactions. We started with simple sequences having only 52 degrees of freedom, more precisely, consisting in a person interacting with a single rigid object, and subsequently, we tested several sequences with more than 80 degrees of freedom. In particular, the most complex one counts 92 degrees of freedom and consists in two people and two rigid objects interacting in the same scene.

In spite of the few number of cameras and the high number of degrees of freedom, our algorithm behaves pretty well detecting the ambiguities that the multitude of occlusions arises. In the next paragraphs, the most relevant sequences are presented and discussed one by one.

In the single person soccer sequences, the actor performs some juggles and some kicks with the ball. One of these sequences is shown in Figure 6.7. Odd rows represent the original frames of the sequence, while even rows, the reconstructed ones.

In this case, the shape matching metric reveals itself as indispensable for an accurate tracking of both the ball and the actor's feet during a kick. Indeed, without this metric, the tracking procedure could fall into a local minima which often consists in a penetration of the ball inside one of the actor's feet.

Figure 6.8 depicts this situation. The first row shows the evolution of the algorithm during a pose estimation without the use of the shape matching metric. A red binary image, representing the silhouette of current model estimate, is superimposed, using a XOR operator, to a white one representing the actual silhouette extracted from the video stream. Red lines represent the silhouette correspondences. At each algorithm iteration the ball is attracted to the actor's foot by a lot of silhouette correspondences which do not take into account for the local shape of their silhouettes. The second row, instead, shows the evolution of the algorithm estimating the same pose but, this time, using the shape matching metric. As the reader can see, most of correspondences pulling the ball inside the actor's foot are discarded and, instead, the correspondences puling the ball to the right place becomes stronger and stronger.

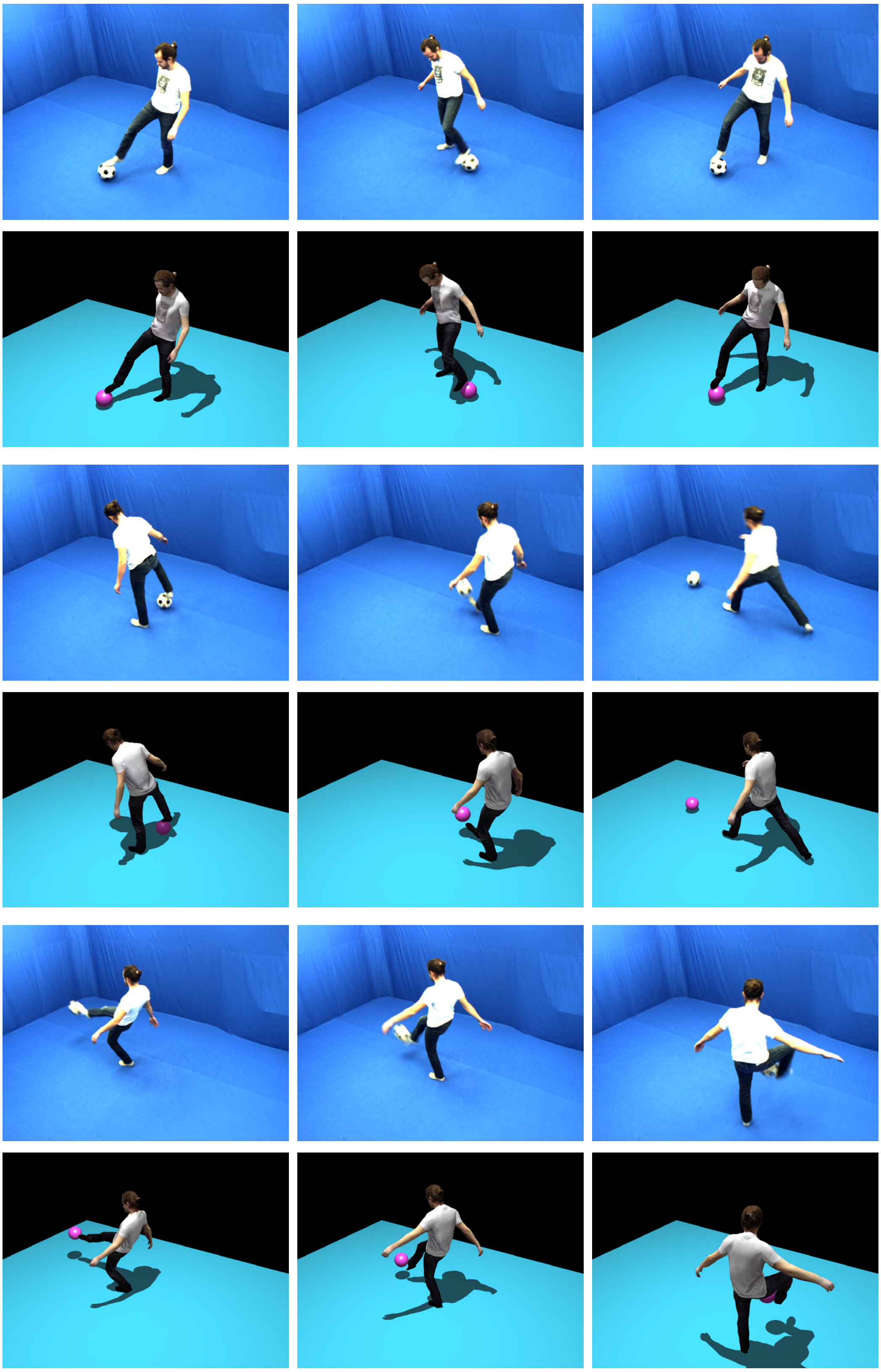


Figure 6.7: *Soccer kicks and juggles. The odd rows represent the original frames of the sequence, while the even rows represent the reconstructed ones.*

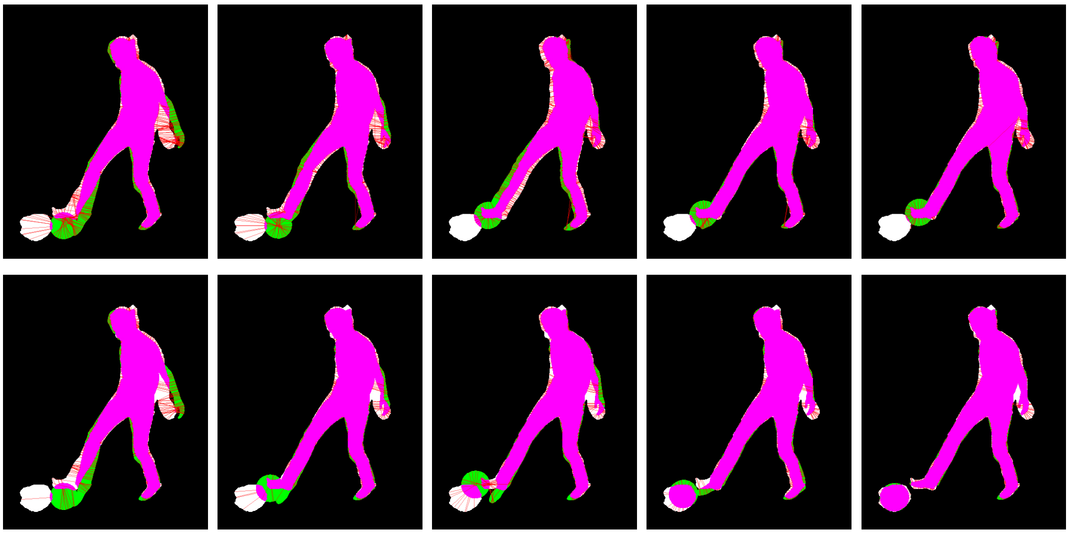


Figure 6.8: *Evolution of the algorithm during an estimation of the pose represented by the white silhouette. (Top row) No shape information is used to compute the silhouettes correspondences. (Bottom row) Shape matching metric is used.*

This sequence reveals also the importance of the optical flow information to recover fast movements. The silhouette information alone is, in fact, not able to capture fast movements because fundamentally, it is a local information. The silhouette correspondences are sought in the surrounding areas of the current pose estimate, therefore, if the actual pose is far away from its initial estimate, the algorithm is not be able to recover it converging into a local minima. Moreover, motion blur alters completely the shape of the silhouettes leading the shape matching metric to be unusable and counterproductive. However, the optical flow compensates for this lack providing the missing information.

Finally, an overall evaluation of the algorithm behavior on this sequence is given by the average PDE which, in this case, as stated by Table 6.1, is equal to 11.32.

Concerning the handshaking sequence, two actors walk alternatively in a circle, occluding each other on each pair of cameras. Afterwards, they stop, shake their hands and say goodbye with an hand gesture while they are walking in opposite directions. The generated occlusions are handled by both the visibility constraint and the shape matching metric. More precisely, the fact that we are considering the two actors as a unique entity (see Sec. 4.4.4) allows us to treat the inter-actor occlusions as self occlusions which are already considered by the visibility constraint. On Figure 6.9 we propose a comparison between the

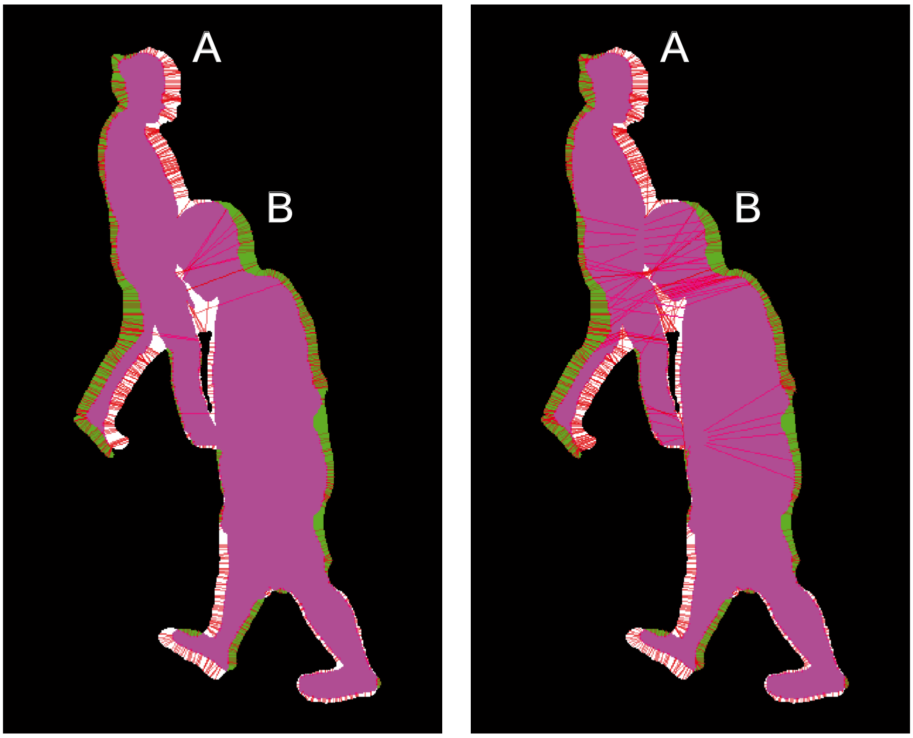


Figure 6.9: *Comparison between our approach to deal with multiple entities (Left) and the naive procedure (Right) consisting in estimating their poses one at a time, neglecting the inter-actor occlusions.*

correspondences detected using our procedure and the correspondences detected applying our pose estimation algorithm separatively on each entity. Let's call A the background actor walking on the right and B the foreground one walking on the left. It is worth to point out that using the latter approach, some points of the A's right foot are considered in the correspondences extraction procedure even if they are occluded by the B's belly. The founded correspondences tend to move, erroneously, the A's foot towards the B's back. The same happens to the A's belly which is attracted by the B's head. The other bad correspondences, visible in this figure, are due to the absence of the shape matching metric. On the other hand, our approach solves most of these mentioned issues as it can be seen in the left of Fig. 6.9.

For this sequence, since the obtained segmentation was accurate we achieved an average PDE of 10.77. However, it is worth pointing out that during the handshake a perfect contact between the two hands was not always achieved, as shown in Fig. 6.10. The hands were, in fact, not well seen by the cameras so their triangulation was hard. Moreover, the accuracy of their geometries was pretty

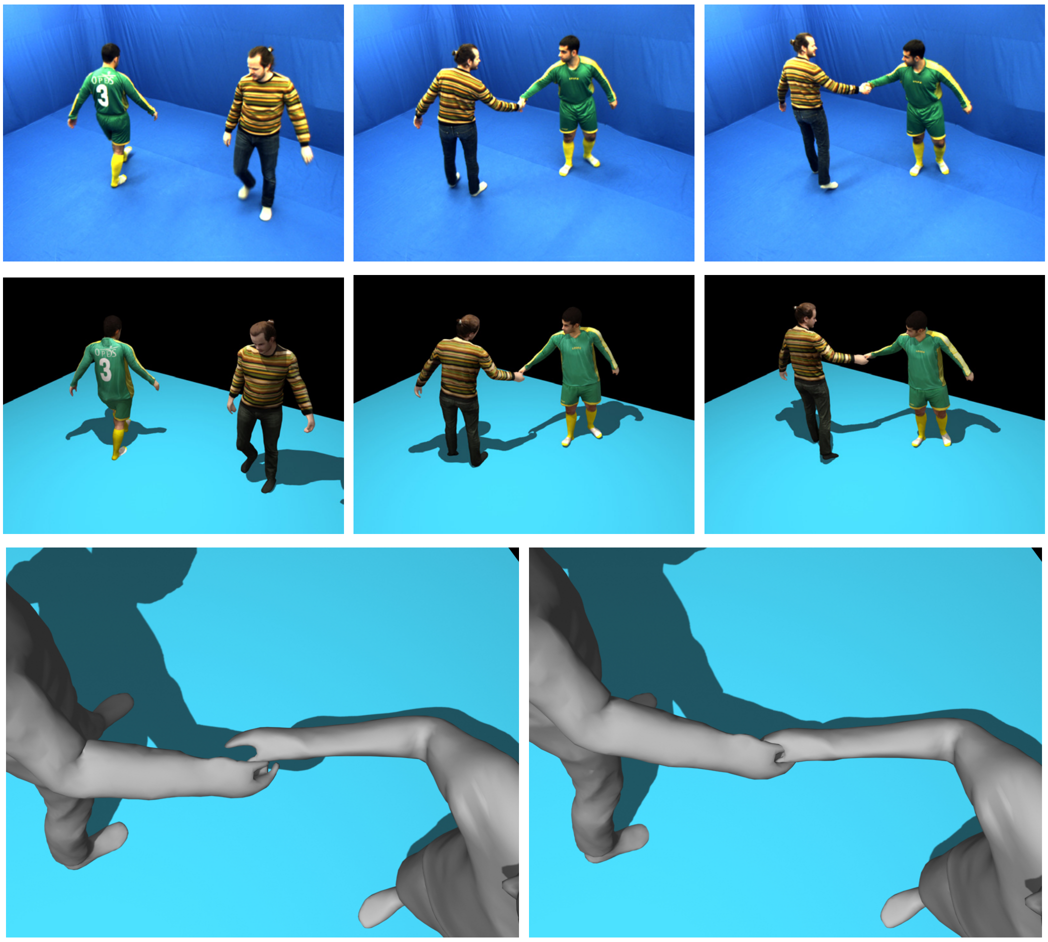


Figure 6.10: *Some real and reconstructed frames of the handshaking sequence. At the bottom two close up views of the hands positions during the handshake are proposed.*

low since the body scanner was not able to capture such a small detail. Indeed, hands are very small with respect to the rest of the human body and they are very sensible to the background subtraction errors.

On the contrary, better results are obtained in the sequences acted by the boxer model. The geometries of his gloves were, in fact, easy to acquired accurately. In particular, in the punching sequence, two actors punch and slap each other in the center of the room. In this case, the contact points between the boxer's gloves and the face of the other actor is very accurate as shown in Figure 6.11.

It is important to note that, in this case, the two actors were very close each other, therefore, the number of occlusions was higher than in other sequences while the number of extractable motion cues was lower. An evident effect of

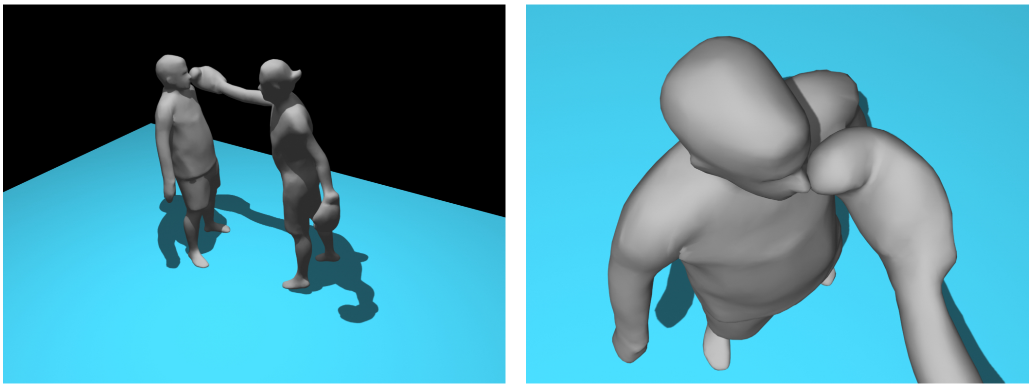


Figure 6.11: *(Left) A reconstructed frame of the punching sequence. (Right) Close up view of the contact point.*

this lack of information was observed in the estimation of the pose of both the actors' pelvis and their first spinal bones. The estimate was, in fact, corrupted by an high frequency noise which decreases the quality of the final rendering. To overcome on this situation a low-pass gaussian filter in the $SE(3)$ domain was applied to the motion capture data of the whole scene.

We finally propose, in Figure 6.12, some frames of the soccer passes sequence where two actors play for 860 frames with a ball. The first two and the last two rows depict the sequence from camera number three while the middle two from camera number one. It is interesting to observe that we were able to recover the actor's pose also during the cross leg pass depicted in the first column row number three and four.

6.2.2 Synthetic results

In order to evaluate the pose estimation error achieved by our algorithm, an experiment in a synthetic environment was performed. A generic human model was drawn, animated and rendered to provide both the synthetic video streams and the ground truth to be compared with the output of our algorithm.

Differently from the human models adopted in the real data tests, the one used for this test has only 39 degrees of freedom arranged in 17 bones. This is due to the fact that, in the drawn sequence, hands, toes and the last spinal bone were not animated, therefore, there was no need to estimate them.

The drawn animation was a kick sequence (see Fig. 6.13) characterized by fast

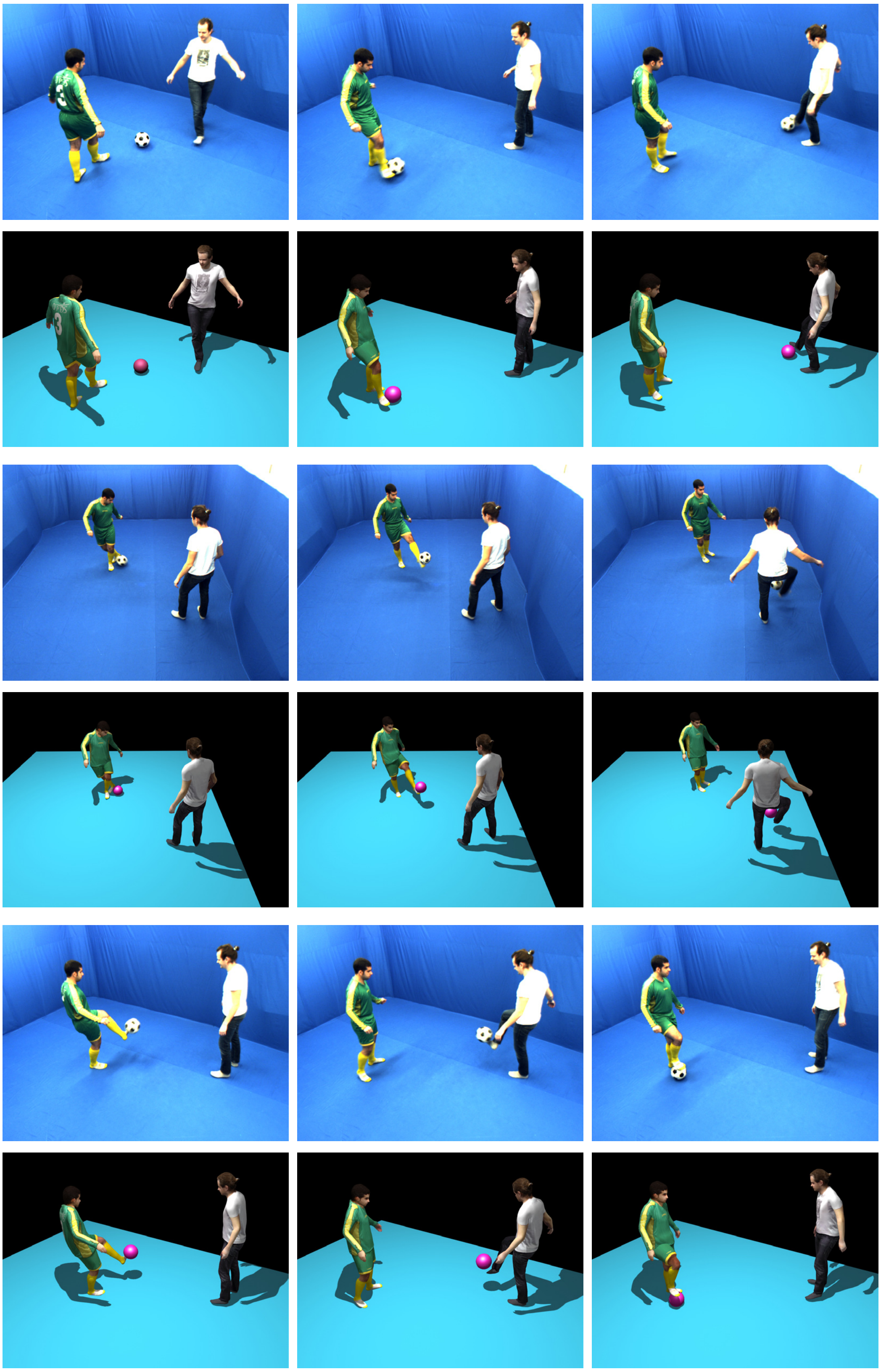


Figure 6.12: *Soccer passes sequence. The odd rows represent the original frames of the sequence, while the even rows represent the reconstructed ones.*

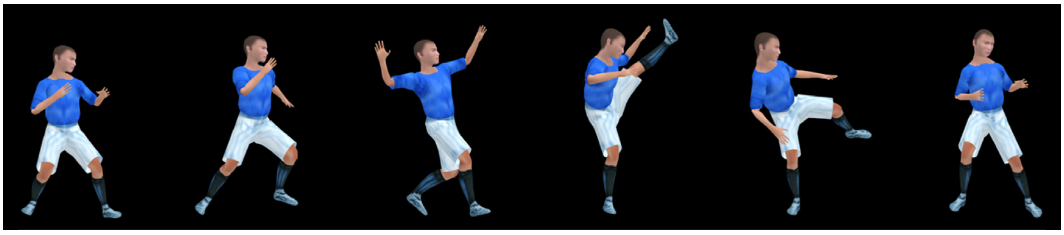


Figure 6.13: *Some frames of the kick sequence.*

and large movements for all the body parts, especially for the clavicles and for the spinal bones. Four cameras, each with a field of view of 45°, were arranged around the model at an average distance of $5.5m$. The videos were rendered at a resolution of $800x600$ pixels, so that, the image of the actor covers, in average, the 2.8% the each frame.

Figure 6.14 and Figure 6.15 show a qualitative evaluation of the pose estimation error superimposing respectively the estimated model on the actual one and the estimated skeletal structure on the actual one. The visual inspection of such superposition shows general agreement even if both models were rendered from a viewpoint different to any of the ones used to generates the video streams. Discrepancies between the two models are visible when some body parts are occluded with respect to every cameras in the scene, or when the model constraints limit some particular movements.

The availability of a ground truth allows to go beyond the qualitative evaluation. In fact, the pose estimation error was evaluated using two quantitative measurements namely, the joint position error and the bone orientation error.

The former represents the error between the joints positions of the estimated skeleton with respect to the joints positions of the ground truth. Table 6.2(Left) reports the statistics of the joint position error over the whole sequence for all the joints of skeleton. Figure 6.16 shows graphically the same error measured at each frame of the sequence for some significant joints namely, the elbows, the knees and the shoulders.

It is worth noting that the maximum error is achieved by left knee exactly during the kick, around frame 40, and it amounts to about $10cm$. This is due to the fact that, the left leg performing the kick makes a so wide movement that it touches, for one instant, the chest of the actor, loosing some silhouette information. Moreover, the left arm partially occludes the knee from the viewpoint of some cameras during exactly the kick (see Fig. 6.13). This lack of information can justify the obtained position error.

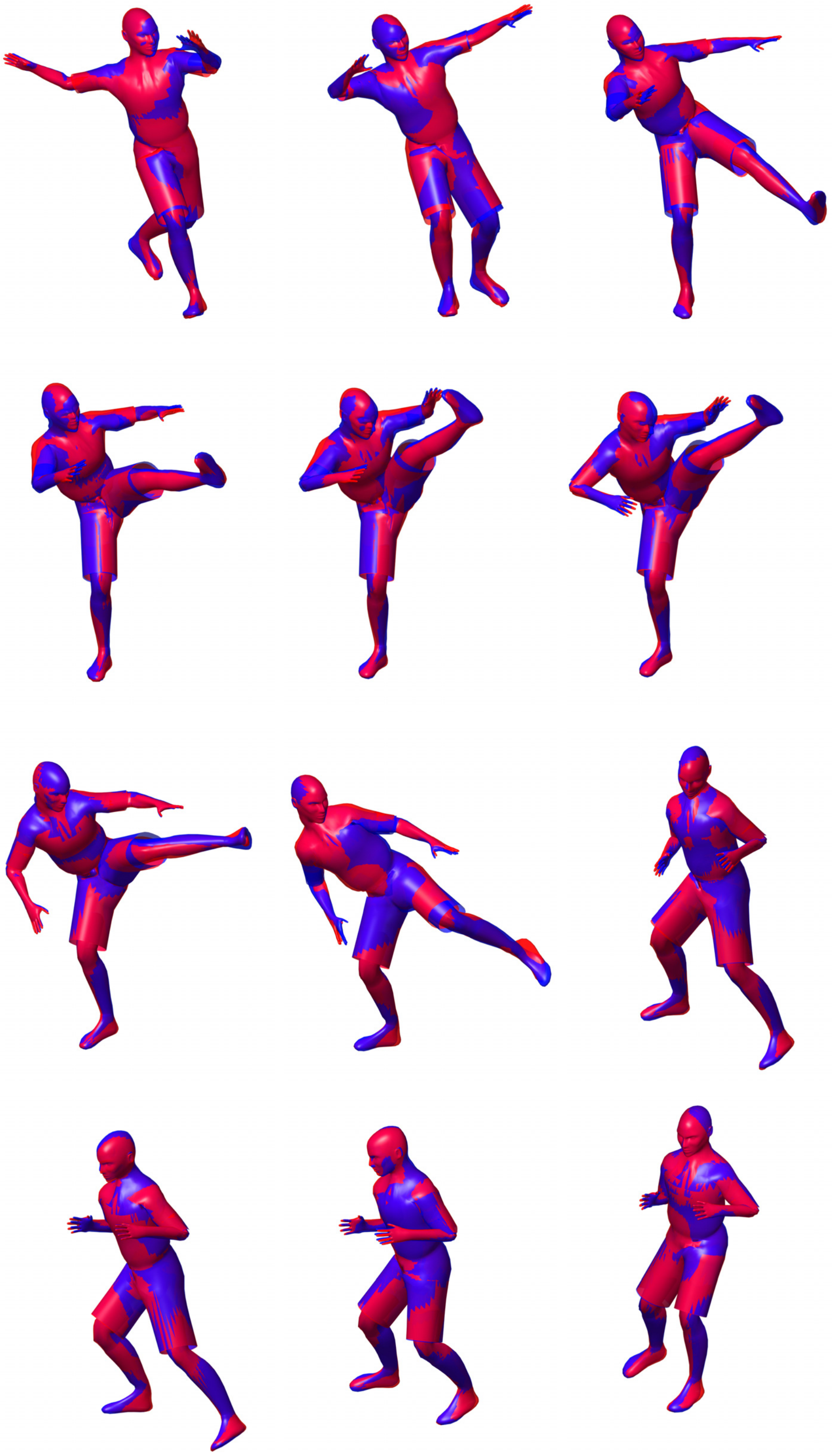


Figure 6.14: *Superposition of the ground truth model (blue) on the reconstructed one (red) seen from a viewpoint not belonging to any of the cameras used to render the sequence.*

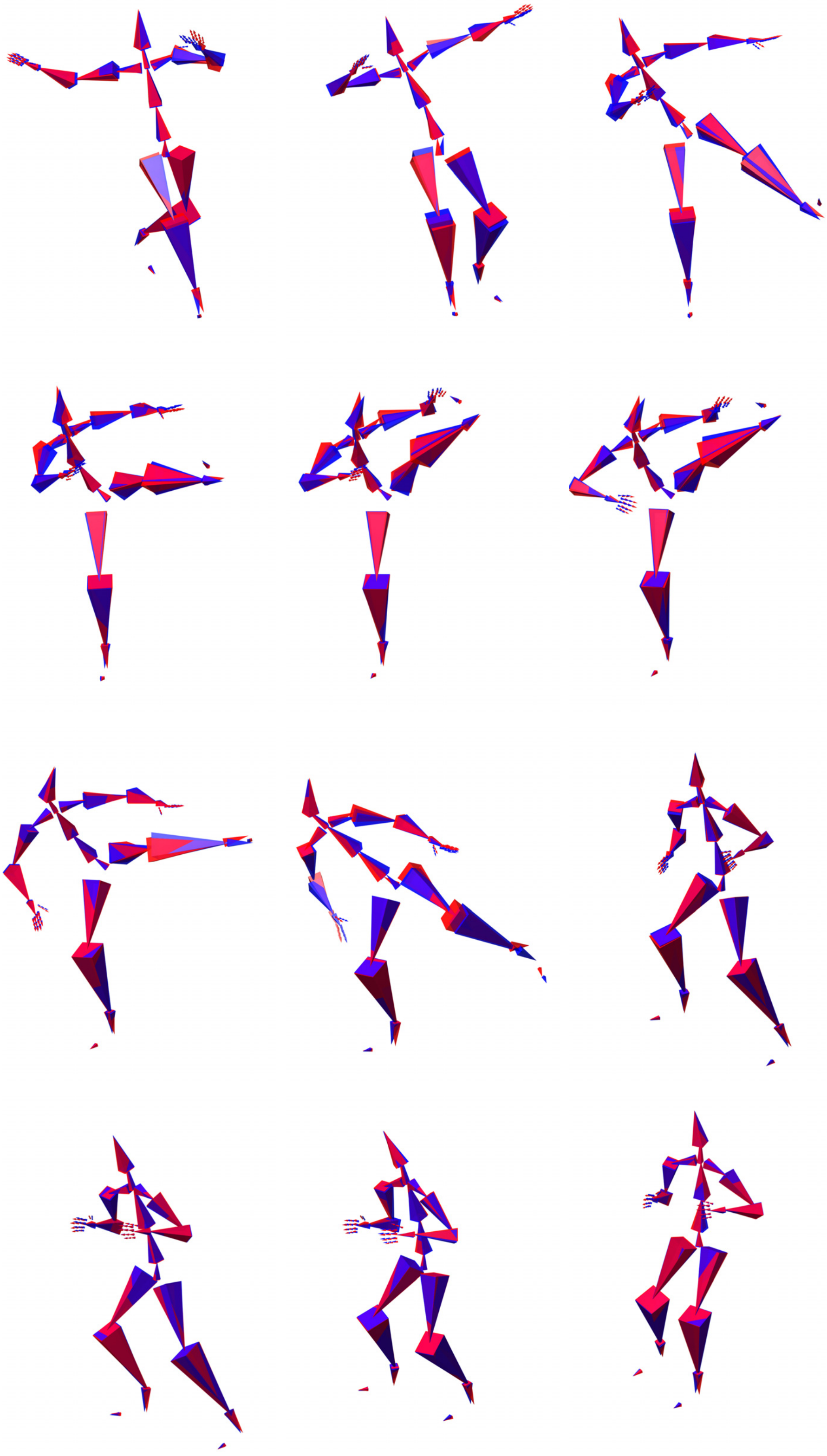


Figure 6.15: *Superposition of the ground truth skeleton (blue) on the reconstructed one (red). All these frames are in one to one correspondence with the frames shown Figure 6.14.*

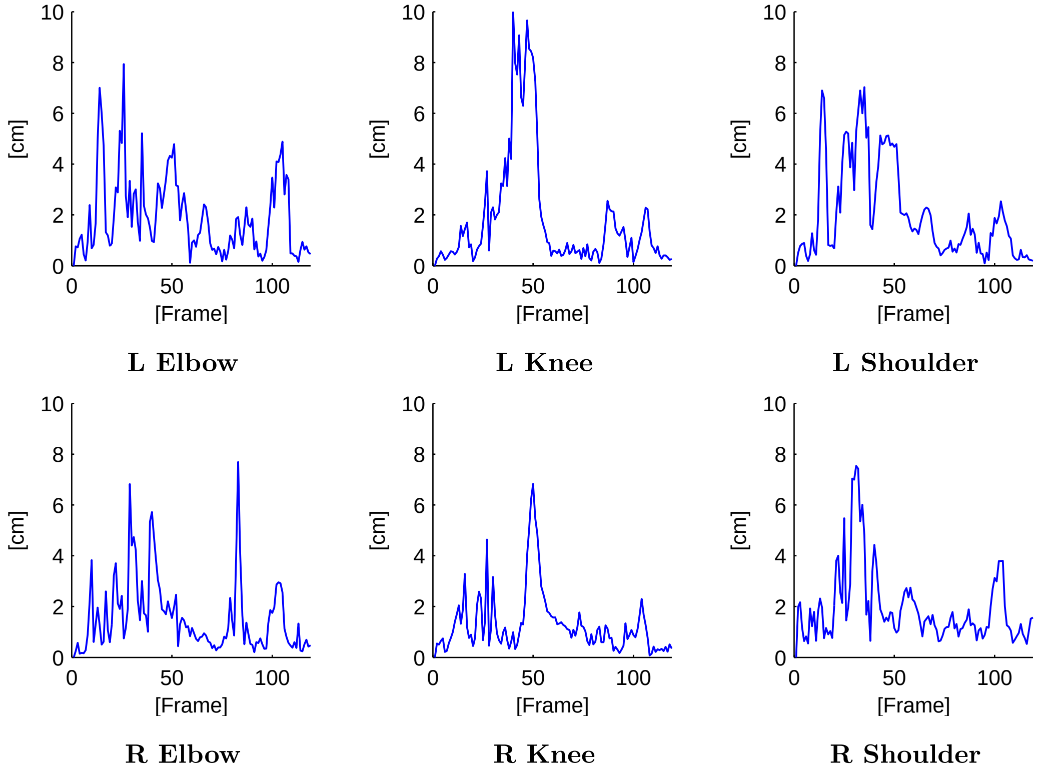


Figure 6.16: *Joint position error measured at each frame of the kick sequence.*

In spite of this situation, all the other joints are correctly estimated with an average error always below 2.5 centimeters. In particular, both shoulders and spinal bones achieve an average error below 2.1cm.

The latter quantitative measurement compares the rotation angles of the bones of the estimated skeleton against those of the ground truth. Namely, we compared the θ_j values related to the pose of this two models. This is a rather demanding comparison since the rotation error of each bone is very sensible to the rotation error of its father.

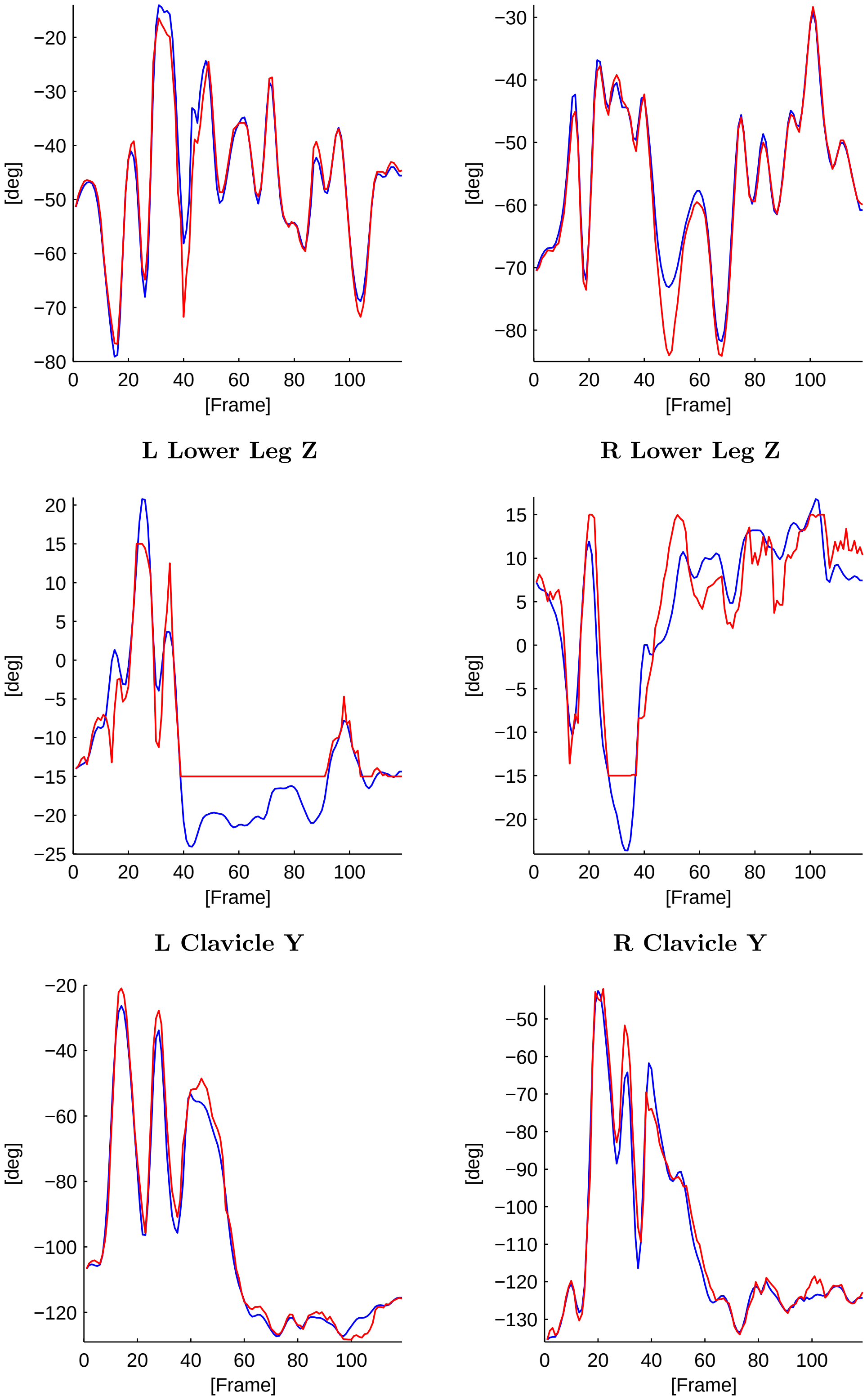
Table 6.2(Right) reports the statistics of the bone orientation error over the whole kick sequence for each bone of the skeleton. Figure 6.17 compares graphically the actual bones orientations with the estimated ones at each frame of the sequence for some significant bones, namely the lower legs, the clavicles and the forearms. Blue lines represents the ground truth angles while red lines, the estimated ones.

As the reader can see from Fig. 6.17, blue and red lines are pretty close to each other in all the graphs, except for the orientation of the left clavicle during the interval starting from frame 40 to frame 90. In fact, the estimated orientation,

	Position [cm]	
	mean	**st. dev.**
L Elbow	1.94	1.58
R Elbow	2.41	2.21
L Shoulder	2.10	1.88
R Shoulder	1.95	1.47
L Clavicle joint	0.96	0.62
R Clavicle joint	1.00	0.66
L Thigh joint	1.80	2.09
R Thigh joint	1.40	1.27
L Knee	1.85	2.36
R Knee	1.30	1.24
L Ankle	1.75	2.25
R Ankle	1.10	0.73
Pelvis	1.03	0.81
Spine lv. 0	1.45	1.18
Spine lv. 1	0.85	0.58
Head Base	1.28	0.82

	Orientation [deg]	
	mean	**st. dev.**
L Forearm Z	2.88	2.58
R Forearm Z	2.58	3.24
L UpperArm X	5.55	6.93
L UpperArm Y	5.37	5.49
L UpperArm Z	6.08	6.49
R UpperArm X	9.29	7.73
R UpperArm Y	5.13	5.80
R UpperArm Z	7.81	6.41
L Clavicle X	3.84	2.70
L Clavicle Y	3.19	2.74
R Clavicle X	4.21	3.79
R Clavicle Y	3.26	2.38
L Thigh X	3.78	4.79
L Thigh Y	4.25	5.56
L Thigh Z	6.21	7.57
R Thigh X	2.63	3.75
R Thigh Y	4.36	5.40
R Thigh Z	5.14	5.17
L Lower Leg Z	1.90	2.33
R Lower Leg Z	1.53	1.99
L Foot X	6.88	3.01
L Foot Z	5.76	7.70
R Foot Y	11.65	2.18
R Foot Z	3.85	2.83
Pelvis X	3.28	4.09
Pelvis Y	4.79	4.73
Pelvis Z	3.85	5.18
Spine lv. 0 X	3.34	2.72
Spine lv. 0 Y	4.81	6.43
Spine lv. 0 Z	6.64	6.91
Spine lv. 1 X	3.50	2.85
Spine lv. 1 Y	2.37	2.18
Spine lv. 1 Z	2.23	1.99
Head X	4.37	3.45
Head Y	3.91	3.27
Head Z	3.70	3.76

Table 6.2: *Error statistics over the whole kick sequence. (Left) Joint position error [cm]. (Right) Bone orientation error [deg].*



L Lower Leg Z

R Lower Leg Z

L Clavicle Y

R Clavicle Y

L Forearm Z

R Forearm Z

Figure 6.17: *Bones orientations comparison for each frame of the kick sequence. (Blue) Ground truth orientation angles. (Red) Estimated orientation angles.*

in this interval, stays stuck at $-15°$. This is due to the fact that the model does not allow such a wide movement constraining the clavicle y-rotation to be above $-15°$ with respect to the Vitruvian man's pose.

In spite of this problem, the average orientation error is about 4.5° with a standard deviation of 4.7°.

6.2.3 Robustness tests

Some tests were performed to evaluate the robustness of the algorithm on missing and wrong input data. In particular we tested our method in sequences where no silhouette information was supplied and in sequences where the used human model differed a lot from the actual actor.

More precisely, we used the soccer player model, used in the synthetic data tests, to track the performance of a real actor during a walk sequence. Clearly, the two models do not match each other especially in the legs, where the soccer player model wears shorts and, instead, the real actor wears jeans. However, even the presence of these discrepancies, the algorithm still tracks rather well but, with an obvious increase of the pixel discrepancy error.

In the no silhouette information test, we observe that only short sequences can be tracked. The optical flow information, used alone, suffers indeed of a drift problem leading to a misalignment of the 3D pose after about two hundred frames.

The algorithm was also tested on sequences with a fewer number of cameras namely, respectively with three and two cameras. In the latter case, we observe that only for simple movements, the full 3D pose can be recovered with a reasonable accuracy. On the contrary the three cameras case does not shows big issues, at least for the single entity sequences. What we explicitly observed is that, if a body part is not seen as silhouette from at least two cameras at the same time its pose estimation accuracy decreases. Nevertheless, a lack of silhouette information can be recovered by an optical flow information, therefore, if a body part has an optical flow correspondence and a silhouette one its pose can be recovered precisely.

Chapter 7

Conclusions

In this thesis, we presented a system for capturing the shape, the appearance and the motion of interacting people and objects using only passive and non-invasive techniques. Given a scene, where multiple people are interacting with each other and with some other objects, our system is able to provide a time-varying description of the whole 3D sequence considering both its geometry and its appearance. A user is therefore able to navigate inside this representation and look at the action from any point of view. Moreover, since the whole scene is modeled by articulated deformable models, the result of the capturing process can be used in any commercial animating software.

The acquisition is performed in two separate steps. First the shape and the appearance of each actor is acquired using a passive body scanner. Subsequently, the actors are invited inside a second location where the actual action will take place. A marker-less motion capture system is used to simultaneously capture their motions and the motion of all the objects which they interact with.

This thesis proposed an optimization framework for the pose estimation capable of handling, simultaneously and in a unified way, multiple entities interacting in the same scene. This framework also take into account the non-rigid deformations of the actors' skin allowing an accurate pose estimation of also the small and high flexible parts of the body, like the spine and the clavicles.

Two distinct sources of information, namely optical flow and silhouette, are extracted from the videos and used synergically to overcome the lacks of data given by the use of a few number of cameras recording the scene (four, in our case) and the high number of degrees of freedom to estimate (more than 80). The analysis-by-synthesis approach was adopted fusing together these two kind of information using a 2D domain functional which also avoid full 3D reconstructions of the entire scene.

The entire system was tested on seven actors and about 120 video sequences covering many different types of motions starting from the single person ones to the multiple people and multiple objects interactions sequences. Each action was recorded by four 0.8 MPixels cameras inside a blue room. The more complex evaluated sequence counts 92 degrees of freedom.

The algorithm performance was evaluated both qualitatively and quantitatively. The pixel discrepancy error was used to evaluate the converging properties of the algorithm and its peculiarity to avoid local minima. A more precise evaluation is provided by the tests made on synthetic data, where the possibility of having the ground truth, allows us to measure the exact pose estimation error in terms of both joint position error and bone angle error. The average errors found in our experiments were respectively of $2.5cm$ for the joint position error and $4.5°$ for the bone angle error.

Experiments show that, the use of the LBS deformation model inside the internal body representation allows us to correctly estimate also the pose of small and high flexible body parts. Numerically, both shoulders and spinal bones achieve an average error below $2.1cm$.

Test reveals that the use of the optical flow information is indispensable to recover sequences with fast motions and multiple occlusions. The silhouette information alone is, in fact, not able to achieve our purposes in these situations because, firstly it is a local information and secondly the motion blur generated by the fast movements corrupts the results of the background subtraction process. Moreover, silhouettes give information only on those body parts belonging to the contours of the subject seen from at least one camera. If this condition is not satisfied, like in most of the body parts during the somersault sequence, the number of pose ambiguities increases. Optical flow, instead, is able to provide motion information over the entire image, overcoming the silhouette lack. However, the optical flow alone cannot reconstruct long sequences as seen from Sec. 6.2.3 because it suffers of drift problems which cannot be neglected after 200 frames. Therefore, only the synergically use of these two kind of information can recover the full motion information of the tested sequences.

The experiments described in Sec. 6.2.1 showed that the use of the shape matching metric and our formulation for multiple entities allows us to recover simultaneously the motion of multiple people and objects interacting in the same scene. Their inter-occlusions are treated as self-occlusions of an unique entity, while, the shape matching discriminates the silhouette belonging to each object.

The proposed body scanner, costing less than two thousand euros, was able to provide us the initial shapes of our actors, counting more than 500 thousands faces and with an accuracy of $4.2mm$. The proposed appearance capture procedure allows us to recover textures of about 21 MPixels free from the illumination artifacts, ghostings and blurs.

Currently, the entire system suffers of two main limitations, namely, the need of a controlled environment to acquire the scene and the need of a separate step to acquire the shape and the appearance of each subject. The latter one could not be considered as a limitation if the purposes of the acquisition are, for instance, character animations, while the former one could be relaxed using some of the recent progresses on the segmentation procedure. In this case, the target is to export our system to acquire accurately the motion also in crowded urban areas.

Moreover, the formulation proposed in Chapter 4 and, in particular, the multiband analysis of the deformable models, suggest an easy future extension of this work for capturing also the deformations which cannot be modeled by the LBS model, like, for instance, the ones of normal clothes.

Appendix A

3D Content Creation by Passive Optical Methods

ATTENTION: This chapter is a copy of my work published in the book "*3D ONLINE MULTIMEDIA AND GAMES: Processing, Visualization and Transmission*" [11] which can be found in the libraries starting from fall 2008.

A.1 Introduction

The term *3D Digitizing/Modeling* is referred to the action of representing a real object by a mathematical description, called *3D model*, that can be processed by a computer. Since the beginning, the possibility of obtaining 3D models of real objects or scene has paved the way to a wide range of new applications in fields such as cultural heritage documentation, medicine, media and entertainment (movies, video games, etc...), virtual simulation, human-computer interaction (HCI), industrial prototyping, reverse engineering, scientific visualization, e-commerce and marketing, surveillance, just to name a few.

The construction of the 3D model of a real object or scene by optical sensors, also referred to as *3D modeling pipeline*, essentially consists of four steps: 1) data acquisition, 2) calibration, 3) reconstruction, and 4) model editing. Any optical sensing device used to collect data can only capture the surface front side and not what is occluded by it. Therefore, a full model must be built from a number of images covering the entire object (data acquisition). In order to perform 3D reconstruction, the camera's parameters must be estimated by a procedure called calibration. Such information can also be obtained from the acquired images if they represent some common regions (by a procedure which is typically called self-calibration). Reconstruction is then performed and the resulting model is stored in an efficient description such as polygonal meshes, implicit surfaces, depth maps or volumetric descriptions. In practical situations, reconstruction may lead to models with some imperfections; thus, a further repairing step is recommend (model editing) [39, 83].

Optical 3D reconstruction methods can be classified into passive or active methods based on the type of sensors used in the acquisition process. Passive sensing refers to the measurement of the visible radiation which is already present in the scene; active sensing refers instead, to the projection of structured light patterns into the scene to scan. Active sensing facilitates the computation of 3D structure by intrinsically solving the correspondence problem which is one of the major issues with some passive techniques. For a detailed description of the operations of the 3D modeling pipeline by active sensing see [131, 129]. In general, active techniques such as those based on laser scanning tend to be more expensive and slower than their passive counterparts. However, the best active methods generally produce more accurate 3D reconstructions than those obtained by any passive technique.

Passive optical methods, as previously mentioned, do not need auxiliary light sources. In this case, the light reflected by the surface of the object comes from natural sources, that is, sources whose characteristics are generally unknown and in most cases, not controllable by the acquisition process. Furthermore, passive methods do not interact in any way with the observed object, not even with irradiation. Passive reconstruction, in principle, can use any kind of pictures, i.e., it does not need pictures taken for 3D reconstruction purposes (even holiday photographs could be used). Passive methods are more robust than their active counterparts, can capture a wider range of objects, can be obtained by inexpensive hardware (such as a simple digital camera) and are characterized by fast acquisition times. Such features are the reason for the attention they received and they continue to receive. Their drawbacks concern accuracy and computational costs. Indeed, passive reconstruction algorithms are complex and time-consuming. Moreover, since their acquisition scenarios are often far from the ideal conditions, noise level is typically much higher than that of active methods which tend to guarantee optimal conditions.

Passive optical methods are classified by the types of visual cues used for

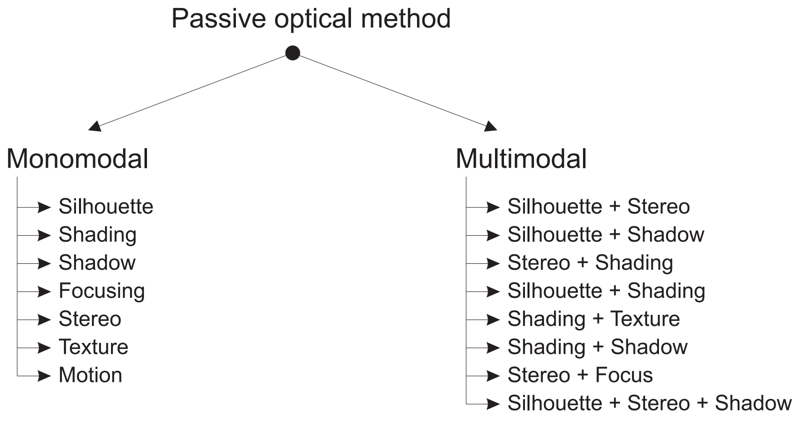


Figure A.1: *Overview of existing passive optical methods.*

3D reconstruction. For this reasons, they are also called *"Shape from X"* (SfX) techniques, where X stands for the cue or the cues used to infer shape. Methods which deal with one single type of visual cue are called monomodal whereas methods jointly exploiting information of different types are called multimodal. The simultaneous use of different cues, in principle, is clearly more powerful than the use of a single one; however, this poses the challenge of how to synergistically fuse different information avoiding mutual conflicts.

Figure A.1 proposes a taxonomy of the passive optical methods. Typical information used for reconstruction are:

- Silhouette or apparent contour;
- Shading;
- Shadow;
- Focusing;
- Pictures differences, i.e., stereo information;
- Texture;
- Motion;

This chapter reviews 3D reconstruction by passive optical methods. This is not an easy task in light of the broad scope of the topic. The spirit we adopted is to give a conceptual outline of the field, referring the reader to the literature for details. We also reserve special attention to recent methods. section A.2 introduces basic concepts and terminology about the image formation process. section A.3 reviews major monomodal methods: shape from silhouette, shape from shading, shape from shadows, shape from focus/defocus and shape from stereo. Finally, in section A.4 we introduce a framework for multimodal methods, focusing on the deformable model technique.

A.2 Basic notation and calibrated images

A calibrated image is an image for which all the parameters of the camera used to take it are known. Formally, a calibrated image is an ordered pair (I, ζ) where I is an image and ζ is an image formation function $\Re^3 \rightarrow \Re^2$ that maps the points from the physical 3D world to the image plane. An image is a function from a rectangular subset of $\Re^2$ representing the image space coordinates to an interval of $\Re$ representing the image intensity values. In this section, the image formation process is approximated using the ideal pinhole camera model (see Fig. A.2) with lens distortion. This means that neither the effects due to finite aperture nor other types of lens aberrations are considered. In this case, ζ can be expressed as the combination of two functions, namely

$$\zeta = \phi \circ V \tag{A.1}$$

where $\phi : \Re^2 \rightarrow \Re^2$ is a nonlinear bijection representing the camera lens distortion and V is a function $\Re^3 \rightarrow \Re^2$, called *view*, incorporating both the pinhole model and the camera point of view information. Function V is a combination of two further functions, i.e., $V = \pi \circ T$. Function π is the pinhole perspective projection[1] simply defined as $\pi\left(x, y, z\right) = \left(\frac{x}{z}, \frac{y}{z}\right)$ for all point $P = (x, y, z)$ in $\Re^3$ with $z > 0$. Function $T : \Re^3 \rightarrow \Re^3$ is an affine bijective transformation which performs 3D coordinates transformation from the world space to the camera space. Let us note that, given a calibrated image (I, ζ), one can always find its non-distorted version $(I \circ \phi, V)$ by estimating camera lens distortion parameters ϕ from image I. This is a classical inverse problem for which a vast literature is available. Popular methods are due to [154] which estimates ϕ using known calibration patterns and to [122] which uses the knowledge that the image represents straight lines of the scene.

Projective Geometry is a powerful framework for describing the image formation process, not adopted in this chapter for reasons of simplicity. Interested readers are referred to [59] for an excellent introduction.

By definition, transformation T can be written as

$$T\left(P\right) = M \cdot P + O \tag{A.2}$$

[1] This model was first proposed by Brunelleschi at the beginning of the 15th century.

where M is an invertible $3x3$ matrix and $O, P \in \Re^3$. Furthermore, M and O form to the so-called projection matrix K, a $3x4$ matrix defined as follows

$$K = \begin{bmatrix} M & O \end{bmatrix} \tag{A.3}$$

Projection matrix K is related to the physical model of the ideal pinhole camera and can be decomposed according to the following scheme

$$K = I \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times E \tag{A.4}$$

where I is the intrinsic matrix, depending on the so-called intrinsic parameters only due to the camera internal characteristics, and E is the extrinsic matrix, depending on the so-called extrinsic parameters only due to the camera position and orientation in the space. Namely, matrix I is defined as follows

$$I = \begin{bmatrix} \frac{f}{p_x} & \frac{(\tan\alpha)f}{p_y} & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

where f (expressed in millimeters) is the camera focal length, that is the distance between the sensor surface (also known as retinal plane or image plane) and pinhole C (also known as center of projection); p_x, p_y respectively are the width and the height in millimeters of a single pixel on the retinal plane; α is the skew angle, measuring how much the image axes x and y are away from orthogonality; $c = (c_x, c_y, 1)$ is the principal point of the camera, i.e., the point at which the optical axis intersects the retinal plane. We recall that the optical axis is the line, orthogonal to the retinal plane, passing through the center of projection C. Another useful parameter is the camera field-of-view along the y axis defined as

$$FOV_y = 2arctan\left(\frac{p_y \, N_y}{2f}\right) \tag{A.6}$$

where N_y is the vertical resolution of the sensor. Figure A.2(above) shows the ideal pinhole camera. Rays of light pass through the pinhole and form an inverted image of the object on the sensor plane. Figure A.2(below) shows an equivalent pinhole model where the image plane is placed in front of the center of projection obtaining a non-inverted image.

Matrix E is defined as

$$E = \begin{bmatrix} R & t^T \\ 0 & 1 \end{bmatrix} \tag{A.7}$$

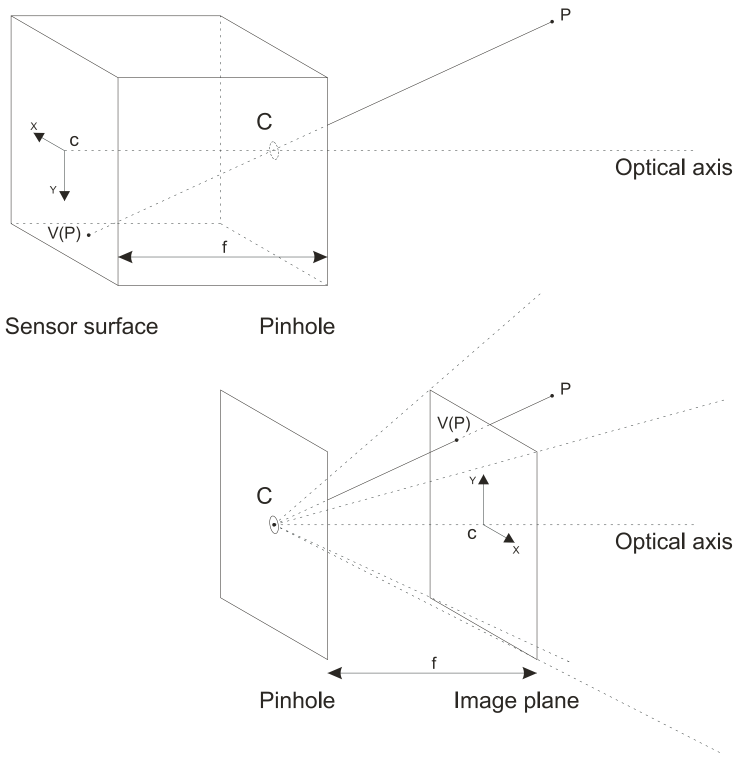


Figure A.2: *Ideal pinhole camera (above) and its equivalent model (below) where the image plane is placed in front of the center of projection.*

where R is $3x3$ rotation matrix and t is a vector belonging to $\Re^3$. For example, given a camera with center of projection C, optical axis D and *up-vector* U (that is the y axis of the camera reference system), the relative extrinsic matrix is:

$$E = \begin{bmatrix} B^{-1} & -B^{-1}C^T \\ 0 & 1 \end{bmatrix} \tag{A.8}$$

where:

$$B = \begin{bmatrix} (U \times D)^T & U^T & D^T \end{bmatrix} \tag{A.9}$$

The center of projection $C = (X_c, Y_c, Z_c)$ can be obtained from the columns of projection matrix $K = \begin{bmatrix} k1 & k2 & k3 & k4 \end{bmatrix}$ as follows

$$X_c = \frac{\det\left(\begin{bmatrix} k2 & k3 & k4 \end{bmatrix}\right)}{Q} \tag{A.10}$$

$$Y_c = -\frac{\det\left(\begin{bmatrix} k1 & k3 & k4 \end{bmatrix}\right)}{Q} \tag{A.11}$$

$$Z_c = \frac{\det\left(\begin{bmatrix} k1 & k2 & k4 \end{bmatrix}\right)}{Q} \tag{A.12}$$

where:

$$Q = -\det\left(\begin{bmatrix} k1 & k2 & k3 \end{bmatrix}\right) \tag{A.13}$$

In order to extract 3D information from a set of images, the related view functions must be estimated for each image of this set. This can be done in two ways: conventional calibration or self-calibration. The first approach uses pictures imaging a known target object such as a planar checkerboard. In this case, function V can be estimated by solving an overconstrained linear system [59]. Self-calibration instead, computes the view functions associated to a set of un-calibrated images without any information about the scene or any object in it. These methods, see for instance [99], essentially extract relevant features from two or more images then, find the matching between them and finally, proceed like conventional calibration methods.

A.3 Monomodal methods

This section introduces the most common monomodal methods namely, the methods using silhouette, shading, shadow, focus and stereo as 3D reconstruction information. Texture and motion are excluded from this analysis, however the interested reader is referred to [51] and [67] for an example of these two techniques.

A.3.1 Silhouette

Algorithms which reconstruct 3D objects using only silhouette information extracted from a set of images are called *"Shape from Silhouette"* methods. They were first proposed in [13] and afterwards formalized in [78], where the concept of *visual hull* was first introduced.

All these methods must face the problem of extracting silhouette information from the set of images. In other words, in each picture, they must identify the points belonging to the object to be acquired with respect to the background. This problem does not have a general solution as it strongly depends on the scene characteristics. The most common approaches to this task are chroma keying (e.g., blue screen matting [29]), background subtraction [117], clustering [120] and many other segmentation techniques. For a comprehensive account see [88]. However, silhouette information is affected by two types of error. The first one is the quantization error due to image resolution and it is directly proportional to the camera-object distance z as

$$\varepsilon_x = \frac{p_x}{2f} z, \quad \varepsilon_y = \frac{p_y}{2f} z \tag{A.14}$$

The second one depends on the specific silhouette extraction method and its amount is usually confined within ±1 pixel.

In order to recall the concept of visual hull, some definitions related to the notion of contour may be useful. Given a view V and a closed surface M in $\Re^3$, let us denote by $V(M)$ the projection (or the *silhouette*) of M on the image plane of V, i.e., the shape of M viewed by V, and by

$$\gamma_M^V = \partial V(M) \tag{A.15}$$

the *apparent contour* of M viewed by V, and by

$$\Gamma_M^V = V^{-1}\left(\gamma_M^V\right) \tag{A.16}$$

the *3D contour* of M viewed by V.

By definition $V(M)$ is a set of points in $\Re^2$ and its boundary γ_M^V is a set of curves in $\Re^2$ which do not intersect each other. As we can easily see with the aid of Figure A.3, neither $V(M)$ nor γ_M^V are generally regular. Indeed, it is likely

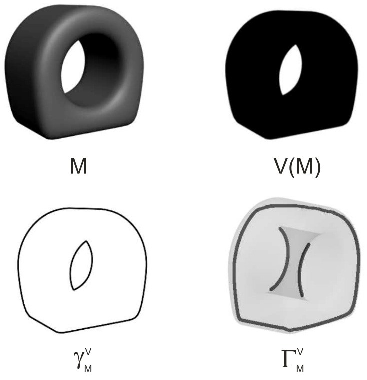


Figure A.3: *M is a 3D object. $V(M)$ represents the silhouette of M, γ_M^V its apparent contour and Γ_M^V its 3D contour. In the figure, Γ_M^V is slightly rotated with respect to the point of view of the other three pictures in order to evidence that Γ_M^V is a set of not necessarily closed 3D curves.*

that they have some singularities. On the contrary, Γ_M^V is a set of not necessarily closed curves in $\Re^3$, with points belonging to M.

Shape from silhouette methods use $V(M)$ as source of information. However, there exists a class of methods, called *"Shape from Contour"* [32], that use the apparent contour γ_M^V instead of $V(M)$ in order to infer shape.

The concept of *visual hull* [78] is a fundamental definition for the shape from silhouette methods.

Definition 1 *Given a set of views $R = (V_1, \ldots, V_n)$ and a closed surface M in $\Re^3$, the visual hull of M with respect to R, denoted as $vh(M, R)$, is the set of points of $\Re^3$ such that $P \in vh(M, R)$ if and only if for every view $V_i \in R$, the half-line starting from the center of projection of V_i and passing through P, contains at least one point of M.*

In other words, the visual hull of a surface M related to a set of views R is the set of all points in the 3D space which are classified as belonging to the object for every view $V_i \in R$. Laurentini [78] proved that the boundary of the visual hull $\partial vh(M, R)$ is the best approximation of M that can be obtained using only silhouette information coming from the projections of M in each view of R. Some implications of this result follow:

- the visual hull always includes the original surface, i.e., $M \subseteq vh(M, R)$, or in other words, the visual hull is an upper-bound of the original object;

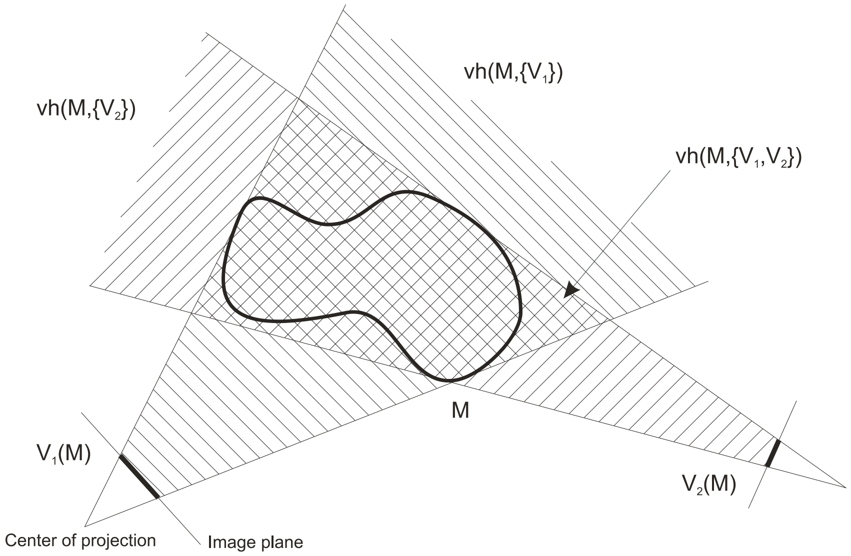


Figure A.4: *Computation of the visual hull as intersection of the visual cones generated by V_1 and V_2.*

- $\partial vh\left(M,R\right)$ and M have the same projections in R, in other words for every $V \in R$, we have:
$$V(M) = V\left(\partial vh\left(M,R\right)\right) \tag{A.17}$$
- If $R_1 \subseteq R_2$ then $vh\left(M,R_2\right) \subseteq vh\left(M,R_1\right)$
- $vh\left(M,R\right) = \bigcap_{V \in R} vh\left(M,\{V\}\right)$

The last property suggests a method for computing the visual hull as the intersection of the visual cones $vh\left(M,\{V\}\right)$ generated by M for every view $V \in R$ (see Fig. A.4). A visual cone $vh\left(M,\{V\}\right)$ is formed by all the half-lines starting from the center of projection of V and intersecting the projection of M on the image plane of V.

All shape from silhouette methods are based on the above principle. They can be classified by the way the visual hull is internally represented, namely by voxels or by polyhedra. The former class, called *"Volume Carving"* algorithms [121], was the first to be proposed. The idea behind it is to divide space into cubic elements of various sizes, called *voxels*, in order to store volume information of the reconstructed object. The latter class of algorithms, recently formulated in [95], represents the boundary of the reconstructed visual hull by polygonal meshes. They are proposed for real-time applications aimed at acquiring, transmitting and rendering dynamic geometry. Indeed, polyhedral visual hull can be rapidly computed and rendered using the projective texture mapping feature of modern

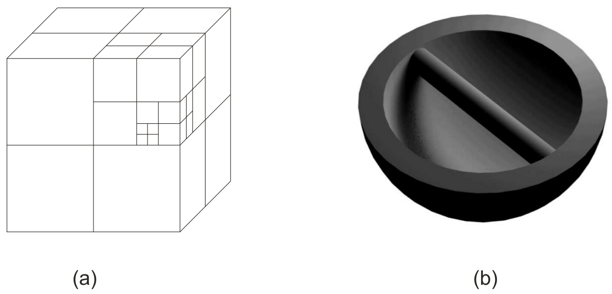


Figure A.5: *(a) Space subdivision by an octree. (b) Example of surface M for which its external visual hull has genus lower than the genus of M. The visual hull cannot completely describe the topology of this surface.*

graphics cards [86]. Besides, polyhedral representations give exact estimations of the visual hulls avoiding the quantization and the aliasing artifacts typical of the voxel approach. However, voxel representations are preferred when the result of shape from silhouette is used as first approximation to be refined by other reconstruction algorithms such as shadow carving (see section A.3.3) and some multimodal methods (see section A.4).

For this reason the remaining of this section focuses on the volume carving algorithms. In this case, the 3D space is divided into voxels which can bear three types of relationship with respect to the visual hull: "belong", "partially belong" or "not belong". In order to verify such characteristics one must check if a voxel completely belongs to every visual cone[2]. In this case the voxel belongs to the visual hull of M. Otherwise, if the voxel is completely outside at least one visual cone, then it does not belong to the visual hull. In any other case, the voxel partially belongs and one must further subdivide it and repeat the check with respect to its sub-voxels until the desired resolution is reached.
Data structures like *octrees* [43] allow for a fast space subdivision and reduce the memory requirements. An octree is a tree where each internal node has 8 children. Every node j is associated with a cube B such that the set of the cubes associated to each child of j is an equipartition of B. The root of the tree represents the whole space under analysis, which is divided into 8 cubes of equal size as shown in Fig. A.5(a). Each of these cubes can be again subdivided into 8 further cubes or alternatively be a leaf of the tree. The possibility of arbitrarily stopping the subdivision is the key characteristic of octrees. In fact, octrees can

[2] Observe that, since voxels are cubes, one can determine whether all their points belong to a visual cone only by checking the eight vertices of the cube.

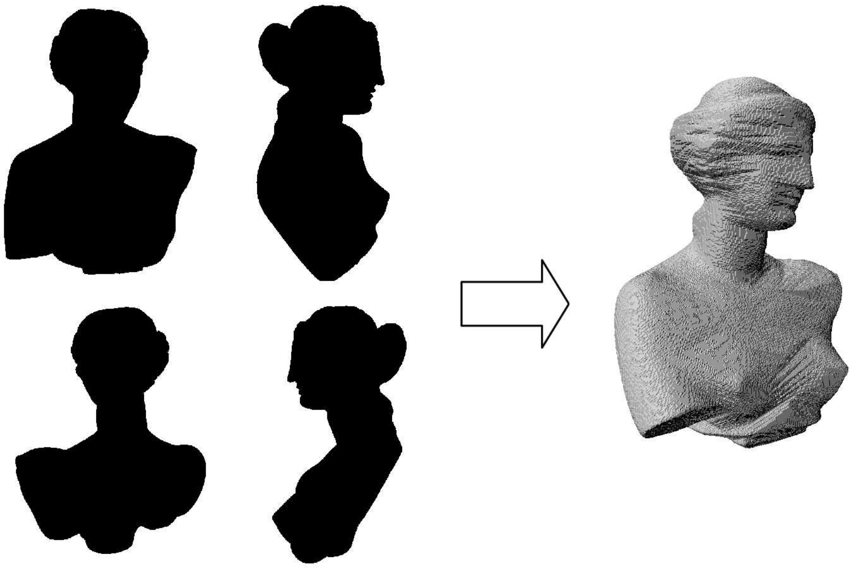


Figure A.6: *Model obtained by volume carving algorithm.*

optimize memory requirements since they allow to describe volumes by a multiresolution grid where detailed regions are described at resolutions higher than those in uniform regions.

Finally, in order to find a polygonal mesh representation of the boundary of the estimated volume, one may resort to the *"marching cubes"* algorithm [33]. Figure A.6 shows an example of a model obtained by a volume carving algorithm.

Let us observe that given the set of all possible views whose centers of projection are outside the convex hull of M, the relative visual hull, called $vh_{\infty}(M)$, is in general not equal to M. In fact, $vh_{\infty}(M)$ cannot describe the concave regions of M which are not visible from viewpoints outside the convex hull of M. As a consequence, in general, the visual hull cannot completely capture the topology of a surface. $vh_{\infty}(M)$ is called the external visual hull and it is a subset of the convex hull of M. Figure A.5(b) shows an object for which its external visual hull has genus lower than that of the original surface.

In conclusion, shape from silhouette algorithms are fast and robust but can only reconstruct a small set of objects, i.e., those objects the visual hulls of which, related to the available views, are similar to their surfaces.

A.3.2 Shading

Shading information is used in both photometric stereo and shape from shading algorithms. The former operates with a series of pictures of the object taken under different lighting conditions. The latter instead, recovers the surface shape from the brightness of a single picture.

Both methods rest on approximations of the reflectance characteristics of the object to be reconstructed, that are the relationships between incoming illumination to a point on the surface and the light reflected by it. For this reason, it may be useful to recall some radiometric definitions.

Light power is the amount of light energy per unit time, measured in Watt $[W]$. The *outgoing radiance* at surface point P in the direction $\omega_o = (\theta_o, \phi_o)$ (where θ_o and ϕ_o are the two angles defining direction ω_o) is the light power per unit area perpendicular to ω_o emitted at P in the unit solid angle of direction ω_o. Such a radiance is denoted as $L_o(P, \omega_o)$ where the subscript o denotes that it is an outgoing radiance. It is measured in $[W][m]^{-2}[sr]^{-1}$, where *Steradian* [sr] is the unit of solid angle. On the other hand, the *incoming radiance* $L_i(P, \omega_i)$ at surface point P in direction $\omega_i = (\theta_i, \phi_i)$ is the incident light power at P per unit area perpendicular to ω_i in the unit solid angle of direction ω_i. Note that, if the surface normal at P forms an angle β with respect to direction ω_i, the infinitesimal area dA centered at P seen from the direction ω_i is $dA\cos(\beta)$. Therefore, the incoming light power per unit area contributed to P by the light sources through the infinitesimal solid angle $d\omega$ of direction ω_i, is $L_i(P, \omega_i)\cos(\beta)\,d\omega$. This quantity is called *incident irradiance* at surface point P in the direction ω_i and it is measured in $[W][m]^{-2}$.

The *bidirectional reflectance distribution function* (BRDF) was introduced in [110] as a unified notation of reflectance in terms of incident and reflected beam geometry. It is defined as the ratio between the outgoing radiance at surface point P in the direction ω_o and the incident irradiance at P in the direction ω_i, i.e.,

$$f_r(P, \omega_o, \omega_i) = \frac{L_o(P, \omega_o)}{L_i(P, \omega_i)\cos(\beta)\,d\omega} \tag{A.18}$$

and it is measured in $[sr]^{-1}$.

The actual BRDF of an object is usually a very complex function and it is difficult to estimate in practical situations, therefore a number of approximations are used instead. For example, *Lambertian* (or ideal diffuse) surfaces, i.e., surfaces that reflect light equally in all directions, lead to a strong simplification namely, they have a constant BRDF

$$f_r\left(P,\omega_o,\omega_i\right)=\rho\left(P\right) \tag{A.19}$$

where ρ is called the *albedo* or the diffuse reflectance of the object. Models for partially specular surfaces were developed by Torrance-Sparrow [153], Phong [116] and Blinn [17]. The last two models are widely used in computer graphics.

The algorithms described in this section consider only Lambertian surfaces and local shading models; thus, neither specularity nor interreflections are considered. However, state of the art of photometric stereo and shape from shading algorithms make use of more general BRDF models such as the simplified Torrance-Sparrow model used in [61].

Some definitions used in both types of algorithms are in order. Let M be the unknown surface in $\Re^3$ and let $I\left(x,y\right)$ be the image intensity seen by a view V. If the surface point $P\in M$ is visible from the viewpoint V then $I\left(V\left(P\right)\right)$ is its brightness. Clearly, $I\left(V\left(P\right)\right)$ is proportional to the outgoing radiance leaving P in direction of the center of projection of V. Therefore, for Lambertian objects illuminated by a single point light source, one can write

$$L_o\left(P,\omega_o\right)=\rho\left(P\right)L_i\left(P,\omega_i\right)\cos\left(\beta\right) \tag{A.20}$$

thus,

$$I\left(V\left(P\right)\right)=\rho\left(P\right)l\left(P\right)L\left(P\right)\cdot N\left(P\right) \tag{A.21}$$

where $l\left(P\right)$ and $L\left(P\right)$ are respectively, intensity and direction of the incident light at P, $\rho\left(P\right)$ is the surface albedo at P and $N\left(P\right)$ is the surface normal.

Photometric stereo

Photometric stereo was first introduced in [163]. Given a set of calibrated images $\left(I_1,V\right),\ldots,\left(I_n,V\right)$ taken from the same point of view V but under different lightings $L_1,\ldots,L_n$, one can estimate surface normal $N\left(P\right)$ for every visible point of M. Let

$$\mathbf{I}\left(x,y\right)=\left[I_1\left(x,y\right),\ldots,I_n\left(x,y\right)\right] \tag{A.22}$$

be the vector of all measured brightness at image point $\left(x,y\right)=V\left(P\right)$, for any visible point P of M, and let

$$\mathbf{L}\left(x,y\right)=\begin{bmatrix} l_1\left(P\right)L_1\left(P\right)\\ \vdots \\ l_n\left(P\right)L_n\left(P\right)\end{bmatrix} \tag{A.23}$$

be the matrix of all light directions and intensities incident at P. From Eq. (A.21), one may write

$$\mathbf{I}^T(x,y) = \rho(P)\,\mathbf{L}(x,y) \times N^T(P) \tag{A.24}$$

which is a linear system of n equations in the three unknowns $\rho(P)\,N(P)$ [3]. Eq. (A.24) has a unique solution when $n > 3$ and it can be solved using least square methods.

Once the values of $\rho(P)\,N(P)$ are available for each visible point P, one can extract the surface albedo and the normal at P using $\rho(P) = \|\rho(P)\,N(P)\|$ and $N(P) = \rho(P)\,N(P) / \|\rho(P)\,N(P)\|$ respectively. Retrieving shape from normals is trivial under the assumption that the view V performs an orthographic projection. Indeed, let us represent M by a Monge patch description, i.e.,

$$M = \{(x, y, z(x,y)) \mid \forall (x,y)\} \tag{A.25}$$

where $z(x,y)$ is the surface depth at (x,y). Consequently, the surface normal at $P = V^{-1}(x,y) = (x, y, z(x,y))$ is

$$N(P) = \frac{(\partial z_x, \partial z_y, -1)}{\sqrt{1 + \partial z_x^2 + \partial z_y^2}} \tag{A.26}$$

where $(\partial z_x, \partial z_y)$ are the partial derivatives of $z(x,y)$ with respect to x and y. $(\partial z_x, \partial z_y)$ can be recovered from $N(P) = (N_x(P), N_y(P), N_z(P))$ using the following

$$(\partial z_x, \partial z_y)(P) = \left(-\frac{N_x(P)}{N_z(P)}, -\frac{N_y(P)}{N_z(P)}\right) \tag{A.27}$$

Surface M can be finally reconstructed by integrating a one-form:

$$z(x,y) = z(x_0, y_0) + \int_\gamma (\partial z_x dx + \partial z_y dy) \tag{A.28}$$

where γ is a planar curve starting at (x_0, y_0) and ending at (x,y). $(x_0, y_0, z(x_0, y_0))$ is a generic surface point of known height $z(x_0, y_0)$. Clearly, if $z(x_0, y_0)$ is unknown, the result will be the actual surface up to some constant depth error.

Unfortunately, errors in surface normal measurements can propagate along the curve γ generating unreliable solutions. For this reason, [155] suggests an alternative height recovery method based on local information only. The more general case where V performs a perspective projection is treated in [149].

[3] $\rho(P)\,N(P)$ has only three degrees of freedom because $N(P)$ is assumed to be normalized.

Shape from shading

Shape from shading algorithm operates only on a single image I, therefore for each image point $(x, y) = V(P)$, we have one equation in three unknowns

$$I(x, y) = \rho(P)\, l(P)\, L(P) \cdot N(P) \tag{A.29}$$

which cannot be solved without imposing additional constraints.

The first attempt to solve Eq. (A.29) was done by Horn in his PhD thesis [63]. Since then, many other solution approaches were developed typically classified into: minimization approaches, propagation approaches, local approaches and linear approaches. For an extensive description of all these methods the reader is referred to [169].

In this chapter we only introduce the minimization approach suggested in [65]. Ikeuchi and Horn [65] reformulated the solution of Eq. (A.29) as the minimization of a cost functional ξ defined as

$$\xi(M) = Bc(M) + \lambda \cdot Sc(M) \tag{A.30}$$

where $Bc(M)$ is the brightness constraint and $Sc(M)$ is the smoothness constraint. The former measures the the total brightness error of the reconstructed image compared with the input image, namely

$$Bc(M) = \int\int \left(I(x, y) - \overline{I}(x, y)\right)^2 dxdy \tag{A.31}$$

where $I(x, y)$ is the input image and $\overline{I}(x, y)$ is the image related to the estimated surface M.

Cost functional $Sc(M)$ penalizes non-smooth surfaces, reducing the degrees of freedom of Eq. (A.29). It is defined as

$$Sc(M) = \int\int \left(\left\| \frac{\partial N}{\partial x}(x, y) \right\|^2 + \left\| \frac{\partial N}{\partial y}(x, y) \right\|^2 \right) dxdy \tag{A.32}$$

Constant λ controls surface smoothness.

In this formulation, $\rho(P)$ is assumed to be known for all $P \in M$ thus, one can add another constraint which imposes normals to be unit. This is what Brooks and Horn [20] did in 1985. The new term was named unit normal constraint and it was defined as follows

$$\int\int \left(1 - \|N(x, y)\|^2\right) dxdy \tag{A.33}$$

The numerical solution is typical achieved using gradient descent algorithms on the Euler-Lagrange equation related to Eq. (A.30) (see section A.4.1 for additional information).

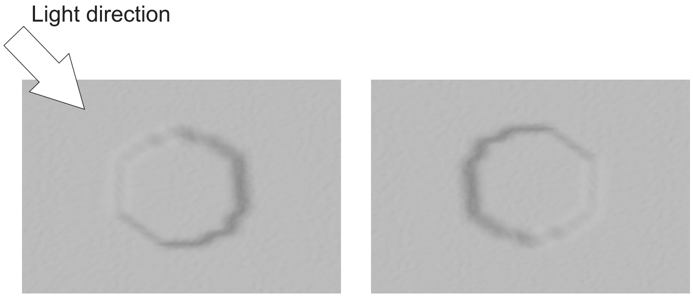


Figure A.7: *An example of the concave/convex ambiguity: it seems that this two images represent two different objects, a concave and a convex one. Nevertheless, the first image is a rotated version of the second one.*

Estimating the light source properties

It can be proven that both photometric stereo and shape from shading become ill-posed problems if light direction, intensity and surface albedo are unknown. This means that a solution may not be unique and it strongly depends on these three parameters[4]. The so-called concave/convex ambiguity, occurring when light orientation is unknown, is a clear example of this ill-posed characteristic. The concave/convex ambiguity refers to the fact that, the same image seems to describe two different objects, one concave and the other convex as shown in Fig. A.7.

More generally, [14] showed that a surface $(x, y, z(x, y))$ is indistinguishable from its *"generalized bas-relief"* (GBR) transformation, defined as

$$\overline{z}(x, y) = \lambda z(x, y) + \mu x + \nu y \tag{A.34}$$

if its albedo and the light properties are unknown. More precisely for all possible values of λ, μ and ν there exists an albedo $\overline{\rho}(x, y)$ and a light $\overline{L}$ such that the brightness image related to the depth map $\overline{z}$ is equal to the one related to z. Moreover, Belhumeur et al. [14] showed that even if self-shadow information is used in addition to shading, the two surfaces $\overline{z}$ and z remain indistinguishable.

Two interesting methods to estimate light direction are due to [74] and [157]. The former recovers the azimuthal angle of the light sources from a single image using texture information. The limit of this approach is the assumption that the textured surface has to be an isotropic gaussian random rough surface with constant albedo.

Instead, [157] use the brightness values of the contour points of the imaged

[4] If we suppose Lambertian surfaces, $\rho(P)$ an $l(P)$ can be grouped together thus, we have only three degrees of freedom.

object in order to estimate light direction by equation Eq. (A.21). Indeed in such points, surface normals can be retrieved knowing that they are perpendicular to the viewing ray connecting these points to the center of projection of the camera.

A.3.3 Shadows

Scene shadows bear a lot of information about the shape of the existing objects. They can give information when no other sources do, indeed shadow regions represent the absence of any other type of information. Methods which exploit this particular visual cue are called either *"Shape form Shadows"* or *"Shape from Darkness"*. They first appear in [140] where shadows were used to relate the orientations of two surfaces. Subsequent works on shadows generally used either the shape of the object casting the shadow in order to constrain the shape of the object being shadowed or vice versa. Indeed, one can infer the shape of an unknown object from the shadows casted on it by a known one. This is the same principle used in structured light projectors with the only difference that the methods based on shadow information use shadow patterns instead of light patterns. The interested reader is sent to [19] for the description of a low cost scanner based on this principle. On the contrary, if an unknown object casts shadows on a known one, for simplicity, let it be a plane, one can use an appropriately modified shape from silhouette method in order to reconstruct its shape. This approach is proposed in [81] in order to avoid segmentation problems implicit in shape from silhouettes methods.

However, in general, shape from darkness methods deal only with the shadow that an object casts on itself, the so-called *self-shadow*. In this case, both the object that casts the shadow and the object being shadowed are unknown as they are all parts of the same unknown surface. Nevertheless, self-shadow can reveal a lot of information. Indeed, let us observe the situation depicted in Figure A.8(a) where p_1 is first shadow boundary points and p_2 is the last one. Knowing their coordinates, one can obtain an upper bound for the shadowed region, i.e., the line η. In other words, a point in such a region cannot be above line η, otherwise it would be a lighted point. Furthermore, all lighted points at the right of p_1 must be above η otherwise they would be shadowed. Thus, η is also a lower bound for the lighted points. Obviously, same results can be obtained if one knows the coordinates of the light source and the coordinates of one of the points p_1 or p_2.

Figure A.8(b) shows a situation similar to the previous one but, in this case, it is assumed that the coordinates of p_1 and p_2 are unknown. Moreover, it is

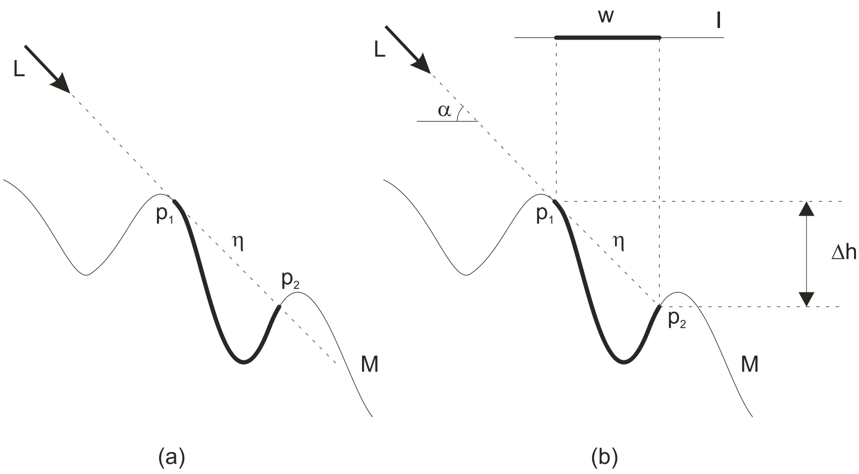


Figure A.8: *Shadowed surfaces: (a) the coordinates of* p_1 *and* p_2 *are assumed to be known; (b)* α *and* w *are known.*

supposed that the camera performs an orthographic projection of the scene and that the light source casts parallel rays of known direction α. This can be obtained by placing both the camera and the light source far away from the scene. The measured shadow width w can be used to estimate the relative height between p_1 and p_2 using the following

$$\Delta h = w \tan(\alpha) \tag{A.35}$$

Moreover, if one assumes that the unknown surface M admits a tangent plane in p_1, such a plane must be parallel to η.

¿From the above considerations, using multiple images taken with different light source positions, one can estimate the unknown surface by constraining a model (e.g. a spline) to fit all the extracted information about relative heights and tangent planes (see [60]).
Furthermore, combining equations of type (A.35) together with the linear inequality constraints related to the various η, one can obtain a set of upper/lower bounds and equations which can be solved by *Linear Programming* algorithms as in [167] or by iterative relaxation methods like in [38].

[145] introduced the concept of *shadowgram*. Let us suppose the situation depicted in Fig. A.9(a) where θ is the angle between the x-axis and the light rays. A shadowgram is a binary function $f(x, \theta)$ recording, for each value of θ, a 0 (black) value at the x coordinates of the shadow points and a 1 (white) value at the x coordinates of the lighted points. Therefore, a shadowgram typically looks like two irregular black stripes of variable thickness. Smith and Kender

[145] demonstrate that the real surface can be reconstructed from the curves representing the discontinuities of the shadowgram $f(x, \theta)$, i.e., the edges of the dark stripes.

The definition of self-shadow consistency follows. Let us assume first that the scene is only illuminated by a point light source positioned at ℓ. Given an object M, the self-shadow generated on M by the light ℓ is the set of all the points on its surface not visible from ℓ. Let $\Theta(M, \ell)$ denote this set. In other words, a generic point P belongs to $\Theta(M, \ell)$ if and only if the segment joining P and ℓ intersects M in at least one point different from P. Therefore, given a calibrated image (I, V), the shadow region generated by ℓ on M and viewed by V is the set of the V-projections of all the points of $\Theta(M, \ell)$ visible from V. Let $\Omega(M, \ell, V)$ denotes this set; then, formally it is

$$\Omega(M, \ell, V) = V(\Theta(M, \ell) \cap \Pi(M, V)) \tag{A.36}$$

where $\Pi(M, V)$ is the set of all the points of M visible from V. Now, given the image I and the estimated shadow regions on I, call them $\omega(I)$, one can say that M is self-shadow consistent with image I if and only if $\omega(I) \subseteq \Omega(M, \ell, V)$. In other words, it is consistent if V does not see shadow points which are not theoretically generated by M. The contrary is not required, since, as we will describe below in this section, in practical situations, only subsets of $\omega(I)$ can be accurately estimated. In this way, the consistent condition is relaxed making consistent surfaces which are not. However, for correctness, one could also estimate $\overline{\omega(I)}$, i.e. the complement of $\omega(I)$, and define that consistency holds when also $\overline{\omega(I)} \subseteq \overline{\Omega(M, \ell, V)}$ holds. Extension to multiple lights is trivial; since, given the set of active lights $(\ell_1, \ldots, \ell_k)$ one can define

$$\Omega(M, L, V) = \bigcup_{\forall \ell_j} \Omega(M, \ell_j, V) \tag{A.37}$$

Besides, consistency for multiple views holds if only if it holds for each singular view. Finally, given an unknown surface Λ and a set of images taken under different lighting conditions, one can build the maximal surface[5] consistent with the extracted shadow information. Let $\Psi(\Lambda)$ denotes this surface, then it is obvious that it contains the actual surface Λ, since Λ is consistent with shadow information.

[5] A maximal surface for a property Q is the surface which satisfied Q and contains every other surfaces that satisfied Q. In order to avoid degeneration, the maximal surface is typically upper bounded.

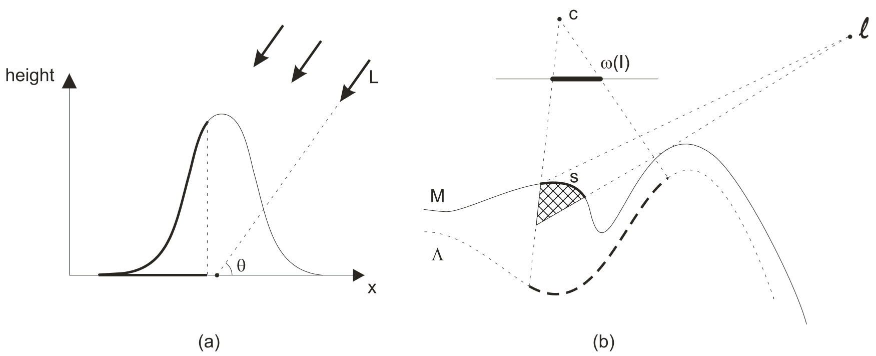


Figure A.9: *(a) Surface to be reconstructed using the shadowgram technique. (b) Conservative shadow carving.*

In [133] a carving approach is proposed to the problem of finding $\Psi(\Lambda)$. The algorithm, called *"Shadow Carving"*, computes first a coarse estimate of the surface using volume carving then it incrementally carves the model removing inconsistencies with self-shadow information. It is known, from section A.3.1 that volume carving computes a volume which certainly contains the original object. The subsequent carving based on shadow cue is performed in a conservative way, i.e., in such a way that the carved model will always contain the actual surface Λ.

Given the situation shown in Fig. A.9(b) where Λ is the actual surface and M is its current estimates. Let (I, V) be a calibrated image, c the center of projection of V and $\omega(I)$ the shadow region on I generated by the light source ℓ. Let us call inconsistent shadow region s, the set of all surface points which are visible from both c and ℓ and such that they project in $\omega(I)$. Savarese et al. [133] proved that the cross-hatched area in Fig. A.9(b) can be removed from M in a conservative way, i.e., obtaining a new estimate that still contains the actual surface Λ.

The major problem of all these algorithms is how to decide whether a surface point P lies on a shadow region or not. This is not a trivial task since it is difficult to distinguish low reflectance points from points in actual shadow regions. The camera only measures radiance coming from some point of the scene. Thus, low radiance measured in a particular direction can be due to a low reflectance (dark textured) point as well as to insufficient illumination. Moreover, insufficient illumination may be due to light sources too far from the object or to an actual shadow region. In the latter case, one must ensure that the shadow is generated by the object itself and not by other objects in the scene. Using only a single image, there is no way to distinguish between these four cases. Furthermore, even if a shadow region is detected, it difficult to accurately extract its boundaries, because shadows, in general, vanish gradually on the surface. Unfortunately, shadow detection plays an important role in reconstruction since small errors can lead to a totally incorrect reconstruction.

Savarese et al. [132] propose a conservative shadow detection method, i.e., a technique which classifies a point as shadow only when it is certain that it is a shadow. The inverse condition is not required so that there can be shadow points classified as non-shadow. Obviously, the more shadow points are detected the more accurate is the reconstruction result. First of all, one must fix a threshold δ which separates light points from dark points. A point P of the surface is *"detectable"* if and only if in at least one picture it appears lighter than δ, otherwise it is *"undetectable"*. This provision ensures that P is not a low reflectance point, but unfortunately, it excludes many points not lighted by the actual light sources. For every image, a point is a shadow point if and only if it is *"detectable"* and it is darker than the threshold δ.

It is finally worth observing that, like shading information, also shadow is subject to the rules of the GBR [76]. Therefore, even if the exact position of the light source is not known, one can reconstruct the observed surface up to a GBR transformation.

A.3.4 Focus/Defocus

There are two techniques to infer depth from a set of defocused images, called *"Shape from Focus"* (SfF) and *"Shape from Defocus"* (SfD). The first one, SfF, acquires a large number of images with small focal settings differences. On the other hand, the second one, SfD, needs only few differently focused images, typically two or three, in order to estimate depth information. In both cases, defocused images are obtained by varying settings like the camera or the lens focal length, the aperture radius or the distance between the object to be acquired and the camera. Afterwards, depth is estimated by comparing the blurriness of different regions in the acquired images.

Both methods are based on the assumption that a defocused image is obtained by convolving the focused one with a kernel h^s_ϕ, called *point spread function* (PSF), that depends on the camera optic ϕ as well as on the scene shape s. Such an assumption comes from the observation that, since pinhole cameras with an infinitesimal aperture are not feasible, each point of the image plane is not

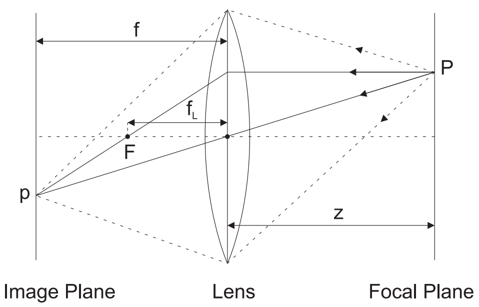


Figure A.10: *Camera with lens: all the light rays coming from a point P in the focal plane are projected into a single point p in the image plane.*

illuminated by a single light ray but by a cone of light rays subtending a finite solid angle. Consequently, these points appear blurry. This effect can be reduced by a proper use of lenses. Indeed, it is well known that in this case, there exists a plane Π, called the *focal plane*, parallel to the retinal plane, the points of which are all in focus, or in other words, each point of Π projects into a single point of the image plane. The situation is shown in Figure A.10, where z is the distance between Π and the center of the lens (the equivalent of the center of projection), f_L is the focal length of the lens and f is the camera focal length defined in section A.2. These quantities are related by the *thin lens equation*

$$\frac{1}{z} + \frac{1}{f} = \frac{1}{f_L} \tag{A.38}$$

Figure A.10 shows that all light rays coming from a point P in the focal plane are projected into a single point p in the image plane. Consequently, an object is perfectly imaged only if it lies exactly on Π, otherwise, it appears blurred. As shown in Fig. A.11, all the rays coming from a point P'' outside the focal plane are projected to a circular region Θ on the image plane.

The image of P'' can be modeled as the integral of the ideal image, where P'' is correctly imaged, weighted by a function (the PSF) which generates the blur effect. Therefore, the relationship between the actual image $\overline{I}$ and the ideal image where all the scene points are correctly imaged I is given by

$$\overline{I}\left(p\right) = \int h^s_\phi\left(p, q\right) I\left(q\right) dq \tag{A.39}$$

If the surface to be acquired is parallel to the focal plane then the PSF can be assumed to be shift invariant, i.e., $h^s_\phi\left(p, q\right) = h^s_\phi\left(p - q\right)$ and Eq. (A.39) can be

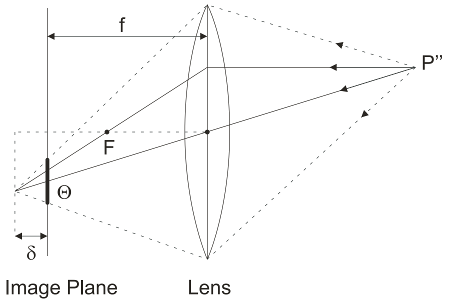


Figure A.11: *All the light rays coming from a point P'' outside the focal plane are projected to a circular region Θ on the image plane.*

rewritten as a convolution

$$\overline{I}\left(p\right) = \int h_{\phi}^{s}\left(p - q\right) I\left(q\right) dq = \left(h_{\phi}^{s} * I\right)\left(p\right) \tag{A.40}$$

As a first approximation, the blur intensity depends on the radius r of Θ, also known as the blurring radius, which is proportional to the distance δ between the actual image plane and an ideal one where P would be correctly imaged (see Fig. A.11). More precisely,

$$r = \frac{\delta R}{f} \tag{A.41}$$

where R is the radius of the lens.

As mentioned above, both SfF and SfD estimate depth from Eq. (A.40). Namely, SfF identifies the regions of the input images where h_{ϕ}^{s} has not been applied, i.e., the in-focus regions. Since h_{ϕ}^{s} is a low pass filter, a defocused region appears poor of high spatial frequency. Furthermore, if the surface to be acquired has high spatial frequency content, i.e., for instance it is a rough surface, a focused region can be recognized by analyzing its local Fourier transform.
The typical approach is to filter each input image $\overline{I}$ with a high pass FIR with impulse response ω and to evaluate the level of blur $v\left(p\right)$ of each point p as

$$v\left(p\right) = \int_{A_{\varepsilon}\left(p\right)} \left(\omega * \overline{I}\right)\left(q\right) dq \tag{A.42}$$

where $A_{\varepsilon}\left(p\right)$ is a neighborhood of p. Once these values are computed for a set of images $\left(\overline{I_1}, \ldots, \overline{I_n}\right)$, shape can be inferred by the following algorithm:

- Let $v_i\left(p\right)$ be the level of blur of the point p of image $\overline{I_i}$
- Let z_i be the depth of the focus plane related to $\overline{I_i}$

- For each point p, find j such that $j = \arg\max \{v_j(p)\}$ (i.e., find the image $\overline{I_j}$ with the sharpest representation of p)
- assign to p depth z_j

For a more precise reconstruction using gaussian interpolation the reader is referred to [108].

SfD methods instead, try to invert directly Eq. (A.40). The difficulty lies in the fact that neither h^s_ϕ nor I are known. Thus, *blind deconvolution* techniques are used in this task. Given a set of blurred images $\left(\overline{I_1}, \ldots, \overline{I_n}\right)$, from Eq. (A.40), one can write

$$\begin{aligned} \overline{I_1} &= h^s_{\phi_1} * I \\ &\vdots \\ \overline{I_n} &= h^s_{\phi_n} * I \end{aligned} \tag{A.43}$$

where ϕ_i is the optical setting used for image $\overline{I_i}$. Many strategies were developed to solve the above ill-posed problem. Classical approaches can be found in [27]. Effective variational and optimization approaches are due to [68] and [48] respectively. In particular, in [49] shape is estimated by inferring the diffusion coefficient of a heat equation.

These methods are widely used in optical microscopy because microscopes have narrow depth of field; therefore, it is easy to obtain pictures containing both blurry and sharp regions.

A.3.5 Picture differences: stereo methods

Algorithms which exploit the differences between two or more pictures of a scene are called *"stereo-matching algorithms"* [93]. They are based on the same process used by human vision system to perceive depth, called *stereopsis*. For this reason, this particular depth cue is typically called stereo information.

In stereo methods, 3D reconstruction is accomplished in two main steps, the first addressing the so-called correspondences (or matching) problem and the second addressing the so-called reconstruction problem. The former recognizes if two or more points belonging to different images are the projection of the same point P of the 3D scene. The latter uses these correspondences in order to estimate the exact position of P in the 3D space.

Reconstruction task is achieved by triangulation. For example, let p_1 and p_2 be a pair of matched points in two different images I_1 and I_2 respectively. Thus, the

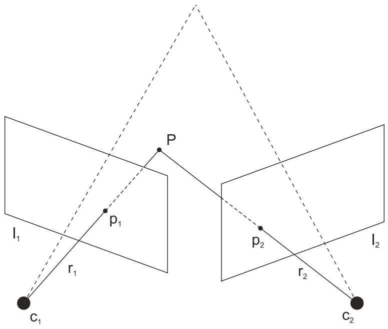


Figure A.12: *From a pair of matched points p_1 and p_2, the 3D coordinates of point P can be computed by triangulation.*

real point P which they refer to, belongs to both the optical rays r_1 and r_2 related to p_1 and p_2 respectively. The situation is schematically depicted in Fig. A.12. Therefore, P must lie at the intersection of r_1 and r_2. In practice, r_1 and r_2 may not intersect due to a imperfect camera calibration or to image discretization errors. The associated depth estimation problem in projective geometry is a linear overconstrained system with three unknowns and four independent equations which can be solved by least squared methods. Details can be found in any computer vision book, for instance in [59].

Stereo methods were widely used in many applications; hence, various versions of these algorithms were developed in order to cope with different types of practical challenges. Recent comparisons of existing stereo matching techniques can be found in [134] and in [138]. A classification of these techniques is not easy because of the number of characteristics to take into account. In the following, stereo methods will be presented according to a basic taxonomy distinguishing them with respect to baselines lengths, number of input images and type of correspondences used. A baseline is a segment connecting the centers of projection of a pair of cameras. Stereo methods which operate with long baselines are called *wide baseline stereo* methods, otherwise they are called *small baseline stereo* methods. Matching problem is different in these two situations. For example, perspective deformations effects can be ignored in the small baseline case but not in the wide baselines case.

Algorithms which use two, three and $n > 3$ images as input are called respectively *binocular stereo*, *trifocal stereo* and *multi-view stereo.* The use of multiple cameras simplifies the reconstruction task reducing errors in the 3D coordinates estimation; moreover, in many situations it eliminates matching ambiguities. In fact, one can use a third camera to check if an hypothetical match is correct or

not.
Binocular stereo stores matching information in a map, called *disparity map*, which associates each pixel of the first input image with the matched pixel of the second input image as follows

$$p_2 = p_1 + d\,(p_1) \tag{A.44}$$

where p_1 and p_2 are the coordinates of the two matched pixels and d is the disparity map.

In a multi-view stereo algorithm the matching and reconstruction tasks are mixed together. Therefore, a disparity map is typically replaced by a complex internal scene representation, such as, a volumetric or a level-sets [47] representation. In particular, using a volumetric representation, reconstruction is achieved by techniques like voxel coloring [137], space carving [77] and max-flow [130, 158]. Space carving applies the above mentioned carving paradigm. In this case, voxels are carved out if they do not project consistently into the set of input images. Therefore, starting from an initial estimate of the surface which includes the actual one, the algorithm finds the maximal surface, called *Photo Hull*, photo-consistent with all the input images. Instead, voxel coloring operates in a single pass through the entire volume of the scene, computing for each voxel a likelihood ratio used to determine whether this voxel belongs to the scene or not.

With respect to the type of correspondences used, an important family of algorithms, called *features based stereo* (FBS), concerns the methods which use image features as stereo information. A feature is a high level data structure that captures some information locally stored in an image. The most used features are edges and corners, but in the literature one can find many other higher order primitives such as regions [34] or topological fingerprints [50]. It is important to note that a feature in the image space does not always correspond to a real feature in the 3D scene.
Restricting matching problem to a small set of a priori fixed features has two big advantages. First of all, features are not affected by photometric variations because they are simple geometrical primitives. Furthermore, since the correspondence search space is highly reduced, the matching task is speeded up. Unfortunately, any feature space gives a discrete description of the scene; thus, reconstruction results in a sparse set of 3D points.

Methods which perform matching between two or more points comparing the regions around them are called *area based stereo* (ABS) methods. These techniques are based on the assumption that given two or more views of the same scene, the image regions surrounding corresponding points look similar. This can be justified by the fact that since corresponding points are the projection of the same point P, their surrounding regions are the projection of the same piece of surface around point P. Therefore, what ABS methods do is to perform matching using only the local reflectance properties of the objects to be acquired.

A formal explanation requires some definitions. Let P be a point of surface M and denote by $A_\epsilon(P) \subset M$ the surface neighborhood of P with radius ϵ. Let (I_1, V_1) and (I_2, V_2) be two calibrated images, assume that P is visible on both images and let $(p_1, p_2) = (V_1(P), V_2(P))$ be a valid correspondence. Therefore, $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ are the projection of $A_\epsilon(P)$ on the image space of I_1 and I_2 respectively. Suppose that the cameras are placed in such a way that the shapes of the image regions $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ look similar, i.e., they are subject to a limited projective distortion. This can be achieved by a pair of parallel cameras with equal up-vectors (see section A.2) and small baseline/depth ratio. In other words, the surface to be acquired has to be far away from the point of views or/and the camera baseline has to be small. Assume that surface M, in $A_\epsilon(P)$, behaves as a pure Lambertian surface. Therefore, the radiance leaving $A_\epsilon(P)$ is independent of the viewpoint. Consequently, the image intensities acquired by the viewpoints V_1 and V_2 in $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ must be equal, up to different camera optical settings (such as focusing, exposure or white balance). More formally, let

$$\begin{aligned} n_1(p_1) &= I_1|_{V_1(A_\epsilon(P))} \\ n_2(p_2) &= I_2|_{V_2(A_\epsilon(P))} \end{aligned} \tag{A.45}$$

be the image intensities around the corresponding points p_1 and p_2, i.e., the restrictions of the images I_1 and I_2 to respectively $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$. Since $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ can be supposed to be equal, images $n_1(p_1)$ and $n_2(p_2)$ are defined in the same domain up to different discretization of the image space. Therefore, one can make a one to one intensities comparison between $n_1(p_1)$ and $n_2(p_2)$ using simple similarity measures such as for example, *Sum of Squared Differences* (SSD), *Sum of Absolute Differences* (SAD) or *Intensity Correlation Distance* (ICD), respectively defined as:

$$\begin{aligned} SSD(p_1, p_2) &= \|n_1(p_1) - n_2(p_2)\|_2 \\ SAD(p_1, p_2) &= \|n_1(p_1) - n_2(p_2)\|_1 \\ ICD(p_1, p_2) &= \langle n_1(p_1), n_2(p_2)\rangle \end{aligned} \tag{A.46}$$

where $\|\cdot\|_1$, $\|\cdot\|_2$ and $\langle\cdot,\cdot\rangle$ are respectively the one-norm, the two-norm and the dot-product in function space. In order to make the above measures invariant to

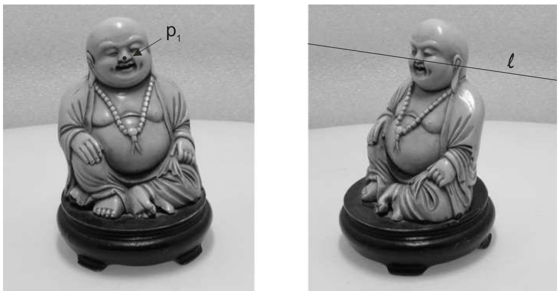


Figure A.13: *Left and right images of a stereo pair: ℓ is the epipolar line associated to p_1.*

camera settings such as white balance and exposure, $n_1(p_1)$ and $n_2(p_2)$ should be replaced by their normalized versions $\overline{n_1}(p_1)$ and $\overline{n_2}(p_2)$, where

$$\overline{n}(p) = \frac{n(p) - \mu}{\sigma} \tag{A.47}$$

with μ the sample mean of $n(p)$ and σ^2 its sample variance.

If the above assumptions were satisfied, one could choose an arbitrary shape for image region $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ and compare them by one of the "metrics"of Eq. (A.46). Usually, square or rectangular shaped windows are preferred since they simplify the computation. Window size plays a crucial role in matching problem. Indeed, small windows are unable to solve matching ambiguities, while large windows make no longer valid the assumption of limited perspective distortion.

In synthesis, given a metric $D(\cdot,\cdot)$, the matching problem is reduced to finding all correspondences (p_1, p_2) such that $D(p_1, p_2)$ is less than a given threshold. Matching task is time expensive since it has to compare each pixel of each image with all the pixels of the other images. However, the knowledge of the calibration parameters can help to restrict the correspondence search space. Indeed, given a scene point P and its projection p_1 on the image I_1, then P certainly belongs to the optical ray r_1 related to p_1 as depicted in Fig. A.12. Ray r_1 starts from the center of projection c_1 of the image I_1 and passes through p_1 in the image plane of I_1. Therefore, if p_2 is the projection of P on the second image I_2, then p_2 must belong to the projection of ray r_1 on I_2, i.e., it must belong to the half-line $\ell = V_2(r_1)$ called the epipolar line associated to p_1 (see Fig. A.13). As a consequence, the correspondence search space related to point p_1 is reduced from a two-dimensional search domain to a one-dimensional one.

In order to improve speed in binocular stereo one may replace the two input images I_1 and I_2 with their rectified versions, i.e., the two equivalent pictures obtained with cameras positioned in such a way to have a common image plane parallel to the baseline and equal up-vectors. Such a process, known as *rectification*, is achieved by projecting the original images I_1 and I_2 into the new image plane. For a comprehensive tutorial on image rectification the reader is referred to [55]. The main characteristic of a rectified image is that its epipolar lines are either all horizontal or all vertical, thus, the search for the correspondences can be performed only along rows or columns. In this case, disparity in Eq. (A.44) can be rewritten as

$$x_2 = x_1 + d\left(x_1, y_1\right), \qquad y_2 = y_1 \tag{A.48}$$

where $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. Consequently, the matching problem is reduced to the following maximization problem

$$d\left(x_1, y_1\right) = -x_1 + \arg\max\left\{D(\left(x_1, y_1\right), \left(x_2, y_1\right)) \mid \forall x_2 \in [1, N_X]\right\} \tag{A.49}$$

where $D\left(\cdot, \cdot\right)$ is a generic similarity metric and N_X is the image width.
Sometimes rectification is used also in multi-view stereo systems with appropriate adjustments.

The physics of the image formation process imposes that each image point has at most one corresponding point in each other image. Therefore, an ambiguity occurs when the solution of the maximization Problem (A.49) is not unique. Such an ambiguity can be solved by adding constraints to the problem, such as surface continuity, disparity bounds or disparity ordering constraint which the scene to be acquired may respect or not. The first type of constraints is obvious while the second says that $d\left(x_1, y_1\right)$ must be less than a given threshold for all possible values of (x_1, y_1). The third imposes that the ordering along the epipolar lines must be preserved. This last one allows one to use *dynamic programming* approaches to the matching problem as in [97].

Computation can be further speeded up if it is organized in a pyramidal structure. In this case, each image is partitioned into different resolution layers (e.g. a Gaussian or a Laplacian pyramid) and the 3D reconstruction is performed at each resolution. At the first iteration, the algorithm runs at the lowest resolution layer creating a first coarse estimate of the surface. At the subsequent stages, the correspondence search interval is restricted using information extracted at the previous layer so that the search is considerably simplified. A detailed account of this method can be found in [98].

Unfortunately, the pure Lambertian assumption for the surface reflectance is too strict for general purpose, indeed objects with constant BRDF are rather rare while surfaces with some kind of specularity are much more common. Therefore, since the radiance reflected by a surface point P changes as a function of the point of view, image intensities $n_1(p_1)$ and $n_2(p_2)$ can be quite different. A typical example is the highlight on a specular surface which moves as the point of view moves. In order to face this problem, one can estimate the object radiance together with its shape as in [69]. Another solution is proposed in [168] which describes a similarity measure invariant with respect to the specularity effects.

Another difficulty in the matching task is due to the fact that it is not always possible to have $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$ within limited projective distortions. Indeed, in general, they are only related by a projective transformation; thus, their shapes can differ in scale, orientation and so on. Sometimes rectification may help to reduce projective distortions. Several techniques were developed to avoid this problem. It is worth recalling the level set method proposed in [47] which uses the current geometry estimate to infer shape and size of the matching windows $V_1(A_\epsilon(P))$ and $V_2(A_\epsilon(P))$. This method iteratively refines the model geometry and performs the match with the estimated windows.

A.4 Multimodal methods

As previously mentioned, multimodal methods reconstruct the shape of an object from more than just one type of information. Since some methods work well in some situations but fail in others, the basic idea of multimodal methods is to integrate information not supplied by one method with that provided by the others. These methods hold the promise of reconstructing a wide range of objects, avoiding the restrictions characterizing individual monomodal methods. Furthermore, the possibility of measuring the same information in different ways allows us to reduce errors typical of specific methods. In short, the characteristics that make these methods superior to monomodal methods, are their robustness and the possibility of acquiring wider ranges of objects.

Unfortunately, the use of more types of information increases algorithmic and time complexity. Indeed, multimodal methods often need a computationally expensive final stage that fuses together all the data extracted and processed in the previous stages. In the literature there exist several ways to combine these data and the specific algorithms depend on the type of data to be fused. For example, [159] proposes a method that combines silhouette and shading information. In particular silhouettes are employed to recover camera motion and to construct the visual hull. This is then used to recover the light source position and finally, the surface is estimated by a photometric stereo algorithm. In [161] a method is described that combines texture and shading cues. More precisely, this latter information is used to solve surface estimation ambiguities of the shape from texture algorithm.

However, most techniques combine multiple cues by classical paradigms like carving or optimization. In particular, as we mentioned before, the carving approach leads to a maximal surface consistent with all the extracted information and certainly including the actual surface. The idea behind multimodal methods based on carving, is to carve all voxels inconsistent with at least one type of information. *"Shadow carving"* and *"Space carving"* are examples of this approach combining respectively shadow and silhouette information and stereo and silhouette information.

On the other hand, the optimization paradigm minimizes a cost functional that takes into account of all the various types of information, delivering as solution a surface fitting the extracted data as much as possible. More formally:

Problem 2 *Given Ω the set of all closed surfaces in $\Re^3$, i.e., the set of all the possible surfaces that can be reconstructed, and $(\alpha_1, \alpha_2, \ldots, \alpha_j)$ a j-tuple, where α_i is information of type i extracted from the input images, the multimodal fusion problem consists in finding M such that*

$$M = \arg\min \{\xi(M) \mid \forall M \in \Omega\} \tag{A.50}$$

where $\xi : \Omega \to \Re$ is the cost functional

$$\xi(M) = \kappa_{int} \cdot \xi_{int}(M) + \sum_i \kappa_i \cdot \xi_i(M, \alpha_i) \tag{A.51}$$

with ξ_{int} a cost functional that penalizes non-smooth surfaces and $\xi_i(\cdot, \alpha_i)$ functionals that penalize surfaces inconsistent with information α_i; κ_{int} and $\kappa_1, \ldots, \kappa_j$ are constants a priori fixed.

Consequently, the solution surface M will be as smooth as possible and consistent with as many data as possible. Constants κ_{int} and $\kappa_1, \ldots, \kappa_j$ balance the impact of the various types of information and the smoothness requirement.

Typically ξ_{int} is related to the mean or to the Gaussian curvature of the surface. For example, it can be defined as

$$\xi_{int} = \int_M \overline{\kappa} ds \tag{A.52}$$

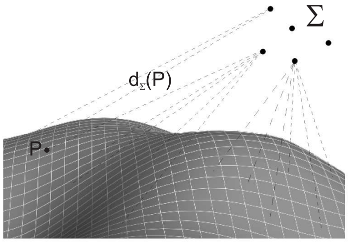


Figure A.14: *In order to evaluate Eq. (A.53), one must measure the distance between Σ and each infinitesimal part of the surface.*

where $\overline{\kappa}$ is the mean curvature.

Functionals $\xi_i(\cdot, \alpha_i)$ instead, depend on the type of information to which they are related. The literature reports many of such functionals accounting for a great variety of visual cues. An interesting functional which penalizes surfaces far from a generic cloud of points Σ is defined as

$$\xi_{cloud}(M) = \left(\int_M d_\Sigma(P)^k ds \right)^{\frac{1}{k}} \tag{A.53}$$

where $d_\Sigma(P)$ is the minimum distance between point $P \in M$ and the points of set Σ (see Fig. A.14). Therefore, Eq. (A.53) can be used as one of the ξ_i, in order to penalize surfaces inconsistent with information extracted, for example, by the stereo-matching algorithm.

Let us observe that Eq. (A.53) accounts for the contribution $d_\Sigma(P)$ of the distance between each $P \in M$ and Σ. Therefore, a surface through empty regions of Σ is bound to have a high value of ξ_{cloud}. Consequently, the solution will be a surface that avoids those regions. This is not always desirable because the empty regions of Σ may be due to actual holes in the object or to failures of the stereo matching algorithm (e.g. in case of dark or poor texture areas).

Several works in the literature address the multimodal fusion problem by an optimization approach. [162] uses both shading and shadow information to reconstruct the lunar surface. [54] fuses together stereo and shading. [58] uses stereo and focus information. [46, 94, 143] fuse stereo and silhouette. [9] combines silhouette, stereo and shadow information.

Problem (2) can be solved in several ways, but, the current trends are the *max-flow/min-cut* and the *deformable models* techniques. Max-flow/min-cut techniques transform the fusion problem into a graph problem where the optimal surface is obtained as the minimum cut solution of a weighted graph. For a recent account see [143]. Instead, deformable models techniques [70, 113] solve

Problem (2) by a gradient descent algorithm on the Euler-Lagrange equation obtained from functional ξ as described in the next section.

A.4.1 Deformable models

A *deformable model* is a manifold deforming itself under forces of various nature. Typically, but not always, these forces make the surface minimize an a priori fixed functional. These forces are classified as internals or externals. The former are generated by the model itself and usually have an elastic nature while the latter depend on the specific problem to solve.

Deformable models appeared for the first time in [70] within the definition of *snake* or *active contour*. A snake is a parametric curve $x(s)$ in the two-dimensional image space that deforms itself maintaining its smoothness and converging to the boundary of a represented object in the image. It is associated to a functional similar to the one of Eq. (A.51) with

$$\xi_{int}(x) = \frac{1}{2}\int_0^1 \left[\alpha \left|x'(s)\right|^2 + \beta \left|x''(s)\right|^2\right] ds \tag{A.54}$$

$$\xi_1(x) = -\int_0^1 \left|\bigtriangledown[G_\sigma * I](x(s))\right|^2 ds \tag{A.55}$$

where $I(x,y)$ is the image intensity function and $G_\sigma(x,y)$ the zero mean bidimensional gaussian function with standard deviation σ. Note that, in this case, the manifold M of Eq. (A.51) is replaced by the snake, $x(s)$, which is a specific parameterization of M.

Since their introduction, deformable models were used in many computer vision tasks, such as: edge-detection, shape modeling [150, 96], segmentation [85, 45] and motion tracking [85, 151]. Actually, in the literature there exist two types of deformable models: the *parametric* (or classical) one [70, 150, 35] and the *geometric* one [23, 24, 113]. The former are the direct evolution of snakes, while the latter are characterized by the fact that their surface evolution only depends on the geometrical properties of the model.

Geometrical framework is based on the level set methods. In this case, the model M is a surface in $\Re^3$, for which there exists a regular function $\psi : \Re^3 \to \Re$ and a constant $c \in \Re$ such that

$$M = \left\{x \in \Re^3 \mid \psi(x) = c\right\} = LevelSet^{\psi}(c) \tag{A.56}$$

In other words, M is the section of level c of a function $\Re^3 \to \Re$ (see Fig. A.15). Besides, the forces are applied to ψ and not directly to M and only when conver-

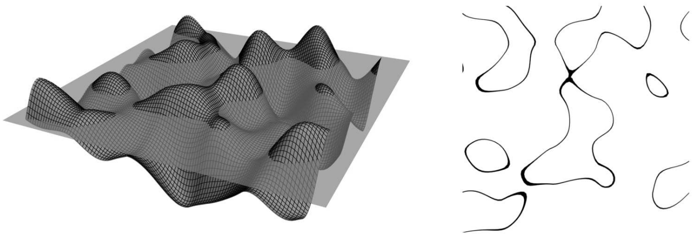


Figure A.15: *Left: representation of LevelSet$^{\psi}(c)$ where $\psi : \Re^2 \rightarrow \Re$. Right: the result of the operation.*

gence is reached, M is computed. Thus, both ψ and M evolve over time according to the partial differential equation

$$\begin{cases} \psi\left(0\right) = \psi_0 \\ \frac{\partial\psi}{\partial t}\left(t\right) = F\left(\psi\left(t\right)\right) \end{cases} \tag{A.57}$$

where $\psi\left(t\right)$ is the function ψ at time t, ψ_0 is its initial state and $F\left(\psi\left(t\right)\right)$ is the force applied to ψ at time t. Hence, since this method operates only on ψ, surface M can dynamically change its topology. Roughly speaking, M can change its number of holes. For example, the reader could imagine to move upwards and downwards the plane of Figure A.15, as the plane moves one obtains sections of ψ with a different number of connected components. Dynamic topology is the key feature that makes the geometrical framework a more powerful tool than the parametric one. The interested reader is sent to [114] for further details.

The remainder of this section is focused on classical deformable model techniques. In this case, in order to solve the minimum problem researchers propose a standard variational approach based on the use of a gradient descent on the Euler-Lagrange equation obtained from functional ξ, which we explain by way of the following example.

Let s be a specific parameterization of M, i.e., s is a function from an open subset $A \subset \Re^2$ to $\Re^3$, and consider the functional

$$\xi\left(s\right) = \kappa_{int} \cdot \xi_{int} + \kappa_{cloud} \cdot \int_A d_\Sigma(s\left(u,v\right))dudv \tag{A.58}$$

where d_Σ is the same as in Eq. (A.53) and

$$\xi_{int} = \int_A \left\|\frac{\partial s}{\partial u}\right\|^2 + \left\|\frac{\partial s}{\partial v}\right\|^2 dudv + \int_A \left\|\frac{\partial^2 s}{\partial u^2}\right\|^2 + \left\|\frac{\partial^2 s}{\partial v^2}\right\|^2 + 2\left\|\frac{\partial^2 s}{\partial v \partial u}\right\|^2 dudv \tag{A.59}$$

where the first term penalizes non-isometric parameterizations of M and the second term is equal to the total curvature of M if s is an isometry, thus penalizing non-smooth surfaces.

The related Euler-Lagrange equation [52] is:

$$-\nabla^2 s(u,v) + \nabla^4 s(u,v) - F_{cloud}(s(u,v)) = 0 \tag{A.60}$$

where $\nabla^2 s$, $\nabla^4 s$ are respectively the laplacian and the bi-laplacian of s and $F_{cloud} : \Re^3 \to \Re^3$ is a field that associates to each point P in the space a unit vector pointing to the point of Σ nearest to P.

The problem of finding M which minimizes ξ has been turned into the problem of finding s, a parameterization of M, which satisfies Eq. (A.60). Therefore, the solution can be computed by a gradient descent algorithm on the following problem

$$\arg\min\left\{\left\|-\nabla^2 s(u,v) + \nabla^4 s(u,v) - F_{cloud}(s(u,v))\right\|, \forall s\right\} \tag{A.61}$$

Consequently, this algorithm can be interpreted as the deformation of a parametric surface s subject to two forces defined as follows

$$\begin{aligned} F_{int} &= \nabla^2 s - \nabla^4 s & \text{(A.62)}\\ F_{ext} &= F_{cloud} & \text{(A.63)} \end{aligned}$$

Let $s(t)$ be the model s at time t, therefore the evolution is described by the following partial differential equation

$$\begin{cases} s(0) = s_0 \\ \frac{\partial s}{\partial t}(t) = \beta \cdot (F_{int} + F_{ext}) \end{cases} \tag{A.64}$$

where s_0 is the initial surface and β determines the evolution speed. In order to find a numerical solution of Eq. (A.64), one can use forward Euler and apply the forces to all the vertices of the mesh. The discrete versions of ∇^2 and ∇^4 on a triangular mesh can be computed using the umbrella $\widetilde{\Delta}$ and the squared umbrella $\widetilde{\Delta}^2$ operators respectively [46].

The advantages and the drawbacks of geometric and parametric deformable models can be summarized as follows. Geometric models have dynamic topology but are not easy to control. Their computation is typically slower than that of parametric models. On the other hand, parametric models have a fixed topology and suffer local minima problems in proximity of concavities. Their computation is faster and by a suitable parameters choice one can also control the parametric characteristics of the final mesh.

A.4.2 Application examples

Multimodal methods, in principle, can use any combination of the visual cues previously seen. Clearly, some combinations can be more effective and manageable than others. This section reviews two multimodal techniques recently proposed in the literature.

A method that combines silhouette and stereo information using classical deformable models is described in [94, 46]. A first estimate s_0 of M is found by volume carving. Starting from s_0, the model evolves subject to three types of forces:

$$\frac{\partial s}{\partial t}(t) = \beta \cdot \left(F_{int} + F_{stereo} + F_{sil}\right) \tag{A.65}$$

where F_{int} is defined as above, F_{stereo} enforces stereo consistency and F_{sil} silhouette information.

In order to avoid local minima problems, [46] defines F_{stereo} as the *gradient vector flow* (GVF) [165] of Σ, that is a vector field solution of a diffusion equation.

Let $P_1, \ldots, P_m$ be the projections (silhouettes) of the real surface Λ viewed by $V_1, \ldots, V_m$ respectively and M be the mesh that currently approximates Λ. Let v be a vertex of mesh M, $F_{sil}(v)$ in [46] is defined as

$$F_{sil}(v) = \alpha(v) \cdot d_{vh}(v) \cdot N(v) \tag{A.66}$$

where $N(v)$ is the surface normal in v, d_{vh} is the signed distance between the visual hull and the projection of vertex v, defined as

$$d_{vh}(v) = \min_j d\left(V_j(v), P_j\right) \tag{A.67}$$

where $d\left(V_j(v), P_j\right)$ is the signed distance between $V_j(v)$ and P_j, i.e., it is positive if v belongs to the visual hull and negative otherwise. $\alpha(v)$ is defined as

$$\alpha(v) = \begin{cases} 1 & if\ d_{vh}(v) \leq 0 \\ \frac{1}{\left(1+d\left(V_c(v), V_c(M)\right)\right)^k} & if\ d_{vh}(v) > 0 \end{cases} \tag{A.68}$$

where $c = \arg\min_j d\left(V_j(v), P_j\right)$ and $V_c(M)$ is the projection of M viewed by V_c. This means that if v is outside the visual hull, $F_{sil}(v)$ is equal to $d_{vh}(v) \cdot N(v)$. Instead, if v is inside the visual hull, $\alpha(v)$ controls the transition of v from a contour point where $d\left(V_c(v), V_c(M)\right) = 0$ to a concave point where $d\left(V_c(v), V_c(M)\right) > 0$. In this way, F_{sil} reduces its intensity as much as v is inside the visual hull and parameter k controls the decreasing factor. Figure A.16 exemplifies the situation.

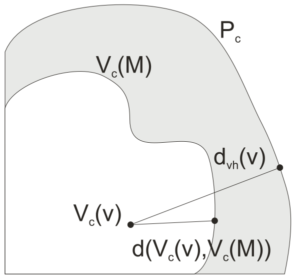


Figure A.16: *Distances involved in F_{sil} computation.*

As we can see in Figure A.17(a) and in Fig. A.17(b), silhouette information cannot describe model concavities which cannot be seen from the acquisition viewpoints, while stereo based methods fail in low variance regions and contours. Silhouette and stereo fusion Fig. A.17(c) makes a better reconstruction of the original surface correcting errors and integrating information missing in each monomodal reconstruction. The final mesh turns out to be smooth and rather uniformly sampled.

An algorithm which combines stereo, silhouette and shadow information using the deformable model framework is proposed in [9]. In particular F_{shadow}, i.e., the force related to shadow information, is defined in a way that minimizes the inconsistency with shadow information. In fact, like in the carving approach, the inconsistent surface portions (for example, portion s in Figure A.9(b)) are pushed inside the surface. More formally,

$$F_{shadow}\left(v\right) = -i\left(v\right) \cdot N\left(v\right) \tag{A.69}$$

where $N\left(v\right)$ is the outer normal to the surface in v and $i\left(v\right)$ is a scalar function equal to 1 if the vertex v is inconsistent with shadow information, and equal to 0 otherwise.

Shadow information can improve the reconstruction obtained from just stereo and silhouette information; indeed, it can describe the shape of the concavities where stereo and silhouette information are missing.

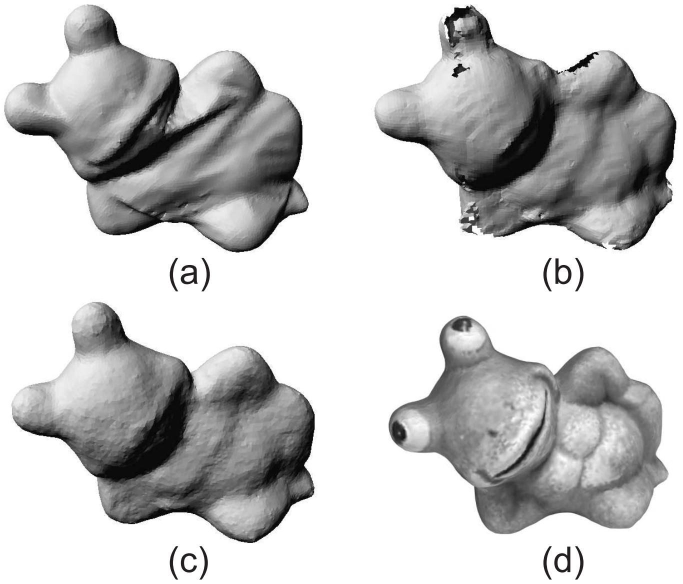


Figure A.17: *Multimodal vs monomodal methods: (a) smoothed model obtained by volume carving; (b) model obtained by fusing together different partial models obtained by stereo matching; (c) model obtained by silhouette and stereo fusion; (d) model obtained by texturing model c.*

Appendix B

Digital Keying

Keying is the most popular technique in visual effects for extracting objects from an image, more precisely, for separating the regions of the image representing foreground objects from the other regions representing background elements.

Differently from *image segmentation* and *background subtraction*, the purpose of keying is to generate a *matte* representing the transparency information about the original image, in such a way that, each pixel is not labeled as belonging or not to a foreground object but, instead, it assumes a continuous value representing how much it is transparent with respect to the background. For this reason, keying is also referred as *matting* [135], [30], [31], [144].

Originally, the term matte refers to the strip of monochrome film that was used in film-making to cover some parts of the original color film strip, so that only parts of the movie were visible. In computer graphics, a matte is a single channel image used to define the transparency of the foreground elements in a composite [164]. The *composition*, in fact, is the direct version of the problem solved by the keying.

According to [119], the composition equation relating the composite image C, the foreground image F, the background image B and the matte α is the following

$$C\left(p\right) = \alpha\left(p\right) F\left(p\right) + \left(1 - \alpha\left(p\right)\right) B\left(p\right) \tag{B.1}$$

where p is a generic point of the composite and $C\left(p\right)$, $F\left(p\right)$, $B\left(p\right)$ are expressed in the same color space. $\alpha\left(p\right)$ instead, is a real value belonging to $[0, 1]$.

The keying aims to recover the values of F, B and α given the observation C. For each single pixel, this results in an under-constrained problem with 7 unknown and 3 equations.

In order to solve this problem, some a priori knowledge about the background and the foreground has to be assumed. $C(p)$, $F(p)$, $B(p)$ and $\alpha(p)$ are usually modeled as stochastic processes and, in particular, the last three are considered independent. Their statistical characteristics can be inferred from the input image and subsequently, given an observation $C(p)$, the value of $\alpha(p)$ can be recovered maximizing a statistical criteria like, for instance, the likelihood.

When dealing with video footage, the previous analysis has to be extended along the time line to the precesses $C(p,t)$, $F(p,t)$, $B(p,t)$ and $\alpha(p,t)$, keeping into account that each of these, is not, in general, independent along the t coordinate. However, most of the keying techniques neglect this fact avoiding the issues that may arise with some hacks.

Literature proposes a wide variety of keying approaches, each providing its advantages and its drawbacks. An excellent classification of these techniques can be found in [30]. Here we focus only on two approaches namely, the difference keying and the chroma keying.

Difference keying presumes that the background is known up to some error modeled, in its turn, as a zero-mean gaussian process. For each point of the image, let's denote with $\overline{B}$ the known background color, with η the error and with B the actual background color, so that $B = \overline{B} + \eta$. Subtracting the known background $\overline{B}$ from the composition color C we obtain

$$C - \overline{B} = \alpha\left(F - \overline{B} + \eta\left(\frac{\alpha - 1}{\alpha}\right)\right) \tag{B.2}$$

Now, if we assumes that F and B differ each other by an amount higher than $\eta\left(\frac{\alpha-1}{\alpha}\right)$, we can state that if $\left(C - \overline{B}\right)$ is equal to zero then α is also equal to zero. In all the other cases, instead, no information about α can be inferred, only that it is different from 0. However $\left(C - \overline{B}\right)$ is often used as matte.

On the contrary, *chroma keying technique* assumes that $C(p)$, $F(p)$, $B(p)$ and $\alpha(p)$ are independently identically distributed stochastic processes and that the distributions of $F(p)$ and $B(p)$ are known and separable. This situation is typically achieved by the use of a solid colored screen placed as background and by ensuring that the foreground objects colors are different from the background color.

Under these assumptions, one can define a surface separating the two color distributions classifying each color as a background or a foreground one. The labeling function related to this surface can be relaxed obtaining a continuous function that can be used as an approximation of the matte α. This approach, however, is still under-constrained therefore, an user has to define manually some parameters of this function, like, for instance, its smoothness.

The chroma keying techniques can be classified according to both the used color space and the allowed shapes of the boundary surface. For instance, the *HLS keying techniques* use the HLS representation of the color while the boundary surfaces are always boxes with edges parallel to the canonical plane. A particular case of HLS keying technique is the *luma keying technique* which surfaces are strips in the luma coordinate. The *3D keying techniques*, instead, use the RGB representation of the color while the boundary surfaces can be spheres, convex polygons or ellipsoids.

Bibliography

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(1):44–58, 2006.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] A. Angelidis and K. Singh. Kinodynamic skinning using volume-preserving deformations. *Proceedings of SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 129–140, 2007.

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rogers, and J. Davis. SCAPE: Shape completion and animation of people. *Proceedings of SIGGRAPH*, 24(3):408–416, 2005.

[5] A. O. Balan, M. J. Black, H. W. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, 2007.

[6] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[7] L. Ballan. Applicazione dei modelli deformabili alla digitalizzazione 3D passiva. *Master thesis, University of Padova*, April 2005.

[8] L. Ballan and G. M. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, Atlanta, GA, USA, June 2008.

[9] L. Ballan and G. M. Cortelazzo. Multimodal 3D shape recovery from texture, silhouette and shadow information. *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2006.

[10] L. Ballan, N. Brusco, and G. M. Cortelazzo. 3D passive shape recovery from texture and silhouette information. In *IEE European Conference on Visual Media Production CVMP*, London, UK, November 2005.

[11] L. Ballan, N. Brusco, and G. M. Cortelazzo. *3D ONLINE MULTIMEDIA AND GAMES: Processing, Visualization and Transmission*, chapter 3D Content Creation by Passive Optical Methods. World Scientific Publishing, 2008.

[12] I. Baran and J. Popovic. Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, 26(3):72, 2007.

[13] B. G. Baumgart. *Geometric Modelling for Computer Vision.* PhD thesis, PhD thesis, Standford University, 1974.

[14] P. Belhumeur, D. Kriegman, and A. Yuille. The bas-relief ambiguity. *Proceedings of IEEE International Conference on Computer Vision*, pages 1060–1066, 1997.

[15] P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements in the computer-generated animated short "tony de peltrie". *Advanced Computer Animation Seminar Notes, Siggraph*, 1985.

[16] P. Besl and H. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[17] J. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH*, pages 192–198, 1977.

[18] J.-Y. Bouguet. Camera calibration toolbox for matlab, 2001. URL `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[19] J.-Y. Bouguet and P. Perona. 3D photography using shadows in dual-space geometry. *International Journal of Computer Vision*, 35(2):129–149, 1999.

[20] M. Brooks and B. Horn. Shape and source from shading. *Proceedings of the International Joint Conference on Artificial Intelligence, Los Angeles*, pages 932–936, 1985.

[21] N. Brusco, L. Ballan, and G. M. Cortelazzo. Passive reconstruction of high quality textured 3D models of works of art. In *6th International Symposium on Virtual Reality, Archeology and Cultural Heritage, VAST*, Pisa, Italy, November 2005.

[22] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Transaction on Computer Graphics*, 22(3), July 2003.

[23] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66(1):1–31, 1993.

[24] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Proceedings 5th Internationa Conference Computer Vision*, pages 694–699, 1995.

[25] E. Catmull. *A subdivision algorithm for computer display of curved surfaces.* PhD thesis, The University of Utah, Salt Lake City, 1974.

[26] J. Chadwick, D. Haumann, and R. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3):243–252, 1989.

[27] S. Chaudhuri and A. N. Rajagopalan. *Depth from Defocus: a real aperture imaging approach.* Springer verlag, 1999.

[28] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[29] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A bayesian approach to digital matting. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2:264–271, 2001.

[30] Y.-Y. Chuang. *New Models and Methods for Matting and Compositing.* PhD thesis, University of Washington, 2004.

[31] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski. Video matting of complex scenes. In *ACM Transactions on Graphics*, pages 243–248, July 2002.

[32] R. Cipolla and P.Giblin. *Visual Motion of Curves and Surfaces.* Cambridge university press, 2000.

[33] H. Cline and W. Lorensen. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–168, 1987.

[34] L. Cohen, L. Vinet, P. Sander, and A. Gagalowicz. Hierarchical region based stereo matching. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 416–421, 1989.

[35] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understand*, 53:211–218, 1991.

[36] F. Cordier and N. Magnenat-Thalmann. A data-driven approach for real-time clothes simulation. *Proceedings on Pacific Conference on Computer Graphics and Applications*, pages 257–266, October 2004.

[37] E. B. Dam, M. Koch, and M. Lillholm. Quaternions, interpolation and animation. Technical report, Department of Computer Science, University of Copenhagen, July 1998.

[38] M. Daum and G. Dudek. Out of the dark: Using shadows to reconstruct 3D surfaces. *Proceedings Asian Conference on Computer Vision, Hong Kong, China*, pages 72–79, 1998.

[39] J. Davis, S. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 428–438, 2002.

[40] A. J. Davison, J. Deutscher, and I. D. Reid. Markerless motion capture of complex full-body movement for character animation. In *Eurographics Workshop on Computer Animation and Simulation*, Manchester, UK, September 2001.

[41] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Rapid animation of laser-scanned humans. In *Proceedings of IEEE Virtual Reality 2007*, pages 223–226, Charlotte, USA, 2007.

[42] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *Proceedings of SIGGRAPH International Conference on Computer Graphics and Interactive Techniques*, page 98, Los Angeles, 2008.

[43] M. de Berg, M. V. Kreveld, M. Overmars, and O. Shwarzkopf. *Computational Geometry*. Springer, 1999.

[44] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 2, pages 315–320, 2001.

[45] R. Durikovic, K. Kaneda, and H. Yamashita. Dynamic contour: A texture approach and contour operations. *Visual Computing*, 11:277–289, 1995.

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

[47] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.

[48] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3): 406–417, 2005.

[49] P. Favaro, S. Osher, S. Soatto, and L. Vese. 3D shape from anisotropic diffusion. *Conference on Computer Vision and Pattern Recognition*, 1:179–186, 2003.

[50] M. M. Fleck. A topological stereo matcher. *International Journal of Computer Vision*, 6(3):197–226, 1992.

[51] D. A. Forsyth. Shape from texture without boundaries. *Proceedings of European Conference on Computer Vision*, pages 225–239, 2002.

[52] C. Fox. *An Introduction to the Calculus of Variations.* Dover Publications, 1987.

[53] P. J. Frey and H. Borouchaki. Surface mesh quality evaluation. *International journal for numerical methods in engineering*, 45:101–118, 1999.

[54] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: combining multi-image stereo shading. *The International Journal of Computer Vision*, 16(1):35–56, 1995.

[55] A. Fusiello, E. Trucco, and A. Verri. Rectification with unconstrained stereo geometry. *Proceedings of the British Machine Vision Conference*, pages 400–409, 1997.

[56] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture - a multi-layer framework. *International Journal of Computer Vision, Special Issue on Evaluation of Articulated Human Motion and Pose Estimation*, 2008.

[57] S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, and L. Van Gool. Articulated multibody tracking under egomotion. In *Proceedings of European Conference on Computer Vision (ECCV).* Springer, October 2008.

[58] I. Gheta, C. Frese, and M. Heizmann. Fusion of combined stereo and focus series for depth estimation. *Workshop Multiple Sensor Data Fusion, Dresden*, 2006.

[59] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge university press, 2000.

[60] M. Hatzitheodour and M. Kender. An optimal algorithm for the derivation of shape from shadows. *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, pages 486–491, 1988.

[61] G. Healey and T. O. Binford. Local shape from specularity. *Computer Vision, Graphics and Image Processing*, pages 62–86, 1988.

[62] D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, February 1983.

[63] B. K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View.* PhD thesis, MIT, 1970.

[64] G. Hua, M.-H. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20–25, San Diego, USA, June 2005.

[65] K. Ikeuchi and B. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[66] D. Jacka, A. Reid, B. Merry, and J. Gain. A comparison of linear skinning techniques for character animation. *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, Afrigraph*, pages 177–186, October 2007.

[67] T. Jebara, A. Azarbayejani, and A. Pentland. 3D structure from 2D motion. *IEEE Signal Processing Magazine*, 16(3):66–84, 1999.

[68] H. Jin and P. Favaro. A variational approach to shape from defocus. *Proceedings of the European Conference on Computer Vision, Part II*, pages 18–30, 2002.

[69] H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo beyond lambert. *Proceedings of IEEE conference on Computer Vision and Pattern recognition*, pages 171–178, 2003.

[70] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[71] L. Kavan, S. Collins, J. Zara, and C. O'Sullivan. Skinning with dual quaternions. *Proceedings on SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 39–46, April/May 2007.

[72] R. Kehl and L. Van Gool. Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, 104(2): 190–209, 2006.

[73] R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:129–136, 2005.

[74] J. J. Koenderink and S. C. Pont. Irradiation direction from texture. *Journal of the Optical Society of America*, 20(10):1875–1882, 2003.

[75] K. Komatsu. Human skin model capable of natural shape variation. *The Visual Computer*, 3:265–271, 1988.

[76] D. J. Kriegman and P. N. Belhumeur. What shadows reveal about object structure. *Journal of the Optical Society of America*, 18(8):1804–1813, 2001.

[77] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):197–216, 2000.

[78] A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[79] M. W. Lee and R. Nevatia. Human pose tracking using multi-level structured models. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 368–381, 2006.

[80] W. Lee, J. Gu, and N. Magnenat-Thalmann. Generating animatable 3D virtual humans from photographs. In *Computer Graphics Forum*, pages 1–10, 2000.

[81] B. Leibe, T. Starner, W. Ribarsky, Z. Wartell, D. Krum, J. Weeks, B. Singletary, and L. Hodges. Toward spontaneous interaction with the perceptive workbench. *IEEE Computer Graphics and Applications*, 20(6):54–65, 2000.

[82] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, 2:164–168, 1944.

[83] M. Levoy, K. Pulli, B. Curless, R. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. *Proceedings of SIGGRAPH Computer Graphics*, pages 131–144, 2000.

[84] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. *Proceedings of SIGGRAPH on Computer graphics and interactive techniques*, pages 165–172, 2000.

[85] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 15:617–634, 1993.

[86] M. Li, M. Magnor, and H. Seidel. Hardware-accelerated visual hull reconstruction and rendering. *Proceedings of Graphics Interface*, pages 65–72, 2003.

[87] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[88] L. Lucchese and S. K. Mitra. Color image segmentation: A state of the art survey. *Proceedings of the Indian National Science Academy*, 67(2): 207–221, 2001.

[89] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 3–19, London, UK, 2000. Springer-Verlag.

[90] N. Magnenat-Thalmann, R. Laperriere, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. *Proceedings on Graphics interface*, pages 26–33, 1988.

[91] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[92] D. Marquardt. An algorithm for least-squares estimation of non-linear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 11: 431–441, 1963.

[93] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings Royal Society of London*, 204:301–328, 1979.

[94] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualiziation for 3D video. *Computer Vision and Image Understanding*, 96(3):393–434, 2004.

[95] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. *Proceedings of 12th Eurographics Workshop on Rendering*, pages 116–126, 2001.

[96] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Comput. Med. Imag. Graph.*, 19: 69–83, 1995.

[97] G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47:275–285, 2002.

[98] C. Menard and N. Brandle. Hierarchical area-based stereo algorithm for 3D acquisition. *Proceedings International Workshop on Stereoscopic and Three Dimensional Imaging, Greece*, pages 195–201, 1995.

[99] P. Mendoca and R. Cipolla. A simple technique for self-calibration. *Proceedings of IEEE Conference on Computer Vision and Pattern recognition*, 1:500–505, 1999.

[100] C. Ménier, E. Boyer, and B. Raffin. 3D skeleton-based body pose recovery. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), Chapel Hill*, June 2006.

[101] B. Merry, P. Marais, and J. Gain. Animation space: A truly linear framework for character animation. *ACM Transactions on Graphics*, 25(4):1400–1423, October 2006.

[102] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and motion capture using voxel data. In *Proceedings of the International Workshop on Articulated Motion and Deformable Objects*, pages 104–118, 2002.

[103] J. Mitchelson and A. Hilton. Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling. In *Proceedings of British Machine Vision Conference (BMVC)*, 2003.

[104] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[105] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006. ISSN 1077-3142.

[106] L. Mundermann, S. Corazza, and T. P. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, June 2007.

[107] H.-H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):5974, May 1988.

[108] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, 1994.

[109] B. Neumann. Natural language description of time-varying scenes. *In Semantic Structures-Advances in Natural Language Processing*, pages 167–206, 1989.

[110] F. Nicodemus. Reflectance nomenclature and directional reflectance and emissivity. *Applied Optics*, 9:1474–1475, 1970.

[111] K. Ogawara, X. Li, and K. Ikeuchi. Marker-less human motion estimation using articulated deformable model. *IEEE International Conference on Robotics and Automation*, pages 46–51, April 2007.

[112] J. O'Rourke and N. I. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2(6):522–536, November 1980.

[113] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, 2003.

[114] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[115] F. I. Parke. *A parametric model for human faces*. PhD thesis, The University of Utah, Salt Lake City, 1974.

[116] B. T. Phong. Illumination for computer generated images. *Communications of the ACM*, 18(6):311–317, 1975.

[117] M. Piccardi. Background subtraction techniques: a review. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 4: 3099–3104, 2004.

[118] R. Plänkers and P. Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1182–1187, 2003.

[119] T. Porter and T. Duff. Compositing digital images. *Computer Graphics*, pages 253–259, July 1984.

[120] M. Potmesil. *Introduction to statistical pattern recognition.* Academic press, 1990.

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

[122] B. Prescott and G. McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1):39–47, 1997.

[123] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 467–474, June 2003.

[124] D. Ramanan and C. Sminchisescu. Tranining deformable models for localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–22, New York, USA, June 2006.

[125] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA, June 2005.

[126] F. Remondino. 3D reconstruction of static human body with a digital camera. In *Videometrics Conference, SPIE*, pages 38–45, 2003.

[127] F. Remondino, N. D'Apuzzo, G. Schrotter, and A. Roditakis. Markerless motion capture from single or multi-camera video sequence. In *Proceedings of International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, pages 8–12, December 2004.

[128] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, volume 1, pages 824–831, 2005.

[129] M. Rioux, F. Blais, A. Beraldin, G. Godin, P. Blulanger, and M. Greenspan. Beyond range sensing: Xyz-rgb digitazing and modeling. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA*, pages 111–115, 2000.

[130] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *Proceedings of IEEE International Conference on Computer Vision*, pages 492–502, 1998.

[131] H. Rushmeier and F. Bernardini. The 3D model acquisition pipeline. *Computer Graphics Forum*, 2(2):149–172, 2002.

[132] S. Savarese, H. E. Rushmeier, F. Bernardini, and P. Perona. Shadow carving. *ICCV*, pages 190–197, 2001.

[133] S. Savarese, M. Andreetto, H. Rushmeier, F. Bernardini, and P. Perona. 3D reconstruction by shadow carving: Theory and practical evaluation. *International Journal of Computer Vision*, 71(3):305–336, 2007.

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

[135] C. Schultz. Digital keying methods, September 2006.

[136] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20:151–160, 1986.

[137] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 38(3):197–216, 2000.

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

[139] H. Seo, Y. I. Yeo, and K. Wohn. 3D body reconstruction from photos based on range scan. In *Edutainment*, pages 849–860, 2006.

[140] S. Shafer and T. Kanade. Using shadows in finding surface orientations. *Computer Vision, Graphics and Image Processing*, 22(1):145–176, 1983.

[141] L. Sigal and M. J. Black. HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[142] K. Singh and E. Kokkevis. Skinning characters using surface oriented free-form deformations. *Proceedings of the Graphics Interface Conference*, pages 35–42, May 2000.

[143] S. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. *Proceedings of IEEE International Conference on Computer Vision*, 1:349–356, 2005.

[144] A. R. Smith and J. Blinn. Blue screen matting. In *SIGGRAPH*, pages 259–268, August 1996.

[145] E. Smith and J. Kender. Shape from darkness: Deriving surface information from dynamic shadows. *In AIII*, pages 664–669, 1986.

[146] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parameterization. In *Proceedings of the conference on Visualization*, pages 355–362, Washington, DC, USA, 2002. IEEE Computer Society.

[147] J. Starck and A. Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, ICCV*, pages 15–21, October 2005.

[148] A. Sundaresan and R. Chellappa. Markerless motion capture using multiple cameras. In *Proceedings of Computer Vision for Interactive and Intelligent Environment*, pages 15–26, November 2005.

[149] A. Tankus and N. Kiryati. Photometric stereo under perspective projection. *Proceedings of IEEE International Conference of Computer Vision*, pages 611–616, 2005.

[150] D. Terzopoulos and K. Fleischer. Deformable models. *Visual Computing*, 4:306–331, 1988.

[151] D. Terzopoulos and R. Szeliski. Tracking with kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–20. Eds. Cambridge, MA, MIT Press, 1992.

[152] C. Theobalt, J. Carranza, M. A. Magnor, and H.-P. Seidel. Combining 3D flow fields with silhouette-based human motion capture for immersive video. *Graphical Models*, 66(6):333–351, 2004.

[153] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal Of Optical Society Of America*, 57:1105–1114, 1967.

[154] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(5):323–344, 1987.

[155] O. Vega. Default shape theory: with applications to the recovery of shape and light source from shading. *Master's thesis, University of Saskatchewan, Computational Sciecne Department*, pages 1474–1475, October 1991.

[156] D. Vlasic, I. Baran, W. Matusik, and J. Popovic. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3), 2008.

[157] G. Vogiatzis, P. Favaro, and R. Cipolla. Using frontier points to recover shape, reflectance and illumination. *Proceedings of IEEE International Conference on Computer Vision*, pages 228–235, 2005.

[158] G. Vogiatzis, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 391–398, 2005.

[159] G. Vogiatzis, C. Hernández, and R. Cipolla. Reconstruction in the round using photometric normals and silhouettes. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 1847–1854, 2006.

[160] X. C. Wang and C. Phillips. Multi-weight enveloping: Least-squares approximation techniques for skin animation. *Proceedings of the SIGGRAPH/Eurographics symposium on Computer animation*, pages 129–138, 2002.

[161] R. White and D. A. Forsyth. Combining cues: Shape from shading and texture. *Conference on Computer Vision and Pattern Recognition*, 2:1809–1816, 2006.

[162] C. Wohler. 3D surface reconstruction by self-consistent fusion of shading and shadow features. *Proceedings of 17th International Conference on Pattern Recognition*, 2:204–207, 2004.

[163] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.

[164] S. Wright. *Digital Compositing for Film and Video*. Focal Press, May 2006.

[165] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, pages 359–369, 1998.

[166] J. Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 712–719, Washington, DC, USA, 2006.

[167] D. K.-M. Yang. Shape from darkness under error. *Ph.D. thesis, Columbia University*, 1996.

[168] R. Yang, M. Pollefeys, and G. Welch. Dealing with textureless regions and specular highlights - a progressive space carving scheme using a novel photo-consistency measure. *Proceedings of the 9th International Conference on Computer Vision*, pages 576–584, 2003.

[169] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8): 690–706, 1999.

[170] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(9), 2004.

Parte di questa tesi è
stata possibile grazie alla
Fondazione Ing. Aldo Gini.
Uno speciale ringraziamento va inoltre
a Marco Toninato e a Stefano Cenci
per aver contribuito alla realizzazione dei sistemi.