3D Object Reconstruction using Computer Vision: Reconstruction and Characterization Applications for External Human Anatomical Structures

Teresa Cristina de Sousa Azevedo

BSc in Electrical and Computer Engineering by
Faculdade de Engenharia da Universidade do Porto (2002)

MSc in Biomedical Engineering by
Faculdade de Engenharia da Universidade do Porto (2007)

Thesis submitted for the fulfilment of the requirements for the
PhD degree in Informatics Engineering by
Faculdade de Engenharia da Universidade do Porto

Supervisor:
João Manuel R. S. Tavares
Associate Professor of the Department of Mechanical Engineering
Faculdade de Engenharia da Universidade do Porto
Co-supervisor:
Mário A. P. Vaz
Associate Professor of the Department of Mechanical Engineering
Faculdade de Engenharia da Universidade do Porto

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Prof. João Manuel R. S. Tavares and Prof. Mário A. P. Vaz, for the opportunity to undertake this research and for their thoughtful guidance throughout.

I should like to thank all my friends and colleagues at *LOME – Laboratório de Óptica e Mecânica Experimental*, from *INEGI – Instituto de Engenharia e Gestão Industrial*, for always being present when I needed help.

I would also like to thank the support of the PhD grant SFRH/BD/27716/2006 from *Fundação para a Ciência e a Tecnologia*, in Portugal.

Finally, my heartfelt thanks go to my husband Igor Terroso, close relatives and friends who have always given me considerable support and encouragement.

ABSTRACT

This PhD project belongs to the Computer Vision domain, more specifically, to the area of Three-Dimensional (3D) Vision. The 3D information of the surrounding environment perceived by human eyes is one of the most important kinds of information offered by sight. Thus, a large effort has been developed by the scientific community throughout time in the development of methods that make possible the acquisition of a scene (or object) using techniques of Computer Vision.

In this Thesis, computational methods were developed and implemented to obtain 3D models of the human body from image sequences. The methods contain several stages, in which the image pixel information is transformed into 3D spatial coordinates. These images are acquired either by moving the camera or the object to be reconstructed, in order to obtain accurate 3D models without imposing severe restrictions, on either the movement involved or on the illumination conditions.

The first method employed adopts the Stereo Vision approach, in which a depth map of scene objects is recovered. The second method is based on a volumetric approach for Shape-from-Silhouettes, which constructs a 3D shape estimate (known as Visual Hull) of an object using its calibrated silhouettes. The developed methodologies were assessed using static objects with dissimilar textures, shapes and sizes, including real human body structures.

The Stereo Vision based method proved to perform better on objects with strong features, easy to extract and correctly match throughout the input image sequences. However, on objects with a smooth surface, with almost no colour or texture variation, an accurate detection and matching of strong features between image sequences was very difficult to attain. Consequently, the calculated epipolar geometry was incorrect, which led to disparity map of poor quality.

For the second method, tests were performed using man-made objects and real human body parts. Generally, the method had no problem to reconstruct objects with smooth surfaces or with complicated morphology. The silhouettes do not need to be perfectly extracted due to the conservative quality of the method. However, this method puts some restrictions, such as backgrounds with low colour variation, suitable calibration apparatus and restrictions on the positions from which the images are acquired.

RESUMO

Esta Tese de Doutoramento insere-se na área da Visão Computacional; nomeadamente, no domínio da Visão Tridimensional (3D). Dados 3D sobre o meio ambiente é uma das mais importantes fontes de informação fornecidas pela visão humana. Assim sendo, tem-se assistido a um grande investimento por parte da comunidade científica, no desenvolvimento de metodologias computacionais para a obtenção da forma 3D de cenas (ou objetos) usando técnicas de Visão Computacional.

No âmbito desta Tese, foram desenvolvidos e implementados métodos computacionais de construção de modelos 3D do corpo humano a partir de sequências de imagens. Os processos de reconstrução desenvolvidos são compostos por várias etapas, nas quais a informação de cada *pixel* das imagens é transformada em coordenadas espaciais 3D. As imagens são obtidas com movimento relativo entre a câmara e o objeto a reconstruir, e de forma a obter modelos 3D precisos, sem impor restrições severas, quer ao nível do movimento quer nas condições de iluminação.

O primeiro método desenvolvido adota a estratégia de Visão Estéreo, na qual é recuperado um mapa de disparidade dos objetos numa cena. O segundo método é baseado numa versão volumétrica do método *Shape-from-Silhouettes*. Este método constrói uma forma 3D aproximada do objeto (conhecida por Envoltória Visual) usando as suas silhuetas previamente calibradas. Ambos os métodos desenvolvidos foram analisados e validados usando objetos estáticos, com diferentes texturas, formas e tamanhos, onde se incluíram estruturas reais do corpo humano.

O método baseado em Visão Estéreo obteve melhores resultados em objetos com suficientes pontos característicos fortes, fáceis de extrair e emparelhar corretamente numa sequência de imagens. Contudo, nos objetos cujas superfícies apresentam transições suaves, ou com pouca variação de cor ou textura, a correta deteção e emparelhamento de pontos característicos entre imagens provou ser bastante difícil. Consequentemente, a geometria epipolar é incorretamente determinada, levando à obtenção de um mapa de disparidade de fraca qualidade.

O segundo método foi analisado usando objetos fabricados e partes do corpo humano reais. Na maior parte dos casos, o método volumétrico obteve bons resultados mesmo em

objetos sem pontos característicos fortes ou com morfologias complexas. As silhuetas obtidas não necessitaram de ser de elevada qualidade devido à propriedade conservativa do método volumétrico utilizado. No entanto, este método coloca algumas restrições, nomeadamente fundos com pouca variação de cor, métodos de calibração adequados e restringe as posições a partir das quais as imagens podem ser adquiridas.

Table of Contents

List of Figures

List of Tables

1
Introduction

In the last decades, as Computers and Imaging Devices became more powerful, attention is being focused on the building of high quality three-dimensional (3D) models. 3D models of real objects are used in a great variety of applications, ranging from cultural heritage, medical 3D imagery, forensic applications, architecture or entertainment.

Although 3D models generated by computational systems have been an intensive and long-lasting research problem in the Graphics and Computer Vision scientific communities, fully automated building of 3D models is still a non-trivial task.

The main objective of this Thesis was to address the automatic image-based 3D reconstruction of objects, in particular, external anatomical structures of the human body without severe restrictions on the image acquisition setup.

One of the key challenges for 3D content production is to build visually realistic models of humans. Since the scientific field of Computer Vision is concerned with developing computational theories and methods for automatic extraction of useful information from images, it offers the opportunity to build these 3D models directly from real-world scenes with visual realism and accuracy, but many difficulties still to be overcome.

This Thesis presents and discusses methods for attaining the 3D shape reconstruction of human body parts from images. The used images can be acquired either by using several cameras at the same time or just one camera at different locations.

1.1 Motivations and objectives

Despite the number of successful applications for 3D reconstruction of human body parts, the construction of a fully automated system remains unobtainable. Main reason for the ill-posed problem that is visual 3D reconstruction is its inverse formulation: it tries to describe a 3D object projected into one or more 2D images and to reconstruct its properties, such as shape and texture.

The image-based reconstruction process involves several steps, and some of these are not yet fully automated or reliable for the generation of accurate 3D models. Among these, the automated markerless estimation of the camera's pose is one of the most attractive and complex research topics in Computer Vision. Nowadays, human interaction is often required, for manual measurements or when using markers or targets.

Furthermore, obtaining an accurate 3D geometric human model that can be used to synthesise realistic novel views of an object is highly desirable in many areas, mainly in Medicine and Engineering, which motivates the continually searching for computational methods to attain such the 3D realistic models in an efficient manner. Moreover, the geometry of human anatomical structures is very complex and accurate 3D reconstruction is important for morphological studies, finite element analysis and rapid prototyping which are common practices in several fields. Although magnetic resonance imaging, computed tomography and laser scanners can be used for reconstructing biological structures, the cost of the equipment is fairly high and specialised technicians are required to operate the equipment, making such approaches limiting in terms of accessibility.

Thus, the ultimate goals of this research were to develop, implement and compare methods for full 3D reconstruction of human external structures from images, addressing some important requirements, such as ease of use, cost effectiveness, flexibility and reliability, mainly in the steps concerning the camera calibration and the 3D information determination. Additionally, an overall automated process for the 3D shape reconstruction of human body parts should be attained. Hence, this Thesis carried out with the following main objectives:

1. investigate if automated markerless camera pose estimation is feasible and under which conditions;
2. study the possibility of recovering complete and detailed 3D models of complex objects using fully automated procedures;
3. explore which kind of (3D) information can be retrieved from acquired images;
4. study the capabilities and limits of image-based methods in dealing with uncalibrated images;
5. comparison between the developed and implemented 3D reconstruction methods, with the identification of the advantages and drawbacks of each one.

On an initial stage of this Thesis, the stereo reconstruction method was studied in detail. This method is based on stereo reconstruction, in which the relative orientation of two images is determined by the epipolar geometry, and then the object can be reconstructed up to a spatial similarity transformation if the images are calibrated or up to a 3D projective transformation if the images are not.

Due to the requirements and limitations of the stereo reconstruction method to address objects with lack of texture, on a second step of this Thesis, volumetric reconstruction methods were studied exhaustively. Typically, the volumetric reconstruction approaches assume a discrete and bounded 3D space containing the object to be reconstructed. In this Thesis, the initial reconstruction volume is divided into voxels and the task is to correctly classify the set of voxels that represent the object involved. The algorithm developed and implemented during this Thesis requires a set of calibrated input images, which was achieved by an off-line method: Zhang's camera calibration. The algorithm also requires some additional classification of the pixels into background and foreground classes: image segmentation.

1.2 Major contributions

This Thesis sets out to solve an ill-posed problem in an effective, cost-saving and practical manner. Hence, it addresses the several difficulties in the development and

implementation of 3D human shape acquisition methods, and presents and evaluates solutions for such problems.

The main contributions of this Thesis are the following:

- development and evaluation of the 3D reconstruction method based on stereo vision, with analysis of its major advantages and drawbacks;
- development and evaluation of a volumetric method for 3D reconstruction, with analysis of its major advantages and drawbacks;
- new hierarchical representation of the 3D space via based on an octree data structure implementation, allowing for a more efficient 3D volumetric reconstruction, both in terms of computation time and memory space;
- combination of silhouettes and texture information, providing for a further refinement of the obtained 3D model from volumetric methods, in which the texture information is introduced as a final consistency check for all surface voxels of the octree model;
- development and implementation of new photo-consistent testing method based on both statistical analysis of the voxel projection and voxel visibility;
- automatic initial bounding box determination for the volumetric reconstruction method, allowing for an automated visual hull computation and a faster approximation of the initial 3D volume with the real 3D object shape;
- implementation of the Zhang's method for camera calibration into two separate steps, allowing an independent and more accurate determination of the intrinsic and extrinsic parameters;
- development of a new chessboard calibration pattern, allowing for a correct and automated estimation of the calibration pattern orientation;
- revision and comparison of unsupervised image segmentation methods based on skin colour detection;
- development and implementation of an image segmentation method that combined both skin colour and contour information, in which the skin-coloured regions are processed using the homogeneity property of the human external shapes reducing many false negative and positive skin extraction; assessment of

the obtained experimental results from both implemented 3D reconstruction methods;

- detailed assessment of 3D models obtained using both implemented 3D reconstruction methods;
- critical evaluation of the experimental results obtained using the volumetric method, comparing them with ground truth data when available or with 3D models built using a commercial laser scanner;
- determination of some subjective (e.g. texture colour, shape) and analytical characteristics from a voxelized 3D model (e.g. volume, size);
- publication of the following articles, whether in scientific journals or conferences, in order to disseminate the work achieved during this project: [Azevedo, 2008b], [Azevedo, 2008a], [Azevedo, 2009], [Azevedo, 2010].

1.3 Outline of the Thesis

After this introduction, in Chapter 2 the existing methods for 3D reconstruction, particularly designed for human anatomical structures, are reviewed. This Chapter assumes particular importance as it presents the theoretical fundaments and framework in 3D reconstruction from images. An attempt was made at reporting the most important and promising methods available at the moment for human 3D reconstruction.

Chapter 3 discusses the problem of camera calibration, which allows modelling the transformation between 3D world points onto their projections onto a 2D image plane. Initially, we present the general concepts on the image formation model and coordinate transformations imposed by a camera. Subsequently, we present some of the most commonly used calibration methods, by classifying them into traditional and auto-calibration methods, with an accent on Zhang's calibration method, used on the developed volumetric reconstruction method.

The first stereo-based reconstruction method developed in this Thesis is reported on Chapter 4. The main steps involved in the stereo reconstruction are: extraction and matching of feature points, epipolar geometry computation, rectification of the stereo image pairs and dense matching. These steps are described and discussed in this Chapter. Chapter 4 also

describes the developed volumetric-based reconstruction method. Based on Shape-From-Silhouettes (SFS) methods, we use a multi-resolution representation of the world space based on octal decomposition of a 3D voxel set. Optionally, we combine visual-hull with photo-hull computation, allowing for the generated 3D models to be refined using photo-consistency tests.

For both implemented reconstruction methods, some qualitative and quantitative results are presented and discussed on Chapter 6.

Finally, Chapter 7 draws the main conclusions and discusses ideas of possible future works.

2
3D Reconstruction

2.1 Introduction

In the 1950s, digital computers became available in research laboratories to process various types of data. By the mid 1950s, it was understood that computers could be used to process digital images. Earliest computer applications on digitized images were simple image processing methods, like image enhancement or image restoration [Rosenfeld, 2001].

First researches on Computer Vision began in the late 1960s, with images of scenes containing simple geometrical objects (the "blocks world"). The first PhD Thesis on this subject was in 1963, [Roberts, 1963]. However, it was only in the 1980s, with the Marr's paradigm [Marr, 1982], that the scientific community finally started to give prominence to Computer Vision research.

Computer Vision is very similar to biological vision, where visual stimuli are transformed into visual perception. In biological vision, human and animal visual perception is studied, resulting in models of how these systems operate in terms of physiological processes. Analogously, Computer Vision conceives artificial vision systems, which are implemented in software and/or hardware. Knowledge exchange between biology and Computer Vision has proven to be extremely fruitful for both fields [Fischler, 1987].

In 3D computational reconstruction, the input data can be acquired by several means, such as image cameras or scanner devices. Since most 3D Computer Vision algorithms require considerable computational and imaging resources, there is always a trade-off between hardware, software, processing speed, accuracy and realism of the results.

In the last decades, the explosive growth in computer's processing power and memory storage and its continuously reducing price, has enabled the common use of 3D Computer Vision solutions in a variety of application fields:

- Security systems, Fig. 2.1a: for instance, in visual surveillance (e.g. [Suresh, 2007], [Haering, 2008], [Grabner, 2010]), to detect human actions (e.g. [Bouchrika, 2007], [Hofmann, 2012], [Tran, 2012]) or even for people identification, etc.
- Recognition: identification of objects or features present in images, like faces (e.g., [Chen, 2007], [Lin, 2011], [Harvey, 2012]) or pedestrians (e.g. [Conci, 2009], [Holley, 2009], [Lindner, 2012], [Andriluka, 2010]);
- Industry, Fig. 2.1b: for instance, in non-destructive quality and integrity inspection, on-line measurements and production line control;
- Navigation systems, Fig. 2.1c: for example, in autonomous robot navigation (e.g. [Do, 2005], [Mishra, 2011]), obstacle detection or traffic analysis (e.g. [Bertozzi, 2012], [Meier, 2012]);
- Virtual reality, Fig. 2.1d: such as to build virtual actors, objects or environments, and augmented/mixed reality (e.g. [Zuo, 2011], [Corral-Soto, 2012]);
- Medical imaging, Fig. 2.1e: for example, in anthropometric studies, detection of tumours or other malformations, design and manufacturing of prosthetic devices and surgery planning (e.g. [Qiang, 2007], [Jacobs, 2011], [Wang, 2012]);
- Architecture/archaeology, Fig. 2.1f: as for 3D architectural modelling and reproduction of archaeological artefacts (e.g. [Fassi, 2007], [Kleber, 2009], [Paliokas, 2010]).

The available methods for 3D reconstruction of objects are usually classified into contact or non-contact methods [Cheng, 1998], Fig. 2.2.

Fig. 2.1 – Some industrial applications of 3D Computer Vision: (a) *ExitSentry* – airport security [ExitSentry, 2012]; (b) *Cognitens WLS400* – industrial 3D inspection [Cognitens, 2012]; (c) *Flexiroad* – traffic analysis [Flexiroad, 2012]; (d) *Virtual Iraq/Afghanistan* – virtual reality [Virtual, 2012]; (e) *ClaroNav* – surgery planning [ClaroNav, 2012]; (f) *PhotoModeler* – architectural modelling [PhotoModeler, 2012].

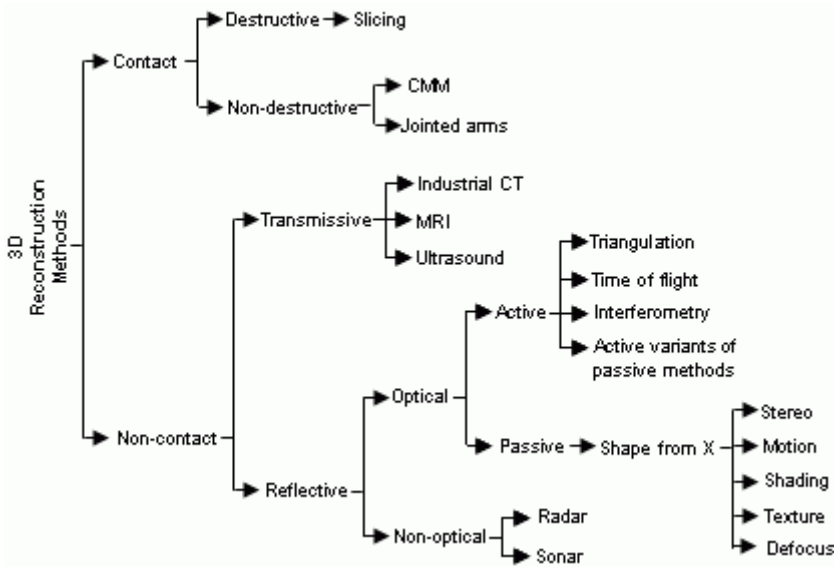

Fig. 2.2 – Usual classification of 3D reconstruction methods.

Contact-based methods probe the object to be reconstructed through physical touch, Fig. 2.3. They can achieve high accuracy rates and are suitable for a wide range of applications. However, as these methods involve mechanical movement from one measurement point to the next, they collect a sparse data set from the object to be reconstructed making possible the unmeasured of relevant regions. Also, the act of scanning an object by touching can modify its shape or even originate damages, in particular, if the objects are soft and delicate.

Nowadays, a 3D model can also be obtained by using non-contact optical methods, which are usually divided into two main groups [Seitz, 1999]:

1. Active: methods that use the projection of some sort of energy (such as structured light or ultra-sounds) or motion between the object and the imaging system, Fig. 2.4;
2. Passive: methods that do not require energy projection or relative motion and work under ambient illumination.

Most common non-contact methods use image data, range sensors or a combination of both. Image-based methods are widely used; in particular, for industrial applications, architectural objects or for precise terrain and city modelling. These methods need that relevant features can be identified in the input images, which sometimes is impossible due to occlusion cases, lack of texture or significant features on the objects' surfaces [Grün, 2003; Remondino, 2006].

Fig. 2.3 – Examples of contact systems used for the 3D reconstruction of objects. From left to right: [FaroArm, 2007], [MDX-15/20, 2007] and [Dimension, 2008].

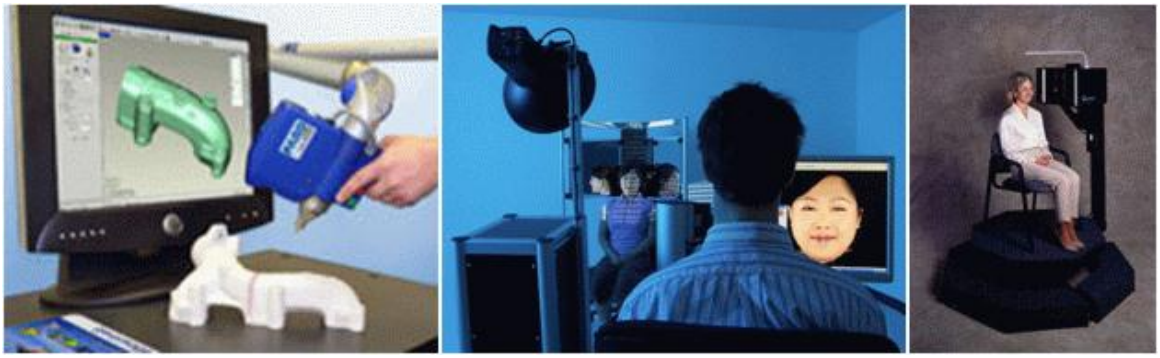

Fig. 2.4 – Examples of non-contact active systems used for the 3D reconstruction of objects. From left to right: [LaserScanArm, 2007], [FaceSCAN3D, 2012] and [Cyberware, 2007].

Range sensors acquire distance measurements from a well-known reference coordinate system to points of the object to be reconstructed. They are very common when highly detailed models are required and are already used in industry (e.g., for design, quality control and rapid prototyping), for documentation of buildings and landscapes or for the recording of objects in archaeology and cultural heritage [Böhler, 2004]. However, they are costly (at least for now), spatially limited, most of the systems available do not provide colour information about the reconstructed objects and the resulting 3D models' quality can be affected by the reflective characteristics of the reconstructed objects' surfaces [Remondino, 2006]. Strongly reflective or translucent objects often violate assumptions crucial for the success of range-based scanners, requiring additional solutions to obtain the 3D models for such problematic objects.

Main difference between image- and range-based methods is that, when using image data, it is necessary to have a mathematical model to derive the object 3D coordinates, which can be sometimes a complex problem to solve [Remondino, 2005]. Building 3D models using range methods is simpler, because the range data acquired already contains the 3D coordinates needed for the 3D reconstruction.

2.2 Methods for Human 3D Reconstruction

The 3D geometric reconstruction of the human body was initially considered in the automotive and aeronautics industry, namely for ergonomic purposes [Thalmann, 1993].

Used models consisted of a simple articulated skeleton to define human pose, with the body being represented using simple geometric primitives, such as cylinders, parallelepipeds or spheres [Dooley, 1982]. Currently, 3D models of the human body are much more realistic and of great interest in a wider variety of application fields, Fig. 2.5, such as:

- cinematographic industry: combination of photographic imagery of people with rendered 3D environments or background imagery (e.g. [Einarsson, 2006]), animated digital actors (e.g. [Alexander, 2010]) or morphing effects (e.g. [Andersen, 2004]);
- clothing industry: virtual fitting (e.g. [Zuo, 2011]), or assist the garment design (e.g. [Liu, 2010]);
- biomedical applications: 3D models for image-guided surgery or surgery planning (e.g. [Yunfeng, 2010], [Wang, 2012]), biological systems modelling (e.g. [Jumadi, 2012]), body asymmetries (e.g. [Atkinson, 2010], [Milczarski, 2011]);
- biomechanics: rehabilitative engineering (e.g. [Sousa, 2007], [Teodoro, 2009]), occupational model simulation (e.g. [Sahai, 2010]), injuries and impacts on sport activities (e.g. [Machado, 2009], [Carroll, 2011]).

Human reconstruction from images is inspired by the need to provide such models with a photo-realistic appearance obtained from images of real people or body parts. Next section provides an overview of human reconstruction solutions and methods.

2.2.1 Contact methods

Contact methods mean the use of some kind of devices that are in touch with parts of the human body. Main disadvantage of these methods is that, because the device must be in touch with the skin, some deviation in the results could appear because of the mechanical contact.

Fig. 2.5 – Examples of nowadays 3D human models and applications (from [D'Apuzzo, 2012]).

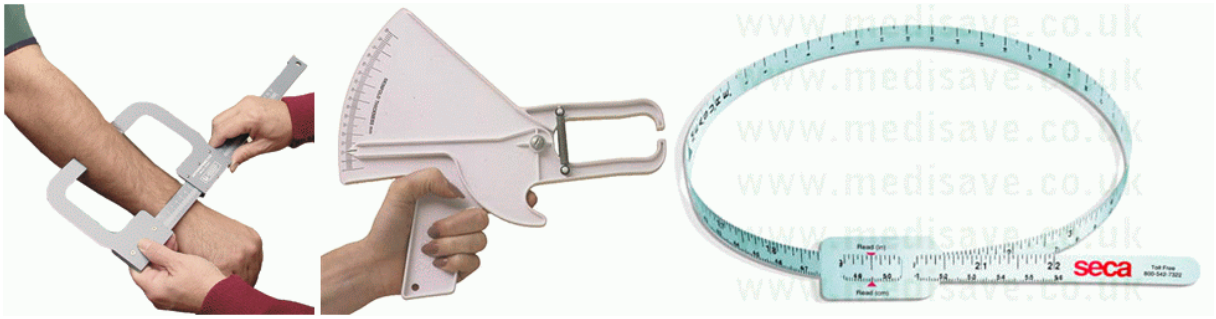

Fig. 2.6 – Three examples of anthropometric devices, from left to right: anthropometer (from [Lafayette, 2007]), calliper (from [NextGen, 2007]) and head tape measure (from [SECA, 2007]).

a) Anthropometric devices

Anthropometric devices are simple mechanical instruments still being utilized to measure the human body at particular regions. Some examples are the calliper, the anthropometer, the sliding compass or the tape measure, Fig. 2.6. Although simple and of low cost, with this kind of instrument it is only possible to measure the length of a cross-section of a limb.

b) Shape tape

A tape device is a wearable flexible ribbon with optical fibres and tensors, which can detect 3D bending and twisting movements, Fig. 2.7. Output signals from the tape sensors array can be used to build a computational 3D real-time model of a shape. The array of sensors obtains its position relative to a fixed origin point defined on the tape and can be used with static or dynamic shapes, including curves and twists [Naik, 2002].

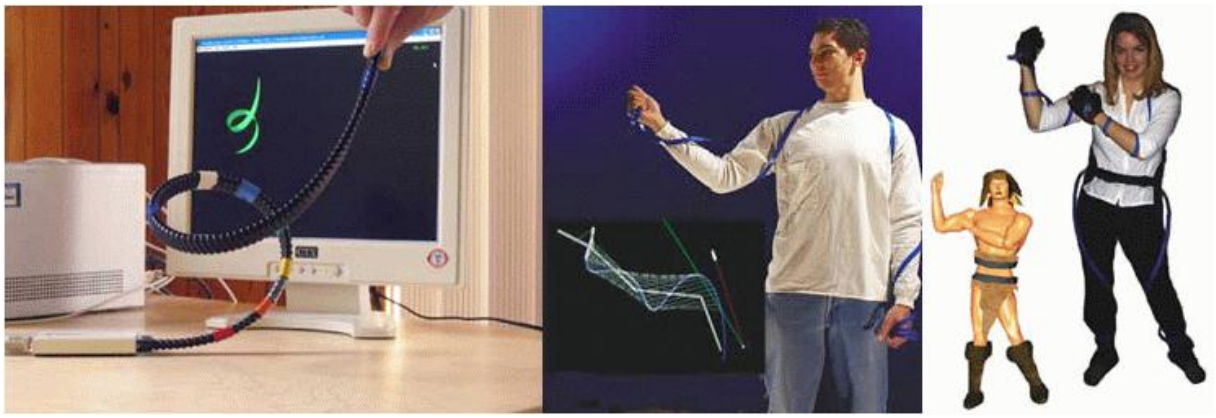

Fig. 2.7 – Examples of 3D models built using Shape Tape™ (from [ShapeTape™, 2007]).

One main advantage of shape tapes is their ability to be dynamically used. In fact, a person does not need to be standing in a fixed position, and it is possible, for example, to measure the shape of a limb in different kind of positions or even in motion.

However, to make a full body geometric model, it is required to cover the whole body skin with shape tapes. Thus, the more complex is the shape of the object to be reconstructed, more difficult it will be to reach all its specific points using the shape tape [Kalkhoven, 2003].

c) Articulated arm digitizer

Articulated arm digitizers are robot-looking devices, usually mounted on a relatively heavy base, with articulated joints that allow movement in any direction. Each joint has a sensor to know the associated angle; the combination of sensors throughout the arm structure allows it to register the position of the tip of the arm in the space. The 3D coordinates measured are feed via a serial interface into a computer, which builds the object 3D geometrical mesh, Fig. 2.8. The actual position of the tip of the external arm is computed by goniometry, which measures the angles between the digitizer arms at the joints.

A problem with this type of devices is the large space that is usually required to operate the moving arm. It is not very easy to reach all points of the object as well, because sometimes they are covered or inaccessible, and therefore, cannot be reached by the probe. Another limitation of the articulated arm digitizer is that it can only digitize a limited volume, meaning they are most suited for small- to medium-sized objects, but not for large objects. Thus, they are usually used to build models of body parts, and not in full body 3D reconstruction.

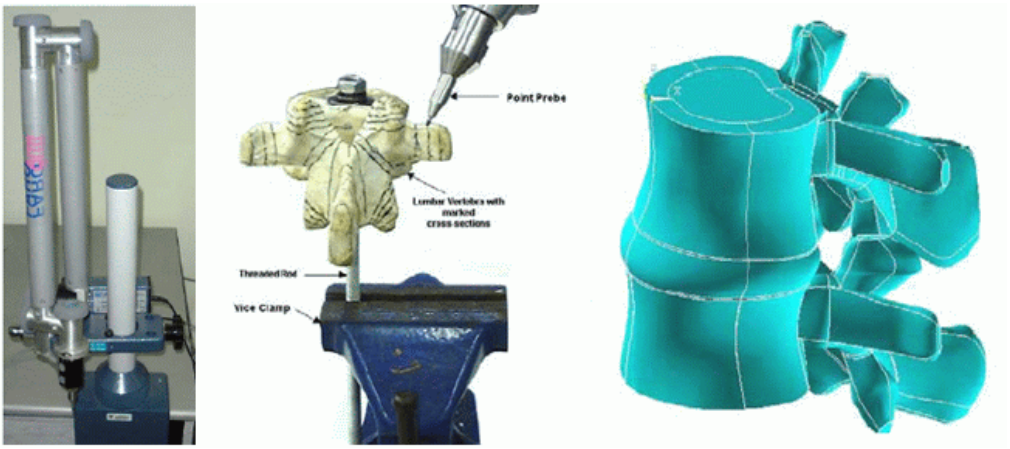

Fig. 2.8 – From left to right: an 3D arm digitizer, capturing a human lumbar vertebra using a digitizer device and the spine volume rendered from the digitized data (adapted from [Lee, 2002]).

Fig. 2.9 – On the left: *FARO Laser ScanArm* [ScanArm, 2012]. On the right, 3D bone modelling using the laser scanning capability of an articulated arm digitizer (from [Harris, 2011]).

Another problem is that these devices must touch the object to be reconstructed, which, in the case of human body reconstruction, may cause deviations in the measures obtained because the flexibility of the tissue [Kalkhoven, 2003].

Main advantages of these systems are their portability, simplicity of use and, mainly, their good precision – they usually achieve precisions under 0.04 mm (see, for example, the specifications of two widely used arms, [ScanArm, 2012] and [AbsoluteArm, 2012]).

Nowadays, several commercial articulated arms have integrated laser scanners, which enable hard-probe measuring of simple point variations, then laser scanning sections for larger volumes of data, Fig. 2.9.

2.2.2 Range-based methods

As stated before, range or depth maps are a collection of distance measurements from a known reference coordinate system to surface points on the object to be reconstructed. Usually, high quality 3D models of a static human body are obtained using commercial scanners (range systems), that are generally expensive, but fast and accurate. Main contribution factors for the success of range-based methods are their precision and the wide variety of available software packages to process the acquired data, build the 3D models and determine characteristic measures.

The 1960's saw the introduction of new scanning technologies, which revolutionized the 3D human model reconstruction. Initial scanning devices were only able to capture one side of the body at a time, until 1985 when Magnant, [Magnant, 1985], developed a system that completely surrounded the body [Simmons, 2003]. The last two decades have seen significant improvements in scanning technologies. Several companies currently manufacture 3D scanners (e.g. [FaceSCAN3D, 2012], [BodySCAN3D, 2012], [FastScan™, 2012], [Mega-Capturor, 2012], [Anthroscan, 2012], among many others), each utilizing somewhat different scanning techniques, capturing a distinct number of data points and producing slightly different results. As the cost of scanning technologies begins to decline, research using 3D scanners is becoming more accessible to both universities and industry.

To create a full 3D model that includes all parts of an object, the 3D scanner must first capture 3D scans of an object from several directions. Once all parts are captured, the next step is to merge the obtained 3D meshes together to form a 3D model, Fig. 2.10. There is some software that can merge the scans directly. However, the process of acquiring 3D scans and later merging and aligning them to create a polygon mesh involves time and requires some practice [Tocheri, 2009]. When scanning the full human body, it may occurs the case where the mesh alignment process needs to be achieved manually, due to strong occlusions on the obtained 3D meshes, Fig. 2.11.

Another solution is to use a rotary table system (also known as turntable), which automates the process of performing multiple scans of an object and automatically aligns and merges the 3D scans together to form a 360° model. This device works with the 3D scanner to reduce 3D scanning and processing time. The rotary table device usually imposes size and weight restrictions on the object to be scanned.

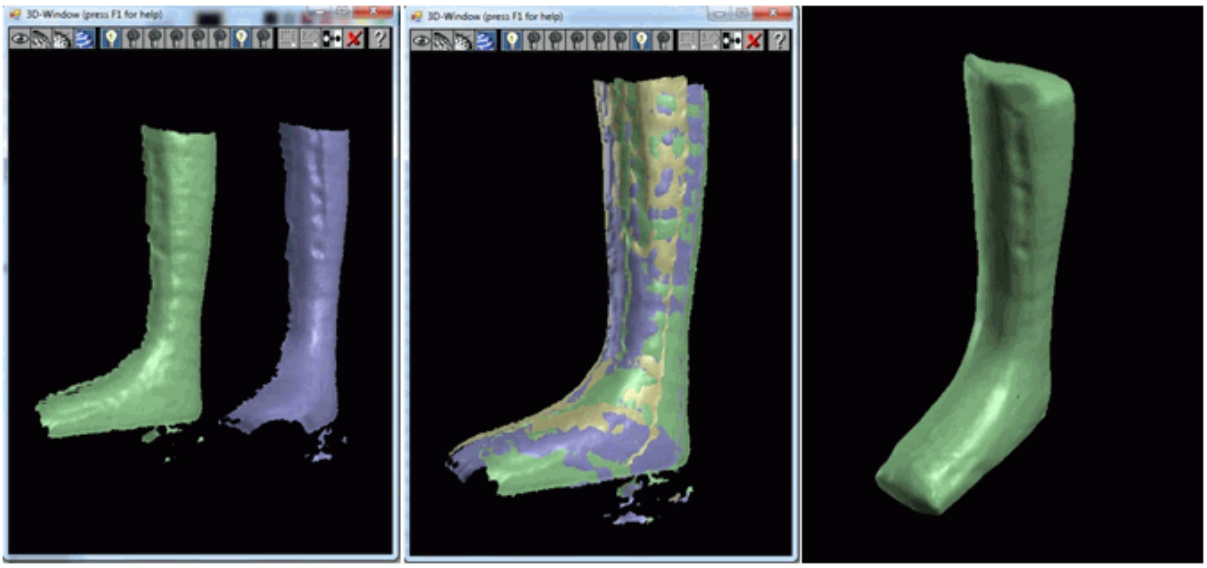

Fig. 2.10 – On the left, two of the many 3D meshes obtained of an orthopedical boot. In the middle, automatic fusion result obtained using the *DAVID Shapefusion* software [DAVID, 2012]. On the left, the final 3D closed geometry (from [Pinto, 2011]).

Fig. 2.11 – Manually rendered combination of frontal and back views of a swimmer (from [Tavares, 2008]). Several difficulties to obtain a closed 3D surface were related to the colour of the swim suits, which are usually very dark.

Another way to automate the process of 3D scanning is to use a 3D scanner that has multiple scan heads (an example is the system on Fig. 2.12). Each scan head takes 3D scans in a different direction so the user does not need to rotate the object. This configuration is good for controlling multiple scan heads connected to a single computer to maximize object coverage. It is especially useful for scanning an object that requires a fast scanning speed such as scanning the human body. This option is more expensive and has a more complex setup.

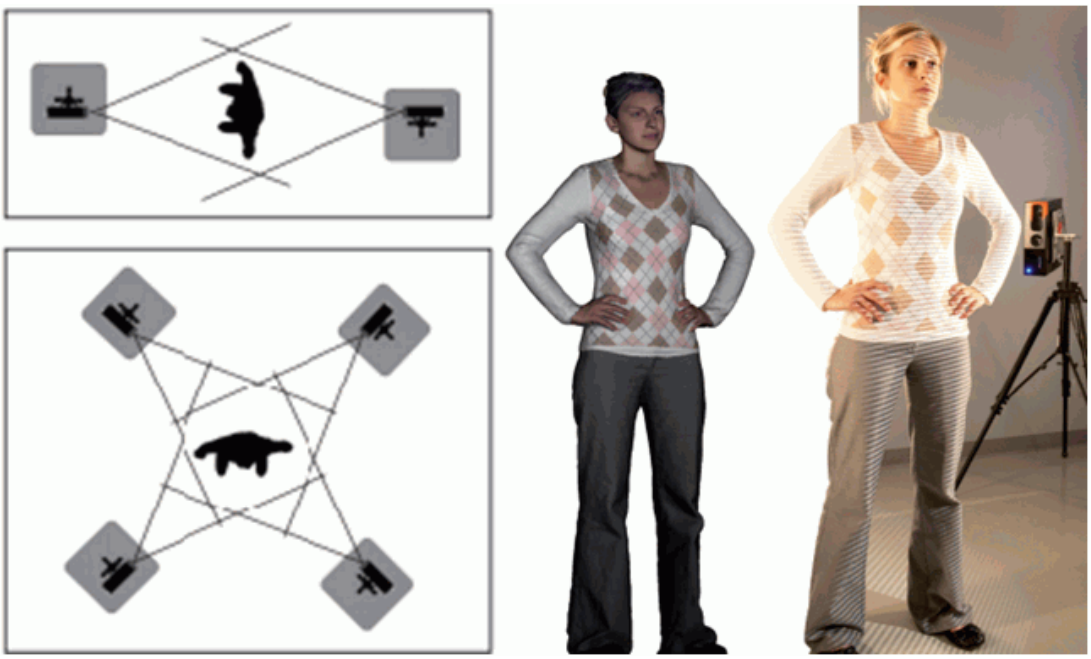

Fig. 2.12 – Textured 3D model built using the structured light system *MegaCapturer* from *Creaform* ([Mega-Capturor, 2012]).

A third way to speed up the reconstruction process is to automate the alignment of 3D scanned data. This process is usually called photogrammetry data alignment based on markers, Fig. 2.13. The markers can be placed either on the object or on the surface the object rests on. After scanning the object and generating a mesh, the scanner software searches for the markers and attempts to align them with the previous scan. There are two ways to place the markers:

- direct placement over the object;
- indirect placement on the surface where the object is placed.

The indirect approach allows getting more details from the object because no section is covered by markers, and it is suitable for objects with textures were direct marker placement will not work. But, in order to be reliable, it requires that the object remains static relative to the markers. If the object needs to be moved, then the all surface were it is placed on must be moved as well.

This leads to the inability to scan the object section directly in contact with the underlying surface without starting a new mesh in which the object is moved to a new position. After this, the several meshes obtained must be aligned to build the final model using a mesh alignment algorithm.

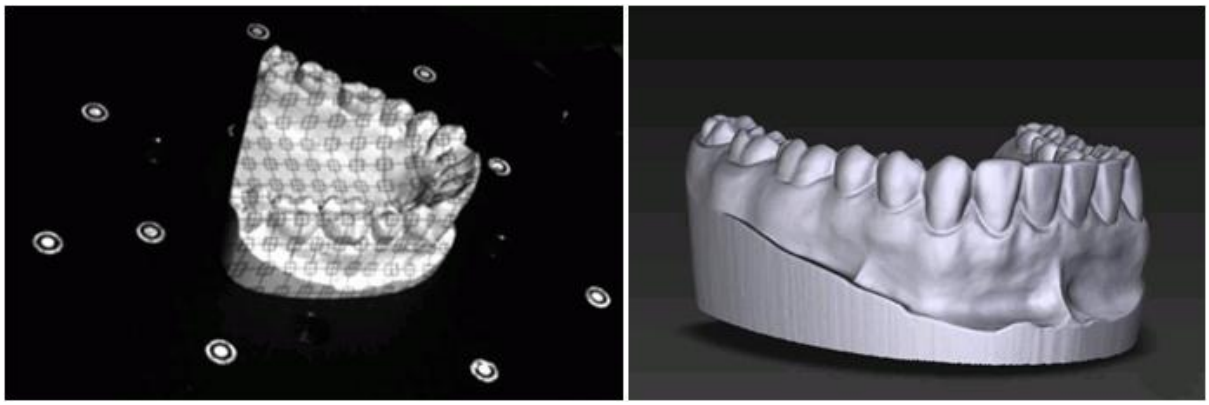

Fig. 2.13 – On the left, dental model being scanned using the structured light system *HDI 3D Scanner* from *3D3Solutions*, [3D-Scanner, 2012], with markers placed on top of the table to facilitate the 3D scanned data alignment.
On the right, the 3D model built.

Fig. 2.14 – The laser scanning system *Anthroscan* (on the left), [Anthroscan, 2012], from *Human-Solutions*, and an obtained 3D model (on the right).

Laser scanning, Fig. 2.14, and structured light, Fig. 2.12, are the range-based methods most widely used to build 3D models of the human body.

Several studies can be found regarding laser scanning methods for human body 3D reconstruction. Many of these studies compare the manual anthropometric measures to the ones obtained from the acquired images (e.g., [Kovacs, 2006], [Park, 2006], [Chiari, 2008]). Other works develop strategies to overcome some common drawbacks in body modelling from images. For example, in [Brunsman, 1997], an experiment was conducted to determine optimal scanning positions; in [Xi, 2007], a morphable human body model was built from a dataset of 3D scanned surfaces, which is often a problem because of incompleteness on the

built surfaces and due to variances in shape, size and pose, that complicate the matching process between surfaces. Noisy point clouds obtained from a laser line scanner are smoothed and existing holes due to occlusion, bad reflectance and body movements effects are reduced using a spatial gray model to predict the unknown points in [Zhu, 2010].

Many other solutions were developed for building 3D models for specific human body parts, as in [Byrne, 2007], where 3D laser scanning was used to aid in the creation of a surgical guide, or in [Singare, 2010], where a laser scanner was used to capture the morphology of a human ear and reconstruct the cast digitally and then physically by rapid prototyping.

Works that use structured light reconstruction methods often emphasize the relatively simple hardware used, reduced scanning time and the fast algorithms involved [Čarnický, 2006]. Moreover, being based on simple light projection, structured light scanners do not have an inherent safety issue like laser scanners have, especially eye-safety danger.

Most structured light systems are named white light scanners because the light coding system is simply based on monochromatic (mostly, white) light. Other works use more complex structured light methods. For example, phase shifting is a time multiplexing strategy by projecting periodic light patterns (e.g. [Tavares, 2008], [Pribanić, 2010], [Wang, 2011]), Fig. 2.15. The fringe projection is a single-frame method which can cope with dynamical events, but the post-processing necessary to attain accurate results can be too intensive to be considered interesting in real-time applications. Other systems use time-shifted colour striped patterns in order to obtain high-resolution models of static scenes (e.g. [Yu, 2010]).

Several techniques for 3D scanning by structured light are currently available and reviewed in detail in [Geng, 2011].

For all range-based methods for 3D reconstruction, acquisition time depends on the object's size and used hardware and software, during which the object being reconstructed must remain stationary. Thus, it is difficult to obtain stable results as humans always move slightly during the data acquisition (e.g. [Calabrese, 2007]). Other drawbacks of these systems are missing data points, where the laser or projected pattern is occluded, Fig. 2.11, and erroneous data due to specularities (e.g. [Forsyth, 2003], [Zhu, 2010]).

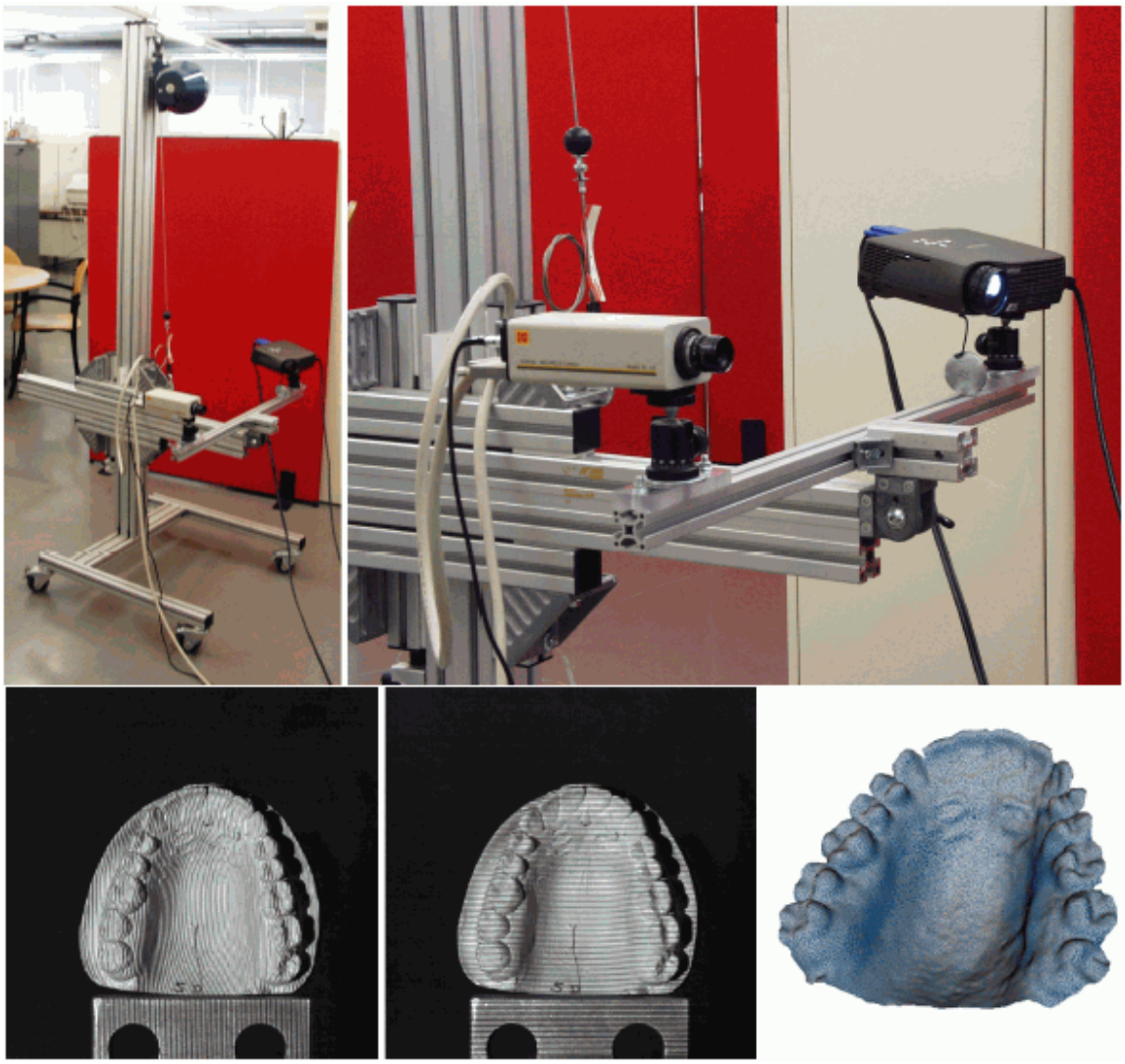

Fig. 2.15 – On the top, a phase-shifting 3D reconstruction system that uses a white light projector and an image camera. On the bottom, two images from a dental model with the projected stripes and the 3D model obtained for the same.

Although not so common for human 3D reconstruction, there are others range-based methods. For example, time-of-flight scanners use a signal transmitter and a receiver to measure the time of flight of the signal during his round trip from the range sensor to the object surface. Time-of-flight systems are compact devices, less expensive than most commercial 3D scanner systems and are able to provide for real-time distance acquisition [Kolb, 2009].

A second approach for range-based methods is based on light coding, projecting a known infrared pattern onto the scene and determining depth based on the pattern's deformation.

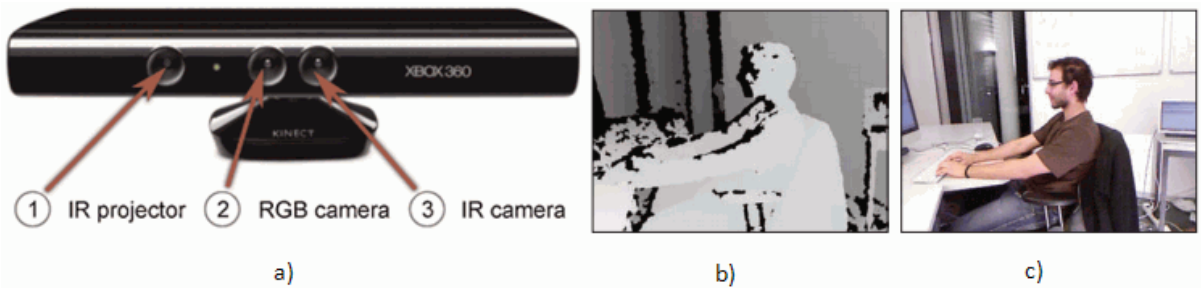

Fig. 2.16 – *Microsoft Kinect* sensor: a) *Kinect* is a device which combines an RGB camera and a 3D scanner, consisting of an infrared (IR) projector and an IR camera (from [Zollhöfer, 2011]); b) and c) are the raw data obtained from *Kinect*, a depth map and a colour image, respectively (from [Weise, 2011]).

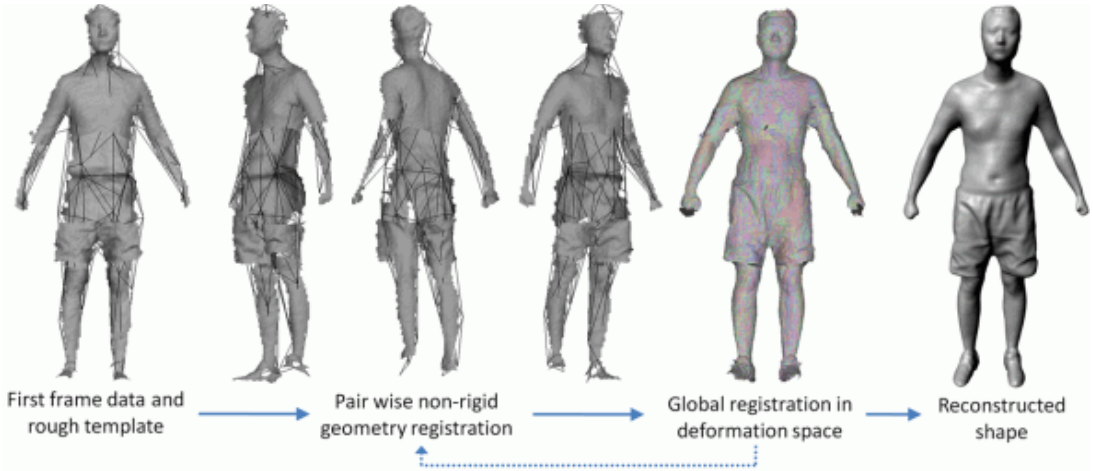

Fig. 2.17 – Algorithm overview of the 3D reconstruction algorithm proposed in [Tong, 2012]: a rough mesh template is constructed and used to deform successive frames pairwisely; then, global alignment is performed to distribute errors in the deformation space.

A newly, and very popular, device is the *Microsoft Kinect* sensor [Kinect, 2012], Fig. 2.16, which is much more inexpensive than time-of-flight sensors. Some researchers have tried to use *Kinect* as 3D scanner (e.g. [Henry, 2010], [Izadi, 2011]), but few have still used it to capture human full body or body parts. In [Zollhöfer, 2011] a personalized avatar is computed from a single RGB image and its corresponding depth map, both obtained from *Kinect*. In the work described in [Weise, 2011] the dynamics of facial expressions are captured and tracked in real-time and then mapped to a digital character.

For full body 3D reconstruction, the main problem with the *Kinect* device is that it has a comparably low X/Y resolution and depth accuracy. To scan a full human shape, the device must be placed around 3 meters away from the person, and the resolution is relatively low. Nevertheless, using the information of multiple frames, as in [Cui, 2011], or using a 3D model of a human body, like in [Weiss, 2011], or finally by capturing data from several *Kinect* sensors, as in [Tong, 2012], final resolution can be enhanced, Fig. 2.17.

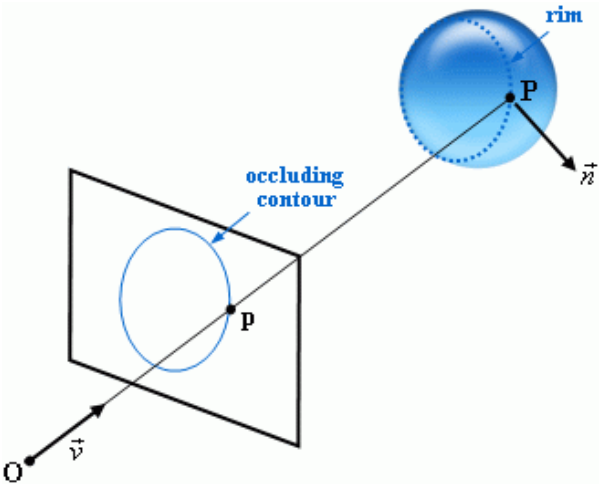

Fig. 2.18 – A rim and occluding contour under perspective projection: the 3D point, **P**, is constrained to be on the back-projection ray going through its image point, **p**, and the camera optical centre, **O**.

2.2.3 Image-based methods

Other usual 3D static human reconstruction methods are image-based. They are an easy, inexpensive and flexible approach, based on image measurements. The required images can be acquired using video or image cameras, which presently can be acquired at low cost and are relatively robust acquisition systems.

a) Silhouette extraction

Considering an object with a smooth surface and camera's perspective projection, a silhouette, or occluding contour, is the rim's projection onto the image plane. A rim is the locus of points on the object's surface where the normal vector, $\vec{n}$, is perpendicular to the viewing direction, $\vec{v}$, Fig. 2.18.

Silhouettes in 2D images directly reflect the 3D shape of the acquired object. Obtaining continuous silhouette from input images is usually a computationally simple task if the images are acquired in a controlled environment. However, it is a very challenging task to apply silhouette-based reconstruction methods in uncontrolled or outdoor environments, where occlusions may occur [Guan, 2007]. A 3D model is built by intersecting the visual cones generated by the silhouettes and the projection centres of each image, Fig. 2.19. This 3D model is denominated visual hull [Laurentini, 1994], a locally convex over-approximation of the volume occupied by an object.

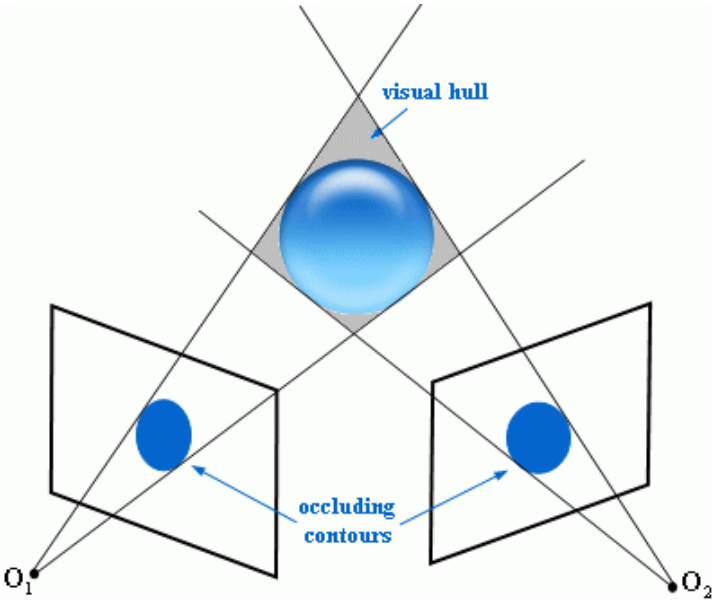

Fig. 2.19 – Visual hull defined by two different viewpoints.

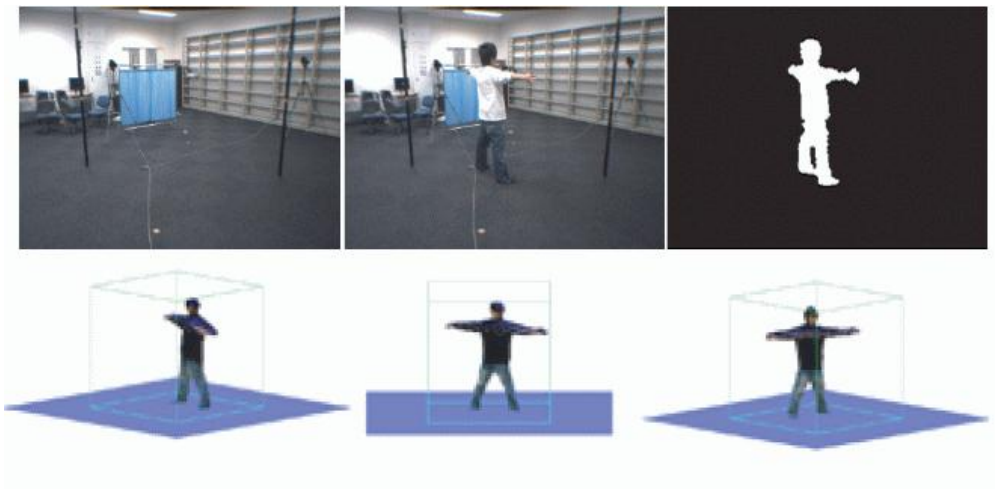

Fig. 2.20 – Silhouette-based reconstruction of a human body (from [Takahashi, 2006]).
On the top: on the left, background image; on the middle, input image; on the right, segmentation result by background subtraction on the input image.
On the bottom: 3D reconstruction from silhouette images.

One important property of visual hull is the conservation constraint [Laurentini, 1994], meaning that the actual object shape is always contained inside the visual hull. Thus, shape-from-silhouette methods are widely used to provide a coarse 3D shape estimate, which can serve as an initialization for more sophisticated 3D shape reconstruction algorithms, like in [Faugeras, 1999], for example.

For the reconstruction of human 3D models, the images are usually acquired around a static person, Fig. 2.20. Then, image segmentation is performed in order to obtain the human body silhouettes.

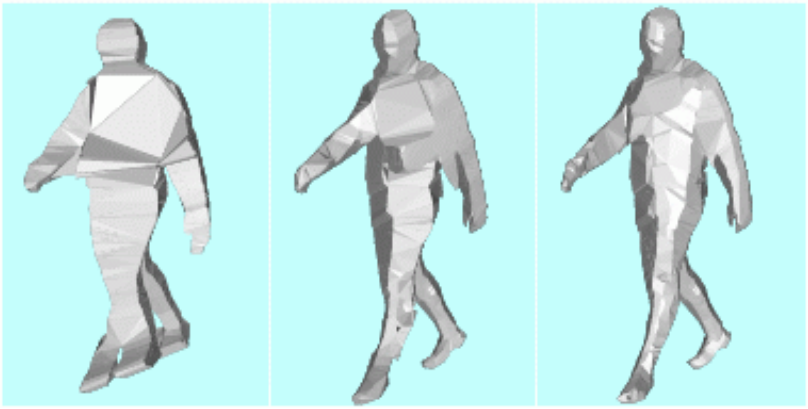

Fig. 2.21 – From left to right: visual hulls of a human body, built from 2, 4 and 6 viewpoints (adapted from [Franco, 2006]).

Some examples of work on 3D human body reconstruction using silhouettes can be found, for example, in [Franco, 2005], [Takahashi, 2006], [Ladikos, 2008], [Zabulis, 2010], [Feldmann, 2010], [Monaghan, 2011] or [Kolev, 2012].

As aforementioned, recovering the shape of an object from silhouette images suffers from the limitation that the visual hull can be a very coarse approximation when there are only a few silhouette images available, especially for objects with complex shapes, like the human body, Fig. 2.21.

Better shape estimates can be obtained if the number of distinct silhouette images is increased: using the across space approach, by increasing the number of physical cameras used, or the across time approach that increases the number of silhouettes by acquiring a higher number of images from each camera over time (while the object is moving) [Cheung, 2005]. Typical across time approaches are shape from rotation (e.g.[Fitzgibbon, 1998], [Mendonça, 2001] and [Hernández, 2007]): placing the object to reconstruct on a turntable device, the motion is known in advance.

Shape-from-silhouettes methods fall in roughly two categories: surface-based and volumetric methods [Kutulakos, 1997].

Surface-based methods compute local surface shape by establishing correspondences between the visual cones. They pose several difficulties, like:

- guaranteeing the validity of curve correspondence across frames, even for simple object shapes, e.g. [Zhao, 1994];
- detecting and handling degenerated cases, like when the surface is flat or has creases, e.g. [Zheng, 1994];

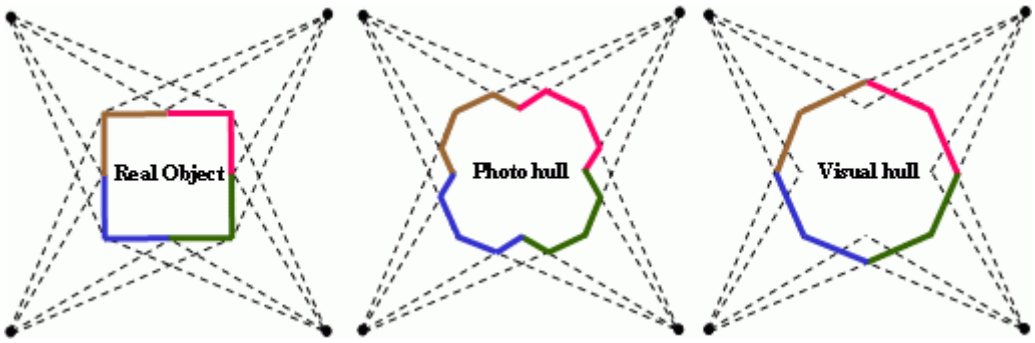

Fig. 2.22 – Relation between photo and visual hull:
real object $\subseteq$ photo hull $\subseteq$ visual hull.

- handling dense image sequences, because optical rays through corresponding curves can become coincident [Kutulakos, 1997].

These aforesaid difficulties usually associated to surface-based methods led to the development of volumetric approaches, which represent the visual hull by 3D volume elements (voxels or 3D pixels) rather than 2D surface patches. The first work to propose representing the visual hull through voxels was [Massone, 1985]. The space of interest is divided into discrete voxels, which are then classified into two categories: inside and outside. The union of all the inside voxels is an approximation of the visual hull.

Even though volumetric reconstruction does not involve curve tracking and produces a global object description, they raise two additional issues:

- the difficulty of computing volume intersections has motivated voxel-based representations of space shape, that limits the reconstruction accuracy to the size of individual voxels, e.g. [Szeliski, 1993];
- volumetric methods cannot recover the shape of regions that project into occluded contours, and consequently, are not part of the silhouette, e.g. [Cipolla, 1992].

Volumetric methods evolved in a way to test if each voxel belongs to the object of interest by using photo-consistency tests. The Space Carving reconstruction method was presented in [Kutulatos, 1998]. With this method, the resulting 3D model is the photo hull, defined as the biggest volume of voxels that is photo-consistent with all viewpoints, Fig. 2.22. Photo-consistency is checked statistically: a voxel is considered consistent if the mean deviation of the pixels' colour, which results from the voxel image projection, is under a predefined threshold.

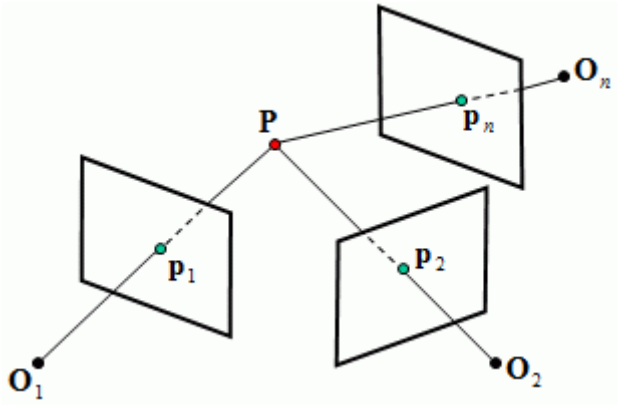

Fig. 2.23 – Multi-view geometry: given the correspondences between pixels $\mathbf{p_i}$, i.e., the ones that are the projections of the same 3D point **P**, this last one can be reconstructed by intersecting the optical rays that pass through the camera centres and corresponding image points.

b) Multi-view geometry

Multi-view geometry consists on reconstructing the 3D shape of an object from correspondences on a sequence of images, acquired from different viewpoints. They are based on searching for the correspondence between image points to infer about the 3D position in a scene from the image plane locations, Fig. 2.23. Hence, these methods are also known as photogrammetric methods. Multi-view geometry is considered an evolution of stereo-based methods, where only two images were used: first, correspondences between points in both images were established (matching) and then their position in 3D space was determined by triangulation [Hartley, 1995]. An example on how this method builds a 3D model of the human body, can be found in [Remondino, 2004], Fig. 2.24.

The use of off-the-shelf cameras for accurate photogrammetric applications is reported in several works, such as in [Chandler, 2005], where the potential of low-cost digital cameras for close range 3D reconstruction using feature-based matching methods is examined.

Reconstruction methods based on multi-view geometry may suffer from difficulties on finding interest points and/or matching them along the input image sequences [Hartley, 2004]. First, if the object to reconstruct has a smooth surface or low texture, the extraction of interest points may be incorrect since the local appearance is uniform within the neighbourhood of each candidate point. Secondly, matching correspondence cannot be established by just comparing local image statistics, unless the object has a Lambertian surface; that is, its appearance does not change with the viewpoint. Finally, occlusions in the scene make the correspondence between images ambiguous or even impossible.

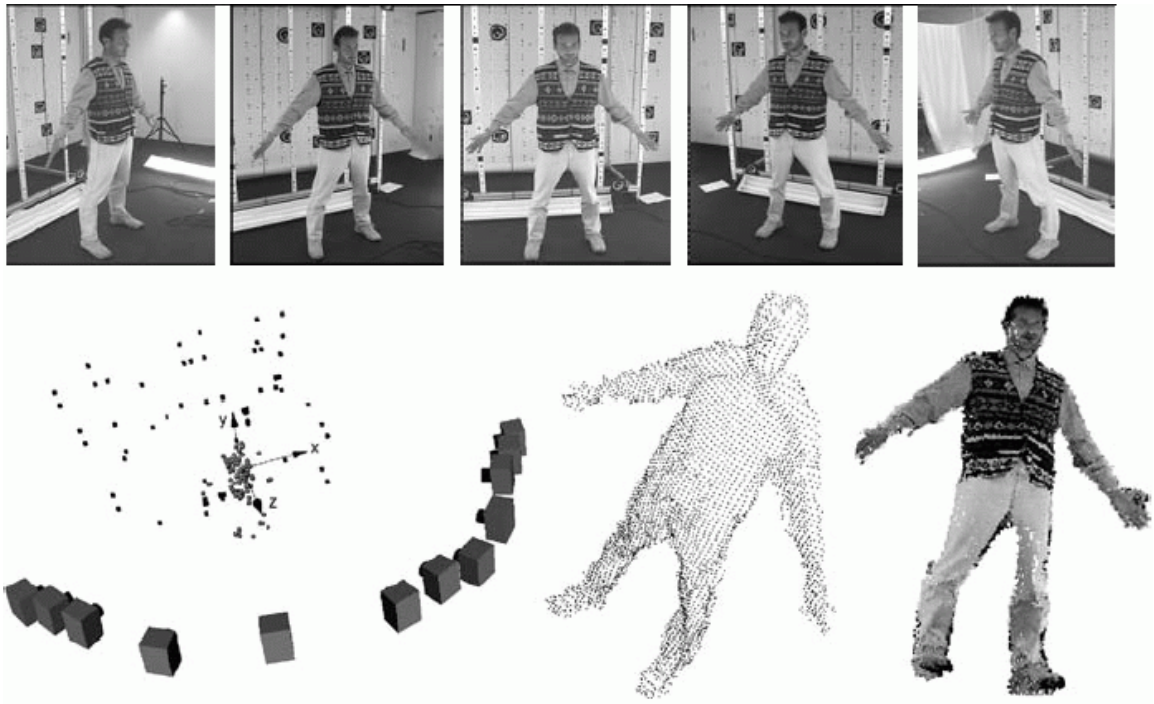

Fig. 2.24 – On the top: five images from an original sequence. On the bottom: on the left, recovered camera positions and objects coordinates; on the middle, 3D point cloud of the human body; on the right, relative visualization with pixel intensity (from [Remondino, 2004]).

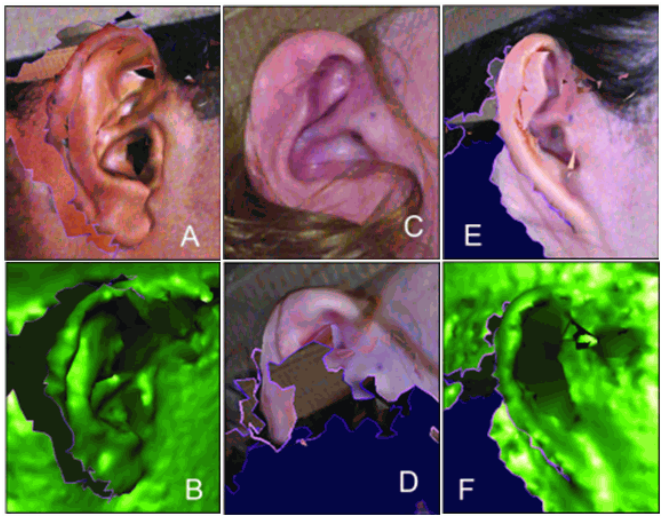

Fig. 2.25 – Examples of inadequate 3D face surface reconstruction using the stereo photogrammetric system *3dMDCranial* [3dMDcranial, 2012]: poor ear coverage occurred due to the angle at which the person was facing relative to the camera, a) and b), due to interference from scalp hair, c) and d), or due to the intricacy of the human external ear, e) and f) (from [Heike, 2010]).

All the difficulties aforementioned are frequently perceptible in images of humans, Fig. 2.25: clothing can have a lack of significant local variation in their appearance or present a repeated pattern, articulation of the body which leads to self-occlusions, clothing or skin exhibits a non-uniform view-dependent appearance and capturing hair can cause interference on the results obtained.

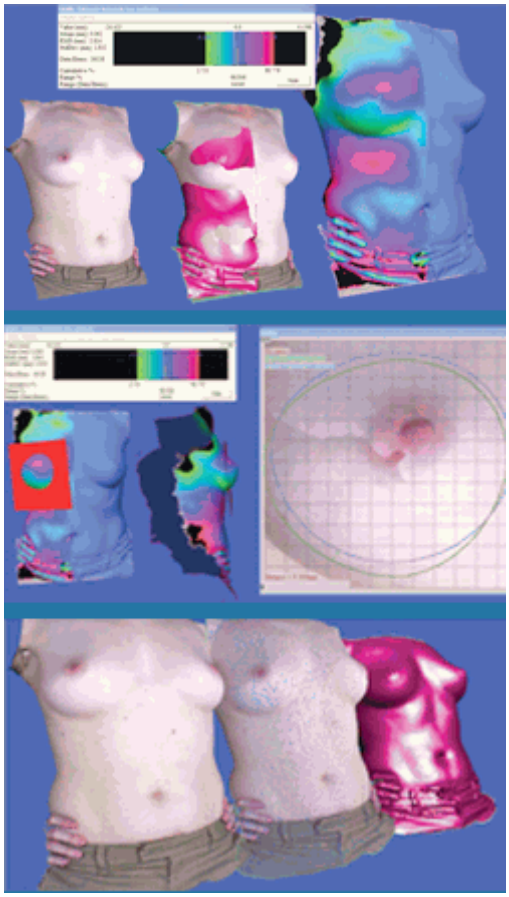

Fig. 2.26 – Breast 3D data acquired using the *3dMDTorso* system based on random light projection [3dMDtorso, 2012].

To overcome these problems, variations to the original method have appeared. The most common variation is to use stereo-photogrammetry, Fig. 2.26, where the human body is marked with artificial fiducials (e.g. points) in order to obtain accurate matches (e.g. [D'Apuzzo, 2002], [Deli, 2009], [Yu, 2010], [PhotoModeler, 2012], [3dMDtorso, 2012]).

2.2.4 Model-based methods

Reconstruction methods described previously aim to deal with the general problem of reconstructing the arbitrary shape and appearance of an unknown object.

In model-based methods, a model of the expected object is refined for shape recovery. Generically, model-based methods use a prior knowledge of the object geometry to constrain the shape recovery in the presence of visual ambiguities. Thus, they can reduce the influence of noisy, sparse or outlier data in shape estimation, providing more accurate, continuous and smooth 3D models than model-free methods [McInerney, 1996], Fig. 2.27.

Fig. 2.27 – Different results for image-based 3D reconstruction. From left to right: original image, 3D model obtained using a visual-hull method, 3D model obtained using a stereo-based method and 3D model obtained using a model-based method (from [Starck, 2003]).

Optimisation of a surface model to match image data was introduced in [Terzopoulos, 1987]. Physically based models were presented, which can be applied to reconstruct the shape and motion of flexible objects from images.

A deformable model was formulated as a continuous elastic surface that deforms dynamically, in order to satisfy a set of imposed shape constraints. Model deformation was defined as an energy minimisation task with an external energy that applies the constraints on model shape and an internal energy that penalises the elastic deviation in the model surface to regularise deformation. The drawback of the approach lies in defining the degree of regularisation required to define the trade-off between fidelity in fitting the constraints and penalising the model deformation [Thompson, 1991].

There has been significant research on deformable models. For example, in [Terzopoulos, 2011], a review of the different representations of deformable models is made, focusing on their applications in medical image analysis. It also reviews functional models, particularly realistic biomechanical models of the human face and body. A review on medical image segmentation using deformable models (among others) is presented and their applications to the female pelvic cavity 3D reconstruction are pointed out in [Ma, 2010]. In [Tavares, 2003], the fundaments of deformable models are presented and some application examples in the medical imaging field are indicated, namely in segmentation, matching, alignment and in reconstruction of 2D and 3D data. The work in [Moura, 2010] focus on the application of deformable models to provide fast 3D reconstruction of the spine that may be accomplished by non-expert users.

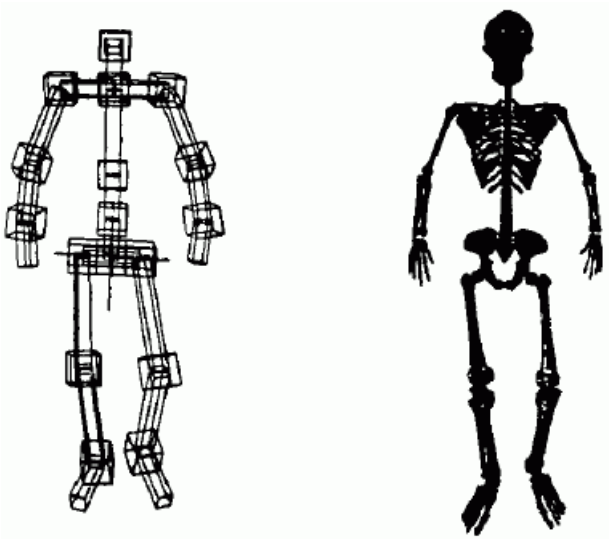

Fig. 2.28 – Mapping of the human bones (on the right) into a human skeleton (on the left) (from [Chadwick, 1989]).

In general, human 3D models follow the multi-layered model approach introduced in [Chadwick, 1989]. The multi-layered model consists of a skeleton structure, representing the bones, Fig. 2.28, with successive layers of details, representing the body tissue, surface shape and external detail such as clothing and hair. The skeleton provides an animator with high-level motion control, and each layer of the model is then successively deformed with the skeleton to produce natural-looking surface deformations.

A variety of geometric modelling methods have been proposed to represent human geometry, including the polygonal mesh, e.g. [Botsch, 2010], parametric surfaces, e.g. [Kaiser, 2011], and implicit surfaces, e.g. [Ilic, 2006].

Human models have been extensively used in Computer Vision, namely for the problem of human motion analysis, Fig. 2.29. The visual analysis of human motion forms a major area of research that addresses the detection, tracking and recognition of different people or human actions in images. Human models are used to impose constraints on the geometry or motion of a person in a set of images in order to estimate the human pose. These models are generally specified a priori and use simplified geometric representations such as a stick figure, 2D contour or 3D volumetric primitives.

There is extensive literature on human motion tracking and several surveys have been presented, for example, in [Moeslund, 2001], [Wang, 2003], [Moeslund, 2006], [Poppe, 2007] and [Ji, 2010].

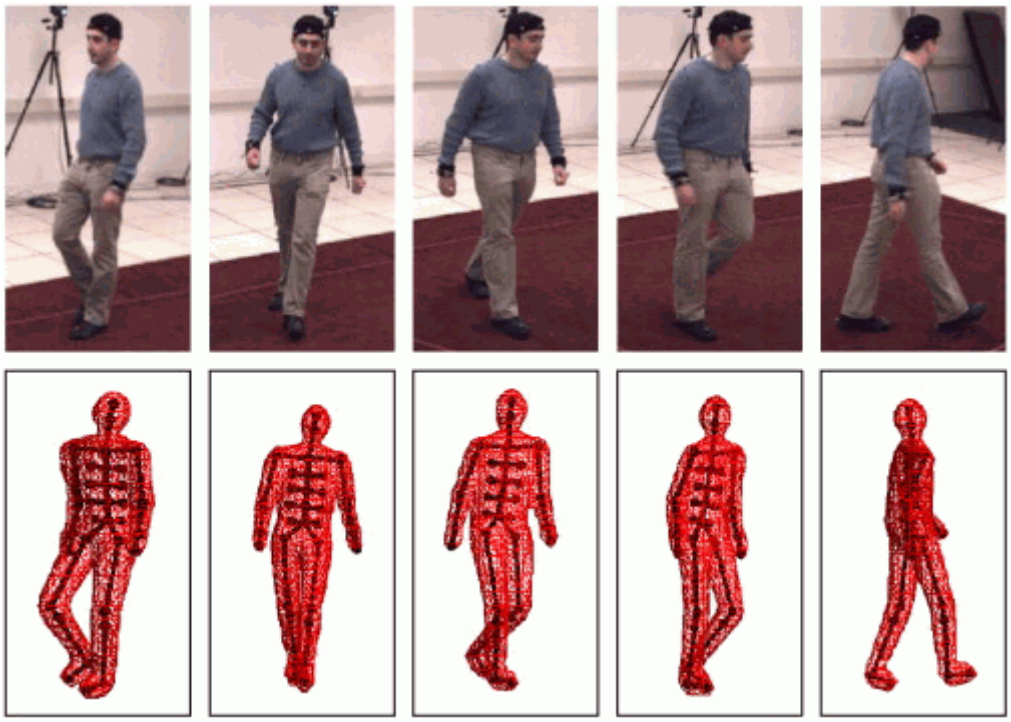

Fig. 2.29 – Results of recovering the poses of a person performing a turning walking motion. Top row shows the original images and the bottom row shows the obtained 3D poses model by a 3D convolution human surface and skeleton models (from [Zhao, 2008]).

2.3 Summary

This Chapter summarizes usual methods for 3D reconstruction, which can be understood as a process that starts from the data acquisition and ends with a virtual 3D model.

The 3D reconstruction of objects and/or scenes is required in many applications and has recently become a very important and fundamental step, especially in cultural heritage, biomedical applications or virtual/augmented reality. The requirements specified by many applications (like digital archiving or recognition) are high geometric accuracy, photo-realism of the results, modelling with complete details as well as automation, low costs, portability and flexibility of the reconstruction method. Therefore, selecting the most appropriate method for a given application is not always an easy task, since no single solution has been developed that covers all types and configurations of input data.

The most general classification of 3D object assessment and reconstruction methods can be done using a contact or non-contact approaches. Contact-based methods are a manual process requiring a large amount of time and skill. Nowadays, the generation of a 3D model

is mainly achieved using non-contact systems based on light waves, in particular *active* or *passive* sensors.

Active sensors provide directly the range data containing the 3D coordinates necessary for the mesh generation phase. Active range scanning technology has been applied to automatically acquire highly accurate geometric data of people. However, generally range-based scanners can only capture a single static pose of a person.

Passive sensors provide for images that need a mathematical model to derive the 3D coordinates. After the measurements, the data must be structured and a consistent polygonal surface generated to build a realistic representation of the modelled objects or scenes. A photo-realistic visualization can afterwards be generated texturing the 3D models with image information.

The use of multiple cameras provides the potential for rapid building of human models in a variety of poses with visual realism. Current methods for image-based modelling of people concentrate on model-based reconstruction with a single camera and provide only the approximate shape and appearance of a person. Reconstruction from multiple cameras can provide improved geometric data and the complex view-dependent appearance of a person to generate more visually realistic computer graphics models. The shape and appearance of a person have been reconstructed from multiple views using multiple-baseline stereo, as well as volumetric reconstruction from image silhouettes and image colour.

3
Camera calibration

3.1 Introduction

Accurate camera calibration is a prerequisite for the extraction of precise and reliable 3D metric information from images. An image camera is usually considered intrinsically calibrated if the focal distance, principal point and lens distortion parameters are known. Complete calibration (known as pose estimation) includes the determination of the above intrinsic parameters together with the extrinsic parameters: translation and rotation.

Most calibration algorithms used in 3D reconstruction are based on perspective projection model. This model, which is described in the next section, starts from the simplest camera model: the pinhole model, which can be ideally modelled as the perspective projection; then, it is progressively generalized through a series of gradations, with the intent to approach the ideal camera model towards a more real one.

Afterwards, the fundamental algorithms considered to be the most well-known in literature [González, 2003; Guerchouche, 2006; Toshio, 2005] for camera calibration and self-calibration are described in detail, using the notation proposed in [Hartley, 2004].

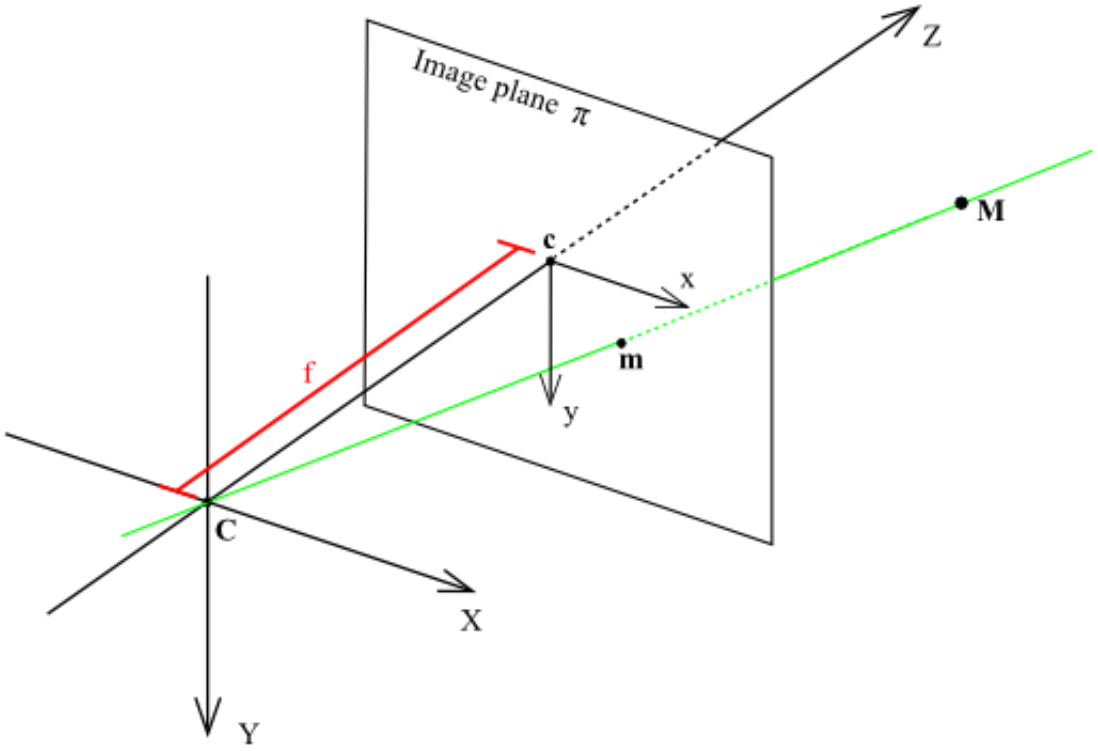

Fig. 3.1 – Pinhole model: point **m**, which belongs to the image plane π, is the 2D projection of 3D world point **M**, according to the 3D Euclidean coordinate system, centred at optical centre **C**; f is the focal distance, i.e., the distance between π and **C**.

3.2 Perspective projection camera model

The pinhole model is usually considered to be the simplest image formation model. However, it is a sufficiently acceptable approach for many applications of Computer Vision [Forsyth, 2003].

In the pinhole model, the origin of the 3D Euclidean coordinate system is the camera's centre of projection or optical centre, $\mathbf{C}$. The 2D image plane or projection plane, π, is located at $Z = f$, where f is the focal distance. The line from the camera centre, perpendicular to the image plane, is called principal axis or principal ray. The point where the principal axis intersects the image plane is called the principal point, $\mathbf{c}$, Fig. 3.1.

The 3D world point $\mathbf{M} = (X, Y, Z)^T$, imaged by a pinhole camera, is related to its image point $\mathbf{m} = (x, y, z)^T$ by the following equations:

$$\begin{cases} x = f\dfrac{X}{Z} = f\dfrac{sX}{sZ} \\ y = f\dfrac{Y}{Z} = f\dfrac{sY}{sZ} \end{cases}. \tag{3.1}$$

Equation (3.1) can be linearly rewritten in the matricial form as:

$$s\mathbf{m} = \mathrm{P}\mathbf{M} \Leftrightarrow \tilde{\mathbf{m}} \cong \mathrm{P}\tilde{\mathbf{M}} \Leftrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.2}$$

where $\tilde{\mathbf{m}}$ and $\tilde{\mathbf{M}}$ are the homogenous coordinates of points $\mathbf{m}$ and $\mathbf{M}$, respectively. The symbol $\cong$ means equality up to a scale factor, s. P is a 3×4 matrix commonly denominated camera projection matrix.

Principal point offset

Equation (3.2) assumed that the origin coordinates in the image plane is at the principal point. As it may not be always true, the principal point coordinates, (c_x, c_y), are introduced:

$$\tilde{\mathbf{m}} \cong \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{M}}. \tag{3.3}$$

Usually the matrix K is defined as a 3×3 sub-matrix of P:

$$\mathrm{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.4}$$

that is known as the camera intrinsic calibration matrix.

Rotation and translation

In general, points in 3D space are expressed in a distinct Euclidean coordinate system from the camera coordinate system, known as the world coordinate system. The two coordinate systems are related via a 3D rigid body transformation, Fig. 3.2, composed by a rotation and a translation:

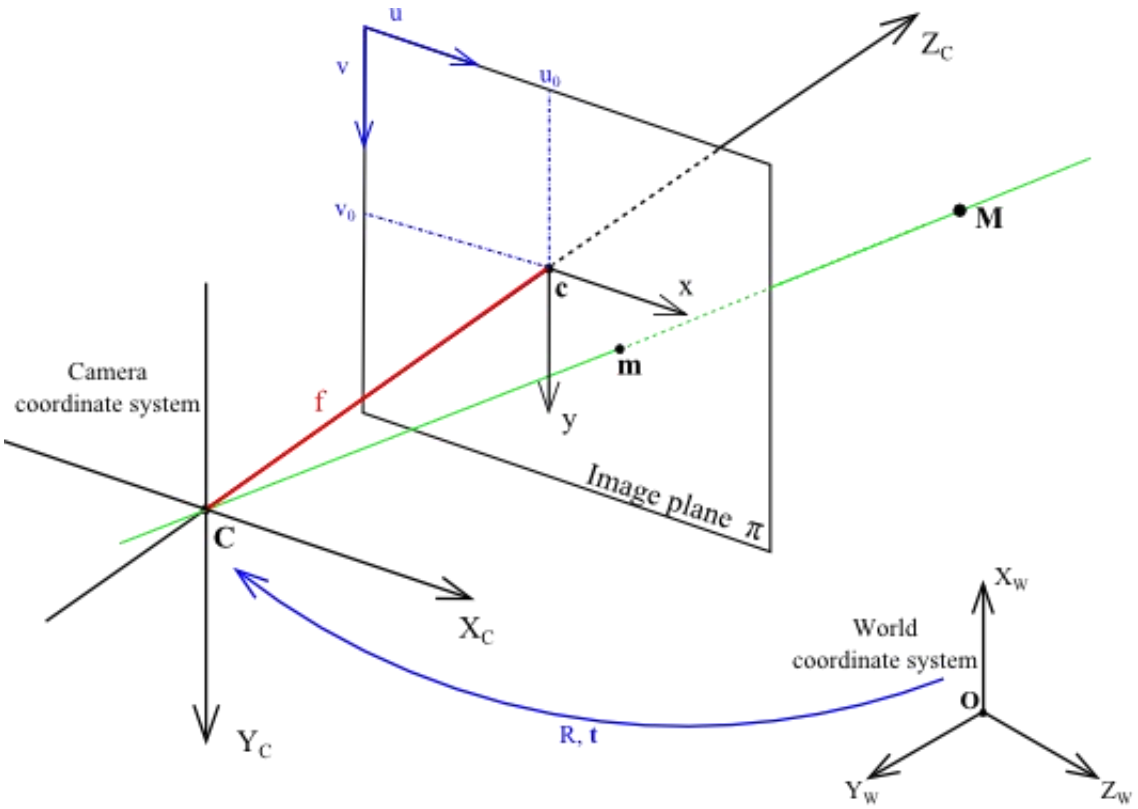

Fig. 3.2 – The geometry of a linear projective camera: matrix R and vector **t** describe the orientation and position (pose) of the camera with respect to the new world coordinate system.

$$\mathbf{M}_C = \mathrm{R}\mathbf{M}_W + \mathbf{t}\,. \tag{3.5}$$

In Equation (3.5), $\mathbf{M}_C = (x_C, y_C, z_C)^T$ and $\mathbf{M}_W = (x_W, y_W, z_W)^T$ are the inhomogeneous vectors of a 3D point, in the camera and world coordinate system, respectively; R is a 3×3 orthonormal rotation matrix, representing the orientation of the camera coordinate system, and **t** is a 3-vector representing the origin of the world coordinate system expressed in the camera coordinate system. These last two elements are also called extrinsic parameters and define the camera pose.

Conversion of units

The pinhole camera model assumes that image coordinates are in Euclidean coordinates, with equal scales. In the case of CCD (Charged-Coupled Device) cameras, there is an additional possibility of having non-square pixels. Using again the notation proposed in [Hartley, 2004], if the number of pixels per unit distance in terms of image coordinates are (m_x, m_y), then the transformation from the world Euclidean coordinates to pixel coordinates is obtained by multiplying Equation (3.4) by $\mathrm{diag}(m_x, m_y, 1)$, yielding:

$$\mathrm{K}=\begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.6}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the camera's focal length in pixel coordinates and $u_0 = c_x m_x$ and $v_0 = c_y m_y$ represent the principal point also in pixel coordinates.

Also, the angle between the inner vertices of a pixel may not be always equal to 90º, due to some optical manufacturing errors. Without loss of generality, one can assume the image u axis aligned with the canonical u axis and the image v axis skewed by θ, Fig. 3.3. Thus, Equation (3.6) takes the following form:

$$\mathrm{K}=\begin{bmatrix} \alpha_x & \cot\theta & u_0 \\ 0 & \alpha_y / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.7}$$

Lens distortion

Real camera lenses do not follow a linear model, like the one that has been assumed. As such, a 3D world point **M**, its 2D image point **m** and the camera centre **C** are not collinear, world lines are not imaged as lines, and so on.

The most important deviation is the radial distortion, but many camera models also consider tangential distortion, Fig. 3.4. To overcome this problem, a correction must be carried out during the initial projection of the world onto the image plane. Using the notation proposed in [Heikkilä, 1996], radial distortion can be modelled as:

$$\begin{cases} x_r = x_p D(r) \\ y_r = y_p D(r) \end{cases}, \tag{3.8}$$

with $D(r) = k_1 r^2 + k_2 r^4 + \ldots + k_n r^{2n}$, where $r = \sqrt{x_p^2 + y_p^2}$ and k_i are the coefficients of radial distortion.

Tangential distortion is due to imperfect centring of the lens. Again, using the notation proposed in [Heikkilä, 1996], it can be modelled as:

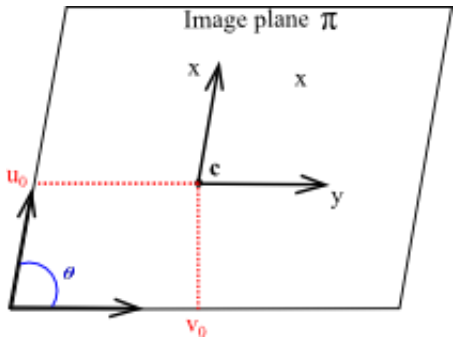

Fig. 3.3 – Representation of the skew angle, θ, of the image coordinate system and the coordinates of the principal point **c**.

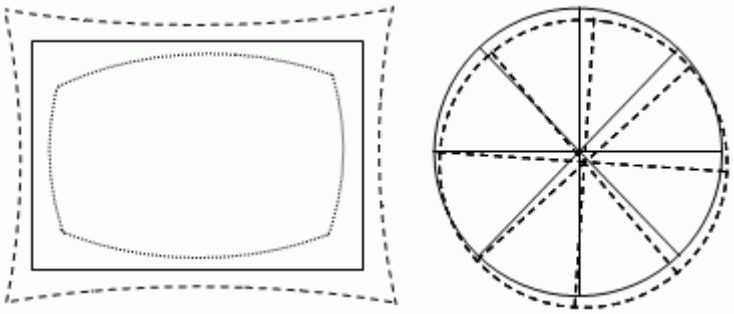

Fig. 3.4 – On the left: effects of radial distortion, where the solid line represents the image with no distortion, the dashed line represents the pincushion effect and the dotted line represents the barrel effect. On the right: effects of tangential distortion; where the solid line is the image with no distortion and the dashed line represents the tangential distortion.

$$\begin{cases} x_t = 2p_1 x_p y_p + p_2(r^2 + 2x_p{}^2) \\ y_t = p_1(r^2 + 2y_p{}^2) + 2p_2 x_p y_p \end{cases}, \tag{3.9}$$

with p_1 and p_2 being the coefficients of tangential distortion.

Correcting the measured (distorted) image points, means to add the modelled coordinates:

$$\begin{cases} x_d = x_p + x_r + x_t \\ y_d = y_p + y_r + y_t \end{cases}. \tag{3.10}$$

Projection matrix

Combining all the transformations that have been presented, excepting for the lens distortion, leads to the following equation:

$$\tilde{\mathbf{m}} \cong \mathrm{P}\tilde{\mathbf{M}} \Leftrightarrow \tilde{\mathbf{m}} \cong \mathrm{K}[\mathrm{R} \mid \mathbf{t}]\tilde{\mathbf{M}}, \tag{3.11}$$

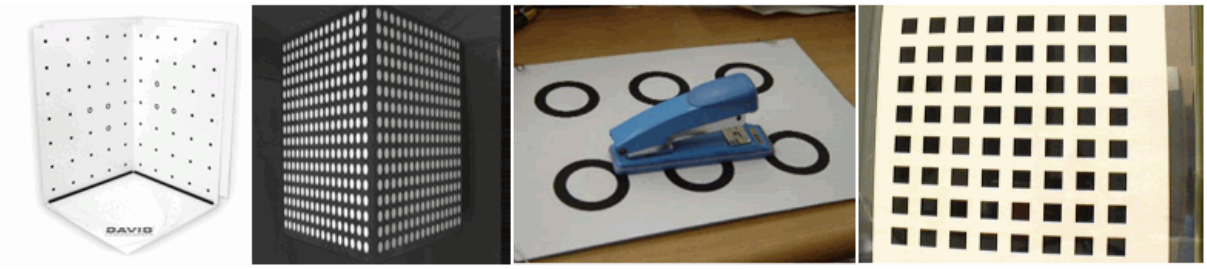

Fig. 3.5 – Some examples of calibration patterns. From left to right: 3D pattern recommended for the *DAVID LaserScanner* system, [DAVID, 2012], consisted by two planar panels with circular patterns; 3D pattern with printed circles used in [Heikkilä, 2000]; planar pattern with concentric circles used in [Ying, 2008]; most commonly used camera calibration pattern, the planar black and white chessboard pattern, used for example in [Qiuyu, 2012].

where P is the 3×4 homogeneous camera projection matrix, that relates the pixel homogeneous coordinates, $\tilde{\mathbf{m}}=(u,v,1)^T$, and the world homogeneous coordinates, $\tilde{\mathbf{M}}=(x,y,z,1)^T$. Matrix K contains the camera's intrinsic parameters and matrix R and vector $\mathbf{t}$ represent the camera's extrinsic parameters.

3.3 Camera calibration methods

Usual calibration methods are applied off-line and require the use of images of a special calibration object (also denominated target), with well-known 3D coordinates, Fig. 3.5. Most commonly used calibration objects are 3D or planar flat plates with a regular pattern marked on the surfaces; others are specifically designed when it is considered camera/object relative motion.

High accuracy can be obtained with off-line camera calibration methods. However, they cannot cope when the camera parameters change during the normal operation (e.g., zooming in or out), or when trying to reconstruct a scene from a pre-acquired image sequence in which the camera's model cannot be obtained, due to, for example, the inexistence of a calibration pattern in the acquired images.

Calibration methods can be classified according to a number of criteria [González, 2003], such as:

- **Linear *versus* nonlinear**
  - **linear methods** (e.g. [Abdel-Aziz, 1971], [Hall, 1982], [Ito, 1991]): they determine the camera parameters using linear equation systems, like the least squares method, [Chapra, 1988]. These methods are fast and easy to implement; however, they are unsuitable when good accuracy is required or when lens distortion needs to be modelled.
  - **non-linear methods** (e.g. [Tsai, 1987], [Weng, 1992], [Faugeras, 1993], [Heikkilä, 1997], [Zhang, 2000]): the camera's parameters are obtained through iterative methods. Although slower than linear calibration methods, the advantage of these is that they can estimate nearly any camera model and that the accuracy usually increases with the number of iterations. However, they require a good initial estimate, which is typically obtained by using linear methods, in order to guarantee convergence.
- **Implicit *versus* explicit**
  - **implicit methods** (e.g. [Faugeras, 1993], [Zhang, 1998], [Ahmed, 1999b]): implicit calibration is the process of calibrating a camera without explicitly computing all its physical parameters (as focal distance and camera's optical centre). Generally, the transformation matrices that contain the set of all parameters are obtained by these methods.
  - **explicit methods** (e.g. [Tsai, 1987], [Heikkilä, 1997], [Batista, 1998]): with these calibration methods, all camera's model parameters are obtained.
- **One single optimization *versus* multi-step**
  - **one single step** (e.g. , [Faugeras, 1993], [Ahmed, 1999b]): for each cycle of the resolution process, all camera's parameters are updated.
  - **multi-step** (e.g. [Tsai, 1987], [Weng, 1992], [Heikkilä, 1997], [Batista, 1998]): in each step, a distinct subset of the camera's model parameters is obtained, using the estimates for those parameters previously determined; good initial estimates can be quickly determined assuming some simplifications on the camera's model, which will be progressively improved in next optimization steps.

- **Planar *versus* non-planar**
  - **planar** (e.g. [Tsai, 1987], [Batista, 1998]): when all the 3D points of a calibration pattern are on the same plane; thus, the inaccuracy of the world points' coordinates is reduced, since one of them is null.
  - **non-planar**: within this group, are methods that require to know the relation between the different planes (usually, it is used a dihedron, i.e., planes making an angle of 90º between each other – as in [Tsai, 1987], [Faugeras, 1993], [Heikkilä, 1997] and [Ahmed, 1999b]) and methods that do not need to know the relation between the planes' positions (usually, image acquisitions of the same planar pattern in different space positions are used, as in [Yu, 1998] and [Zhang, 1998]).

3.3.1 DLT method

The DLT – Direct Linear Transformation – was proposed in [Abdel-Aziz, 1971]. It is the simplest method for camera calibration, but remains a classical and very well-known linear calibration method.

Usually, it is used as an initial approximation step in many iterative calibration methods, to obtain the camera projection matrix. It assumes a perfect projective camera and a set of non-planar pattern calibration points.

The DLT method requires the knowledge on the 3D to 2D correspondences of the calibration points. It is based on the pinhole camera model, and it ignores the nonlinear radial and tangential distortion components. For each correspondence $\mathbf{m}_i \leftrightarrow \mathbf{M}_i$, Equation (3.2) can be rewritten in terms of the cross product:

$$\mathbf{m}_i \times \mathrm{P}\mathbf{M}_i = \mathbf{0}, \tag{3.12}$$

which enables the deriving of a simple linear solution for camera projection matrix P.

Three equations can be obtained from the left side of Equation (3.12); but, eliminating the unknown scale factor, each point correspondence gives origin to just two equations:

$$\begin{bmatrix} \mathbf{0}^T & -\mathbf{M}_i^T & v_i\mathbf{M}_i^T \\ \mathbf{M}_i^T & \mathbf{0}^T & -u_i\mathbf{M}_i^T \end{bmatrix} \begin{bmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{bmatrix} = \mathbf{0}, \tag{3.13}$$

with $\mathbf{p} = \begin{bmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{bmatrix} = [p_{11} \quad p_{12} \quad p_{13} \quad p_{14} \quad p_{21} \quad p_{22} \quad p_{23} \quad p_{24} \quad p_{31} \quad p_{32} \quad p_{33} \quad p_{34}]^T$.

For the n non-planar correspondence points, the above relationship can be described by the system:

$$\mathrm{A}.\mathbf{p} = \mathbf{0}, \tag{3.14}$$

with $\mathbf{0}$ being a $2n \times 1$ null vector and matrix A obtained by stacking up the two equations in Equation (3.13) for each 3D to 2D point correspondence.

System of equations (3.14) can be solved by applying the least squares regression method [Chapra, 1988]. Thus, it must be over-determined; that is, the number of equations must be higher than the number of unknowns. Since the system represents two equations with 12 unknowns p_{ij}, applying these equations to $n > 6$ non-planar known locations in the object space and their corresponding image points, the unknown transformation coefficients p_{ij} can be computed.

In order to avoid a trivial solution, like $p_{ij} = 0$, a proper normalization must be addressed. The constraint $p_{34} = 1$ was used in [Abdel-Aziz, 1971]. The problem with this normalization is that a singularity is introduced if the correct value of p_{34} is close to zero. Instead of $p_{34} = 1$, in [Faugeras, 1987] it was suggested to introduce the constraint $\|p_{31} \quad p_{32} \quad p_{33}\| = 1$ that is singularity free.

Parameters p_{ij} have no physical meaning, and thus DLT can be considered an implicit camera calibration method. There are techniques for extracting some of the physical camera parameters from the projection matrix P (e.g., [Melen, 1994]), but few are able to solve all the intrinsic (excluding the lens distortion) and extrinsic parameters.

The main advantage of DLT is that it is very fast (requires only one image) and simple, because only linear equations need to be solved. Its drawbacks are: results are sensitive to errors, and it does not consider camera distortion, which may severely affect the accuracy of posterior measurements or reconstructions.

3.3.2 Tsai's method

Presented in [Tsai, 1986; Tsai, 1987], Tsai's calibration method is one of the most referenced calibration methods in literature, and probably it was the first to include distortion in the camera model parameters: it considers radial distortion, modelled by a single coefficient.

Tsai proposed two versions of his calibration method: a single view algorithm and an adaptation to be used with multiple views of the calibration pattern. Tsai's method is an explicit two-step approach:

- First, it tries to obtain estimates of as many camera's parameters as possible using a linear least squares method. In this initial step, constraints between parameters are not enforced, and what is minimized is not the error in the image plane, but a quantity that simplifies the analysis and leads to linear equations. This does not affect the calibration result, since these estimated parameter values are used only as starting values for the second and final optimization step.
- Secondly, the remainder parameters are obtained using a non-linear optimization method that finds the best fit between the observed image points and those predicted from the target model. Parameters estimated in the first step are refined in this process.

Tsai's method provides different versions for planar and non-planar calibration patterns. Accurate planar targets are easier to make and maintain than 3D targets, but limit calibration in ways that will become apparent. Since Tsai uses a distinct camera model than the one presented in section 3.2, its differences are briefly described in the following.

Tsai's camera model

In [Tsai, 1986; Tsai, 1987], the transformation from 3D world coordinates into camera coordinates is based on a Radial Alignment Constraint (RAC), Fig. 3.6.

According to this model, only radial distortion is taken into consideration and a correction is due, after the perspective transformation (Equation (3.1)), such that:

$$\begin{cases} x_D + D_x = x_U \\ y_D + D_y = y_U \end{cases}, \tag{3.15}$$

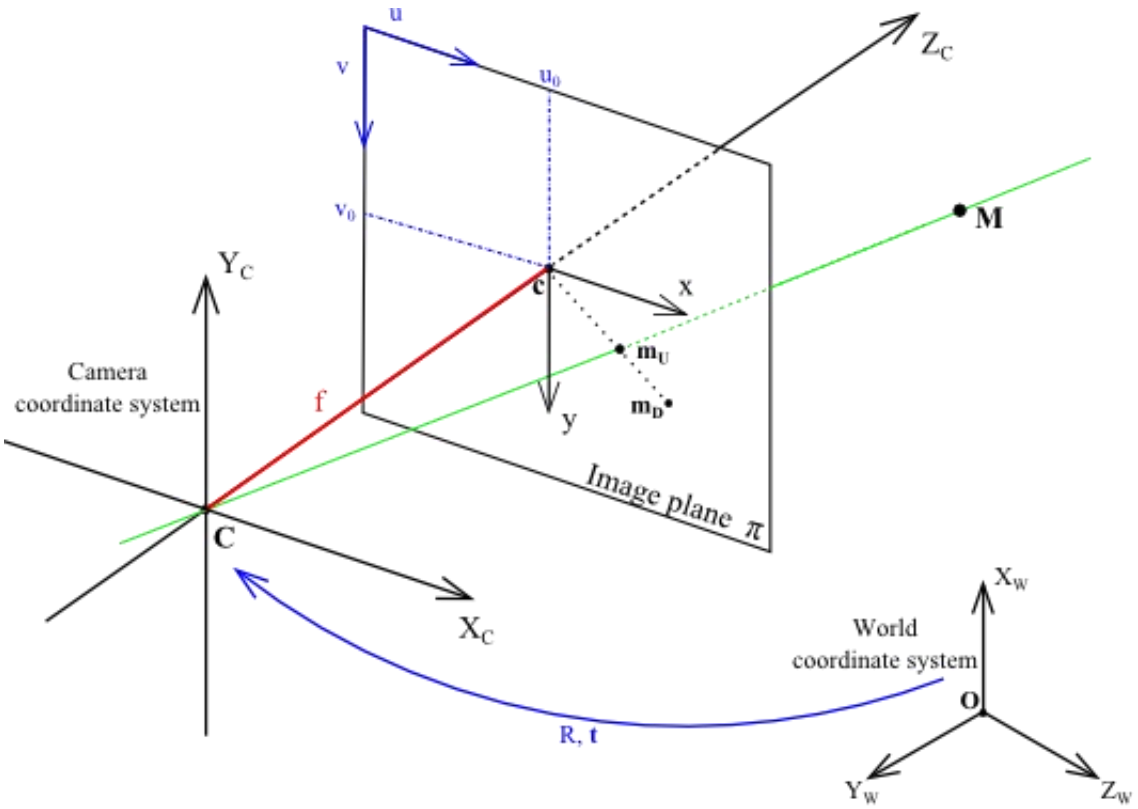

Fig. 3.6 – Tsai's camera model, which includes perspective projection and radial distortion.

where $\mathbf{m}_D = (x_D, y_D)$ are the distorted image coordinates on the image plane, π, and factors D_x and D_y are given by:

$$\begin{cases} D_x = x_D k_1 r^2 \\ D_y = y_D k_1 r^2 \end{cases}, \tag{3.16}$$

with $r = \sqrt{x_D{}^2 + y_D{}^2}$ and k_1 as the first term of the radial distortion.

The transformation from real (distorted) image coordinates, $\mathbf{m}_D$, to computer (pixel) image coordinates, $\mathbf{m}_P = (u, v)$, is given by:

$$\begin{cases} u = s_x (d_x^{'})^{-1} x_D + u_0 \\ v = (d_y^{'})^{-1} y_D + v_0 \end{cases}, \tag{3.17}$$

where s_x is the horizontal scale factor and $(d_x^{'}, d_y^{'})$ are the pixels' dimensions (in mm). In typical CCD or CMOS (Complementary Metal Oxide Semiconductor) cameras, the frame grabber sampling of the initially sensor signal is different for the horizontal and vertical directions. The sampling in the horizontal direction is typically not equal to the spacing of

sensor cells, and is not known accurately. In general, the horizontal spacing between pixels in the sampled image does not correspond to the horizontal spacing between cells in the image sensor. In contrast, the sampling is controlled in vertical direction by the spacing of rows of sensor cells. Some digital cameras avoid this problem, but many cameras, particularly cheaper ones, do not. In those cases, the ratio of picture cell size in the horizontal and in the vertical direction is not known a priori from the dimensions of the sensor cells and needs to be determined. Therefore, the extra scaling parameter, s_x, needs to be recovered as part of the camera calibration process.

First step – non-planar pattern

All pattern points' coordinates in pixels (u, v) are converted into Euclidean coordinates (x_D, y_D), using Equation (3.17) and assuming that there is no lens distortion. In this step, it is considered that $s_x = 1$ and that (u_0, v_0) are equal to the geometrical image centre.

Given the calculated image coordinates (x_D, y_D) and corresponding world coordinates $\mathbf{M}_W = (x_W, y_W, z_W)$, the following linear system can be built, still considering the camera distortion null:

$$\begin{bmatrix} y_D x_W & y_D y_W & y_D z_W & y_D & -x_D x_W & -x_D y_W & -x_D z_W & -x_D \end{bmatrix} \begin{bmatrix} s_x r_{11} \\ s_x r_{12} \\ s_x r_{13} \\ s_x t_1 \\ r_{21} \\ r_{22} \\ r_{23} \\ t_2 \end{bmatrix} = 0, \qquad (3.18)$$

where r_{ij} and t_i are the elements of the rotation matrix R and translation vector $\mathbf{t}$, respectively.

Since the equations in (3.18) are homogeneous, there is an unknown scale factor associated to the solution of above system. One way to convert them into inhomogeneous is to set one unknown equal to one; for example, $t_2 = 1$. If the number of points n is superior to seven, this system of equations can be solved by applying a simple least squares regression method [Chapra, 1988].

After obtaining a solution for the eight unknowns of system (3.18) (seven unknowns, plus the one that was set equal to one), the correct scale factor, the proportion factor s_x, all elements of the rotation matrix r_{ij} and the first two elements of translation vector $\mathbf{t}$ can be found by noting that rotation matrix R is orthonormal.

First step – planar pattern

With a planar pattern with $z_W = 0$, the system of equations (3.18) is re-arranged:

$$[y_D x_W \quad y_D y_W \quad y_D \quad -x_D x_W \quad -x_D y_W \quad -x_D] \begin{bmatrix} r_{11} \\ r_{12} \\ t_1 \\ r_{21} \\ r_{22} \\ t_2 \end{bmatrix} = 0 . \tag{3.19}$$

As in the non-planar case, there is an unknown scale factor because the equations are homogeneous. So, one solution is to set one unknown to one; for example, $t_2 = 1$.

Once again, after obtaining a solution for the six unknowns of system (3.19) (five unknowns, plus the one set equal to one), the correct scale factor, all elements of the rotation matrix r_{ij} and the first two elements of translation vector $\mathbf{t}$ can be determined by noting that rotation matrix R is orthonormal.

Second step

The focal distance f and the translation along z, t_3, are determined from the following linear equation system:

$$[y_C \quad -y_D] \begin{bmatrix} f \\ t_3 \end{bmatrix} = z_r y_D , \tag{3.20}$$

with $y_C = r_{21} x_W + r_{22} y_W + r_{23} z_W + t_2$ and $z_r = r_{31} x_W + r_{32} y_W + r_{33} z_W$. This system can be solved by applying a least squares regression method [Chapra, 1988].

Since accuracy improves with the depth range of pattern points, if a planar pattern is used, it means that it must not be parallel to the image plane.

At this point, all elements of rotation matrix, R, and translation vector, **t**, are estimated, as well as the focal distance, f. In order to find the first coefficient for radial distortion, k_1, and to refine the values for f and t_3, a non-linear optimization is processed by using, for example, the Levenberg-Marquardt algorithm [Moré, 1977], in equation:

$$d_y' v + d_y' v k_1 r^2 = f \frac{r_{21} x_W + r_{22} y_W + r_{23} z_W + t_2}{r_{31} x_W + r_{32} y_W + r_{33} z_W + t_3}, \tag{3.21}$$

where $r = \sqrt{(s_x^{-1} d_x' u)^2 + (d_y' v)^2}$.

Main advantage of this method is that it includes a coefficient for the radial distortion and simple equation systems of easy resolution. Its drawbacks are that parameters determined in the first step are not refined in the second. Thus, matrix R and parameters t_1, t_2 and s_x will contain the error introduced by discarding the distortion effect in the first linear step. Another disadvantage of the Tsai's calibration method is that the radial component of the image points are discarded completely and only the perpendicular components are used. As a consequence, the RAC constraint is very sensitive to noise, especially when the angle between the optical axis and the calibration plane is small. According to [Batista, 1998], the angle of incidence between the optical axis of the camera and the calibration plane must be at least 30 degrees when a coplanar set of control points is used.

3.3.3 Zhang's method

Described in [Zhang, 1998; Zhang, 2000], Zhang's calibration method performs in two-steps: first, an analytic solution to solve the camera's intrinsic and extrinsic parameters is proposed, and then the best approximation using a minimization method is obtained.

In contrast with the Tsai's method, Zhang's method requires at least three different views of a planar calibration pattern. However, only two images are necessary, if some intrinsic parameters are known. Lens distortion is modelled using two radial coefficients. It is an implicit method, i.e., an intrinsic matrix is obtained whose elements are functions of the camera's intrinsic parameters.

First step

Projection matrix P is calculated from each one of the n images used. First, an initial estimate is obtained using a linear approach, except for that z-coordinate of world points is zero. Let's denote the i^{th} column of the rotation matrix R by $\mathbf{r}_i$. From Equation (3.11), one has:

$$\tilde{\mathbf{m}} \cong \mathrm{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathrm{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{3.22}$$

Therefore, a world model point, $\mathbf{M}$, and its image coordinates, $\mathbf{m}$, are related by a 3×3 homography matrix, H:

$$\tilde{\mathbf{m}} \cong \mathrm{H}\tilde{\mathbf{M}}, \tag{3.23}$$

with $\mathrm{H} = \mathrm{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]$.

Given that the columns of the rotation matrix are orthonormal, the following constraints can be derived:

$$\begin{aligned} \mathbf{h}_1^T \mathrm{K}^{-T}\mathrm{K}^{-1}\mathbf{h}_2 &= 0 \\ \mathbf{h}_1^T \mathrm{K}^{-T}\mathrm{K}^{-1}\mathbf{h}_1 &= \mathbf{h}_2^T \mathrm{K}^{-T}\mathrm{K}^{-1}\mathbf{h}_2 \end{aligned}. \tag{3.24}$$

Because a homography has 8 degrees of freedom and there are 6 extrinsic parameters (3 for the rotation and 3 for the translation), only 2 constraints on the intrinsic parameters can be obtained. It should be noted that the symmetric matrix $\mathrm{K}^{-T}\mathrm{K}^{-1}$ actually describes the image of the absolute conic B [Faugeras, 1992a]:

$$\mathrm{K}^{-T}\mathrm{K}^{-1} = \mathrm{B} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}. \tag{3.25}$$

Since matrix B is symmetric, only six elements need to be determined. Converting this matrix into a 6D vector $\mathbf{b} = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]^T$ and considering $\mathbf{h}_i$ as the i^{th} column of the homography matrix H, one has:

$$\mathbf{h}_i^T \mathrm{B}\mathbf{h}_j = \mathbf{v}_{ij}\mathbf{b}, \tag{3.26}$$

where $\mathbf{v}_{ij} = [h_{1i}h_{1j} \quad h_{1i}h_{2j} + h_{2i}h_{1j} \quad h_{2i}h_{2j} \quad h_{3i}h_{1j} + h_{1i}h_{3j} \quad h_{3i}h_{2j} + h_{2i}h_{3j} \quad h_{3i}h_{3j}]$. Therefore, the two fundamental constraints (3.24) can be rewritten as two homogeneous equations in $\mathbf{b}$:

$$\begin{bmatrix} v_{12} \\ (v_{11} - v_{22}) \end{bmatrix} \mathbf{b} = 0 . \tag{3.27}$$

Thus, for n images of the calibration pattern, the n equivalent equations can be stacked into a system in the form:

$$\mathrm{V}\mathbf{b} = 0, \tag{3.28}$$

where V is a $2n \times 6$ matrix. Because this system has 6 unknowns, $n \geq 3$ is enough to obtain a unique solution, defined up to a scale factor. However, imposing the skewless constraint $\gamma = 90^{\circ} \Leftrightarrow s = 0$, the system can be solved with $n = 2$. The solution to system (3.28) is well known as the eigenvector of $\mathrm{V}^{\mathrm{T}}\mathrm{V}$ associated with the smallest eigenvalue and can be found, for example, via SVD (Singular Value Decomposition) of Q [Golub, 1983].

From matrix B, the intrinsic parameters and the scale factor s can be computed:

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2), \\ s &= B_{33} - (B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23}))/B_{11}, \\ \alpha_x &= \sqrt{s/B_{11}}, \\ \alpha_y &= \sqrt{sB_{11}/(B_{11}B_{22} - B_{12}^2)}, \\ \gamma &= -B_{12}\alpha_x{}^2\alpha_y/s, \\ u_0 &= v_0 s/\alpha_y - B_{13}\alpha_x{}^2/s. \end{aligned} \tag{3.29}$$

Once matrix K is known, the extrinsic parameters for each image are computed using:

$$\begin{aligned} \mathbf{r_1} &= \lambda \mathrm{K}^{-1}\mathbf{h_1}, \\ \mathbf{r_2} &= \lambda \mathrm{K}^{-1}\mathbf{h_2}, \\ \mathbf{r_3} &= \mathbf{r_1} \times \mathbf{r_2}, \\ \mathbf{t} &= \lambda \mathrm{K}^{-1}\mathbf{h_3}, \end{aligned} \tag{3.30}$$

where $\lambda = 1/\left\|\mathrm{K}^{-1}\mathbf{h_1}\right\| = 1/\left\|\mathrm{K}^{-1}\mathbf{h_2}\right\|$.

Second step

All camera's parameters are refined using an optimization procedure that minimizes the geometric image error; i.e., the distance between the n 2D image points, $\mathbf{m}_i$, and their

projections onto the image, using all obtained calibration parameters, i.e. the intrinsic, extrinsic and distortion coefficients, and the 3D world calibration points, $\mathbf{M}_i$. This is a non-linear problem, that can be solved using a Levenberg-Marquardt algorithm [Moré, 1977] to refine matrix P, by minimizing:

$$\sum_{i=1}^{n} \left\| \mathbf{m}_i - \hat{\mathbf{m}}_i \right\|^2 , \tag{3.31}$$

with $\mathbf{m}_i$ the real image coordinates of all pattern points and

$$\hat{\mathbf{m}}_i = \frac{1}{\mathbf{p}_3^T \mathbf{M}_i} \begin{bmatrix} \mathbf{p}_1^T \mathbf{M}_i \\ \mathbf{p}_2^T \mathbf{M}_i \end{bmatrix}, \tag{3.32}$$

where $\mathbf{p}_1 = [p_{11} \quad p_{12} \quad p_{13}]^T$, $\mathbf{p}_2 = [p_{21} \quad p_{22} \quad p_{23}]^T$ and $\mathbf{p}_3 = [p_{31} \quad p_{32} \quad p_{33}]^T$.

Finally, initial estimates of the first two coefficients of radial distortion, (k_1, k_2), are calculated using a least squares method [Chapra, 1988], solving the following equations for each image point:

$$\begin{bmatrix} (u_p - u_0)r & (u_p - u_0)r^2 \\ (v_p - v_0)r & (v_p - v_0)r^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} u_d - u_p \\ v_d - v_p \end{bmatrix}, \tag{3.33}$$

with $r = (x_C^2 + y_C^2) / z_C^2$, (u_d, v_d) the distorted (real) pixel coordinates and (u_p, v_p) the ideal pixel coordinates calculated using the previously determined camera's parameters.

This method differs from others, because it corrects the camera distortion before the points are projected onto the image plane; i.e., its coordinates are not affected by focal distance. Thus, the obtained radial distortion coefficients are not comparable to the ones obtained from other calibration methods.

3.3.4 Heikkilä's method

Presented in [Heikkilä, 1996; Heikkilä, 1997], Heikkilä's method is composed by four steps, and compensates for radial and tangential distortion by using circular features. Therefore, Heikkilä's method uses a non-planar calibration pattern with circles as control points.

First step

It uses the DLT calibration method, described in Section 3.3.1, to obtain an initial approximation for the camera's extrinsic parameters, focal distance and principal point.

The camera's model is the same as in Tsai's method (Fig. 3.6). The physical parameters, from the encountered solution for vector **p** (Equation (3.14)), are extracted by using QR decomposition [Melen, 1994]:

$$\mathrm{P} = \lambda \mathrm{V}^{-1}\mathrm{B}^{-1}\mathrm{FRT}, \tag{3.34}$$

where R and T are the rotation and translation matrices, respectively, and λ is a scaling factor. Matrices V, B and F contain the intrinsic parameters: principal point (u_0, v_0), focal distance f and coefficients for linear distortion (b_1, b_2):

$$\mathrm{V} = \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 0 & -v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{B} = \begin{bmatrix} 1+b_1 & b_2 & 0 \\ b_2 & 1-b_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{F} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.35}$$

The linear distortion correction is used to compensate for the orthogonality errors of the image coordinate axes, which are very close to null for CCD cameras.

Second step

An iterative non-linear method adjusts the previously obtained parameters and determines the remaining ones: the proportion factor and radial and tangential distortion coefficients. To do so, a Levenberg-Marquardt algorithm [Moré, 1977] is used to minimize the following function:

$$\sum_{i=1}^{n} \left[(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 \right], \tag{3.36}$$

where (u_i, v_i) are the real image coordinates and $(\hat{u}_i, \hat{v}_i)$ the estimated coordinates using the calculated parameters, for all n points.

In this method, the camera's intrinsic matrix K is slightly different from the one presented in Section 3.2:

$$\mathrm{K} = \begin{bmatrix} fD_u s_u & 0 & u_0 \\ 0 & fD_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.37}$$

where (D_u, D_v) are the coefficients needed to change from metric units to pixels and s_u is a scale factor.

Third step

In this step, a process to correct asymmetric projection distortion is applied. Thus, perspective projection distorts the shape of the circular features in the image plane depending on the angle and displacement between the object surface and the image plane.

This method corrects the image coordinates of the circle centres (U_i, V_i) using the following equation:

$$\begin{aligned} U_i' &= U_i - D_u s_u (\tilde{u}_{c,i} - \tilde{u}_{0,i}) \\ V_i' &= V_i - D_v (\tilde{v}_{c,i} - \tilde{v}_{0,i}) \end{aligned}, \tag{3.38}$$

where $(\tilde{u}_c, \tilde{v}_c)$ are the image coordinates of the ellipse centre, since the usual image projection of a circle is an ellipse, and $(\tilde{u}_0, \tilde{v}_0)$ the estimated projection of the circle centre using the previously calculated parameters.

Fourth step

This final step focuses on the back-projection problem. Image correction is performed by using a new implicit model that interpolates the correct image points based on the physical camera's parameters derived from previous steps.

3.3.5 Methods based on Neural Networks

Artificial neural networks have been commonly used to solve some Computer Vision problems, such as static stereo, computation of optical flow, image restoration and object recognition [Zhou, 1992].

Neural networks are parallel information processing systems consisting of a number of simple neurons (also called nodes or units), which are organized in layers that are connected by links. The artificial neural networks imitate the highly interconnected structures of the brain and the nervous system of animals and humans whereby the neurons correspond to the brain cells, and the links are equivalent to the axons in biology [Dieterle, 2003].

The existent literature on camera calibration using neural networks is quite extensive, with some of the earliest works dating back to the early 1990s (e.g., [Wen, 1991], [Lynch, 1992], [Choi, 1994]). These methods used neural networks to learn the mapping from 3D world to 2D images without specifying the camera's model, e.g. [Lynch, 1992], or as an additional stage to improve the performance of other existing methods, e.g. [Wen, 1991] and [Choi, 1994]. In [Ahmed, 1999b] the term neurocalibration was proposed not only to learn the mapping from 3D points to 2D image pixels but also to extract the projection matrix and camera's parameters.

Ahmed's method

In [Ahmed, 1999a], a method was presented that goes beyond the previous ones by having a neural network to specify the camera's intrinsic and extrinsic parameters. A priori, the camera's distortion is not considered. It does not require an initial approximation of the camera's parameters; however, in [Ahmed, 1999b], the experimental results are based on an initial approximation obtained using a linear method to accelerate the calibration process.

Ahmed's calibration method trains a two-layer feedforward neural network, Fig. 3.7. The network intends to obtain the projection matrix, which transforms the 3D pattern calibration points (inputs) into image pixel coordinates (outputs). Two weight matrices, V and W, are associated to each layer. Matrix V (first level) relates to the extrinsic parameters and W to the intrinsic parameters (output level):

$$\mathrm{V}=\begin{bmatrix}\mathrm{R} & \mathbf{t}\\ \mathbf{0}_3^T & 1\end{bmatrix},\quad \mathrm{W}=\begin{bmatrix}\alpha_x & -\alpha_x\cot\theta & u_0 & 0\\ 0 & \alpha_y/\sin\theta & v_0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}. \tag{3.39}$$

For any input pattern point i $(1\le i\le N)$, the input vector is formed as $\mathbf{M}_i=[X_i \quad Y_i \quad Z_i \quad 1]^T$. The hidden neuron j from the first level produces the output:

$$Y_j=\sum_{l=1}^{4}V_{jl}M_{il}\,, \tag{3.40}$$

where i is the number of input coordinates $\mathbf{M}$. The output neuron k is given by:

$$O_{ik}=\sum_{j=1}^{4}\gamma_i W_{kj}Y_j\,, \tag{3.41}$$

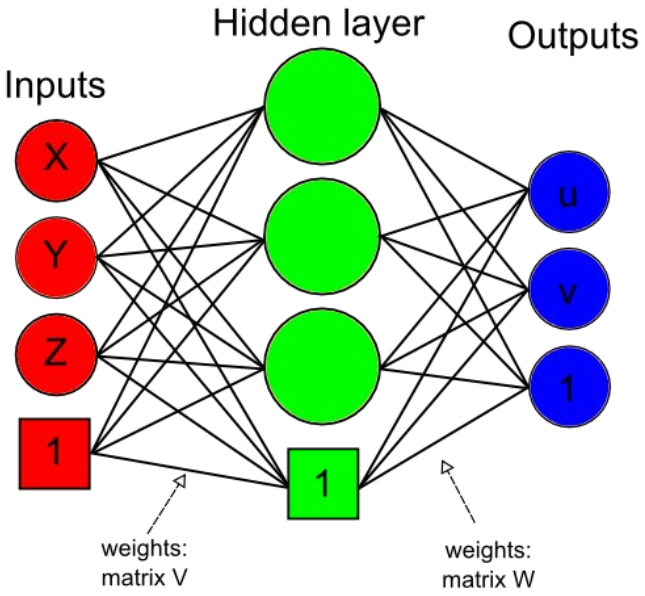

Fig. 3.7 – Ahmed's neurocalibration network, with a 4-4-3 topology: four inputs (three neurons plus one augmented fixed at 1), four neurons in the hidden layer (three plus one dummy) and three outputs.

where $o_i = [\hat{u}_i \quad \hat{v}_i \quad \hat{w}_i]^T$ are the image projections homogeneous coordinates, with $\hat{w}$ a scale factor. All points have a distinct factor γ, that is initially set equal to 1 (one). The weights of the network V_{jl} and W_{kj} are initialized at random values in the range [-1:1].

The network weights and γ_i are updated according to a gradient descent rule applied to the following error measure equation:

$$E = \sum_{i=1}^{N} (\gamma_i o_{i1} - u_i)^2 + (\gamma_i o_{i2} - v_i)^2 + (\gamma_i o_{i3} - 1)^2 \ . \tag{3.42}$$

For simplicity of the network learning phase, input and output coordinates must be normalized. Therefore, 2D and 3D points must be divided by their norm, s_1 and s_2, respectively. After training the network, the projection matrix P can be obtained using:

$$\mathrm{P} = \mathrm{S}_1 \mathrm{W} \mathrm{V} \mathrm{S}_2 \,, \tag{3.43}$$

with $\mathrm{S}_1 = \mathrm{diag}(s_1, s_1, 1)$ and $\mathrm{S}_2 = \mathrm{diag}(s_2^{-1}, s_2^{-1}, s_2^{-1}, 1)$.

Moreover, to go beyond just obtaining the projection matrix P, each network weight is mapped to one camera's parameter. First, some elements of intrinsic matrix W have values zero or one; therefore, the corresponding weights W_{kj} are set accordingly and are not allowed to update during network learning. Second, the neuronal net is trained using the

orthogonality constraint of the rotation matrix and by minimizing the disparity of each point image projection; thus, the error measure per input as additional terms and is set as:

$$E_{tot} = E_{2D} + \beta E_{orth}, \tag{3.44}$$

with β a small positive weighting factor,

$$E_{2D} = \frac{1}{2}\sum_{i=1}^{N}\left[(\hat{u}_i - u_i)^2 + (\hat{v}_i - v_i)^2 + (\hat{w}_i - 1)^2\right] = \frac{1}{2}\sum_{k=1}^{3}(d_{ik} - O_{ik})^2, \tag{3.45}$$

where (u_i, v_i) are the desired 2D pixel coordinates of the input point pattern and:

$$E_{orth} = \sum_{l=1}^{3}(V_{1l}^2 + V_{2l}^2 + V_{3l}^2 + a)^2 + \left(\sum_{c=1}^{3} V_{c1}V_{c2}\right)^2 + \left(\sum_{c=1}^{3} V_{c1}V_{c3}\right)^2 + \left(\sum_{c=1}^{3} V_{c2}V_{c3}\right)^2, \tag{3.46}$$

where a is a constant that accounts for any scaling in V weights (initially should be 1).

Major advantages of this method are its simplicity and the relaxation of the requirement of a good initial starting point, which is common to other non-linear optimization techniques (e.g., Levenberg-Marquardt's algorithm). Another advantage is that the proposed network can be used for single- or multi-image calibration [Ahmed, 1999a] and can be easily modified for zoom-lenses camera calibration [Ahmed, 2000].

Major drawbacks are the use of a 3D calibration pattern and the fact that lens distortion parameters are entirely discarded.

3.4 Self-calibration methods

In Computer Vision, the term self-calibration or auto-calibration is used when no pattern is employed in the calibration procedure. Thus, camera's parameters are recovered from a set of images, using only constraints on the referred parameters or on the scene acquired.

Since usual calibration methods do not allow an on-line camera calibration and assume that internal parameters of the camera are constant during image acquisition, self-calibration methods appeared as more versatile and efficient than the previous ones [Hartley, 2004].

In general, two types of constraints are applied: relative motion between scene and camera and/or constraints on the camera's intrinsic parameters.

- **Relative motion**
  - **General** (e.g. [Triggs, 1997]): these methods allow for any kind of camera or scene motions; thus, they are usually used for cameras placed in mobile platforms, like robots. They can be problematic with certain critical motion sequences [Sturm, 1997a] in which the motion of the camera is not generally sufficient to allow for recovering the calibration parameters and an ambiguity remains in the 3D reconstruction.
  - **Pure rotation** (e.g. [Hartley, 1994], [Stein, 1995], [Agapito, 1998], [Fitzgibbon, 1998], [Liu, 2000], [Hassanpour, 2004]): generally used in video-vigilance cameras, these methods assume that the relative motion between scene and used camera is described by rotations about a single fixed axis.
  - **Pure translation** (e.g. [Jang, 1996], [Ruf, 1998]): assuming camera's displacement along a straight line, equations for calibration can be greatly simplified.
  - **Planar motion** (e.g. [Li, 2004], [Espuny, 2007]): images are acquired by a camera which can rotate about a fixed axis and can move in any direction orthogonal to that axis; used, for example, in cameras mounted on vehicles moving on planar surfaces.
- **Intrinsic parameters**
  - **Fixed** (e.g. [Faugeras, 1992b], [Hartley, 1993], [Pollefeys, 1996], [Heyden, 1996], [Triggs, 1997], [Fitzgibbon, 1998], [Espuny, 2007]): the majority of self-calibration methods described in the Computer Vision literature treat intrinsic camera's parameters as constant but unknown.
  - **Variable** (e.g. [Heyden, 1997], [Agapito, 1998], [Pollefeys, 1998], [Li, 2004]): usually, only the effect of zoom variation is addressed in these methods, since they assume that zoom variation changes the focal distance.

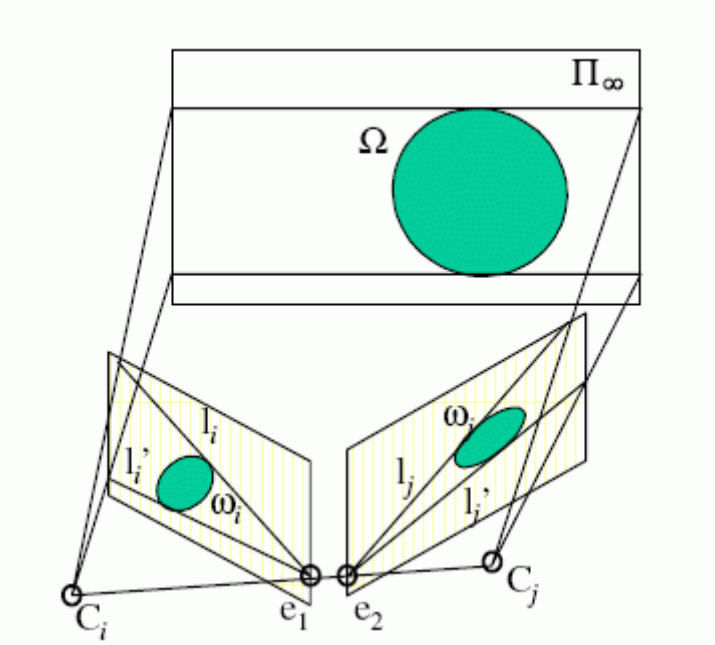

Fig. 3.8 – The absolute conic Ω, a particular conic in the plane at infinity Π_∞, and its image projections, ω_i and ω_j.(from [Hemayed, 2003]).

3.4.1 General motion

Fixed intrinsic parameters

These self-calibration methods allow for any kind of camera's motion, keeping the intrinsic parameters constant. They were the first to appear [Hartley, 2004] and are based on the replacement of the known calibration object by an abstract one, the absolute conic, Fig. 3.8.

The absolute conic, Ω, is a particular conic in the plane at infinity, which is invariant to transformations in the 3D space; that is, if two images are acquired at different camera positions and its internal parameters stay constant, the image of the absolute conic will be the same in both image planes.

The image of the absolute conic, ω_∞, is related to the camera's intrinsic matrix, K, by the following equation:

$$\omega_\infty = \mathrm{K}^{-T}\mathrm{K}^{-1}. \tag{3.47}$$

With ω_∞, the intrinsic matrix, K, can then be extracted using, for example, Cholesky decomposition [Golub, 1983].

The Kruppa equations relate the epipolar geometry (or the fundamental matrix) to the image of the absolute conic, ω_∞. Each point $\mathbf{p}$, belonging to the image of the absolute conic in the second image, satisfies:

$$\mathbf{p}^T \omega_\infty \mathbf{p} = \mathbf{0} \Leftrightarrow \mathbf{p}^T \mathrm{K}^{-T} \mathrm{K}^{1} \mathbf{p} = 0. \tag{3.48}$$

Fig. 3.8 shows two planes tangent to the absolute conic Ω passing through the two camera centres, C_i and C_j. Thus, those planes intersect the image planes at two pairs of epipolar lines, which are tangent to the image of the absolute conic. This fact is better expressed with the aid of duality: conics are self-dual objects, i.e., the envelope of lines tangent to a conic is also a conic, known as the dual conic, ω_∞^*, defined as:

$$\mathbf{l}^T \mathrm{K}\mathrm{K}^T \mathbf{l} = \mathbf{0} \Leftrightarrow \mathbf{l}^T \omega_\infty^* \mathbf{l} = 0, \tag{3.49}$$

where $\mathbf{l}$ is the line tangent to ω_∞. This implies that for a point $\mathbf{q}$ on any of the tangents to ω_∞ in the second image holds the constraint:

$$(\mathbf{e}_2 \times \mathbf{q})^T \mathrm{K}\mathrm{K}^T (\mathbf{e}_2 \times \mathbf{q})^T = 0, \tag{3.50}$$

where $\mathbf{e}_2$ is the epipolar point on the second image. The epipolar line $\mathrm{F}^T\mathbf{q}$, corresponding to $\mathbf{q}$ in the first image, is also tangent to ω_∞. Thus, the invariance of ω_∞ under rigid transformations yields:

$$(\mathrm{F}^T\mathbf{q})^T \mathrm{K}\mathrm{K}^T (\mathrm{F}^T\mathbf{q})^T = 0. \tag{3.51}$$

Combining the Equations (3.50) and (3.51), the following can be obtained:

$$\mathrm{F}\omega_\infty^* \mathrm{F}^T = \beta [\mathbf{e}_2]_\times^T \omega_\infty^* [\mathbf{e}_2]_\times = \beta [\mathbf{e}_2]_\times \omega_\infty^* [\mathbf{e}_2]_\times^T, \tag{3.52}$$

where β is an arbitrary nonzero scale factor and $[\mathbf{e}_2]_\times$ is the anti-symmetric matrix of vector $\mathbf{e}_2$. Citing [Lourakis, 1999], Equation (3.52) can be explained as: "*the Kruppa equations express the constraint that epipolar lines in the second image, that correspond to epipolar lines of the first image that are tangent to* ω_∞*, are also tangent to* ω_∞ *and vice versa*".

Since $\mathrm{F}\omega_\infty^*\mathrm{F}^T$ is a symmetric matrix, Equation (3.52) corresponds to the following equations, obtained by eliminating β:

$$\frac{\mathrm{F}\omega_\infty^*\mathrm{F}_{11}{}^T}{([\mathbf{e}_2]_\times\omega_\infty^*[\mathbf{e}_2]_\times^T)_{11}}=\frac{\mathrm{F}\omega_\infty^*\mathrm{F}_{12}{}^T}{([\mathbf{e}_2]_\times\omega_\infty^*[\mathbf{e}_2]_\times^T)_{12}}=\frac{\mathrm{F}\omega_\infty^*\mathrm{F}_{13}{}^T}{([\mathbf{e}_2]_\times\omega_\infty^*[\mathbf{e}_2]_\times^T)_{13}}$$
$$=\frac{\mathrm{F}\omega_\infty^*\mathrm{F}_{22}{}^T}{([\mathbf{e}_2]_\times\omega_\infty^*[\mathbf{e}_2]_\times^T)_{22}}=\frac{\mathrm{F}\omega_\infty^*\mathrm{F}_{23}{}^T}{([\mathbf{e}_2]_\times\omega_\infty^*[\mathbf{e}_2]_\times^T)_{23}}=\frac{\mathrm{F}\omega_\infty^*\mathrm{F}_{33}{}^T}{([\mathbf{e}_2]_\times\omega_\infty^*[\mathbf{e}_2]_\times^T)_{33}}\quad, \tag{3.53}$$

where F_{ij} is the fundamental matrix between images i and j. There are only two independent equations among the set of six in Equation (3.53), which are the second order polynomials of ω_∞^*. When using the Kruppa equations, given in Equation (3.53), for self-calibration, it is common to start by estimating ω_∞^* and then the intrinsic matrix K can be extracted using Cholesky decomposition [Golub, 1983], as referred before.

Computing the camera's extrinsic parameters has two classical solutions, both based on the fundamental matrix that was calculated during the estimation of the intrinsic parameters. A direct factorization approach uses the essential matrix, E, obtained from:

$$\mathrm{E}=\mathrm{K}^T\mathrm{F}\mathrm{K}\,. \tag{3.54}$$

Then, finding the rotation matrix R and translation vector $\mathbf{t}$ has well known solutions, which are explained in [Hartley, 1992].

An alternative approach is to directly use the error function that has been employed in the determination of the fundamental matrix:

$$\sum_i(\mathbf{m}_i'\mathrm{K}^{-T}\mathrm{E}\mathrm{K}\mathbf{m}_i)^2\,, \tag{3.55}$$

where $(\mathbf{m}_i,\mathbf{m}_i')$ are the matching points between a pair of images and $\mathrm{E}=[\mathbf{t}]_\times\mathrm{R}$.

First algorithms that use the Kruppa equations can be found in [Maybank, 1992] and [Faugeras, 1992a]. In these works, point correspondences from at least three different viewpoints are required.

In [Luong, 1997], a global method is used, which refines the camera's parameters using a non-linear minimization, highly reducing the calibration error.

In [Zeller, 1996], an image video sequence was used and therefore, it is considered a generalization to a large number of images of the method developed in [Faugeras, 1992a]. Robustness was improved with statistical tools, such as robust model fitting and covariance matrices.

The first method to use the SVD of the fundamental matrix [Golub, 1983], which leads to a particularly simple form of the Kruppa equations, was presented in [Lourakis, 1999]. Apart from the fundamental matrix itself, no other quantities that can be extracted from it (e.g. the epipoles) are needed. Since an accurate estimation of the epipoles is difficult in the presence of noise and/or degenerated camera motions, this method proved to be simpler and more efficient. The camera's parameters were recovered by a non-linear optimization, using a classical Levenberg-Marquardt algorithm [Moré, 1977].

Some other methods use a stratified approach: they use the projection matrices, obtained from the fundamental matrix, to determine a projective reconstruction and then transform it to an Euclidean reconstruction (in some cases an affine reconstruction is obtained in between).

In [Hartley, 1993], the presented method can be applied to a sequence of images, from which camera calibration is determined from matching points. Then, the scene is reconstructed, relative to the placement of one of the images used as reference, up to an unknown scaling.

Modulus constraint was used in [Pollefeys, 1996] to obtain an affine reconstruction from at least four different viewpoints. Compared with [Hartley, 1993], it has the advantage that the non-linear optimization only takes place in a three dimensional parameter space, which means that it always converges to the optimal solution.

Later, in [Pollefeys, 1997], the infinite homography property is used as a restriction to get the plane at infinity for each image pair. It is more robust because it does not consider only one image as reference and also reduces the minimal necessary images to three.

In [Heyden, 1996], the calibration process is simplified because it proceeds directly towards an Euclidean reconstruction and, again, only three images are needed.

Another method was proposed in [Triggs, 1997], that also achieves an Euclidean structure right after the projective one, but uses the absolute quadric, equivalent to the traditional absolute conic but simpler to use.

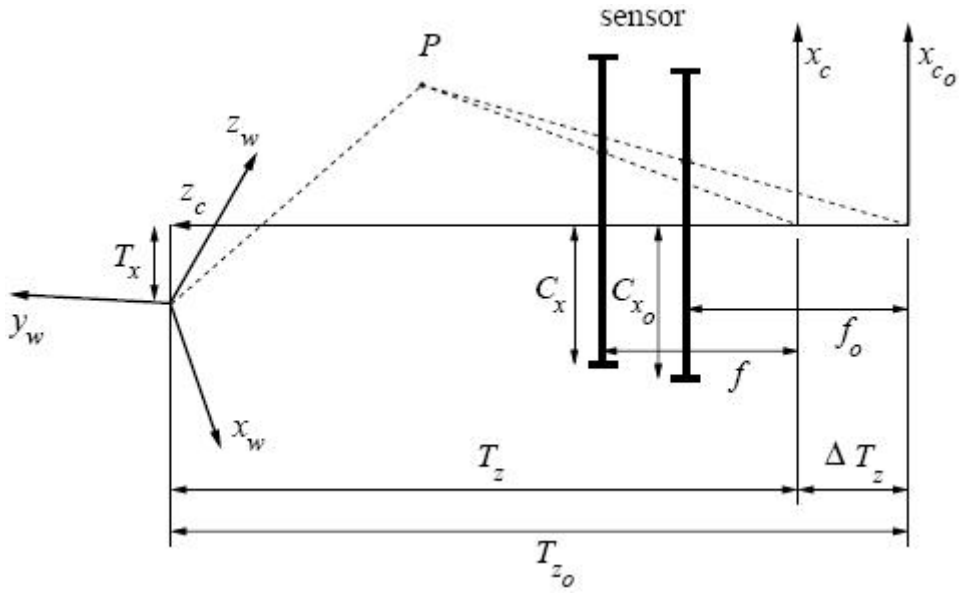

Fig. 3.9 – Change of the extrinsic and intrinsic camera's parameters with lens settings from camera plane $x_C z_C$ to $x_{C_0} z_{C_0}$ (from [Willson, 1994]).

Variable intrinsic parameters

These self-calibration methods allow for any kind of camera motion and also model the variation of the intrinsic parameters, during the image sequence acquisition. This generally happens when the camera has the auto-zoom and/or auto-focus capacities enabled. Therefore, the image-formation process is an adjustable function of the lens control settings, and thus the terms in the camera models must also be variable, Fig. 3.9.

In [Willson, 1994], the influence of zoom and focus in the camera's model parameters was studied. It was concluded that both focal distance and principal point are affected by zoom and focus variation. Thus, the following methodology was proposed: first, a conventional fixed camera model is calibrated at a number of settings spanning the desired range of lens settings; then, variation of the fixed model parameters is characterized, by alternately setting polynomials to individual parameters. The resulting adjustable camera model can interpolate between the original sampled lens settings to produce, for any lens setting, a set of values for the parameters in the fixed camera model. In [Heyden, 1997; Heyden, 1998], an iterative method was presented, based on bundle adjustment techniques. It only requires the camera motion to be sufficiently general; thus, pure translational or rotational movements lead to failure in the proposed method. The work presented in [Lourakis, 2000] is an adaptation of the method introduced in [Lourakis, 1999], where the SVD of the fundamental matrix [Golub, 1983] is used to simplify the Kruppa equations, in

the case of variable intrinsic camera parameters. These are recovered through a non-linear minimization scheme, assuming perfect orthogonality and square pixels. In [Pollefeys, 1996b], a stratified approach was used, which went from a projective through affine to metric reconstruction. Like in previous works, the modulus constraint was used. For practical purposes and in order to get a good initialization, the first two images must be acquired using a pure translation motion. Later, in [Pollefeys, 1999], an iterative method was developed that explores the absolute quadric. Depending on camera intrinsic parameter constraints set, it gives the minimum number of images necessary to perform self-calibration, assuming only the absence of skew. This method also detects critical motion sequences [Sturm, 1997a]. In [Hartley, 1992], a non-iterative method was developed. Based on the properties of the essential matrix, it determines the focal length and external camera parameters of two cameras, assuming all other internal camera parameters are known. An essential matrix is obtained from the fundamental matrix by a transformation involving the intrinsic parameters of the pair of cameras associated with the two input images. Thus, constraints on the essential matrix can be translated into constraints on the intrinsic parameters of the pair of cameras. Later, in [Hartley, 1999], the previous method to self-calibrate a camera was adapted to a sequence of images. Equations are simplified, because square pixels and the principal point were considered as the image centre. In [Mendonça, 1999], the method developed in [Hartley, 1992] was extended to the case of multiple varying intrinsic parameters and larger image sequences. In [Sturm, 1997b], it was considered that the principal point position depends on the camera's zoom. A pre-calibration of the camera was proposed, with the intent to model the inter-dependence of the intrinsic parameters, concretely between the principal point and pixel scale factors. Thus, self-calibration came down to the estimation of only one intrinsic parameter. This method exploited the latter fact and did not need an initialization of the intrinsic parameters.

3.4.2 Pure rotation

These self-calibration methods are applied to cameras that undergo pure rotations, like in many surveillance systems and in the broadcast of sport events. Having null translation allows eliminating the three parameters of the translation vector $\mathbf{t}$ and, thus, the calibration is simplified, Fig. 3.10.

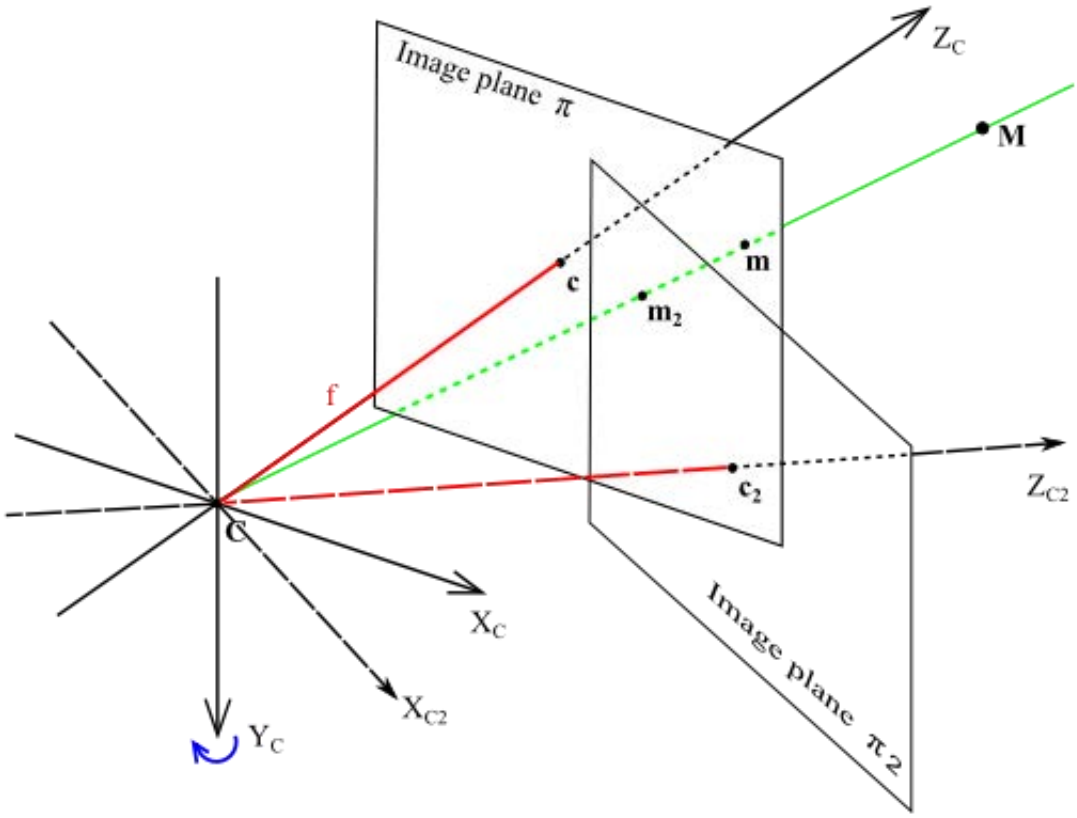

Fig. 3.10 – Pure camera rotation in turn of the Y-axis.

Sometimes a problem arises when translation is not assured to be null; due to camera's mechanical issues, the rotation axis does not contain the optical centre or, when zoom is applied, a displacement of the optical centre occurs [Hayman, 2002].

Fixed intrinsic parameters

Some self-calibration methods of this group assume that the rotation angle, between consecutive images, is known. Some examples are the methods proposed in [Du, 1993], [Stein, 1993; Stein, 1995], [Viéville, 1994] and [McLauchlan, 1996].

In [Stein, 1993], a self-calibration method was developed using pairs of images where the camera has rotated around a constant axis of rotation. Given an initial estimate of the camera's intrinsic parameters, the angle and axis of rotation, results are refined by minimizing the sum of squared distances between the feature points in the second image and those computed from the first image.

Computation of the points assumes a pure rotation around an axis $\vec{W} = (w_x, w_y, w_z)$, through an angle of $-\theta$ degrees. Given a world point $\mathbf{P} = (X, Y, Z)$ expressed in the camera coordinate system, its coordinates after one rotation step are given by:

$$\mathbf{P}' = \mathrm{R}\mathbf{P} = [\cos\theta \mathrm{I} + \sin\theta \mathrm{Q} + (1 - \cos\theta) WW^T]\mathbf{P}, \tag{3.56}$$

where I is the identity matrix and matrix Q is defined as:

$$\mathrm{Q} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}. \tag{3.57}$$

Similarly to Equation (3.13), point $\mathbf{P}' = (X', Y', Z')$ is projected as point $\mathbf{p}' = (u', v')$ in the image plane:

$$\begin{aligned} u' &= f\frac{X'}{Z'} = f\frac{r_{11}x + r_{12}y + r_{13}f}{r_{31}x + r_{32}y + r_{33}f} \\ v' &= f\frac{Y'}{Z'} = f\frac{r_{21}x + r_{22}y + r_{23}f}{r_{31}x + r_{32}y + r_{33}f} \end{aligned}. \tag{3.58}$$

Without loss of generality, the rotation can be considered around the Y-axis; thus, $\vec{W} = (0, 1, 0)$ and the rotation matrix becomes:

$$\mathrm{R} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{3.59}$$

Combining Equations (3.58) and (3.59), yields:

$$\begin{aligned} u' &= f\frac{x\cos\theta + f\sin\theta}{-x\sin\theta + f\cos\theta} \\ v' &= f\frac{y}{-x\sin\theta + f\cos\theta} \end{aligned}. \tag{3.60}$$

A major disadvantage is that this method requires knowing the radial distortion parameters, and errors in these parameters can cause errors in the estimates of the focal length and of principal point.

The other group of methods does not require the knowledge of the rotation angle. The first proposed was the one described in [Hartley, 1994]. This method requires at least three images, but two are sufficient if some assumptions about the intrinsic parameters can be

made, like considering the skew $s_x = 0$, or that the camera has square pixels, i.e., $k_u = k_v$. For convenience, the coordinate axes are chosen to be aligned with the 0-th image projection centre, so that $\mathrm{R}_0 = \mathrm{I}$. Thus, for all other images, the projection matrix P_j can be written in the following form:

$$\mathrm{P}_j = \mathrm{K}\mathrm{R}_j\mathrm{K}^{-1}. \tag{3.61}$$

Since the rotation matrix obeys to $\mathrm{R} = \mathrm{R}^{-T}$, the previous equation can be re-arranged into:

$$(\mathrm{K}\mathrm{K}^T)\mathrm{P}_j^{-T} = \mathrm{P}_j(\mathrm{K}\mathrm{K}^T). \tag{3.62}$$

With sufficiently views and theirs corresponding matrices P_j, Equation (3.61) can be used to solve for the entries of matrix $\mathrm{K}\mathrm{K}^T$. It is a non-iterative algorithm that uses the Cholesky decomposition [Golub, 1983] to extract the intrinsic parameters from $\mathrm{K}\mathrm{K}^T$. This method was further improved in [Hartley, 1997].

Variable intrinsic parameters

The works described in [Agapito, 1998] and [Seo, 1998; Seo, 1999] can be included in this group. Iterative methods have been developed, based on the infinite homography constraint, obtained from image matches. This constraint relates the unknown calibration matrices to the computed inter-image homographies. For the optimization process, the first work uses the Levenberg-Marquardt algorithm [Moré, 1977], and the second uses the Kanatani's method [Kanatani, 2000]. Both considered that the principal point and aspect ratio were fixed and that the image axes were orthogonal.

Later, the same authors have developed linear versions of their methods, [Agapito, 1999] and [Seo, 2001]. In [Agapito, 1999], zero skew, square pixels, known pixel aspect ratio and known principal point were assumed. It was based on the absolute conic and required at least two images.

The method developed in [Seo, 2001] also needed a minimum of two images and the knowledge of the principal point. Later, a complete method was developed in [Agapito, 2001], decomposed into three steps: a first initial approximation given by the linear method [Agapito, 1999]; the iterative method [Agapito, 1998]; and a final parameter refinement, by

minimizing geometric errors, using MLE (Maximum Likelihood Estimation) and MAP (Maximum a Posteriori Estimation).

In [Agapito, 2001], the negative effect of radial distortion in the self-calibration process was also studied and some possible solutions to overcome it were pointed out.

3.4.3 Pure translation

Pure translational motion means that there are no rotations involved and allows for linear calibration methods.

Linear constraints are derived by means of planar homographies between images. Assuming that $\mathbf{m} = (u, v, 1)^T$ and $\mathbf{m}' = (u', v', 1)^T$ are the homogeneous coordinates of two corresponding points in two images, they are related by:

$$s\mathbf{m}' = \mathrm{H}\mathbf{m}, \tag{3.63}$$

where s is an unknown scale factor, and H is 3×3 homography matrix induced by the epipolar plane, Π [Hartley, 2004], Fig. 3.11. It can be shown that the homography between the two images can be expressed as:

$$\mathrm{H} = \sigma(\mathrm{KRK}^{-1} + \mathrm{K}\frac{\mathbf{t}\vec{n}^T}{\mathbf{d}}\mathrm{K}^{-1}), \tag{3.64}$$

with σ an unknown scale factor, $\vec{n}^T$ the unit vector perpendicular to plane Π and $\mathbf{d}$ the distance from the origin of the world coordinates system to Π. If the camera undergoes a pure translation motion, then the homography becomes:

$$\mathrm{H} = \sigma(\mathrm{I} + \mathrm{K}\frac{\mathbf{t}\vec{n}^T}{\mathbf{d}}\mathrm{K}^{-1}). \tag{3.65}$$

Considering two camera planar orthonormal translations, $\mathbf{t}_1$ and $\mathbf{t}_2$, the relation between the homographies for both translations are derived from previous equation, leading to the constraint:

$$(\mathrm{H}_1^T - \sigma_1\mathrm{I})\mathrm{K}^{-T}\mathrm{K}^{-1}(\mathrm{H}_2 - \sigma_2\mathrm{I}) = \frac{\sigma_1\sigma_2}{\mathbf{d}^2}\vec{n}_1\mathbf{t}_1^T\mathbf{t}_2\vec{n}_2 = \mathbf{0}. \tag{3.66}$$

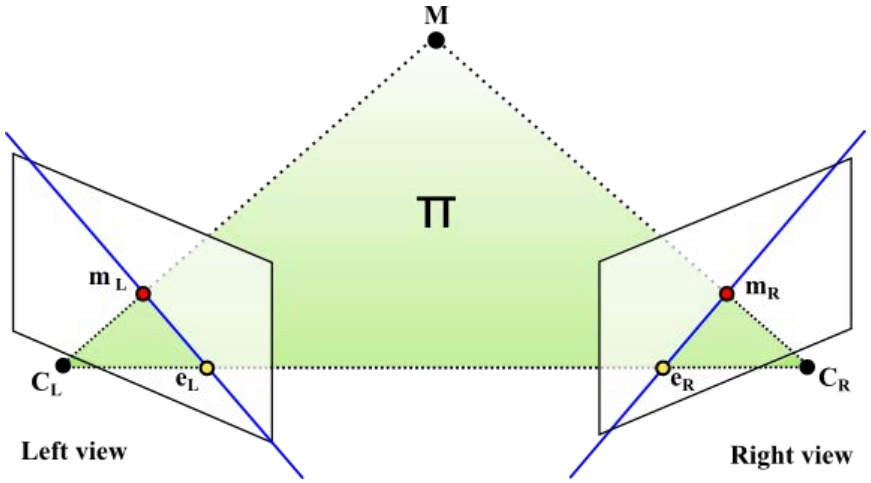

Fig. 3.11 – Epipolar geometry: the epipolar plane Π is formed by 3D point **M** and both cameras' centres **C**. The plane intersects both cameras' image planes, where it forms the epipolar lines (in blue). Points $\mathbf{e}_L$ and $\mathbf{e}_R$ are the epipoles.

In order to uniquely obtain $K^{-T}K^{-1}$ up to a scale factor, at least 5 sets of two planar orthogonal translations or 2 sets of three mutually orthogonal motions are acquired.

In this group, we can include the work described in [Ma, 1996], which was shown that if the camera's intrinsic parameters are known a priori, the camera's pose can be solved using three translational motions. If the intrinsic parameters are unknown, then two sequences, each consisting of three mutual orthogonal translations, are necessary to determine the camera's intrinsic and extrinsic parameters. The method developed in [Yang, 1998] requires four sequences of two orthogonal planar translations. Both methods assumed orthogonal pixels. This restriction was avoided in [Li, 2002]; however, it was necessary to use the scene planar information and to acquire at least five sequences of two orthogonal motions.

3.4.4 Planar motion

Another restriction on the camera's motion is to assume planar motion, i.e. the camera can move in any direction contained in a plane and can rotate about an axis perpendicular to that plane. An example of this kind of motion is when the camera is mounted on a vehicle moving on a planar surface.

Some works on camera self-calibration based on planar motion were described, for example, in [Beardsley, 1995] and [Armstrong, 1996b].

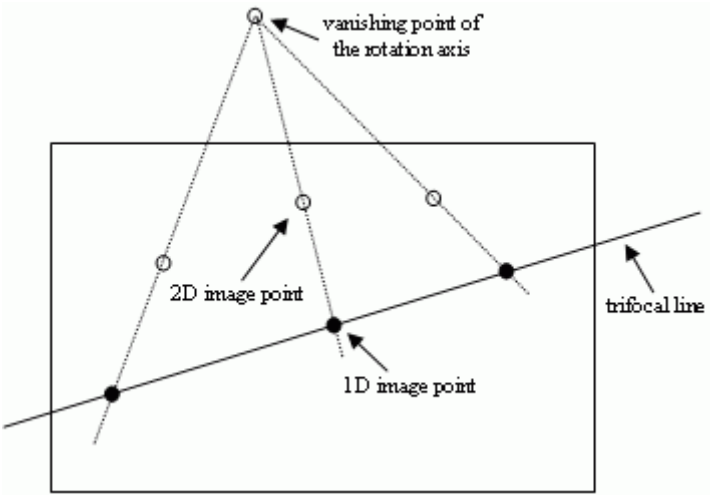

Fig. 3.12 – Converting 2D image points into 1D image points in the image plane is equivalent to a projective projection from the image plane to the trifocal line, with the vanishing point of the rotation axis as the projection centre (from [Faugeras, 2000]).

More recently, in [Faugeras, 2000], the non-linear problem of self-calibration of a camera undergoing planar motions is simplified to a linear one. This approach is based on the concept that, given planar motion, the camera trifocal plane, i.e. the plane through the projection centres, is coincident with the motion plane where the camera is moving on. Therefore, the image location of the motion plane is the same as the trifocal line which could be determined from fundamental matrices [Armstrong, 1996b], Fig. 3.12. Thus, the 2D images of a camera undergoing any planar motion reduce to 1D images by projecting the 2D image points onto the trifocal line. A 1D camera is represented by a 2×3 projection matrix $\tilde{\mathrm{P}}$, which can be decomposed into:

$$\tilde{\mathrm{P}} = \tilde{\mathrm{K}}[\tilde{\mathrm{R}} \mid \tilde{\mathbf{t}}], \tag{3.67}$$

where $\tilde{\mathbf{t}}$ is a 2×1 translation vector, $\tilde{\mathrm{R}}$ is a 2×2 rotation matrix:

$$\tilde{\mathrm{R}} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \tag{3.68}$$

and $\tilde{\mathrm{K}}$ represents the two camera intrinsic parameters:

$$\tilde{\mathrm{K}} = \begin{bmatrix} f & u_0 \\ 0 & 1 \end{bmatrix}. \tag{3.69}$$

For three views from a camera undergoing planar motion, with fixed intrinsic parameters, there are four fixed points, three of which are collinear, Fig. 3.13:

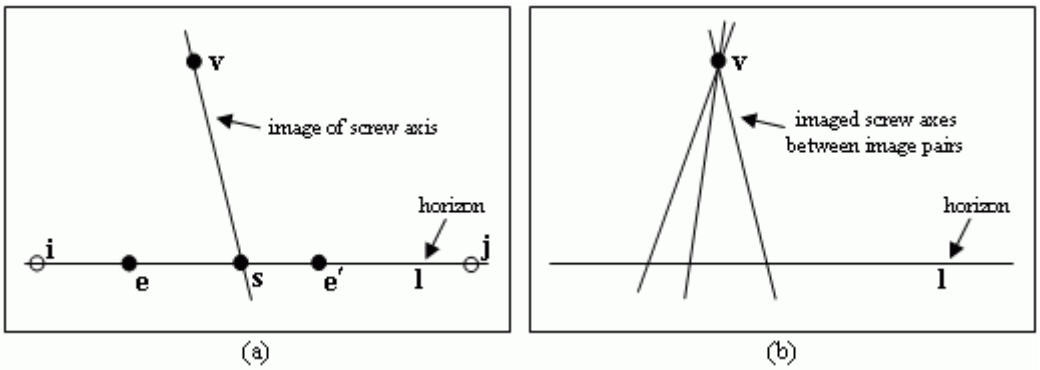

Fig. 3.13 – Fixed image entities for planar motion: (a) For two views, the imaged screw axis is a line of fixed points in the image and the horizon is a fixed line under the motion, where the epipoles lie. (b) Relation between the fixed lines obtained pairwise for three images under planar motion; the image horizon lines for each pair are coincident, and the imaged screw axes for each pair intersect in the vanishing point of rotation; all the epipoles lie on the horizon line (from [Armstrong, 1996a]).

1. the vanishing point of the rotation (or screw) axes, $\mathbf{v}$;
2. two complex points, the images of the two circular points, $\mathbf{i}$ and $\mathbf{j}$, on the horizon line;
3. a third point on the horizon line and peculiar to the image triplet, $\mathbf{x}$.

The relationship between the image of the circular points, $\mathbf{i}$, and the internal parameters of the 1D camera follows directly by projecting one of the circular points $I = (i, 1, 0)^T$ by the 2×3 camera's projection matrix $\tilde{\mathrm{P}}$:

$$\lambda \mathrm{i} = e^{-i\theta} \begin{bmatrix} u_0 + if \\ 1 \end{bmatrix}, \tag{3.70}$$

with $i = \sqrt{-1}$. It clearly appears that the real part of the projective coordinates' ratio of the image of the circular point $\mathbf{i}$ is the position of the principal point u_0 and the imaginary part is the focal length f.

This approach avoids the unstable convergence of Kruppa equations. However, it also brings some new problems to be solved, like the detection of planar motions and robust estimation of the 1D trifocal tensors under noise, which is magnified by the projection from 2D image points to 1D image points.

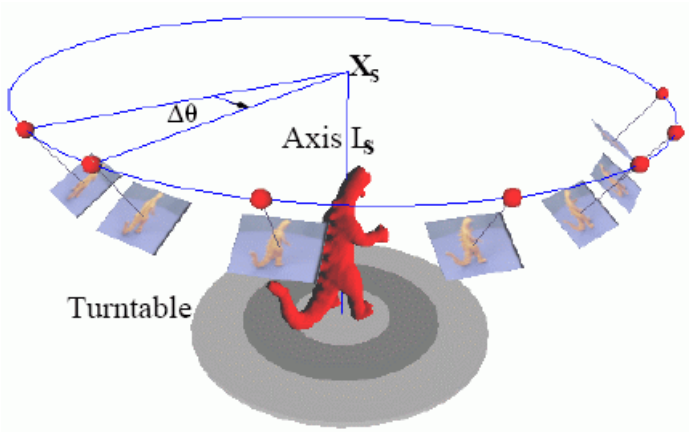

Fig. 3.14 – Turntable motion: the red spheres are the camera projection centres and the point X_S is the intersection of the horizontal plane containing the projection centres, with the rotation axis L_S (from [Fitzgibbon, 1998]).

a) Single axis or turntable motion

One special case inside the ordinary planar motion is the single axis motion or circular motion. This particular motion can be described as set of rotations about a single, fixed screw axis, with zero translation along the same screw axis.

The most common case where this motion occurs when a static camera is viewing an object placed on a turntable device, Fig. 3.14. Some works that use this special motion for camera self-calibration can be seen, for example, in [Fitzgibbon, 1998], [Jiang, 2004], [Fremont, 2004] and [Zhang, 2006].

The image invariants under this kind of motion are derived from the ones explained before and presented in Fig. 3.13. The horizontal plane, Π_h, containing the projection centres, is orthonormal to the rotation axis, L_S, Fig. 3.14. The image of Π_h is the horizon or vanishing line, $\mathbf{l}_h$. The image of the rotation axis, L_S, is the line $\mathbf{l}_s$. Let the plane defined by the image camera centre and the rotation axis be Π_s.

Consider three orthogonal directions N_x, N_y and N_z, given by the normal direction of Π_s, the Y-axis, and $N_x \times N_y$, respectively. These three directions will have vanishing points $\mathbf{v}_x$, $\mathbf{v}_y$ and $\mathbf{v}_z$, respectively. Since $\mathbf{l}_s$ is also the image of Π_s, $\mathbf{v}_x$ and $\mathbf{l}_s$ form a pole-polar relationship with respect to the image of the absolute conic [Faugeras, 1992a]. By construction, N_x is parallel to Π_h, and N_z is parallel to both Π_h and Π_s. Hence, $\mathbf{v}_x$ must lie on $\mathbf{l}_h$ and $\mathbf{v}_z$ is given by the intersection of $\mathbf{l}_h$ and $\mathbf{l}_s$.

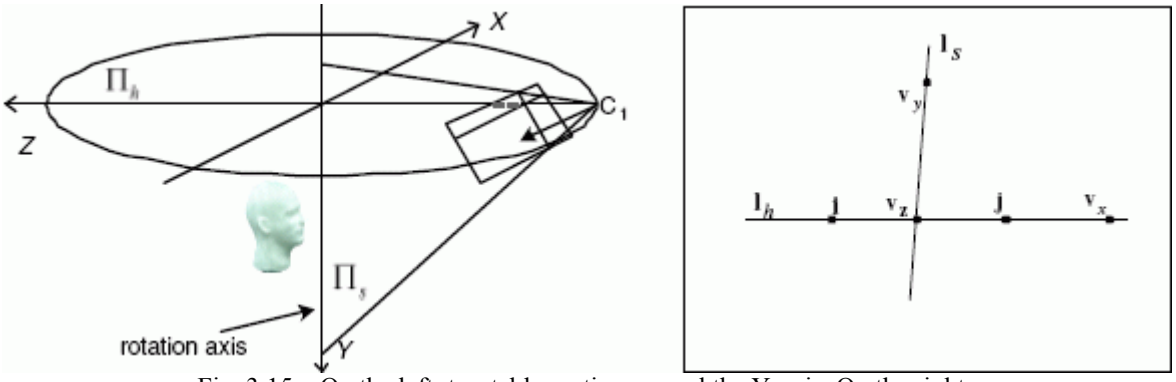

Fig. 3.15 – On the left: turntable motion around the Y-axis. On the right: 2D image invariants under turntable motion (from [Zhang, 2005]).

The pair of imaged circular points $\mathbf{i}$ and $\mathbf{j}$ must also lie on $\mathbf{l}_h$, hence $\mathbf{l}_h = \mathbf{i} \times \mathbf{j}$. If the camera intrinsic parameters are kept constant, then all the mentioned image entities will be fixed, Fig. 3.15. The fundamental matrix relating any two images acquired by turntable motion can be explicitly parameterized in terms of the image invariants [Fitzgibbon, 1998; Mendonça, 2001], and is given by:

$$\mathrm{F}(\theta) = [\mathbf{v}_x]_\times + k \tan\frac{\theta}{2}(\mathbf{l}_s\mathbf{l}_h^T + \mathbf{l}_h\mathbf{l}_s^T), \tag{3.71}$$

where θ is the angle between the two views and k an unknown scalar. The terms in Equation (3.71) are in homogeneous coordinates and therefore, the term k is necessary to account for the different scales used in the representations. The scalar k is unknown but fixed for any angle θ and cannot be obtained from two images alone.

In Equation (3.71), the only variable is $k\tan\theta/2$. This parameter can be found by using a triplet of images, with relative angles θ_{ij}. Let $\gamma_1 = k\tan\theta_{12}/2$, $\gamma_2 = k\tan\theta_{23}/2$ and $\gamma_3 = k\tan\theta_{13}/2$. Since $\theta_{13} = \theta_{12} + \theta_{23}$, the fixed scalar k can be obtained using:

$$k = \sqrt{\frac{\gamma_1\gamma_2\gamma_3}{\gamma_3 - \gamma_1 - \gamma_2}}. \tag{3.72}$$

Then, the rotation angles between each pair of images are determined by dividing γ by k.

3.5 Summary

Camera calibration is a process to determine the relationship between a given 3D coordinate system (world coordinates) and a 2D coordinate system on the image plane (pixel coordinates). More specifically, it is to determine the camera's model involved in the transformation from the 3D world to the 2D image acquired by the camera.

In this Chapter, the classical pinhole camera model was described. In reality, real image cameras are much more complicated devices, and if they are used in the 3D reconstruction process, a proper calibration procedure should be used in order to achieve accurate results.

Considerable research has been done in this field. As an indication of that, several camera calibration methods have been suggested. An explanation of the most known classic and auto calibration methods was given in this Chapter, pointing out some of their drawbacks and specificities.

The camera calibration literature does not pay much attention to the errors originating in the calibration process. Some of the errors are caused by the insufficient parameters of the cameras' models, and others are due to some external factors. The commonly used cameras' models compensate for only radial and tangential lens distortions. As such, the models do not address the effect of the light intensity, wavelength, focus, iris or electrical distortions. Also, some external factors, like outliers in the measured data, inaccurate calibration targets, and asymmetric feature projection, can cause bias in the parameters estimated. These sources of errors should therefore be identified through other means and actions for preventing their influence on the accuracy of the 3D reconstructions should be made.

4

Developed methodologies for 3D reconstruction

4.1 Introduction

This Thesis refers to the Computer Vision field; namely, to the 3D Reconstruction. There have been considerable efforts made by the scientific community, mainly in the recent decades, in developing computational methodologies for obtaining the 3D shape of real objects, from acquired images of the same ones.

During the work developed in the scope of this Thesis, two methodologies that are commonly used in 3D reconstruction were addressed: Stereo Reconstruction and Volumetric Reconstruction. The central topic of this Thesis was to consider methodologies that allow obtaining the 3D models of objects, without imposing several restrictions on the relative motion between the camera used and the object to be reconstructed. They were also developed with the purpose of being of reduced cost, accurate, portable and easy-to-use.

Stereo reconstruction requires at least two perspective views of the object of interest. These views may be acquired simultaneously as in a stereo rig, or acquired sequentially, for example, by a camera moving relatively to the object. On the other hand, starting with a set of

correctly calibrated images, the volumetric based method reconstructs the 3D shape of an object, as well as achieves the coloration of the reconstructed 3D geometrical model.

4.2 Stereo-based reconstruction

The depth of an object's point along the corresponding projection ray is not directly accessible in a single image. With at least two images, depth can be determined through triangulation [Hartley, 1995]. This is the reason why most animals have two eyes or why autonomous robots are equipped with a stereo image system or acquire images associated to two distinct views when they have just one camera.

In the present case, it was pretended not to impose any kind of restrictions to the involved movement. So, starting with two uncalibrated images from an object, the goal was to extract the relative movement between the camera viewpoints, and finally to obtain the 3D geometry of the object involved. The following sections describe in detail each step of the developed stereo reconstruction methodology, Fig. 4.1.

4.2.1 Feature point detection and matching

Image feature points (or strong points) are those who have a strong 2D component over the imaged object.

Feature points reflect relevant discrepancies between their intensity values and those of their neighbours. Usually, they represent visible vertices of the correspondent object.

For this first step, it was employed an intensity-based algorithm that computes a measure to indicate the presence of an interest point directly from the greyscale image values. On a greyscale image, the gradient covariance matrix $\mathbf{A}$ averages derivatives of the image I in a window W around a pixel point (u,v):

$$\mathbf{A} = \begin{bmatrix} \sum_{W} \partial_u^2 \mathrm{I} & \sum_{W} (\partial_u \mathrm{I})(\partial_v \mathrm{I}) \\ \sum_{W} (\partial_u \mathrm{I})(\partial_v \mathrm{I}) & \sum_{W} \partial_v^2 \mathrm{I} \end{bmatrix}, \tag{4.1}$$

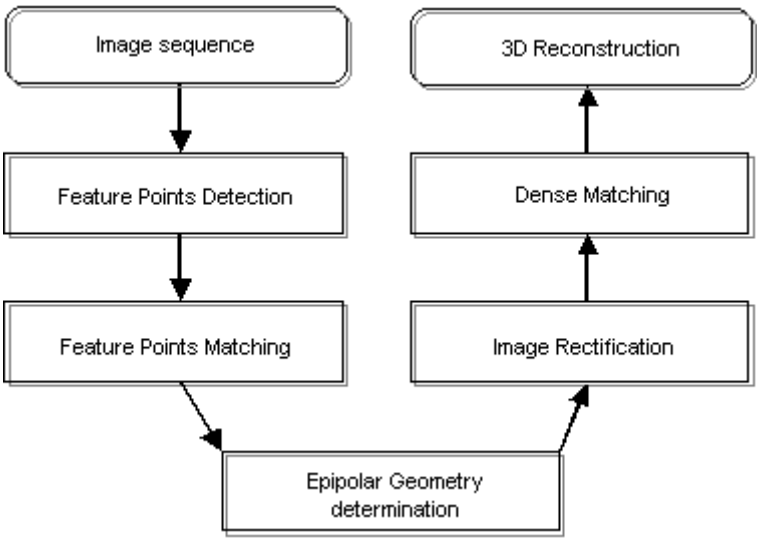

Fig. 4.1 – Methodology for 3D reconstruction based on stereo vision.

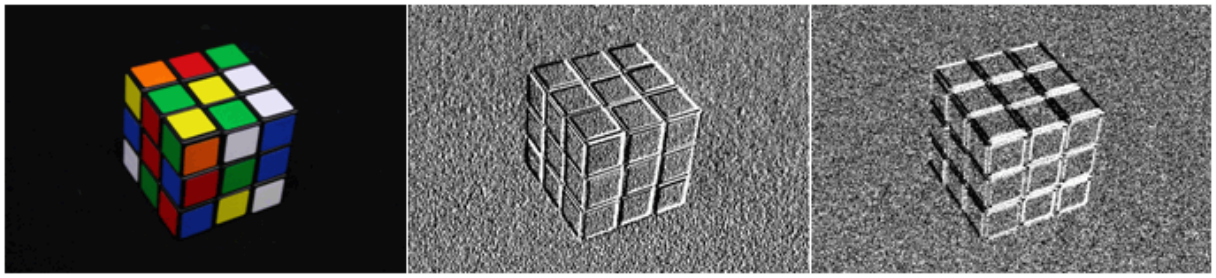

Fig. 4.2 – Original colour image (on the left), the correspondent horizontal derivative ∂_u (on the middle) and vertical derivative ∂_v (on the right).

where ∂_u and ∂_v are the right and left derivatives, Fig. 4.2. Here, the image derivatives are calculated using the Sobel operator, with an aperture size of 3, which combines Gaussian smoothing and differentiation. Therefore, the result is more robust to the presence of noise.

If covariance matrix $\mathbf{A}$ is of rank two, meaning that both of its eigenvalues, λ_1 and λ_2, are large, an interest point is detected. A matrix of rank one indicates an edge and rank zero a homogeneous region.

In [Harris, 1988], a weight function was used to determine the feature points:

$$R = \det(\mathbf{A}) - k(\mathrm{tr}(\mathbf{A}))\,, \tag{4.2}$$

where $\det(\mathbf{A}) = \lambda_1\lambda_2 = (\partial_u)^2(\partial_v)^2 - (\partial_u\partial_v)^2$ and $\mathrm{tr}(\mathbf{A}) = \lambda_1 + \lambda_2 = (\partial_u)^2 + (\partial_v)^2$. Local maxima of R are the feature point locations in the original image, Fig. 4.3. It was also ensured that all feature points are distanced enough one from another by considering some Euclidean distance, so features too close to other stronger features were removed.

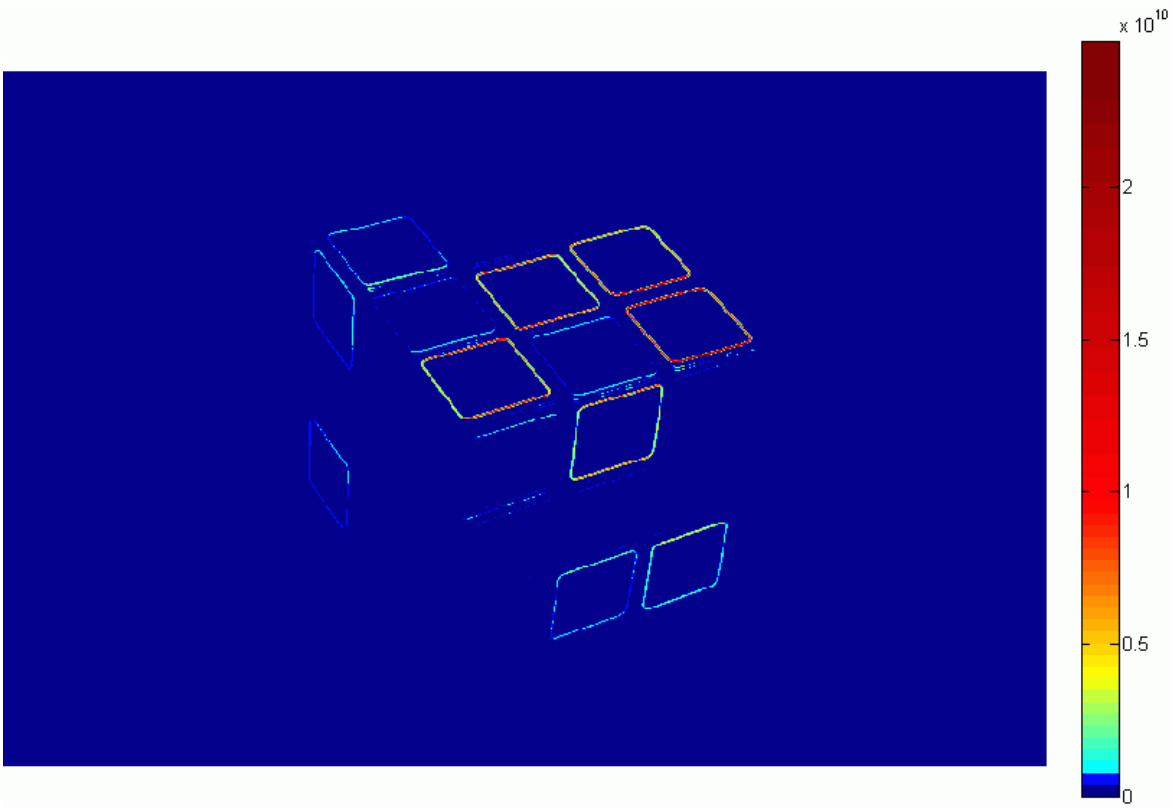

Fig. 4.3 – Colour representation of the Harris's weight function:
warmer colours for higher values (Equation (4.2)).

The matching process associates 2D points among sequential images, which are the projections of the same 3D object's point. Automatic detection of matching points between two images can be achieved using one cross-correlation method. These methods use small image windows from a first image as templates for matching in the subsequent images [Gonzalez, 1987].

In this Thesis, the Lucas-Kanade (LK) algorithm [Bouguet, 1999], which is one of the most popular methods for tracking features in Computer Vision, was used to match the object's image points.

Consider point $\mathbf{m} = (u, v)$ in image I. The goal of the LK algorithm is to find its matching point $\mathbf{m}' = \mathbf{m} + \mathbf{d} = (u + d_u, v + d_v)$ in image J, with vector $\mathbf{d}$ known as optical flow at $\mathbf{m}$, Fig. 4.4. This vector $\mathbf{d}$ is the one that minimizes the following residual function:

$$\varepsilon(\mathbf{d}) = \sum_{w_u} \sum_{w_v} (\mathrm{I}(u, v) - \mathrm{J}(u + d_u, v + d_v))^2 , \tag{4.3}$$

with $\mathbf{w} = (w_u, w_v)$ the search window size. Perfect matching results means $\varepsilon(\mathbf{d}) = 0$.

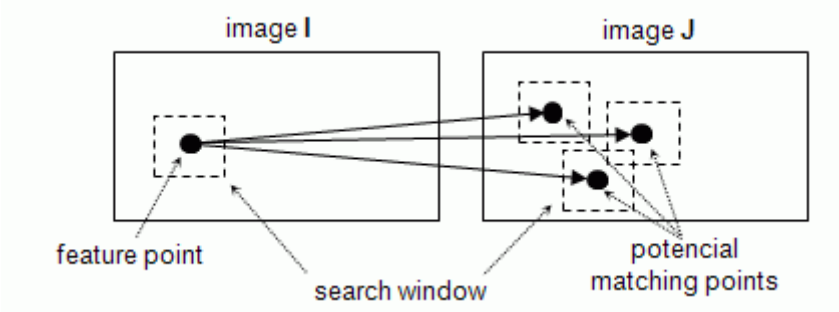

Fig. 4.4 – Matching points between two images by neighbourhood analysis of their feature points.

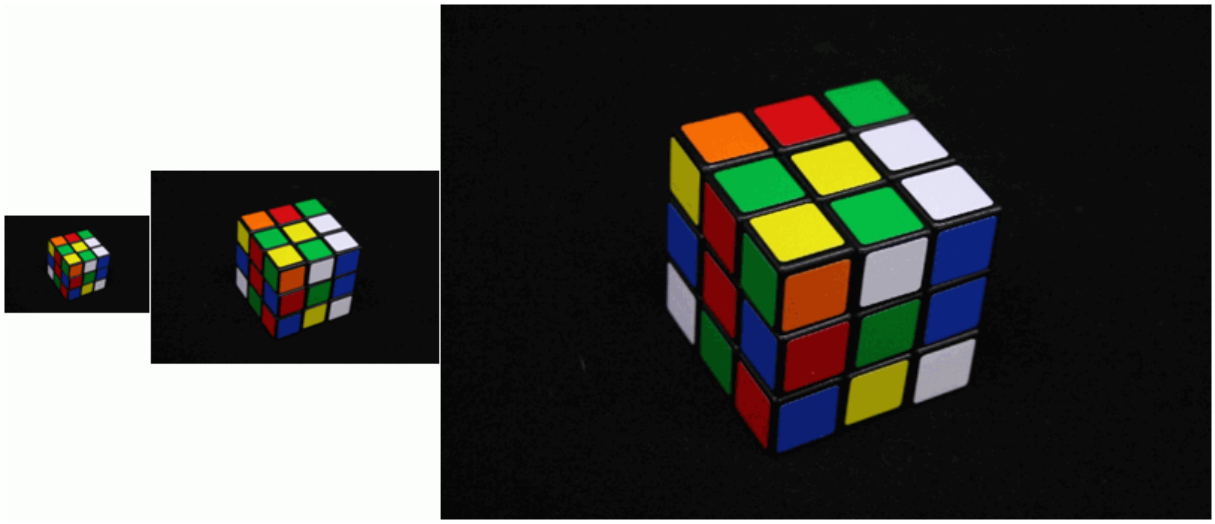

Fig. 4.5 – Coarse to fine 3-level pyramidal image resolution.

The LK algorithm uses a multi-resolution (pyramidal) approach, from coarse to fine image resolution, Fig. 4.5, to successfully handle relatively large displacements between the images. It assumes that, since vector **d** is small, Equation (4.3) can be minimized into the following system of two equations:

$$\mathbf{d} = \begin{bmatrix} \sum_w \partial_u \mathrm{I}(u_i, v_i)\mathrm{I}_\mathrm{t}(u_i, v_i) \\ \sum_w \partial_v \mathrm{I}(u_i, v_i)\mathrm{I}_\mathrm{t}(u_i, v_i) \end{bmatrix} \mathrm{A}^{-1}, \tag{4.4}$$

where A is the matrix described in Equation (4.1) and $\mathrm{I}_\mathrm{t}(u,v) = \mathrm{I}(u,v) - \mathrm{J}(u,v)$.

4.2.2 Epipolar geometry

Epipolar geometry determines a pairwise relative orientation and allows for rejection of false matches (or outliers). When the interior orientation parameters of both images are the

same, it mathematically expresses itself by the fundamental matrix, F, a projective singular correlation between two images [Faugeras, 1992a].

In epipolar geometry, $\mathrm{F}\mathbf{m}$ and $\mathrm{F}\mathbf{m}'$ describe the epipolar lines on which the corresponding points $\mathbf{m}'$ and $\mathbf{m}$, respectively, must lie on the other image, Fig. 4.6:

$$\begin{aligned} \mathbf{l} &= \mathrm{F}\mathbf{m}' \\ \mathbf{l}' &= \mathrm{F}\mathbf{m} \end{aligned}, \tag{4.5}$$

with $\mathbf{m} = (u, v, 1)^T$ and $\mathbf{m}' = (u', v', 1)^T$ the homogeneous image coordinates of corresponding points in a stereo image pair, respectively. This means that for all pairs of corresponding points, the epipolar geometry holds that:

$$\mathbf{m}'\mathrm{F}\mathbf{m} = 0\,. \tag{4.6}$$

Given at least eight matching points, Equation (4.6) can be used to obtain a set of linear equations of the form:

$$\mathrm{A}\mathbf{f} = 0\,, \tag{4.7}$$

with $\mathbf{f}$ the column vectorization of matrix F and each row of matrix A is vector $[uu' \;\; uv' \;\; u \;\; vu' \;\; vv' \;\; v \;\; u' \;\; v' \;\; 1]$. The fundamental matrix F is defined only up to an unknown scale. For this reason, and to avoid the trivial solution $\mathbf{f} = \mathbf{0}$, the additional constraint $F_{33} = 0$ is adopted.

The SVD algorithm solves the equation system (4.7). If rank of matrix A is smaller than 8, then there are multiple solutions for fundamental matrix F, i.e., the corresponding points lie on a plane.

It is trivial to reconstruct the fundamental matrix F from the solution vector $\mathbf{f}$. However, in the presence of noise, this matrix will not satisfy the rank-2 constraint. This means that there will not be real epipoles through which all epipolar lines pass. A solution to this problem is to obtain F as the closest rank-2 approximation of the solution coming out of the linear equations. With this new constraint, only seven corresponding points are necessary.

The described 8- and 7-point algorithms to compute the fundamental matrix assume that the matching points are accurate, i.e., the algorithms only accounting for some expected noise and they cannot cope with outliers.

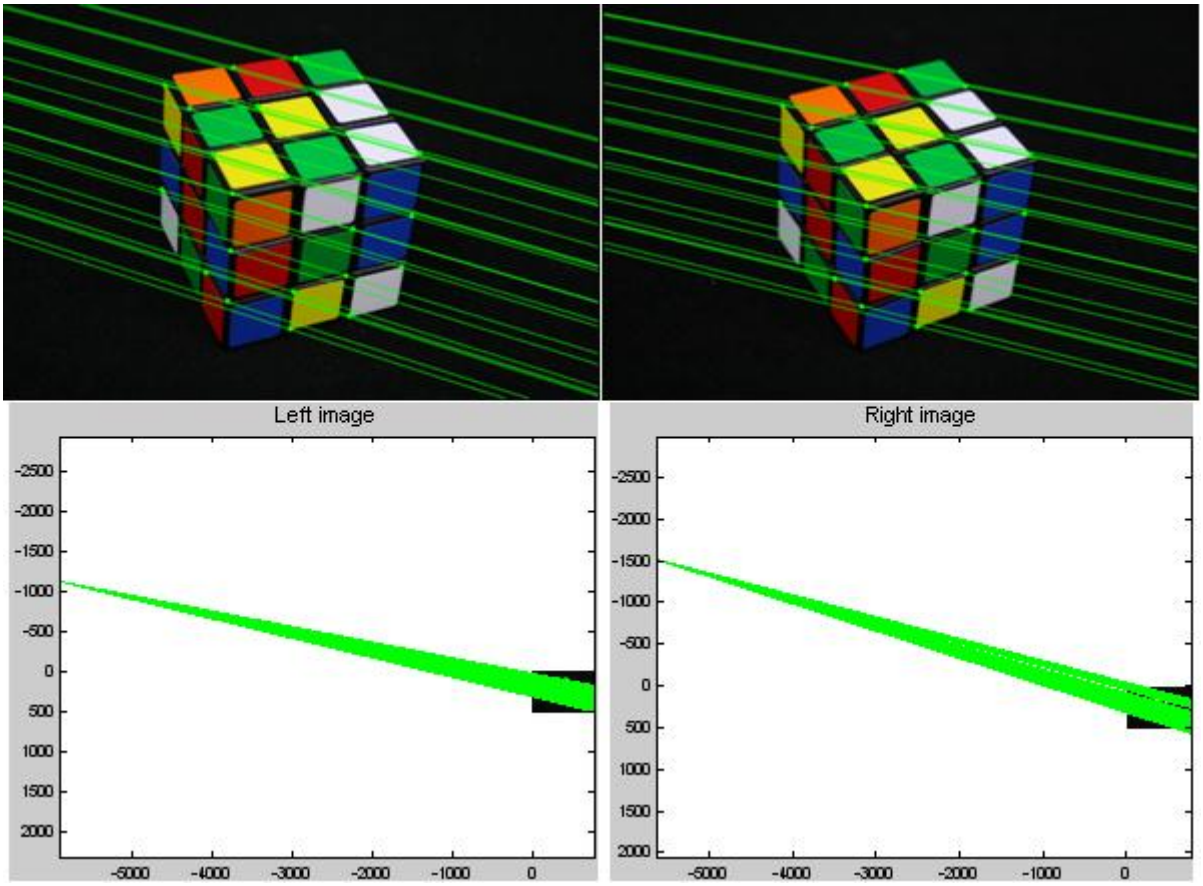

Fig. 4.6 – On the top, the epipolar lines (green lines) for a stereo image pair, on which the corresponding points must lie (green dots). On the bottom, the epipole as the point intersection of an epipolar line set. Both axes are in pixels units.

Thus, more robust methods of estimation are required. In general, the RANSAC (RANdom Sampling Consensus) algorithm [Fischler, 1981] achieves a robust estimation of the epipolar geometry. RANSAC is an iterative method to estimate matrix F from a pair of sets of observed image points that can contain outliers. It is also a non-deterministic approach in the sense that it produces a reasonable result only with a certain probability.

The RANSAC algorithm is summarized in Table 4.1. The algorithm terminates when the probability of finding a better ranked consensus set drops below a certain threshold.

4.2.3 Rectification

Rectification is the act of projecting two stereo images onto a common plane, such that pairs of conjugate epipolar lines, which are derived from the fundamental matrix, become collinear and parallel to the horizontal image axis, Fig. 4.7.

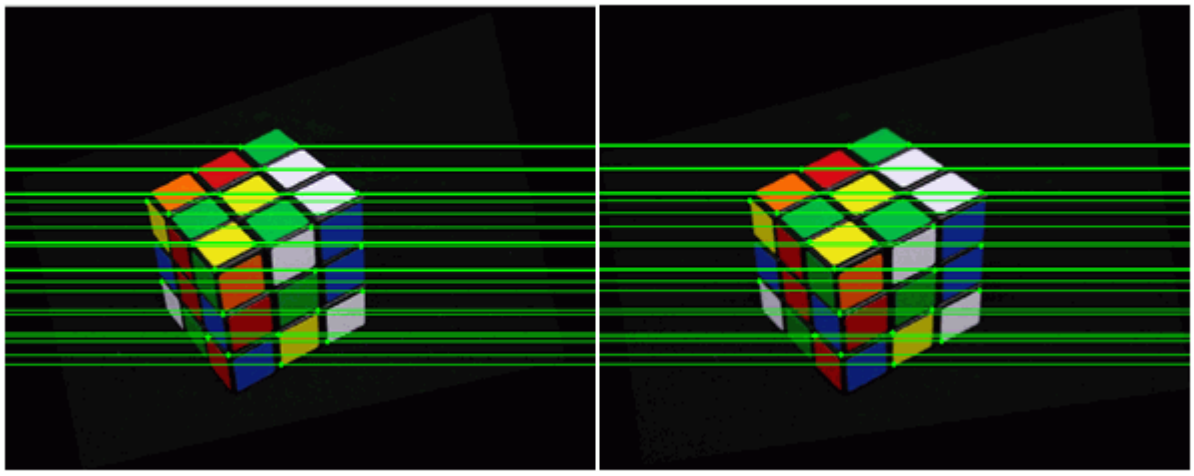

Fig. 4.7 – Rectified image pair: resampled images in which the epipolar lines run parallel with the image *u*-axis.

Performing this step simplifies the posterior process of dense matching, because the search problem is reduced to 1D, since there is no vertical disparity.

As the fundamental matrix is the essential matrix modified by the camera calibration parameters [Hartley, 2004], it is possible to rectify a stereo image pair into a common image space given enough knowledge of the camera parameters:

$$\mathrm{H} = \mathrm{K}_{new}\mathrm{R}\mathrm{K}_{old}\,, \tag{4.8}$$

where K_{old} and K_{new} are the intrinsic parameters of the "old" and "new" camera, respectively, and R is the rotation that is applied to the "old" camera in order to rectify it. This is denominated as calibrated (Euclidean) rectification. Geometrically, rectification is achieved by a suitable rotation of both image planes. Unfortunately, no knowledge of the camera parameters exists in the uncalibrated case. Essentially, the only information that is available must be obtained from the images themselves. However, if epipolar lines are to be transformed to lines parallel with the image horizontal axis, then the epipoles should be mapped to infinity, i.e., to the particular infinite point $(1,0,0)^T$. This leaves some degree of freedom to compute a projective transformation, H. If an inappropriate homography H is chosen, severe projective image distortion can take place [Hartley, 1998].

A few competing rectification methods are present in the literature (e.g. [Zhang, 1995], [Loop, 1999], [Isgrò, 1999]). In this Thesis, the method described in [Fusiello, 2008] was adopted, since it aims to achieve a good approximation to the calibrated epipolar rectification, that is referred as a *quasi*-Euclidean rectification.

Table 4.1 – Outline of the RANSAC algorithm.

RANSAC Algorithm	
Step 1:	Randomly select the 7 points required to determine the model parameters
Step 2:	Solve for the parameters of the model using the 7-point algorithm
Step 3:	Determine how many points from the set of all points fit with a predefined tolerance
Step 4:	If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold, re-estimate the model parameters using all the identified inliers and terminate; otherwise, repeat steps 1 through 4 (maximum of N times).

Assuming the knowledge of the fundamental matrix F and matching points $\mathbf{m}_i$ and $\mathbf{m}'_i$, most rectification methods exploit the fact that the fundamental matrix of a pair of rectified images has a very special form:

$$\overline{\mathrm{F}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.9}$$

Let H_R and H_L be the unknown rectifying homographies. The transformed matching points must satisfy the epipolar geometry of a rectified image pair:

$$(\mathrm{H}_R\mathbf{m}'_i)^T\,\overline{\mathrm{F}}(\mathrm{H}_L\mathbf{m}_i) = 0\,. \tag{4.10}$$

The fundamental matrix F between the image pair can be described as:

$$\mathrm{F} = \mathrm{H}_R^T\overline{\mathrm{F}}\mathrm{H}_L\,. \tag{4.11}$$

The approach proposed in [Fusiello, 2008], is to minimize the Sampson error [Hartley, 2004]; i.e. to minimize a first order approximation of the geometric reprojection error:

$$E_S = \frac{(\mathbf{m}_i'^T\mathrm{F}\mathbf{m}_i)^2}{\left\|\overline{\mathrm{F}}\mathrm{F}\mathbf{m}_i\right\|^2 + \left\|\mathbf{m}_i'^T\mathrm{F}\overline{\mathrm{F}}\mathbf{m}_i\right\|^2}\,. \tag{4.12}$$

Homographies H_R and H_L are forced to have the same structure as in the calibrated case. Each homography depends on five intrinsic plus three, concerning the rotation involved, unknown parameters.

The intrinsic parameters are reduced by making some suitable guesses: they remain unchanged during rectification, are equal for both images, have no skew, the principal point is in the image centre, and the aspect ratio is equal to one:

$$\mathrm{K}_R = \mathrm{K}_L = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.13}$$

The only remaining unknown is the focal length f. Combining Equations (4.8) and (4.11), the minimization of Equation (4.12) can be carried using a Levenberg-Marquardt algorithm and starting with all the unknown parameters set equal to zero.

4.2.4 Disparity map

In a stereo system, the 3D information of a scene can be represented by the disparity between stereo images. This step is also known as dense matching, where the matching is performed addressing as many pixels as possible. It is a necessary step as the goal is to recover the detailed geometry of an object.

It should be noted that without a significant amount of extra information and calculations, it is generally impossible to ascertain an exact depth measurement, but rather "planes" of depth can be isolated, which means to localize which parts of the scene are at relatively close depth level. Thus, a disparity map codifies the distances between object and camera: closer points have maximal disparity (usually 255 - white) and farther points get zero disparity (0 - black). For short, a disparity map gives perception of discontinuity in terms of depth, which is known as 2.5D reconstruction.

A comprehensive overview on dense stereo matching can be found in [Scharstein, 2002], [Lazaros, 2008], [Szeliski, 2010], among many others. In this Thesis, it the segment-based approach described in [Klaus, 2006] was addressed. This approach performs on four consecutive steps:

1. regions of homogeneous colour or intensity values are located using a colour segmentation method;
2. a local window-based matching method is used to determine the disparities of reliable points;

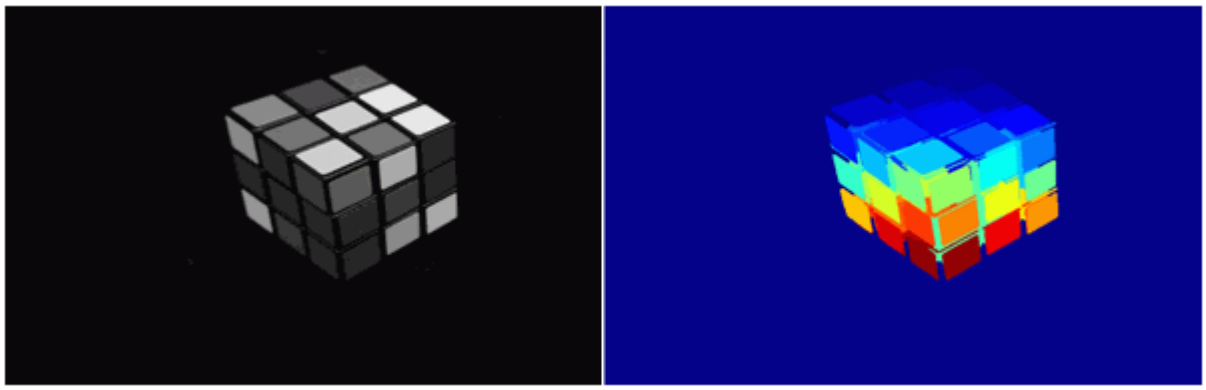

Fig. 4.8 – On the left, reference image, i.e. rectified left image of the stereo pair. On the right, colour representation of regions with homogeneous greyscale values.

3. a plane fitting technique is applied to obtain disparity planes that are considered as a label set;
4. finally, an optimal disparity plane assignment, based on optimal labelling, is approximated using belief propagation.

The first step is to decompose the left image, which is taken as reference, into regions of homogeneous colour or greyscale, Fig. 4.8. To accomplish this step, the mean-shift analysis approach described in [Comaniciu, 2002] was adopted. This approach assumes that disparity values vary smoothly in low-texture regions and that depth discontinuities only occur on region boundaries. It should be noted that over-segmentation is preferred, since it helps to meet the assumption in practice.

For the second step, in which local matching is performed, a self-adapting dissimilarity measure was used. It combines the SAD (Sum of Absolute intensity Difference) and SGRAD (Sum of GRadient Absolute Difference) cost functions:

$$\begin{aligned} C_{SAD} &= \sum_{w} I_L(u_i, v_i) - I_R(u_i + d_i, v_i) \\ C_{SGRAD} &= \sum_{w_u} \left|\partial_u I_L(u_i, v_i) - \partial_u I_R(u_i + d_i, v_i)\right| + \sum_{w_v} \left|\partial_v I_L(u_i, v_i) - \partial_v I_R(u_i + d_i, v_i)\right| \end{aligned}, \tag{4.14}$$

with I_L being the left image of the stereo pair which is the one taken as reference, I_R the left image, w a 3×3 square window surrounding point (u_i, v_i), w_u a surrounding window without the rightmost column, w_v a surrounding window without the leftmost column and d the optical flow between the stereo image pair.

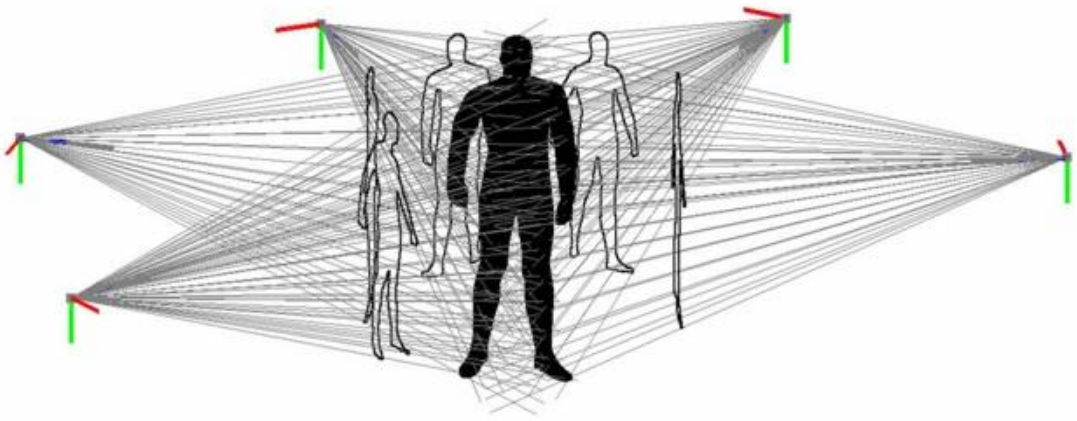

Fig. 4.9 – Visual hull defined by projecting the calibrated silhouettes together (from [Corazza, 2006]).

An optimal weighting ω between C_{SAD} and C_{SGRAD} is determined by maximizing the number of reliable correspondences that are filtered out by applying a cross-checking test, i.e., by comparing left-to-right and right-to-left disparity maps, and by choosing the disparity with the lowest matching cost:

$$C = (1-\omega)C_{SAD} + \omega C_{SGRAD}. \tag{4.15}$$

A set of disparity planes is derived using reliable correspondences. This third step is achieved by applying a robust plane fitting method, and a final refinement step is performed by applying a belief propagation optimization method [Felzenszwalb, 2006].

4.3 Volumetric-based reconstruction

For the 3D reconstruction of smooth objects, volumetric-based methods for 3D reconstruction have been quite popular for some time [Seitz, 1997]. These methods are based on SFS (Shape-From-Silhouettes) reconstruction methods: intersecting the visual cones generated by the silhouettes and the projection centres of each image, it is possible to determine a 3D model, Fig. 4.9. This 3D model is denominated as visual hull [Laurentini, 1994], a locally convex over-approximation of the volume occupied by the object under reconstruction.

Major two advantages of volumetric reconstruction using silhouettes are:

- it handles 3D shape reconstruction from arbitrarily placed cameras successfully;
- since no constraints on the camera positions are imposed, the solution is a global reconstruction, eliminating the need of partial reconstructions and merges.

Moreover, as photo-consistency is integrated into the volumetric method, the solution volume provides the tightest possible bound of the original object that can be computed from the given views.

In the following, a summary of the standard volumetric reconstruction method theory is presented.

Assume an object in the 3D space with an unknown volume shape O and a set of $i = \{1,...,k\}$ perspective projection silhouettes, S_i, obtained respectively from a set of known locations, C_i. This means that it is assumed that the camera(s) is (are) calibrated with $\mathrm{P}_i() : \Re^3 \to \Re^2$ the perspective projection function of viewpoint i. In other words, $\mathbf{m} \cong \mathrm{P}_i(\mathbf{M})$ are the image points of a 3D point $\mathbf{M}$ in space in the i^{th} image.

A volume A is said to exactly explain all silhouettes S_i if and only if its projection onto the i^{th} image exactly coincides with S_i, i.e., $\mathrm{P}_i(A) = S_i$. If there exists at least one non-empty volume which explains the silhouette images exactly, the set of silhouette images is said to be consistent. Normally, a set of silhouette images obtained from an object is consistent, unless there are camera calibration errors or silhouette image noise.

A visual hull is therefore defined as the intersection of all the visual cones, Fig. 4.10, each formed by projecting the silhouette image S_i into the 3D space through the camera centre C_i. Generally, for a consistent set of silhouette images S_i, there is an infinite number of volumes, including the object O itself, that exactly explain the silhouettes.

A second definition for visual hull is that it is the largest possible volume that explains S_i. Though abstract, this second definition implicitly expresses one of the useful properties of the visual hull: it provides an upper bound on the object which forms the silhouettes. The upper bound given by the visual hull gets tighter if we increase the number of distinct silhouette images. Therefore, the accuracy of the reconstruction obtained depends on the number of images used, the positions of each viewpoint considered, the precision of the camera calibration and the complexity of the object's shape [Mundermann, 2005].

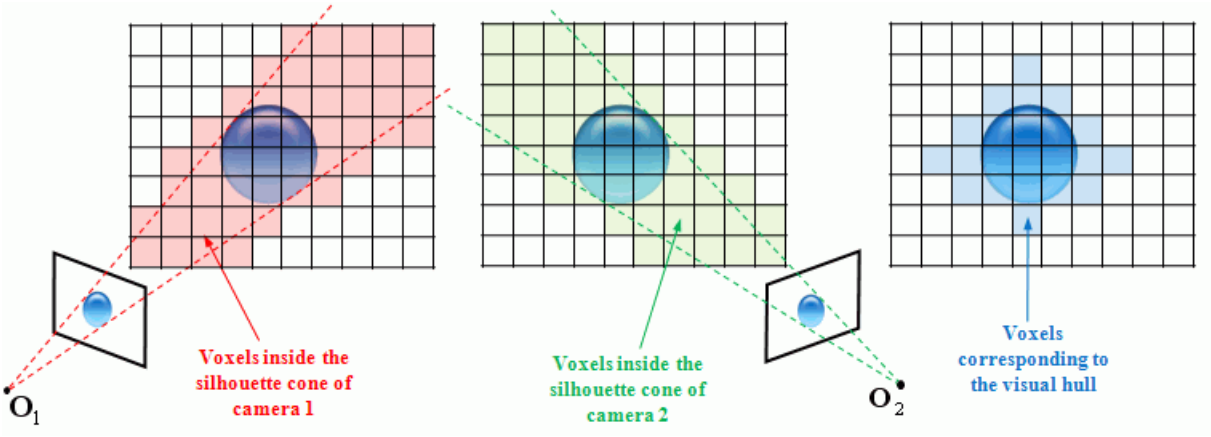

Fig. 4.10 – Silhouette reconstruction based on voxel carving when two cameras with optical centres $\mathbf{O_1}$ and $\mathbf{O_2}$ are used.

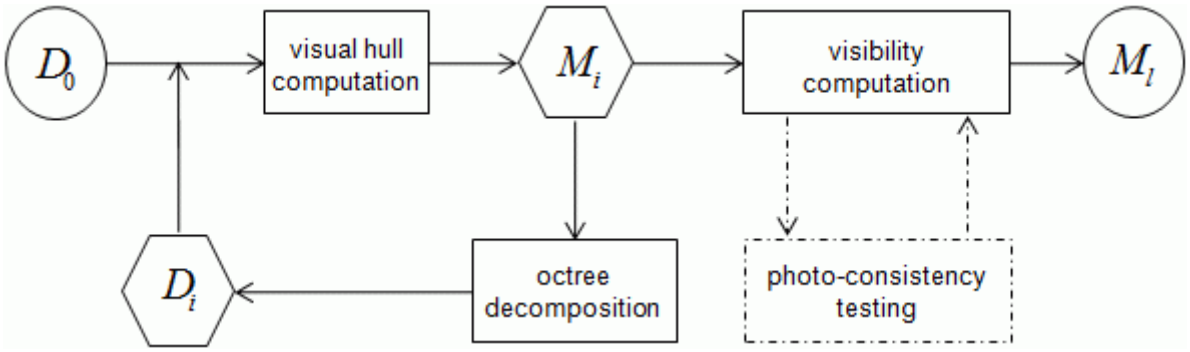

Fig. 4.11 – Developed volumetric reconstruction methodology.

Asymptotically, if we have every possible silhouette images of a convex object, the visual hull is exactly equal to the object. If the object is not convex, the previous is not guaranteed.

Volumetric methods represent the 3D space model O by using voxels, which are regular volumetric structures also known as 3D pixels. The space of interest is divided into $N \times N \times N$ discrete voxels v_n, with $n = \{1,...,N^3\}$, that are then classified into two categories: inside and outside. If a particular voxel does not belong to the object, it is set as outside; whereas, voxels within the object remain as inside and can additionally be coloured. The union of all the inside voxels is an approximation of the visual hull. Table 4.2 presents an overview of the volumetric method described above for the visual hull computation.

Fig. 4.11 presents the general structure of the developed volumetric method, which includes three major steps: visual hull estimation, voxel visibility computation and photo-consistency estimation.

Table 4.2 – Outline of the standard voxel-based volumetric reconstruction method based on silhouettes.

Volumetric visual hull computation algorithm
Divide the space of interest into $N \times N \times N$ voxels v_n.
Initialize all the N^3 voxels as inside voxels.
For $n = 1$ to N^3 do
For $i = 1$ to k do
Project v_n into the i^{th} image plane, using the projection function P_i;
If the projected area $\mathrm{P}_i(v_n)$ lies completely outside S_i, then v_n is outside.
The visual hull is the union of all inside voxels.

To the best of our knowledge, the implemented methodology is the first to include photo-consistency testing into an octree-based volumetric reconstruction method.

The following sub-sections describe the steps of the novel algorithm, pointing out the main differences relatively to the standard voxel-based volumetric methods.

4.3.1 Image acquisition

In the volumetric methodology developed, two image sequences should be acquired:

- a first one, acquired moving a planar calibration pattern freely in 3D space;
- for the second sequence, the object to be reconstructed is placed above or in front of the calibration pattern, and the images are acquired by moving the camera freely in space.

The necessity to acquire two image sequences was due to the fact that the calibration pattern is partially occluded by the object of interest. Although the calibration algorithm works even if the pattern is not completely visible, it may occur that the size of the object comparatively to the calibration pattern is such that the number of pattern points available for calibration is small, which will lead to an increase in the calibration error.

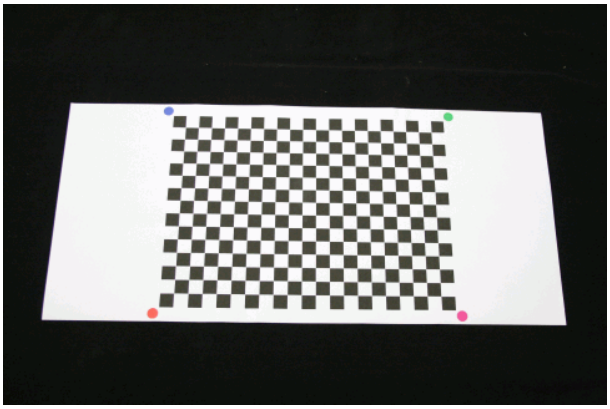

Fig. 4.12 – Planar calibration pattern used in this Thesis to perform the camera calibration using Zhang's method (Section 3.3.3).

No restrictions are made on the number of images acquired. The only constraint is to fully observe the calibration pattern on all acquired images and to keep the camera's intrinsic parameters unchanged during the acquisition process, i.e., the zoom and focus should remain constant.

The calibration pattern is an adaptation of the classical chessboard by adding four coloured circles, Fig. 4.12. The classical black and white chessboard was improved by adding four coloured circles in its extremities, for correct detection of the pattern orientation in an image set captured from multiple, unconstrained positions, as is demonstrated next.

4.3.2 Camera calibration

The camera used is calibrated in order to find the transformation that maps the 3D world in the associated 2D image space. The calibration procedure developed was based on Zhang's algorithm [Zhang, 2000] (see Section 3.3.3). The camera's intrinsic (focal length and principal point) and distortion parameters (radial and tangential) are obtained from the first image sequence. Then, using the second image sequence, the extrinsic parameters (3D rotation and translation) associated with each viewpoint considered in the reconstruction process are determined.

To recover the calibration parameters for all images, Zhang's method sets the world coordinates on one of the four outer vertices of the chessboard pattern and requires the knowledge of the remainder world coordinates, X_W and Y_W, for all visible vertices of the pattern's squares; .i.e. the calibration pattern is on the world plane $Z_W = 0$.

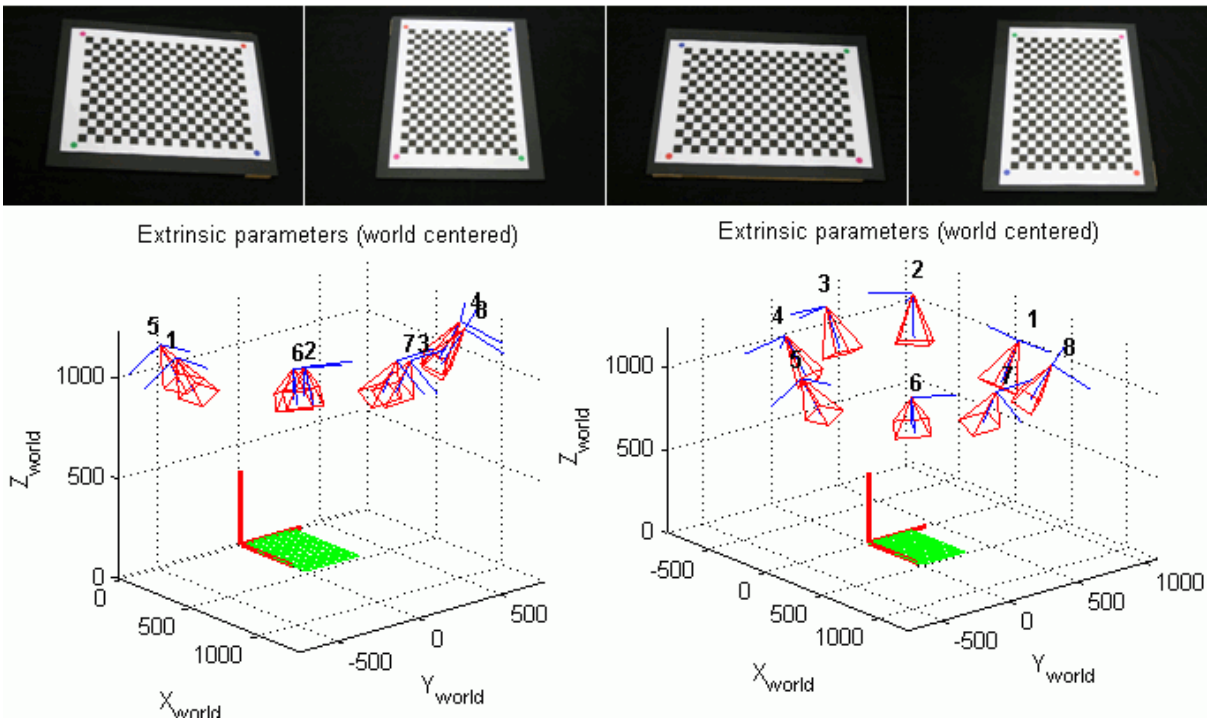

Fig. 4.13 – On the top: four images of the chessboard calibration pattern used, acquired by performing a full rotation around it with the camera. On the bottom: 3D representation of the camera's poses, obtained without using the pattern's circles as reference (on the right) and obtained by using the pattern's circles as reference (on the left); for both graphs, the calibration pattern is represented as a green grid and the green pyramids are the camera's poses for each acquired image.

Automatic and robust algorithms are available to detect and match chessboard vertices used as calibration points (e.g., [Shu, 2003], [Wang, 2010]). However, they commonly fail when the pattern is partially occluded or at steep angles between pattern and image planes. Moreover, it is impossible to distinguish the origin of the world coordinate system on images with the calibration pattern suffering 180º rotations around the z-axis (turned upside-down). For example, when using images of a pattern rotating on a turntable device, the camera calibration returns a set of projection matrices that describe a semi-circle in the 3D space, Fig. 4.13.

The manual identification of the points in a chessboard pattern is monotonous and can be unreliable. Some solutions currently available make use of self-identifying markers, with some or all squares represented in binary code (e.g. [Fiala, 2004], [Atcheson, 2010], [Grochulla, 2011]), or using a coloured pattern (e.g. [Nowakowski, 2009]).

In this Thesis, the choice of an appropriate calibration object was a compromise between the calibration accuracy and the manufacturing complexity of the target itself. Planar

calibration targets are by far the most used both in experimental and industrial setups in virtue of their ease of construction.

It was equally important to select an appropriate feature pattern for the detection of the reference points. Usual choices are symmetric markers such as circular points (e.g. [Heikkilä, 2000]) or chessboards. Both these marker types have shown to be detectable with high sub-pixel accuracy, but the chessboard pattern deals better with the misplacement error introduced by radial distortion [Mallon, 2007].

Therefore, the chessboard pattern was chosen because of its simple geometry, proven good results and has features which are easy to detect and order. As such, the new algorithm developed to extract the chessboard vertices on the second image sequence includes the following steps:

1. image binarization, Fig. 4.14: in order to convert from colour to binary images an adaptive thresholding is applied, by computing an average of a window around each pixel:

$$B(u,v) = \begin{cases} 1, \text{ if } I(u,v) > \text{avg}_w(I(u,v)) \\ 0, \text{ otherwise} \end{cases}. \tag{4.16}$$

   The size of the window, w, was chosen to be approximately 10% of the minimum imaged square area. The adaptive threshold is useful when, relatively to the general intensity gradient, there are strong illumination or reflectance gradients.

2. squares segmentation: image dilation, using a 3×3 structuring element, is performed to properly separate the chessboard squares, Fig. 4.15;

3. retrieval of the pattern squares and circles, Fig. 4.16: all image contours are retrieved and polygonized; first, non-convex, inner contours and contours with too small or too big areas are rejected; from the remaining, circles are chosen if they have more than 4 sides and its centre colour belongs to one of the four coloured pattern circles; finally, the non-quadrangles are rejected, which are those with more than 4 sides or which are more rectangular than square;

4. finding square neighbours: due to image dilation, pattern squares were detached, originating four vertices for each square; in order to remove duplicates, connected vertices are detected and their average pixel location is retrieved;

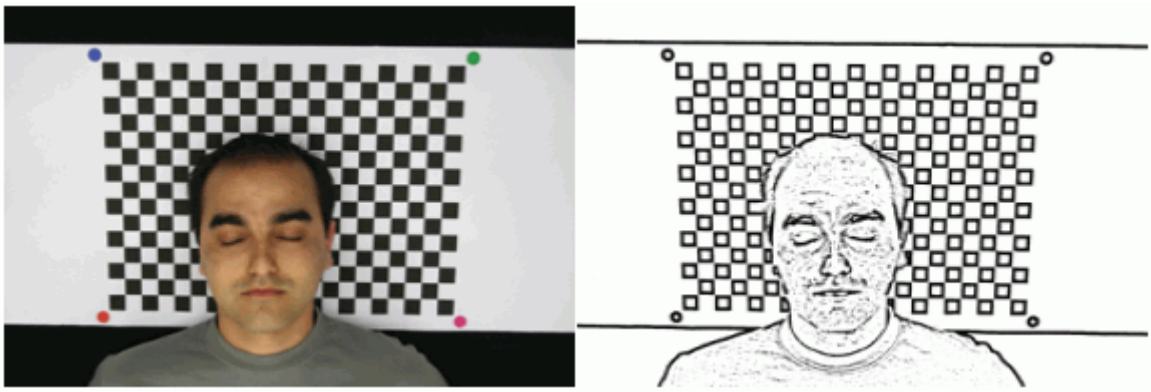

Fig. 4.14 – Original image (on the left) and the correspondent binary image obtained using an adaptive thresholding (on the right).

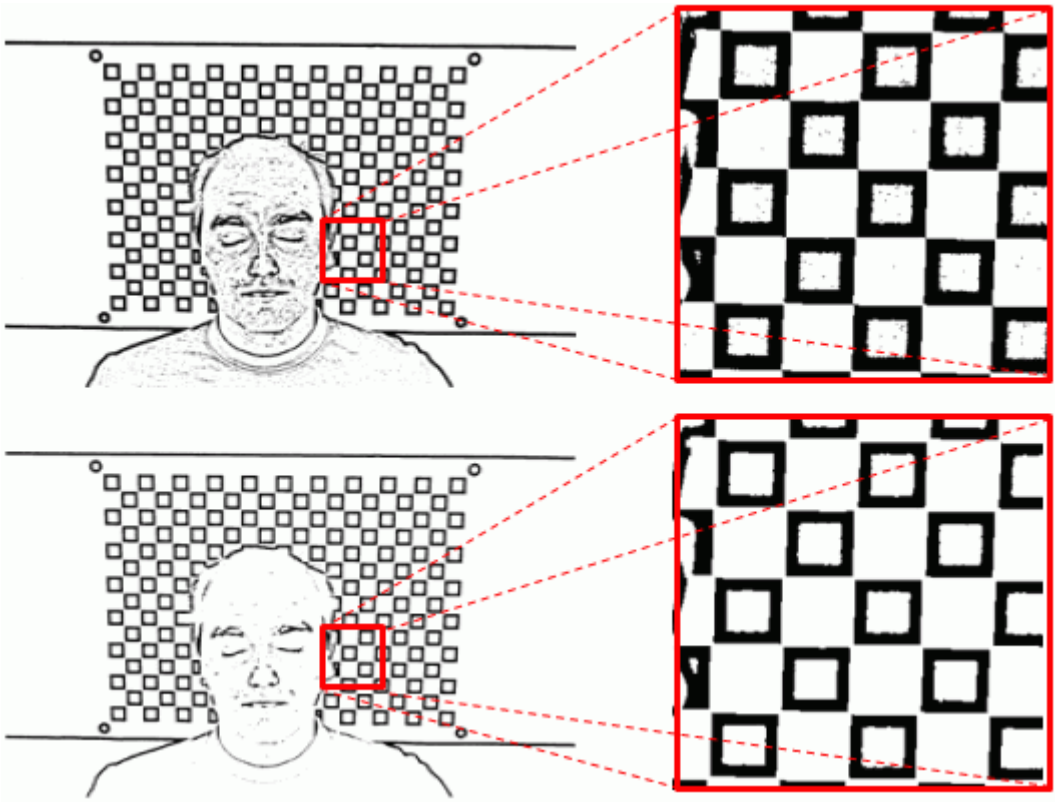

Fig. 4.15 – On the top: original binary image and a close-up to observe some squares that share the same vertex. On the bottom: dilated binary image and a close-up to confirm the good segmentation between adjacent squares.

5. vertex location refinement: taking two points, one is the found vertex, q, and the other is a nearby pixel, p, the dot product of the gradient at p and the vector $q - p$ is computed. If the nearby pixel p is on a flat region or over an edge, the dot product is always zero, Fig. 4.17. A linear system is formed, by stacking several equations of the dot product for different locations of p, which can be solved in order of q. One example result can be seen in Fig. 4.18.

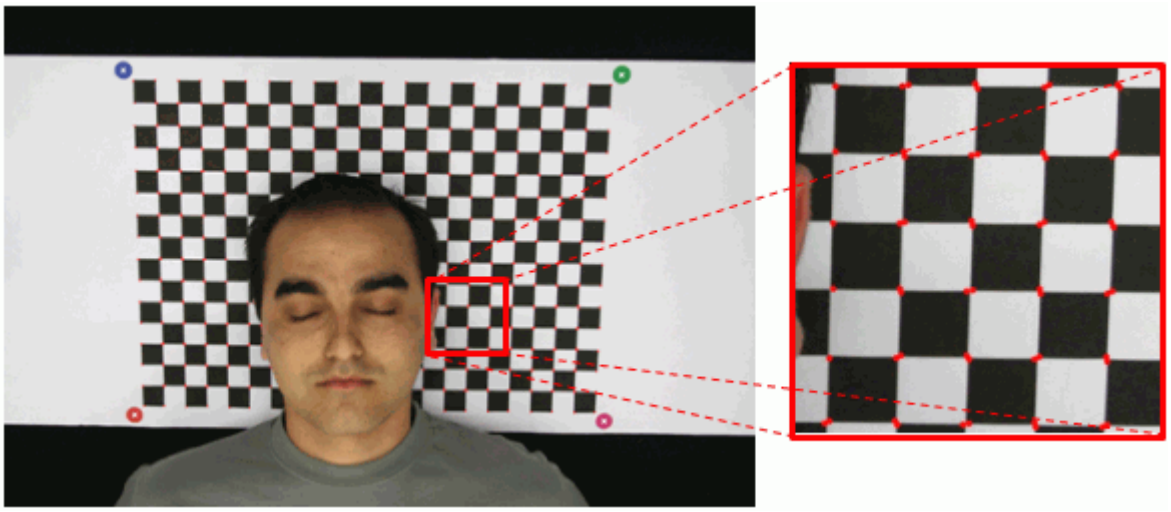

Fig. 4.16 – On the left: result of the square vertices (the red dots) and circles' centre detection (the white crosses). On the right: close-up of a region in which vertices of adjacent squares were detected twice.

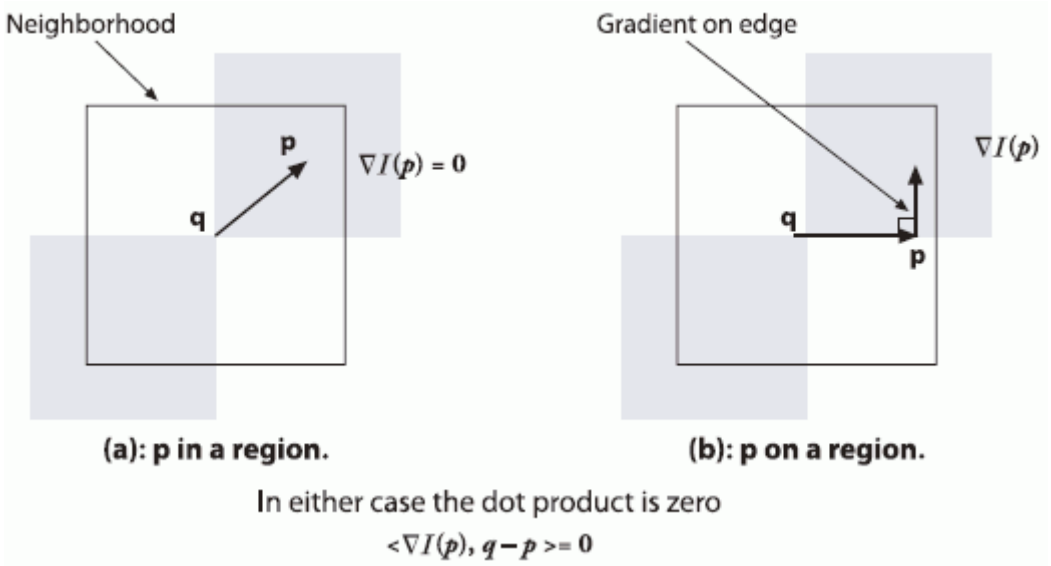

Fig. 4.17 – Finding vertices to sub-pixel accuracy: (a) the image area around the point p is uniform and so its gradient is 0 (zero); (b) the gradient at the edge is orthogonal to the vector q-p along the edge; in either case, the dot product between the gradient at p and the vector q-p is 0 (zero) (from [Bradski, 2008]).

To match the chessboard vertices between images, $\mathbf{p}$ and $\mathbf{p}'$, in the second sequence, the 4 circle centres are used to determine the homography matrix, H, which relates both images. Images of points on a plane are related to corresponding image points in a second view by a projective transform (homography) [Hartley, 2004]:

$$\mathbf{p} \cong \mathrm{H}\mathbf{p}' . \tag{4.17}$$

With the homography and the chessboard vertices from the previous image, all image vertices from the next image in the sequence are estimated, using Equation (4.17).

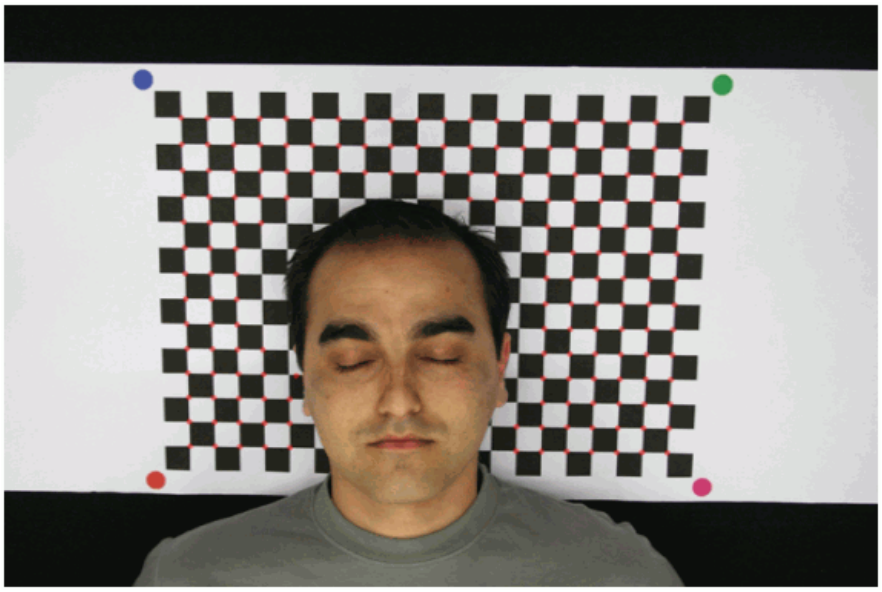

Fig. 4.18 – Final result of the pattern vertices detection (the red circles).

Vertices are matched with the previous image if the distance between estimated and founded vertices is inferior to 20 pixels, Fig. 4.19.

Having all images with chessboard vertices correctly ordered and matched, the camera calibration is a straightforward process. The first image sequence provides the camera's intrinsic parameters and for the second sequence, with the object to be reconstructed on the calibration plate, the camera's extrinsic parameters, i.e. camera's pose, are computed for each image.

To verify the calibration results' accuracy, 3D coordinates of the chessboard pattern were reprojected and the average and standard deviations (in pixels) were calculated. Another way to verify the quality of the camera calibration results was to graphically represent the viewpoints considered in the second image acquisition process, considering the world coordinate system fixed on the first vertex of the chessboard pattern.

4.3.3 Image segmentation

To obtain the object silhouettes from the input images, image segmentation is performed. This step is required to build the visual hull. Also, since the calibration pattern used is imaged with the object to be reconstructed, if not segmented it will not be considered has scene background and, consequently, will be reconstructed as if it was part of the object of interest.

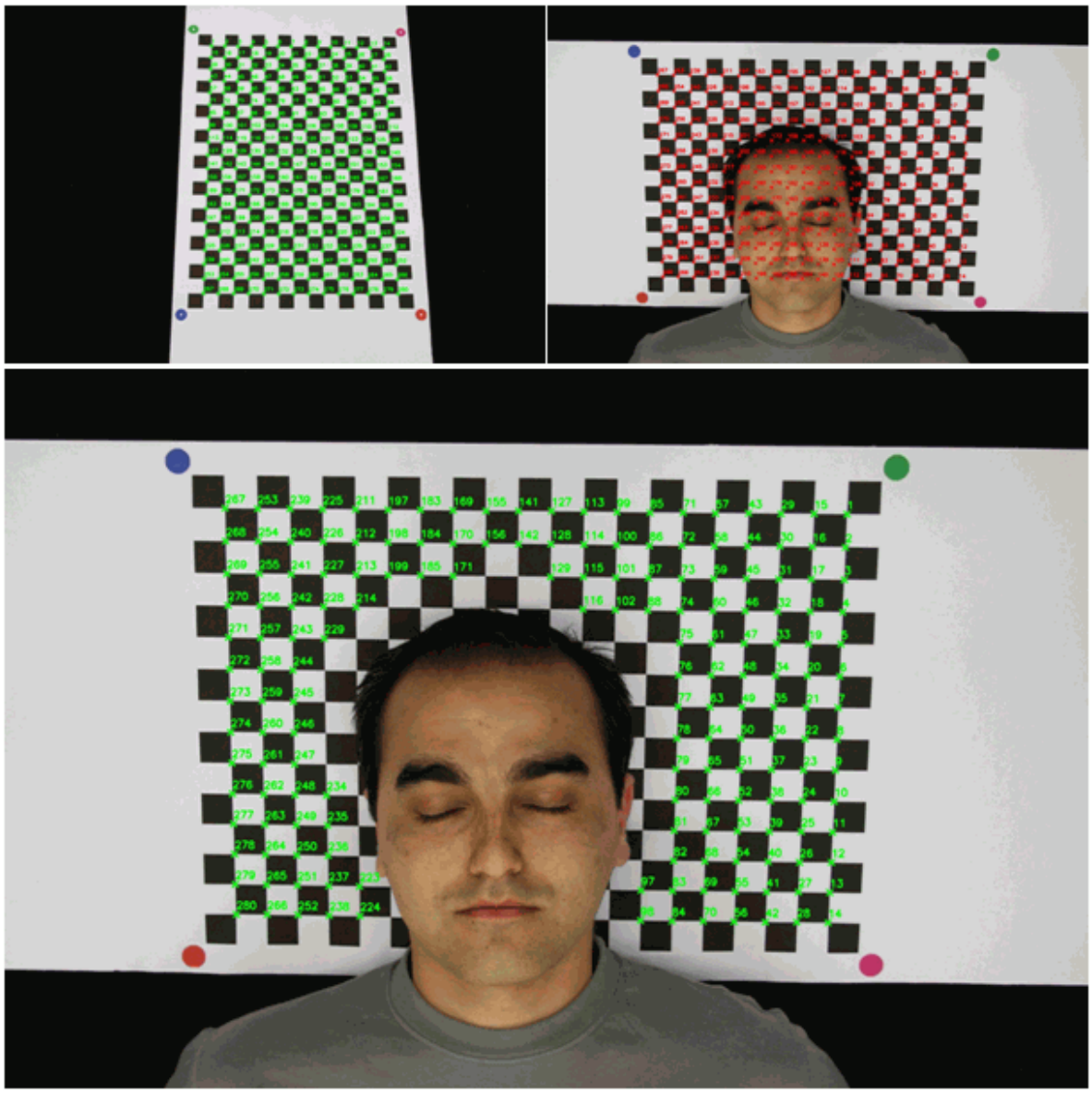

Fig. 4.19 – On the top: green crosses (on the left) represent vertices' locations from last image of the first sequence and red crosses (on the right) are the estimated chessboard vertices, using calculated homography. On the bottom: final result, with the found chessboard vertices correctly ordered.

Since the main goal of this Thesis concerns the 3D reconstruction of human body parts, a skin detection method was developed for the segmentation process. Skin detection can be defined as the process of selecting which pixels of a given image correspond to human skin.

It may be naively considered that skin detection is a trivial task as the human visual system can easily detect and differentiate skin surfaces; however, it is not easy to train a computer to do so.

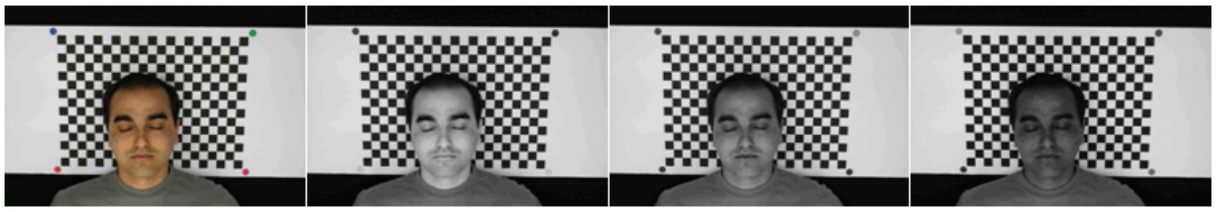

Fig. 4.20 – From left to right: original image, and greyscale images correspondent to the R, G and B channels.

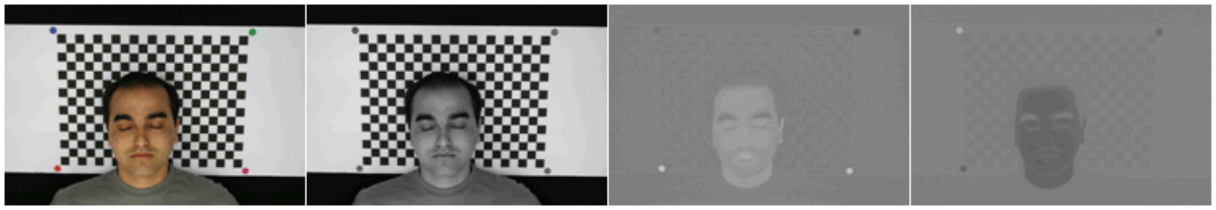

Fig. 4.21 – From left to right: original image, and greyscale images correspondent to Y, Cr and Cb channels.

The inspiration to use skin colour analysis for initial classification of an image into foreground and background regions stems from a number of simple but powerful characteristics of skin colour. Under certain lighting conditions, colour is orientation invariant. The major difference between skin tones is regarding the intensity, e.g. due to varying lighting conditions and different human races [Menser, 2000]. Also, the colour of human skin is different from the colour of most other natural objects.

One important factor that should be considered while building a statistical model for skin is the choice of a colour space. Several colour spaces have been proposed in the literature for skin detection applications.

The RGB (Red-Green-Blue) is the default colour space for most available image formats, Fig. 4.20. However, high correlation between channels and mixing of chrominance and luminance data make RGB a not very favourable choice for colour analysis. Any other colour space can be obtained from a linear or non-linear transformation from RGB space.

Here, the YCbCr (Luminance, Blue-difference chrominance and Red-difference chrominance) orthogonal colour space was used, Fig. 4.21. Segmentation of skin coloured regions becomes robust if only the chrominance component is used in analysis. Therefore, the variations of the luminance component are eliminated as much as possible by choosing the CbCr plane (chrominance components) of the YCbCr colour space to build the model. Research has shown that skin colour is clustered in a small region of the chrominance space [Jones, 2002]. The orthogonal colour spaces reduce the redundancy present in RGB colour channels and represent the colour with statistically almost independent components.

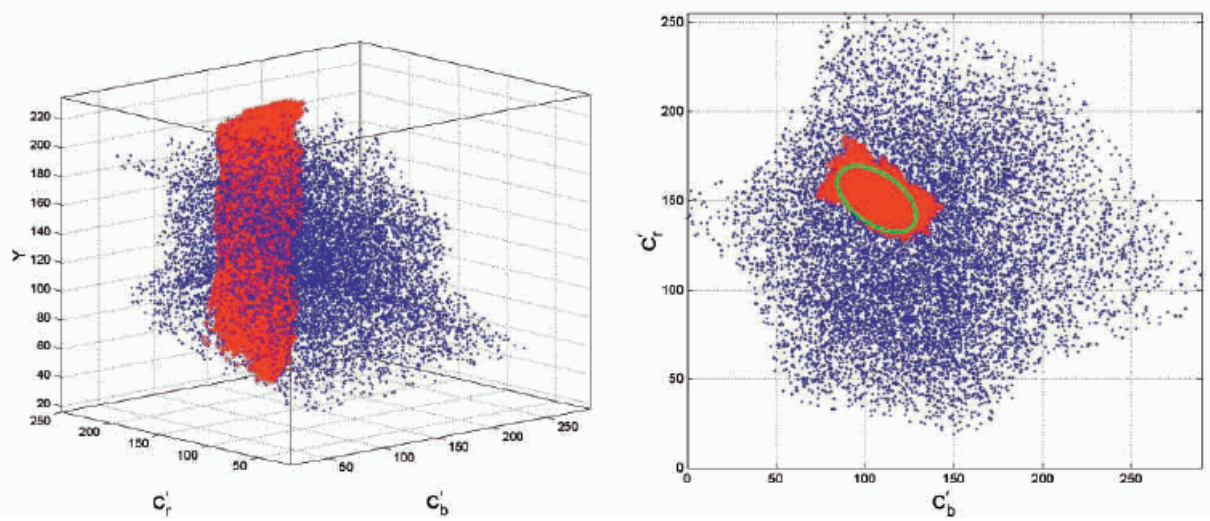

Fig. 4.22 – On the left: non-linear transformation of the YCbCr colour space (blue dots represent the reproducible colour in a monitor, and the red dots are the skin colour samples). On the right, is a 2D projection, in which the elliptical skin model is overlaid on the skin cluster (from [Hsu, 2002]). For both graphics, the axis units represent possible colour values for individual pixels in a monitor (0-255).

The transformation simplicity and explicit separation of chrominance (Cr and Cb) and luminance (Y) components, make these spaces an even more favourable choice for skin detection:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ C_r &= R - Y \\ C_b &= B - Y \end{aligned} \quad . \tag{4.18}$$

Then, the image pixels are compared against an elliptical cluster model for skin tones, Fig. 4.22:

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} C_b - c_x \\ C_r - c_y \end{bmatrix}, \\ R &= \frac{(x - e_{c_x})^2}{a^2} + \frac{(y - e_{c_y})^2}{b^2}, \end{aligned} \tag{4.19}$$

where c_x, c_y, θ, e_{c_x}, e_{c_y}, a and b are parameters evaluated from training samples of skin patches [Hsu, 2002]. When a pixel chromaticity, channels C_r and C_b, leads to a pair of values x and y that is inside the ellipse, $R \leq 1$, then this colour corresponds to the skin colour, Fig. 4.23.

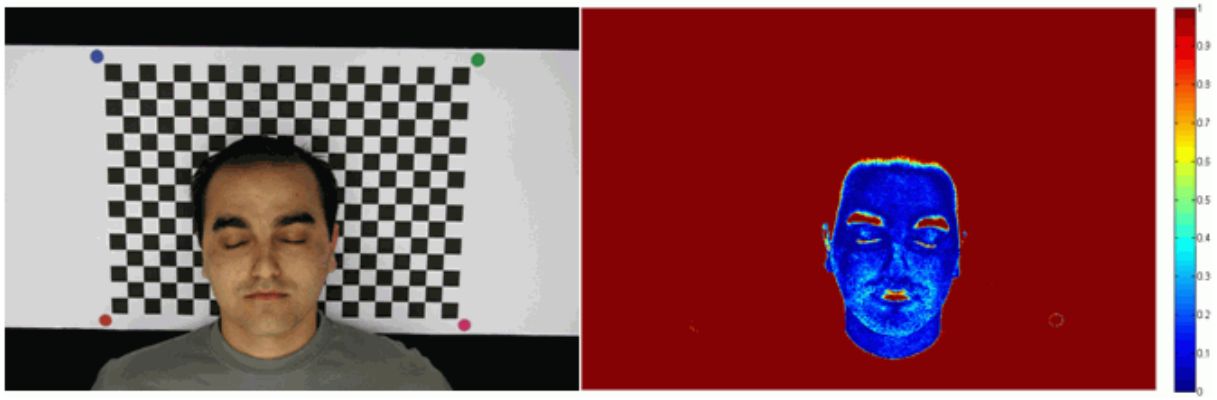

Fig. 4.23 – Original image (on the left) and its pixel classification accordingly to the adopted elliptical model (on the right).

Fig. 4.24 – Skin colour segmentation using the elliptical model adopted (on the left) and selection of the biggest contour (on the right).

Some results of this model are shown in Fig. 4.24, where one can observe some false-positive background pixels, in the mouth and eyebrow areas. These are eliminated by finding the contour with the biggest area, since it was assumed that the imaging system is acquiring one person, i.e. a single fully-connected object.

4.3.4 Volumetric reconstruction

Combining the image sequence and associated silhouette images, together with the parameters found by the calibration procedure, the objects' models were built using two volumetric methodologies: one only based on silhouettes, and a second one that combines silhouettes and photo-consistency criterion. The two methodologies allowed for the evaluation of the influence on the 3D reconstruction accuracy of some of the algorithms' parameters, such as the type of projection used for the voxels (rectangular or accurate), the initial volume size defined and the effect of the colour information.

Volumetric approaches for silhouette-based reconstruction are quite robust, somewhat easy to implement and guarantee results proportional to the chosen resolution. Further, they can be accelerated using dedicated graphics hardware (e.g., [Hasenfratz, 2004]).

a) Voxel projection

Volumetric structure from silhouettes is generally based on voxel classification: a voxel remains part of the estimated shape if it lies inside the visual hull. There are two choices available for the classification process:

- voxel classification is performed in the 3D space: the silhouettes are each back-projected as a cone, and each voxel is tested to see if it lies inside of these cones;
- voxel classification is performed in the images: each voxel is projected into each image to determine if it lies inside the silhouette.

It is generally more efficient to perform 2D boolean operations on shapes, and thus the second approach was chosen.

The simplest voxel projection onto an image plane (footprint) is a single point, Fig. 4.25a. Here, only the center of the voxel is projected, which leads to some problems such as: the rounding operation used to identify the pixel and artifacts on the 3D reconstruction due to the excessive simplification of the projection.

The accurate footprint of a voxel has to consider its cubic shape. All 8 cube vertices are projected into the image plane, and the footprint is its convex hull, Fig. 4.25b.

To determine the 2D convex hull from a given set of points, a variant of the Graham Scan algorithm introduced in [Andrew, 1979] was used, also known as Andrew's Monotone Chain Algorithm. It is a $O(n \log n)$ algorithm to construct a convex hull given a set of n unsorted 2D points. It does so by first sorting the image points lexicographically (first by u-coordinate, and in case of a tie, by v-coordinate), and then constructing the upper and lower hulls of the points. An upper hull is the part of the convex hull that is visible from above the line segment defined by the two points containing the rightmost and the leftmost X coordinates, respectively. On the other hand, the lower hull is the remaining part of the convex hull. It runs from its rightmost point to the leftmost point in counter clockwise order, for both upper and lower hulls. Once the two hull chains have been found, the final convex hull is the union of both upper and lower hulls.

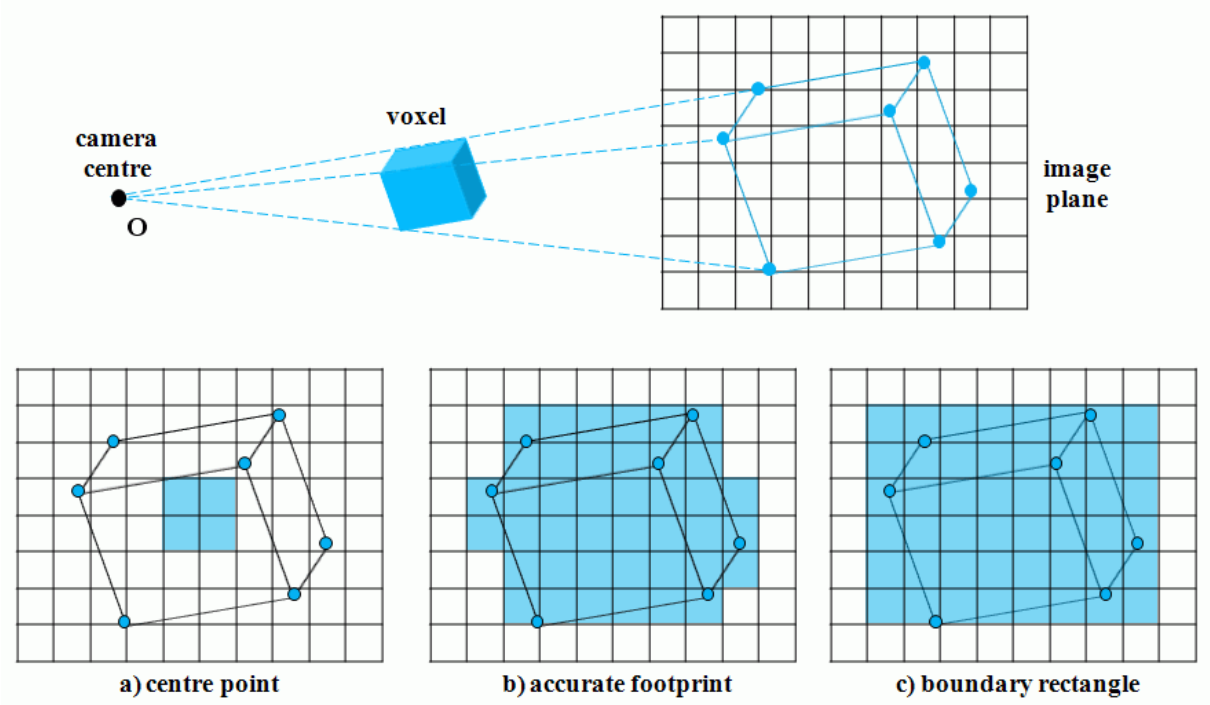

Fig. 4.25 – Techniques for the image projection of the voxels.

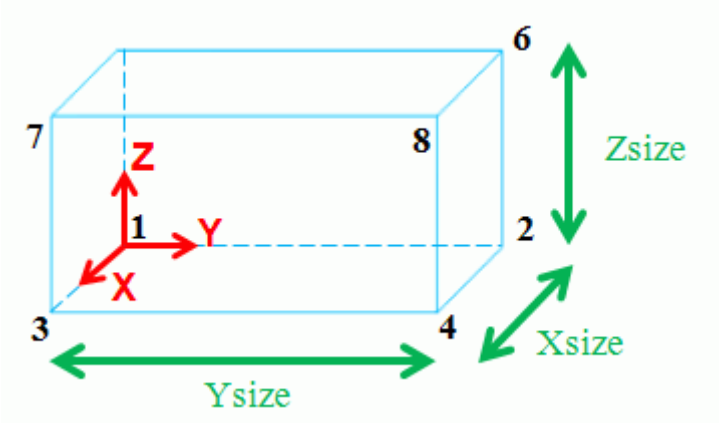

Fig. 4.26 – Voxel's vertices indexation for footprint determination.

Usually, a voxel is a regular cube. However, a parallelepiped voxel was implemented, i.e., not all sides of the voxel are required to have the same size, Fig. 4.26. This improvement of the classical voxel shaped reconstructions allowed for an unconstrained determination of the initial volume. The voxel vertices' coordinates are defined as follows:

- vertex 1: [X, Y, Z];
- vertex 2: [X, Y + Ysize, Z];
- vertex 3: [X+ Xsize, Y, Z];
- vertex 4: [X+ Xsize, Y + Ysize, Z];
- vertex 5: [X, Y, Z + Zsize];

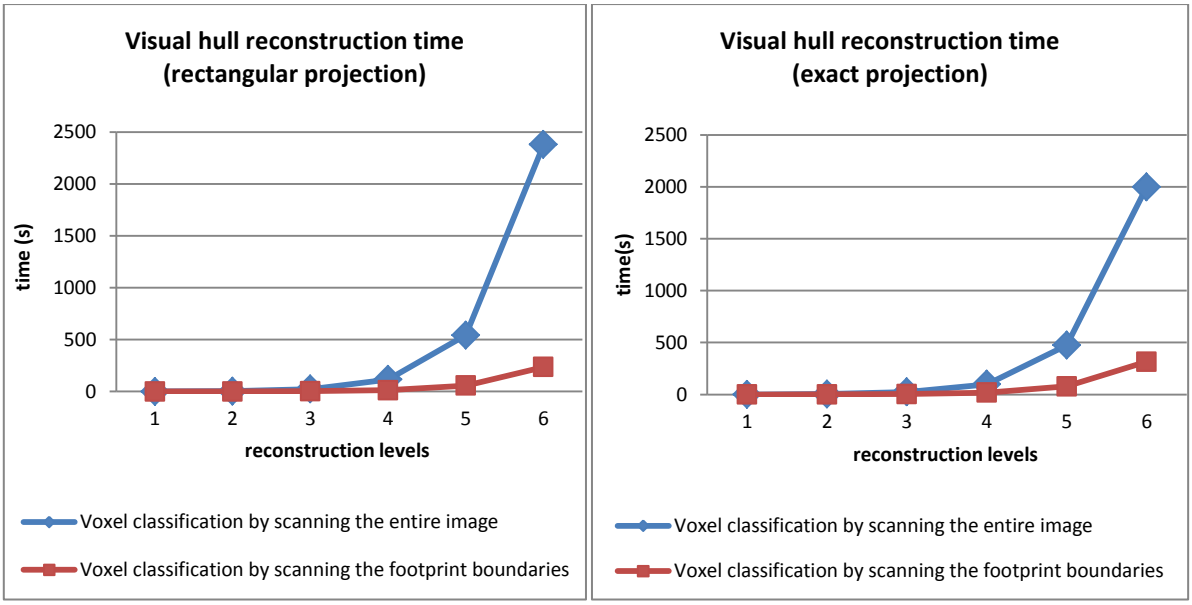

Fig. 4.27 – Comparison of the visual hull reconstruction times using the same parameters and regarding the same object, and only varying the image scan processing during the voxel classification process.

- vertex 6: [X, Y + Ysize, Z + Zsize];
- vertex 7: [X+ Xsize, Y, Z + Zsize];
- vertex 8: [X+ Xsize, Y + Ysize, Z + Zsize].

The eight vertices are back-projected onto all acquired images and the projected vertices are inputs for the Monotone Chain Algorithm, which returns an either 4- or 6-sided footprint polygon for each image, like a regular cubic-shaped voxel does, which is subsequently filled using image contour filling. Afterwards, 2D pixel comparisons are performed between the voxel footprint and the correspondent silhouette image.

Since this is one of the steps that contributes more to the overall processing time of a volumetric reconstruction method - the other is the computation of the voxel's visibility -, it is important to reduce the amount of comparisons, Fig. 4.27. An improvement contribution achieved in this Thesis is the approach developed to perform such comparisons using a rectangle defined by the minimum and maximum values of the *u*- and *v*- pixel coordinates of the voxels' footprints. This option revealed to be effective, since the projection of a voxel is a convex polygon; therefore, it is only necessary to track the right and left edges from the top to the bottom of the correspondent polygon.

A more computationally attractive technique to approximate the accurate footprint is to use the bounding rectangle that can be computed as the bounding box of the projections of the eight vertices of a voxel, Fig. 4.25c.

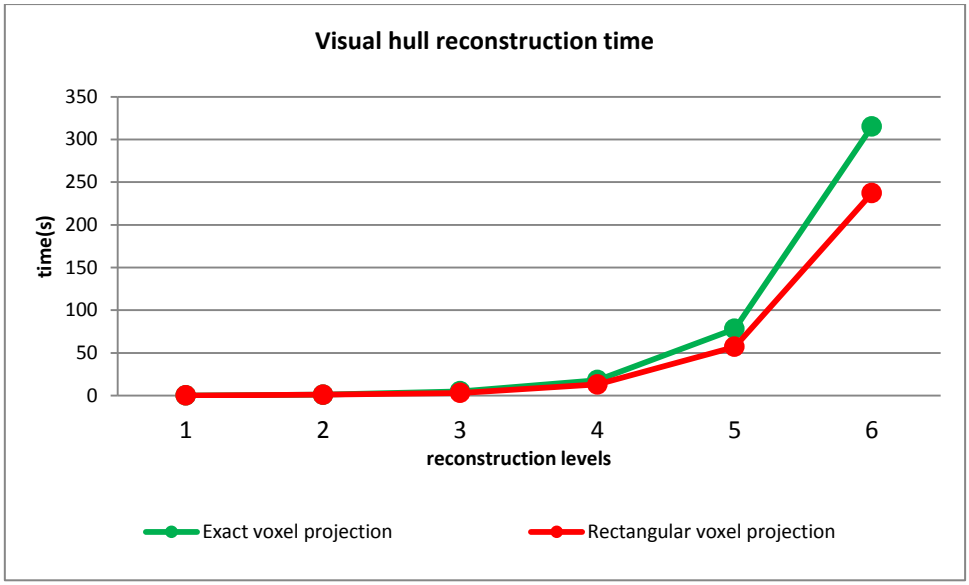

Fig. 4.28 – Comparison of the visual hull reconstruction times for the same object, but varying the voxel projection method: exact versus rectangular and using the improved image scan processing during the voxel classification process.

With the bounding box defined, a 4-sided footprint polygon is determined, which does not need to be filled neither requires comparisons with the silhouette. One must just search over the silhouette within a ROI (Region Of Interest) defined by the 4-sided footprint, and so decrease the computational effort of the reconstruction algorithm, Fig. 4.28.

b) Visual hull computation

In this Thesis, the visual hull is computed following an octree structure. The concept of octrees was introduced in [Chien, 1984]. 3D octrees are hierarchical structures, based on a recursive subdivision of the 3D space into eight octants, also known as voxels, which are usually regular cubes, Fig. 4.29. With an octree, cubes can be recursively subdivided until a desired resolution is reached. Starting from an initial resolution, each parent voxel is divided into eight child voxels. The procedure is repeated until the desired resolution level L is obtained. Octrees are a computationally efficient way to represent objects in the 3D space; especially, if the objects are highly coupled, as in the case of the human body.

The multi-resolution volumetric reconstruction starts with a coarse resolution (8 voxels) of the 3D voxel space, D_0, delimitating a parallelepiped volume containing the object of interest. Octree decomposition is performed on voxels classified as ambiguous. Voxel classification is done using the following criteria:

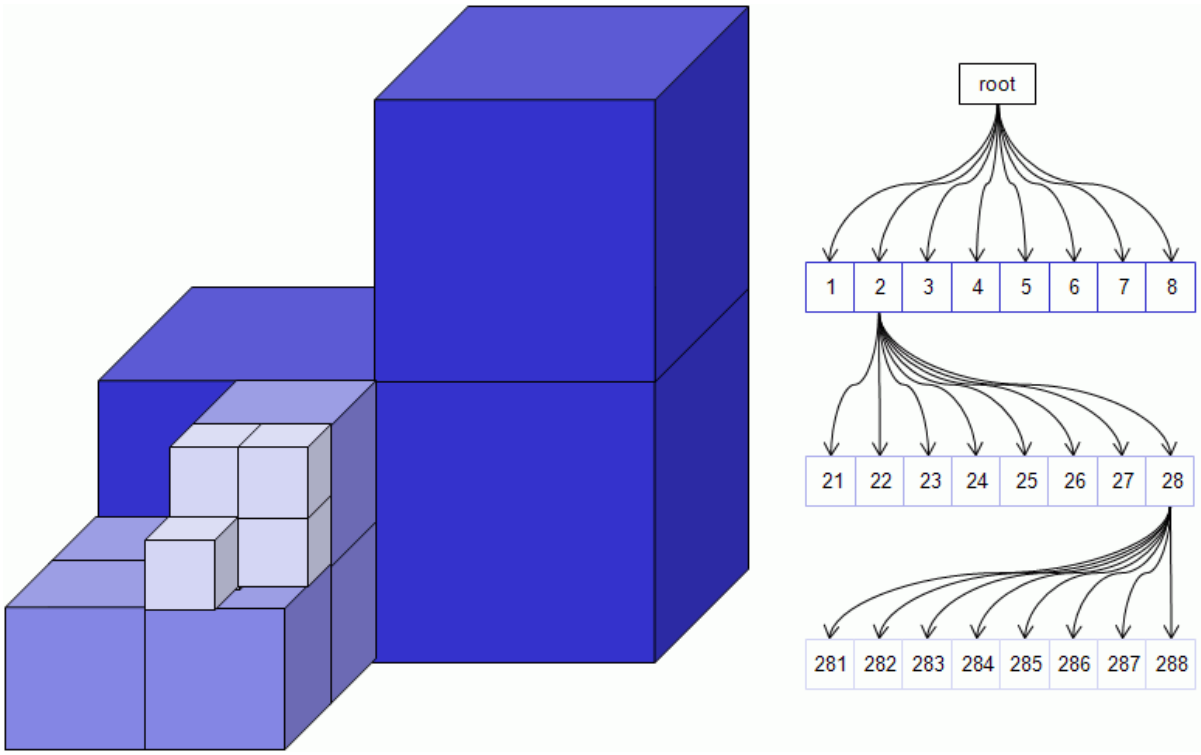

Fig. 4.29 – 3D octree as a hierarchical data structure, used to represent volumes in 3D space. On the right: voxel octal decomposition. On the left: octree data structure as a tree structure with allocation of voxel identifiers.

- inside: if all voxel footprints project inside the image silhouettes;
- outside: if at least one voxel footprint projects outside an image silhouette;
- ambiguous: neither of the above.

The first model M_0 is obtained by estimating the visual hull of the initial set of voxels, D_0. At each resolution level i, with $i \geq 1$, a new set of voxels D_i is obtained by octree decomposition, containing $q_i \leq 8 \times q_{i-1}$ voxels, with q_{i-1} the number of voxels in model M_{i-1}. Only voxels classified as ambiguous are decomposed. The number of iterations L is defined by the user according to the desired resolution and precision of the final reconstructed model. A model M_i is obtained by estimating the visual hull of D_i, consisting on the union of all voxels classified as inside or ambiguous.

As stated before, the voxels are parallelepipeds; therefore, each decomposition produces eight equally shaped sub-parallelepipeds. Each node (voxel) in the octree stores not only an identifier, but also the X-, Y-, Z- sizes, the 3D coordinates of its first vertex and its classification (inside, outside or ambiguous).

Table 4.3 – Outline of the developed octree-based volumetric reconstruction using silhouettes.

Developed octree-based reconstruction algorithm
Initialize the octree's root node as a cube that completely encloses the object to be reconstructed;
Subdivide octree into 8 sub-voxels and classify them as inside;
For level $i = 1$ to L do:
For all voxels in level i, v_i
For all images I_k
Project the 8 vertices of v_i into the k^{th} image plane, using P_k;
Determine the v_i footprint;
If v_i footprint lies completely outside silhouette S_k
Classify v_i as outside;
Break the inner for-loop;
Else if v_i footprint lies partially inside silhouette S_k
Classify v_i as ambiguous;
Subdivide all voxels v_i classified as ambiguous;
The visual hull is the union of all inside and ambiguous voxels.

Data representation of the octree is a tree structure, left on Fig. 4.29. All searches on the octree data structure are recursive and stored in a level-list for faster access. An octree can be searched for all voxels in a particular level, for all ambiguous voxels or for all non-outside voxels (ambiguous or inside) of the octree.

The overall algorithm developed for visual computation using an octree data structure and silhouette-based reconstruction is summarized in Table 4.3.

c) Initial bounding volume estimation

As previously stated, the visual hull computation with an octree requires a 3D bounding volume as a root node, delimitating a parallelepiped containing the object of interest.

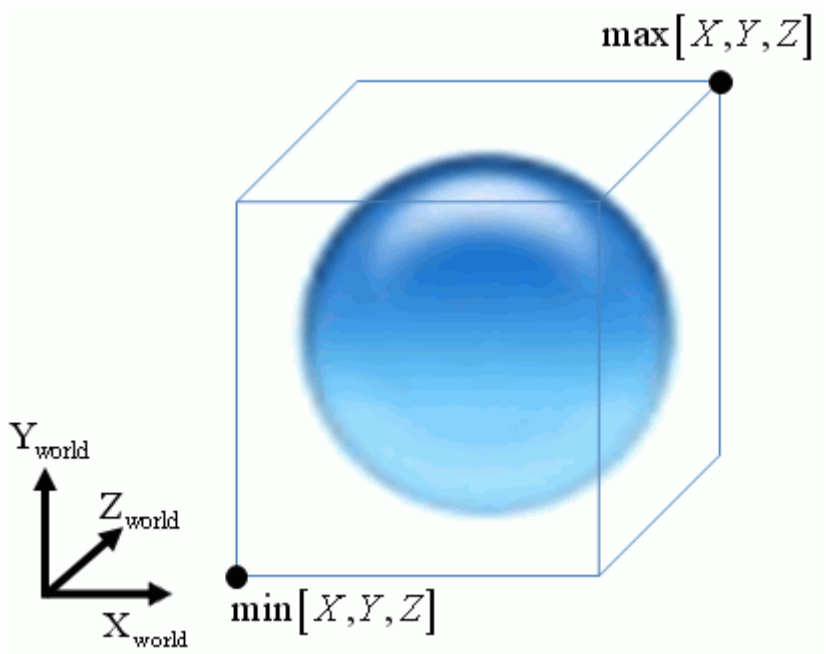

Fig. 4.30 – Initial bounding volume definition as an AABB, provided as the root node for the developed octree-based visual hull computation.

In this Thesis, the initial bounding parallelepiped is aligned with the 3D world axis coordinates, which makes it an AABB (Axis Aligned Bounding Box), a volume definition known in Computer Graphics. It is defined as the parallelepiped having the minimum volume that still includes the object being reconstructed.

Since the main goal of the developed 3D reconstruction methods was for them to be as automated as possible, it was important to automatically determine the position and size of the initial AABB.

A minimum bounding volume can be computed by optimization methods for each of the six variables defining the AABB:

$$AABB = \left\{ (X,Y,X) : \begin{array}{r} \min_X \le X \le \max_X \\ \min_Y \le Y \le \max_Y \\ \min_Z \le Z \le \max_Z \end{array} \right\}, \tag{4.20}$$

where $\min_X$, $\max_X$, $\min_Y$, $\max_Y$, $\min_Z$ and $\max_Z$ are the minimum and maximum 3D world coordinates of the AABB, Fig. 4.30. Given a set of image silhouettes, S_i, and the projection matrices, P_i, $4i$ inequations are defined:

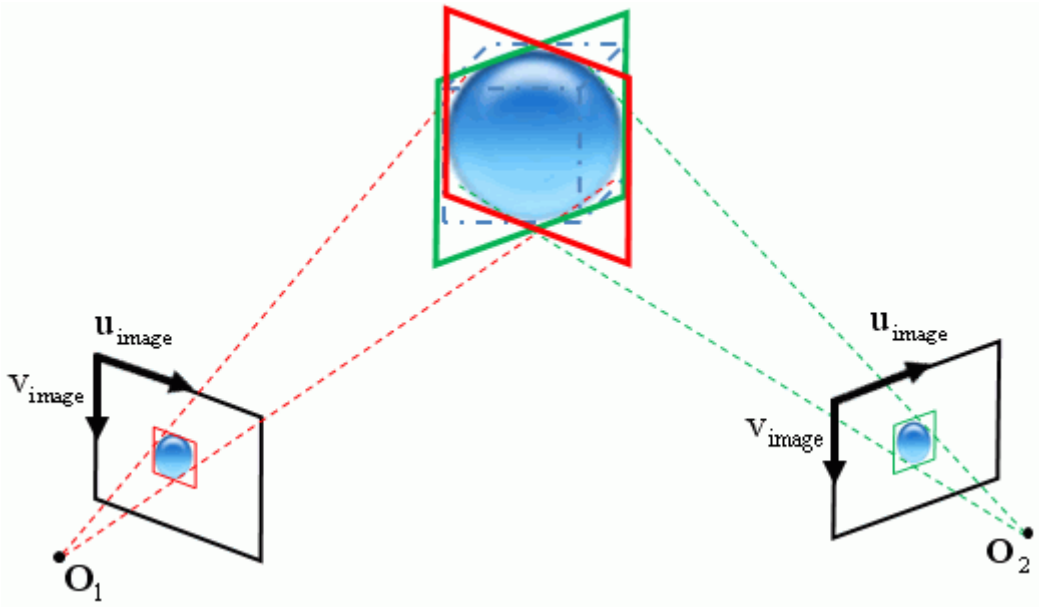

Fig. 4.31 – Back-projection of two silhouette bounding rectangles, from 2D image coordinates into the 3D world coordinates (red and green polygons). The back-projected box defines the boundaries of the object AABB (blue doted cube).

$$\begin{cases} u^i_{\min} \le \dfrac{\mathrm{p}^i_{11}X + \mathrm{p}^i_{12}Y + \mathrm{p}^i_{13}Z + \mathrm{p}^i_{14}}{\mathrm{p}^i_{31}X + \mathrm{p}^i_{32}Y + \mathrm{p}^i_{33}Z + \mathrm{p}^i_{34}} \le u^i_{\max} \\ v^i_{\min} \le \dfrac{\mathrm{p}^i_{21}X + \mathrm{p}^i_{22}Y + \mathrm{p}^i_{23}Z + \mathrm{p}^i_{24}}{\mathrm{p}^i_{31}X + \mathrm{p}^i_{32}Y + \mathrm{p}^i_{33}Z + \mathrm{p}^i_{34}} \le v^i_{\max} \end{cases}, \tag{4.21}$$

with $u^i_{\min}$, $u^i_{\max}$, $v^i_{\min}$ and $v^i_{\max}$ as the 2D pixel coordinates of the silhouette bounding rectangle, Fig. 4.31, and p^i_{mn} the matrix elements of projection matrix P_i.

Hence, the automatic computation of a minimum axis aligned bounding volume can be generalized as how to calculate the maximum and minimum (X,Y,Z) world coordinates under $4i$ constraints. Currently proposed solutions involve iterative methods, such as the Levenberg-Marquardt optimizer or genetic algorithms, as proposed in [Thormählen, 2008] and [Song, 2009], respectively. Other solutions determine the bounding boxes through rotations about three vanishing points defined by the scene [Rudek, 2005], by exploiting the central axis if the camera performs a circular motion around the object [Yilmaz, 2002], or even estimating the base edges from a single image of objects placed on the world ground plane [Jung, 2009].

Since in the developed reconstruction method, the object is placed directly on top of the calibration pattern, Equation (4.21) is relaxed because $\max_Z$ is known to be equal to 0

(zero). Therefore, the projections P_i are reduced from 3×4 matrices into 3×3 square matrices, P'_i, as proved by Equation (3.22):

$$\mathrm{P}'_i=\mathrm{K}[\,\mathbf{r}_1^i \quad \mathbf{r}_2^i \quad \mathbf{t}^i\,]. \tag{4.22}$$

Consequently, variables $\min_X$, $\max_X$, $\min_Y$ and $\max_Y$ can be derived non-iteratively using the left and right lower vertices of the silhouettes' bounding rectangles, $[u^i_{\min L}, v^i_{\min L}]$ and $[u^i_{\min R}, v^i_{\min R}]$, respectively:

$$\begin{cases} \begin{bmatrix} \min_X \\ \min_Y \end{bmatrix} = \min\left(\mathrm{inv}(\mathrm{P}'_\mathrm{i}).\begin{bmatrix} u^i_{\min L} \\ v^i_{\min L} \end{bmatrix} \right) \\ \begin{bmatrix} \max_X \\ \max_Y \end{bmatrix} = \max\left(\mathrm{inv}(\mathrm{P}'_\mathrm{i}).\begin{bmatrix} u^i_{\min R} \\ v^i_{\min R} \end{bmatrix} \right) \end{cases}. \tag{4.23}$$

With the i^{th} camera's optical centre coordinates, for the i^{th} silhouette image, one obtains:

$$CC_i = -\mathbf{t}_i \mathrm{R}_i^{-1}. \tag{4.24}$$

The variable $\max_Z$ is determined by $\min(CC^i_Z)$, with CC^i_Z the Z world coordinate of the optical centre of camera i. However, since the camera's relative motion around the object is unknown, one must verify if the projection of the computed value for $\max_Z$ does, in fact, include all image silhouettes. This can occur on higher objects, where the camera's optical centres are below a virtual 3D plane passing by the object's highest 3D position. The computed AABB must be therefore back-projected using P_i and if any silhouette is above the projected AABB, variable $\max_Z$ is increased by a factor of 10%. This process iterates until the image projection of the AABB fully encloses all silhouettes. It was found that with a factor of 10%, the process stops after not more than 7 iterations, for the 3D reconstructions obtained in this Thesis.

On the other hand, unnecessary octree refinements should also be avoided in case the image acquisition is performed from a relatively high viewpoint. Like in the previous case, only the variable $\max_Z$ is changed: it is decreased by a factor of 10%, until its back-projection collides with any of the silhouettes.

Equation (4.23) and, in some cases also Equation (4.24), returns an over-approximation of the AABB, but it is faster and simpler to compute when comparing with other iterative methods. If provided, the object's real measures are introduced on the octree data structure root node, and some calculated values for the AABB determined on this step can be overruled.

d) Voxel visibility

Finally, all voxels from final model M_1 are colourized. For realistic and accurate model texture, voxels visibility must be estimated. Moreover, if the photo-consistency option is included in the 3D reconstruction, accurate voxel footprints can only be correctly determined if the visibility of a certain voxel, regarding its position in the octree and the image plane, is available.

A number of methods have been proposed to solve the visibility problem for volumetric reconstruction methods. The fundamental problem is that of discerning which voxels of the octree are occluded by others in the view of a particular camera. This typically requires a search over the all voxels of the 3D reconstructed model. It is usually solved by making multiple plane-sweep passes, using each time only the cameras lying in front of the plane, and iterating until convergence.

Voxel visibility consists on computing the views from which each voxel is visible, and it is achieved by taking into account the occlusions with the other voxels, and the distance in 3D space between the considered voxel' and the camera' centres. A voxel colour is the RGB average of all its visible projections.

In order to determine if a voxel is visible from a certain camera's viewpoint, first, a unique ID is assigned to each voxel on the octree. Then visibility maps are constructed for all images used on the reconstruction process. A visibility map contains a voxel ID for every pixel in the corresponding image. Voxel visibility starts by building a visibility map structure with invalid voxel IDs: for example, a matrix filled with zeros. Then, the Euclidean distance is stored for every pair voxel/image. Closer voxels are first rendered into the visibility map, and valid voxel IDs are never over-written. Thus, after a set of voxels have been rendered, some pixels will contain the ID of the closest voxel that projects onto it.

Table 4.4 – Outline of the developed voxel visibility algorithm.

Voxel Visibility
Search for all voxels v_i in the Visual Hull classified as ambiguous or inside;
For all images I_k do:
Sort all v_i by its distance to the image optical centre
Initialize visibility map VIS_k with zeros;
For all v_i do:
Compute voxel footprint on I_k ;
Update VIS_k using v_i ID;
Add RGB pixel information into v_i colour;
Increase v_i number of pixels where it is visible;
For all v_i do:
Compute v_i colour average.

Once valid visibility maps have been computed for the images, it is then possible to compute the set of all pixels from which the voxel is visible: for each pixel in the visibility map, if the value equals the voxel's ID, then the pixel is added to the set of pixels from which the voxel projects.

The voxel visibility algorithm developed and implemented in this Thesis is outlined in Table 4.4.

e) Photo-consistency estimation

If a photo-consistency option is included, all voxels from final model M_l are tested using the footprints from images where they are visible. A voxel is said to be photo-consistent if its footprints have the same or similar colour.

Since this is the most demanding algorithm, in terms of computational time, it was decided to incorporate it as a last refinement step of the previously computed visual hull. Thus, unnecessary voxel visibility determination steps and hierarchical thresholds for the consistency estimation were prevented.

The original consistency check for voxel colouring was introduced in [Seitz, 1997]. In line with the assumption adopted in this Thesis, they assumed approximately Lambertian objects, i.e., objects containing only matte surfaces. As such, a set of voxels S is photo-consistent with images $I_1,...,I_N$ if, for every voxel $V \in S$ and image pixels $\mathbf{p} \in I_i$ and $\mathbf{q} \in I_j$:

$$V = S(\mathbf{p}) = S(\mathbf{q}) \Rightarrow colour(\mathbf{p}, I_i) = colour(\mathbf{q}, I_j) = colour(V, S)\,. \tag{4.24}$$

This means that, if a voxel V is not fully occluded in image I_i, its projection will overlap a nonempty set of image pixels, π_i. Without noise or quantization effects, a consistent voxel should project to a set of pixels with identical colour values for all images where V is visible. The main statement in [Seitz, 1997] was that to account for some image noise, and so voxel consistency is determined using the standard deviation σ_V of the colour values of all the pixels a voxel projects to:

$$\sigma_{V,colour} = \sigma\left(\bigcup_{i=1}^{k} \pi_i\right) = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(c_i - \bar{c})^2}\,, \tag{4.25}$$

with k the number of images from where the voxel is visible, c_i the pixel value for one of the RGB image colour channels and $\bar{c}$ the average colour for one of the RGB channel for all images. A pixel is said to be photo-consistent if:

$$\sigma_{V,c} < T \text{ and } k > 0\,, \tag{4.26}$$

for all colour channels, with T a predefined threshold value.

A major problem with this consistency check is that there is no optimal threshold: areas with little texture are reconstructed best with a low threshold, while areas that are highly textured or with sharp edges need very high threshold values. Worse, during the visual hull computation, the visibility maps are not set correctly, and the errors are propagated along the rays from the camera's optical centres through the incorrectly classified voxel. If the threshold is set to low, some voxels are wrongly removed in areas with high variation or belonging to an edge; on the other hand, if the threshold is set high, small colour variations are ignored, resulting in cusping in the areas with low variation – cusping is a distortion in which a surface in the reconstruction is warped toward the cameras, relative to the actual surface.

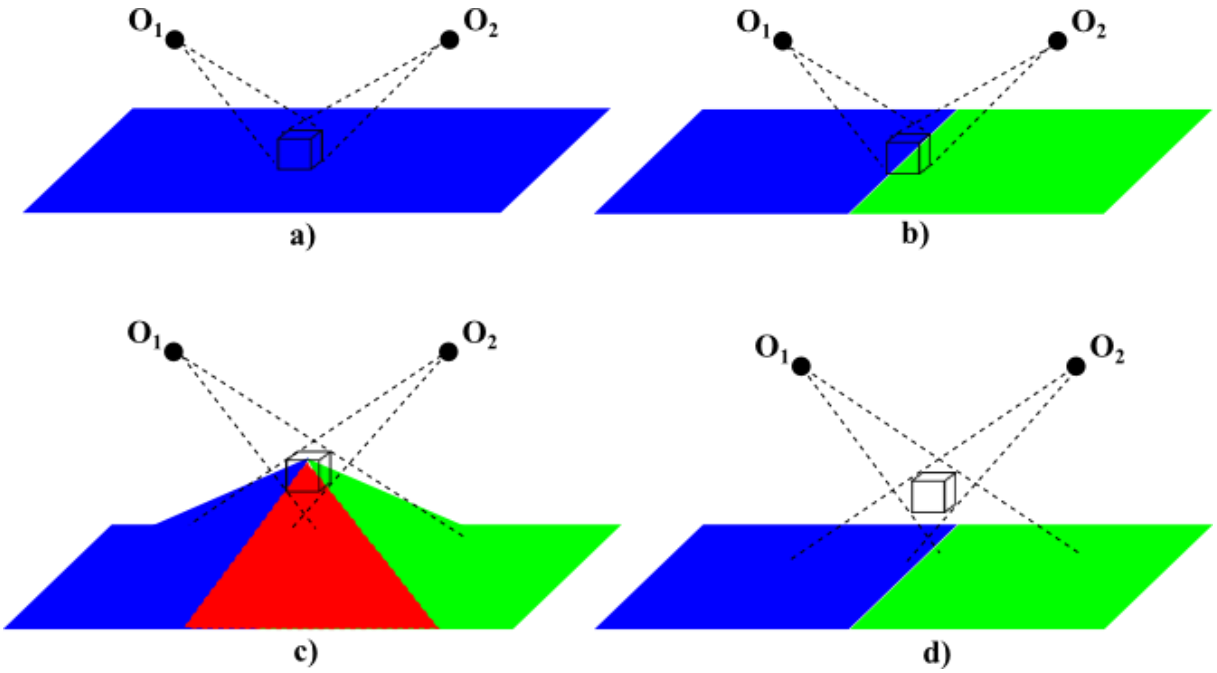

Fig. 4.32 – Voxel photo-consistency check: a) the voxel represents a homogeneous region, for which both σ and $\bar{\sigma}$ are small; on a textured region, b), or on an edge, c), both σ and $\bar{\sigma}$ are high; d) when the voxel is outside the object, σ is high but $\bar{\sigma}$ is small.

The solution is to use an adaptive colour threshold as introduced in [Slabaugh, 2002]. This adaptive consistency check calculates the standard deviation over all pixels in the individual images where a voxel projects. These standard deviations are averaged, resulting in the mean standard deviation per image, $\bar{\sigma}_V$. The overall deviation, σ_V, is now determined taking into account $\bar{\sigma}_V$:

$$\sigma_V < T_1 + \bar{\sigma}_V T_2, \tag{4.27}$$

with T_1 and T_2 being predefined threshold values. If there are very high standard deviations in all images, which may be caused by an edge or high texture variations, then the overall threshold is increased, preventing false voxel removals, Fig. 4.32.

In [Slabaugh, 2002], all RGB colour channels are combined into a single standard deviation:

$$\sigma_V = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(R_i^2 + G_i^2 + B_i^2) - \bar{R}^2 - \bar{G}^2 - \bar{B}^2}\ . \tag{4.28}$$

Table 4.5 – Outline of the developed voxel consistency check algorithm.

Voxel Consistency
Set all leaf voxels V in the Visual Hull as consistent;
For N iterations do:
NoChanges = TRUE
For every voxel V do:
Find the set S of images where V is visible;
If S>1 and V as a consistent colour
Assign V the average colour of all pixels in S;
Else
NoChanges = FALSE;
Remove V from Visual Hull;
If NoChanges = TRUE
Quit loop.

However, for accuracy purposes, in this Thesis a decision was made to use the original definition of standard deviation and to keep the analysis of the colour channels separated. Thus, a voxel V is said to be photo-consistent if:

$$\sigma_{V,colour} < T_1 + \overline{\sigma}_{V,colour} T_2, \tag{4.29}$$

where *colour* is one of the considered colour channels. This criterion compares the photo-consistency individually for all three colour channels, which in turn provides for a more accurate decision about the voxel classification as inside or outside the reconstructed volume.

To optimize the photo-consistency testing, a minimum allowable percentage of image pixels in the voxel's footprint was ensured. To perform that, at least 20% of the non-occluded image pixels projection would have to correspond to the complete footprint of a voxel.

In addition, a voxel is tested for photo-consistency if the number of images from where it is visible is higher than 1 (one). Both conditions not only reduce the noise from calibration data but also allow some tolerance to erroneous segmentation.

The voxel consistency algorithm developed and implemented in this Thesis is summarized in Table 4.5. When the visibility maps computation is finished, the voxel's photo-consistency is calculated using Equation (4.29). If it is found to be inconsistent, it is set to outside, which in turn may change underlying voxels' visibility status. Thus, the visibility

map computation and photo-consistency calculation of the entire 3D model only stops when all those voxels that remain are projected into consistent colours in the images from which they are visible.

4.3.5 3D model assessment

To evaluate the developed and implemented volumetric reconstruction methodology, some analysis on the obtained 3D models was performed. Comparison of some geometrical measures between real objects and the reconstructed 3D models are the first analysis to be performed when assessing the validity of a reconstruction method. However, an extensive evaluation must be carried out for a complete validation of the reconstruction accuracy and precision.

Following sections describe in detail the analysis performed on the obtained 3D models.

a) Geometrical measures

From the voxelized 3D models obtained, some geometrical measures can be determined, such as:

- height, length and width;
- volume and surface;
- specific measures using mesh processing software (e.g., [Meshlab, 2012], [Paraview, 2012], [MiniMagics, 2012], among others).

In the algorithm developed for surface extraction all 2D faces that are used by only one 3D voxel, i.e. boundary faces, are extracted. Therefore, it is necessary to previously decompose all octree's voxels in order for them to have the same (smallest) size.

b) Reprojection error

The reprojection error is another useful measure for evaluating the results of reconstructions, as well as a measure of colour consistency. In order to perform an objective evaluation of the reprojection error, the obtained 3D model is projected (rendered) on images

that are not used in the reconstruction process. Further on, these images will be called evaluation images. Rendering is performed using the camera information that is obtained from the calibration step.

A common distance, between evaluation and rendered images, is the square of differences per RGB colour component:

$$\forall \alpha \in \{R,G,B\}, E_\alpha = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(Ie_i - Ir_i)^2}\,, \tag{4.30}$$

with Ie as the evaluation images, Ir the rendered images and N the number of pixels on both images. Since $E \in [0,255]$, a normalized criterion can be computed as the percentage of colour similarity between evaluation and rendered images:

$$p_E = 100\left(1 - \frac{avg(E_\alpha)}{255}\right). \tag{4.31}$$

c) Hausdorff distance

Another way to assess a reconstructed 3D model is to compare it with another 3D model of the same object obtained using another reconstruction method.

Having two discrete 3D surfaces of the same object, a useful tool to measure distances between the two surfaces is the Hausdorff distance [Aspert, 2002]. This distance, or metric, is a mathematical construct to measure how far two meshes are from each other. To understand this metric, let's define the distance $d(\mathbf{p}, S')$ between a point $\mathbf{p}$ belonging to a surface S and a surface S' as:

$$d(\mathbf{p}, S') = \min_{\mathbf{p}' \in S'} \|\mathbf{p} - \mathbf{p}'\|. \tag{4.32}$$

From Equation (4.32), the one-sided Hausdorff distance from surface S to S' is given as:

$$d(S, S') = \max_{\mathbf{p} \in S} d(\mathbf{p}, S'). \tag{4.33}$$

Since $d(S,S') \neq d(S',S)$, Fig. 4.33, the Hausdorff distance is computed as:

$$d_{s,s'} = \max[d(S,S'), d(S',S)]. \tag{4.34}$$

The point-to-surface distance defined in Equation (4.32) can be used to define a mean error, d_m:

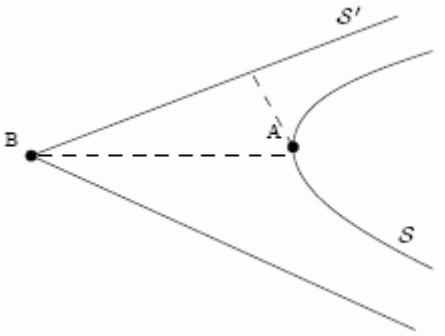

Fig. 4.33 – The one-sided Hausdorff distance from surface S to S', $d(S,S')$, is considerable smaller than $d(S',S)$, since in this case $d(\mathbf{A},S') << d(\mathbf{B},S)$. Thus, a small one-sided distance does not imply a small distortion (from [Aspert, 2002]).

$$d_m(S,S') = \frac{1}{|S|}\iint_{\mathbf{p}\in S} d(\mathbf{p},S')dS\,, \tag{4.35}$$

where $|S|$ is the area of S. From this, the root mean square d_{rmse} follows as:

$$d_{rmse}(S,S') = \sqrt{\frac{1}{|S|}\iint_{\mathbf{p}\in S} d(\mathbf{p},S')^2 dS}\,. \tag{4.36}$$

d) Polygonal smoothing

The generated 3D model from the volumetric method developed is composed by voxels. Therefore, its surface patches and edges are always aligned with the world coordinate system, which produces a jagged surface. A straightforward approach to overcome this is to smooth the polygonized surface. The simplest smoothing algorithm that satisfies the linear complexity, required for large datasets, is the Laplacian smoothing [Taubin, 2000].

The Laplacian smoothing proceeds as follows. For each mesh vertex, its coordinates are moved according to a weighted average of its connected vertices. A relaxation factor is available to control the amount of displacement of the vertex. When the process is repeated for each vertex, a single iteration is completed.

There are some variables used to control the execution of the Laplacian smoothing algorithm. The convergence variable limits the maximum motion of any vertex point. It is expressed as a fraction of the length of the diagonal of the bounding box of the mesh. If the maximum point motion during a smoothing iteration is less than the convergence value, the

smoothing operation terminates. The number of iterations is the maximum number of smoothing passes. More iterations produce higher smoothing.

A problem with Laplacian smoothing is shrinkage. When a large number of Laplacian smoothing steps are iteratively performed, the shape undergoes substantial deformations, eventually converging to the centroid of the original data.

4.4 Summary

During this Thesis, two image-based methods for 3D object reconstruction were developed and implemented: Stereo Reconstruction and Volumetric Reconstruction. The main goal was to obtain the 3D shape of human external surfaces, without imposing severe restrictions on the relative motion between the used camera and the object to be reconstructed. In both the reconstruction of a rigid object with Lambertian surfaces was assumed.

The first 3D reconstruction method was based on stereoscopic vision. It requires two perspective views of the object to reconstruct, usually named stereo pair. So, starting with two images of an object, the epipolar geometry, which is denoted by the fundamental matrix, is extracted and finally, a disparity map of the object is obtained.

For the second 3D reconstruction method, two sets of images are required. Combined sets allow for a more accurate camera calibration and pose estimation when the object to be reconstructed partially occludes the calibration pattern. Developed calibration pattern is an adaptation of the usual planar black and white chessboard. The modifications introduced on the pattern allowed for a correct orientation, when the camera's or object's relative motion is not known a priori.

Then, using the acquired images, obtained silhouettes and calculated camera calibration data, a 3D multi-resolution model is reconstructed and colorized. A 3D octree data structure was included in order to achieve faster computational times.

The typical square shape of a 3D voxel is relaxed, which avoids unnecessary carving steps and adapts the initial bounding box to the real 3D shape of the object. This initial bounding volume is automatically determined from correspondent silhouettes and projection matrices; instead, the voxel footprint is not determined over the entire silhouette. The voxel's eight vertices are back-project onto the image plane and its boundary rectangle imposes the minimum and maximum values for the pixel comparisons against the silhouette information.

Colour consistency may be applied to refine the obtained models for objects with significant texture information. It is not only determined by statistical photogrammetric analysis, but it also takes into account the voxel visibility on each individual image.

Finally, subjective and analytical characteristics can be obtained to analyze the resultant 3D model, to assess the reconstruction accuracy if a ground-truth is available or even to compare results with other methods.

5

Experimental results and Discussion

5.1 Introduction

This Chapter presents the results of the developed 3D reconstruction methods as their discussion. As stated in previous Chapter, the work developed in this Thesis followed two main methodologies: Stereo and Volumetric based reconstruction methodologies.

Therefore, the first group of results presented were obtained using the stereo-based reconstruction methodology, described in Section 4.2. Then, follows the results obtained using the volumetric-based reconstruction methodology, described in Section 4.3.

For all the presented experimental results, the same digital camera was used: a CANON EOS 400D, with an image sensor of 22.2×14.8 mm^2, approximately 10.1M effective pixels and a maximum resolution of 3888×2592 pixels. The used lens was a CANON EF-S, allowing for focal distances ranging from 18 to 55 mm.

The developed 3D reconstruction methods were ran on a 32-bit desktop computer with an Intel Pentium 3GHz processor, 2GB of RAM and a Nvidia GeForce 7800 GT graphics card.

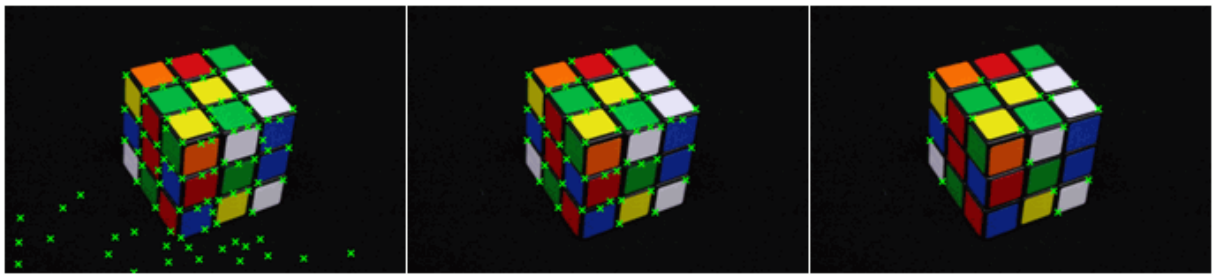

Fig. 5.1 – Green crosses represent the feature points found using the Harris detector.
From left to right, the quality threshold was increased by a factor of 10.

Fig. 5.2 – Green crosses represent the feature points found using the Harris detector.
From left to right, the minimal distance was equal 1, 50 and 100, respectively.

5.2 Stereo-based reconstruction results

To test the stereo-based reconstruction methodology, two images of the same object were used. Both were acquired on a simple black background, and with a resolution of 774×515 pixels.

5.2.1 Rubik cube

First tests were concerning a Rubik cube. This object has a straightforward topology, with flat orthogonal surfaces, whose vertices can be easily detected in each image and simply matched in the stereo image pair.

Feature points extraction

Fig. 5.1 and Fig. 5.2 present some results obtained using the Harris detector (Section 4.2.1). On the first figure, the influence of the accepted quality for the image features was compared. The quality was measured as the threshold for the minimal eigenvalue of the derivative covariation matrix for every pixel (Equation (4.1)). A careful decision on the quality threshold must be performed, since a very low threshold allows for false-positive

features and, on the other hand, a high threshold discards features on image areas with low contrast, Fig. 5.1. On the other hand, in the images on Fig. 5.2, the minimal possible Euclidean distance between the returned vertices was varied. This value can be inferred directly from the acquired image, by measuring the minimal distance between potential feature points.

Feature points matching

Using the best feature point detection results on the first stereo image, next step was to find the matches on the second stereo image. For that, the LK method, described in Section 4.2.1, was used and tested, by varying the search window size and the pyramid level.

Fig. 5.3 shows the results obtained from varying the pyramid level, ranging from 1 to 3. The accuracy was improved by increasing the number of pyramids; however, for a large number (higher than 8), no matches were successfully found.

Fig. 5.4 shows different matching results obtained by varying the search window size. Again, a compromise must be made, because small windows can be the source of false matches, and large windows cause some points not to be matched.

Since the Rubik cube presents flat surfaces, with strong colour variations, obviously few wrong matches were detected.

Epipolar geometry computation

To determine the epipolar geometry for the stereo image pairs, the RANSAC algorithm, described in Section 4.2.2 and summarized on Table 4.1, was used.

RANSAC improves the previous matching results, classifying them into inliers and outliers, Fig. 5.5. This matching refinement is performed through the determination of the epipolar geometry, Fig. 5.6.

In each image, the direction of the camera associated to the other image may be inferred from the intersection of the pencil of epipolar lines. In this case, both epipoles lie outside of the visible image.

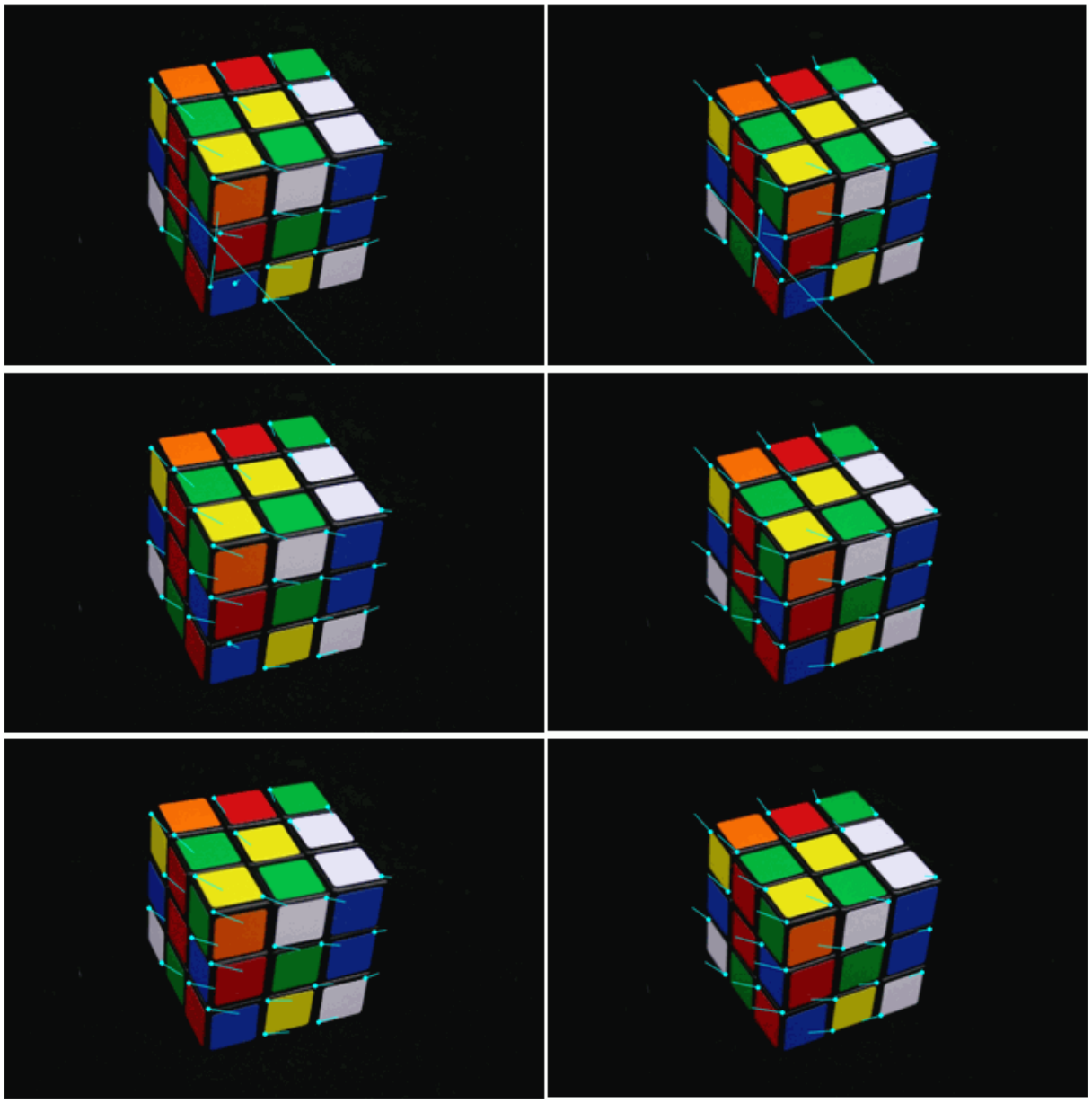

Fig. 5.3 – Matching results using the LK method. Cyan dots represent the supposed matches and cyan lines represent the connection with the other stereo image point match. From top to bottom: 1, 2 and 3 pyramid levels were adopted, respectively.

Image rectification

With the epipolar geometry computed, the image pair was rectified using the method described in Section 4.2.3. Obtained results can be observed in Fig. 5.7.

Image distortion is proportional to the slope variation of the epipolar lines. This means that close epipoles to the image centre originate highly distorted rectification images. From the results, one can observe that rectified left image is more distorted than the right one.

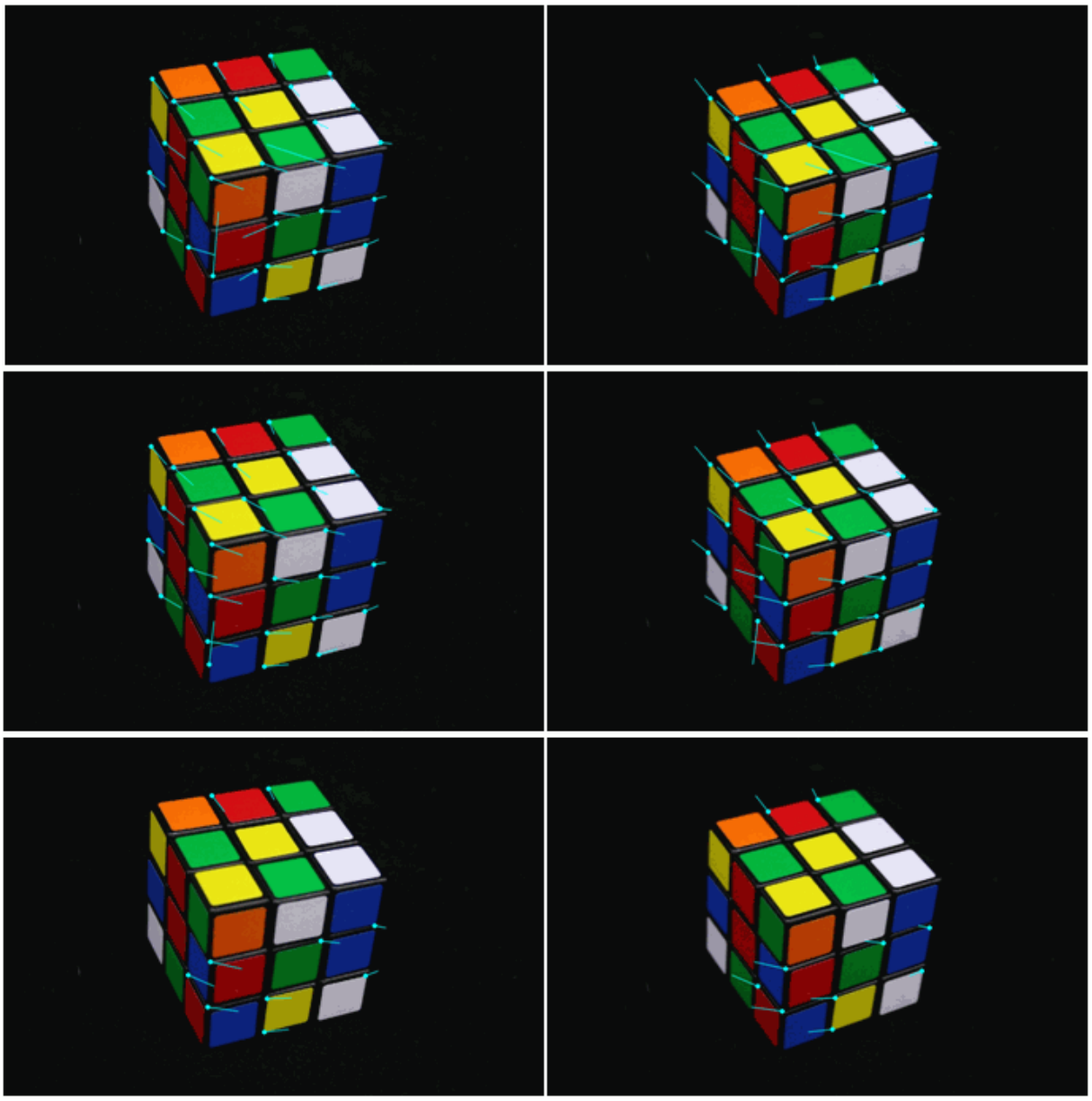

Fig. 5.4 – Matching results using the LK method. Cyan dots represent the supposed matches and cyan lines represent the connection with the other stereo image point match. From top to bottom, the window size was equal to 2, 10 and 30, respectively.

Dense matching

The dense matching resulted for the Rubik cube is shown in Fig. 5.8. This result was obtained using the method described in Section 4.2.4. Fusing the rectified images from the two viewpoints and exploiting the disparity between them, allowed a stronger sense of depth. However, since the image acquisition system was not calibrated, there is a projective ambiguity in the reconstruction achieved.

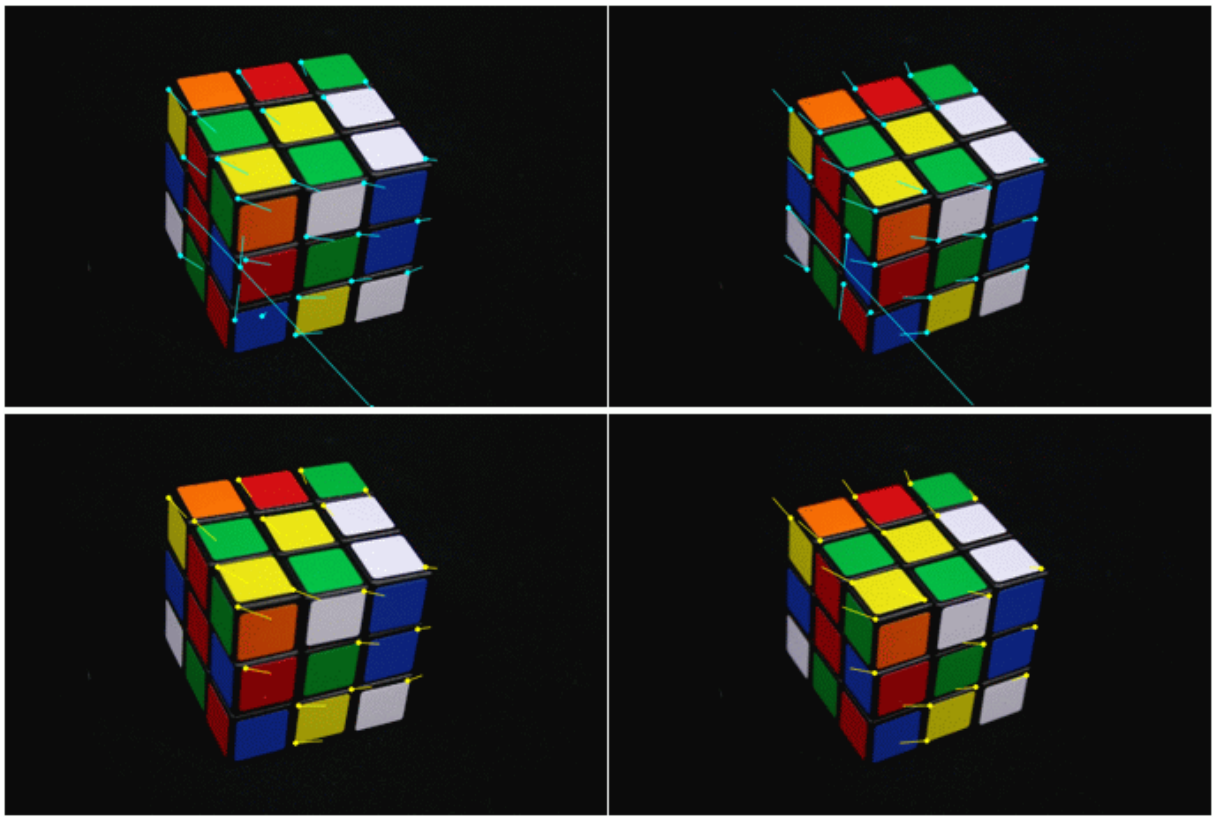

Fig. 5.5 – On the top: matching results using the LK method. On the bottom: matching results improved by using the RANSAC algorithm.

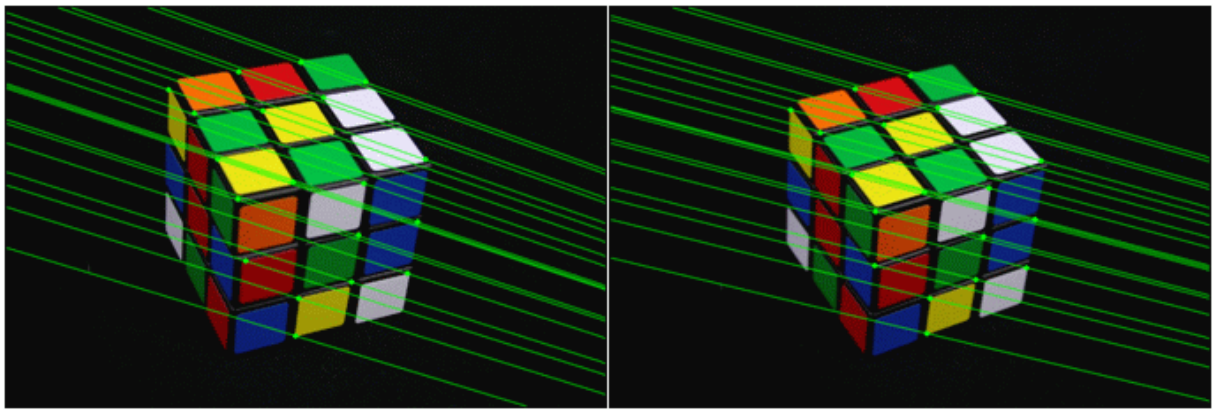

Fig. 5.6 – Epipolar lines (the green lines) for the inlier matches (the green dots).

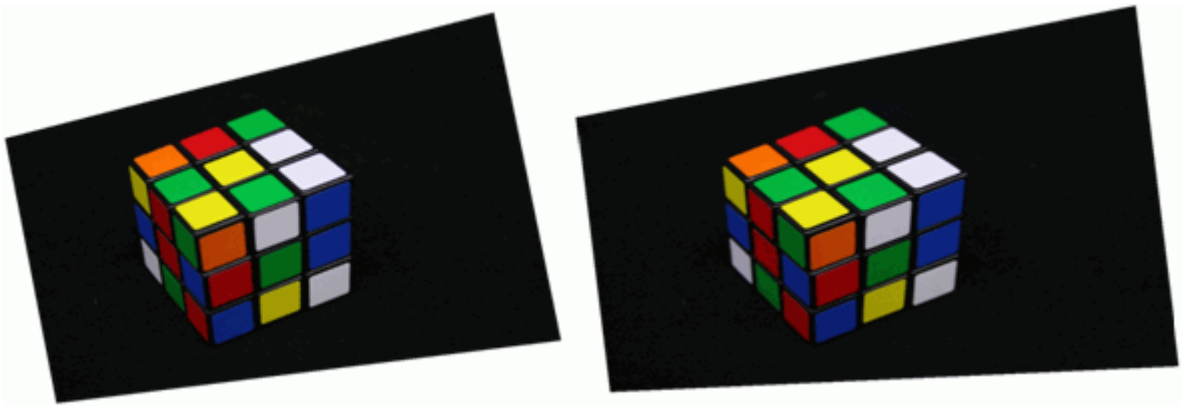

Fig. 5.7 – Rectified image pair for the Rubik cube object.

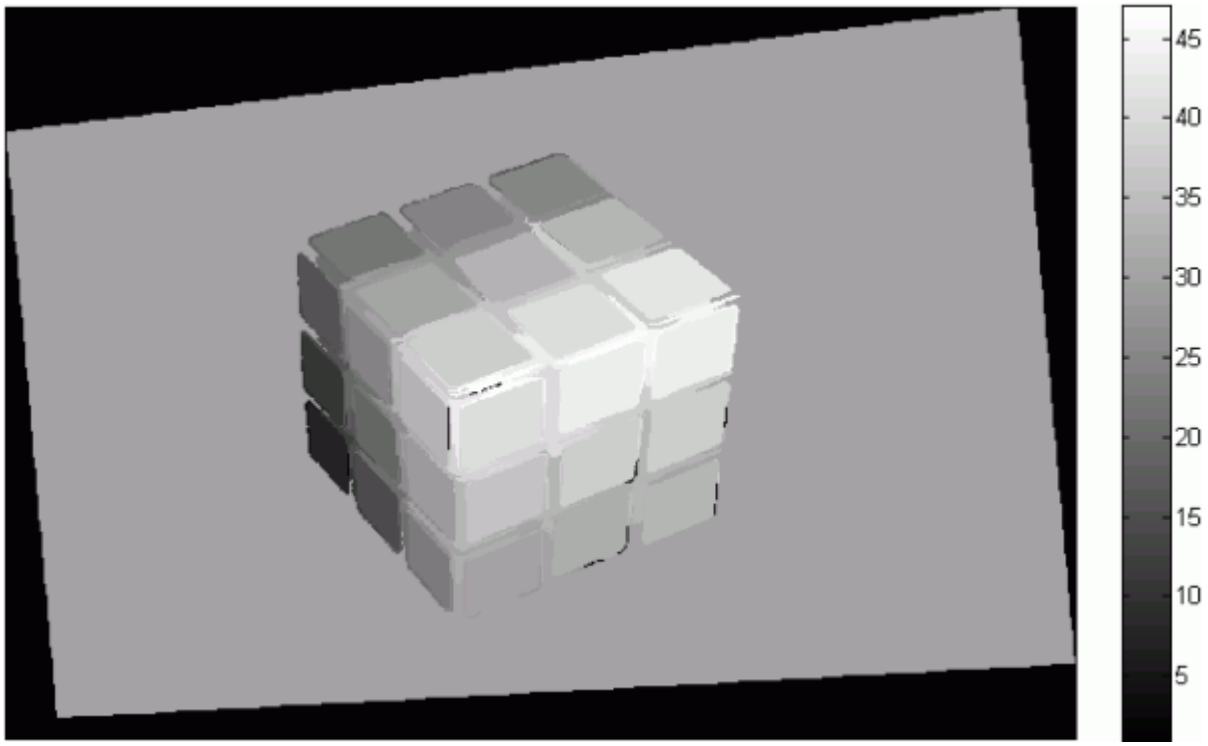

Fig. 5.8 – Disparity map for the Rubik cube object: higher (lighter) disparity values mean closer points, and lower (darker) disparity mean farther points. The image scale range displayed was adjusted to the minimum and maximum values of the disparity map, which were 1 and 49, respectively.

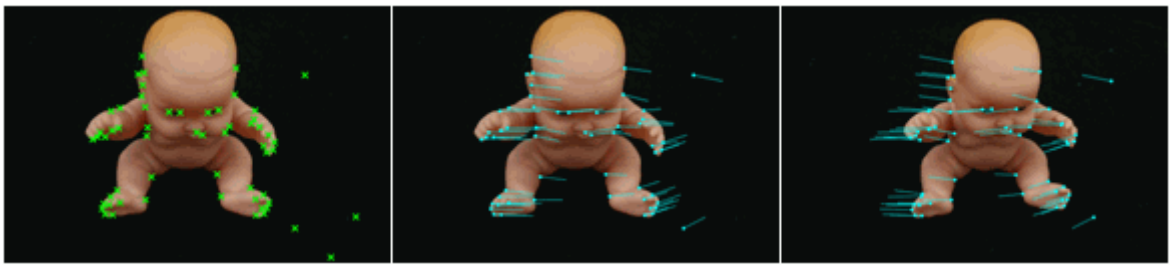

Fig. 5.9 – On the left: green crosses represent the feature points found on the left image using the Harris detector. In the middle and on the right: matching results using the LK method. Cyan dots represent the supposed matches, and cyan lines represent the connection with the other stereo image point match.

5.2.2 Plastic baby toy

Next tests were performed using a plastic baby toy that has a smooth surface and complicated shape. Unlike the Rubik cube, few strong colour or texture variations could be found.

Feature points extraction and matching

Fig. 5.9 shows the result of the feature points' extraction and matching obtained for the second test object.

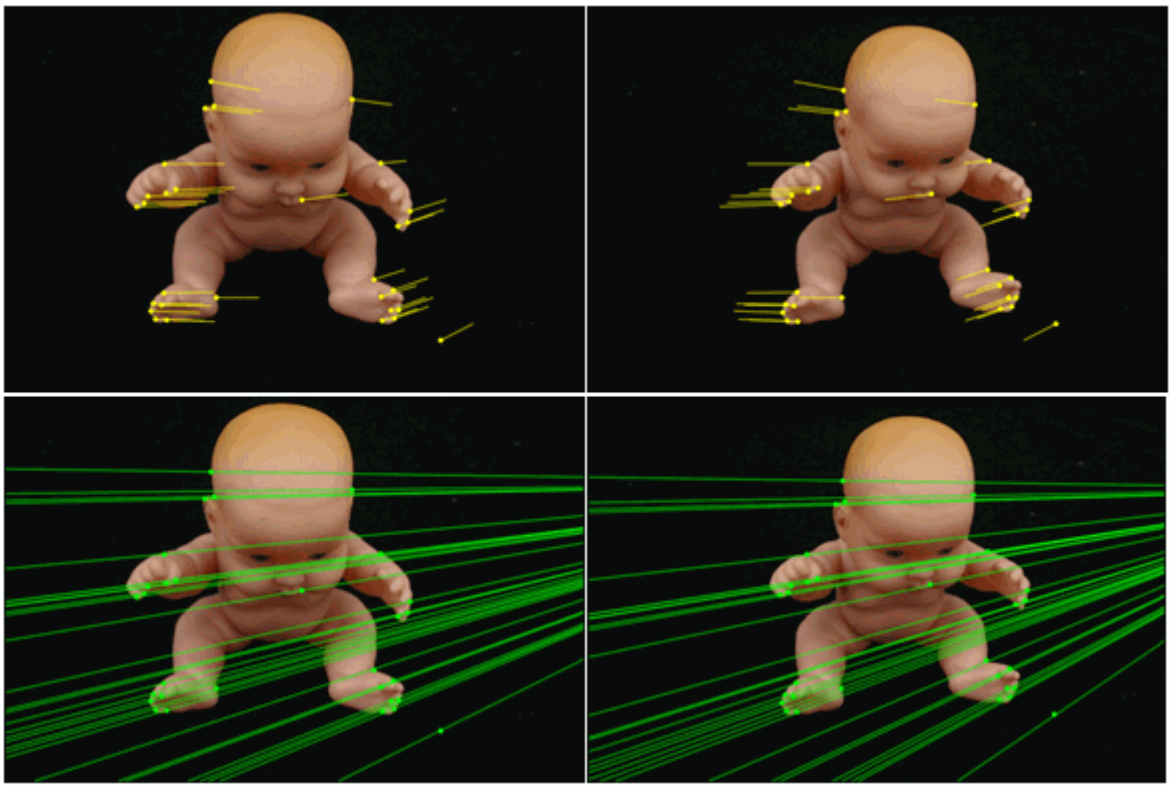

Fig. 5.10 – On the top: matching results improved by the RANSAC algorithm.
On the bottom: epipolar lines for the inlier matches.

Since the plastic baby toy presents a smooth surface, obviously many wrong matches were detected. In low textured areas, local maxima of Harris weight function R are the point located on the boundary between object and background. This induces wrong matches between the image pair.

Epipolar geometry computation

Since many wrong matches were detected, consequently, the determined epipolar geometry was incorrectly estimated. Fig. 5.10 shows the results obtained.

Image rectification and dense matching

Fig. 5.11 shows the rectification result for the baby toy object. As with the Rubik cube, distortion of the rectified images was proportional to the slope variation of the epipolar lines.

In comparison with the Rubik cube, the result of the dense matching was of worst quality, Fig. 5.12.

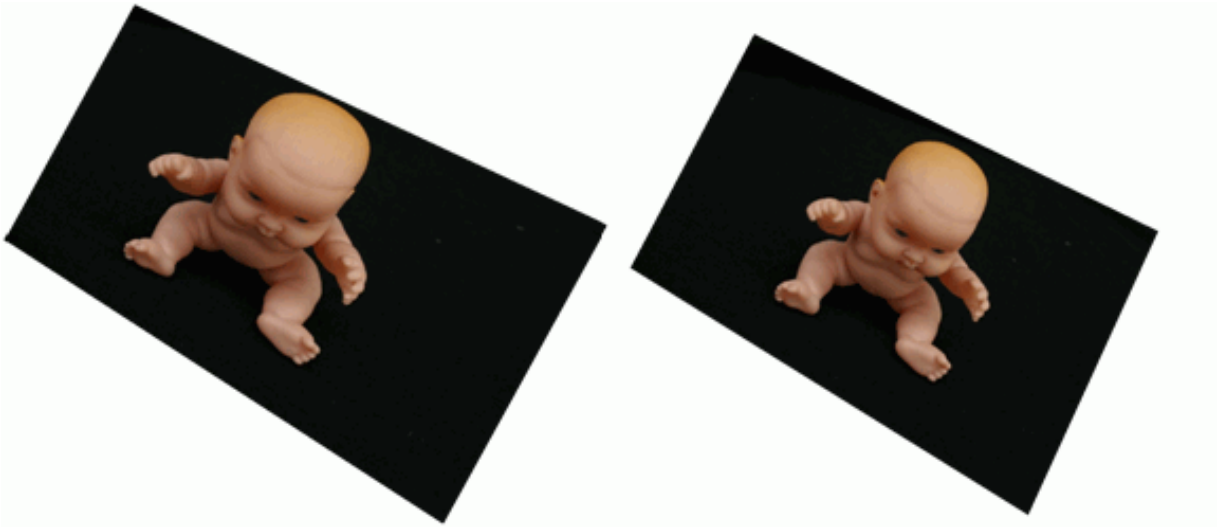

Fig. 5.11 – Rectified image pair for the plastic baby toy object.

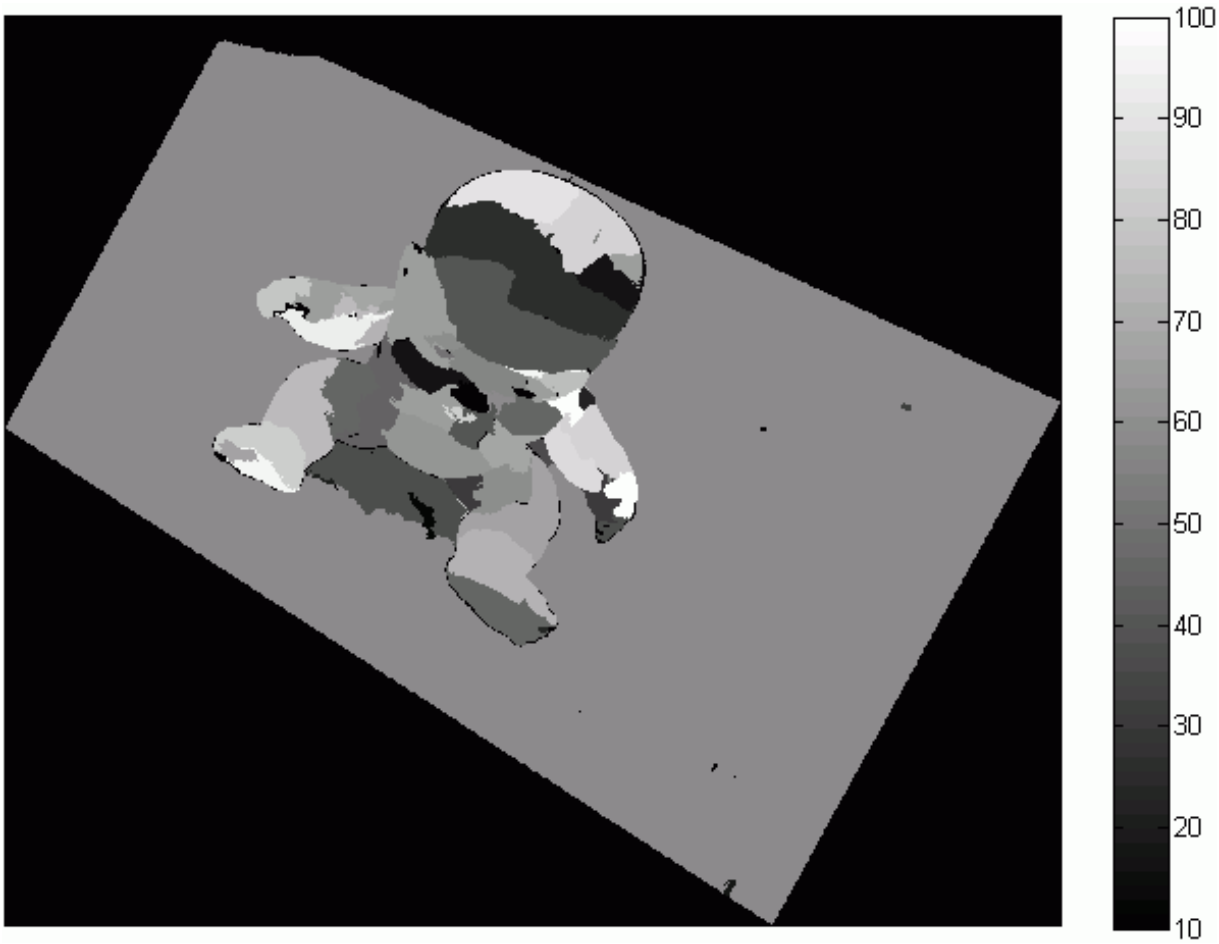

Fig. 5.12 – Disparity map obtained for the plastic baby toy object: higher (lighter) disparity values mean closer points and lower (darker) disparity mean farther points. The image scale range displayed was adjusted to the minimum and maximum values of the disparity map, which were 10 and 100, respectively.

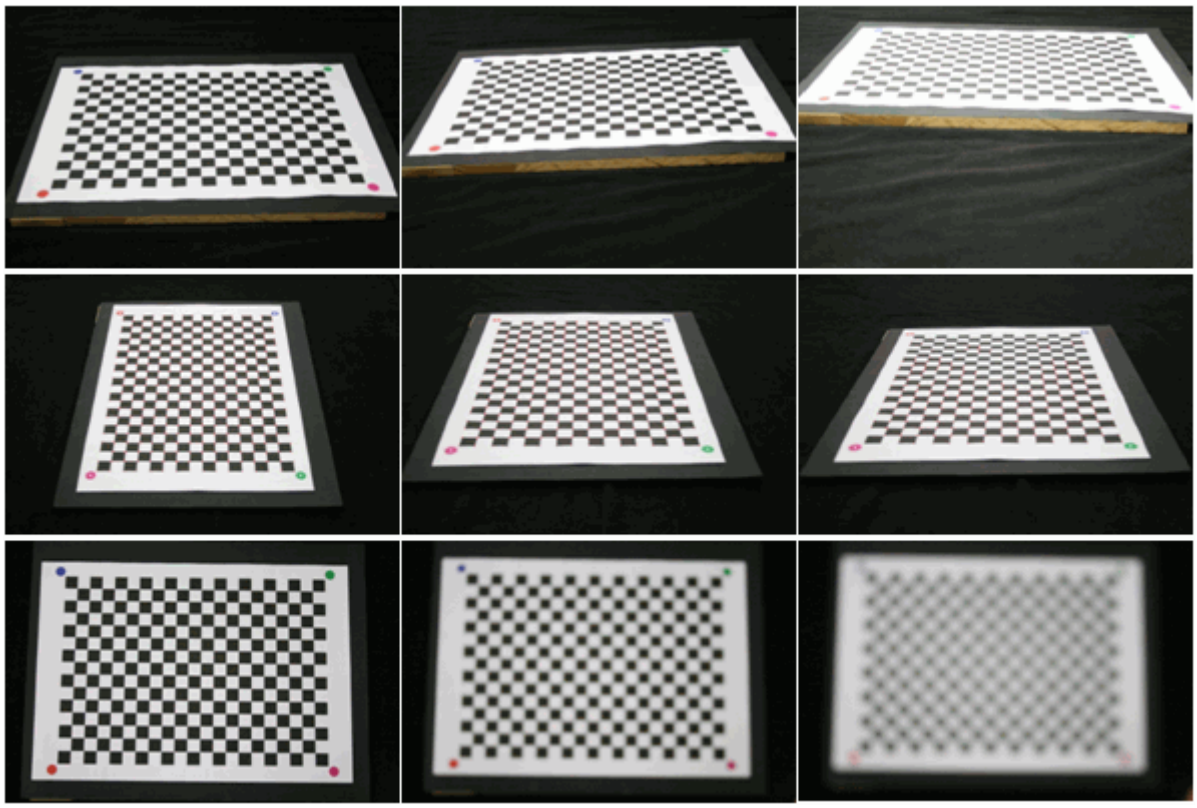

Fig. 5.13 – On the top: 3 of the 6 images obtained by approximating the camera towards the calibration pattern, while maintaining the camera's centre with a constant X coordinate. On the middle: 3 of the 6 images obtained by approximating the camera towards the calibration pattern, while maintaining the camera's centre with a constant X coordinate. On the bottom: 3 of the 4 images acquired by increasing the defocus.

5.3 Volumetric-based reconstruction results

To test the volumetric-based reconstruction methodology, two image sequences were used, as explained in Section 4.3.1.

5.3.1 Rubik cube

The first experimental test was performed using the same Rubik cube used in Section 5.2.1. Initially, 10 images of the chessboard calibration pattern were acquired. Then, some extra images were acquired for the second sequence, in order to evaluate some conditions of the image acquisition process for a successful calibration, Fig. 5.13: 6 images were obtained by decreasing the Z coordinate of the camera's centre with a constant X coordinate, 6 more were obtained by decreasing the Z coordinate of the camera's centre with a constant Y coordinate, and 4 images were acquired by decreasing the focus.

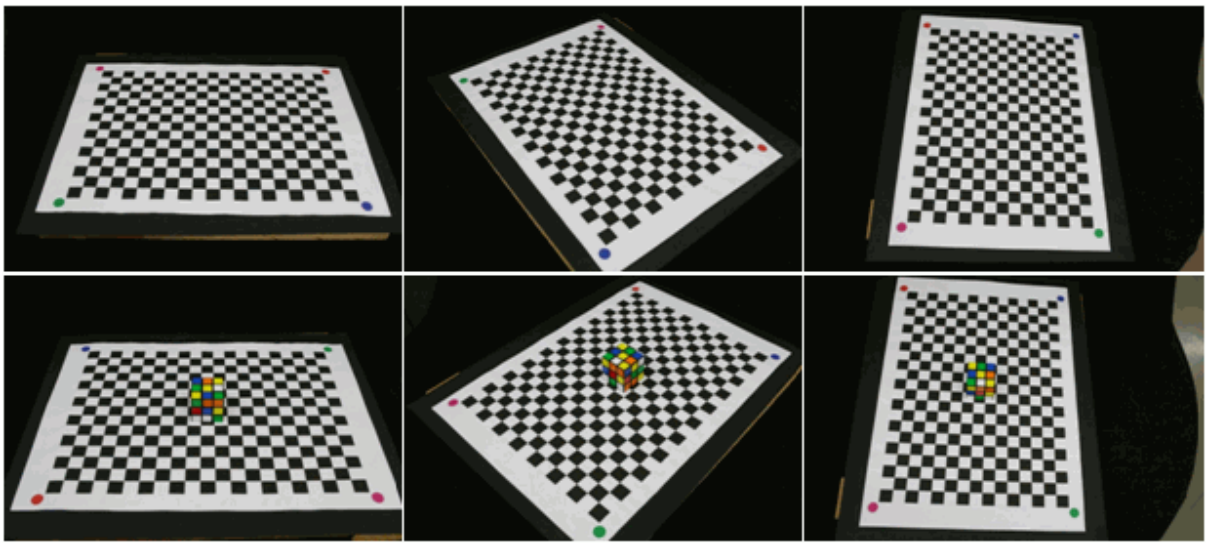

Fig. 5.14 – Top and bottom: three examples of the first and second image sequence, respectively, acquired for the Rubik cube.

Table 5.1 – Camera's intrinsic parameters obtained for the Rubik cube object.

Intrinsic Parameters		
Focal distance (pixel-related unit)	f_x	2108.124
	f_y	2093.228
Principal point (pixel)	c_x	980.476
	c_y	595.073
Radial distortion coefficients	k_1	-0.10253
	k_2	0.08663
Tangential distortion coefficients	p_1	-0.00122
	p_2	-0.00143

Finally, the Rubik cube was placed over the calibration pattern and more 11 images were acquired, Fig. 5.14.

All images were acquired on a simple black background with a resolution of 1936×1288 pixels.

Camera calibration

For 10 images of the first sequence, the 280 (20×14) chessboard vertices were successfully extracted and matched. Table 5.1 shows the camera's intrinsic parameters obtained using this first set of images.

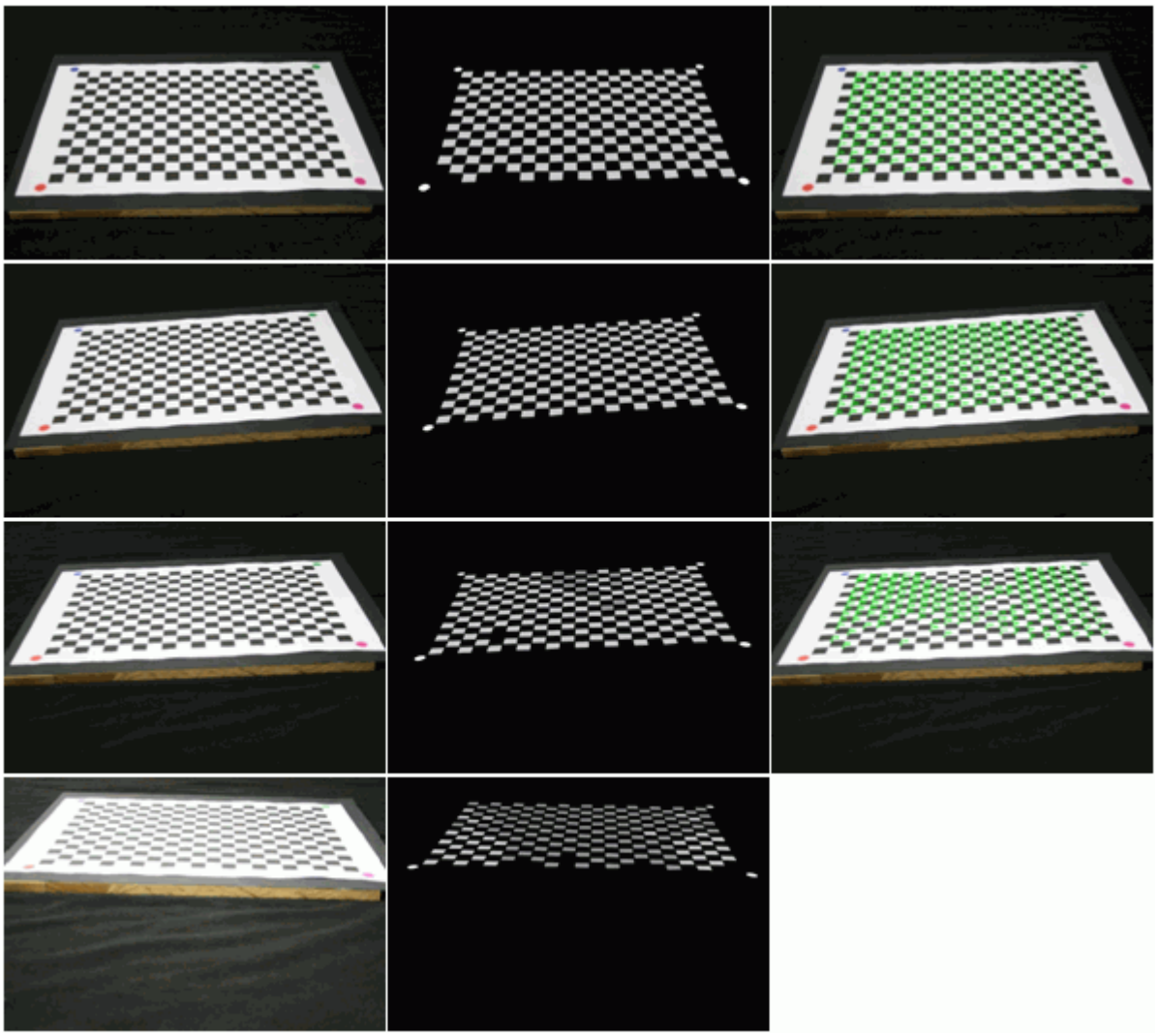

Fig. 5.15 – From left to right: 4 of the 6 original images obtained by approximating the camera towards the calibration pattern, while maintaining the camera's centre with a constant X coordinate; results obtained after pattern circles (brighter masks) and squares detection (medium gray masks); and final calibration inner vertices (green crosses) detected and used for camera's pose estimation.

The camera's extrinsic parameters, i.e. camera's poses, were impossible to determine for some images of the extra set of images of the first sequence. For example, on the image from the set obtained by keeping the camera's optical centre X world coordinate, the blue circle of the calibration pattern was not successfully detected, Fig. 5.15. The camera's angle is defined as the line of sight when viewed from the side; in this case the angle between the line of sight and the calibration pattern. With this definition, it was observed that for an angle less than 35 degrees, the camera could not be calibrated, mainly due to failure in the detection of the four circles.

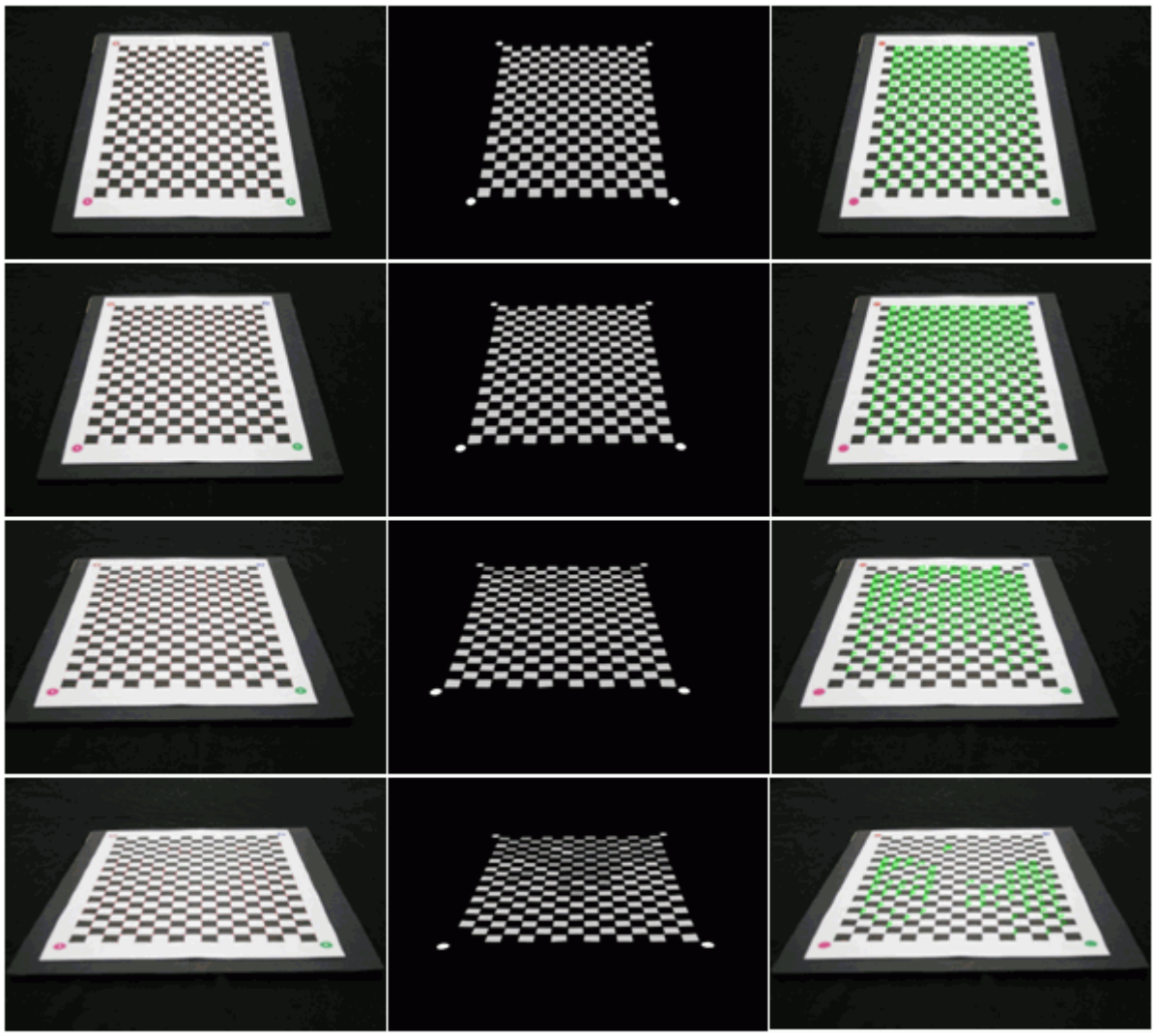

Fig. 5.16 – From left to right: 4 of the 6 original images obtained by approximating the camera towards the calibration pattern, while maintaining the camera's centre with a constant Y coordinate; results obtained after pattern circles (brighter masks) and squares detection (medium gray masks); and final calibration inner vertices (green crosses) detected and used for camera's pose estimation.

On the other hand, for all images obtained by keeping the Y world coordinate of the camera's optical centre, all pattern circles were successfully detected, Fig. 5.16. Similarly to the previous set of images, the smallest angle between the calibration pattern and the camera's centre was of 32 degrees. However, since on this set the calibration pattern was more centred on the image plane, the developed pattern detection method was able to effectively extract the four circles and some of the chessboard vertices.

For 2 of the 4 images acquired by decreasing the focus, the 4 circles were not successfully detected, Fig. 5.17. With decreasing the focus, image processing techniques used to extract chessboard vertices and coloured circles start to fail, due to the blending of pixel information amongst neighbours.

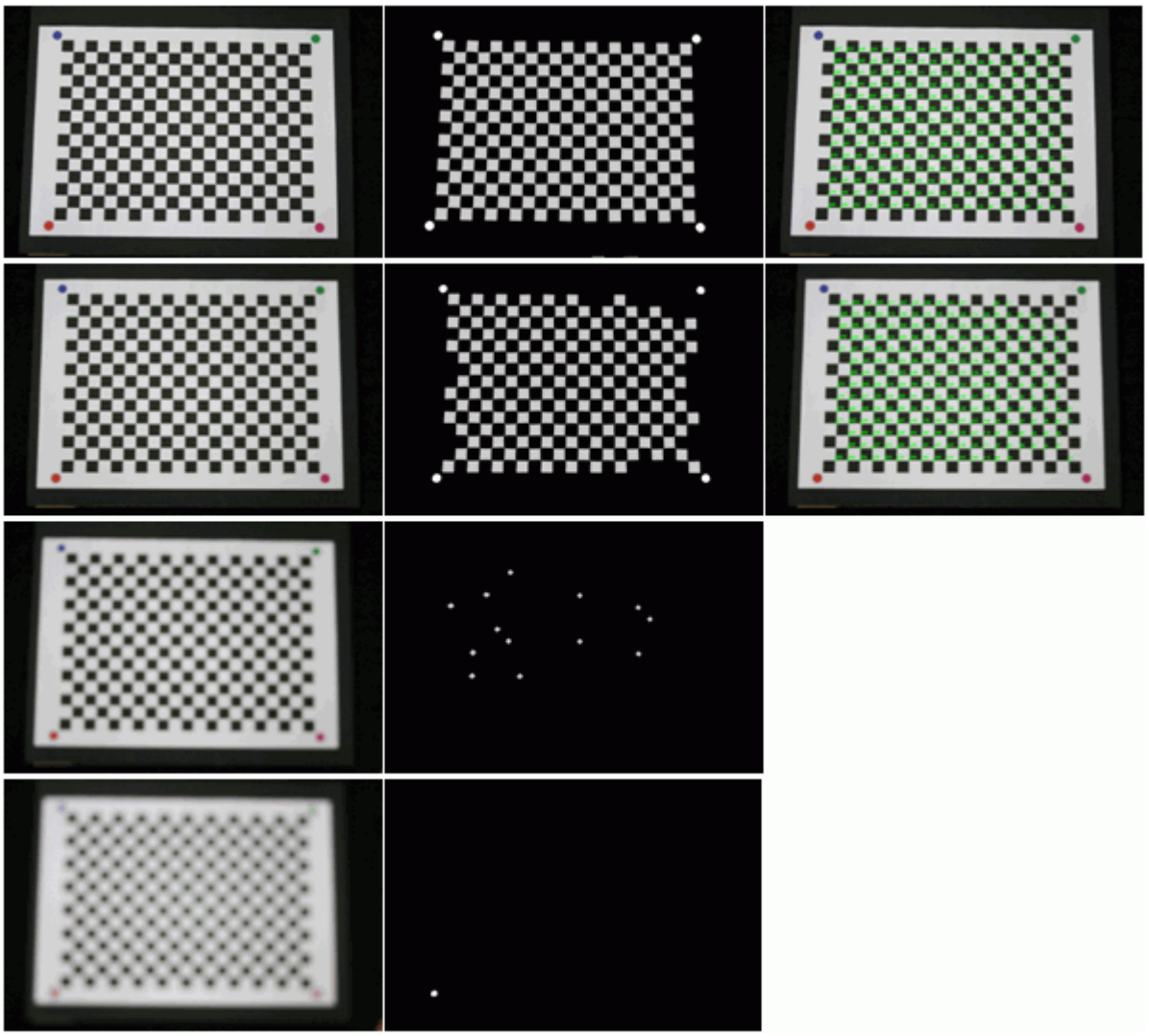

Fig. 5.17 – From left to right: original images obtained by decreasing the camera's focus, results after pattern circles (brighter masks) and squares detection (medium gray masks) and final calibration inner vertices detected and used for camera's pose estimation (green crosses).

Based on these results, one can conclude that even if the calibration pattern is fully observed in the image, not all viewpoints allow for camera's pose estimation; namely, those where the calibration pattern features suffer from strong distortion and down-sizing due to image projection. Also, a good camera focus is fundamental for a correct detection of the pattern's circles and vertices and, since the calibration method adopted assumes fixed intrinsic parameters, the distance between the different viewpoints and the calibration pattern should not change very much.

Finally, all circles and chessboard vertices were successfully extracted and matched among the last 11 images of the second sequence (the ones with the Rubik cube placed over the calibration pattern).

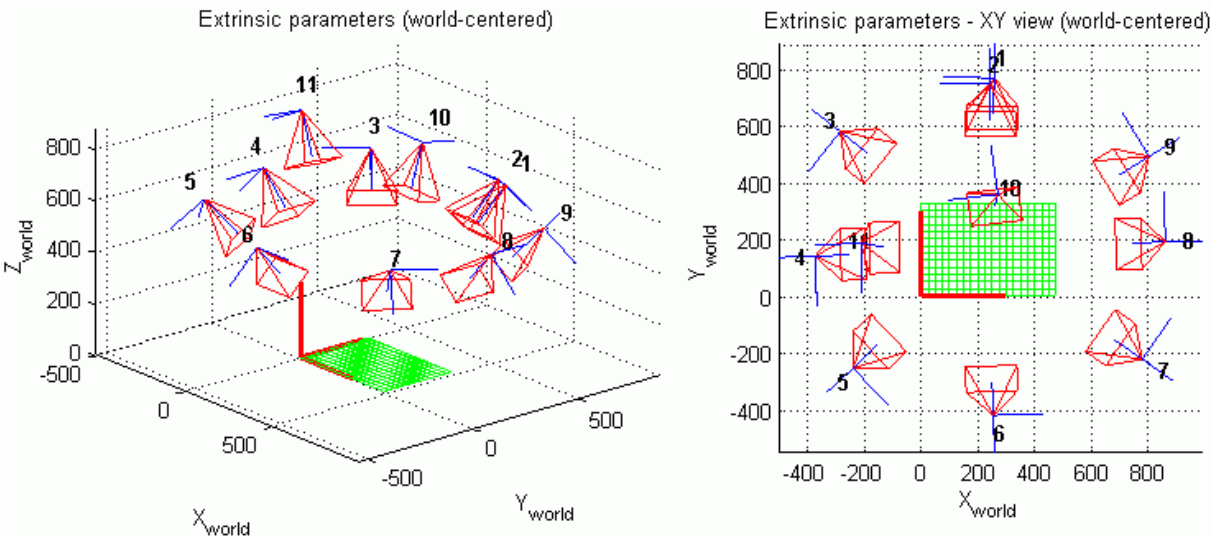

Fig. 5.18 – 3D representation of the camera's extrinsic parameters obtained with the second image sequence of the Rubik cube object. World 3D axes (in red) are located on the bottom-left vertex of the chessboard (the green grid). In both graphics, the scale is in mm.

Although the cube has red, green and blue surface blocks, these were not mistakenly confused with the calibration pattern circles. This was achieved by eliminating image contours with less than 5 vertices, when searching for the pattern circles.

In Fig. 5.18 a graphical 3D representation of the camera's extrinsic parameters obtained for the 11 images of the second image sequence can be observed. Average reprojection error for the second image sequence was of $e_{avg} = (0.1967,\ 0.4964)$ mm and with a standard deviation of $e_{std} = (0.1614,\ 0.3784)$ mm.

Image segmentation

Since the test object does not have a surface colour similar to the human skin, automation using image processing techniques was difficult to implement due to the similarity between object and calibration pattern, and so the segmentation was done semi-automatically.

First, edge detection was performed using a Sobel approximation to the derivative, followed by image dilation to join line segments, with a 5-pixels length cross as the structuring element, and then a manual flood-filling of the holes of the Rubik cube imaged patches, Fig. 5.19. The flood-fill operation changes connected background pixels, i.e. black pixels, to foreground pixels, i.e. white pixels, starting from the points specified manually. The boundary of the fill operation is 4-connected.

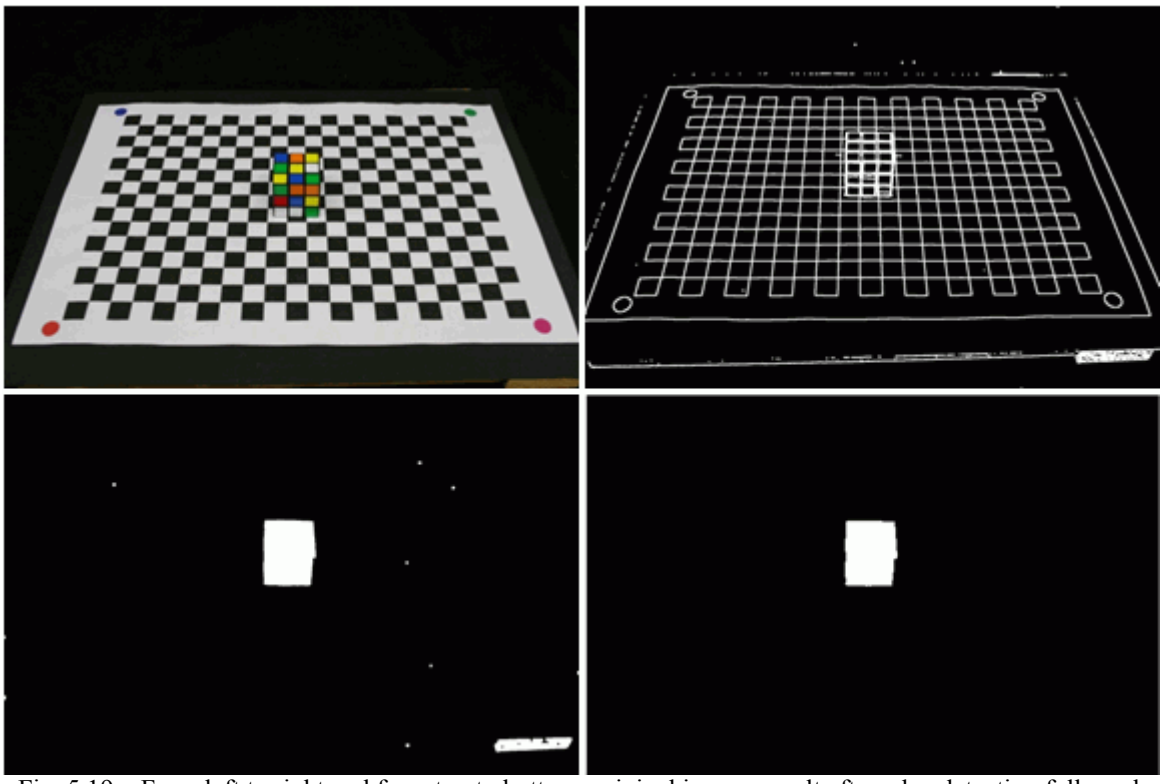

Fig. 5.19 – From left to right and from top to bottom: original image; result after edge detection followed by dilation; connected regions obtained by manually hole flood-filling; and the final silhouette determined by selecting the region with the biggest area.

Finally, since the flood-filling operation sometimes missed image parts containing the Rubik cube, all closed contours were extracted, and the correct silhouette was obtained by selecting the contour with the biggest area and again, filling its holes.

Volumetric-based reconstruction

As explained in Section 4.3.4, combining the original and segmented images and the calibration results, 3D models could be obtained through the previously described volumetric-based method.

From image analysis alone, one can infer some of the values for the initial bounding box and compare them with the ones obtained from the method proposed in Section 4.3.4, Table 5.2. For the X- and Y- coordinates, the determined values were close to the real ones, Fig. 5.20. The calculation of $\max_Z$ took five iterations in order the decrease the initial value determined using Equation (4.24), Fig. 5.21.

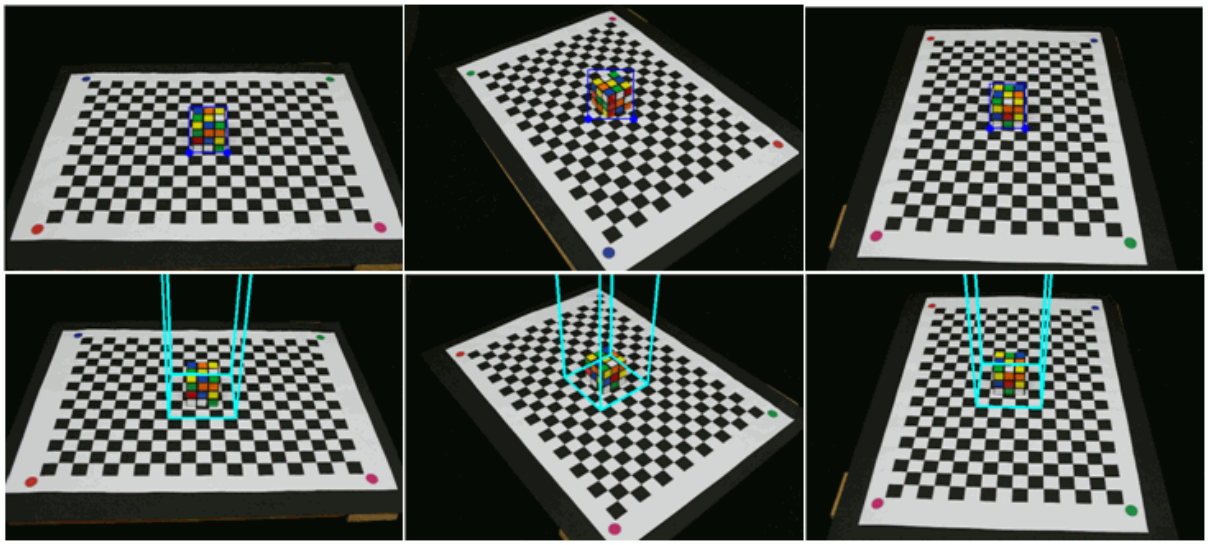

Fig. 5.20 – On the top: back-projection of calculated silhouette bounding box (blue rectangle) with the bottom vertices marked with blue circles. On the bottom: back-projection of the computed initial bounding volume.

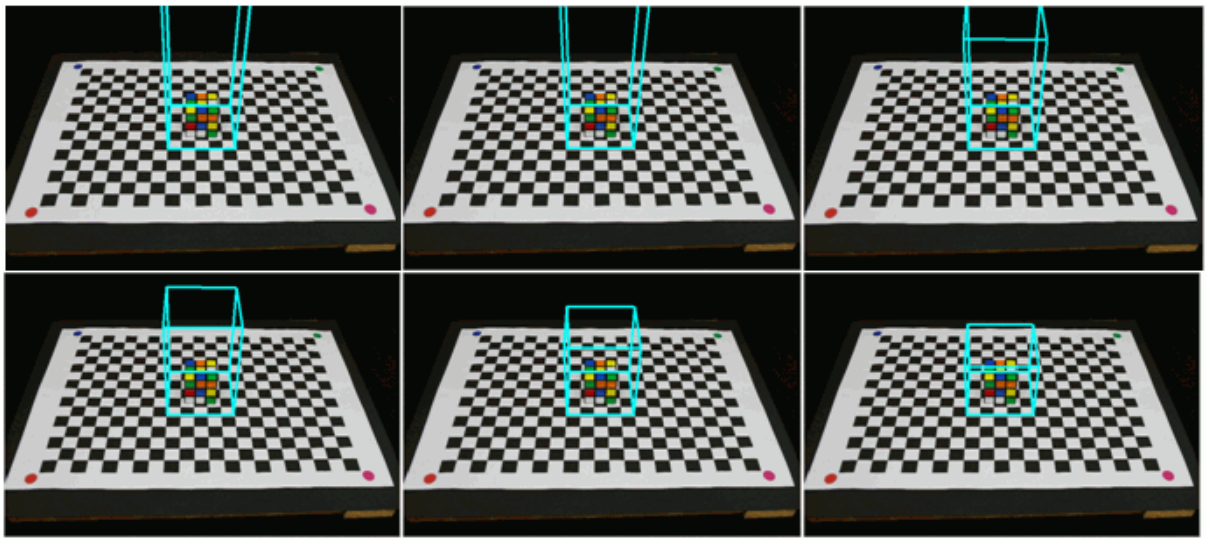

Fig. 5.21 – From left to right, from top to bottom: six back-projections (blue parallelepiped) on one of the acquired images. A total of seven iterations was required for the initial bounding volume computation of the Rubik cube object, by decreasing the maximum height of the volume.

Fig. 5.22 demonstrates the evolution of the visual hull computation for different levels of the volumetric reconstruction, using the inferred initial bounding values presented in Table 5.2.

After a few iterations, the 3D model can be considered a good approximation of the real object, excepting the cusping effect, which becomes more evident as the voxels are carved away.

Table 5.2 – Initial bounding volume measures
for the Rubik cube object (in mm).

	Inferred from images (number of pattern squares times square's size)	Automatically computed (using the method described in Section 4.3.4)
$\min_X$	8 x 25 = 200	177
$\max_X$	11 x 25 = 275	298
$\min_Y$	5 x 25 = 125	100
$\max_Y$	8 x 25 = 200	222
$\max_Z$	75 (same size as X and Y)	115

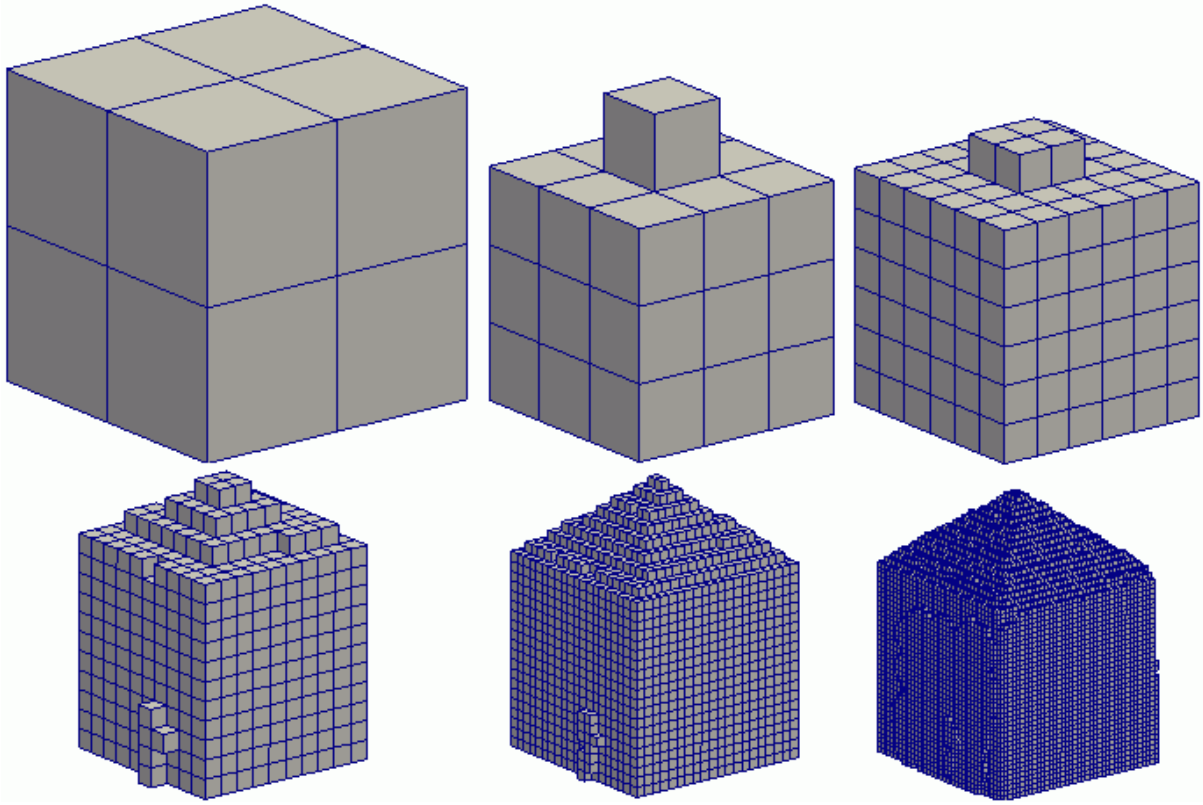

Fig. 5.22 – Visual hull computation of the Rubik cube object: from left to right and from top to bottom, evolution of the 3D visual hull from level 1 to 6, respectively. The blue lines are the wireframe representation of individual voxels (gray squares).

Fig. 5.23 shows a cropped 3D octree-based model, built adopting 5 levels. The voxels classified as inside on early levels are not sub-divided, saving both processing time and memory usage.

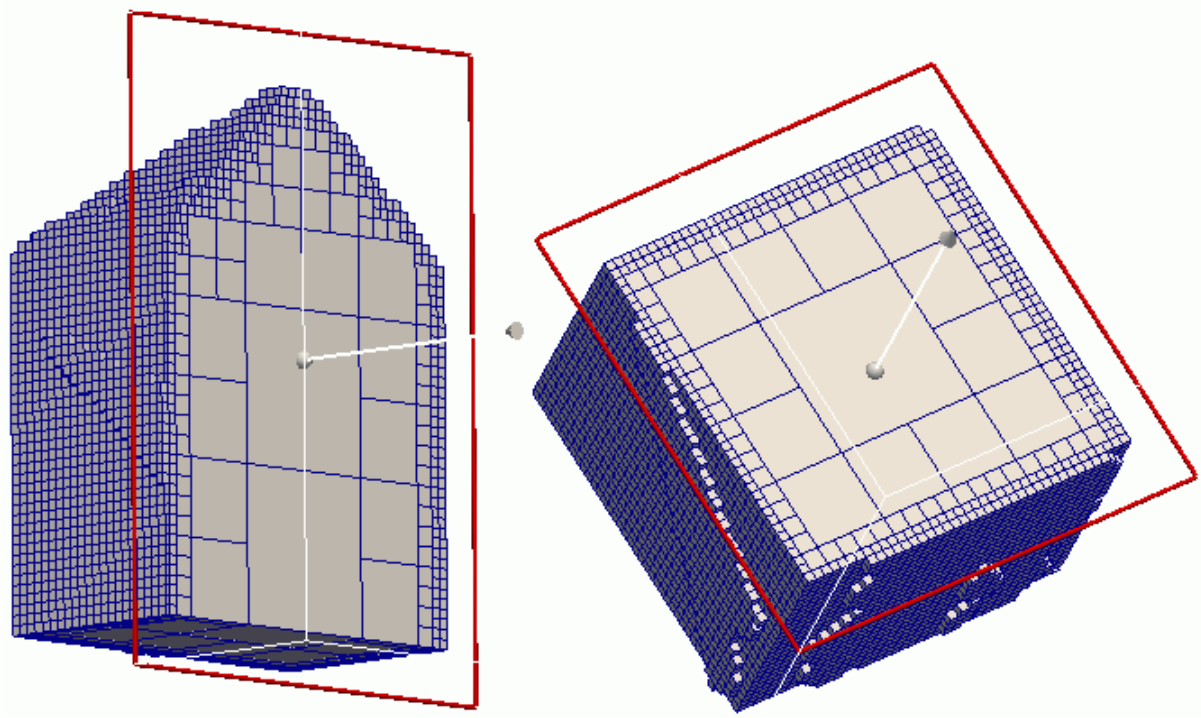

Fig. 5.23 – Visualization of the 3D octree data, with 5 levels, representing the computed visual hull. A plane perpendicular to the world Y-axis, on the left, and to the Z-axis, on the right, was used to crop a 3D model of Rubik cube object. All voxels with the same size belong to equal level of the octree data structure.

Fig. 5.24 and Fig. 5.25 shows two sets of the reconstruction results for the Rubik cube using the volumetric-based method using only silhouettes and with the automatically computed initial bounding values presented in Table 5.2. The sets differ on the projection criterion of the voxels: rectangular or exact (see Fig. 4.25).

The finer resolution models obtained from both voxel projection criteria were very similar. However, with exact projection, the visual hull was faster to approximate the real object's shape, as can be realized when comparing the rectangular projection, Fig. 5.24, with the exact projection, Fig. 5.25.

Fig. 5.26 shows two graphs comparing the time required on the two sets of reconstructions of the Rubik cube presented in Fig. 5.24 and Fig. 5.25, as well as the final number of voxels that constitute the volumetric 3D models built. On the left panel of this figure, the relation between the total amount of computational time required to reconstruct a 3D model of the Rubik cube, using rectangular and exact projections, and the number of iterations is depicted. On the right panel, the total number of voxels belonging to a 3D model, using rectangular and exact projections, is also compared for the different number of reconstruction levels considered.

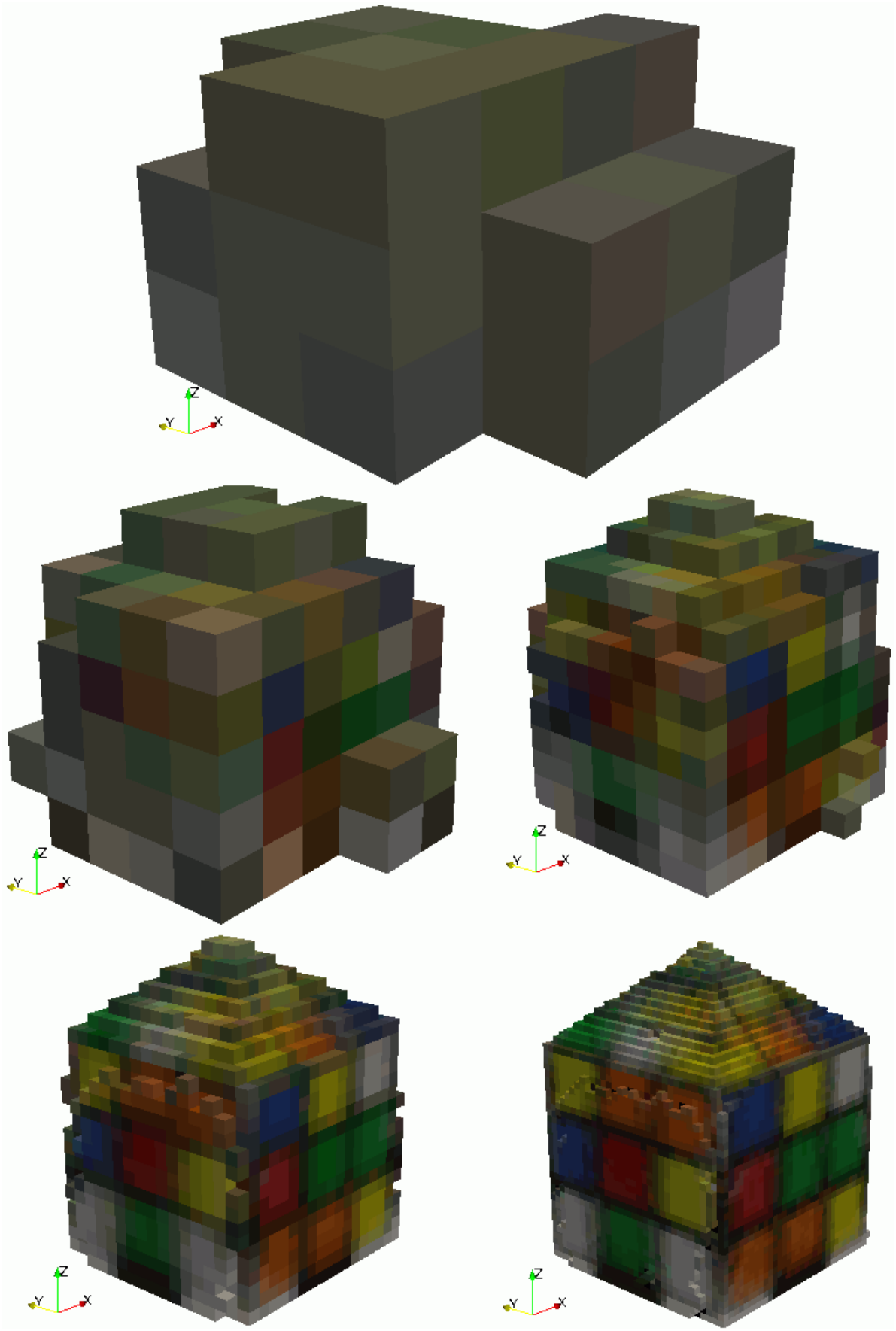

Fig. 5.24 – 3D models obtained for the Rubik cube object using rectangular voxel projection. From top to bottom and from left to right: the octree refinement level was sequentially increased from 2 to 6 levels, respectively.

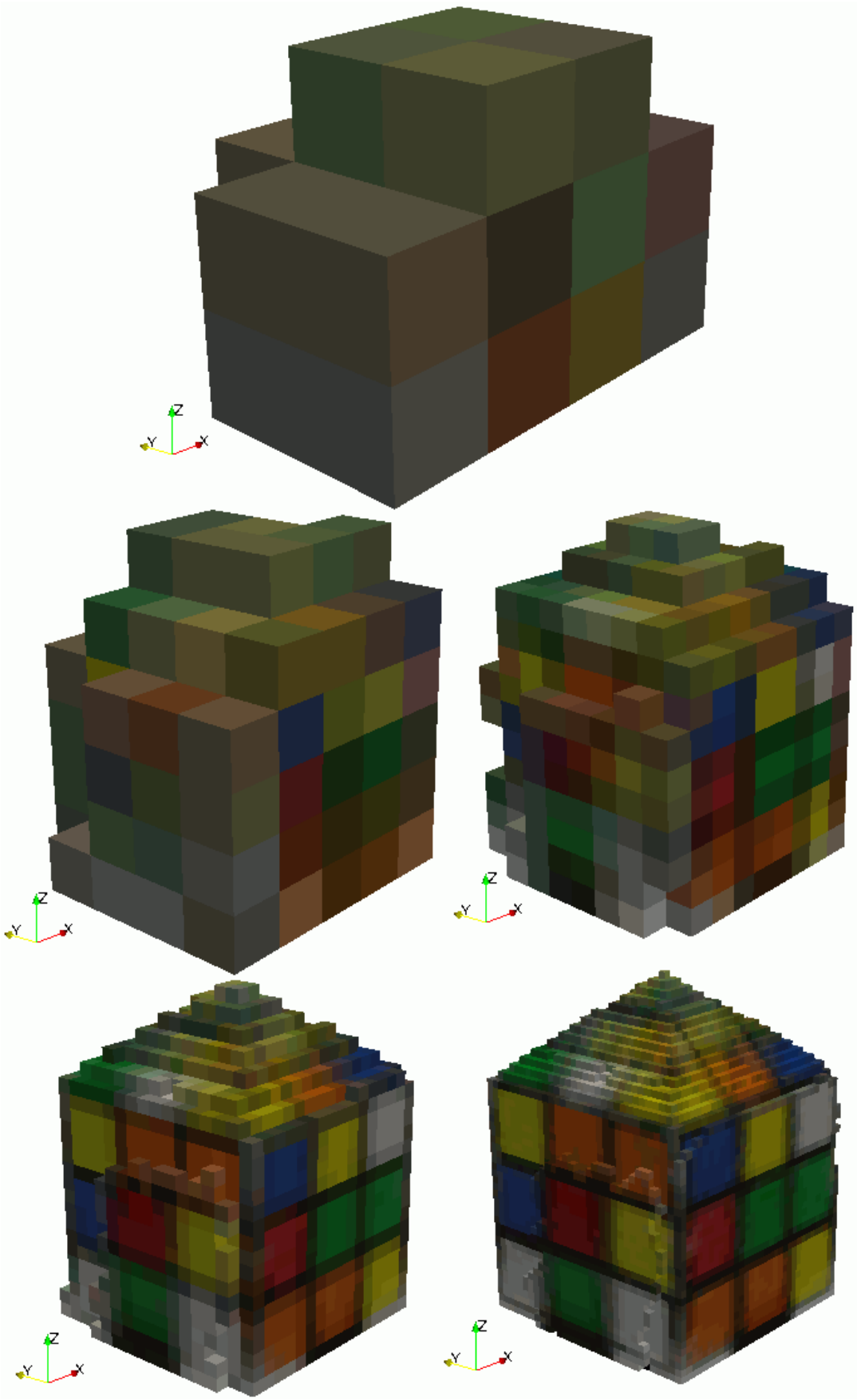

Fig. 5.25 – 3D models obtained for the Rubik cube object using exact voxel projection. From top to bottom and from left to right: the octree refinement level was sequentially increased from 2 to 6 levels, respectively.

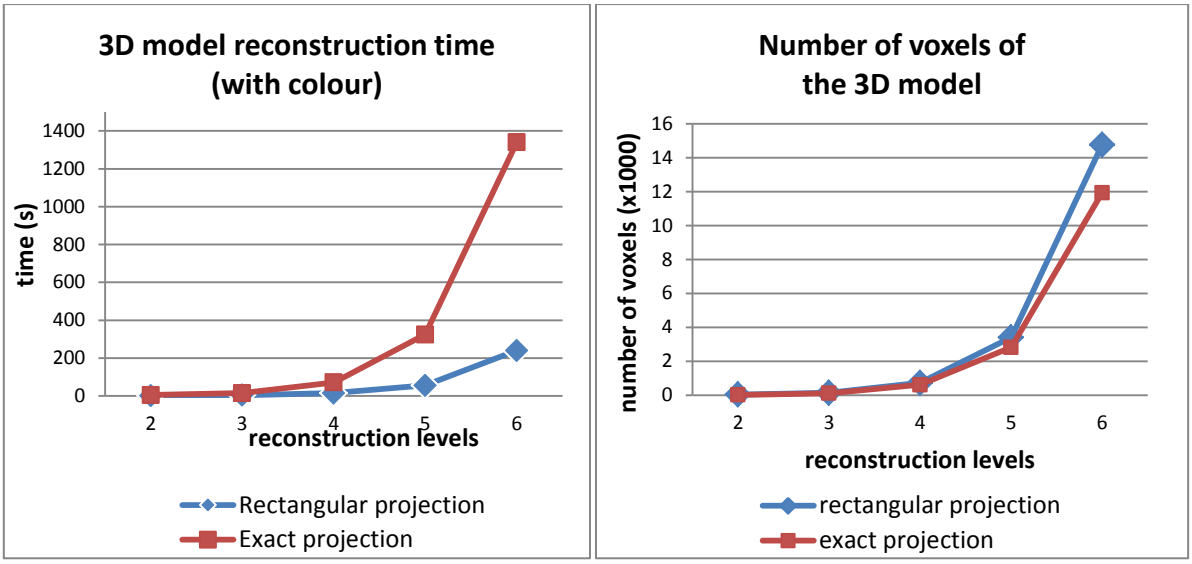

Fig. 5.26 – On the left: total amount of time required to reconstruct the Rubik cube, using rectangular and exact projections. On the right: number of voxels of the reconstruct 3D models for the Rubik cube using rectangular and exact projections.

On the left panel of Fig. 5.26, if a trendline was drawn for both reconstruction types, the best approximation would be traduced by an exponential line. In fact, the reconstruction time grew exponentially with the octrees final level L, because the maximum number of voxels is 8^L, as confirmed by the right panel. However, this is not the only factor that contributes for the higher time taken by the reconstructions using exact footprints. Another reason is the smaller computational effort required to compute the voxels' image footprints for the rectangular case:

- for an exact footprint, it is necessary to fill the footprint polygon and then compare it with the silhouette;
- for a rectangular footprint, the search is performed directly over the silhouette within the ROI defined by the 4-sided rectangular footprint's polygon.

Analysing the graph on the left panel of Fig. 5.27, it can be noticeable that determining the voxels' colour of the final 3D models can take around 20% of the total reconstruction time when using rectangular projection, but, when using exact projection, it is increased to around 80% of the total reconstruction time. This difference is explained by the considerable inferior number of footprints determined when computing the visual hull than when it is being coloured. Since the colour determination for the final visual hull needs more computations of footprints, its influence in the exact projection case becomes more influent on the overall processing time.

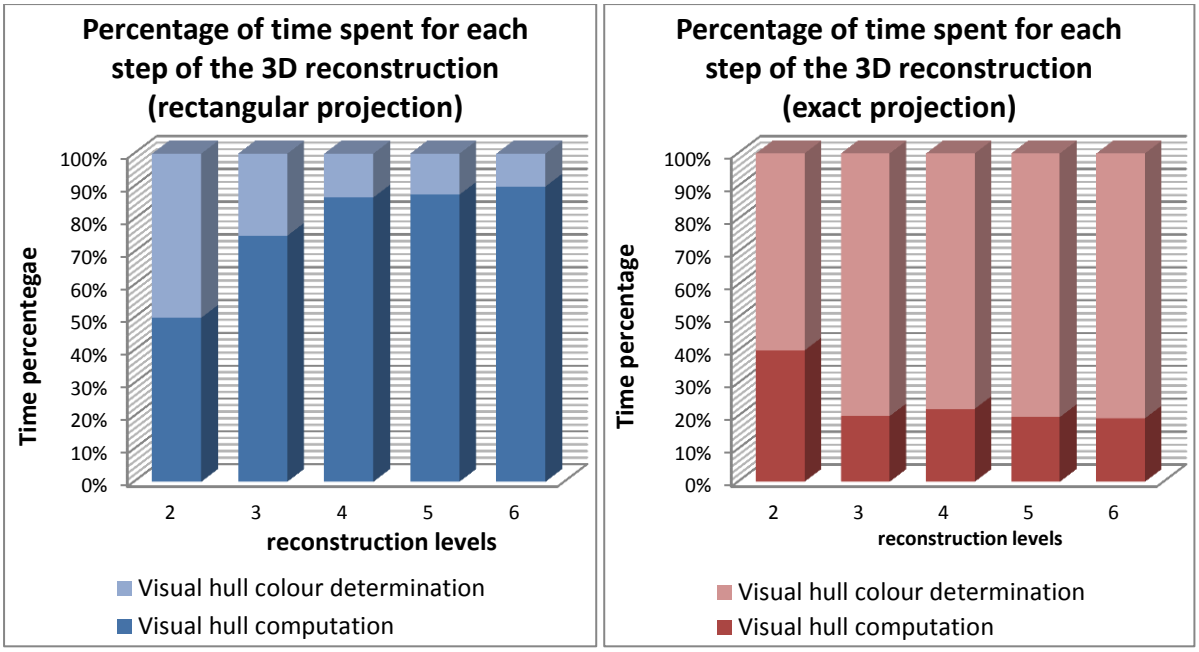

Fig. 5.27 – For both graphics, the column bars represent the percentage of time spent on the two major steps of the developed volumetric reconstruction method using only silhouettes: visual hull computation and voxel colouring. On the left, the graph reflects the times using rectangular projection to determine a voxel's footprint, and on the right, the graph reflects the times using exact projection.

The computational time difference is even more important when using the photo-consistency option, because in this case, the voxel visibility must be computed every time a voxel is considered inconsistent and removed from the final 3D model.

In order to evaluate the influence of the camera calibration method on the 3D reconstruction accuracy, two new 3D models of the Rubik were built: one using the original camera's parameters and a second one with the parameters obtained by disregarding the step of calibration pattern's vertices refinement with sub-pixel precision. For the Rubik cube object, the maximum deviation of the vertices' location was about 1.5 pixels, Fig. 5.28.

In terms of the overall reprojection error, it increases without the sub-pixel refinement. The values obtained with the sub-pixel refinement for the average and standard deviation of the reprojection errors were equal to $e_{avg} = (0.1967, 0.4964)$ pixels and $e_{std} = (0.1614, 0.3784)$ pixels, respectively. Without the refinement step, the new values obtained where of $e_{avg} = (0.3529, 0.5567)$ and $e_{std} = (0.2662, 0.4207)$, respectively. When comparing the calibration results obtained in both cases, there was no surprise when the 3D models built were very similar. For an objective comparison, the one-sided Hausdorff distance, from accurate to less-accurate, was computed.

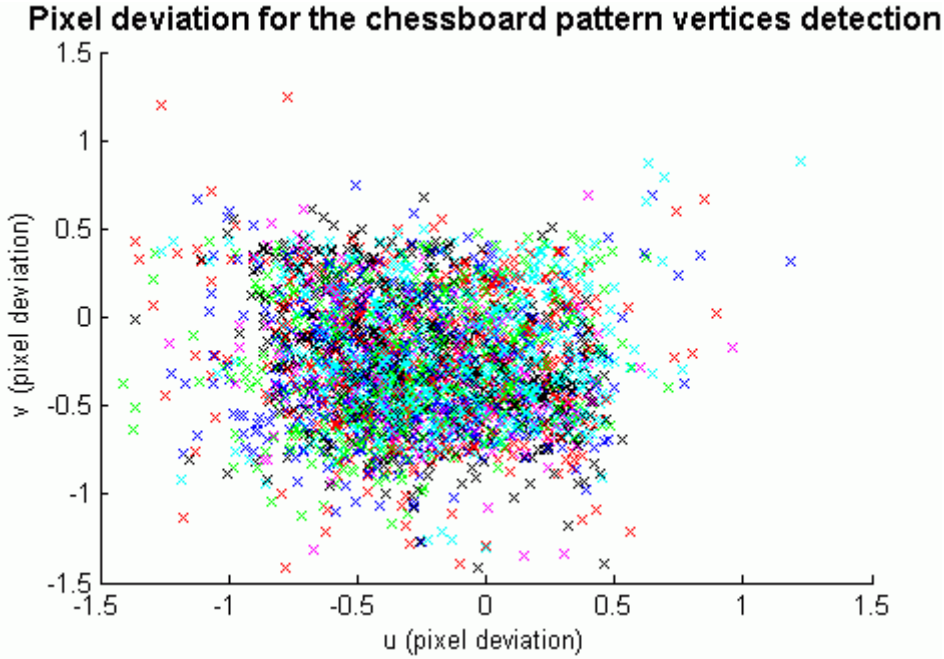

Fig. 5.28 – Pixel deviation between the vertices founded with and without sub-pixel precision during the calibration procedure for the Rubik cube object (different colours represent distinct images).

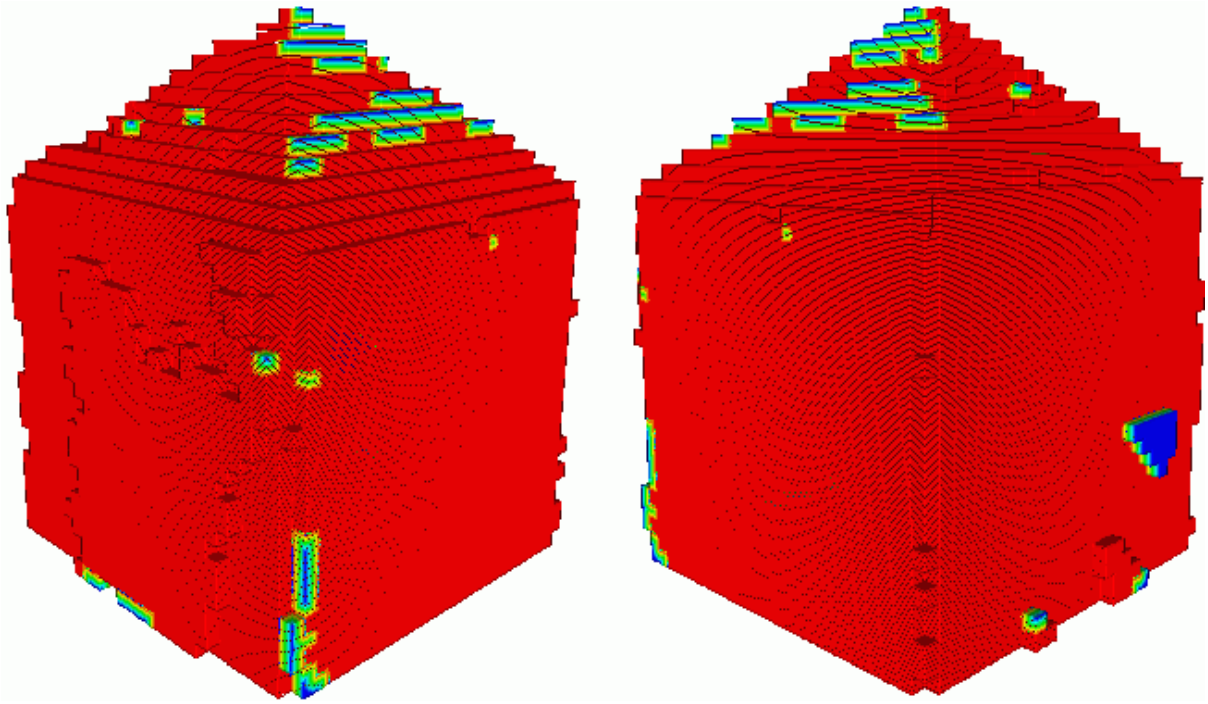

Fig. 5.29 – Two viewpoints of the Hausdorff distance computed between the 3D models of the Rubik cube built with and without sub-pixel precision of the pattern vertices' detection during the camera calibration procedure.

Fig. 5.29 shows two views of the computed Hausdorff distance with the distance values colourized into a RGB colourmap: since that it is a red-green-blue map, then red is min and blue is max. As such, in this case, red means zero distance and blue higher distance.

Table 5.3 – Hausdorff distance statistics between the 3D models of the Rubik cube built with and without sub-pixel precision of the pattern vertices' detection during the camera calibration procedure.

Hausdorff distance (mm)	
min	0.000000
max	2.343994
mean	0.045885
RMS	0.270126

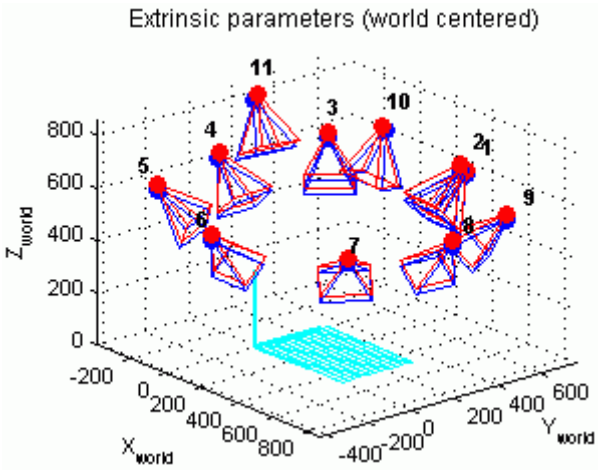

Fig. 5.30 – 3D representation of the camera's extrinsic parameters obtained with accurate (in red) and erroneous (in blue) pattern vertices detection. Coloured spheres represent the camera's centres. The world 3D axes are located on the bottom-left vertex of the chessboard (both in cyan).

Table 5.3 presents statistics concerning the computed Hausdorff distance. Considering a bounding box of the 3D models as reference, it can be observed that the average Hausdorff distance error is about 0.03% of its diagonal, which is very low.

For a further analysis on the influence of the calibration results in the 3D reconstruction accuracy, a second experience was made. A random noise of +/- 5 pixels was added to all pattern vertices detected, and, from the corrupted vertices' coordinates, the camera's intrinsic and extrinsic parameters were computed. Fig. 5.30 allows the comparison of the camera's extrinsic parameters obtained for the second image sequence.

Again, a new 3D model was built, using the same parameters as before. Fig. 5.31 shows two views of the computed Hausdorff distance, with the distance values colourized into a RGB colourmap, and Table 5.4 presents some statistics of the computed Hausdorff distance.

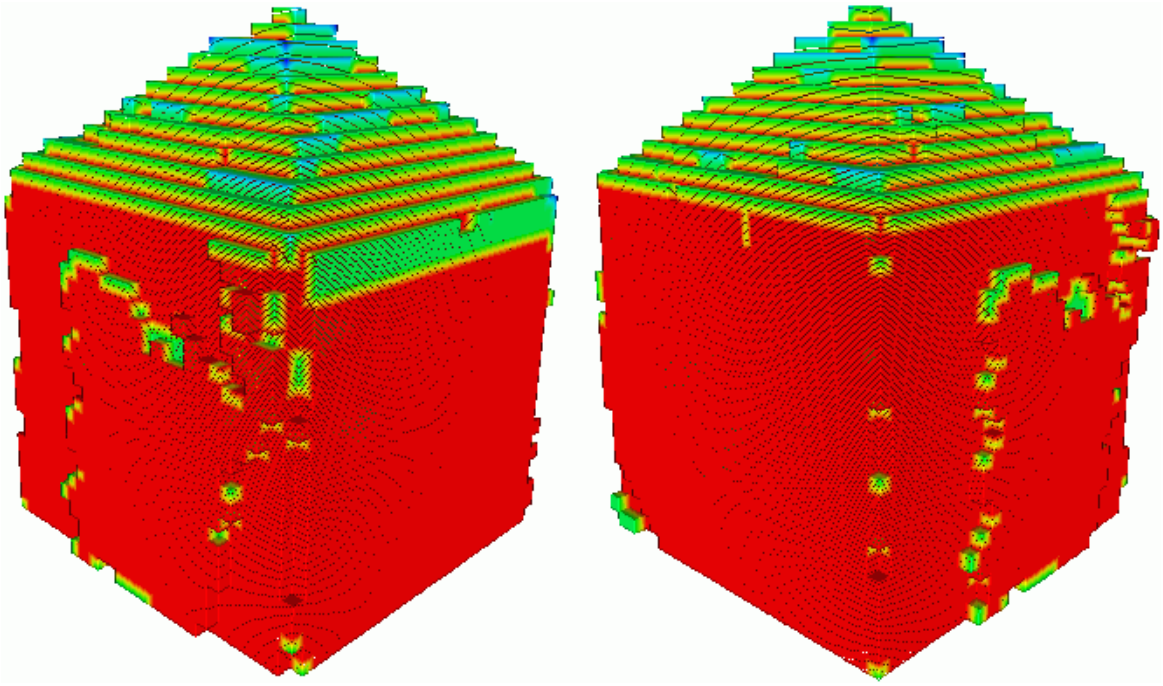

Fig. 5.31 – Two viewpoints of the Hausdorff distance computed between the 3D models of the Rubik cube built with and without random noise introduction after the pattern vertices' detection during the camera calibration procedure.

Table 5.4 – Hausdorff distance statistics between the 3D models of the Rubik cube built with and without random noise introduction after the pattern vertices' detection during the camera calibration procedure.

Hausdorff distance (mm)	
min	0.000000
max	4.059804
mean	0.516281
RMS	0.998691

Again, considering a bounding box of the 3D models as reference, it can be observed that the average Hausdorff distance error was about 0.4% of its diagonal, which approximately 10 times the previous value. Due to the configuration of the used setup, where the calibration pattern defines the XY world plane, errors induced by a defective calibration will mostly induce worst reconstructions on the upper part of the reconstructed objects, Fig. 5.32.

Another 3D model of the Rubik cube object was built, this time to compare the results obtained when the photo-consistency tests are used to evaluate the model's voxels. Since the cube has a highly textured surface, some experiments were performed in order to evaluate if the initial 3D model built only using silhouettes could be refined.

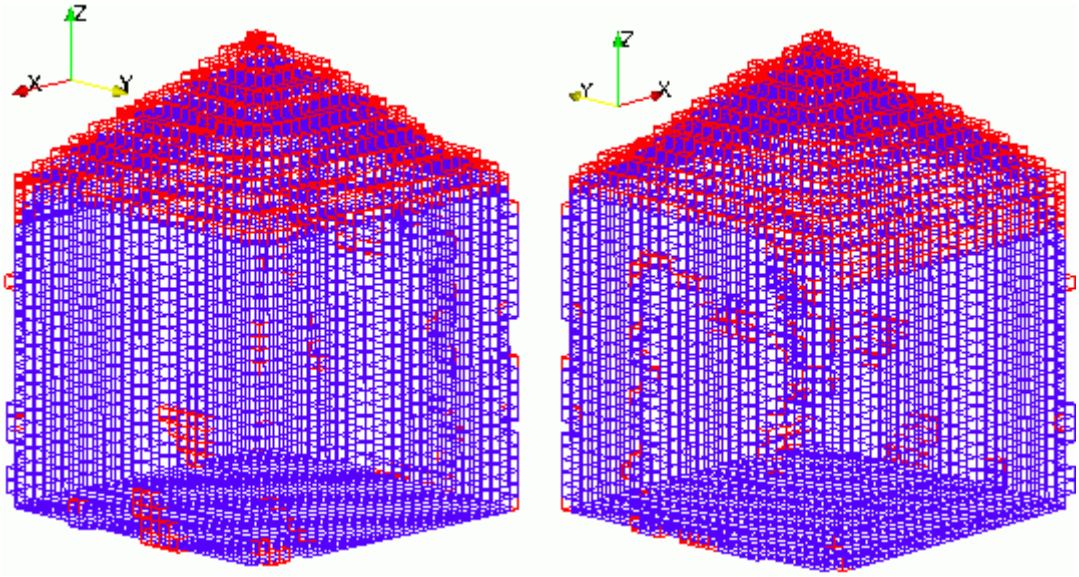

Fig. 5.32 – Wireframe representation of the 3D models built for the Rubik cube built with (in blue) and without (in red) random noise introduction after the pattern vertices' detection using the camera calibration procedure.

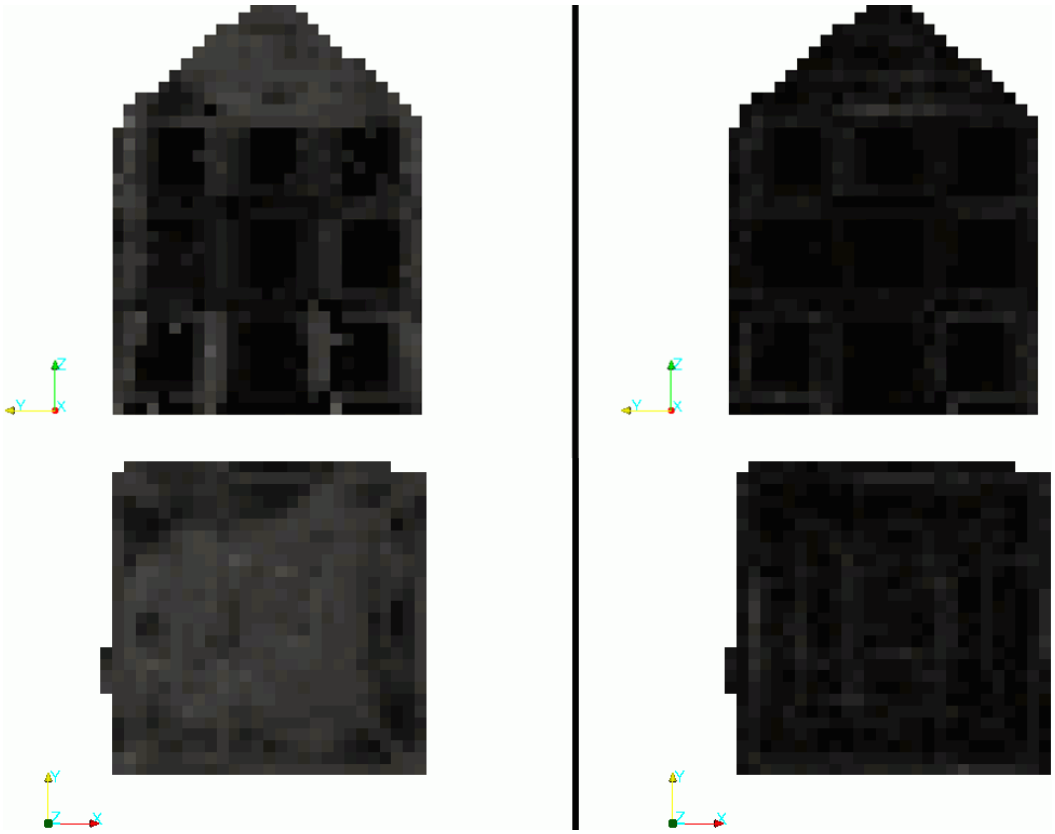

Fig. 5.33 – Two viewpoints (by column) of the 3D models textured with colour variance: darker colour mean lower variance. On the left, the variance is computed as the standard deviation over all image pixels from which a certain voxel is visible; on the right, the variance is computed as the mean standard deviation per image from which a certain voxel is visible.

On Fig. 5.33 the influence of texture for reconstruction methods based on photo-consistency is depicted. Using a standard consistency check, along with outside voxels, also voxels belonging to highly textured areas would be considered inconsistent.

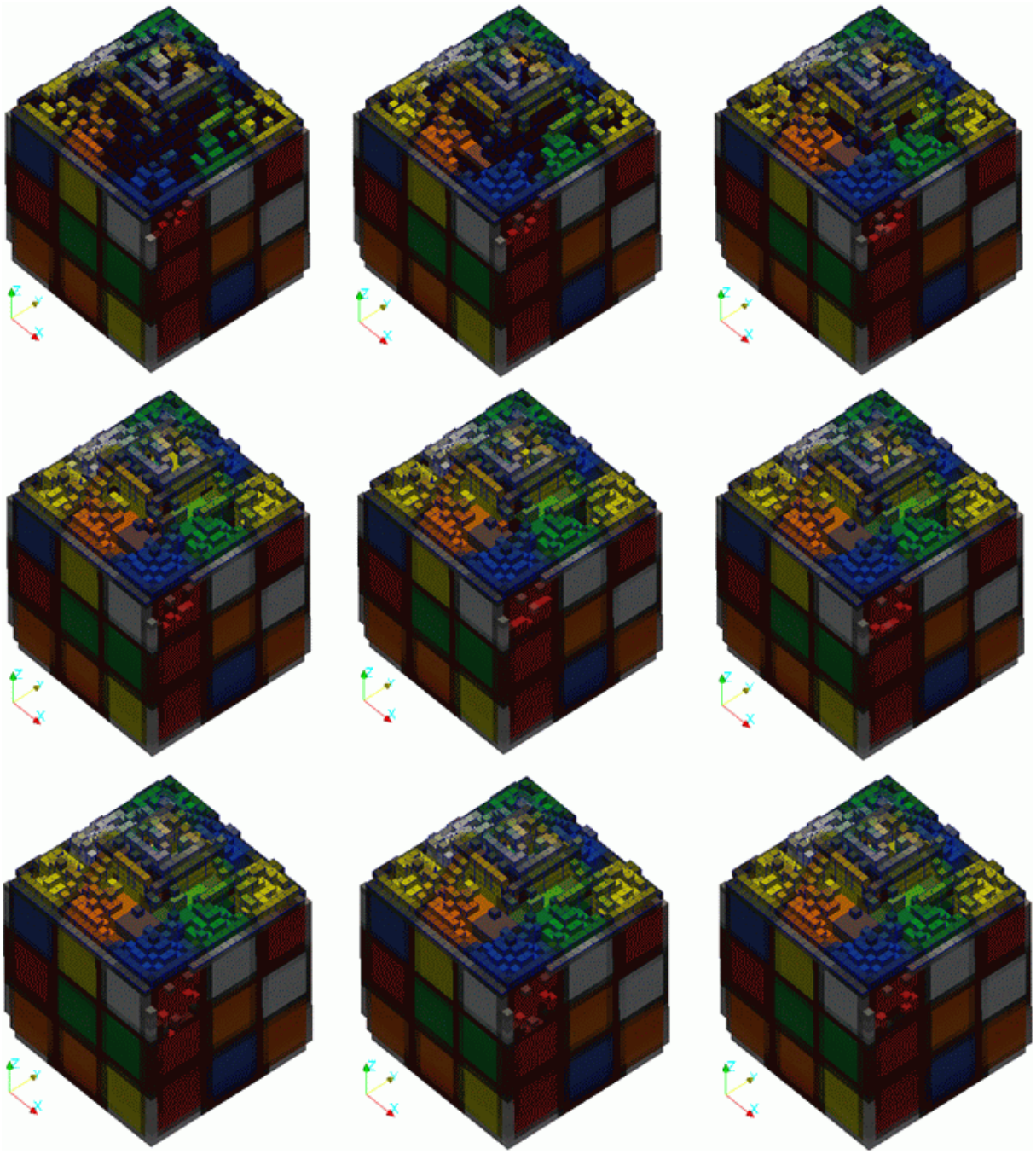

Fig. 5.34 – From left to right, and from top to bottom: 3D model reconstruction evolution of the Rubik cube using the volumetric reconstruction method based on photo-consistency with the following parameters: number of iterations equal to 7; exact voxel projection; photo-consistency thresholds $T_1 = T_2 = 5$. The voxels on the topmost surface have their edges highlighted in blue for better visualization.

Fig. 5.34 shows some of the reconstruction results for the same object using the volumetric method based on silhouettes and photo-consistency. For this set, accurate voxel projection was adopted, and the threshold values were defined as $T_1 = T_2 = 5$ (see Equation (4.29)), as these values were the ones that led to the best results in the experiments conducted.

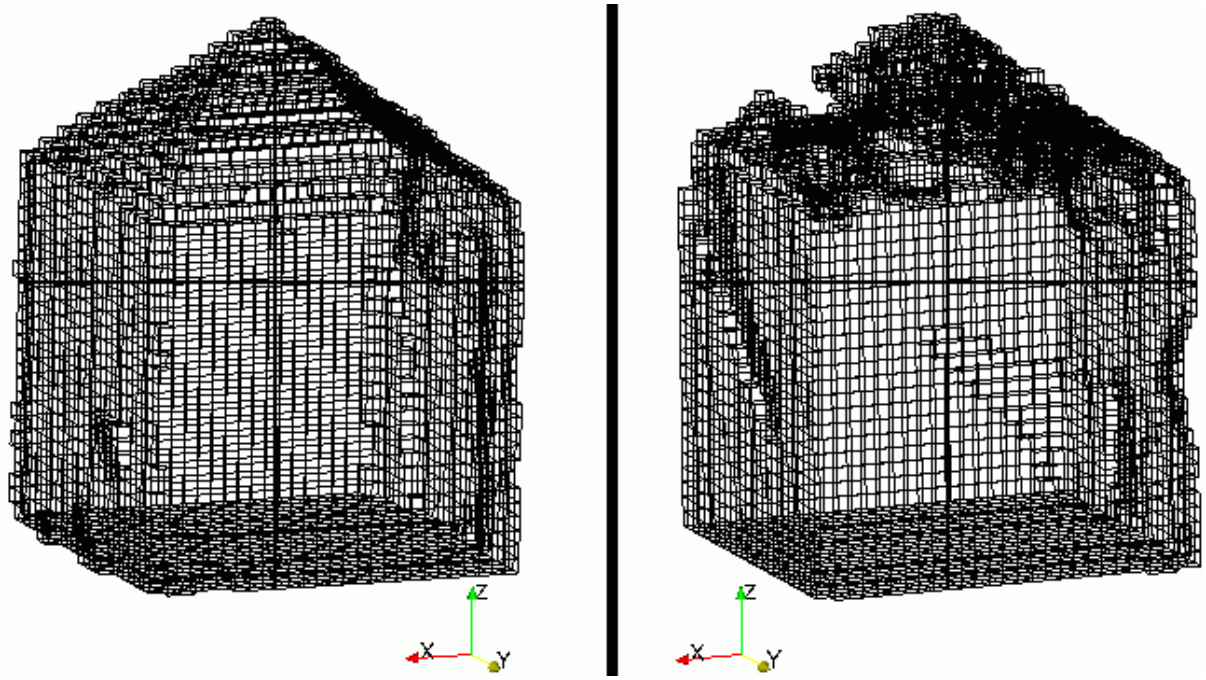

Fig. 5.35 – Wireframe representation of the obtained 3D models for the Rubik cube object using just silhouettes (on the left) and using also photo-consistency tests (on the right).

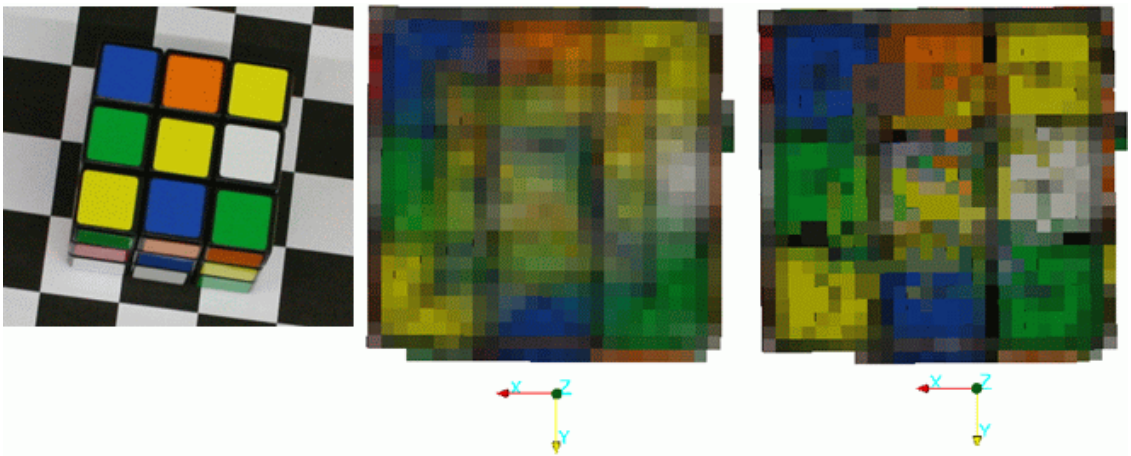

Fig. 5.36 – On the left: original top view of the Rubik cube object. On the middle: reconstructed 3D model based only on silhouettes. On the right: reconstructed 3D model based on silhouettes and photo-consistency.

Due to the removal of 1742 voxels from the initial 3D model composed by 6912 voxels, the total volume was decreased by 16%. Fig. 5.35 highlights the voxels edges for both 3D models built using only silhouettes and with photo-consistency testing for better evaluation of the voxel carving. Even using so low thresholds, not all of the top voxels were removed; however, it was possible to observe that this happened because the colours almost match with the real Rubik cube, Fig. 5.36.

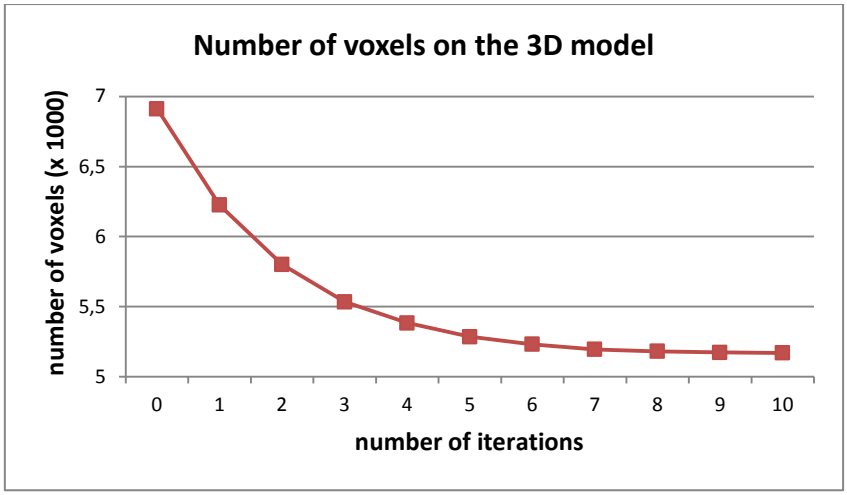

Fig. 5.37 – Evolution of the number of voxels on the reconstructed 3D model for the Rubik cube during the photo-consistency test iterations. Most of the voxel carving was done with the first four iterations of the photo-consistency test algorithm.

Fig. 5.37 despites a problem with the adopted photo-consistency test algorithm: although the first few iterations of the algorithm removed a large number of voxels, as it evolves, much of the surface voxels were found to be photo-consistent and were not removed. The result was that the already photo-consistent voxels were unnecessarily tested again on the subsequent iterations.

3D model assessment

Fig. 5.38 shows two graphs comparing some measures of the obtained 3D models and the real object, the Rubik cube.

With good calibration and reasonable silhouettes, volumetric-based methods are demonstrated to be conservative, i.e., voxels belonging to the actual object are never removed. The volume of the obtained 3D models drops very fast, achieving a suitable approximation after 6 iterations on the octree decomposition. It is slowed by the height (Z coordinate) because of the cusping effect of the visual hull.

Fig. 5.39 shows the result of applying a smoothing Laplacian filter to the obtained surface of the reconstructed 3D model. Fig. 5.40 shows the vertices displacements, which are much higher on stiffer edges.

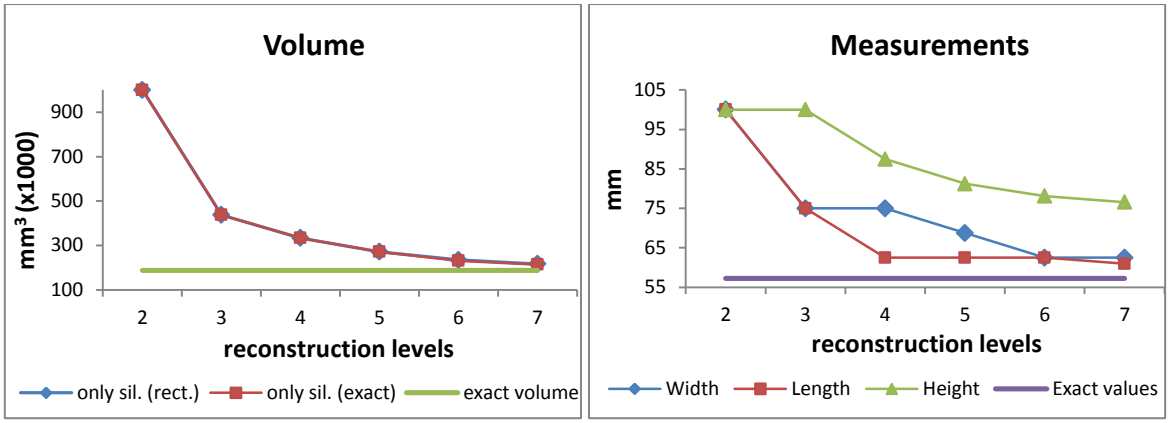

Fig. 5.38 – Measurements comparison between reconstructed 3D models and real Rubik cube (accurately measured using a calliper). On the right panel, it is possible to observe the cusping effect of the visual hull, which causes the height of Rubik cube to never approximate to its real value.

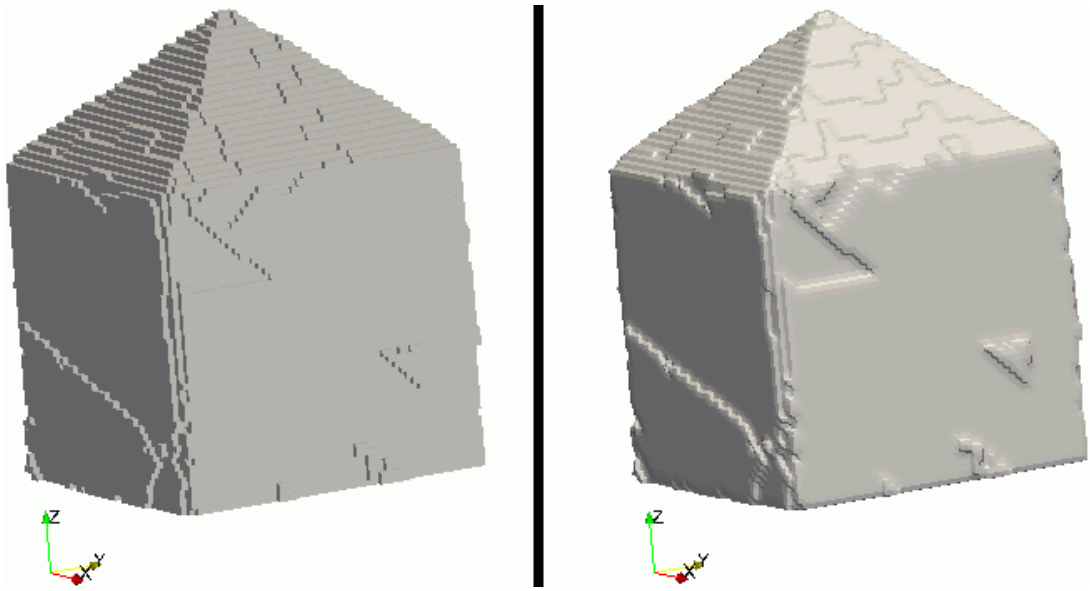

Fig. 5.39 – On the left: original 3D model surface obtained for the Rubik cube object. On the right: 3D model smoothed by performing 100 iterations of the Laplacian smoothing filter (voxel colour was removed for an easier visualization).

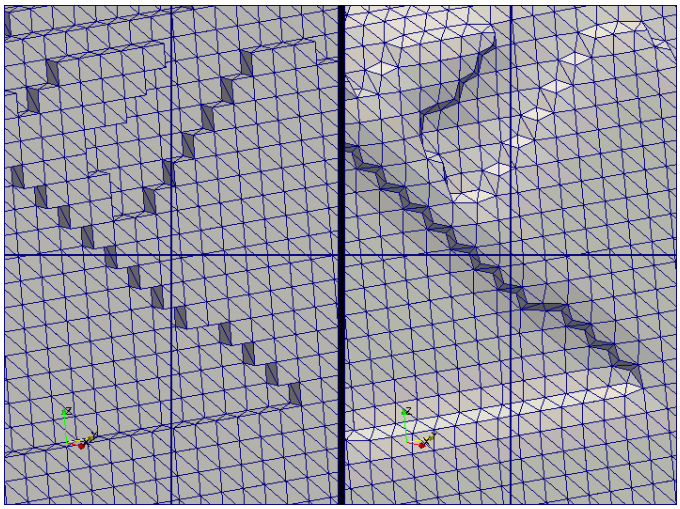

Fig. 5.40 – Zoom of the 3D models presented in Fig. 5.39, highlighting the polygonal meshes (blue lines).

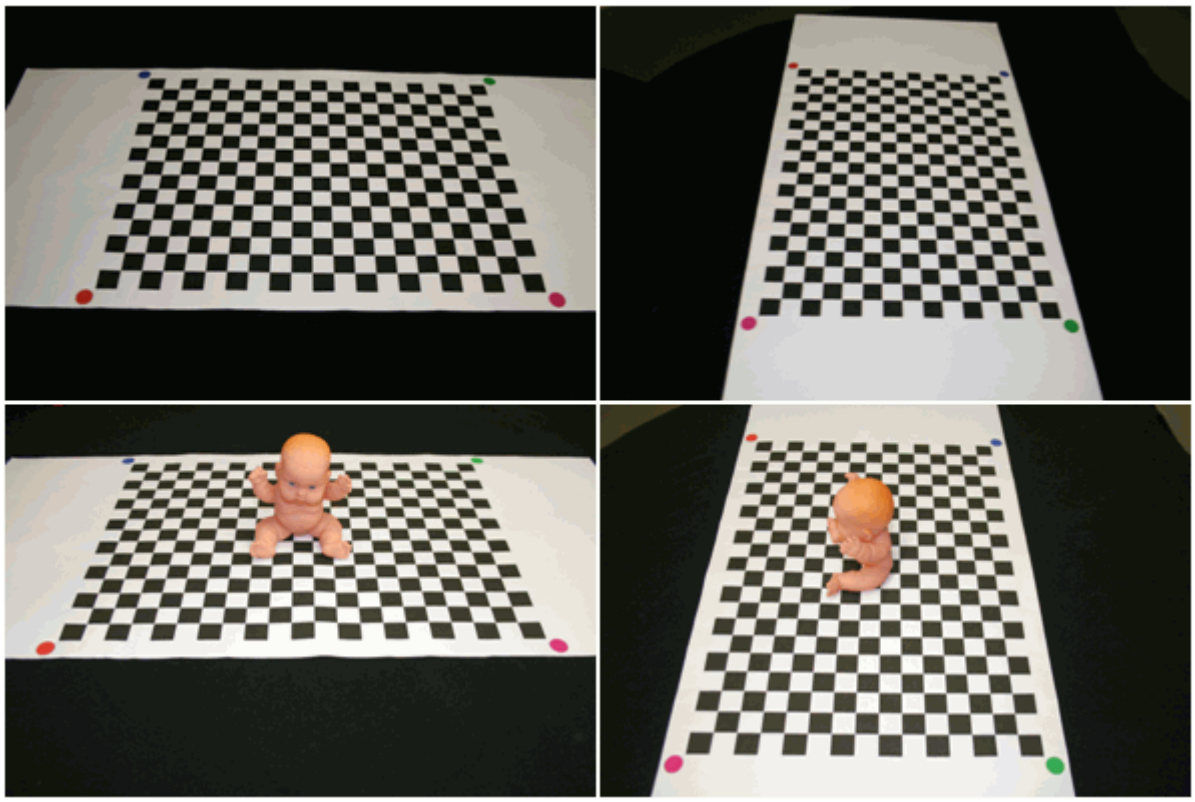

Fig. 5.41 – Example of two images of the first (on the top) and second image sequence (on the bottom) acquired for the baby toy object.

5.3.2 Plastic baby toy

The test results discussed hereby were performed using the same plastic baby toy address in Section 5.2.2.

First image sequence consisted on 8 images of the chessboard calibration pattern. For the second image sequence, 11 images were acquired with the plastic baby toy positioned on top of the chessboard pattern and moving the camera around the plastic toy, Fig. 5.41.

All images were acquired on a simple black background and with a resolution of 1936×1288 pixels.

Image segmentation

Since the object has a surface colour similar to the human skin, the segmentation results were good, Fig. 5.42. The worst results obtained were due to the effects of directional lighting and shadowing effects, Fig. 5.43.

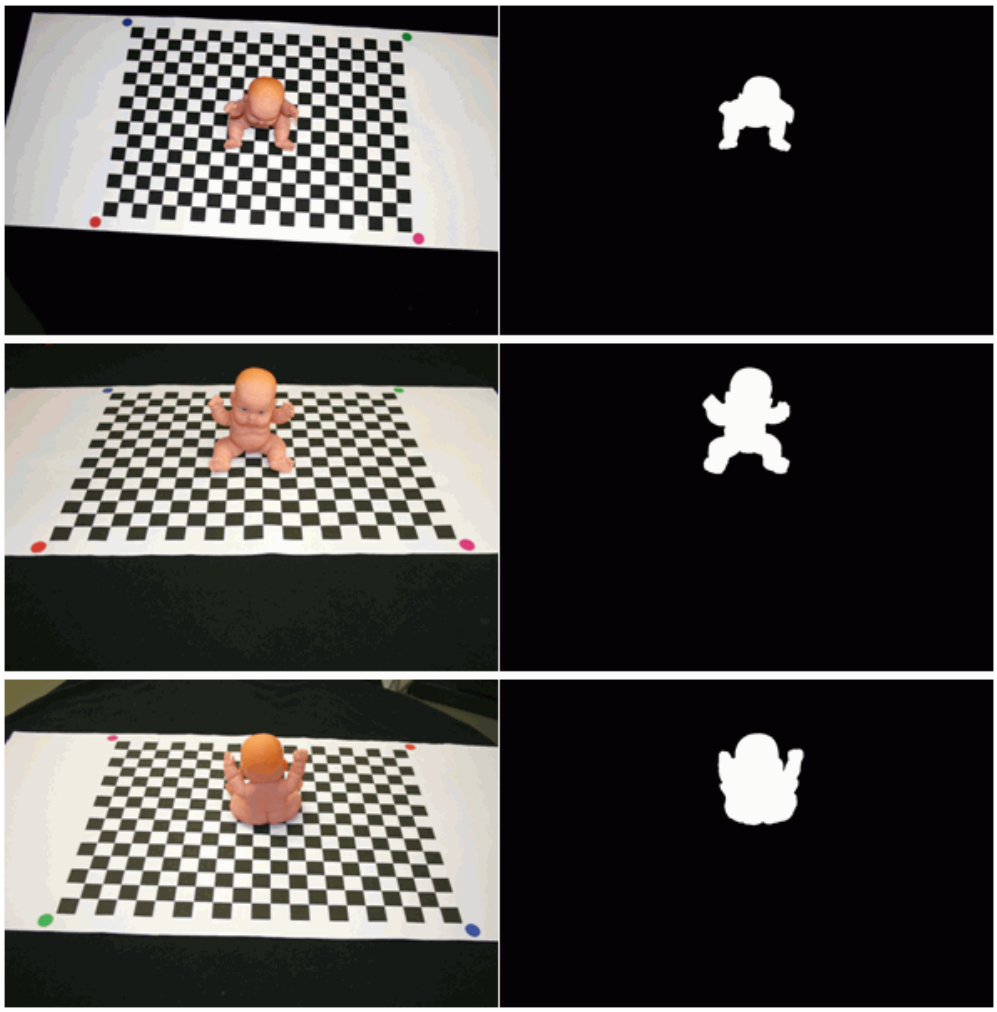

Fig. 5.42 – Original images of the plastic baby toy (on the left) and silhouettes obtained (on the right) through skin colour segmentation.

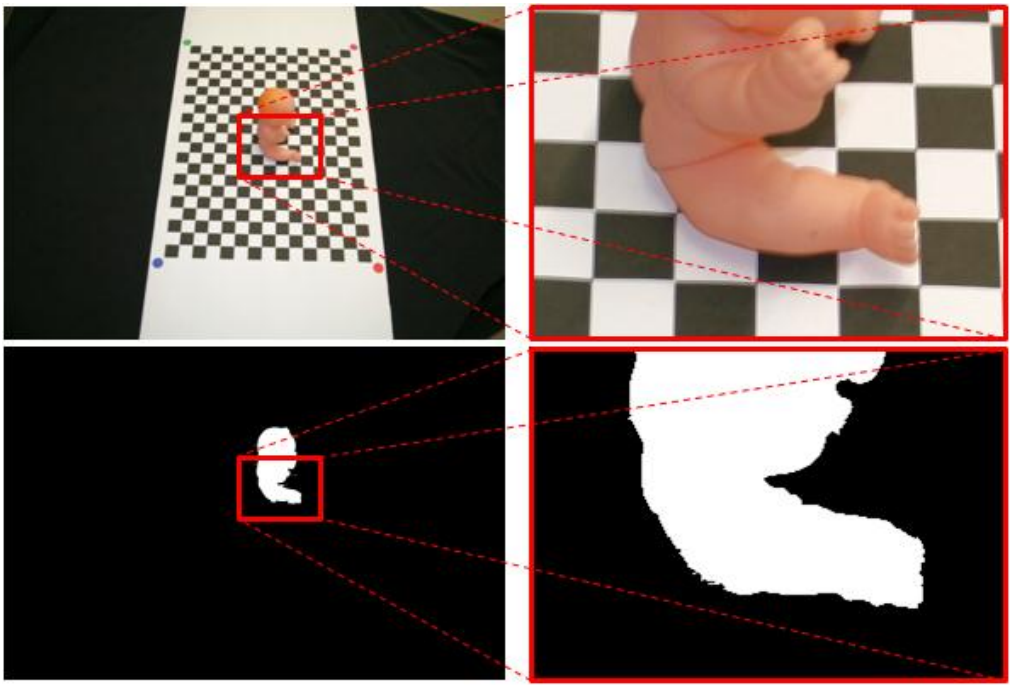

Fig. 5.43 – Effects of shadows in the image segmentation process based on skin colour: original image (on the top) and the correspondent silhouette image (on the bottom).

Fig. 5.44 – Combination of the acquired image with its silhouette image in order to facilitate the detection of the coloured calibration circles.

Table 5.5 – Camera's intrinsic parameters obtained for the plastic baby toy object.

Intrinsic Parameters		
Focal distance (pixel-based unit)	f_x	1924.234
	f_y	1908.741
Principal point (pixel)	c_x	980.247
	c_y	631.398
Radial distortion coefficients	k_1	-0.11948
	k_2	0.00691
Tangential distortion coefficients	p_1	0.00703
	p_2	-0.00236

Camera calibration

In all images of the first sequence, the 280 (20×14) chessboard vertices were successfully extracted and matched. Table 5.5 shows the camera's intrinsic parameters obtained from this image sequence.

Since the object to be reconstructed had some superficial areas with colours close to red or pink, those areas could be confused with the red or pink calibration circles. A simple solution was to allow the calibration procedure to remove the object of interest from the images using also the silhouettes, whenever these were available, Fig. 5.44.

The four outer circles were not successfully detected in 2 of the 11 acquired images for the second sequence due to the projection distortion introduced by a small angle between the camera's line of sight and the calibration pattern. Therefore, those images were not considered for the next steps for 3D reconstruction.

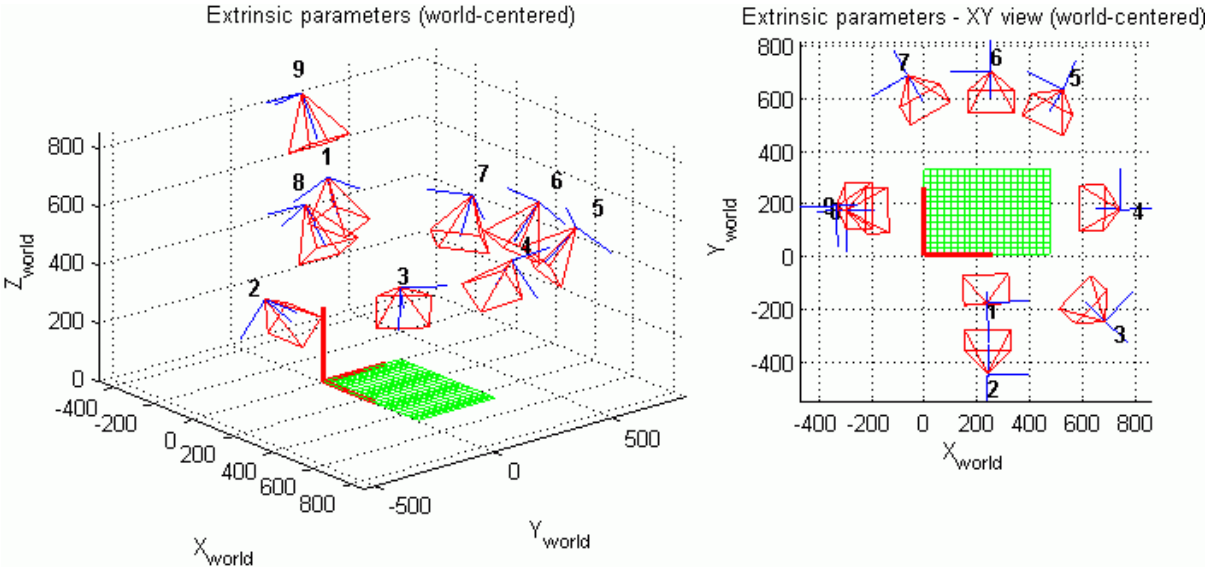

Fig. 5.45 – 3D representation of the camera's extrinsic parameters obtained with the second image sequence of the plastic baby toy object. World 3D axes (in red) are located on the bottom-left vertex of the chessboard (the green grid). In both graphics, the scale is in mm.

A graphical 3D representation of the camera's extrinsic parameters, i.e., the camera's pose, obtained for each of the 9 remainder images can be observed in Fig. 5.45. Average reprojection error for the second image sequence was of $e_{avg} = (1.5527, 1.8262)$ pixels and with a standard deviation of $e_{std} = (2.2757, 2.0522)$ pixels.

When compared to the Rubik cube object, the increase in the reprojection error for the Y-axis was because the images were acquired from viewpoints not equality distanced from one another in the 3D space.

Volumetric reconstruction

The determined initial bounding box provided the following results for the octree root node 3D coordinates:

- $\min_X = 170;\ \max_X = 320;$
- $\min_Y = 50;\ \max_Y = 204;$
- $\min_Z = 0;\ \max_Z = 149.$

The final value obtained for $\max_Z$ required five iterations to decrease its initial value, determined by the calibration parameters using Equation (4.24), Fig. 5.46.

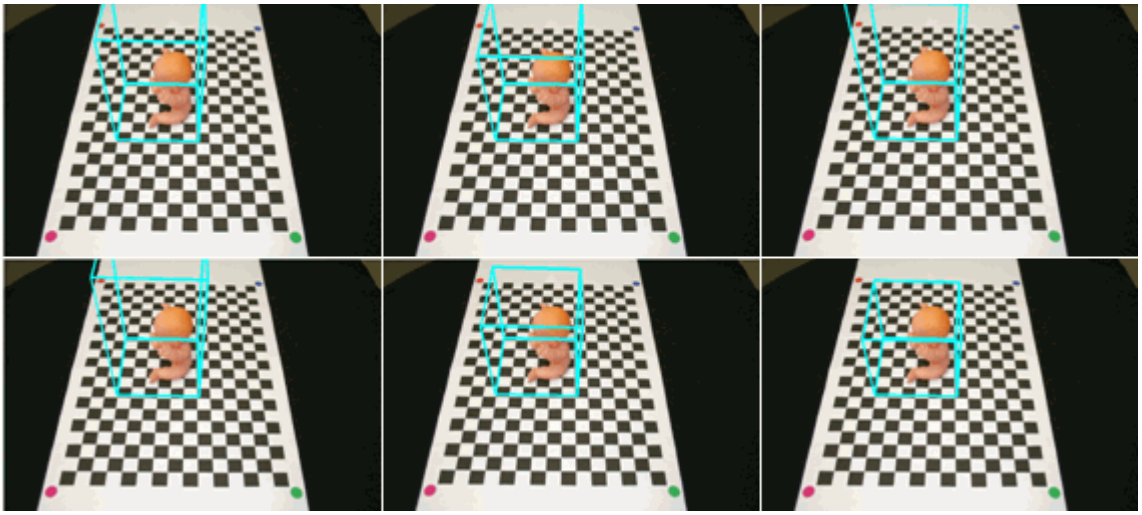

Fig. 5.46 – From left to right, from top to bottom: back-projections of the five necessary iterations to estimate the initial bounding volume for the baby toy object by decreasing the maximum height of the volume.

Fig. 5.47 – Four different views of the obtained 3D models for the baby toy object using the volumetric reconstruction method (parameters: number of iterations equal to 7; only silhouettes): using rectangular voxel projection (on the top) and using exact voxel projection (on the bottom).

The obtained results were confirmed not only by the object's real measures, but also from the measures of the 3D reconstructed model of the baby toy started with an initial voxel volume of $300 \times 200 \times 200$ mm^3, Fig. 5.47:

- $\min_X = 171.09$; $\max_X = 297.65$;
- $\min_Y = 101.56$; $\max_Y = 190.62$;
- $\min_Z = 0$; $\max_Z = 148.44$.

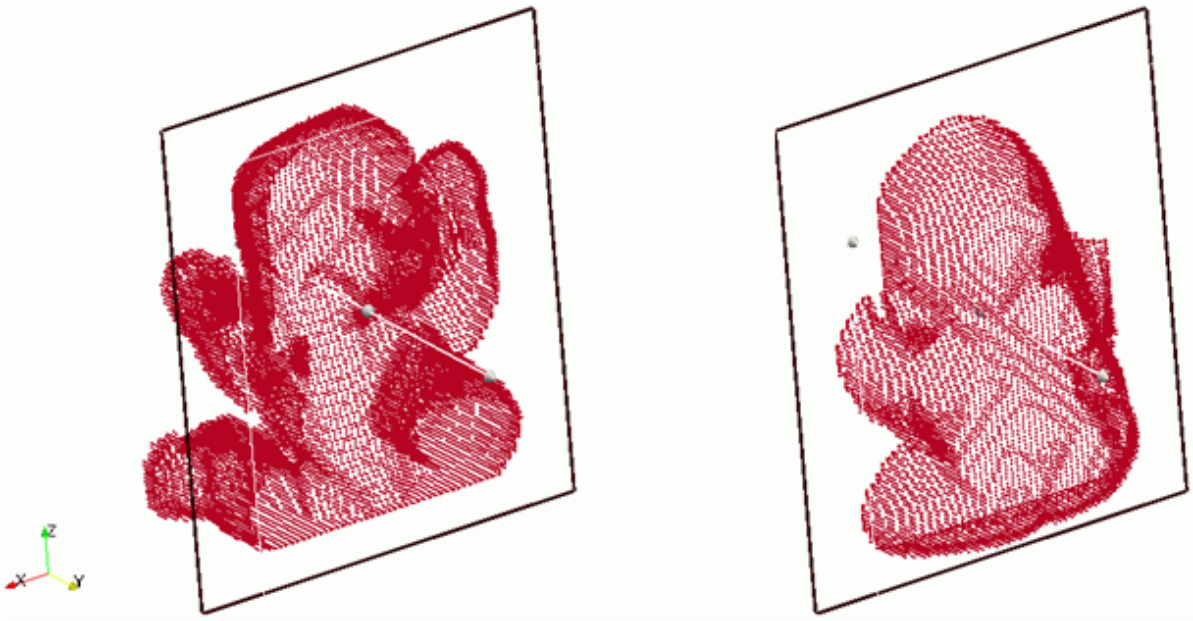

Fig. 5.48 – Surface points of the 3D model obtained for the baby toy object clipped by a plane normal to the world Y-axis.

Fig. 5.47 shows four viewpoints of the two 3D reconstructed models using the volumetric method only based on silhouettes, differing on the used projection criterion of the voxels: rectangular and exact. The 3D visual hull built with the rectangular projection was composed by a total of 54985 voxels and required a processing time of 14 minutes and 7 seconds. On the other hand, the visual hull reconstructed by using exact projection was composed by a total of 42078 voxels and required a processing time of 17 min and 38 seconds. Fig. 5.49 compares the evolution of the visual hull for both projection criteria.

From the 9 images calibrated, only 6 were used to build the 3D models, and the remaining were used to validate the results obtained using photo-consistency, whose results are discussed later. Some voxels of the reconstructed 3D models are coloured black because they were not visible from any of the acquired images. However, they belong to the 3D model, has can be seen in Fig. 5.48.

The surface in Fig. 5.48 was constructed from the 3D model built with exact projection, by subdividing all voxels from lower levels on the octree data structure: the initial 42078 were subdivided into 106429 equal sized voxels, which in turn only the surface faces are maintained and then converted into a polygonal mesh with a total of 59064 triangles.

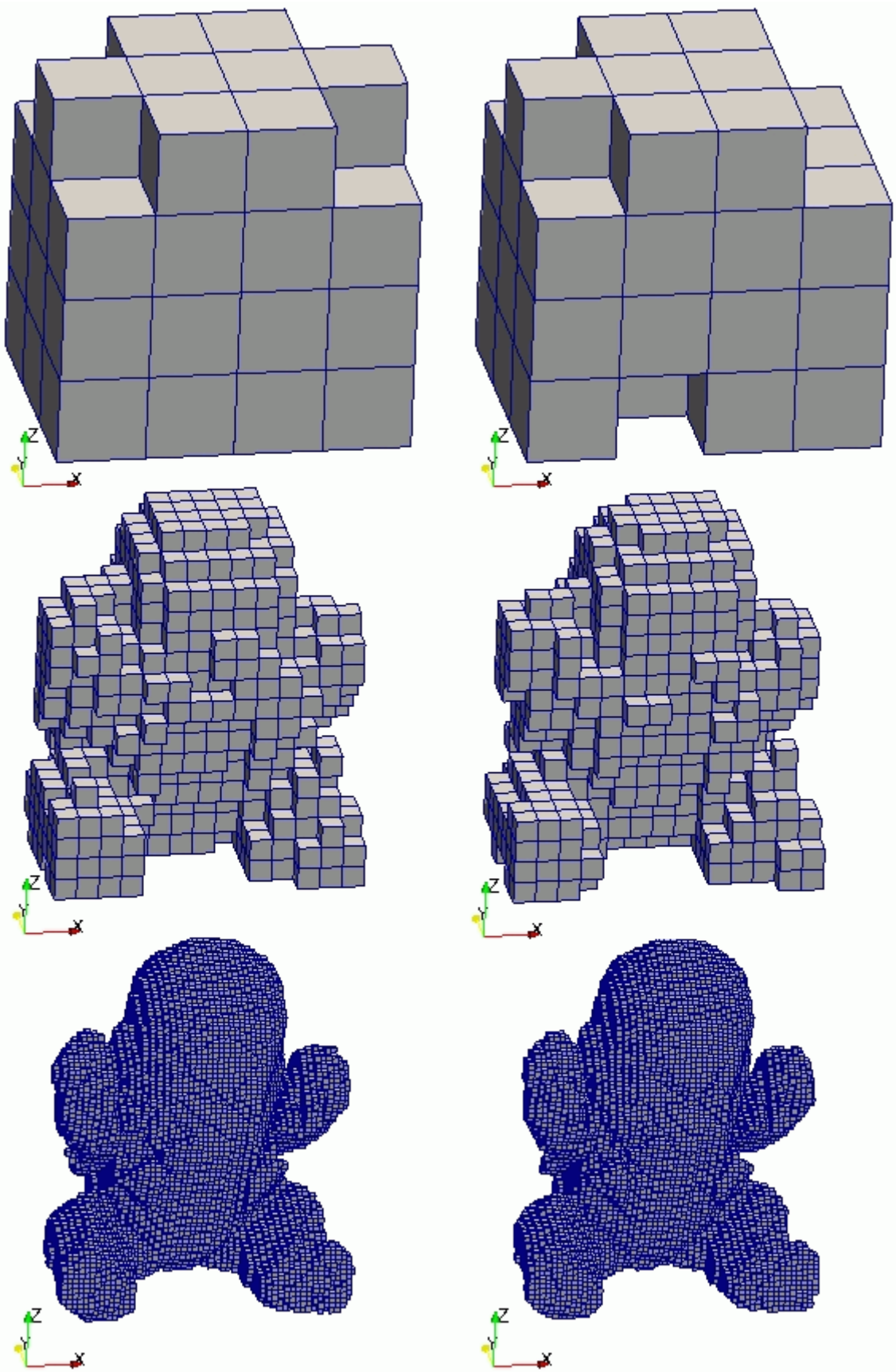

Fig. 5.49 – The 3D models obtained for the baby toy object, using rectangular (left) and exact (right) voxel projection. From top to bottom: octree refinement level of 2, 4 and 6 levels, respectively.

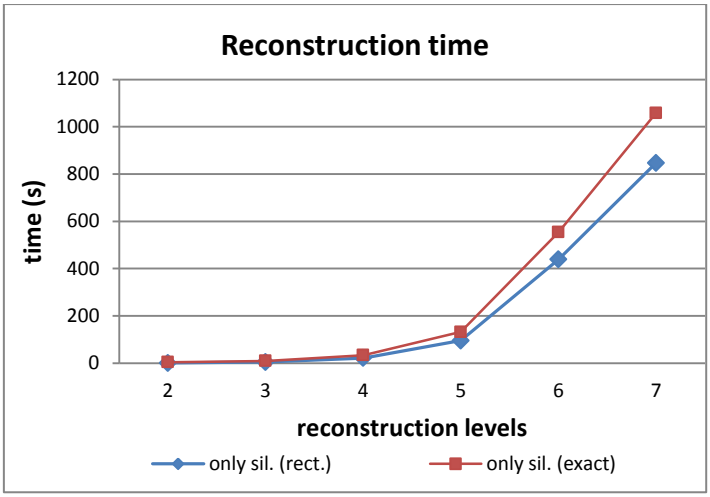

Fig. 5.50 – Total amount of time required to build the visual hull of the baby toy object using rectangular and exact projections.

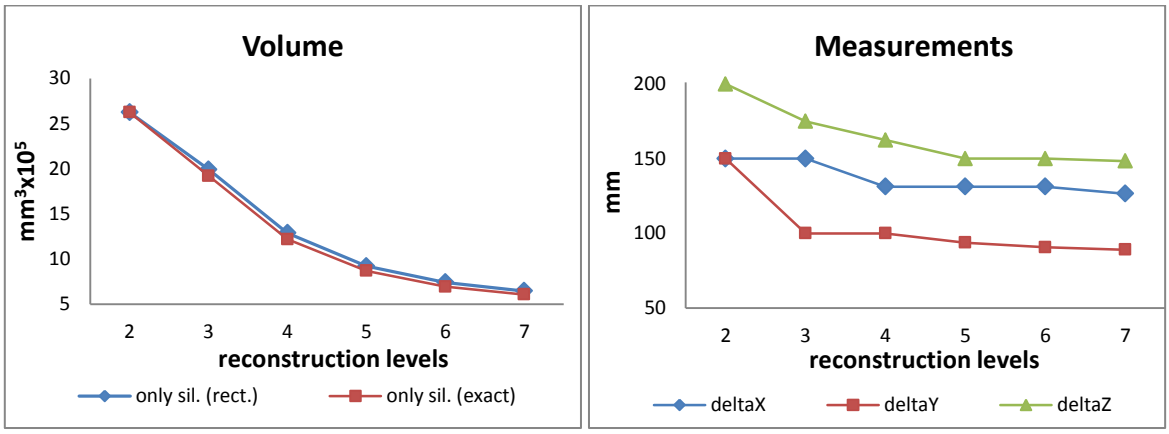

Fig. 5.51 – Measurements performed on the reconstructed 3D models of the plastic baby toy.

3D model assessment

Fig. 5.50 show one graph regarding the computational time required to determine the visual hull for the plastic baby toy, using different levels of refinement. In order to build a textured 3D model with 6 levels of refinement, an extra 51 seconds were required when using rectangular projection, and an extra 34 minutes and 25 seconds if the voxels' footprints are determined with exact projection. Again, the reconstruction time increased exponentially with the octrees final level L.

Fig. 5.51 shows two graphs that allow the comparison among measures of the obtained 3D models for the plastic baby toy. Again, the volume of the obtained 3D models dropped very fast. However, the boundaries of the 3D model dropped even faster and remain almost constant after 4 iterations.

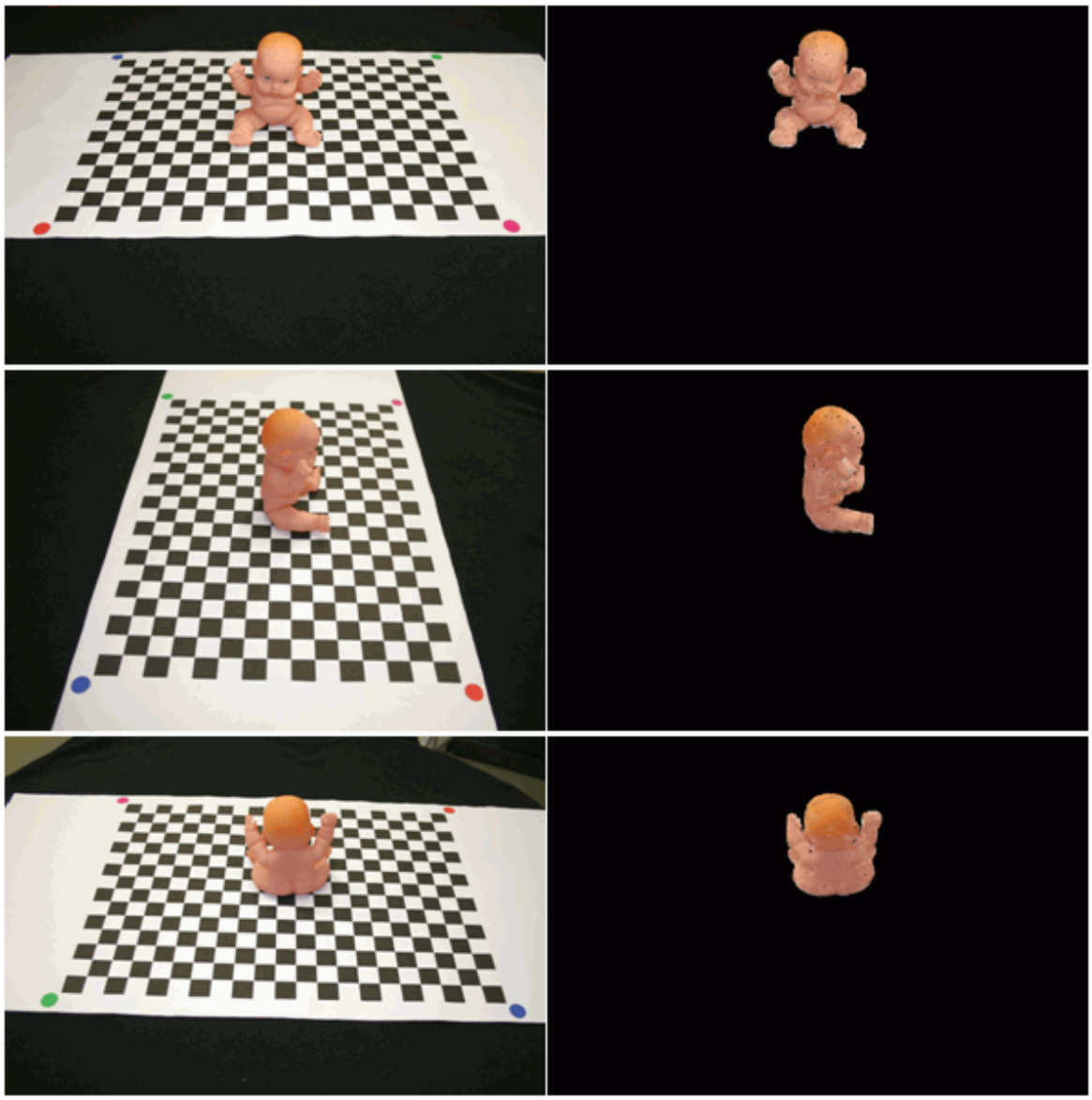

Fig. 5.52 – Images used in the rendering process: evaluation images (on the left) and rendered images (on the right).

The 3 images calibrated, but not included in the reconstruction process, were used to validate shape and texture of the obtained results, by back-projecting the 3D model built with exact projection and with 7 levels of refinement into their image planes.

Fig. 5.52 presents the three evaluation and rendered images used for the baby toy object. Qualitatively, it is noticeable that the resulting reconstructed 3D models offer a good visual quality.

For an objective evaluation of the 3D reconstruction, Table 5.6 summarizes the calculated reprojection errors and colour similarity for the three evaluation images.

Table 5.6 – Reprojection error and colour similarity between the obtained 3D model for the plastic baby toy object and rendered images.

Image	Reprojection error [0-255]		Colour similarity (%)
Top of Fig. 5.52	E_R	10.0772	96.15%
	E_G	9.3953	
	E_B	9.9518	
Middle of Fig. 5.52	E_R	8.6237	96.58%
	E_G	8.2579	
	E_B	9.2552	
Bottom of Fig. 5.52	E_R	13.0438	95.22%
	E_G	11.3998	
	E_B	12.1312	

5.3.3 Human hand model

These test results were performed using a human hand model manufactured by rapid prototyping.

The first image sequence consisted on 27 images of the chessboard calibration pattern. For the second image sequence, 12 images were acquired with the hand model positioned on top of the chessboard pattern and moving the camera around them. All images were acquired on a simple black background and with an image resolution of 1936×1288 pixels.

The image segmentation was successfully performed by thresholding the images' blue channel, Fig. 5.53. Human skin segmentation was not performed on this case because:

1) the hand model was sprayed with a white powder, so its surface was not similar to the human skin;

2) the red squares of the chessboard pattern could be mistaken with skin colour.

Camera calibration

In all images of the first sequence, the 77 (11×7) chessboard vertices were successfully extracted and matched. Table 5.7 shows the camera's intrinsic parameters obtained from this sequence.

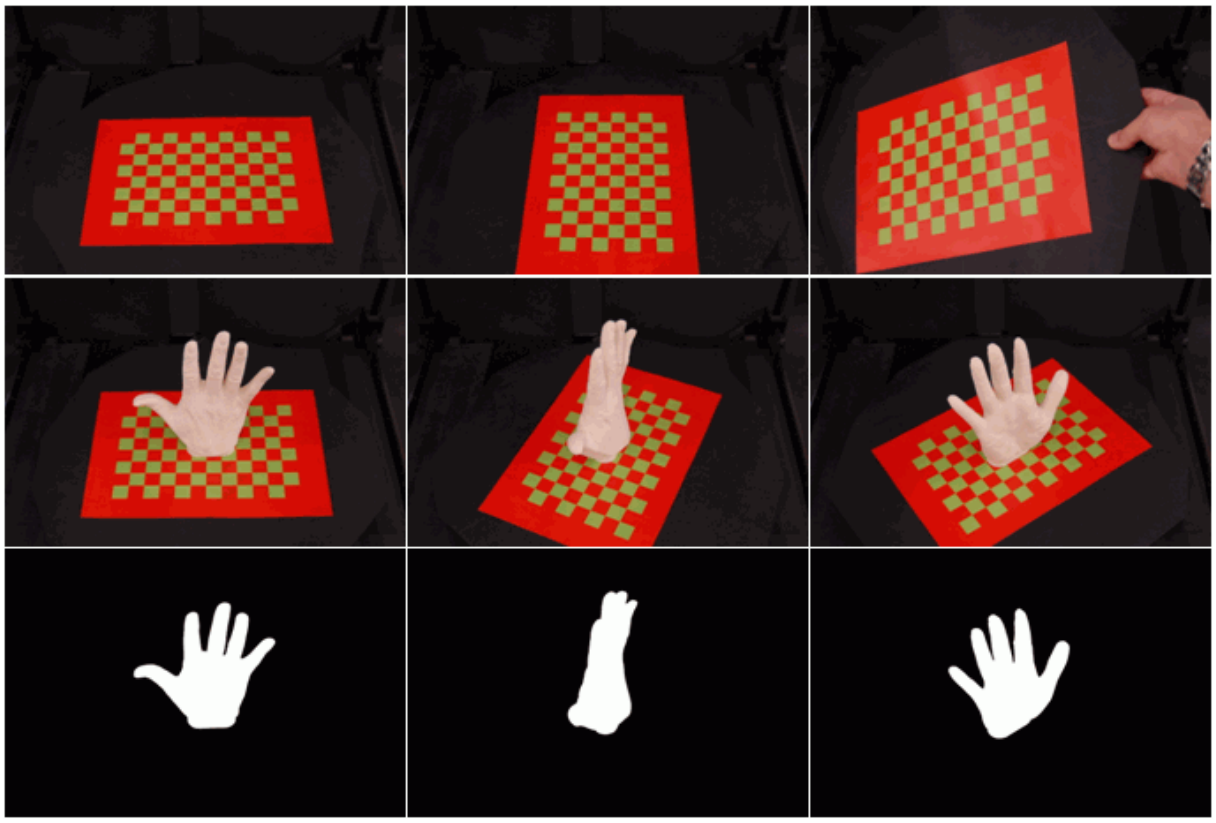

Fig. 5.53 – On the top and on the middle: three examples of the first and second image sequence acquired for the hand model object. On the bottom: obtained silhouettes for the middle images.

Table 5.7 – Camera's intrinsic parameters obtained for the human hand model object.

Intrinsic Parameters		
Focal distance (pixel-based unit)	f_x	2910.295
	f_y	2923.991
Principal point (pixel)	c_x	952.978
	c_y	633.967
Radial distortion coefficients	k_1	-0.11820
	k_2	1.31027
Tangential distortion coefficients	p_1	-0.00681
	p_2	-0.00041

A graphical 3D representation of the camera's extrinsic parameters can be observed in Fig. 5.54. For the corresponding image sequence, the average reprojection error was of $e_{avg} = (0.2009, 0.5422)$ pixels, with a standard deviation of $e_{std} = (0.6428, 0.6406)$ pixels.

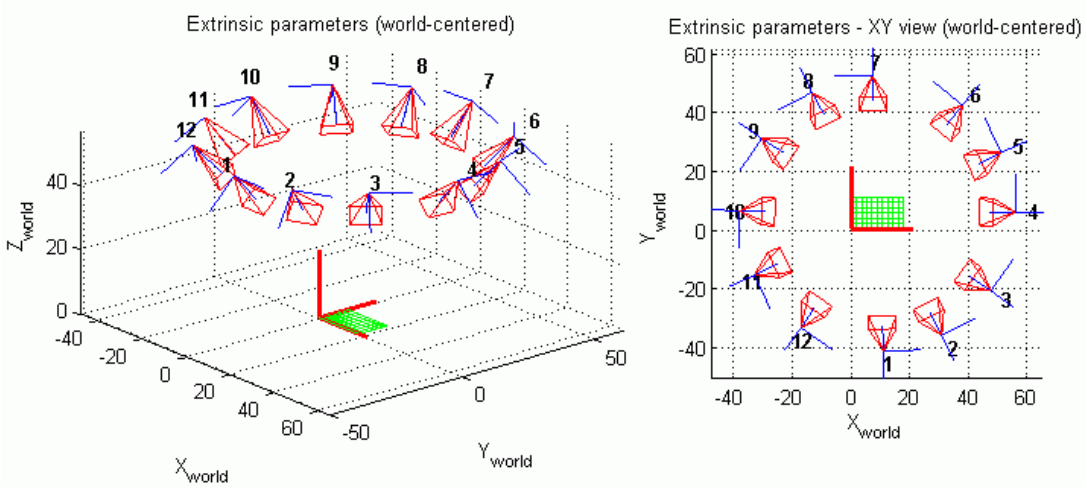

Fig. 5.54 – 3D representation of the camera's extrinsic parameters obtained with the second image sequence of the human hand model object. World 3D axes (in red) are located on the bottom-left vertex of the chessboard (the green grid). In both graphics, the scale is in cm.

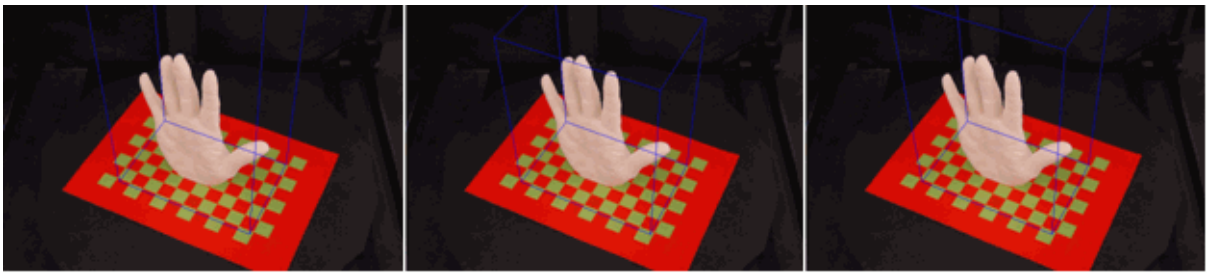

Fig. 5.55 – From left to right: back-projection (blue parallelepiped) of three of a total of five iterations, required to determine the initial bounding volume for the human hand model object, by decreasing the maximum height of the volume.

Volumetric reconstruction

In the 3D reconstruction of the human hand model were used 8 images from the 12 images acquired: images number 1, 3, 4, 6, 7, 8, 10 and 11 in Fig. 5.54. The four images, used to validate the results obtained using photo-consistency, whose results are discussed on the next section, were selected in a way that allows the XY distance between each other to be as high as possible, meaning that they could observe the hand model from four very different viewpoints.

The computation of the initial bounding box 3D coordinates required five iterations to decrease the initial value for the maximum height of the volume, Fig. 5.55, retrieving in the end the following results:

- $\min_X = 1;\ \max_X = 18;$
- $\min_Y = 0;\ \max_Y = 10;$

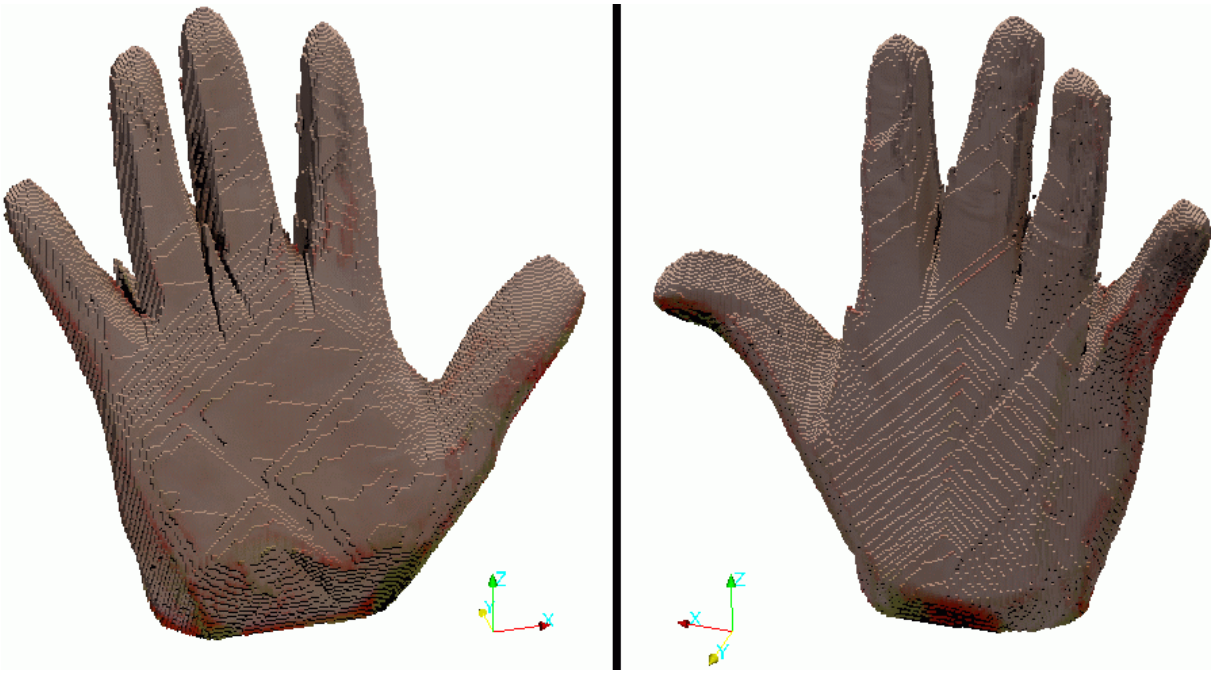

Fig. 5.56 – Two different views of the obtained 3D model for the human hand model using the volumetric reconstruction method and the following parameters: number of iterations equal to 8, only silhouettes and exact voxel projection.

- $\min_Z = 0$; $\max_Z = 22$.

The obtained results are confirmed not only by the object's real measures, but also from the measures of a 3D reconstructed model of the human hand model started with an initial voxel volume of $20 \times 20 \times 20$ cm^3, Fig. 5.56:

- $\min_X = 1.56$; $\max_X = 17.11$;
- $\min_Y = 2.42$; $\max_Y = 7.66$;
- $\min_Z = 0$; $\max_Z = 17.89$.

Fig. 5.56 shows two viewpoints of the 3D reconstructed model using the volumetric method only based on silhouettes and exact voxel projection. The reconstruction was completed after 8 iterations, and included a total of 150862 voxels.

The second reconstruction started with an initial voxel volume of $200 \times 100 \times 200$ mm^3. When the initial volume is nearer to the real bounding volume, for the same refinement level of the octree, the reconstructed 3D model volume is closer to the real one. However, the required reconstruction time increased, Fig. 5.57, since the number of voxels is higher in the second reconstruction.

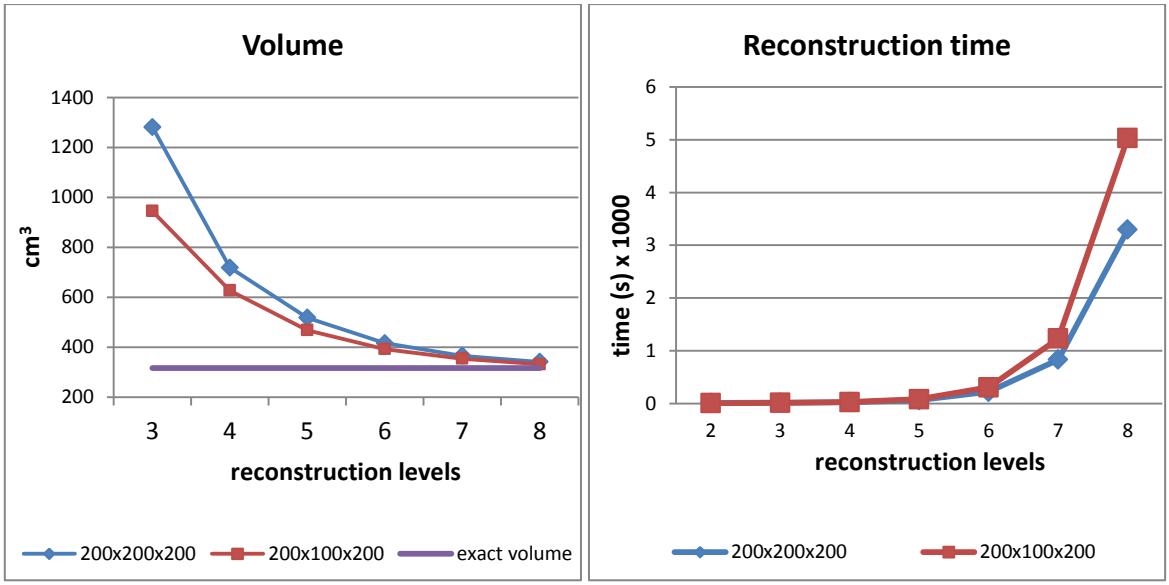

Fig. 5.57 – On the left: volume comparison between the reconstructed 3D models and the real human hand model. On the right: the total amount of time required to reconstruct the human hand models using exact projection. In both graphs, the reconstructions differ on the initial voxel volume.

Table 5.8 – Measures obtained from the real and reconstructed hand model.

Measures	Real model	Reconstructed model
Volume (mm^3)	316848	353235
Width (mm)	159	156
Length (mm)	54.6	52.3
Height (mm)	182	179

3D model assessment

The hand model was manufactured by a rapid prototyping technique, and its real measures could be directly obtained from the correspondent stereolithography CAD (Computer-Aided Design) file. Therefore, a ground-truth was provided and an accurate validation of the obtained 3D models was possible. Table 5.8 compares some measures calculated from the original model and the reconstructed one, Fig. 5.56. Again, the conservative characteristic of the volumetric-based reconstruction methods were evident, since all the measures determined from the 3D model were higher than the real hand model. Nevertheless, those measures were close to the real ones.

Besides the quantitative evaluation of the reconstructed visual hull, some qualitative and also qualitative evaluations were performed on the coloured 3D model. Fig. 5.58 presents the four original images and resultant rendered images concerning the hand model object.

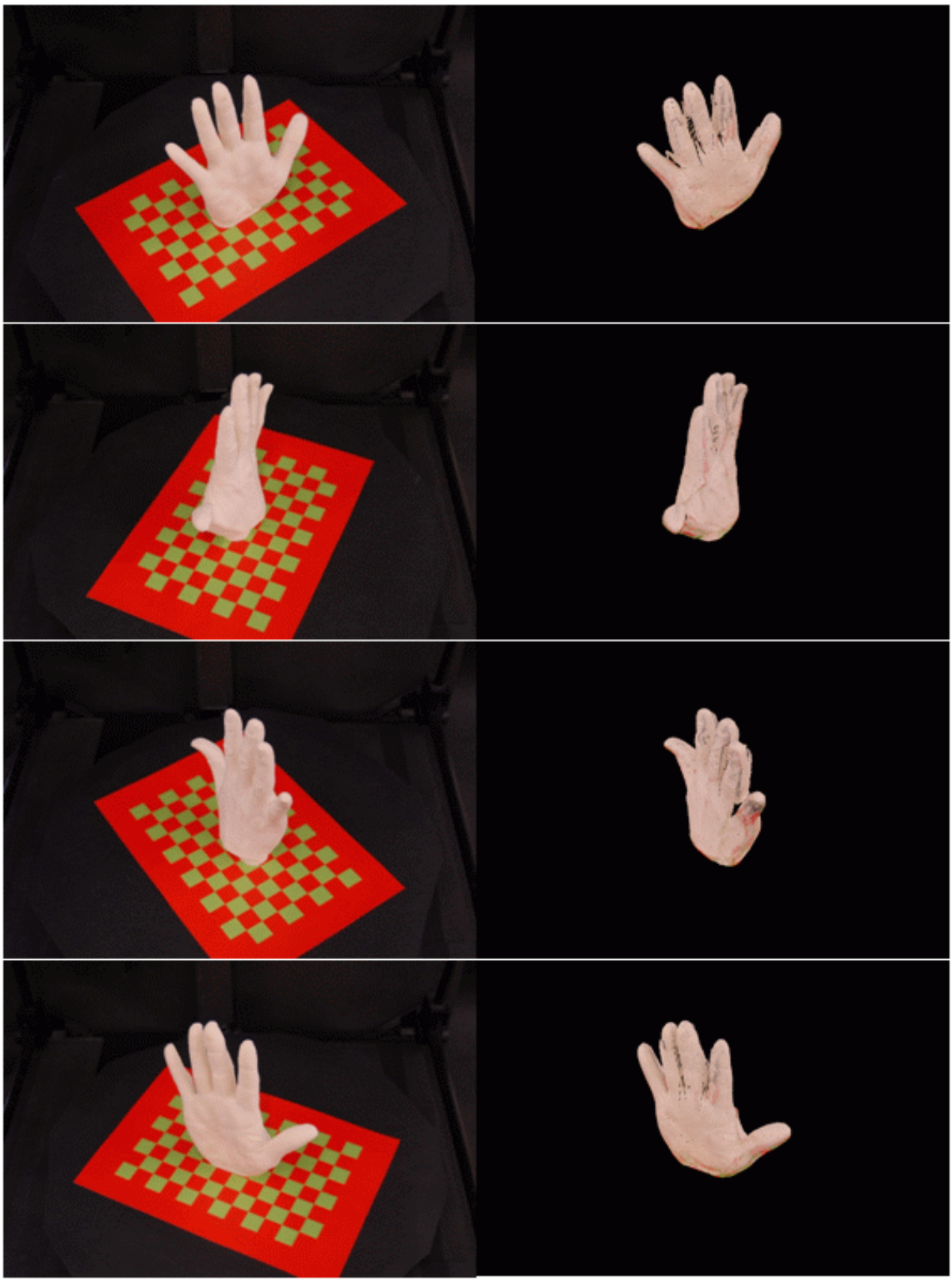

Fig. 5.58 – Images used in the rendering process: original images (on the left) and resultant rendered images (on the right).

Table 5.9 – Reprojection error and colour similarity between the obtained 3D model for the human hand model object and the associated rendered images.

Image	Reprojection error [0-255]		Colour similarity (%)
1st (top) of Fig. 5.58	E_R	19.5984	91.30
	E_G	22.7542	
	E_B	24.2335	
2nd of Fig. 5.58	E_R	17.1741	91.77
	E_G	22.1718	
	E_B	23.6245	
3rd of Fig. 5.58	E_R	17.3320	91.97
	E_G	21.6240	
	E_B	22.4733	
4th (bottom) of Fig. 5.58	E_R	13.5695	94.38
	E_G	14.3874	
	E_B	15.0390	

Qualitatively, it can be noticeable that the resulting reconstructed 3D models offer a good visual quality. However, if the evaluation images were also considered in the reconstruction process, some errors of the obtained 3D model would be avoided; namely, an enhanced reconstruction could be obtained between the fingers, and better colour attributions could be attained, Fig. 5.59.

For an objective evaluation of the 3D reconstruction obtained, Table 5.9 summarizes the calculated reprojection errors and colour similarity relatively to the original images used. Since the real 3D model, from which the hand object was built, was available, the two one-sided Hausdorff distances - from real to reconstructed 3D models and vice-versa - were computed. Fig. 5.60 shows two views of the Hausdorff distance from the real to reconstructed 3D models. In his figure, the distance values are colourized according to a RGB colourmap: red represents the minimum and blue the maximum values. As such, in this case, red means low distance (lower error) and blue high distance (higher error).

Fig. 5.61 shows the same two views for Hausdorff distance from reconstructed to real 3D models. Since the reconstructed model as substantial fewer points, the image is more scattered, but yet visible. Even so, the two measures are not perfectly symmetric, given that the results depend on what mesh was set as sample, Table 5.10.

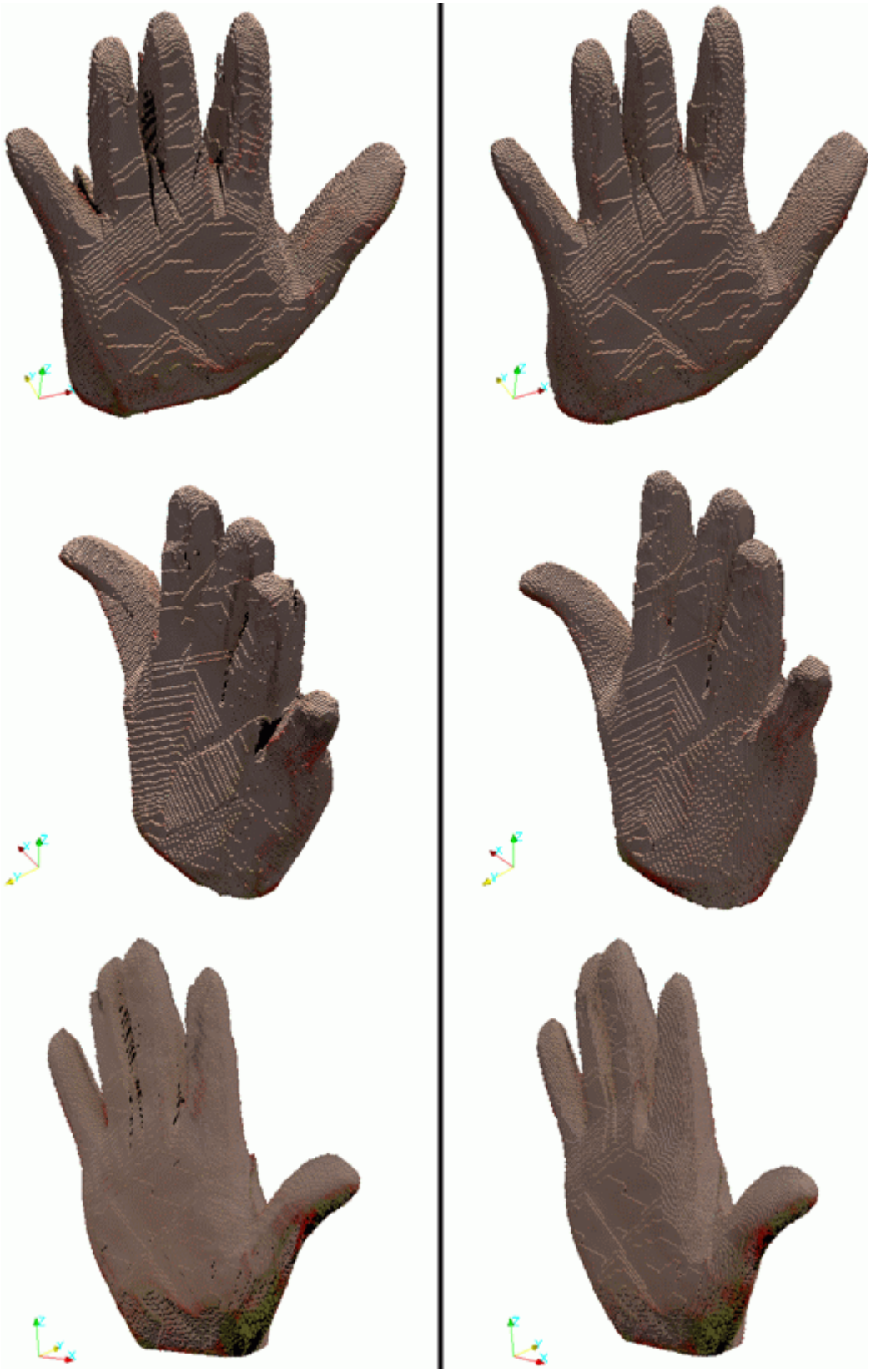

Fig. 5.59 – Three different views of the obtained 3D model for the human hand model when: not all acquired images were used (on the left) and all acquired images were used (on the right) in the reconstruction process.

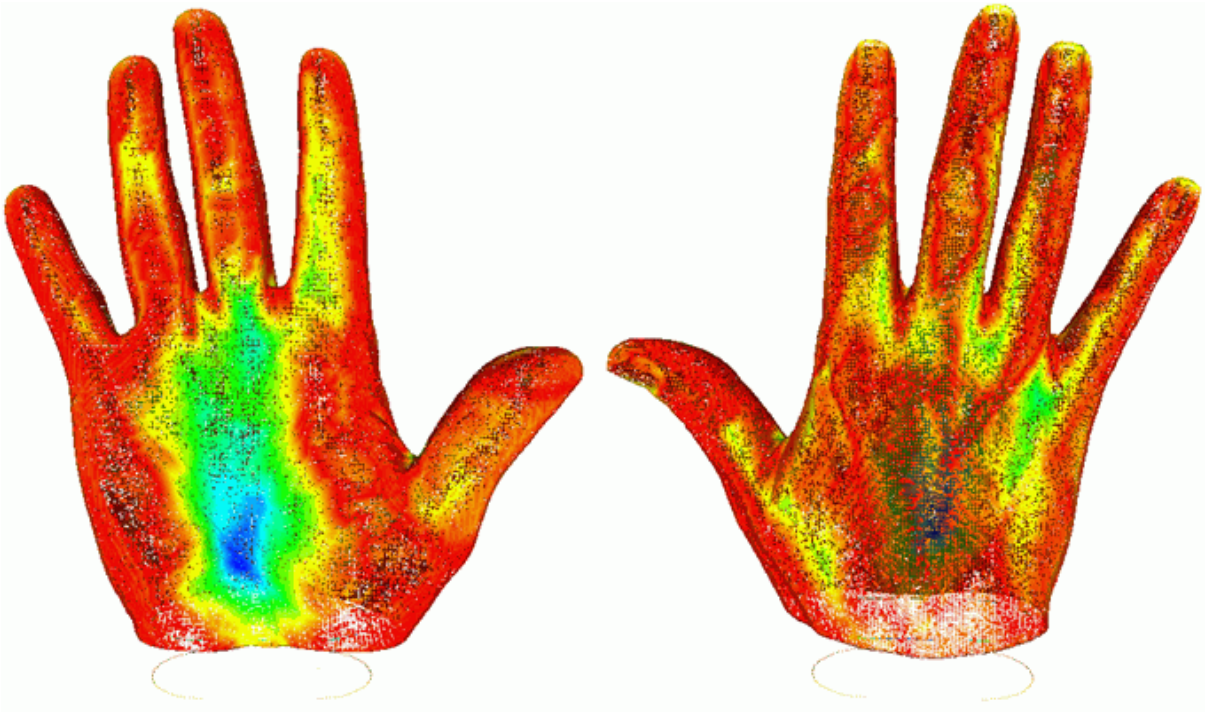

Fig. 5.60 – Two views of the Hausdorff one-side distance from real (with a total of 438516 points) to reconstructed model: red means lower distance and blue higher distance.

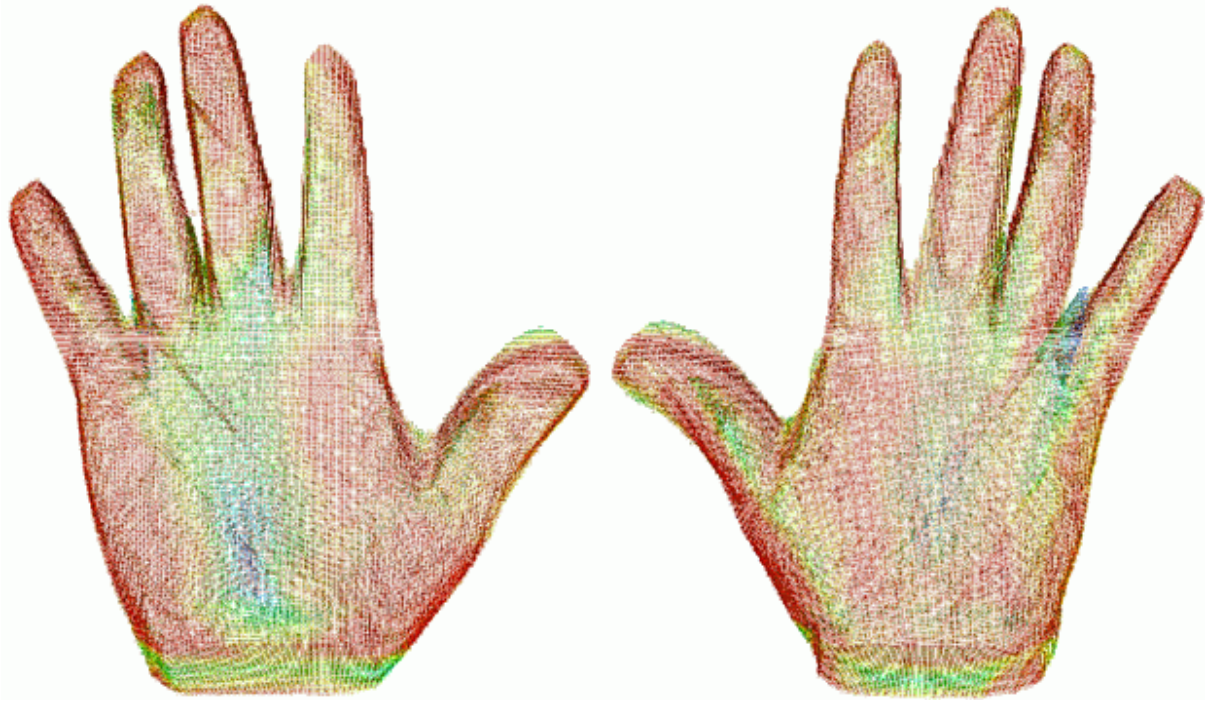

Fig. 5.61 – Two views of the Hausdorff one-side distance from reconstructed (with a total of 98581 points) to real model: red means lower distance and blue higher distance.

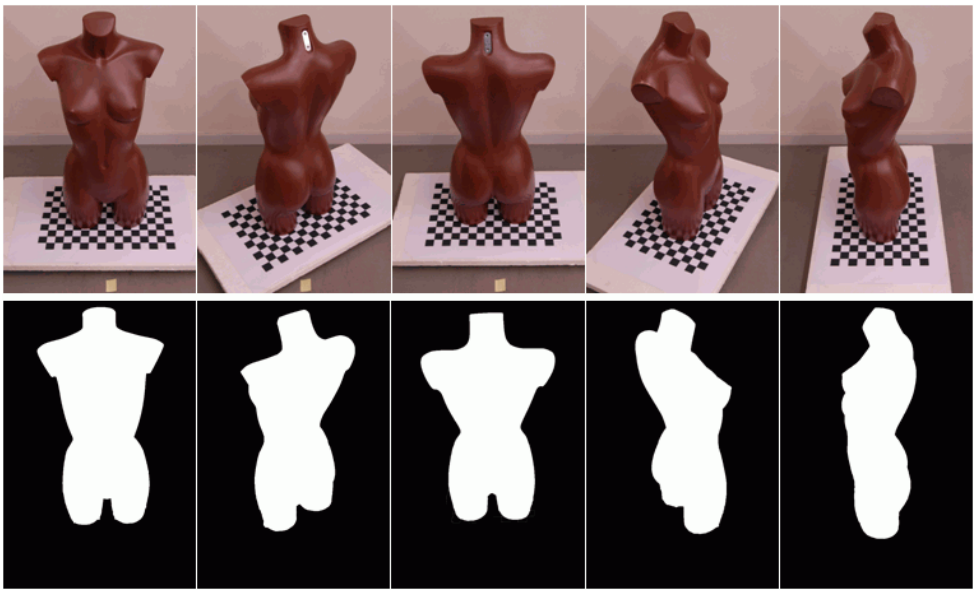

Fig. 5.62 – Original images of the plastic torso model (on the top) and obtained silhouettes (on the bottom).

Table 5.10 – Computed Hausdorff distances between the real and obtained 3D models for the human hand model object (in mm).

Hausdorff distance	Real → Reconstructed	Reconstructed → Real
min	0.000001	0.000029
max	8.969360	9.641207
mean	1.247338	1.432365
RMS	1.966115	2.025856

It can also be observed that the higher errors were found on the frontal middle area and, obviously, where the reconstructed model mismatches the real one: around the fingers and on the bottom of the hand object. The middle area errors were due to the concavity of the object that was not correctly rendered by the reconstruction method.

5.3.4 Human torso model

These test results were performed using a plastic human torso model. First image sequence consisted on 16 images of the chessboard calibration pattern. For the second image sequence, 12 images were acquired with the torso model positioned on top of the chessboard pattern and moving the camera around the torso model. All images were acquired with a resolution of 1296×1936 pixels, Fig. 5.62.

Table 5.11 – Camera's intrinsic parameters obtained for the human torso model object.

Intrinsic Parameters		
Focal distance (pixel-related unit)	f_x	3017.509
	f_y	3025.834
Principal point (pixel)	c_x	618.206
	c_y	1045.190
Radial distortion coefficients	k_1	-0.0882
	k_2	0.2849
Tangential distortion coefficients	p_1	0.0054
	p_2	-0.0015

The torso model as uniform colour, as so its silhouettes were easily extracted using the developed algorithm of skin colour segmentation. However, semi-manual segmentation was also required since some regions of the background were mistaken as skin. The same segmentation steps, used for the Rubik cube object, were followed: edge detection using the Sobel method, followed by image dilation to join line segments and then a flood-filling of the holes. Some results can be observed on Fig. 5.62.

Camera calibration

In all images of the first sequence, the 150 (15×10) chessboard vertices were successfully extracted and matched. Table 5.11 shows the camera's intrinsic parameters obtained from this sequence.

With the second image sequence, the camera's extrinsic parameters were determined and the graphical 3D representation that can be observed in Fig. 5.63 was built. For the referred sequence, the average reprojection error was of $e_{avg} = (0.1068, 0.4788)$ pixels, with a standard deviation of $e_{std} = (0.1562, 0.3959)$ pixels.

Volumetric reconstruction

The computation of the initial bounding box 3D coordinates required two iterations to increase the initial value for the maximum height of the volume, Fig. 5.64, leading at the end to the following results, in cm:

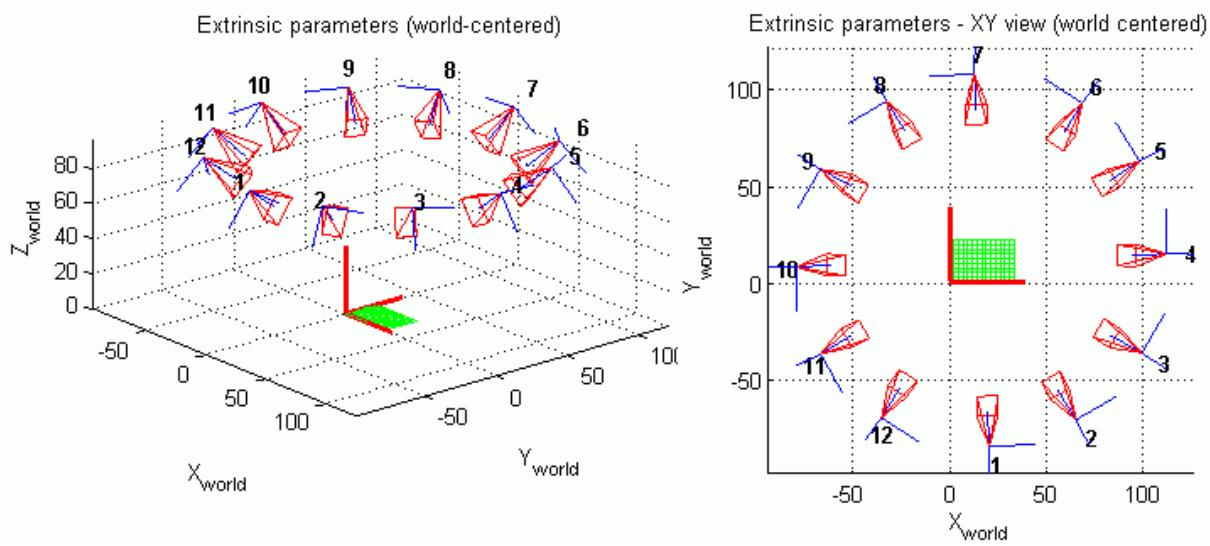

Fig. 5.63 – 3D representation of the camera's extrinsic parameters obtained with the second image sequence of the plastic torso model object. World 3D axes (red) are located on the bottom-left vertex of the chessboard (green grid). In both graphics, the scale is in cm.

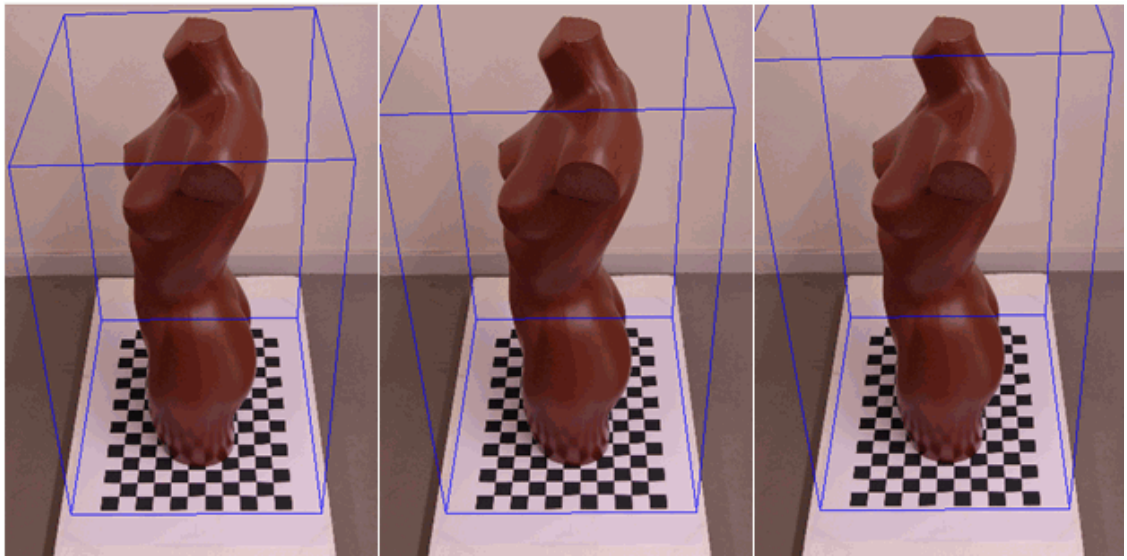

Fig. 5.64 – From left to right: back-projection (blue parallelepiped) of the initial calculated bounding volume for the human torso model and after two iterations by increasing the maximum height of the volume.

- $\min_X = -5\,;\ \max_X = 36\,;$
- $\min_Y = -6\,;\ \max_Y = 28\,;$
- $\min_Z = 0\,;\ \max_Z = 65\,.$

The obtained results were confirmed not only using the real measures of the human torso model, but also from the measures of a 3D reconstructed model started with an initial voxel volume of $40\times40\times70$ cm^3, Fig. 5.65:

- $\min_X = 9.37$; $\max_X = 31.62$;
- $\min_Y = 12.50$; $\max_Y = 20.31$;
- $\min_Z = 0$; $\max_Z = 60.70$.

Fig. 5.65 shows six viewpoints of the 3D reconstructed model for the plastic torso model, using the volumetric method only based on silhouettes, and voxel footprint determined using exact projection.

Since the torso has a reflective surface, the colour attribution of the bottom voxels of the model was highly affected by the black and white squares of the calibration pattern. As all images were acquired from a high viewpoint, the 3D model was worst reconstructed in the lower part of the object.

3D model assessment

Fig. 5.66 shows the original reconstructed 3D model and the one obtained after applying the Laplacian smooth filter. From this figure it can be clearly verified how the results were improved by the smoothing operation. This is especially true for objects with smoother surfaces, like this one.

To obtain a reference 3D model of the torso object, a commercial laser scanner, model *Handyscan* from *DeltaCad* [Handyscan, 2011], was used. This device is a handheld laser scanner with a volumetric accuracy up to 0.05 mm. Fig. 5.67 allows the comparison between the frontal view of the obtained 3D model points using the volumetric-based reconstruction method and the laser scanner.

In order to analyze, both qualitatively as quantitatively, the difference between both 3D models, the Hausdorff distance from the scanned to our model was computed, Fig. 5.68. Again, the distance values are colourized into a RGB colourmap: red meaning lower distance (better), and blue higher distance (worst). As with the hand model, concave areas had larger errors: middle chest and back. Also, higher errors were observed in the lower part of the torso, due to the high plane from where the images were acquired. The highest error distance was of 53.6 mm, and the mean error and standard deviation were equal to 4.98 mm and 6.53 mm, respectively.

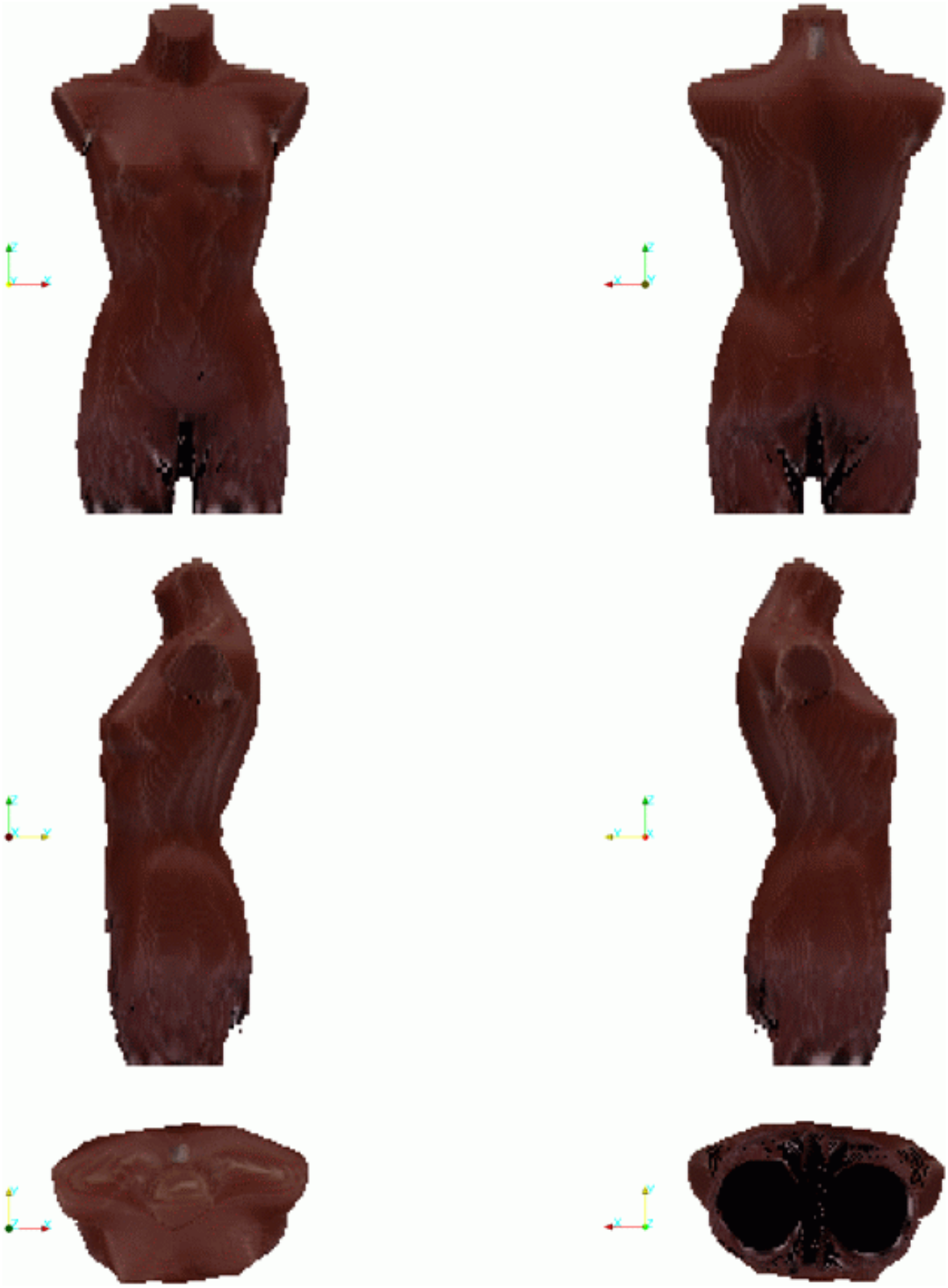

Fig. 5.65 – Six different views of the obtained 3D model for the human torso model using the volumetric-based reconstruction method and the following parameters: number of iterations equal to 7, only silhouettes and exact voxel projection.

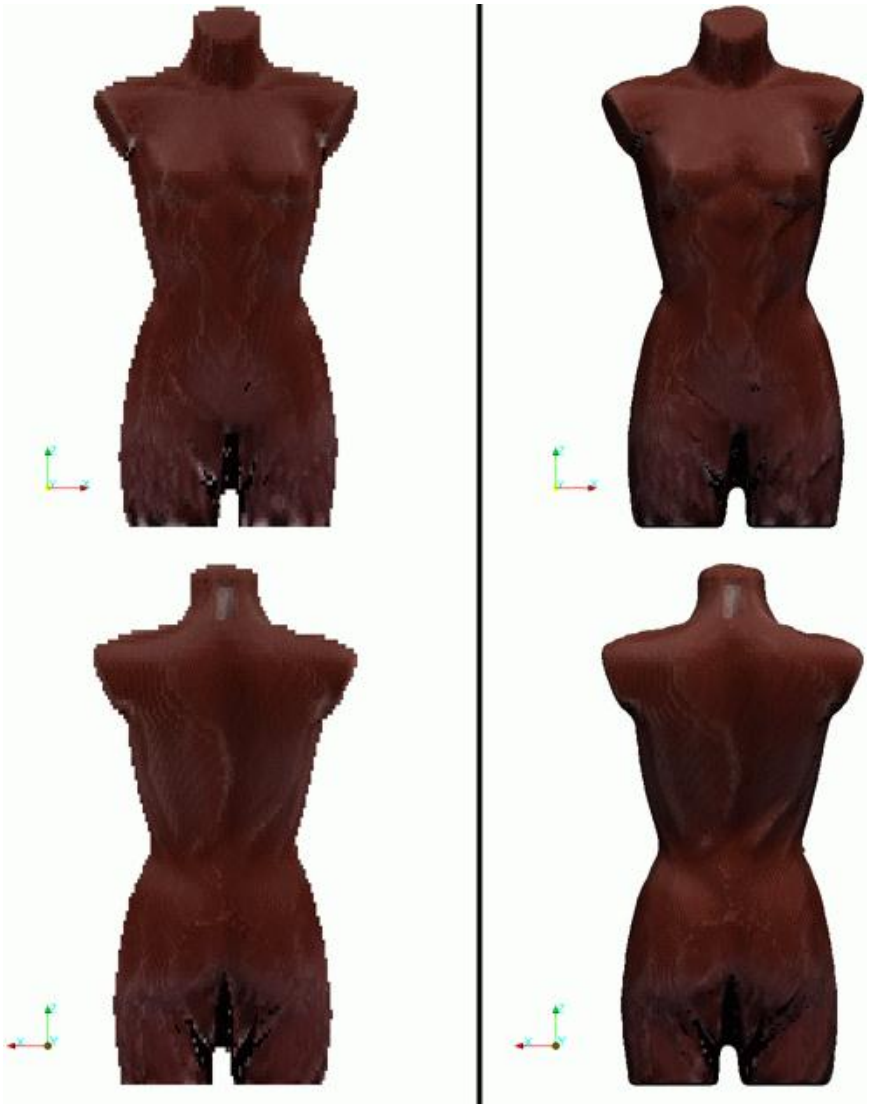

Fig. 5.66 – Original 3D model surface obtained for the human torso model object using the volumetric-based reconstruction method (on the left) and after 500 iterations of the Laplacian smoothing filter (on the right).

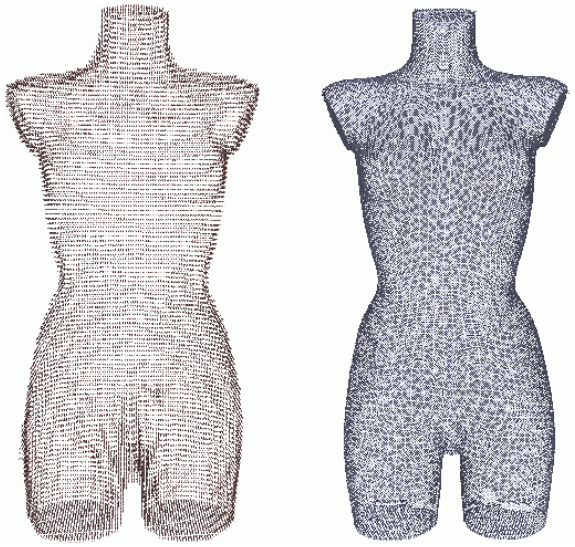

Fig. 5.67 – 3D model points obtained for the human torso model object using the volumetric-based reconstruction method (on the left, 40810 points) and using the *Handyscan* laser scanner (on the right, 65603 points).

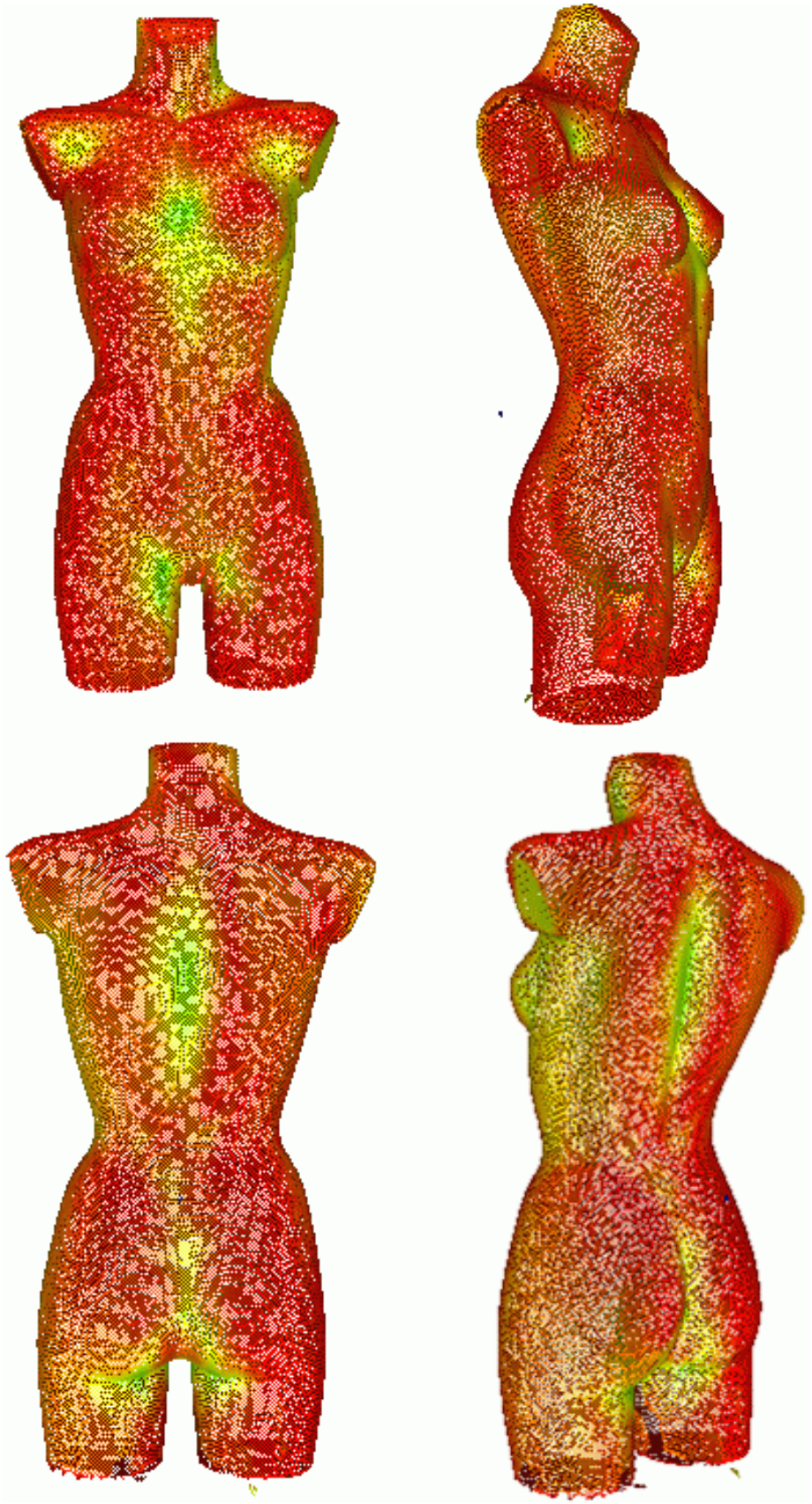

Fig. 5.68 – Colour visualization of the Hausdorff one-side distance from the 3D model built using a laser scanner to the reconstructed 3D model obtained by the volumetric-based method (four views, red means lower distance (better) and blue higher distance (worst)).

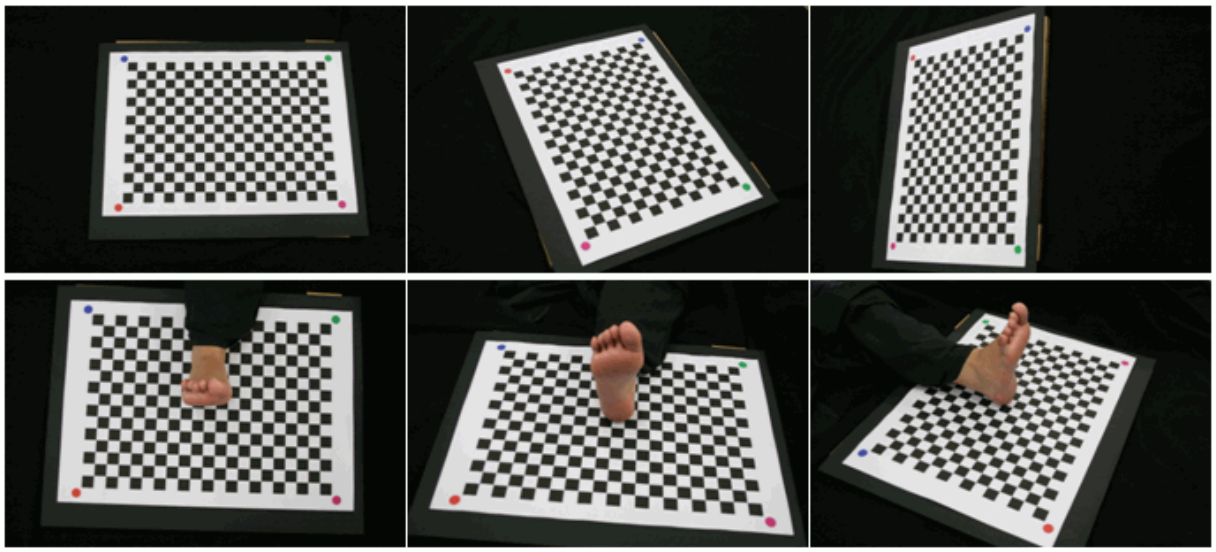

Fig. 5.69 – Three examples of the first (on the top) and second image sequence (on the bottom) acquired for the human foot object.

5.3.5 Human foot

These test results were performed using a real human foot.

The first image sequence consisted on 7 images of the chessboard calibration pattern, and for the second image sequence, 14 images were acquired with the foot positioned on top of the chessboard pattern and moving the camera around it, Fig. 5.69. All images were acquired with a resolution of 1936×1288 pixels.

Camera calibration

In all images of the first sequence, the 280 (20×14) chessboard vertices were successfully extracted and matched. Table 5.12 shows the camera's intrinsic parameters obtained from the first image sequence.

A graphical 3D representation of the camera's extrinsic parameters can be observed in Fig. 5.70. For the same image sequence, the average reprojection error was of $e_{avg} = (0.1663,\ 0.4868)$ pixels, with a standard deviation of $e_{std} = (0.2001,\ 0.4134)$ pixels.

Like with the baby toy object, the increase in the reprojection error for the Y-axis, when compared to the other objects, is due to the lack of homogeneity of dispersion from where the images were acquired.

Table 5.12 – Camera's intrinsic parameters obtained for the human foot object.

Intrinsic Parameters		
Focal distance (pixel related unit)	f_x	2338.420
	f_y	2345.132
Principal point (in pixel)	c_x	975.463
	c_y	692.281
Radial distortion coefficients	k_1	-0.1754
	k_2	1.4218
Tangential distortion coefficients	p_1	0.0059
	p_2	-0.0020

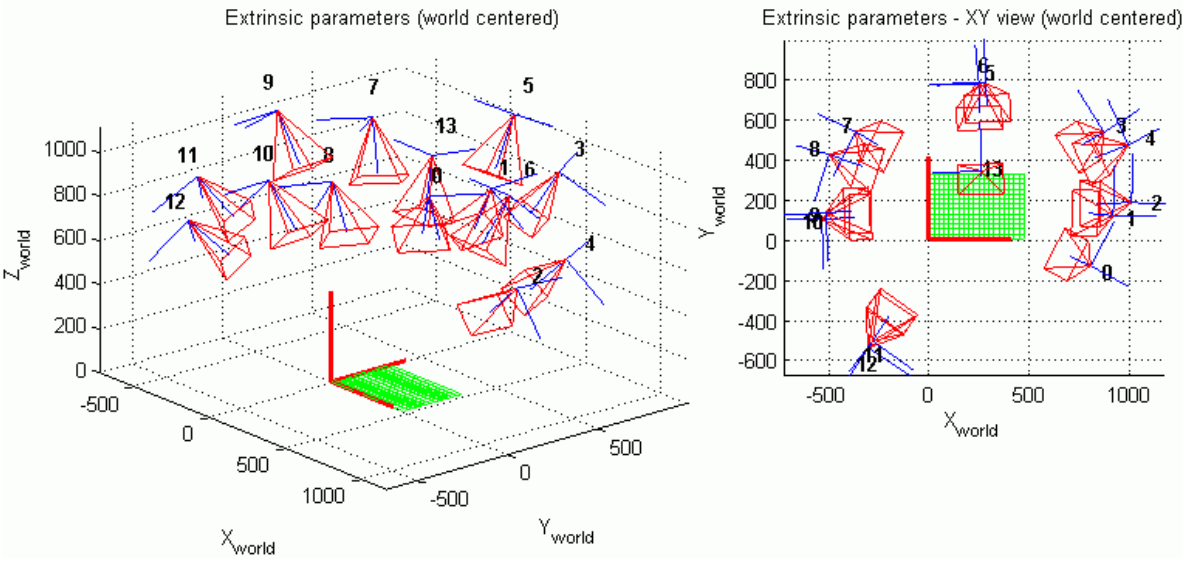

Fig. 5.70 – 3D representation of the camera's extrinsic parameters obtained with the second image sequence of the human foot object. World 3D axes (in red) are located on the bottom-left vertex of the chessboard (the green grid). In both graphics, the scale is in mm.

Image segmentation

The silhouette images were obtained using the algorithm developed of skin colour segmentation and the results were good, but not perfect, Fig. 5.71. However, the volumetric-based reconstruction compensates the defective silhouette extraction because only voxels that lie inside all silhouette images remain in the final 3D model.

The worst results observed were due to the shadows originated by the human foot in regions where it is closer the chessboard pattern, Fig. 5.72.

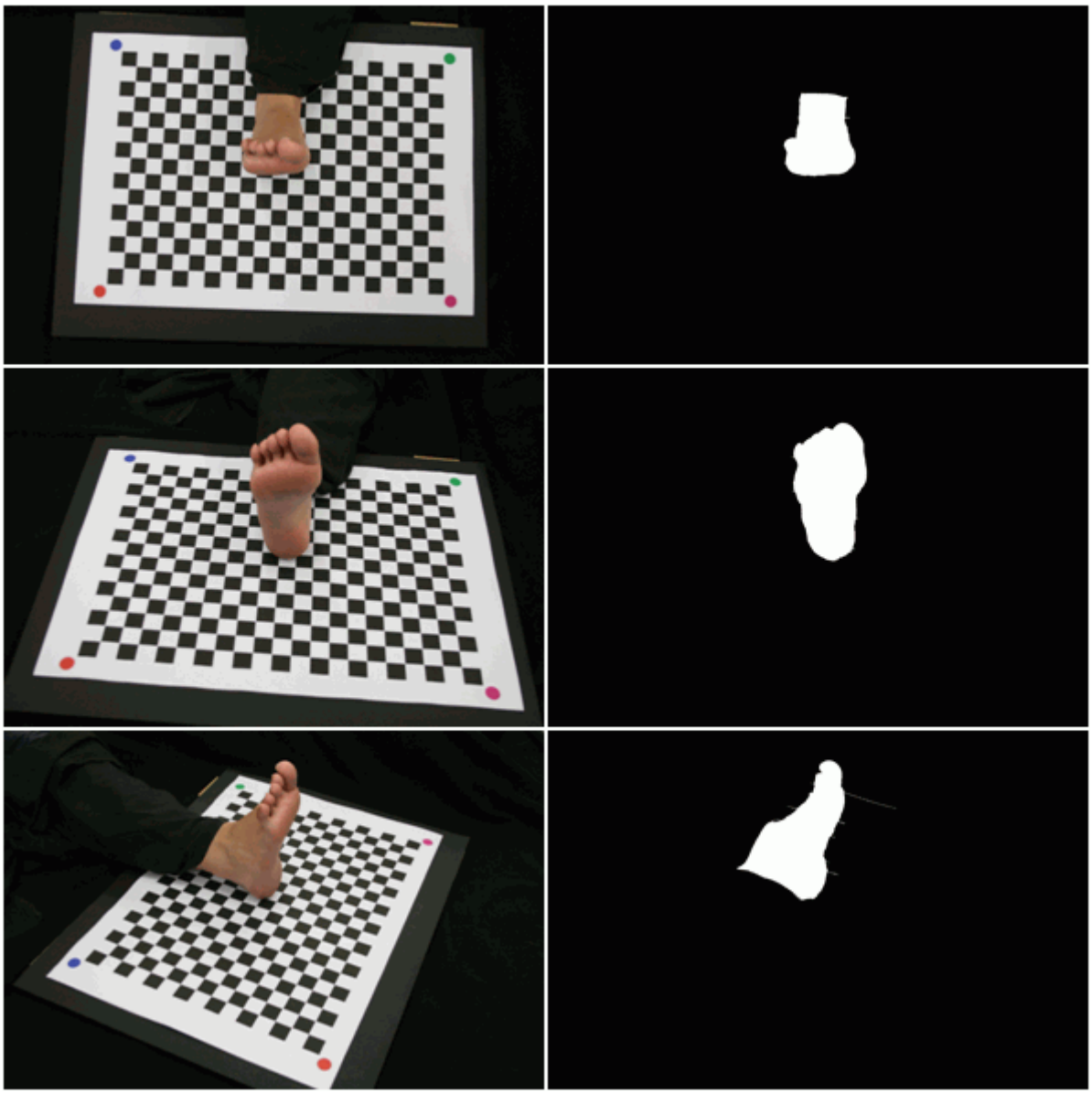

Fig. 5.71 – Original images of the human foot (on the left) and the silhouettes obtained (on the right) using the algorithm of skin colour segmentation.

Volumetric-based reconstruction

The computation of the initial bounding box 3D coordinates required four iterations to decrease the initial value for the maximum height of the volume, leading to the following results, in mm:

- $\min_X = 170$; $\max_X = 319$;
- $\min_Y = 25$; $\max_Y = 286$;
- $\min_Z = 0$; $\max_Z = 290$.

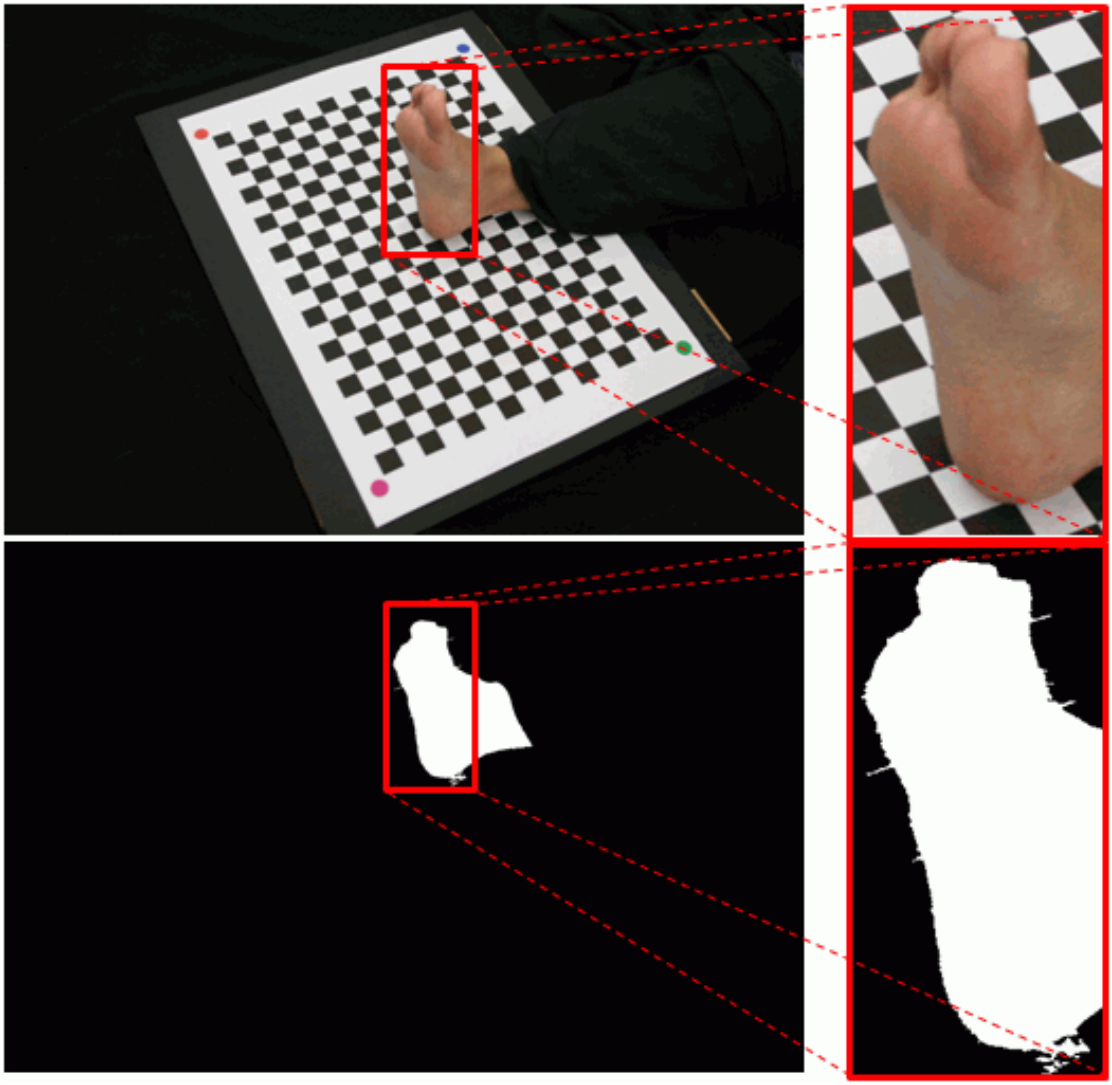

Fig. 5.72 – Effects of shadows in the image segmentation based on skin colour: original (on the top) and obtained silhouette images (on the bottom).

The obtained results were evaluated using the measures of a 3D reconstructed model started with an initial voxel volume of $300 \times 300 \times 300$ mm^3, Fig. 5.73:

- $\min_X = 201.56;\ \max_X = 297.66;$
- $\min_Y = 63.28;\ \max_Y = 196.87;$
- $\min_Z = 0;\ \max_Z = 255.47.$

Fig. 5.73 shows six viewpoints of the 3D reconstructed model for the human foot using the volumetric method only based on silhouettes, and with the voxel footprint determined by exact projection.

Fig. 5.73 – Six different views of the obtained 3D model for a real human foot using the volumetric-based reconstruction method with the following parameters: number of iterations equal to 7, only silhouettes and exact voxel projection.

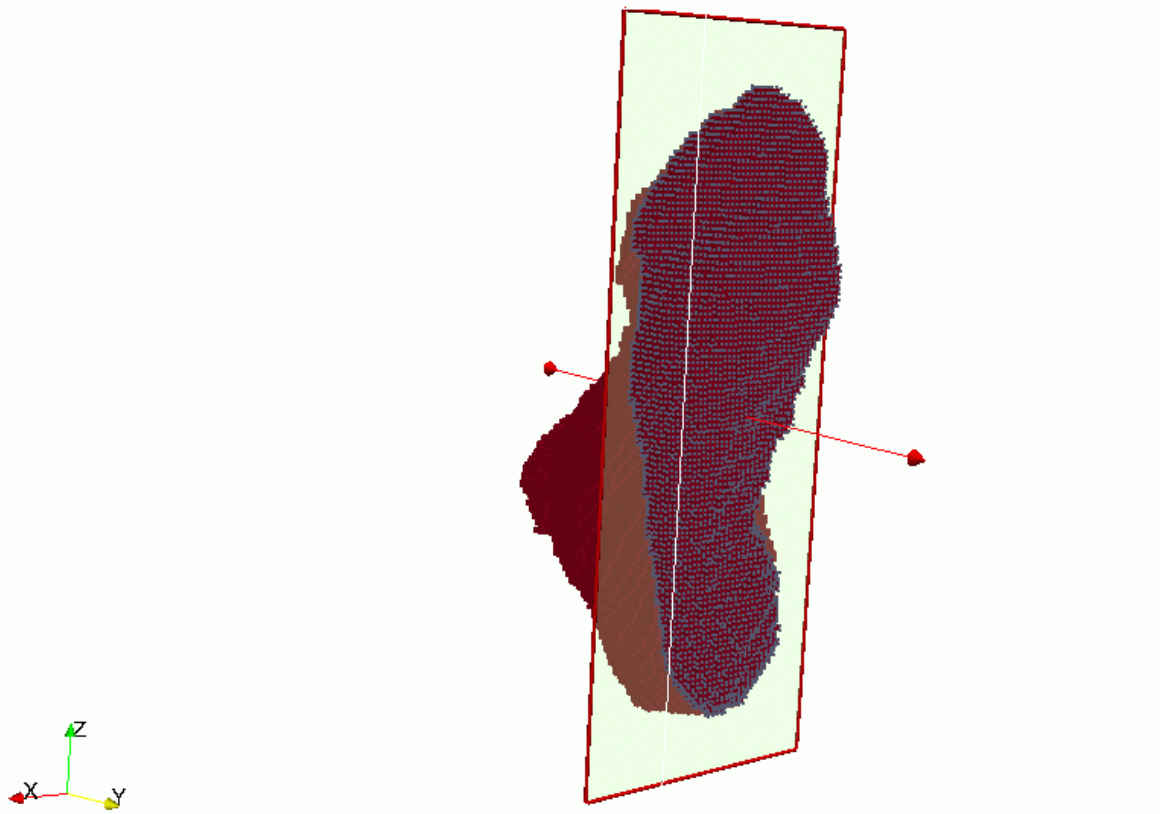

Fig. 5.74 – Surface points (in grey) of the 3D model obtained for the human foot object, clipped by a plane normal to the Y-axis and overlapped with the corresponding 3D volume voxels (in red).

Some voxels had the colour black because they were not visible in any of the acquired images. However, they do belong to the 3D model, as can be confirmed in Fig. 5.74. The 3D visual hull built with the rectangular projection was composed by a total of 40414 voxels and required a processing time of 12 minutes and 57 seconds. On the other hand, the visual hull reconstructed by using exact projection was composed by a total of 30375 voxels and required 17 min and 13 seconds.

3D model assessment

Fig. 5.75 compares the measures obtained from the real foot with the ones calculated from the reconstructed 3D model. Again, the conservative characteristic of the volumetric-based reconstruction method were evident. Nevertheless, the values were close. Fig. 5.76 shows the original reconstructed 3D model and the one obtained after applying the Laplacian smooth filter. As discussed before, it can be clearly verified how the results could be improved with the smoothing operation, which is especially true for objects with smoother surfaces, like this one.

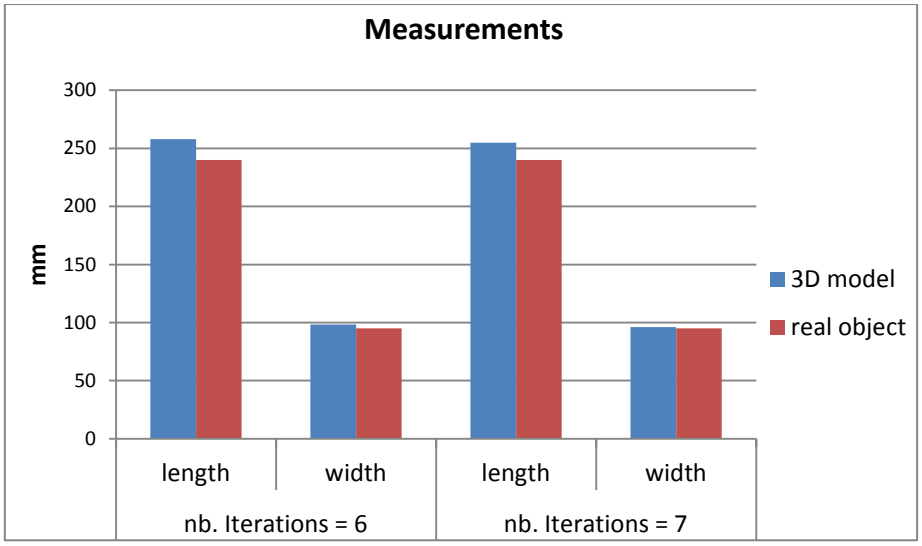

Fig. 5.75 – Measurements obtained from the reconstructed 3D models and from the real human foot.

Fig. 5.76 – Original 3D model surface obtained for the human foot object (on the left) and smoothed 3D model obtained after 1000 iterations of the Laplacian smoothing filter (on the right).

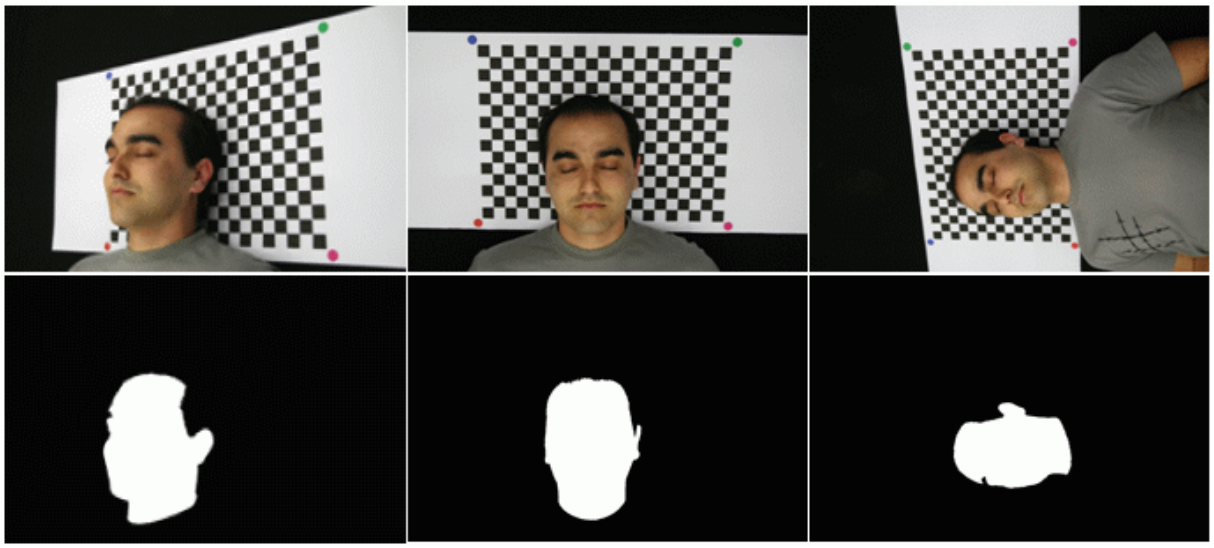

Fig. 5.77 – Original images of the human face (on the top) and silhouettes (on the bottom) obtained using the developed algorithm of skin colour segmentation

5.3.6 Human face

Image acquisition and segmentation

These test results were performed using a real human face. The first image sequence consisted on 6 images of the chessboard calibration pattern. For the second image sequence, 14 images were acquired with the subject's face positioned on top of the chessboard pattern and moving the camera around it, Fig. 5.77. All images were acquired with a resolution of 1936×1288 pixels.

The silhouette images were obtained using the developed algorithm of skin colour segmentation and, again, the results were good, but not perfect, as the subject's eyebrows were not successfully extracted in all images in which they were in the face's rim on the image, Fig. 5.77.

As the volumetric-based reconstruction not compensates for those cases of bad silhouettes, since it takes only one image in which the voxel projects outside the silhouette to be classified as outside, the solution was to manually include the eyebrows on the silhouettes.

Camera calibration

The 280 (20×40) chessboard vertices were successfully extracted and matched in all images of the first sequence. Table 5.13 shows the camera's intrinsic parameters obtained.

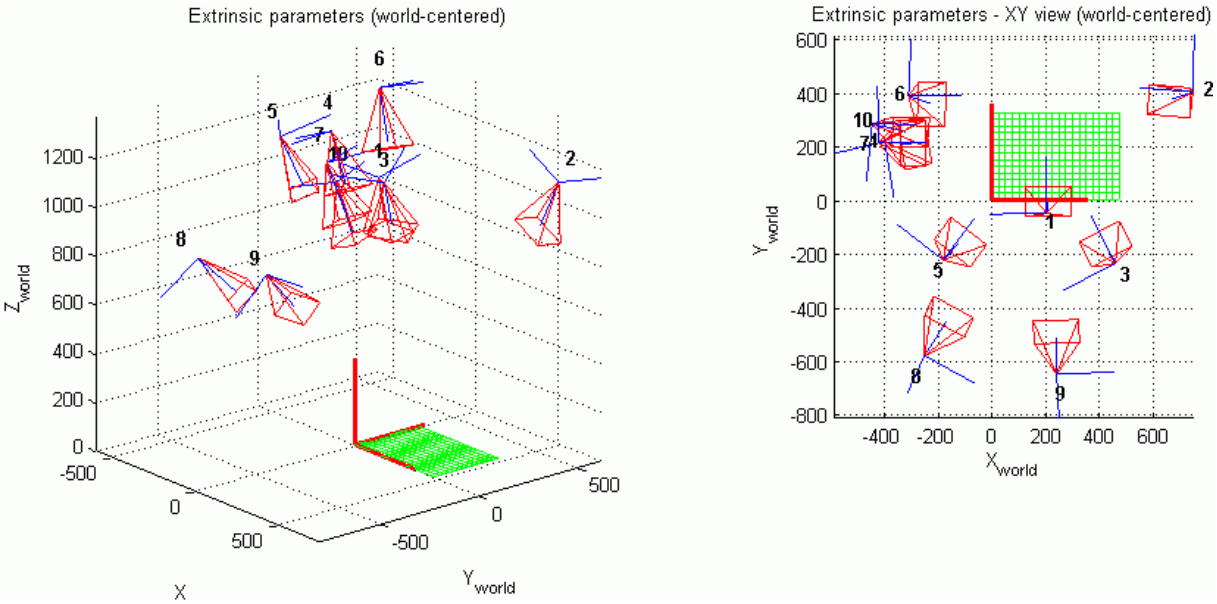

Fig. 5.78 – 3D representation of the camera's extrinsic parameters obtained with the second image sequence of the human face object. World 3D axes (in red) are located on the bottom-left vertex of the chessboard (the green grid). In both graphics the scale is in mm.

Table 5.13 – Camera's intrinsic parameters obtained for the human face.

Intrinsic Parameters		
Focal distance (pixel related unit)	f_x	2719.303
	f_y	2717.580
Principal point (in pixel)	c_x	981.531
	c_y	687.485
Radial distortion coefficients	k_1	-0.1270
	k_2	0.9481
Tangential distortion coefficients	p_1	0.0065
	p_2	-0.0031

From the 14 images acquired for the second sequence, 4 images were discarded as the four outer circles from the calibration pattern were not fully detected in those images.

A graphical 3D representation of the camera's extrinsic parameters for the images of the second sequence can be observed in Fig. 5.78. The average reprojection error was of $e_{avg} = (0.8320, 1.0893)$ pixels, with a standard deviation of $e_{std} = (0.7093, 0.9060)$ pixels.

Volumetric-based reconstruction

The computation of the initial bounding box 3D coordinates required five iterations to decrease the initial value for the maximum height of the volume, leading to the following results, in mm:

- $\min_X = 146$; $\max_X = 528$;
- $\min_Y = 37$; $\max_Y = 478$;
- $\min_Z = 0$; $\max_Z = 282$.

The obtained results were evaluated using the measures of a 3D reconstructed model started with an initial voxel volume of $400 \times 400 \times 400$ mm^3, Fig. 5.79:

- $\min_X = 162.50$; $\max_X = 356.25$;
- $\min_Y = 81.25$; $\max_Y = 362.5$;
- $\min_Z = 0$; $\max_Z = 268.75$.

The 3D model reconstruction was performed using all the acquired images. Fig. 5.79 shows four viewpoints of the 3D reconstructed model for the human face using the volumetric method only based on silhouettes and with the voxel footprint determined by exact projection.

Clearly, the volumetric-based method failed to reconstruct the human face, making even more evident the importance of the viewpoints from which the images used were acquired.

To remove the cusping effect on the centre of the face, images acquired covering all subject's head or at least some acquired from its profile would be necessary. This was impossible, due to the requirement of full visibility of the calibration pattern in the images necessary for camera calibration. A possible solution to overcome this problem could be a calibrated multi-camera system to acquire the necessary images for the developed volumetric-based reconstruction. Also, the deformation of the reconstructed 3D model caused wrong voxel colouration, Fig. 5.80.

The obtained results gave emphasis to the idea that the visual hull is dependent both on the number and position of the viewpoints. In order to try to improve the 3D model, the face's colour information was included on a second reconstruction attempt, by including photo-consistency tests into the reconstruction process.

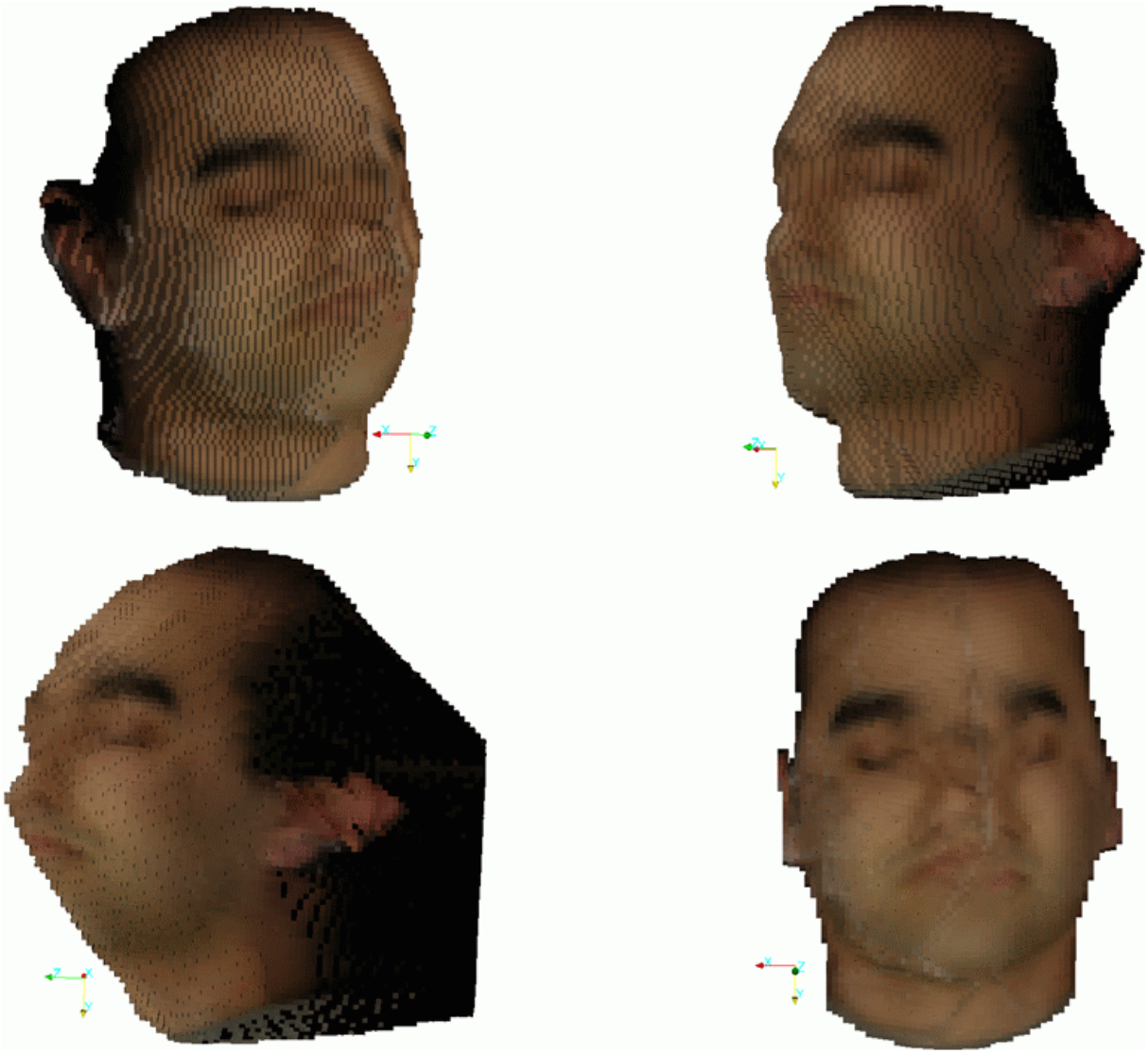

Fig. 5.79 – Four different views of the obtained 3D model for a real human face using the volumetric reconstruction method with the following parameters: number of iterations equal to 7, only silhouettes and exact voxel projection.

Fig. 5.81 allows the comparison between the previous 3D model against the one obtained by using silhouettes and photo-consistency. In this case, the used threshold values were $T_1 = 10$ and $T_2 = 5$ (see Equation (4.29)). It was decided to maintain a lower threshold on the average standard deviation, since the face has lower colour variation. Even so, not significant improvements were achieved, since only 2367 voxels were removed from the initial volume that was composed by a total of 49042 voxels, which means a voxel removal around 5%.

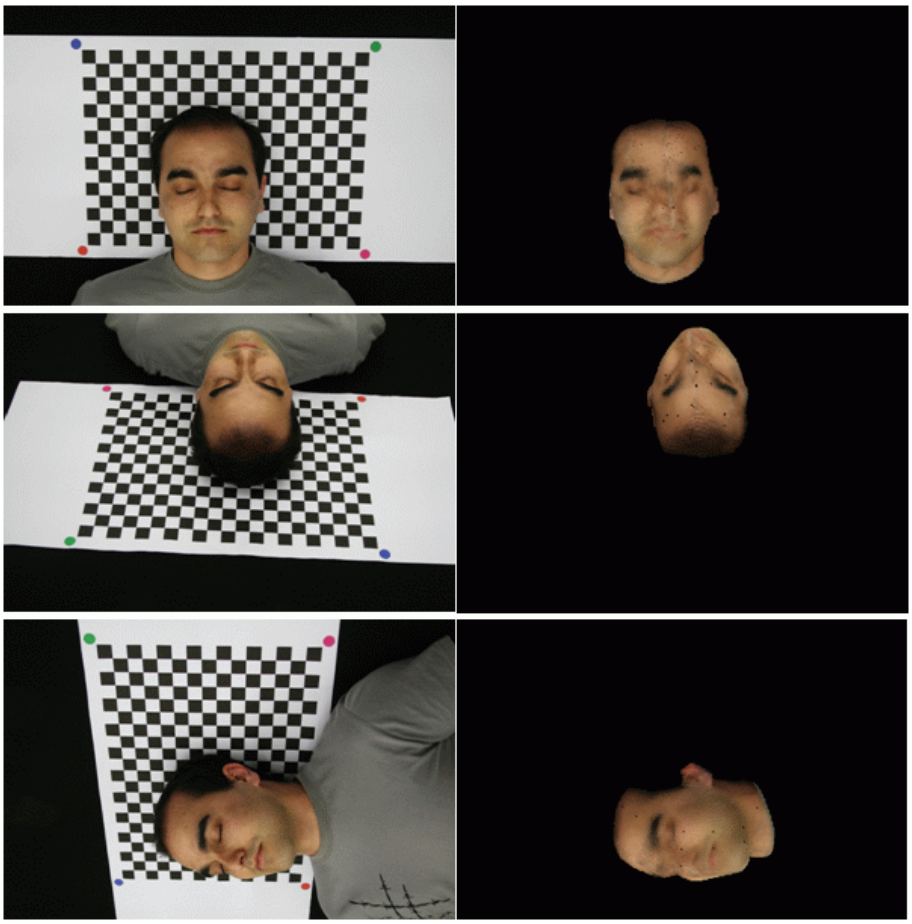

Fig. 5.80 – Three examples of the rendering results: original or evaluation images (on the left) and rendered images (on the right).

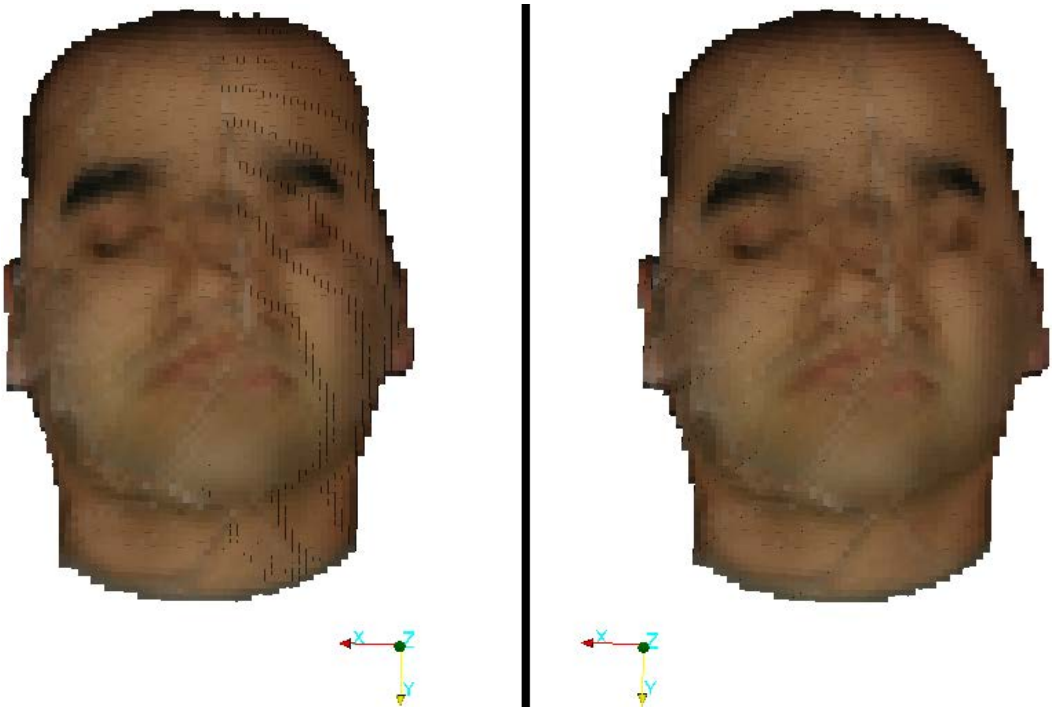

Fig. 5.81 – 3D models built using only silhouettes (on the left) and built using silhouettes and photo-consistency, with the thresholds $T_1 = 10$ and $T_2 = 5$ (on the right).

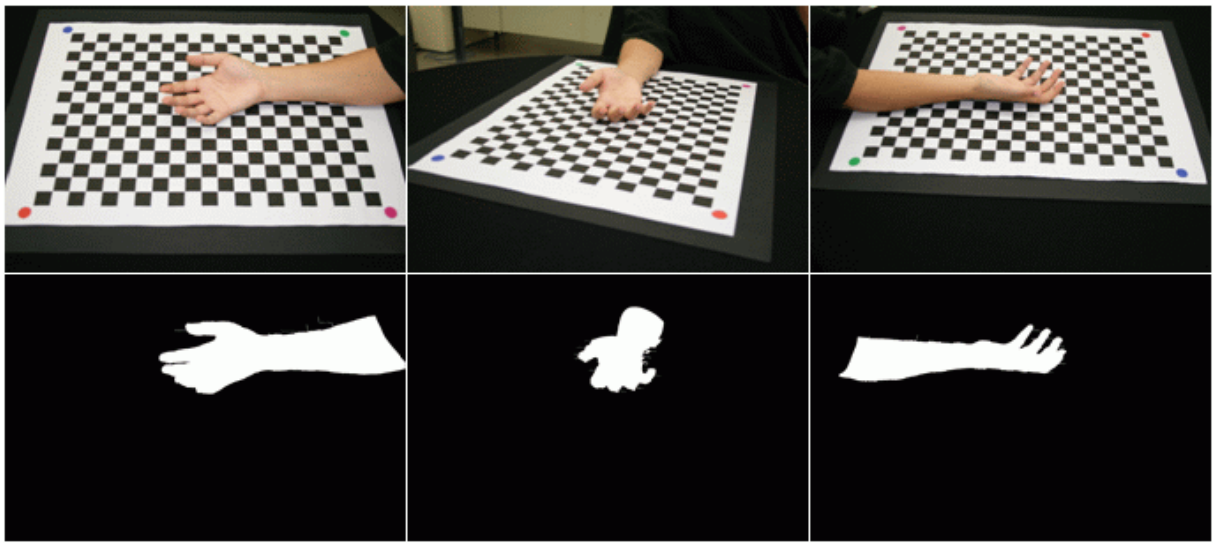

Fig. 5.82 – Original images of the human hand (on the top) and silhouettes images obtained using the developed algorithm of skin colour segmentation (on the bottom).

5.3.7 Human hand

Image acquisition

These test results were performed using a real human hand.

The first image sequence consisted on 9 images of the chessboard calibration pattern. On the other hand, for the second image sequence, 16 images were acquired with the hand positioned on top of the chessboard pattern and moving the camera around the hand, Fig. 5.82. All images were acquired with a resolution of 1936×1288 pixels.

Image segmentation

The silhouette images were obtained using the algorithm developed of skin colour segmentation, Fig. 5.82; again, the results were good, but not perfect, Fig. 5.83. The worst results were due to shadowing effects and similarity of the background with the skin.

However, the volumetric-based reconstruction compensates for those bad silhouettes, because only voxels that lie inside all silhouette images remain in the final reconstructed 3D model.

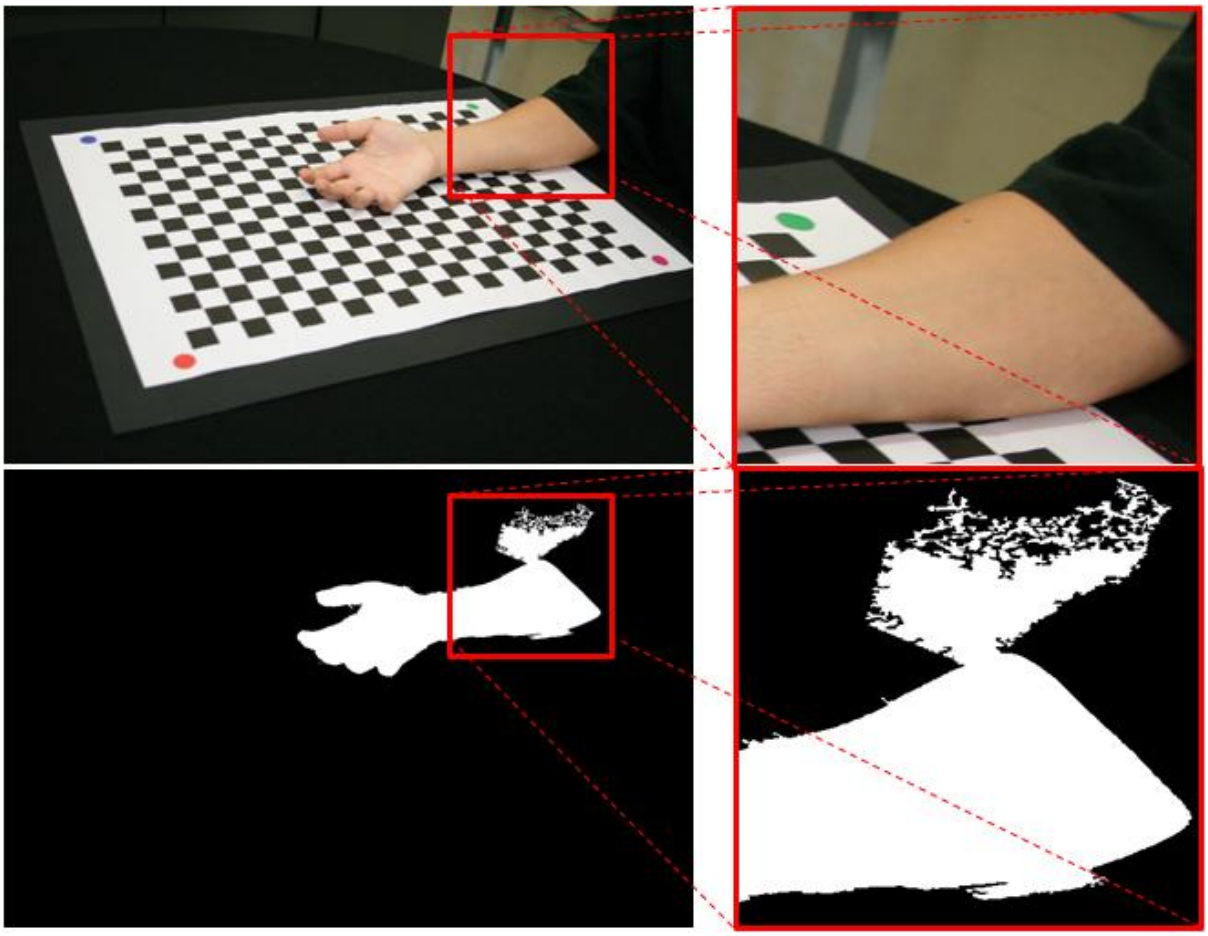

Fig. 5.83 – Effects of shadows in the image segmentation based on skin colour: original (on the top) and obtained silhouette images (on the bottom).

Camera calibration

In all images of the first sequence, the 280 (20×14) chessboard vertices were successfully extracted and matched, and the camera's intrinsic parameters were obtained Table 5.14.

A graphical 3D representation of the camera's extrinsic parameters for the 10 images of the second image sequence, used in the 3D reconstruction process, can be observed in Fig. 5.84. The remaining 6 images were not calibrated because the four outer circles of the calibration pattern were not successfully detected.

For the referred images, the average reprojection error was of $e_{avg} = (0.1842, 0.6130)$ pixels, with a standard deviation of $e_{std} = (0.1649, 0.5561)$ pixels.

Table 5.14 – Camera's intrinsic parameters obtained for the human hand object.

Intrinsic Parameters		
Focal distance (pixel related unit)	f_x	2279.309
	f_y	2295.194
Principal point (in pixel)	c_x	973.221
	c_y	706.056
Radial distortion coefficients	k_1	-0.0747
	k_2	0.0033
Tangential distortion coefficients	p_1	-0.0008
	p_2	-0.0020

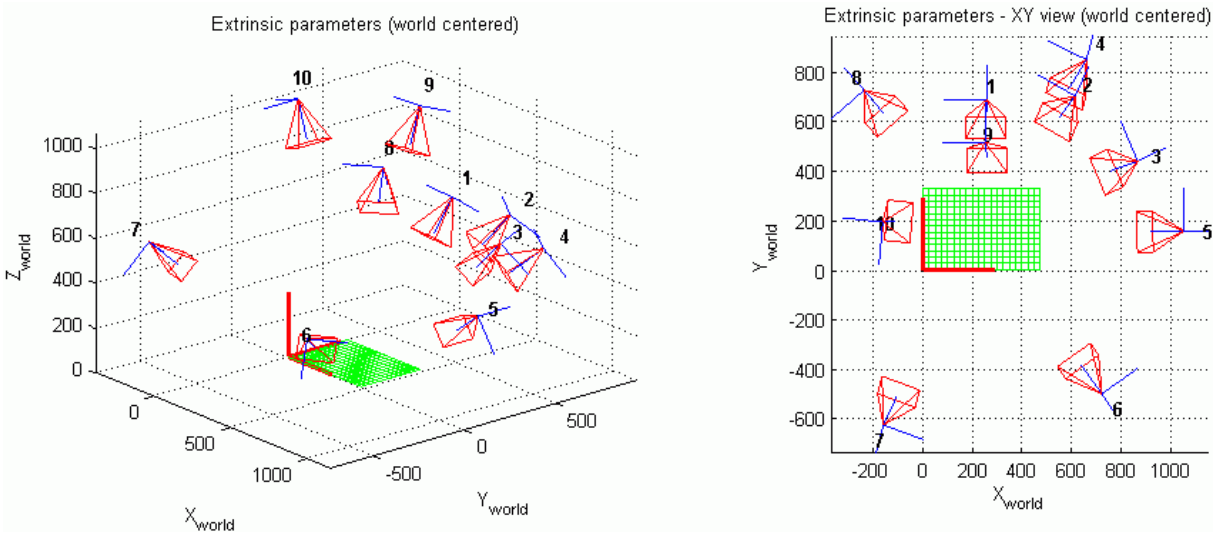

Fig. 5.84 – 3D representation of the camera's extrinsic parameters obtained for the 10 images of the second sequence of the human hand object used in the 3D reconstruction. World 3D axes (in red) are located on the bottom-left vertex of the chessboard (the green grid). In both graphics the scale is in mm.

Volumetric-based reconstruction

The computation of the initial bounding box 3D coordinates required seven iterations to decrease the initial value for the maximum height of the volume, leading to the following results, in mm:

- $\min_X = -102;\ \max_X = 402;$
- $\min_Y = -48;\ \max_Y = 301;$
- $\min_Z = 0;\ \max_Z = 114.$

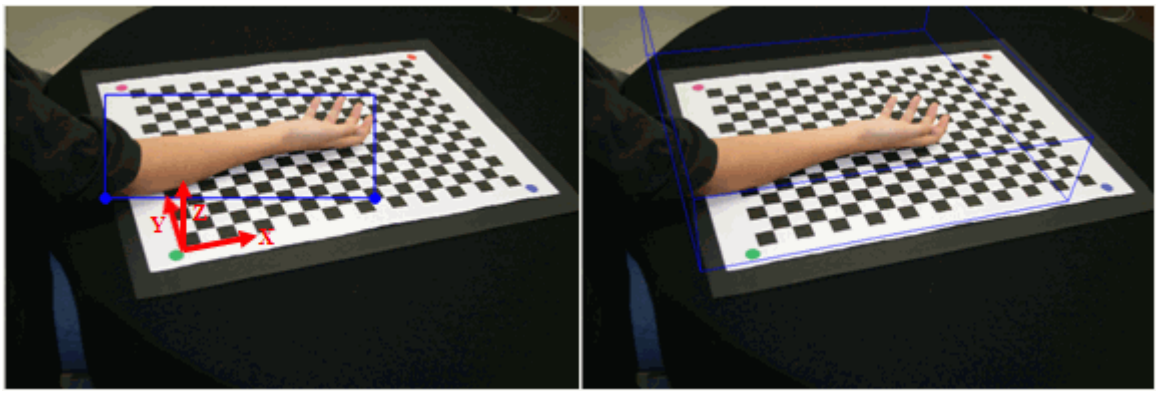

Fig. 5.85 – On the left: example of one image where one of the two lower vertices (blue circles) of the hand silhouette's bounding rectangle (blue rectangle) is outside the 3D world coordinates (in red). On the right: back-projection of the initial bounding box (blue parallelepiped), where the real X- and Y- vertices' coordinates are far from the real object.

Because part of the person's arm was segmented along with the hand, Fig. 5.85, the initial bounding volume was not the minimum bounding volume possible. However, the requirement for a successful 3D volumetric reconstruction is that the octree's initial volume must contain the real object, which was attained.

The 3D model reconstruction was performed using 10 calibrated images of the 16 acquired. Fig. 5.86 shows five viewpoints of the 3D reconstructed model for the human hand using the volumetric method only based on silhouettes and considering voxel footprint determined using exact projection.

Some voxels of the model built were coloured black because they were not visible in the acquired images; however, they do belong to the model, as can be seen in Fig. 5.87.

3D model assessment

Using the 6 uncalibrated images, the reprojection error of the reconstructed 3D model was computed. This was performed by calibrating these images with manual detection of the pattern's circles.

Fig. 5.88 presents the six evaluation and rendered images used for the human hand. Qualitatively, it is noticeable that the resulting reconstructed 3D models offer a good visual quality. The worst reconstructed part of the hand was the palm, since it had a concave shape. However, if the evaluation images were also used for reconstruction purposes, some errors of the obtained 3D model would be corrected; namely, a better reconstruction would result in between the fingers.

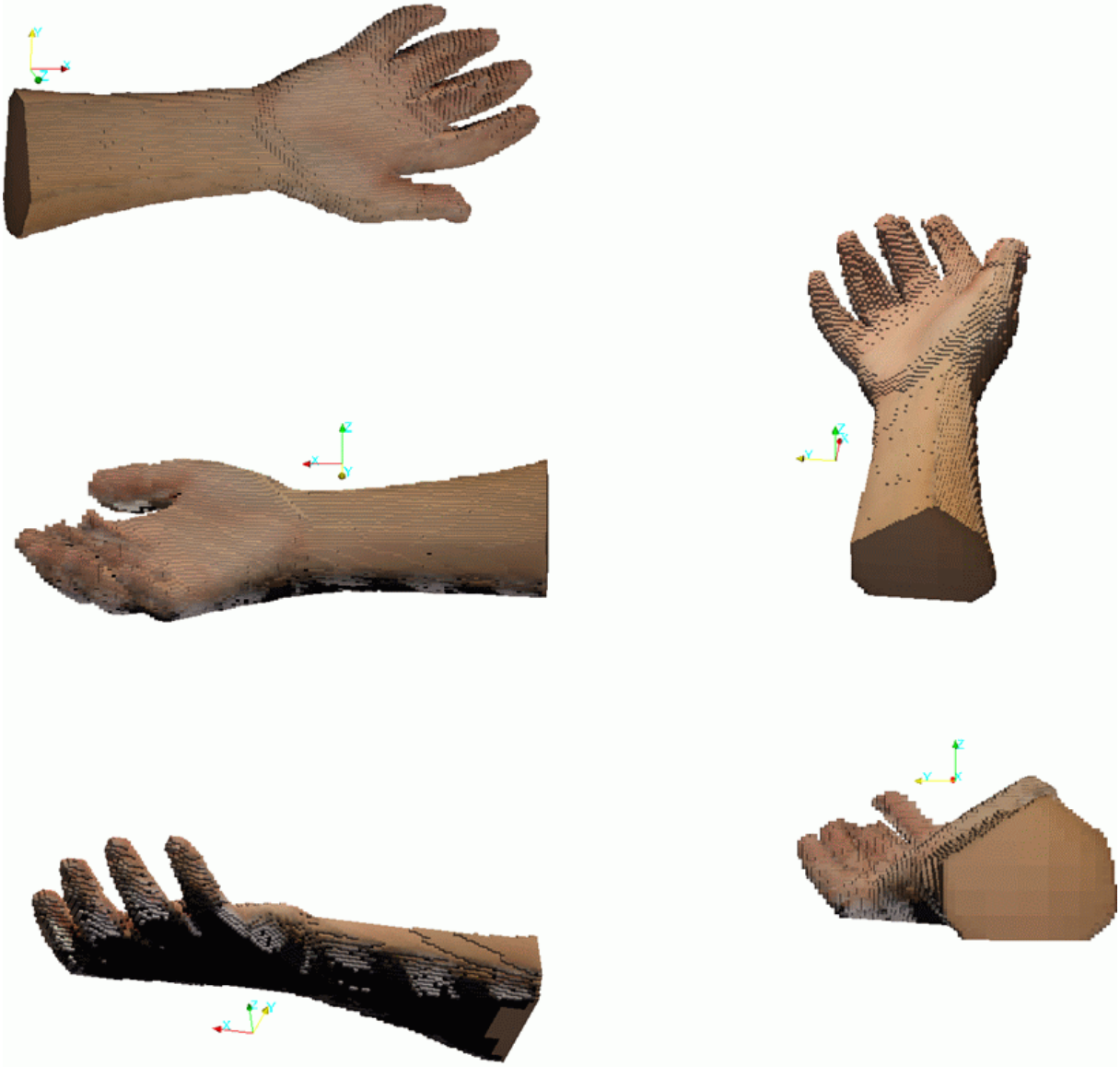

Fig. 5.86 – Five different views of the obtained 3D model for a real human hand, using the volumetric reconstruction method, with the following parameters: number of iterations equal to 7, only silhouettes and exact voxel projection.

The calculated reprojection errors and colour similarity measures relatively to the considered evaluation images were computed, Table 5.15, for an objective evaluation of the 3D reconstruction.

Fig. 5.89 allows the comparison between the original reconstructed model and the smoothed models. From this figure, one can realize the improvement attained by the smoothing, but also perceived that some shrinkage of the models occurred; however, the 3D shape of the hand was not significantly deformed.

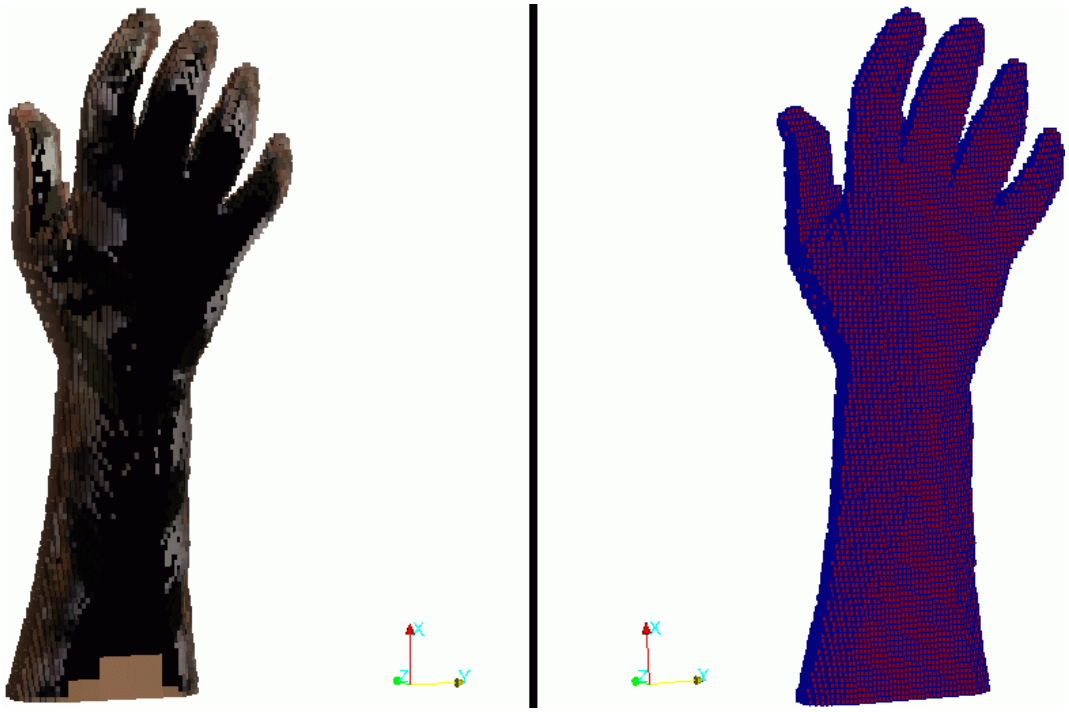

Fig. 5.87 – Original 3D model obtained for the human hand object (on the left) and surface edges (in blue) overlapped with the correspondent 3D volume voxels (in red) (on the right).

Table 5.15 – Reprojection error and colour similarity between the obtained 3D model for the real human hand and the rendered images.

Image	Reprojection error [0-255]		Colour similarity (%)
1st (top) of Fig. 5.88	E_R	15.1928	94.19
	E_G	14.3798	
	E_B	14.8957	
2nd of Fig. 5.88	E_R	25.5393	90.33
	E_G	24.2380	
	E_B	24.1892	
3rd of Fig. 5.88	E_R	21.3753	91.49
	E_G	21.3563	
	E_B	22.3950	
4th of Fig. 5.88	E_R	26.1948	89.85
	E_G	25.5189	
	E_B	25.9139	
5th of Fig. 5.88	E_R	26.2720	89.81
	E_G	25.5788	
	E_B	26.0748	
6th (bottom) of Fig. 5.88	E_R	22.9882	90.84
	E_G	23.0794	
	E_B	24.0168	

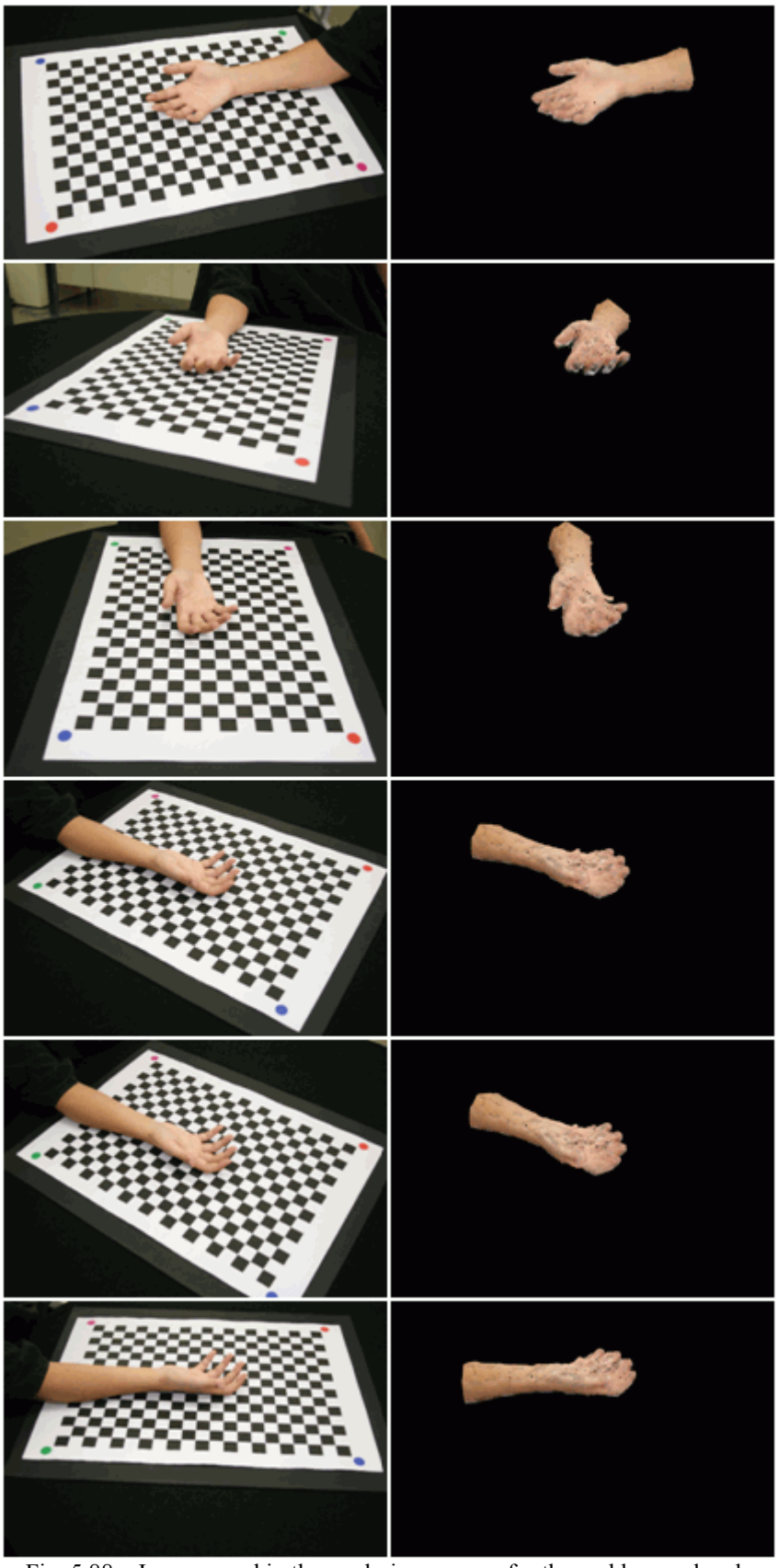

Fig. 5.88 – Images used in the rendering process for the real human hand: evaluation (on the left) and rendered images (on the right).

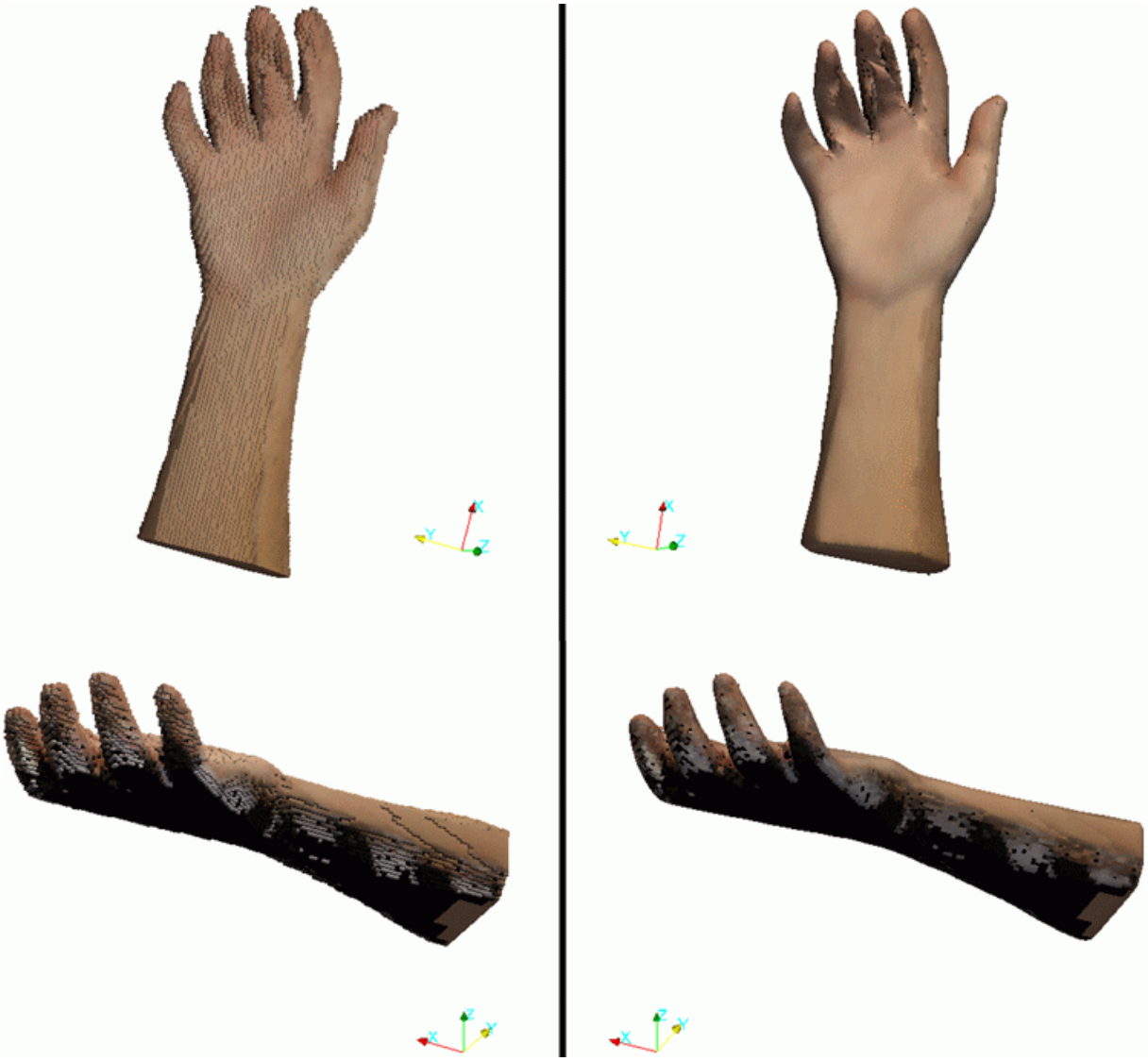

Fig. 5.89 – Original 3D model surface obtained for the human hand object (on the left) and 3D model smoothed by applying 1000 iterations of the Laplacian smoothing filter (on the right).

5.4 Summary

In this Chapter, experimental results, obtained with two methods for 3D object reconstruction – *Stereo-based Reconstruction* and *Volumetric-based Reconstruction* - were presented and discussed. Since the main goal of this Thesis was to develop, implement and compare methodologies to obtain the 3D shape of objects without imposing severe restrictions on the image acquisition process, all images were acquired with a simple black background and under normal indoor lighting conditions.

The first method addressed is based on stereoscopic vision and requires two perspective views of the object to be reconstructed. As such, starting with two uncalibrated images of an

object, usually known as stereo image pair, the epipolar geometry, specified by the fundamental matrix, was extracted and, finally, a disparity map of the object could be obtained.

The stereo-based reconstruction method was tested with two real objects: a Rubik cube and a plastic baby toy. For the first one, which has surfaces with strong features, a good disparity map was obtained. However, since the baby toy presented a smooth surface, with almost no colour or texture variation, the detection and matching of strong features between the stereo image pair were very difficult. Consequently, the epipolar geometry calculated was incorrect, which led to a poor disparity map.

The 3D stereo-based reconstruction was stopped at the disparity map computation, due to the results obtained for smooth, low-textured objects. However, it should be pointed out that an upgrade from projective to a metric reconstruction could pass by adding some additional information on the cameras, e.g. translational motion, or scene, like parallel lines or orthogonal planes.

For the volumetric-based reconstruction, the tests started with static objects: a Rubik cube, a plastic baby toy and some human body part models, mainly a hand and a torso. Then, the developed volumetric-based method was tested with real human body parts: a foot, a head and a hand. Using silhouette and camera calibration data, 3D multi-resolution models were obtained and colourized for each test object. Colour consistency was applied to refine the obtained models for objects with significant texture information, like the Rubik cube. Finally, subjective and analytical characteristics were obtained to evaluate the 3D models built.

The performed camera calibration method was found to be very precise, with reprojection errors around 10^{-4} pixels. This is even more important since accurate calibration is an essential requirement for exact volumetric reconstruction, since errors on the camera's parameters will reflect on errors on the built 3D models, especially on areas that are more separated from the calibration pattern.

Generally, the volumetric-based reconstruction method was able to successfully reconstruct objects with smooth surfaces or with complicated morphology, like a human hand. For most of the tested objects, the 3D models effectively reconstructed and the calculated measures were very close to original ones. However, this method revealed some restrictions, such as a background with low colour variation and a suitable calibration apparatus. Also, since the acquired images were used both for calibration and reconstruction

purposes, limitations on the camera's viewpoints should be addressed: both object and calibration pattern must be visible in all acquired images. Imposing these restrictions could lead to insufficient viewpoints for the successfully reconstruction of the object, and consequently, to poorly reconstructed 3D models, like in the case of the human head. In the experiments, it was also verified that the silhouettes did not need to be perfect, as some errors could be overcome if there were adequate viewpoints to compensate such errors.

6
Conclusions and Future work

6.1 Conclusions

The final goal of this Thesis was to develop, implement and evaluate computational methods to build accurate 3D models of external human body parts. Thus, initially, the existing methods for 3D reconstruction, particularly designed for human anatomical structures, were reviewed. This assumed particular importance as it presents the current theoretical fundaments and frameworks for the 3D reconstruction from images.

Afterwards, the problem of camera calibration, which allows knowing the transformation between 3D world points onto their projections on 2D image planes, was addressed. Initially, general concepts about image formation and coordinate transformations were introduced. Subsequently, the most commonly used calibration methods were described, and classified into traditional and auto-calibration methods.

Afterwards, two image-based methods for the 3D reconstruction of static objects from images were developed, implemented and evaluated.

The first method is based on stereo vision and involves the following:

- identify strong features in the input images;

- find the matching of strong features detected between the two input images;
- estimate the epipolar geometry involved between the two input images;
- rectify one of the input images based on the epipolar geometry found;
- establish the dense matching between the rectified images and compute the disparity map.

Briefly, from the experimental results obtained, it can be affirmed that the stereo vision based method produces good results when applied to objects with strong features, such as texture or edges, straightforwardly detected in the input images, which are consequently easy to extract and correctly match. As observed from the experimental tests presented and discussed in the previous chapter, the results of lowest quality results were related to features that were wrong extracted and/or matched. These errors were then conveyed to the next steps, decreasing the quality of the calculated epipolar geometry to an inappropriate level, which led to a disparity map of poor quality. Thus, with objects having smooth surfaces, with almost no colour or texture variations, the accurate detection and matching of strong features between the input images are very difficult to achieve and consequently, the 3D models built will be of low quality.

The second 3D reconstruction method developed, implemented and evaluated is based on silhouette reconstruction trough volume, or space, carving, and integrates the following steps:

- calibration of the camera used in order to obtain global coordinates - this step is also needed in the first reconstruction method addressed, if global coordinates are needed, or the coordinates will be affected by a global scale factor as well as by the distortions imposed by the optical system of the image acquisition system;
- segmentation of the input images;
- computation of the visual hull;
- and, optionally, photo hull determination.

In the volumetric-based reconstruction solution developed, two sets of images are required. A first set is acquired using a chessboard calibration pattern and used in the camera intrinsic calibration, i.e. to find the camera's internal parameters. Then, placing the object to

be reconstructed on the same calibration pattern, the second image set is acquired and used in the camera extrinsic calibration, i.e. to find the camera's pose (orientation and location).

In order to allow the correct estimation of the pattern orientation, the traditional chessboard pattern was modified by adding four coloured circles on its four outer vertices. This modification was required due to the symmetry of the classical chessboard pattern. As such, the 3D spatial orientation of the new calibration pattern could be correctly found, and the camera calibration procedure was totally automated.

The main restriction of the camera calibration method implemented, is that the calibration pattern must be entirely visible in all acquired images. Apart from this, the images can be acquired by moving freely the camera.

For better image segmentation results, a simple black background was used, but this is not a constraint. When the object to be reconstructed had a skin-like colour surface, the segmentation algorithm developed and based on skin colour produced good results.

The visual hull computation combines both calibration and silhouette information to build the 3D models. The developed carving process starts with an initial box containing the object to be reconstructed and uses an octree data structure to store and refine the voxels belonging to the 3D model under building. The initial bounding box is automatically established from the calibration parameters and silhouette information. The octree structure adopted store voxels with flexible width, length and height, meaning that they are not constrained to being cubes as in the traditional volumetric-based methods. During the 3D reconstruction process, voxels' footprint calculations address the fact that they are convex polygons, which avoids comparisons over the entire silhouettes.

Finally, the voxel visibility is established allowing for a correct voxels' colour computation (texture). Due to its conservative properties, the visual hull may not successfully carve some parts of the original object, especially its concavities. If there is enough surface colour variation, a final photo-consistency test can improve the obtained 3D model. However, the consistency test is highly dependent on the reflectivity of the objects surface and amount of visual information available.

For the second reconstruction approach, tests included man-made objects and real human body parts. Generally, the volumetric-based approach had no difficulty to reconstruct objects with smooth surfaces or with complicated shapes and topologies. Silhouettes do not need to be perfectly extracted due to the conservative property of the volumetric-based

approach. However, this method has some restrictions, such as backgrounds with low colour variation to be easily segmented, suitable calibration procedures and limited viewpoints at which the images could be acquired.

The developed volumetric-based method is a very good solution when a full 3D model is required. It is fast when low resolutions are demanded; meaning that it is suitable in applications that require a coarse but reliable approximation of the objects' shape, e.g. in gait analysis and human posture classification. The method does not require any special knowledge about 3D vision algorithms, since it is a fully automated method. Furthermore, it does not require a posterior mesh registration, which is one of the most common drawbacks of other 2.5D reconstruction methods, like the stereo-based ones.

The developed volumetric-based method provides better results for objects with sizes proportional to the calibration pattern used. When the objects are too big, the calibration pattern is further away from the image plane and not so accurate results could be obtained. On the other hand, for very small objects, since the camera must completely acquire the calibration pattern, the object can be further away from the image plane and important pixel information could be lost in the reconstruction process. However, if the object's and calibration pattern's sizes are well balanced, the camera's poses are not so restricted and more silhouette and pixel information can be used in the reconstruction process.

Our final conclusions are about the type of human external structures to be reconstructed and the suitability of the two computational methods developed for their reconstruction. Human limbs (arms, hands, legs or feet) or a full person body can be successfully reconstructed using the volumetric-based method. The 3D reconstruction of a face is better achieved using the stereo-based method, because: 1) faces have good imaged features than can be tracked on a small baseline camera motion and, 2) if the volumetric-based method is used, then a calibration pattern needs to be placed on the back of the face, which puts severe constrains on the camera viewpoints. One can extrapolate these conclusions for a wider range of objects: both methods assume static objects with lambertian surfaces. The stereo-based method is more suitable for highly textured objects and with small baseline motions involved and the volumetric-based method performs more accurately for objects without or reduced texture information and with wide baseline motions. Also, if an object has relevant texture information, then the volumetric-based reconstruction can be considerably improved by adopting a photo-consistency testing into the obtained 3D visual

hull. In fact, this consistency criterion turns this method more suitable for a wider range of situations and/or objects.

6.2 Future work

Although the volumetric-based method implemented originated experimental results that were quite satisfactory and promising, it can be improved in different aspects.

The camera calibration procedure can be improved by using self-identifying markers (e.g. [Fiala, 2004], [Atcheson, 2010]). This would remove the need to segment the calibration pattern based on its colour, providing a more robust process, and would eliminate the constraint of having to visualize the full calibration pattern on every viewpoint at which the images are acquired.

Another improvement in the camera calibration procedure could be the elimination of the calibration pattern, by implementing an auto-calibration method that uses the information relative to a specific motion (e.g. turntable motion, [Zhang, 1995], [Fitzgibbon, 1998], [Gibson, 1998], [Liu, 2000]) or by using a relatively small amount of markers strategically placed on or around the object.

Whilst the adopted multi-resolution approach reduce the computation time and provides more accurate 3D models, mainly in complex regions of the object's surface, the resulting computation times (from some minutes to a few hours) can still be prohibitive in many possible applications. However, if the models do not need to include the texture information, only a few minutes are necessary for a truthful 3D reconstruction. During the 3D reconstruction process using the volumetric-based method, the voxel footprint and visibility computations are the most time consuming steps. Therefore, an effort should be made to decrease these computation times, using other solutions to estimate the voxels' visibility and photo-consistency, such as the solutions proposed in [Guerchouche, 2008] or [Wong, 2010]. Other possible approaches are to perform the texture analysis on the extracted surface patches of the voxelized octree or to build a texture mapped model instead of mapping the visual hull onto the images.

Most of the voxel-based 3D reconstructions perform in non-real time. The camera calibration, silhouette extraction, visual hull determination and voxel colouring slow down the entire computational process. However, a decrease in the volumetric reconstruction time

can be achieved, either by using optimized processes (e.g., [Sainz, 2002], [Li, 2003]), by adopting dedicated hardware for image processing or graphical computation (e.g. [Sharp, 2008], [Feldmann, 2010], [Gobron, 2010]), or by employing a mixture of pipelined and parallel computer system architecture (e.g. [Hasenfratz, 2004]). If the goal is human body reconstruction, processing time can be improved by including prior knowledge about human body shape (e.g. [Mikić, 2001], [Starck, 2005], [Feldmann, 2010]).

Fusing active range sensors with passive intensity images can facilitate the building of completed and of high-resolution 3D models, since the combination of active and passive techniques may compensate the weakness of each technique. For example, the integration of laser scanning and photogrammetric techniques has a great potential in several mapping applications, by increasing the accuracy, level of automation and robustness (e.g. [Beraldin, 2004], [Rönnholm, 2007], [Huhle, 2010], [Izadi, 2011]). Based on this idea of hybrid methodologies, the visual hull reconstruction combined with stereo feature matching had already proven to attain results of good quality (e.g. [Starck, 2007]).

With the possibility of decreasing the 3D reconstruction computational time, the building of 3D models for deformable or non-static objects is a natural evolution step. For example, the reconstructing and updating of the 3D shape of a moving person is an ongoing topic of research; especially, with the objective of creating 3D models of people that accurately reflect their time-varying shape and appearance (e.g. [Vedula, 2000], [Aans, 2002], [Dewaele, 2004], [Starck, 2009], [Feldmann, 2010], etc.).

Bibliography

[3D-Scanner, 2012] 3D-Scanner, *3D3Solutions*, http://www.3d3solutions.com/products/3d-scanner/, 2012.

[3dMDcranial, 2012] 3dMDcranial, *3dMD*, http://www.3dmd.com/3dmdcranial/ (retrieved on May 2012), 2012.

[3dMDtorso, 2012] 3dMDtorso, *3dMD*, http://www.3dmd.com/3dmdtorso/ (retrieved on May 2012), 2012.

[Aans, 2002] H. Aans and F. Kahl, *Estimation of Deformable Structure and Motion*, Vision and Modelling of Dynamic Scenes Workshop, Copenhagen, Denmark, 2002.

[Abdel-Aziz, 1971] Y. I. Abdel-Aziz and H. M. Karara, *Direct Linear Transformation into Object Space Coordinates in Close-Range Photogrammetry*, Close-Range Photogrammetry, Urbana, Illinois, IL, USA, pp. 1-18, 1971.

[AbsoluteArm, 2012] AbsoluteArm, *Portable Measuring Arms, ROMER*, http://www.romer.eu/portable-measuring-arms_524.htm (retrieved on May 2012), 2012.

[Agapito, 1998] L. d. Agapito, E. Hayman and I. Reid, *Self-Calibration of a Rotating Camera with Varying Intrinsic Parameters*, British Machine Vision, Southampton, United Kingdom, pp. 105-114, 1998.

[Agapito, 1999] L. d. Agapito, R. I. Hartley and E. Hayman, *Linear Calibration of a Rotating and Zooming Camera*, Computer Vision and Pattern Recognition, Fort Collins, Colorado, CO, USA, vol. 1, pp. 15-21, 1999.

[Agapito, 2001] L. d. Agapito, E. Hayman and I. D. Reid, *Self-Calibration of Rotating and Zooming Cameras*, Computer Vision, vol. 45, n. 2, pp. 107-127, 2001.

[Ahmed, 1999a] M. Ahmed, E. Hemayed and A. Farag, *A Neural Approach for Single- and Multi-Image Camera Calibration*, Image Processing, Kobe, Japan, vol. 3, pp. 925-929, 1999a.

[Ahmed, 1999b] M. T. Ahmed, E. E. Hemayed and A. A. Farag, *Neurocalibration: A Neural Network That Can Tell Camera Calibration Parameters*, Computer Vision, Kerkyra, Greece, pp. 463-468, 1999b.

[Ahmed, 2000] M. T. Ahmed and A. A. Farag, *A Neural Optimization Framework for Zoom Lens Camera Calibration*, Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, SC, USA, vol. 1, pp. 403-409, 2000.

[Alexander, 2010] O. Alexander, M. Rogers, W. Lambeth, J.-Y. Chiang, W.-C. Ma, C.-C. Wang and P. Debevec, *The Digital Emily Project: Achieving a Photoreal Digital Actor*, Computer Graphics and Applications, vol. 30, n. 4, pp. 20-31, 2010.

[Andersen, 2004] B. Andersen and K. Noone, *Controlled Texture Pushing and Crossing Seams in UV Space Using Maya and Photorealistic Renderman*, Graphics

Tools, Computer Graphics for Motion Picture Production, vol. 9, n. 4, pp. 57-67, 2004.

[Andrew, 1979] A. M. Andrew, *Another Efficient Algorithm for Convex Hulls in Two Dimensions*, Information Processing Letters, vol. 9, n. 5, pp. 216-219, 1979.

[Andriluka, 2010] M. Andriluka, S. Roth and B. Schiele, *Monocular 3D Pose Estimation and Tracking by Detection*, Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.

[Anthroscan, 2012] Anthroscan, *Human-Solutions*, http://www.human-solutions.com/fashion/front_content.php?idcat=140&lang=7 (retrieved on May 2012), 2012.

[Armstrong, 1996a] M. N. Armstrong, *Self-Calibration from Image Sequences*, PhD Thesis, University of Oxford, United Kingdom, 1996a.

[Armstrong, 1996b] M. N. Armstrong, A. Zisserman and R. Hartley, *Self-Calibration from Image Triplets*, Computer Vision, Cambridge, United Kingdom, vol. 1, pp. 3-16, 1996b.

[Aspert, 2002] N. Aspert, D. Santa-Cruz and T. Ebrahimi, *MESH: Measuring Errors Between Surfaces using the Hausdorff Distance*, Multimedia and Expo, Lausanne, Switzerland, vol. 1, pp. 705-708, 2002.

[Atcheson, 2010] B. Atcheson, F. Heide and W. Heidrich, *CALTag: High Precision Fiducial Markers for Camera Calibration*, Vision, Modeling and Visualization, Siegen, Germany, pp. 41-48, 2010.

[Atkinson, 2010] G. A. Atkinson, M. L. Smith, L. N. Smith and R. P. Warr, *Correlating Plagiocephaly Skull Deformations with Facial Asymmetry using Projected Pattern 3D Imaging*, Biomedical Engineering, Innsbruck, Austria, pp. 249-256, 2010.

[Azevedo, 2008a] T. C. S. Azevedo, J. M. R. S. Tavares and M. A. P. Vaz, *Análise do Método de Calibração de Câmaras Proposto por Zhang*, Congresso Luso-Moçambicano de Engenharia, Maputo, Moçambique, 2008a.

[Azevedo, 2008b] T. C. S. Azevedo, J. M. R. S. Tavares and M. A. P. Vaz, *3D Object Reconstruction from Uncalibrated Images using an Off-the-Shelf Camera*, Advances in Computational Vision and Medical Image Processing: Methods and Applications, ISBN: 978-1-4020-9085-1(print) - 978-1-4020-9086-8 (online), Springer, pp. 117-136, 2008b.

[Azevedo, 2009] T. C. S. Azevedo, J. M. R. S. Tavares and M. A. P. Vaz, *3D Reconstruction of External Anatomical Structures from Image Sequences*, Computational Mechanics, Columbus, Ohio, USA, vol. 13, n. 3, pp. 359-369, 2009.

[Azevedo, 2010] T. C. S. Azevedo, J. M. R. S. Tavares and M. A. P. Vaz, *Three-dimensional Reconstruction and Characterization of Human External Shapes from Two-dimensional Images using Volumetric Methods*, Computer Methods in Biomechanics and Biomedical Engineering, vol. 13, n. 3, pp. 359-369, 2010.

[Batista, 1998] J. Batista, H. Araújo and A. T. d. Almeida, *Iterative Multistep Explicit Camera Calibration*, Computer Vision, Bombay, India, vol. 15, pp. 709-714, 1998.

[Beardsley, 1995] P. A. Beardsley and A. Zisserman, *Affine Calibration of Mobile Vehicles*, Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision, Xi'an, China, pp. 214-221, 1995.

[Beraldin, 2004] J.-A. Beraldin, *Integration of Laser Scanning and Close-Range Photogrammetry - The Last Decade and Beyond* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science Congress, Istanbul, Turkey, pp. 972-983, 2004.

[Bertozzi, 2012] M. Bertozzi, *Image Processing for Vehicular Applications*, Handbook of Intelligent Vehicles, 2012.

[BodySCAN3D, 2012] BodySCAN3D, *3D-Shape GmbH*, www.3d-shape.com/produkte/face_e.php (retrieved in May 2012), 2012.

[Böhler, 2004] W. Böhler and A. Marbs, *3D Scanning and Photogrammetry for Heritage Recording: A Comparison*, Geoinformatics, Gävle, Sweden, pp. 291-298, 2004.

[Botsch, 2010] M. Botsch, L. Kobbelt and B. Levy, *Polygon Mesh Processing*, A. K. Peters Series, 2010.

[Bouchrika, 2007] I. Bouchrika and M. S. Nixon, *Model-Based Feature Extraction for Gait Analysis and Recognition*, Model-based Imaging, Rendering, Image Analysis and Graphical special Effects, INRIA Rocquencourt, France, pp. 150-160, 2007.

[Bouguet, 1999] J.-Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker*, Intel Corporation, Microprocessor Research Labs, Technical Report, 1999.

[Bradski, 2008] G. Bradski and A. Kaehler, *Learning OpenCV*, O'Reilly Media, 2008.

[Brunsman, 1997] M. A. Brunsman, H. Daanen and K. M. Robinette, *Optimal Postures and Positioning for Human Body Scanning*, Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Canada, pp. 266-273, 1997.

[Byrne, 2007] P. J. Byrne and J. R. Garcia, *Autogenous Nasal Tip Reconstruction of Complex Defects: A Structural Approach Employing Rapid Prototyping*, Archives of Facial Plastic Surgery, vol. 9, n. 5, pp. 358-364, 2007.

[Calabrese, 2007] S. Calabrese, A. Gandhi and C. Zhou, *Full Body 3D Scanning*, 3D Photography: Final Project Report, Computer Science Department, Columbia University, New York, USA, 2007.

[NextGen, 2007] B. S. Calipers, *Baseline® Skinfold Calipers*, NextGen Ergonomics, www.nexgenergo.com/medical/caliper3.html (retrieved in October 2007), 2007.

[Čarnický, 2006] J. Čarnický and D. C. Jr., *Three-Dimensional Measurement of Human Face with Structured-Light Illumination*, Measurement Science Review, vol. 6, n. 1, pp. 1-4, 2006.

[Carroll, 2011] M. Carroll, M.-E. Annabell and K. Rome, *Reliability of Capturing Foot Parameters Using Digital Scanning and the Neutral Suspension Casting Technique*, Foot and Ankle Research, vol. 4, n. 1, pp. 4-9, 2011.

[Chadwick, 1989] J. E. Chadwick, D. R. Haumann and R. E. Parent, *Layered Construction for Deformable Animated Characters*, Computer Graphics and Interactive Techniques, Boston, Massachusetts, MA, USA, vol. 23, n. 3, pp. 243-252, 1989.

[Chandler, 2005] J. H. Chandler, J. G. Fryer and A. Jack, *Metric Capabilities of Low-cost Digital Cameras for Closerange Surface Measurement*, Photogrammetric Record, vol. 20, n. 109, pp. 12-26, 2005.

[Chapra, 1988] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, McGraw-Hill, 1988.

[Chen, 2007] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen and W. Gao, *Enhancing Human Face Detection by Resampling Examples Through Manifolds*, Systems, Man and Cybernetics, Part A, vol. 37, n. 6, pp. 1017-1028, 2007.

[Cheng, 1998] M. Cheng, *Visualisation of Superficial Breast Changes*, MSc first year report, Department of Computer Science, The University of Western Australia, Nedlands, Australia, 1998.

[Cheung, 2005] K.-M. Cheung, S. Baker and T. Kanade, *Shape-From-Silhouette Across Time - Part I: Theory and Algorithms*, Computer Vision, vol. 62, n. 3, pp. 221-247, 2005.

[Chiari, 2008] Y. Chiari, B. Wang, H. Rushmeier and A. Caccone, *Using Digital Images to Reconstruct Three-dimensional Biological Forms: A New Tool for Morphological Studies*, Biological Journal of the Linnean Society, vol. 95, pp. 425-436, 2008.

[Chien, 1984] C. H. Chien and J. K. Aggarwal, *A Volume/SurfaceRepresentation*, Pattern Recognition, Montreal, Canada, pp. 817-820, 1984.

[Choi, 1994] D.-H. Choi, S.-Y. Oh, H.-D. Chang and K.-I. Kim, *Nonlinear Camera Calibration using Neural Networks*, Neural, Parallel & Scientific Computations, vol. 2, n. 1, pp. 29-42, 1994.

[Cipolla, 1992] R. Cipolla and A. Blake, *Surface Shape from the Deformation of Apparent Contours*, Computer Vision, vol. 9, n. 2, pp. 83-112, 1992.

[ClaroNav, 2012] ClaroNav, *Claron Technology Inc.*, http://www.clarontech.com/applications_claronav.php (retrieved on March 2012), 2012.

[Cognitens, 2012] Cognitens, *WLS400 White Light Scanners*, http://www.cognitens.com/products_98.htm (retrieved on March 2012), 2012.

[Comaniciu, 2002] D. Comaniciu and P. Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, 2002.

[Conci, 2009] A. Conci, J. E. R. d. Carvalho and T. W. Rauber, *A Complete System for Vehicle Plate Localization, Segmentation and Recognition in Real Life Scene*, Latin America Transactions, vol. 7, n. 5, pp. 497-506, 2009.

[Corazza, 2006] S. Corazza, L. Mündermann, A. M. Chaudhari, T. Demattio, C. Cobelli and T. P. Andriacchi, *A Video-Based, Markerless Motion Tracking System for*

Biomechanical Analysis in an Arbitrary Environment, Annals Of Biomedical Engineering, vol. 34, n. 6, pp. 1019-1029, 2006.

[Corral-Soto, 2012] E. R. Corral-Soto, R. Tal, L. Wang, R. Persad, L. Chao, C. Solomon, B. Hou, G. Sohn and J. H. Elde, *3DTown: The Automatic Urban Awareness Project*, Virtual Reality, Costa Mesa, California, CA, USA, pp. 87-88, 2012.

[Cui, 2011] Y. Cui and D. Stricker, *3D Shape Scanning with a Kinect*, Siggraph, Vancouver, Canada, 2011.

[Cyberware, 2007] Cyberware, *Head & Face Color 3D Scanner*, www.cyberware.com/products/scanners/ps.html (retrieved in June 2007), 2007.

[D'Apuzzo, 2002] N. D'Apuzzo, *Modeling human faces with multi-image photogrammetry*, Three-Dimensional Image Capture and Applications, San Jose, CA, USA, vol. 4661, pp. 191-197, 2002.

[D'Apuzzo, 2012] N. D'Apuzzo, *3D Human Body Scanning Technologies - Overview, Trends, Applications*, Imagina, Monaco, 2012.

[DAVID, 2012] DAVID, *LaserScanner*, http://www.david-laserscanner.com/ (retrieved on May 2012), 2012.

[Deli, 2009] R. Deli, E. D. Gioia, L. M. Galantucci and G. Percoco, *Automated Landmark Extraction for Orthodontic Measurement of Faces Using the 3-Camera Photogrammetry Methodology*, Craniofacial Surgery, vol. 21, n. 1, pp. 87-93, 2009.

[Dieterle, 2003] F. Dieterle, *Multianalyte Quantifications by Means of Integration of Artificial Neural Networks, Genetic Algorithms and Chemometrics for Time-Resolved Analytical Data*, PhD Thesis, Institute of Physical and Theoretical Chemistry, Universität Tübingen, Germany, 2003.

[Dimension, 2008] Dimension, *Column-Type CMMs, THOME Präzision*, http://www.thome-praezision.de/eng/column_cmm_eng.html (retrieved in January 2008), 2008.

[Do, 2005] Q. V. Do, P. Lozo and L. C. Jain, *Vision-Based Autonomous Robot Navigation*, Innovations in Robot Mobility and Control, Studies in Computational Intelligence, vol. 8, 2005.

[Dooley, 1982] M. Dooley, *Anthropometric Modeling Programs - A Survey*, Computer Graphics and Applications, vol. 2, n. 9, pp. 17-25, 1982.

[Du, 1993] F. Du and M. Brady, *Self-Calibration of the Intrinsic Parameters of Cameras for Active Vision Systems*, Computer Vision and Pattern Recognition, Madison, WI, USA, pp. 477-482, 1993.

[Einarsson, 2006] P. Einarsson, C.-F. Chabert, A. Jones, W.-C. Ma, B. Lamond, T. Hawkins, M. Bolas, S. Sylwan and P. Debevec, *Relighting Human Locomotion with Flowed Reflectance Fields*, Eurographics Symposium on Rendering, Nicosia, Cyprus, 2006.

[Espuny, 2007] F. Espuny, *A New Linear Method for Camera Self-Calibration with Planar Motion*, Mathematical Imaging and Vision, vol. 27, n. 1, pp. 81-88, 2007.

[ExitSentry, 2012] ExitSentry, *Cernium*, http://www.cernium.com/ (retrieved on March 2012), 2012.

[FaceSCAN3D, 2012] FaceSCAN3D, *3D-Shape GmbH*, www.3d-shape.com/produkte/face_e.php (retrieved in May 2012), 2012.

[FaroArm, 2007] FaroArm, *FARO Swiss Holding GmbH*, http://www.faro.com/content.aspx?ct=us&content=pro&item=2 (retrieved in June 2007), 2007.

[Fassi, 2007] F. Fassi, *3D Modeling of Complex Architecture Integrating Different Techniques - A Critical Overview*, 3D-ARCH, 3D Virtual Reconstruction and Visualization of Complex Architectures, ETH-Zurich, Switzerland, vol. 36, 2007.

[FastScan™, 2012] FastScan™, *Polhemus*, http://polhemus.com/?page=Scanning_Fastscan (retrieved on May 2012), 2012.

[Faugeras, 1987] O. Faugeras and G. Toscani, *Camera Calibration for 3D Computer Vision*, Industrial Applications of Machine Vision and Machine Intelligence, Silken, Japan, pp. 240-247, 1987.

[Faugeras, 1992a] O. Faugeras, Q.-T. Luong and S. J. Maybank, *Camera Self-Calibration: Theory and Experiments*, Lecture Notes in Computer Vision, Springer-Verlag, vol. 588, pp. 321-334, 1992a.

[Faugeras, 1992b] O. Faugeras, T. Luong and S. Maybank, *Camera self-calibration: theory and experiments*, Computer Vision, Santa Margherita, Ligure, Italy, pp. 321-334, 1992b.

[Faugeras, 1993] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.

[Faugeras, 1999] O. Faugeras and R. Keriven, *Variational Principles, Surface Evolution, PDE's, Level Set Methods and the Stereo Problem*, Image Processing, vol. 7, n. 3, pp. 336-344, 1999.

[Faugeras, 2000] O. Faugeras, L. Quan and P. Sturm, *Self-calibration of a 1D Projective Camera and its Application to the Self-calibration of a 2D Projective Camera*, Pattern Analysis and Machine Intelligence, vol. 22, n. 10, pp. 1179-1185, 2000.

[Feldmann, 2010] T. Feldmann, I. Mihailidis, D. Paulus and A. Wörner, *Online Full Body Human Motion Tracking Based on Dense Volumetric 3D Reconstructions from Multi Camera Setups*, Artificial Intelligence, Karlsruhe, Germany, pp. 74-81, 2010.

[Felzenszwalb, 2006] P. F. Felzenszwalb and D. P. Huttenlocher, *Efficient Belief Propagation for Early Vision*, Computer Vision, vol. 70, n. 1, pp. 41-54, 2006.

[Fiala, 2004] M. Fiala, *ARTag Revision 1. A Fiducial Marker System Using Digital Techniques*, Report NRC/ERB-1117, Institute for Information Technology, Canada, 2004.

[Fischler, 1981] M. A. Fischler and R. Bolles, *RANdom SAmpling Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated*

Cartography, Communications of the ACM, New York, NY, USA, vol. 24, n. 6, pp. 381-395, 1981.

[Fischler, 1987] M. A. Fischler and O. Firschein, *Intelligence: The Eye, the Brain and the Computer*, Addison-Wesley Longman Publishing Co., Inc., 1987.

[Fitzgibbon, 1998] A. W. Fitzgibbon, G. Cross and A. Zisserman, *Automatic 3D Model Construction for Turn-Table Sequences*, SMILE Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Freiburg, Germany, vol. 1506, pp. 154-170, 1998.

[Flexiroad, 2012] Flexiroad, *Capflow - Systèmes de Vision*, http://www.capflow.com/ (retrieved on March 2012), 2012.

[Forsyth, 2003] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall Series in Artifical Intelligence, 2003.

[Franco, 2005] J.-S. Franco, *Modélisation à Partir de Silhouettes*, PhD Thesis, Institut National Polytechnique de Grenoble, France, 2005.

[Franco, 2006] J.-S. Franco, M. Lapierre and E. Boyer, *Visual Shapes of Silhouette Sets*, 3D Processing, Visualization and Transmission, University of North Carolina, Chapel Hill, NC, USA, 2006.

[Fremont, 2004] V. Fremont and R. Chellali, *Turntable-Based 3D Object Reconstruction*, Cybernetics and Intelligent Systems, Singapore, vol. 2, pp. 1277-1282, 2004.

[Fusiello, 2008] A. Fusiello and L. Irsara, *Quasi-Euclidean Uncalibrated Epipolar Rectification*, Pattern Recognition, Tampa, Florida, FL, USA, pp. 1-4, 2008.

[Geng, 2011] J. Geng, *Structured-light 3D Surface Imaging: a Tutorial*, Advances in Optics and Photonics, vol. 3, pp. 128-160, 2011.

[Gobron, 2010] S. Gobron, C. Marx, J. Ahn and D. Thalmann, *Real-time Textured Volume Reconstruction using Virtual and Real Video Cameras*, Computer Graphics, Singapore, 2010.

[Golub, 1983] G. H. Golub and C. F. V. Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, Maryland, MD, USA, 3rd ed., 1983.

[González, 2003] J. I. González, *Estudio Experimental de Métodos de Calibración y Autocalibración de Cámaras*, PhD Thesis, División de Inteligencia Artificial y Sistemas, Universidad de Las Palmas de Gran Canaria, Spain, 2003.

[Gonzalez, 1987] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd ed., Addison Wesley, 1987.

[Grabner, 2010] H. Grabner, J. Matas, L. V. Gool and P. Cattin, *Tracking the Invisible: Learning Where the Object Might be*, Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.

[Grochulla, 2011] M. Grochulla, T. Thormählen and H.-P. Seidel, *Using Spatially Distributed Patterns for Multiple View Camera Calibration*, MIRAGE - Computer Vision/Computer Graphics Collaboration Techniques, Rocquencourt, France, pp. 110-121, 2011.

[Grün, 2003] A. Grün, F. Remondino and L. Zhang, *Image-based Automated Reconstruction of the Great Buddha of Bamiyan, Afghanistan*, Videometrics VII, SPIE Electronic Imaging, Santa Clara, CA, USA, vol. 5013, pp. 129-136, 2003.

[Guan, 2007] L. Guan, J.-S. Franco and M. Pollefeys, *3D Occlusion Inference from Silhouette Cues*, Computer Vision and Pattern Recognition, Minneapolis, Minnesota, MN, USA, pp. 1-8, 2007.

[Guerchouche, 2006] R. Guerchouche, F. Coldefya and T. Zahariab, *Accurate Camera Calibration Algorithm Using a Robust Estimation of the Perspective Projection Matrix*, Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications IX, San Diego, CA, USA, vol. 6315, pp. 1-12, 2006.

[Guerchouche, 2008] R. Guerchouche, O. Bernier and T. Zaharia, *Multiresolution Volumetric 3D Object Reconstruction for Collaborative Interactions*, Pattern Recognition and Image Analysis, vol. 18, n. 4, pp. 621-637, 2008.

[Haering, 2008] N. Haering, P. L. Venetianer and A. Lipton, *The Evolution of Video Surveillance: An Overview*, Machine Vision and Applications, 2008.

[Hall, 1982] E. L. Hall, J. B. K. Tio, C. A. McPherson and F. A. Sadjadi, *Measuring Curved Surfaces for Robot Vision*, Computer, vol. 15, n. 12, pp. 42-54, 1982.

[Handyscan, 2011] Handyscan, *DeltaCad*, http://www.deltacad.pt/, 2011.

[Harris, 1988] C. G. Harris and M. J. Stephens, *A Combined Corner and Edge Detector*, Vision, University of Manchester, England, vol. 15, pp. 147-151, 1988.

[Harris, 2011] S. Harris, *3D Bone Modelling Software May Help Police Identify Bodies*, The Engineer, http://www.theengineer.co.uk/sectors/medical-and-healthcare/news/3d-bone-modelling-software-may-help-police-identify-bodies/1009153.article (retrieved on March 2012), 2011.

[Hartley, 1992] R. Hartley, *Estimation of Relative Camera Positions for Uncalibrated Cameras*, Lecture Notes In Computer Science, Springer-Verlag, vol. 588, pp. 579-587, 1992.

[Hartley, 1993] R. Hartley, *Euclidean Reconstruction from Uncalibrated Views*, Lecture Notes In Computer Science, Springer-Verlag, vol. 825, pp. 237-256, 1993.

[Hartley, 1994] R. Hartley, *Self-Calibration from Multiple Views with a Rotating Camera*, Computer Vision, Stockholm, Sweden, vol. 1, pp. 471-478, 1994.

[Hartley, 1995] R. Hartley and P. Sturm, *Triangulation*, Lecture Notes in Computer Science, vol. 970, pp. 190-197, 1995.

[Hartley, 1997] R. Hartley, *Self-calibration of Stationary Cameras*, Computer Vision, vol. 22, n. 1, pp. 5-23, 1997.

[Hartley, 1998] R. Hartley, *Theory and Practice of Projective Rectification*, Computer Vision, 1998.

[Hartley, 1999] R. Hartley, E. Hayman, L. d. Agapito and I. Reid, *Camera Calibration and the Search for Infinity*, Computer Vision, Kerkyra, Greece, pp. 510-517, 1999.

[Hartley, 2004] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd ed., 2004.

[Harvey, 2012] A. Harvey, *CV Dazzle - Camouflage from Computer Vision*, http://cvdazzle.com/ (retrieved on May 2012), 2012.

[Hasenfratz, 2004] J.-M. Hasenfratz, M. Lapierre and F. Sillion, *A Real-Time System for Full Body Interaction with Virtual Worlds*, Virtual Environments, Grenoble, France, pp. 147-156, 2004.

[Hassanpour, 2004] R. Hassanpour and V. Atalay, *Camera Auto-Calibration using a Sequence of 2D Images with Small Rotations*, Pattern Recognition Letters, vol. 25, n. 9, pp. 989-997, 2004.

[Hayman, 2002] E. Hayman and D. W. Murray, *The Effects of Translational Misalignment in the Self-Calibration of Rotating and Zooming Cameras*, Oxford University Engineering Library Technical Report OUEL 2250/02, University of Oxford, United Kingdom, 2002.

[Heike, 2010] C. L. Heike, K. Upson, E. Stuhaug and S. M. Weinberg, *3D Digital Stereophotogrammetry: A Practical Guide to Facial Image Acquisition*, Head & Face Medicine, vol. 6, n. 18, 2010.

[Heikkilä, 1996] J. Heikkilä and O. Silvén, *Calibration Procedure for Short Focal Length Off-the-Shelf CCD Cameras*, Pattern Recognition, Vienna, Austria, vol. 1, pp. 166-170, 1996.

[Heikkilä, 1997] J. Heikkilä and O. Silvén, *A Four-Step Camera Calibration Procedure with Implicit Image Correction*, Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 1106, 1997.

[Heikkilä, 2000] J. Heikkilä, *Geometric Camera Calibration using Circular Control Points*, Pattern Analysis and Machine Intelligence, vol. 22, n. 10, pp. 1066-1077, 2000.

[Hemayed, 2003] E. E. Hemayed, *A Survey of Camera Self-Calibration*, Advanced Video and Signal Based Surveillance, Coral Gables, FL, USA, pp. 351-357, 2003.

[Henry, 2010] P. Henry, M. Krainin, E. Herbst, X. Ren and D. Fox, *RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments*, Experimental Robotics, Delhi, India, 2010.

[Hernández, 2007] C. Hernández, F. Schmitt and R. Cipolla, *Silhouette Coherence for Camera Calibration under Circular Motion*, Pattern Analysis and Machine Intelligence, vol. 29, n. 2, pp. 343-349, 2007.

[Heyden, 1996] A. Heyden and K. Åström, *Euclidean Reconstruction from Constant Intrinsic Parameters*, Pattern Recognition, Vienna, Austria, vol. 1, pp. 339-343, 1996.

[Heyden, 1997] A. Heyden and K. Åström, *Euclidean Reconstruction from Image Sequences with Varying and Unknown Focal Length and Principal Point*, Computer Vision and Pattern Recognition, Puerto Rico, PR, USA, pp. 438-443, 1997.

[Heyden, 1998] A. Heyden and K. Åström, *Minimal Conditions on Intrinsic Parameters for Euclidean Reconstruction*, Computer Vision, Hong Kong, China, vol. 2, pp. 169-176, 1998.

[Hofmann, 2012] M. Hofmann and D. M. Gavrila, *Multi-view 3D Human Pose Estimation in Complex Environment*, Computer Vision, vol. 96, n. 1, pp. 103-124, 2012.

[Holley, 2009] R. Holley, *How Good Can It Get? Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs*, D-Lib Magazine, vol. 15, n. 3/4, 2009.

[Hsu, 2002] R.-L. Hsu, M. A. Mottaleb and A. K. Jain, *Face Detection in Color Images*, Pattern Analysis and Machine Intelligence, vol. 24, n. 5, pp. 696-706, 2002.

[Huhle, 2010] B. Huhle, P. Jenke and W. Straßer, *Fusion of Range and Color Images for Denoising and Resolution Enhancement with a Non-Local Filter*, Intelligent Systems Technologies and Applications, vol. 5, n. 3/4, pp. 255-263, 2010.

[Ilic, 2006] S. Ilic and P. Fua, *Implicit Meshes for Surface Reconstruction*, Pattern Analysis and Machine Intelligence vol. 28, n. 2, pp. 328-333, 2006.

[Virtual, 2012] V. Iraq/Afghanistan, *Post-Traumatic Stress Disorder Assessment and Treatment project, University of Southern California, CA, USA*, http://ict.usc.edu/projects/ptsd/ (retrieved on March 2012), 2012.

[Isgrò, 1999] F. Isgrò and E. Trucco, *Projective Rectification Without Epipolar Geometry*, Computer Vision and Pattern Recognition, Fort Collins, Colorado, CO, USA, vol. 1, pp. 94-99, 1999.

[Ito, 1991] M. Ito, *Robot Vision Modelling - Camera Modelling and Camera Calibration*, Advanced Robotics, vol. 5, n. 3, pp. 321-335, 1991.

[Izadi, 2011] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison and A. Fitzgibbon, *KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera*, User Interface Software and Technology, Santa Barbara, CA, USA, 2011.

[Jacobs, 2011] S. Jacobs, M. Gessat, T. Walther and V. Falk, *Three-Dimensional Template-based Planning for Transapical Aortic Valve Implantation*, Thoracic and Cardiovascular Surgery, vol. 141, n. 6, pp. 1541-1543, 2011.

[Jang, 1996] J.-H. Jang and K.-S. Hong, *Self-Calibration of a Stereo-Camera by Pure Translational Motion*, Image Processing, Lausanne, Switzerland, vol. 2, pp. 297-300, 1996.

[Ji, 2010] X. Ji and H. Liu, *Advances in View-Invariant Human Motion Analysis: A Review*, Systems, Man, and Cybernetics, vol. 40, n. 1, pp. 13-24, 2010.

[Jiang, 2004] G. Jiang, L. Quan and H.-T. Tsui, *Circular Motion Geometry Using Minimal Data*, Pattern Analysis and Machine Intelligence, vol. 26, n. 6, pp. 721-731, 2004.

[Jones, 2002] M. J. Jones and J. M. Rehg, *Statistical Color Models with Application to Skin Detection*, Computer Vision, vol. 46, n. 1, pp. 81-96, 2002.

[Jumadi, 2012] N. A. Jumadi, K. B. Gan, M. A. M. Ali and E. Zahedi, *Development of Physical 3D Model of Maternal-fetal to Study Light Interaction through Tissue*, Biomedical Engineering, Penang, Malaysia, pp. 557-561, 2012.

[Jung, 2009] S. Jung, D. Jang and M. Kim, *3D Localization of Projected Objects for Surveillance*, Distributed Smart Cameras, Como, Italy, pp. 1-6, 2009.

[Kaiser, 2011] M. Kaiser, G. Heym, N. Lehment, D. Arsic and G. Rigoll, *Dense Point-to-point Correspondences Between 3D Faces using Parametric Remeshing for Constructing 3D Morphable Models*, Workshop on Applications of Computer Vision, Kona, Hawaii, USA, pp. 39-44, 2011.

[Kalkhoven, 2003] T. Kalkhoven and N. Naber, *3D Measurement Methods*, Research project, Computer-Aided Design and Engineering, Faculty of Industrial Design, University of Technology, Delft, The Netherlands, 2003.

[Kanatani, 2000] K. Kanatani, N. Ohta and Y. Kanazawa, *Optimal Homography Computation with a Reliability Measure*, Information and Systems, vol. E83-D, n. 7, pp. 1369-1374, 2000.

[Kinect, 2012] Kinect, *Xbox*, http://www.xbox.com/kinect (retrieved on May 2012), 2012.

[Klaus, 2006] A. Klaus, M. Sormann and K. Karner, *Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure*, Pattern Recognition, Hong Kong, China, pp. 15-18, 2006.

[Kleber, 2009] F. Kleber, *A Survey of Techniques for Document and Archaeology Artefact Reconstruction*, Document Analysis and Recognition, Barcelona, Spain, 2009.

[Kolb, 2009] A. Kolb, E. Barth, R. Koch and R. Larsen, *Time-of-Flight Sensors in Computer Graphics*, Eurographics, STAR – State of The Art Report, pp. 119-134, 2009.

[Kolev, 2012] K. Kolev, T. Brox and D. Cremers, *Fast Joint Estimation of Silhouettes and Dense 3D Geometry from Multiple Images*, Pattern Analysis and Machine Intelligence, vol. 34, n. 3, pp. 493-505, 2012.

[Kovacs, 2006] L. Kovacs, A. Zimmermann, G. Brockmann, M. Gühring, H. Baurecht, N. A. Papadopulos, K. Schwenzer-Zimmerer, R. Sader, E. Biemer and H. F. Zeilhofer, *Three-Dimensional Recording of the Human Face with a 3D Laser Scanner*, Plastic, Reconstructive and Aesthetic Surgery, vol. 59, n. 11, pp. 1193-1202, 2006.

[Kutulakos, 1997] K. N. Kutulakos, *Shape from the Light Field Boundary*, Computer Vision and Pattern Recognition, San Juan, Puerto Rico, PR, USA, pp. 53-59, 1997.

[Kutulatos, 1998] K. N. Kutulatos and S. M. Steiz, *A Theory of Shape by Space Carving*, Technical Report TR692, Computer Science Department, University of Rochester, New York, NY, USA, 1998.

[Ladikos, 2008] A. Ladikos, S. Benhimane and N. Navab, *Efficient Visual Hull Computation for Real-Time 3D Reconstruction using CUDA*, Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, 2008.

[LaserScanArm, 2007] LaserScanArm, *FARO Swiss Holding GmbH*, www.faro.com/content.aspx?ct=us&content=pro&item=1 (retrieved in June 2007), 2007.

[Laurentini, 1994] A. Laurentini, *The Visual Hull Concept for Silhouette-based Image Understanding*, Pattern Analysis and Machine Intelligence, vol. 16, n. 2, pp. 150-162, 1994.

[Lazaros, 2008] N. Lazaros, G. C. Sirakoulis and A. Gasteratos, *Review of Stereo Vision Algorithms: From Software to Hardware*, Optomechatronics, vol. 2, n. 4, pp. 435-462, 2008.

[Lee, 2002] K.-K. Lee, T.-X. Qiu and E.-C. Teo, *3-D Finite Element Modeling of Lumbar Spine (L2/L3) using Digitizer*, Information Technology, vol. 8, n. 2, pp. 161-163, 2002.

[Li, 2002] H. Li, H. Zhang, F.-C. Wu and Z.-Y. Hu, *A New Linear Camera Self-Calibration Technique*, Computer Vision, Melbourne, Australia, 2002.

[Li, 2004] Y. Li and Y. S. Hung, *A Stratified Self-Calibration Method for a Stereo Rig in Planar Motion with Varying Intrinsic Parameters*, Pattern Recognition, Tübingen, Germany, pp. 318-325, 2004.

[Lin, 2011] J. Lin, J. Ming and D. Crookes, *Robust Face Recognition with Partial Occlusion, Illumination Variation and Limited Training Data by Optimal Feature Selection*, IET Computer Vision, vol. 5, n. 1, pp. 23-32, 2011.

[Lindner, 2012] M. Lindner, M. Block and R. Rojas, *Object Recognition using Summed Features Classifier*, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 7267, 2012.

[Liu, 2010] Y.-J. Liu, D.-L. Zhang and M. M.-F. Yuen, *A Survey on CAD Methods in 3D Garment Design*, Computers in Industry, vol. 61, pp. 576-593, 2010.

[Liu, 2000] Y. Liu, H. Tsui and C. Wu, *Resolving Ambiguities of Self-Calibration in Turntable Motion*, Pattern Recognition, Barcelona, Spain, vol. 3, pp. 865-868, 2000.

[Loop, 1999] C. Loop and Z. Zhang, *Computing Rectifying Homographies for Stereo Vision*, Computer Vision and Pattern Recognition, Colorado, CO, USA, 1999.

[Lourakis, 1999] M. I. A. Lourakis and R. Deriche, *Camera Self-Calibration Using the Singular Value Decomposition of the Fundamental Matrix: From Point Correspondences to 3D Measurements*, Research Report 3748, INRIA Sophia Antipolis, France, 1999.

[Lourakis, 2000] M. I. A. Lourakis and R. Deriche, *Camera Self-Calibration Using the Kruppa Equations and the SVD of the Fundamental Matrix: The Case of Varying Intrinsic Parameters*, Technical Report 3911, INRIA - Sophia Antipolis, France 2000.

[Luong, 1997] Q.-T. Luong and O. D. Faugeras, *Self-Calibration of a Moving Camera From Point Correspondences and Fundamental Matrices*, Computer Vision, vol. 22, n. 3, pp. 261-289, 1997.

[Lynch, 1992] M. B. Lynch and C. H. Dagli, *Backpropagation Neural Network for Stereoscopic Vision Calibration*, Machine Vision Architectures, Integration, and Applications, vol. 1615, pp. 289-298, 1992.

[Ma, 1996] S. D. Ma, *A Self-Calibration Technique for Active Vision Systems*, Robotics and Automation, vol. 12, n. 1, pp. 114-120, 1996.

[Ma, 2010] Z. Ma, R. N. M. Jorge, J. M. R. S. Tavares and T. Mascarenhas, *A Review of Algorithms for Medical Image Segmentation and their Applications to the*

Female Pelvic Cavity, Computer Methods in Biomechanics and Biomedical Engineering, vol. 13, n. 2, pp. 235-246, 2010.

[Machado, 2009] M. Machado, D. Lopes, J. Ambrósio, M. Silva and P. Flores, *Development of a Freeform Surface Contact Methodology for Dynamic Analysis of the Human Knee*, Biomechanics, Bragança, Portugal, pp. 505-510, 2009.

[Magnant, 1985] D. Magnant, *Capteur Tridimensional Sans Contact*, Photo-Optical Instrumentation Engineers, vol. 602, pp. 18-22, 1985.

[Mallon, 2007] J. Mallon and P. F. Whelan, *Which Pattern? Biasing Aspects of Planar Calibration Patterns and Detection Methods*, Pattern Recognition Letters, vol. 28, n. 8, pp. 921-930, 2007.

[Marr, 1982] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W. H. Freeman & Co., San Francisco, CA, USA, 1982.

[Massone, 1985] L. Massone, P. Morasso and R. Zaccaria, *Shape from Occluding Contours*, Intelligent Robots and Computer Vision, Cambridge, MA, USA, vol. 521, pp. 114-120, 1985.

[Maybank, 1992] S. J. Maybank and O. D. Faugeras, *A Theory of Self-Calibration of a Moving Camera*, Computer Vision, vol. 8, n. 2, pp. 123-151, 1992.

[McInerney, 1996] T. McInerney and D. Terzopoulos, *Deformable Models in Medical Image Analysis: A Survey*, Medical Image Analysis, vol. 1, n. 2, pp. 91-108, 1996.

[McLauchlan, 1996] P. F. McLauchlan and D. W. Murray, *Active Camera Calibration for a Head-Eye Platform Using the Variable State-Dimension Filter*, Pattern Analysis and Machine Intelligence, vol. 18, n. 1, pp. 15-22, 1996.

[MDX-15/20, 2007] MDX-15/20, *Desktop 3D Scanning and Milling, Roland® Advanced Solutions Division*, www.rolanddga.com/asd/resources/pdf/brochure_MDX20.pdf (retrieved in June 2007), 2007.

[Mega-Capturor, 2012] Mega-Capturor, *Creaform*, http://www.creaform3d.com/pt/3d-body-digitizer/mega-capturor.aspx (retrieved on May 2012), 2012.

[Meier, 2012] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer and M. Pollefeys, *PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight using Onboard Computer Vision*, Autonomous Robots, 2012.

[Melen, 1994] T. Melen, *Geometrical Modelling and Calibration of Video Cameras for Underwater Navigation*, PhD thesis, Institutt for Teknisk Kybernetikk, Norges Techniske Hogskole, Trondheim, Norway, 1994.

[Mendonça, 1999] P. R. S. Mendonça and R. Cipolla, *A Simple Technique for Self-Calibration*, Computer Vision and Pattern Recognition, Fort Collins, Colorado, CO, USA, vol. 1, pp. 500-505, 1999.

[Mendonça, 2001] P. R. S. Mendonça, K.-Y. K. Wong and R. Cipolla, *Epipolar Geometry from Profiles under Circular Motion*, Pattern Analysis and Machine Intelligence, vol. 23, n. 6, pp. 604-616, 2001.

[Menser, 2000] B. Menser and M. Wien, *Segmentation and Tracking of Facial Regions in Color Image Sequences*, Visual Communications and Image Processing, Perth, Australia, pp. 731-740, 2000.

[Meshlab, 2012] Meshlab, *Meshlab Open Source Project*, http://meshlab.sourceforge.net/ (retrieved in May 2012), 2012.

[Mikić, 2001] I. Mikić, M. Trivedi, E. Hunter and P. Cosman, *Articulated Body Posture Estimation from Multi-Camera Voxel Data*, Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, vol. 1, pp. 455-460, 2001.

[Milczarski, 2011] P. Milczarski, *A New Method for Face Identification and Determing Facial Asymmetry*, Studies in Computational Intelligence, vol. 381, pp. 329-340, 2011.

[MiniMagics, 2012] MiniMagics, *MiniMagics Free Stl Viewer 2.0, Materialise*, http://software.materialise.com/materialise-premium-software-additive-manufacturing-professionals (retrieved on May 2012), 2012.

[Mishra, 2011] P. Mishra, G. Koshy, K. Murarka, K. Mall, N. Agarwal and S. B. S., *Vision Based Tracking of Multiple Dynamic Obstacles for Autonomous Robot Navigation*, Computational Intelligence, Communication Systems and Networks, Bali, Indonesia, 2011.

[Moeslund, 2001] T. B. Moeslund and E. Granum, *A Survey of Computer Vision-Based Human Motion Capture*, Computer Vision and Image Understanding, vol. 81, n. 3, pp. 231-268, 2001.

[Moeslund, 2006] 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, vol. 104, n. 2, pp. 90-126, 2006.

[Monaghan, 2011] D. Monaghan, P. Kelly and N. E. O'Connor, *Quantifying Human Reconstruction Accuracy for Voxel Carving in a Sporting Environment*, ACM Multimedia, Scottsdale, Arizona, AZ, USA, 2011.

[Moré, 1977] J. Moré, *The Levenberg-Marquardt Algorithm, Implementation and Theory*, Numerical Analysis, Lecture Notes in Mathematics, Springer-Verlag, vol. 630, pp. 105-116, 1977.

[Moura, 2010] D. C. Moura, J. Boisvert, J. G. Barbosa, J. M. R. S. Tavares and H. Labelle, *Fast 3D Reconstruction of the Spine by Non-expert Users Using a Statistical Articulated Model*, Studies in Health Technology and Informatics, vol. 158, pp. 268-269, 2010.

[Mundermann, 2005] L. Mundermann, A. Mundermann, A. M. Chaudhari and T. P. Andriacchi, *Conditions that Influence the Accuracy of Anthropometric Parameter Estimation for Human Body Segments using Shape-from-silhouette*, Human Body Modeling vol. 5665, pp. 268-277, 2005.

[Naik, 2002] S. Naik, *Real-time Avatar Construction Using Shape Tape*, Technical Report Number 02-036, Effective Virtual Environments Research Group, Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA, 2002.

[Nowakowski, 2009] A. Nowakowski and W. Skarbek, *Colour in Calibration Points Indexing*, Photonics Letters of Poland, vol. 1, n. 1, pp. 43-45, 2009.

[Paliokas, 2010] I. Paliokas, *Archaeo Viz - a 3D Explorative Learning Environment of Reconstructed Archaeological Sites and Cultural Artefacts*, Signal Processing and Multimedia Applications, Athens, Greece, 2010.

[Paraview, 2012] Paraview, *Open Source Cientific Visualization*, http://www.paraview.org/ (retrieved May 2012), 2012.

[Park, 2006] H.-K. Park, J.-W. Chung and H.-S. Kho, *Use of Hand-held Laser Scanning in the Assessment of Craniometry*, Forensic Science International, vol. 160, pp. 200-206, 2006.

[PhotoModeler, 2012] PhotoModeler, *PhotoModeler scanner*, http://www.photomodeler.com/ (retrieved on March 2012), 2012.

[Pinto, 2011] V. C. Pinto, M. A. P. Vaz, R. Santos, B. Nunes, S. Abreu, M. Castro, J. P. V. Boas, R. Silva and M. J. Ferreira, *Relatório Técnico Cientifico - QREN*, LOME - Laboratório de Óptica e Mecânica Experimental, FEUP, Porto, Portugal, 2011.

[Pollefeys, 1996] M. Pollefeys, L. V. Gool and A. Oosterlinck, *The Modulus Constraint: A New Constraint for Self-Calibration*, Pattern Recognition, Vienna, Austria, vol. 1, pp. 349-353, 1996.

[Pollefeys, 1997] M. Pollefeys and L. V. Gool, *A Stratified Approach to Metric Self-Calibration*, Computer Vision and Pattern Recognition, Puerto Rico, PR, USA, pp. 407-412, 1997.

[Pollefeys, 1998] M. Pollefeys, R. Koch and L. V. Gool, *Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters*, International Journal of Computer Vision, vol. 32, n. 1, pp. 7-25, 1998.

[Pollefeys, 1999] M. Pollefeys, R. Koch and L. V. Gool, *Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters*, Computer Vision, vol. 32, n. 1, pp. 7-25, 1999.

[Poppe, 2007] R. Poppe, *Vision-based Human Motion Analysis: An overview*, Computer Vision and Image Understanding, vol. 108, n. 1-2, pp. 4-18, 2007.

[Pribanić, 2010] T. Pribanić, S. Mrvoš and J. Salvi, *Efficient Multiple Phase Shift Patterns for Dense 3D Acquisition in Structured Light Scanning*, Image and Vision Computing, vol. 28, n. 8, pp. 1255-1266, 2010.

[Qiang, 2007] W. Qiang, Z. Pan, C. Chun and B. Jianjun, *Surface Rendering for Parallel Slices of Contours from Medical Imaging*, Computing in Science and Engineering vol. 9, n. 1, pp. 32-37, 2007.

[Qiuyu, 2012] Z. Qiuyu and L. Yichun, *A Practical Estimating Method of Camera's Focal Length and Extrinsic Parameters from a Planar Calibration Image*, Intelligent System Design and Engineering Application, Sanya, Hainan, China, pp. 138-142, 2012.

[Remondino, 2004] F. Remondino, *3-D Reconstruction of Static Human Body Shape from Image Sequence*, Computer Vision and Image Understanding, vol. 93, n. 1, pp. 65-85, 2004.

[Remondino, 2005] F. Remondino, A. Guarnieri and A. Vettore, *3D Modeling of Close-Range Objects: Photogrammetry or Laser Scanning?*, Electronic Imaging, vol. 5665, pp. 216-225, 2005.

[Remondino, 2006] F. Remondino and S. El-Hakim, *Image-Based 3D Modelling: A Review*, The Photogrammetric Record, vol. 21, n. 115, pp. 269-291, 2006.

[Roberts, 1963] L. G. Roberts, *Machine Perception of 3-D Solids*, PhD Thesis, MIT Lincoln Laboratory, 1963.

[Rönnholm, 2007] P. Rönnholm, E. Honkavaara, P. Litkey, H. Hyyppä and J. Hyyppä, *Integration of Laser Scanning and Photogrammetry*, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, vol. 3, n. 3/W52, pp. 355-362, 2007.

[Rosenfeld, 2001] A. Rosenfeld, *From Image Analysis to Computer Vision: An Annotated Bibliography, 1955-1979*, Computer Vision and Image Understanding, vol. 84, n. 2, pp. 298-324, 2001.

[Rudek, 2005] M. Rudek, P. Roberto and G. Kurka, *3-D Measurement from Images using a Range Box*, Mechanical Engineering, Ouro Preto, Brasil, 2005.

[Ruf, 1998] A. Ruf, G. Csurka and R. Horaud, *Projective Translations and Affine Stereo Calibration*, Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, pp. 475-481, 1998.

[Sahai, 2010] N. Sahai, R. P.Tewari and L. Singh, *Mechanical Behavior of Muscles During Flexion and Extension of Lower Limb on Variable Age Group by Using BRG.LifeMOD*, Development of Biomedical Engineering in Vietnam, Ho Chi Minh City, Vietname, vol. 27, pp. 48-50, 2010.

[ScanArm, 2012] L. ScanArm, *FARO Laser ScanArm V3*, http://measuring-arms.faro.com/ (retrieved on March 2012), 2012.

[Scharstein, 2002] D. Scharstein and R. Szeliski, *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*, Computer Vision, vol. 47, n. 1, pp. 7-42, 2002.

[SECA, 2007] SECA, *Measuring Tape for Head Circumference, SECA - Precision for Health*, http://www.medisave.co.uk/seca-measuring-tape-for-head-circumference-baby-infant-p-1414.html (retrieved in October 2007), 2007.

[Seitz, 1997] S. Seitz and C. R. Dyer, *Photorealistic Scene Reconstruction by Voxel Coloring*, Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 1067-1073, 1997.

[Seitz, 1999] S. M. Seitz, *An Overview of Passive Vision Techniques*, SIGGRAPH 2000 Course on 3D Photography, Course Notes, New Orleans, Louisiana, LA, USA, 1999.

[Seo, 1998] Y. Seo and K. S. Hong, *Auto-Calibration of a Rotating and Zooming Camera*, Machine Vision Applications, Makuhari, Chiba, Japan, pp. 274-277, 1998.

[Seo, 1999] Y. Seo and K. S. Hong, *About the Self-Calibration of a Rotating and Zooming Camera: Theory and Practice*, Computer Vision, Corfu, Greece, pp. 183-189, 1999.

[Seo, 2001] Y. Seo and K. S. Hong, *Theory and Practice on the Self-Calibration of a Rotating and Zooming Camera from Two Views*, Vision, Image and Signal Processing, vol. 148, n. 3, pp. 166-172, 2001.

[ShapeTape™, 2007] ShapeTape™, *Motion Capture By Measurand*, http://www.measurand.com/products/ShapeTape.html (retrieved in October 2007), 2007.

[Sharp, 2008] T. Sharp, *Implementing Decision Trees and Forests on a GPU*, Microsoft Research, Cambridge, United Kingdom, 2008.

[Shu, 2003] C. Shu, A. Brunton and M. Fiala, *Automatic Grid Finding in Calibration Patterns Using Delaunay Triangulation*, Technical Report, Computational Video Group, Institute for Information Technology, National Research Council, Ontario, Canada, 2003.

[Simmons, 2003] K. P. Simmons and C. L. Istook, *Body Measurement Techniques: Comparing 3D Body-Scanning and Anthroprometric Methods for Apparel Applications*, Fashion Marketing and Management, vol. 7, n. 3, pp. 306-332, 2003.

[Singare, 2010] S. Singare, Z. Shouyan, X. Guanghui, W. Weiping and Z. Jianjun, *The Use of Laser Scanner and Rapid Prototyping to Fabricate Auricular Prosthesis*, E-Product E-Service and E-Entertainment, Henan, China, 2010.

[Slabaugh, 2002] G. Slabaugh, *Novel Volumetric Scene Reconstruction Methods for New View Synthesis*, PhD Thesis, Georgia Institute of Technology, 2002.

[Song, 2009] P. Song, X. Wu and M. Y. Wang, *A Robust and Accurate Method for Visual Hull Computation*, Information and Automation, Zhuhai, Macau, China, pp. 784-789, 2009.

[Sousa, 2007] D. S. S. Sousa, J. M. R. S. Tavares, M. V. Correia, J. G. Barbosa, C. V. Chã and E. Mendes, *Registration between Data from Visual Sensors and Force Platform in Gait Event Detection*, Measurement Analysis and Modeling of Human Function, Cascais, Portugal, pp. 331-340, 2007.

[Starck, 2003] J. Starck and A. Hilton, *Model-Based Multiple View Reconstruction of People*, Computer Vision, Nice, France, vol. 2, pp. 915-922, 2003.

[Starck, 2005] J. Starck, G. Miller and A. Hilton, *Video-Based Character Animation*, Computer Animation, Los Angeles, CA, USA, pp. 49-58, 2005.

[Starck, 2007] J. Starck and A. Hilton, *Surface Capture for Performance-Based Animation*, Computer Graphics and Applications, vol. 27, n. 3, pp. 21-31, 2007.

[Starck, 2009] J. Starck, A. Maki and S. Nobuhara, *The Multiple-Camera 3-D Production Studio*, Circuits and Systems for Video Technology, vol. 19, n. 6, pp. 856-869, 2009.

[Stein, 1993] G. P. Stein, *Internal Camera Calibration using Rotation and Geometric Shapes*, Technical Report: AITR-1426, Massachusetts Institute of Technology, Cambridge, MA, USA, 1993.

[Stein, 1995] G. P. Stein, *Accurate Internal Camera Calibration using Rotation, with Analysis of Sources of Error*, Computer Vision, Cambridge, Massachusetts, MA, USA, pp. 230-236, 1995.

[Sturm, 1997a] P. Sturm, *Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction*, Computer Vision and Pattern Recognition, Puerto Rico, PR, USA, pp. 1100-1105, 1997a.

[Sturm, 1997b] P. Sturm, *Self-Calibration of a Moving Zoom-Lens Camera by Pre-Calibration*, Image and Vision Computing, vol. 15, n. 8, pp. 583-589, 1997b.

[Suresh, 2007] K. V. Suresh, G. M. Kumar and A. N. Rajagopalan, *Superresolution of License Plates in Real Traffic Videos*, Intelligent Transportation Systems, vol. 8, n. 2, pp. 321-331, 2007.

[Szeliski, 1993] R. Szeliski, *Rapid Octree Construction from Image Sequences*, Computer Vision, Graphics and Image Processing: Image Understanding, vol. 58, n. 1, pp. 23-32, 1993.

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

[Takahashi, 2006] K. Takahashi, Y. Nagasawa and M. Hashimoto, *Remarks on 3D Human Body Posture Estimation System Using Simple Multi-Camera System*, Industrial Electronics Society, Paris, France, pp. 3284-3289, 2006.

[Taubin, 2000] G. Taubin, *Geometric Signal Processing on Polygonal Meshes*, Eurographics - State of the Art Report, Interlaken, Switzerland, 2000.

[Tavares, 2003] J. M. R. S. Tavares, J. Barbosa and A. J. Padilha, *Modelos Deformáveis em Imagem Médica*, Internal Report, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2003.

[Tavares, 2008] P. Tavares, *Caracterização Tridimensional de Geometrias utilizando Campos de Luz Estruturada*, PhD Thesis, Departamento de Engenharia Mecânica e Gestão Industrial, FEUP, Porto, Portugal, 2008.

[Teodoro, 2009] P. Teodoro, P. Pires, J. Martins and J. S. d. Costa, *Proximal Femur Parametrization and Contact Modelling for Total Hip Resurfing Surgery*, Biomechanics, Bragança, Portugal, pp. 71-76, 2009.

[Terzopoulos, 1987] D. Terzopoulos, J. Platt, A. Barr and K. Fleischert, *Elastically Deformable Models*, Computer Graphics, vol. 21, n. 4, pp. 205-214, 1987.

[Terzopoulos, 2011] D. Terzopoulos, *Deformable and Functional Models*, Comoutational Vision and Medical Image Processing, Computational Methods in Applied Sciences, vol. 19, pp. 125-143, 2011.

[Thalmann, 1993] D. Thalmann, *Human Modelling and Animation*, Eurographics State of the Art Reports, 1993.

[Thompson, 1991] A. M. Thompson, J. C. Brown, J. W. Kay and D. M. Titterington, *A Study of Methods of Choosing the Smoothing Parameter in Image Restoration by*

Regularization, Pattern Analysis and Machine Intelligence, vol. 13, n. 4, pp. 326-339, 1991.

[Thormählen, 2008] T. Thormählen and H.-P. Seidel, *3D-Modeling by Ortho-Image Generation from Image Sequences*, ACM Transactions on Graphics, SIGGRAPH, vol. 27, n. 3, 2008.

[Tocheri, 2009] M. W. Tocheri, *Laser Scanning: 3D Analysis of Biological Surfaces*, Advanced Imaging in Biology and Medicine, Techonology, Software Environments, Applications, Springer, ch. 4, 2009.

[Tong, 2012] J. Tong, J. Zhou, L. Liu, Z. Pan and H. Yan, *Scanning 3D Full Human Bodies Using Kinects*, Visualization and Computer Graphics, vol. 18, n. 4, pp. 643-650, 2012.

[Toshio, 2005] U. Toshio, O. Takayuki and S. Tomokazu, *A Survey of Camera Calibration Techniques*, SIG Technical Reports, Information Processing Society of Japan vol. 18, pp. 1-18, 2005.

[Tran, 2012] C. Tran, *3-D Posture and Gesture Recognition for Interactivity in Smart Spaces*, Industrial Informatics, vol. 8, n. 1, pp. 178-187, 2012.

[Triggs, 1997] B. Triggs, *Autocalibration and the Absolute Quadric*, Computer Vision and Pattern Recognition, Puerto Rico, PR, USA, pp. 609-614, 1997.

[Tsai, 1986] R. Y. Tsai, *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision*, Computer Vision and Pattern Recognition, Miami Beach, FL, USA, pp. 364-374, 1986.

[Tsai, 1987] R. Y. Tsai, *A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology using Off-the-shelf TV Cameras and Lenses*, Robotics and Automation, vol. 3, n. 4, pp. 323-344, 1987.

[Vedula, 2000] S. Vedula, S. Baker, S. Seitz and T. Kanade, *Shape and Motion Carving in 6D*, Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, vol. 2, pp. 592-598, 2000.

[Viéville, 1994] T. Viéville, *Autocalibration of Visual Sensor Parameters on a Robotic Head*, Image and Vision Computing, vol. 12, n. 4, pp. 227-237, 1994.

[Wang, 2012] D. Wang and A. H. Tewfik, *Real Time 3D Visualization of Intraoperative Organ Deformations Using Structured Dictionary*, Medical Imaging, vol. 31, n. 4, pp. 924-937, 2012.

[Wang, 2003] L. Wang, W. Hu and T. Tan, *Recent Developments in Human Motion Analysis*, Pattern Recognition, vol. 36, n. 3, pp. 585-601, 2003.

[Wang, 2011] Y. Wang and S. Zhang, *Superfast Multifrequency Phase-shifting Technique with Optimal Pulse Width Modulation*, Optics Express, vol. 16, n. 6, pp. 5149-5155, 2011.

[Wang, 2010] Z. Wang, Z. Wang and Y. Wu, *Recognition of Corners of Planar Checkboard Calibration Pattern Image*, Control and Decision, Xuzhou, China, pp. 3224-3228, 2010.

[Weise, 2011] T. Weise, S. Bouaziz, H. Li and M. Pauly, *Realtime Performance-Based Facial Animation*, ACM Transactions on Graphics, Vancouver, Canada, 2011.

[Weiss, 2011] A. Weiss, D. Hirshberg and M. J. Black, *Home 3D Body Scans from Noisy Image and Range Data*, Computer Vision, Barcelona, Spain, 2011.

[Wen, 1991] J. Wen and G. Schweitzer, *Hybrid Calibration of CCD Cameras using Artificial Neural Nets*, Neural Networks, Singapore, Republic of Singapore, vol. 1, pp. 337-342, 1991.

[Weng, 1992] J. Weng, P. Cohen and M. Herniou, *Camera Calibration with Distortion Models and Accuracy Evaluation*, Pattern Analysis and Machine Intelligence, vol. 14, n. 10, pp. 965-980, 1992.

[Willson, 1994] R. G. Willson, *Modeling and Calibration of Automated Zoom Lenses*, PhD Thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, PA, USA, pp. 170-186, 1994.

[Wong, 2010] S. S. Wong and K. L. Chan, *3D Object Model Reconstruction from Image Sequence based on Photometric Consistency in Volume Space*, Pattern Recognition and Image Analysis, vol. 13, n. 4, pp. 437-450, 2010.

[Xi, 2007] P. Xi, W.-S. Lee and C. Shu, *Analysis of segmented human body scans*, Graphics Interface, Montreal, Canada, vol. 234, pp. 19-26, 2007.

[Yang, 1998] C. Yang, W. Wang and Z. Hu, *An Active Vision Based Self-Calibration Technique*, Chinese Journal of Computers, vol. 5, pp. 428-435, 1998.

[Yilmaz, 2002] U. Yilmaz, A. Y. Mülayim and V. Atala, *Reconstruction of Three Dimensional Models from Real Images*, 3D Data Processing Visualization and Transmission, Padova, Italy, pp. 554-557, 2002.

[Ying, 2008] X. Ying and H. Zha, *Efficient Detection of Projected Concentric Circles using Four Intersection Points on a Secant Line*, Pattern Recognition, Tampa, Florida, USA, pp. 1-4, 2008.

[Yu, 2010] W. Yu and B. Xu, *A Portable Stereo Vision System for Whole Body Surface Imaging*, Image Vision Computing, vol. 28, n. 4, pp. 605-613, 2010.

[Yu, 1998] Y. Yu and J. Malik, *Recovering Photometric Properties of Architectural Scenes from Photographs*, Computer Graphics, vol. 32, pp. 207-217, 1998.

[Yunfeng, 2010] L. Yunfeng, D. Xingtao and P. Wei, *Study on Digital Data Processing Techniques for 3D Medical Model*, Bioinformatics and Biomedical Engineering, Chengdu, China, 2010.

[Zabulis, 2010] X. Zabulis, T. Sarmis, K. Tzevanidis, P. Koutlemanis, D. Grammenos and A. A. Argyros, *A Platform for Monitoring Aspects of Human Presence in Real-time*, Advances in Visual Computing, Las Vegas, NV, USA, 2010.

[Zeller, 1996] C. Zeller and O. Faugeras, *Camera Self-Calibration from Video Sequences: the Kruppa Equations Revisited*, Research Report 2793, INRIA Sophia Antipolis, France, 1996.

[Zhang, 2006] G. Zhang, H. Zhang and K.-Y. K. Wong, *1D Camera Geometry and Its Application to Circular Motion Estimation*, Machine Vision, Edinburgh, United Kingdom, vol. 1, pp. 67-76, 2006.

[Zhang, 2005] H. Zhang, G. Zhang and K.-Y. K. Wong, *Auto-Calibration and Motion Recovery from Silhouettes for Turntable Sequences*, Machine Vision, Oxford Brookes University, Oxford, United Kingdom, vol. 1, pp. 79-88, 2005.

[Zhang, 1995] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry*, Artificial Intelligence Journal, vol. 78, pp. 87-119, 1995.

[Zhang, 1998] Z. Zhang, *A Flexible New Technique for Camera Calibration*, Technical Report MSR-TR-98-71, Microsoft Research, Microsoft Corporation, Redmond, Washington, WA, USA, 1998.

[Zhang, 2000] Z. Zhang, *A Flexible New Technique for Camera Calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, n. 11, pp. 1330-1334, 2000.

[Zhao, 1994] C. Zhao and R. Mohr, *Global Three-Dimensional Surface Reconstruction from Occluding Contours*, Computer Vision and Image Understanding, vol. 64, n. 1, pp. 62-96, 1994.

[Zhao, 2008] X. Zhao and Y. Liu, *Generative tracking of 3D human motion by hierarchical annealed genetic algorithm*, Pattern Recognition, vol. 41, n. 8, pp. 2470-2483, 2008.

[Zheng, 1994] J. Y. Zheng, *Acquiring 3-D Models from Sequences of Contours*, Pattern Analysis and Machine Intelligence, vol. 16, n. 2, pp. 163-178, 1994.

[Zhou, 1992] Y.-T. T. Zhou and R. Chellappa, *Artificial Neural Networks for Computer Vision*, Springer-Verlag, New York, NY, USA, 1992.

[Zhu, 2010] G. Zhu and X. Li, *Reduction and Hole-repairing of 3D Scan Line Point Clouds Based on the Portable Body Measurement System*, Computer and Automation Engineering, Singapore, vol. 4, pp. 310-313, 2010.

[Zollhöfer, 2011] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger and J. Süßmuth, *Automatic Reconstruction of Personalized Avatars from 3D Face Scans*, Computer Animation and Virtual Worlds, vol. 22, n. 2-3, pp. 195-202, 2011.

[Zuo, 2011] P. Zuo and Y. Zhao, *A Design of 3D Modeling Virtual Fitting Project for Online Shopping*, Industrial Engineering and Engineering Management, Singapore, 2011.