Methods and applications of “sensor fusion” for mechatronic systems

Metodi e applicazioni di sensor fusion per sistemi meccatronici

Francesco Setti

January 28, 2010

To Stefania

Acknowledgments

It is with great pleasure that I am here, once again, to thank those who made this possible. First I want to thank my advisor, prof. Mariolino De Cecco; already three years ago he believed in me, he gave me the opportunity to reach this new goal and has always guaranteed its support.

For the same support I must also be grateful to all my colleagues in the Measurement Group and, more generally, in the Mechatronics Group at the University of Trento; to list all them would take too long, thanks collectively!

This work was partially supported by Delta R&S, thanks to them, which I hope does not end there.

Many thanks to Alessio Del Bue and all the "Lisboetas" for three intense months of work, but not only...

I owe a special thanks to those who accompanied me in these years: Matteo and Bicio. Like three years ago, we are, this time away, to celebrate a new degree.

I would also like to thank my parents Silvano and Patrizia, my sister Chiara, Robby and all my relatives for the support they provided me through my entire life and in particular in these last three years.

I can not forget Marco, Elena and Andrea who are now part of my family.

I must acknowledge my best friends Matteo, Elena, Fabio, Anna, Roberto and Veronica simply for being with me.

However, remember that the greatest acknowledgment is not a thanks but a dedication!

Abstract

In this paper we present a system for 3D shapes digitization. During the research we developed algorithms for 3D shapes reconstruction from multiple stereo images; this algorithms provide the use of colored markers lying over the object surface. The procedure developed provides steps of image processing for markers detection, 3D reconstruction based on epipolar geometry and data fusion from different stereo-pairs. Particular attention was paid to the latter point, developing an algorithm based on uncertainty analysis of each 3D point and compatibility analysis by the Mahalanobis distance. Were then developed algorithms for modeling 3D objects in the scene based on a images sequence of moving bodies. A physical prototype was created at the MeccaLab at University of Trento and it has been used for an experimental verification of the proposed algorithms.

Sommario

In questo lavoro viene presentato un sistema di digitalizzazione di forme 3D. Nel corso della ricerca sono stati sviluppati degli algoritmi per la ricostruzione di forme 3D a partire da immagini acquisite da più stereocamere; questi algoritmi prevedono l'utilizzo di marker colorati posizionati sull'oggetto da digitalizzare. La procedura sviluppata prevede fasi di elaborazione delle immagini per la localizzazione dei marker, ricostruzione 3D basata sulla geometria epipolare e fusione dei dati provenienti dalle diverse stereocamere. Particolare attenzione è stata posta a quest'ultimo punto, sviluppando un algoritmo basato su analisi dell'incertezza di ogni punto e analisi di compatibilità dei punti mediante distanza di Mahalanobis. Vengono poi proesentati degli algoritmi per la modellazione degli oggetti 3D presenti nella scena sulla base di una sequenza di immagini dei corpi in movimento. Un prototipo fisico è stato realizzato presso il MeccaLab dell'Università di Trento ed è stato utilizzato per una fase sperimentale di verifica degli algoritmi proposti.

Contents

List of Figures

Chapter 1

Introduction

A 3D scanner is a device that analyzes a real-world object or environment to collect data on its shape and, if interesting, its appearance (e.g. color). The collected data can then be used to construct digital, three dimensional models useful for a wide variety of applications. These devices are used extensively by the entertainment industry, in the production of movies and video games, but also in industrial design, orthotics and prosthetics, reverse engineering and prototyping, quality control/inspection and documentation of cultural artifacts.

The purpose of a 3D scanner is usually to create a point cloud of geometric samples on the surface of the subject. These points can then be used to extrapolate the shape of the subject. Many different technologies can be used to build these 3D scanning devices; each technology comes with its own limitations, advantages and costs.

For most situations, a single scan will not produce a complete model of the subject. Multiple scans, even hundreds, from many different directions are usually required to obtain information about all sides of the subject. These scans have to be brought in a common reference frame, a process that is usually called alignment or registration, and then merged to create a complete model.

There are two types of 3D scanners: contact and non-contact. Non-contact 3D scanners can be further divided into two main categories, active scanners and passive scanners. There are a variety of technologies that fall under each of these categories.

Contact 3D scanners probe the subject through physical touch. A *Coordinate Measuring Machine* (CMM)(Spitz, 1999) is an example of a contact 3D scanner. It is used mostly in manufacturing and can be very precise. The disadvantage of CMMs though, is that it requires contact with the object being scanned. Thus, the act of scanning the object might modify or damage it. This fact is very significant when scanning delicate or valuable objects such as historical artifacts. The other disadvantage of CMMs is that they are relatively slow compared to the other scanning methods. Physically moving the arm that the probe is mounted on can be very slow and the fastest CMMs can only operate on a few hundred hertz.

Non-contact active scanners emit some kind of radiation or light and detect its reflection in order to probe an object or environment. Possible types of emissions used include light, ultrasound or x-ray. Some examples of non-contact active devices are time-of-flight laser scanner, triangulation laser scanner and structured light.

The time-of-flight 3D laser scanner is an active scanner that uses laser light to probe the subject. At the heart of this type of scanner is a time-of-flight laser rangefinder. The laser rangefinder finds the distance of a surface by timing the round-trip time of a pulse of light. A laser is used to emit a pulse of light and the amount of time before the reflected light is seen by a detector is timed. Since the speed of light c is known, the round-trip time determines the travel distance of the light, which is twice the distance between the scanner and the surface. The accuracy of a time-of-flight 3D laser scanner depends on how precisely we can measure the time. The laser rangefinder only detects the distance of one point in its direction of view. Thus, the scanner scans its entire field of view one point at a time by changing the range finder's direction of view to scan different points. The view direction of the laser rangefinder can be changed by either rotating the range finder itself, or by using a system of rotating mirrors. The latter method is commonly used because mirrors are much lighter and can thus be rotated much faster and with greater accuracy. Typical time-of-flight 3D laser scanners can measure the distance of $10.000 \div 100.000$ points every second.

The triangulation 3D laser scanner is also an active scanner that uses laser light to probe the environment. With respect to time-of-flight 3D laser scanner the triangulation laser shines a laser on the subject and exploits a camera to look for the location of the laser dot. Depending on how far away the laser strikes a surface, the laser dot appears at different places in the camera's field of view. This technique is called triangulation because the laser dot, the camera and the laser emitter form a triangle. The length of one side of the triangle, the distance between the camera and the laser emitter is known. The angle of the laser emitter corner is also known. The angle of the camera corner can be determined by looking at the location of the laser dot in the camera's field of view. These three pieces of information fully determine the shape and size of the triangle and gives the location of the laser dot corner of the triangle. In most cases a laser stripe, instead of a single laser dot, is swept across the object to speed up the acquisition process. The National Research Council of Canada was among the first institutes to develop the triangulation based laser scanning technology in

1978(Mayer, 1999).

Structured-light 3D scanners project a pattern of light on the subject and look at the deformation of the pattern on the subject. The pattern may be one dimensional or two dimensional. An example of a one dimensional pattern is a line. The line is projected onto the subject using either an LCD projector or a sweeping laser. A camera, offset slightly from the pattern projector, looks at the shape of the line and uses a technique similar to triangulation to calculate the distance of every point on the line. In the case of a single-line pattern, the line is swept across the field of view to gather distance information one strip at a time. An example of a two-dimensional pattern is a grid or a line stripe pattern. A camera is used to look at the deformation of the pattern, and an algorithm is used to calculate the distance at each point in the pattern. Structured-light scanning is still a very active area of research. The advantage of structured-light 3D scanners is speed. Instead of scanning one point at a time, structured light scanners scan multiple points or the entire field of view at once. This reduces or eliminates the problem of distortion from motion. Some existing systems are capable of scanning moving objects in real-time(Zhang and Yau, 2006).

Non-contact passive scanners do not emit any kind of radiation themselves, but instead rely on detecting reflected ambient radiation. Most scanners of this type detect visible light because it is a readily available ambient radiation. Other types of radiation, such as infrared could also be used. Passive methods can be very cheap, because in most cases they do not need particular hardware.

Stereoscopic systems usually employ two video cameras, slightly apart, looking at the same scene. By analyzing the slight differences between the images seen by each camera, it is possible to determine the distance at each point in the images. This method is based on human stereoscopic vision(Young, 1994).

Silhouette 3D scanners use outlines created from a sequence of photographs around a three-dimensional object against a well contrasted background. These silhouettes are extruded and intersected to form the visual hull approximation of the object. With these kinds of techniques some kind of concavities of an object (like the interior of a bowl) are not detected.

In this work we are interested in the development of a 3D scanner for the reconstruction of human body's parts, like hands or feet. We need a system able to reconstruct a 3D object with a “single shot”, for this reason we choose a vision system.

This kind of systems are nowadays widely used in 3D shape reconstruction because of its flexibility. However, the increasing resolution of digital image sensors is bringing actual measurement performance toward limits that were not available until few years ago.

As regards hardware setup, the main difference is that between *multi-camera* and *multi-stereo*. In the first approach the cameras are located uniformly in the space and each camera is considered as associated with each other; in the second one the system reconstruct shapes by associating camera couples. The use of multiple pairs of cameras allows the reconstruction of different portions, visible to each pair and partially overlapping. Compared with the multi-camera procedure, this approach allows a better match between the two views, which are commonly very closed to each other. However, the short baseline is prone to high depth uncertainty. In order to increase shape accuracy, the different parts can be matched by means of *Iterative Closest Points* (ICP) methods (Trucco et al., 1999; Eggert et al., 1997) and then, for each point, a compatibility analysis can be performed with their neighbors in order to fuse each estimate coming from different couples.

Several methods can be used to match the information on different cameras: shape detection, edge detection, correlation analysis, marker matching and others. Celenk et al. (Celenk and Bachnak, 1990) describes a method for surface reconstruction that employs a Lagrangian polynomial for surface initialization and a quadratic variation method to improve the results. In Esteban et al. (Hernandez Esteban and Schmitt, 2002) they recovers a first approximation of the shape through the object silhouettes seen by the multiple cameras, and then the shape is improved by a carving approach, employing local correlation analysis between images taken by different cameras. This approach is based on the hypothesis that, if a 3D point belongs to the object surface, its projection into the different cameras which really see it will be closely correlated. Nedevschi et al. (Nedevschi et al., 2004) presents a method for spatial grouping by a multiple stereo system. The grouping algorithm comprises a 3D space compressing step in order to map the 3D points into a space of even density, that allows a easier grouping by a neighborhood approach; a subsequent decompressing step preserves the adjacencies of the compressed space and helps the fusion of grouped points seen by different cameras.

Each step of the measurement process is affected by uncertainty, which propagates to the final 3D estimates. Uncertainty sources are different, such as digitalization and noise in image acquisition, feature extraction algorithm and intrinsic and extrinsic calibration parameters. One drawback of the above approaches is that they do not evaluate the uncertainty of the reconstructed object. When system is used to perform 3D measurements, a region of confidence of the measured 3D points should be evaluated to a desired level of confidence. In this work we present a method that develops a symbolic uncertainty estimation that merges the measurements performed with different stereo pairs and yields the uncertainty associated with the measured quantities.

In Chen et al. (Chen et al., 2008) an uncertainty analysis is presented for a binocular stereo reconstruction, but it does not describe a method to compare and fuse the measurements of different stereo pairs. In addition, in our method the covariance of the parameters estimated during the calibration phase is obtained by means of a Monte Carlo simulation avoiding linearization; correlation between the different parameters is analyzed in depth, giving rise to a covariance that can be considered sparse but not simply diagonal, as in (Chen et al., 2008).

A method which takes uncertainty into consideration in order to choose the best combination of camera pairs for stereo triangulation is described in Amat et al. (Amat et al., 2002). In this case, however, the uncertainties associated with the intrinsic and extrinsic camera calibration parameters are not taken into account, and a simplified geometrical uncertainty estimation and a propagation algorithm that uses scalar instead of vector quantities is employed. In this way, cross-correlation between the different sources of uncertainty are neglected.

The interest on the extension of 3D reconstruction to motion analysis is growing due to the wide application of these systems in different industrial and scientific fields. The recent advancements in Computer Vision have impacted highly in the movie and advertisement industries (Boujou, 2009), in the medical analysis area, in video-surveillance applications (Ioannidis et al., 2007) and in biomechanics studies of the human body (Corazza et al., 2007; Fayad et al., 2009). However, the strongest limitation for several systems is their restriction to deal with rigid bodies only. A shape which is deforming introduces new challenges, the object can vary arbitrary and the observed shape may have different articulations not known a priori. How to model and identify such variations is still an open issue even if successful systems are already available in the market (OrganicMotion, 2009; Vicon, 2009).

In this work we present a multi-stereo system for the 3D scanning of anatomical parts. Particular interest will be on the problem of position fusion of 3D points reconstructed from more stereo-pairs. In the second part we address the problem of motion segmentation and joint parameters reconstruction, applied to human body in order to build automatically a human body model.

The present work will be organized in 3 parts that discuss the theory of computer vision, the innovative proposed algorithms and the experiments for the validation of the algorithms.

In Part I we discuss an introduction to the mathematical models used for computer vision and the calibration of its the parameters (Chapter 2), and the geometry and mathematical methods used for stereo-vision (Chapter 3).

In Part II we discuss a detailed description of the algorithms for the 3D reconstruction of static objects (Chapter 4) and for the extension to motion segmentation and joint reconstruction (Chapter 5).

In Part III we present the developed experimental set-up and the experiments provided for the experimental verification of the proposed algorithms (Chapter 6).

Part I

Introduction to computer vision

Chapter 2

Camera model

In this chapter we introduce a mathematical model of the geometry of image formation process. We are now interested in the discussion of what is an image and how it is formed, in the reference frames that will be used in the following chapters and in a rigorous description of the notation and conventions. In the second part of the chapter will be presented the parameters used to characterize the camera, related to the previously presented camera model.

2.1 What's an image?

An image is a two-dimensional brightness array, and we talk about a *gray level image*, or a set of three such array, and we talk about a *RGB image* (red, green and blue). In other words the image I is a map defined on a compact region Ω of a two-dimensional surface, taking values in the positive real numbers. In a camera Ω is a planar rectangular region occupied by the CCD[1] sensor. So I is a function:

$$I \, : \, \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}_+ \, ; \;\; (x, y) \longmapsto I(x, y) \qquad (2.1)$$

In the case of a digital image, both the domain Ω and the range $\mathbb{R}_+$ are discretized. For instance, $\Omega = [1, 640] \times [1, 480] \subset \mathbb{Z}^2$, and $\mathbb{R}_+$ is an interval of integers $[0, 255] \subset \mathbb{Z}_+$ (in this case we talk about VGA resolution and 8-bit encoding).

The values of the image I depend upon physical properties of the scene being viewed, such as:

- the shape;
- the material reflectance properties;
- the distribution of the light sources (Forsyth and Ponce, 2002).

[1] In this work we always refer to "CCD sensor" but in general it could be a CMOS sensor or a photographic film

2.2 The thin lens model

A camera is composed by a set of lenses used to direct the light toward the CCD. Thereby we perform a change of direction of propagation of the light, using the properties of diffraction, refraction and reflection of a glass. For simplicity we neglect the effects of diffraction and reflection in a lens system, and consider only the refraction; for more details about the lenses models see (Born and Wolf, 1999). Therefore we first consider the *thin lens* model.

A *thin lens* (see Figure 2.1) is a mathematical model defined by an axis (*optical axis*) and a plane perpendicular to the axis (*focal plane*) with a circular aperture centered at the intersection between the optical axis and the focal plane (*optical center*).

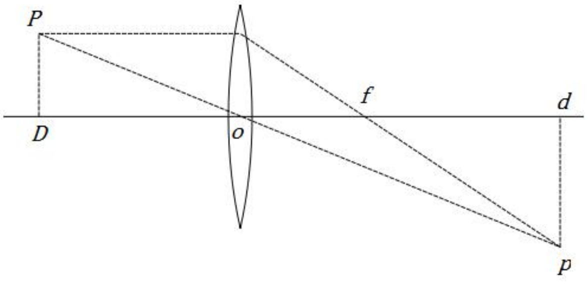


Figure 2.1: Graphic representation of the *thin lens model*.

The thin lens model has two parameters: the *focal length* (f) and the *diameter* (d). Its function is characterized by two properties:

- all rays entering the aperture parallel to the optical axis intersect on the optical axis at a distance f from the optical center; the point of intersection is called *focus* of the length.

- all rays through the optical center are undeflected.

Consider a point $\mathbf{P} \in \mathbb{R}^3$ at a distance D along the optical axis from the optical center. We can draw two rays from the point $\mathbf{P}$: the first one is parallel to the optical axis until the aperture and then intersect the optical axis at the focus; the second one through the optical center and remains undeflected. We can call $\mathbf{p}$ the point where the two rays intersect, and let d be the distance from the optical center along the optical axis. Using similar triangles we can obtain the *fundamental equation of the thin lens*:

$$\frac{1}{D} + \frac{1}{d} = \frac{1}{f} \tag{2.2}$$

Therefore the irradiance $I(\mathbf{x})$ at the point $\mathbf{x}$ on the image plane is obtained by integrating all the energy emitted from the region of space that project on the point $\mathbf{x}$, compatibly with the geometry of the lens.

2.3 The pinhole model

If we let the aperture of the thin lens decrease to zero, all rays are forced to go through the optical center ($\mathbf{o}$), remaining undeflected. Consequently the only points that contribute to the irradiance at the image point $\mathbf{p} = [x, y]^T$ are on a line through the points $\mathbf{p}$ and $\mathbf{o}$.

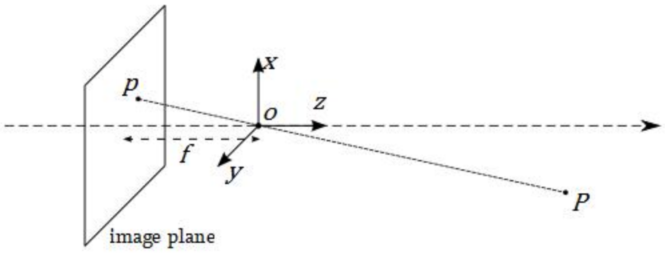


Figure 2.2: Graphic representation of the *pinhole model*.

Let us consider a point $\mathbf{P} = [X, Y, Z]^T$, relative to a reference frame centered at the optical center $\mathbf{o}$ with the z-axis parallel to the optical axis (see Figure 2.2), from similar triangles we can see that the coordinates of $\mathbf{P}$ and its image $\mathbf{p}$ are related by the so-called *ideal perspective projection*:

$$x = -f\frac{X}{Z} \ , \ y = -f\frac{Y}{Z} \tag{2.3}$$

We can also write the projection as a map π:

$$\pi \ : \ \mathbb{R}^3 \to \mathbb{R}^2 \ ; \ \mathbf{X} \longmapsto \mathbf{x} \tag{2.4}$$

This model, called *ideal pinhole camera model*, is an idealization of the thin lens model when the aperture decrease to zero. Note that in this conditions the diffraction effects become dominant and therefore the thin lens model does not hold (Born and Wolf, 1999). Furthermore, as the aperture decrease to zero, the energy going through the lens also become zero. Otherwise, today it is possible to build devices that approximate the pinhole model, and so we can consider this model just as a good geometric approximation.

Notice that in this model we have a negative sign in each of the formula (2.3); this makes the image to be upside down. In order to eliminate this effect we introduce the *frontal pinhole camera model*, placing the image plane in front of the optical center (see Figure 2.3). This corresponds to flip the image: $(x, y) \mapsto (-x, -y)$. In this case, the image $\mathbf{p} = [x, y]^T$ of the point $\mathbf{P}$ is given by:

$$x = f\frac{X}{Z} \ , \ y = f\frac{Y}{Z} \tag{2.5}$$

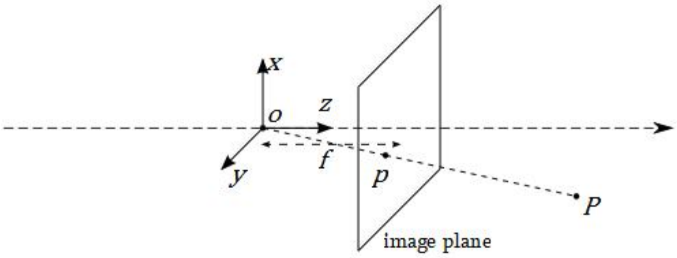


Figure 2.3: Graphic representation of the *frontal pinhole model*.

For the present work we used always the frontal pinhole model; the interested readers can refer to (Ma et al., 2003) and (Forsyth and Ponce, 2002) for more details.

2.4 Reference frames and conventions

As described in (Ma et al., 2003), a camera can be modeled by a line (*optical axis*), a point belonging to the line (*optical center*) and a plane orthogonal to the optical axis at a distance f from the optical center (*image plane*)[2].

As shown in Figure 2.4, in the rest of the work we refer to three kinds of reference frames:

world it's a global reference frame, it can be related to other devices in the workspace or to the calibration set-up (Horn, 2000).

camera it's a reference frame with origin in the optical center ($\mathbf{o}$), z-axis aligned with the optical axis and x-axis aligned with the columns of the CCD sensor.

image it's a reference frame with origin in the upper left corner of the CCD sensor, x-axis aligned with the rows in the CCD and y-axis aligned with the columns in the CCD.

[2] We are now referring to the *frontal pinhole model*, a more detailed description of the model parameters will be discuss below.

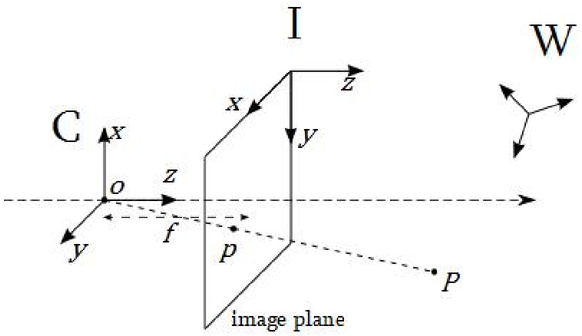


Figure 2.4: The ideal pinhole camera model with all the reference frames used in this work. Notice the *world* reference frame (W), the *camera* reference frame (C) and the *image* reference frame (I).

If we consider a set of points $\mathbf{P}$ in the space, we refer to the i-th point with express in the reference frame F with the symbol:

$$^{F}\mathbf{P}_i$$

2.5 Geometric model of projection

Considering the just described frontal pinhole camera model, the generic position of a point comprised in the field of view of a camera is given by:

$$^{C}\mathbf{P} = \lambda \, ^{C}\mathbf{p} \tag{2.6}$$

where $\lambda \in \mathbb{R}_{+}$ is a positive scalar parameter associated with the depth of the point.

The camera is characterized by a set of *intrinsic* calibration parameters, as described below, that defines the relationship between the camera reference frame and the image reference frame. Referring to the Figure 2.4, an ideal pinhole camera reveals the following direct model:

$$^{I}\mathbf{p} = \mathbf{K} \, ^{C}\mathbf{p} = \begin{bmatrix} 0 & f_m \cdot s & ^{I}x_0 \\ -f_m & 0 & ^{I}y_0 \\ 0 & 0 & 1 \end{bmatrix} \, ^{C}\mathbf{p} \tag{2.7}$$

and the inverse model becomes:

$$^{C}\mathbf{p} = \mathbf{K}^{-1} \, ^{I}\mathbf{p} = \begin{bmatrix} 0 & -\frac{1}{f_m} & \frac{^{I}y_0}{f_m} \\ \frac{1}{f_m \cdot s} & 0 & -\frac{^{I}x_0}{f_m \cdot s} \\ 0 & 0 & 1 \end{bmatrix} \, ^{I}\mathbf{p} \tag{2.8}$$

where $f_m = f \cdot S_x$; $s = \frac{S_y}{S_x}$; $S_x = \frac{pixels}{length\ unit}$ along x axis; $S_y = \frac{pixels}{length\ unit}$ along y axis[3]; (x_0, y_0) are the coordinates of the *principal point* of the CCD, defined as the intersection between the principal axis and the image plane.

2.6 Camera parameters

Usually when we refer to *camera calibration* we mean the recovery of the principal distance (or focal length) f and the principal point $(x_0, y_0)^T$ in the image plane; or, equivalently, recovery of the position of the center of projection $(x_0, y_0, f)^T$ in the camera reference frame. This is referred to as interior orientation in photogrammetry; for us these will be the *intrinsic parameters.*

A calibration target can be used to recover, from the correspondences between points in the space and points in the image, a relationship between the camera (or image) reference frame and the world reference frame. This is referred to as exterior orientation in photogrammetry, for us these will be the *extrinsic parameters.*

Since cameras often have appreciable geometric distortions, camera calibration is often taken to include the recovery of power series coefficients of these distortions. Furthermore, an unknown scale factor in image sampling may also need to be recovered, because scan lines are typically resampled in the frame grabber, and so picture cells do not correspond discrete sensing elements.

In this work we have developed a calibration procedure based on Tsai's method (Horn, 2000). This method for camera calibration recovers the intrinsic parameters, the extrinsic parameters, the power series coefficients for distortion, and an image scale factor that best fit the measured image coordinates, corresponding to known target point coordinates. This is done in stages, starting off with closed form least-squares estimates of some parameters and ending with an iterative non-linear optimization of all parameters simultaneously, using these estimates as starting values.

2.6.1 Intrinsic parameters

Interior Orientation is the relationship between camera reference frame and image reference frame. Camera coordinates and image coordinates are related by the matrix K in equation 2.7. As we can see, matrix K has four degrees of freedom. The problem of intrinsic parameters calibration is the recovery of x_0, y_0, f_m and s.

This is the basic task of camera calibration. However in practice we also need to recover the position and attitude of the calibration target in the

[3] S_x and S_y are defined with reference to the directions x and y in the camera reference frame and not in the image one.

camera coordinate system (extrinsic parameters).

2.6.2 Extrinsic parameters

Exterior Orientation is the relationship between world reference frame and camera reference frame. The transformation from world to camera consists of a rotation and a translation. This transformation has six degrees of freedom (three for rotation and three for translation). The world coordinate system can be any system convenient for the particular design of the target.

The relationship between these two reference frames is given by:

$$ {}^{W}\mathbf{P} ={}^{W} \mathbf{R}_C \, {}^{C}\mathbf{P} +{}^{W} \mathbf{T}_C \tag{2.9}$$

where ${}^{W}\mathbf{R}_C$ is the rotation matrix from camera reference frame to world reference frame, and ${}^{W}\mathbf{T}_C$ is the position of the origin of camera reference frame expressed in world frame.

The problem of extrinsic parameters calibration is the recovery of three rotation angles, used to generate the rotation matrix, and three coordinates of translation.

2.6.3 Distortion

Projection in an ideal imaging system is governed by the frontal pinhole model. Real optical systems suffer from a number of inevitable geometric distortions. In optical systems made of spherical surfaces, with centers along the optical axis, a geometric distortion occurs in the radial direction. A point is imaged at a distance from the principal point that is larger (*pincushion* distortion) or smaller (*barrel* distortion) than the predicted one by the perspective projection equations; the displacement increasing with distance from the center. It is small for directions that are near parallel to the optical axis, growing as some power series of the angle. The distortion tends to be more noticeable with wide-angle lenses than with telephoto lenses.

The displacement due to radial distortion can be modelled using the equations:

$$\begin{cases} \delta x = x(\kappa_1 r^2 + \kappa_2 r^4 + \ldots) \\ \delta y = y(\kappa_1 r^2 + \kappa_2 r^4 + \ldots) \end{cases} \tag{2.10}$$

where x and y are measured from the center of distortion, which is typically assumed to be at the principal point. Only even powers of the distance r from the principal point occur, and typically only the first, or perhaps the first and the second term in the power series are retained.

Equivalently we can express this distortion as function of r as:

$$\delta r = \kappa_1 r^3 + \kappa_2 r^5 + \ldots \tag{2.11}$$

Electro-optical systems typically have larger distortions than optical systems made of glass. They also suffer from tangential distortion, which is at right angle to the vector from the center of the image. Like radial distortion, tangential distortion grows with distance from the center of distortion.

$$\left\{ \begin{array}{l} \delta x = -y(\epsilon_1 r^2 + \epsilon_2 r^4 + \ldots) \\ \delta y = +x(\epsilon_1 r^2 + \epsilon_2 r^4 + \ldots) \end{array} \right. \tag{2.12}$$

In calibration, we attempt to recover the coefficients (κ_1, κ_2 , ..., ϵ_1, ϵ_2, ...) of these power series.

In this work we consider only the radial distortions because of the tangential distortions are negligible.

Chapter 3

Geometry of two cameras

A stereo system comprises two cameras, that we can call cam-*1* and cam-*2*, that frame the same field of view. The geometry of stereo systems is based on the *epipolar geometry.* In this chapter we describe the epipolar geometry and the triangulation process.

3.1 Epipolar geometry

Consider two images of the same scene taken from two distinct vantage points. If we assume that the cameras are calibrated, we know the position and orientation of the two camera frames with reference to the world frame (see Section 2.6), and therefore the relative orientation and translation between the two camera frames.

The intersections of the line $(\mathbf{o}_1, \mathbf{o}_2)$ with each image plane are called *epipoles* and denoted by $\mathbf{e}_1$ and $\mathbf{e}_2$ (see Figure 3.1). Consider a point $\mathbf{P}$ in the 3D space in the field of view of both cameras; we can define the *epipolar plane* as the plane through $\mathbf{P}$, $\mathbf{o}_1$ and $\mathbf{o}_2$. Notice that the projections of point $\mathbf{P}$ on the image planes belong to the rays $(\mathbf{o}_1, \mathbf{P})$ and $(\mathbf{o}_2, \mathbf{P})$, both belonging to the epipolar plane; therefore the projections belongs to the epipolar plane too.

The lines l_1 and l_2 are called *epipolar lines*, which are the intersections of the epipolar plane with the two image planes.

Mathematically this considerations are expressed by the *epipolar constraint*:

$$\langle \mathbf{p}_2, \mathbf{T} \times \mathbf{R}\mathbf{p}_1 \rangle = 0 \tag{3.1}$$

where $(\mathbf{R}, \mathbf{T})$ is the relative pose between the two cameras.

The power of this constraint is applied to the matching of feature points between the two images. Once we have the projection $\mathbf{p}_1$ of the point $\mathbf{P}$ on the image plane π_1, we have a description of the epipolar plane $(\mathbf{o}_1, \mathbf{o}_2, \mathbf{p}_1)$ and so we can compute the epipolar line on image plane π_2. The projection $\mathbf{p}_2$ of the point $\mathbf{P}$ on the image plane π_2 must belong to this line.

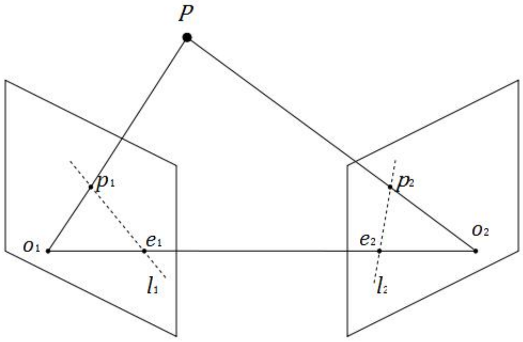


Figure 3.1: The epipolar geometry representation.

3.2 Stereo camera model

For each camera we can define a camera reference frame, as described in Section 2.4. Considering the camera model described in Section 2.5, the position of a point in the field of view of the i-th camera is given by:

$$ {}^{i}\mathbf{P} =^{i} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \lambda_i \, \mathbf{p}_i \tag{3.2} $$

where ${}^{i}\mathbf{P}$ is the point position expressed in the reference frame of cam-i; $\mathbf{p}_i$ is the projection of this point onto an ideal camera aligned with cam-i with a focal length equal to 1 (in length units)[1] and $\lambda_i \in \mathbb{R}_+$ is a scalar parameter associated with the depth of the point.

By using the intrinsic parameters of the camera model, evaluated during camera calibration, we can define the relation between projection $\mathbf{p}_i$, expressed in length units, and projection $\mathbf{p}_i\prime$, expressed in pixels, being $x_i\prime$ and $y_i\prime$ respectively the column and row number, from the upper left corner of the sensor. The ideal pinhole camera model give the following relationship:

$$ \begin{cases} x\prime &= f_m \cdot s \cdot y + x_0\prime \\ y\prime &= -f_m \cdot x + y_0\prime \end{cases} \tag{3.3} $$

and so:

$$ \mathbf{p}_i\prime = \begin{bmatrix} x_i\prime \\ y_i\prime \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & f_m \cdot s & x_{0i}\prime \\ -f_m & 0 & y_{0i}\prime \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \mathbf{K} \cdot \mathbf{p}_i \tag{3.4} $$

[1] This is usually called *rectified image.*

where $f_m = f \cdot S_x$; $s = \frac{S_y}{S_x}$; $S_x = \frac{pixels}{length\ unit}$ along x axis; $S_y = \frac{pixels}{length\ unit}$ along y axis[2].

3.3 Triangulation

The algorithm used for computing the depth of a point in the field of view of both cameras in a stereo-pair is called triangulation. In the following paragraphs we present the ideal case of triangulation and, later, the real case with the *middle point* approach. More complex approaches, like *epipolar optimization* algorithm, are described in (Ma et al., 2003).

3.3.1 Ideal case

When a point in space is in the field of view of both cameras in a stereo-pair, the rays through the optical center and the projection of the point on the image plane of each cameras intersect in the point itself.

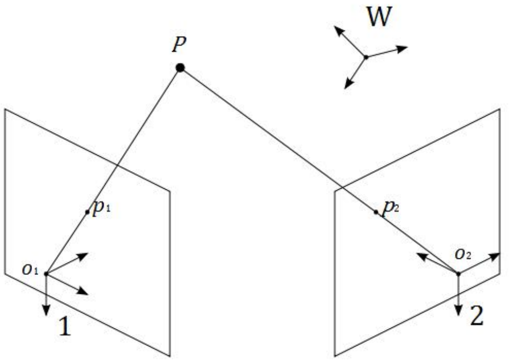


Figure 3.2: The triangulation geometry in an ideal case.

With reference to Figure 3.2, the rays through (o_1, p_1) and (o_2, p_2) intersect exactly in the point P. In mathematical language it can be express as:

$$ {}^{W}\mathbf{P} = \lambda_1({}^{W}\mathbf{p}_1 - {}^{W}\mathbf{o}_1) = \lambda_2({}^{W}\mathbf{p}_2 - {}^{W}\mathbf{o}_2) \tag{3.5} $$

This is a system of three equations in two variables (λ_1 and λ_2) but, in this ideal case, there are only two linearly independent equations. The position of the point $\mathbf{P}$ will be computed from one of the rays in Equation 3.5.

[2] Notice that the parameters S_x and S_y are referred to x and y-axis and not to $x\prime$ and $y\prime$-axis.

3.3.2 Real case

Because of the influence of uncertainty, in the real case the two rays do non-intersect. With reference to the Figure 3.3, the Equation 3.6 becomes:

$$ {}^W\mathbf{P} \cong \lambda_1({}^W\mathbf{p}_1 - {}^W\mathbf{o}_1) \cong \lambda_2({}^W\mathbf{p}_2 - {}^W\mathbf{o}_2) \tag{3.6} $$

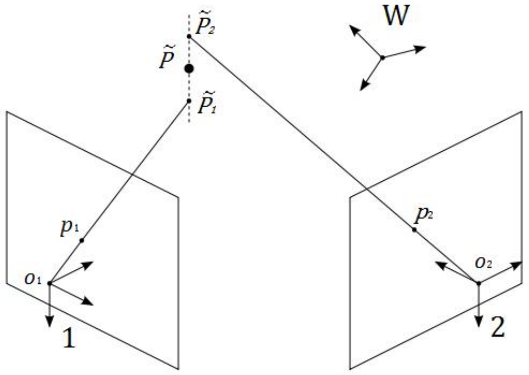


Figure 3.3: The triangulation geometry in an real case.

In this case the system has no solution in variables (λ_1 and λ_2). For this reason we have to use a cost function.

We can define two points $\tilde{\mathbf{P}}_1 \in r_1$ and $\tilde{\mathbf{P}}_2 \in r_2$ with the minimum distance. These two points defines a segment orthogonal to the two rays; middle point of this segment ($\tilde{\mathbf{P}}_m$) is selected as the measured 3D point.

A generic point $\mathbf{P}_1$ belonging to the ray r_1 is described from Equation 2.6. We can compute the position of two generic points $\mathbf{P}_1$ and $\mathbf{P}_2$ in the world reference frame as:

$$ {}^W\mathbf{P}_1 = \lambda_1 \cdot^W \mathbf{R}_1 \cdot^1 \mathbf{p}_1 + {}^W\mathbf{o}_1 \;, \; {}^W\mathbf{P}_2 = \lambda_2 \cdot^W \mathbf{R}_2 \cdot^2 \mathbf{p}_2 + {}^W\mathbf{o}_2 \tag{3.7} $$

In order to find points $\tilde{\mathbf{P}}_1$ and $\tilde{\mathbf{P}}_2$ with minimum distance, the following cost function g is defined and then minimized by imposing gradient $\left[\frac{\partial g}{\partial \lambda_1}, \frac{\partial g}{\partial \lambda_2}\right]$ equal to zero:

$$ \begin{aligned} g &= \left\| {}^W\mathbf{P}_1 -^W \mathbf{P}_2 \right\|^2 = \left({}^W\mathbf{P}_1 -^W \mathbf{P}_2\right)^T \left({}^W\mathbf{P}_1 -^W \mathbf{P}_2\right) \\ &= \left(\lambda_1 \cdot^W \mathbf{R}_1 \cdot^1 \mathbf{p}_1 + {}^W\mathbf{o}_1 - \lambda_2 \cdot^W \mathbf{R}_2 \cdot^2 \mathbf{p}_2 + {}^W\mathbf{o}_2\right)^T \cdot \\ &\quad \cdot \left(\lambda_1 \cdot^W \mathbf{R}_1 \cdot^1 \mathbf{p}_1 + {}^W\mathbf{o}_1 - \lambda_2 \cdot^W \mathbf{R}_2 \cdot^2 \mathbf{p}_2 + {}^W\mathbf{o}_2\right) \end{aligned} \tag{3.8} $$

Defining ${}^W\mathbf{v}_{12} =^W \mathbf{o}_1 -^W \mathbf{o}_2$ as the vector from ${}^W\mathbf{o}_2$ to ${}^W\mathbf{o}_1$ express in world reference frame, and using the combination of rotation matrices, the

Equation 3.8 becomes:

$$\begin{array}{rcl} g & = & \lambda_1^2 \cdot^1 \mathbf{p}_1^T \cdot^1 \mathbf{p}_1 - 2 \cdot \lambda_1 \cdot \lambda_2 \cdot^1 \mathbf{p}_1^T \cdot^1 \mathbf{R}_2^2 \mathbf{p}_2 - 2 \cdot \lambda_1 \cdot^1 \mathbf{p}_1^T \cdot^1 \mathbf{v}_{21} + \\ & & -2 \cdot \lambda_2 \cdot^2 \mathbf{p}_2^T \cdot^2 \mathbf{v}_{12} + {\lambda_2}^2 \cdot^2 \mathbf{p}_2^T \cdot^2 \mathbf{p}_2 +^W \mathbf{v}_{12}^T \cdot^W \mathbf{v}_{12} \end{array} \tag{3.9}$$

Taking partial derivatives and assigning a value of zero to the gradient yields the following equation system:

$$\left\{ \begin{array}{l} \frac{\partial g}{\partial \lambda_1} = \lambda_1 \left(2\,{}^1\mathbf{p}_1{}^T\,{}^1\mathbf{p}_1 \right) + \lambda_2 \left(-2\,{}^1\mathbf{p}_1{}^T\,{}^1\mathbf{R}_2\,{}^2\mathbf{p}_2 \right) + \left(-2\,{}^1\mathbf{p}_1{}^T\,{}^1\mathbf{v}_{21} \right) = 0 \\ \frac{\partial g}{\partial \lambda_2} = \lambda_1 \left(-2\,{}^1\mathbf{p}_1{}^T\,{}^1\mathbf{R}_2\,{}^2\mathbf{p}_2 \right) + \lambda_2 \left(2\,{}^2\mathbf{p}_2{}^T\,{}^2\mathbf{p}_2 \right) + \left(-2\,{}^2\mathbf{p}_2{}^T\,{}^2\mathbf{v}_{12} \right) = 0 \end{array} \right. \tag{3.10}$$

The solution of this system are the $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ values that define the minimum distance segment between the two rays. The symbolic solution of the system is:

$$\left\{ \begin{array}{l} \tilde{\lambda}_1 = \frac{\left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{R}_2 \cdot {}^2\mathbf{p}_2\right) \cdot \left({}^2\mathbf{p}_2{}^T \cdot {}^2\mathbf{v}_{12}\right) + \left({}^2\mathbf{p}_2{}^T \cdot {}^2\mathbf{p}_2\right) \cdot \left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{v}_{21}\right)}{\left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{p}_1\right) \cdot \left({}^2\mathbf{p}_2{}^T \cdot {}^2\mathbf{p}_2\right) - \left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{R}_2 \cdot {}^2\mathbf{p}_2\right)^2} \\ \tilde{\lambda}_2 = \frac{\left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{p}_1\right) \cdot \left({}^2\mathbf{p}_2{}^T \cdot {}^2\mathbf{v}_{12}\right) + \left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{R}_2 \cdot {}^2\mathbf{p}_2\right) \cdot \left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{v}_{21}\right)}{\left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{p}_1\right) \cdot \left({}^2\mathbf{p}_2{}^T \cdot {}^2\mathbf{p}_2\right) - \left({}^1\mathbf{p}_1{}^T \cdot {}^1\mathbf{R}_2 \cdot {}^2\mathbf{p}_2\right)^2} \end{array} \right. \tag{3.11}$$

Thus, the extreme points $\tilde{\mathbf{P}}_1$ and $\tilde{\mathbf{P}}_2$ of the minimum distance segment are:

$${}^W\tilde{\mathbf{P}}_1 = \tilde{\lambda}_1 \cdot {}^W\mathbf{R}_1 \cdot {}^1\mathbf{p}_1 + {}^W\mathbf{o}_1 \; , \; {}^W\tilde{\mathbf{P}}_2 = \tilde{\lambda}_2 \cdot {}^W\mathbf{R}_2 \cdot {}^2\mathbf{p}_2 + {}^W\mathbf{o}_2 \tag{3.12}$$

and the middle point associated with the points is:

$${}^W\tilde{\mathbf{P}}_m = \frac{{}^W\tilde{\mathbf{P}}_1 + {}^W\tilde{\mathbf{P}}_2}{2} \tag{3.13}$$

Part II

Proposed algorithms

Chapter 4

Algorithm for static reconstruction

In this chapter we discuss the description of the static reconstruction algorithm developed in this work. In Figure 4.1, you can see the flow chart of the algorithm. It is substantially based on three steps:

single camera stage in which, for each camera, the image is processed in order to extract the feature points (*200. feature detection*) and provide a description of each detected feature (*300. feature description*). See Section 4.1.

single stereo-pair stage in which the two sets of feature points, one for each camera in stereo-pair, are combined in order to reconstruct the 3D points cloud (*400. Stereo-reconstruction*). See Section 4.2.

complete system stage in which the points clouds provided from each stereo-pair are combined and fused in order to obtain a unique points cloud (*500. Points fusion*). See Section 4.3.

Externally to these stages, is the calibration stage (*600. System calibration*). The calibration algorithm used is based on the Tsai's approach (Horn, 2000) [1]. In this work we don't describe the calibration algorithm, for informations about the calibration parameters refer to Section 2.6.

What about the image acquisition stage (*100. Image acquisition*), a compiled file provide to the initialization, setting and synchronized image acquisition. In this step the most important things are:

- synchronism in image acquisition, because if the subject is moving, time-different acquisitions generate errors in triangulation of the feature points.

[1] This algorithm was developed for previous works of the Mechanical Measurement Research Group at University of Trento, and was adapted to the particular case.

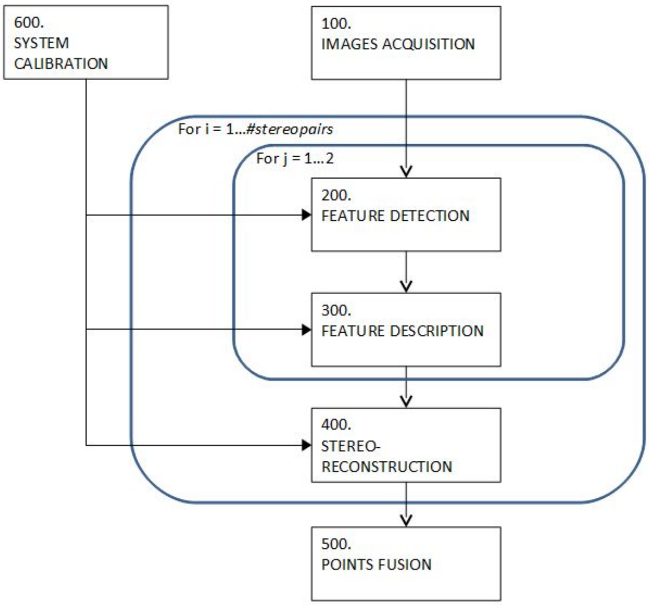


Figure 4.1: Functional description of the algorithm for static 3D reconstruction

- settings of camera parameters, because different settings of two cameras in a single stereo-pair can provide different feature descriptions and so errors in feature matching.

4.1 Feature extraction

The colored circular markers in the image shall be recognized in order to create a list of descriptor array. The image segmentation algorithm is based on edge detection using *Canny operator* (Canny, 1986). The feature extraction algorithm proposed in this work is composed of two steps: feature detection, which refers to the edge detection and filtering in order to select only the colored markers, and feature description, which refers to the generation of an array of descriptors used in the feature matching stage.

4.1.1 Feature detection

We start from an RGB image and we transform it into a gray-scale image in order to compute the edges using the Canny algorithm. In literature there are a lot of edge detectors (sobel, roberts, . . .), but we chose the canny one because of it is the best way to obtain closed contours. Thus we have a binary image that represents the contours of the markers but also contours of other objects in the field of view of the camera, and so we have to filter the image.

In the binary image we can consider each set of connected pixel as a distinct object. In this way we have a set of objects in the image and we have to determine if they are markers or not markers. For each object we can compute a set of geometric parameters:

Euler Number It is a scalar value that represents the total number of objects in the image minus the total number of holes in those objects. Since we consider one object singularly, the Euler number can be 1, if the contour is open; 0, if the object has only one hole; or less then 0, if the object has more then one hole.

Area It is the number of pixels in the object. In this case the object is the contour of a real object in the scene and so this value could be the perimeter of the marker.

Filled Area It is the area of the object plus the area of all the holes in the object. In this case this value could represents the area of the marker.

Major Axis Length It is a scalar value specifying the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region.

Minor Axis Length It is a scalar value specifying the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region.

Using these parameters we can select two conditions in order to establish if the object is the contour of a marker or something else.

The first condition is that the recognized object shall be closed contour with a single hole inside. This condition is mathematically described by;

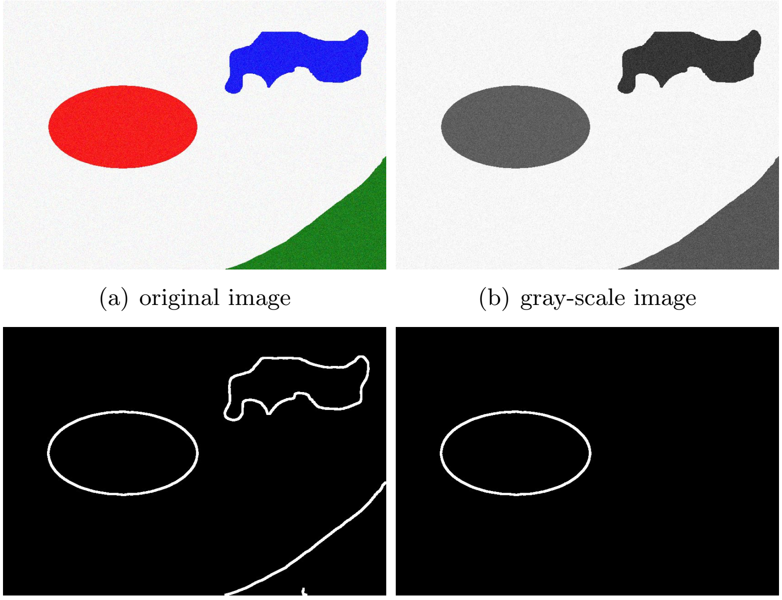
$$EulerNumber = 0$$

The second condition is based on the consideration that the projection of a circle, although deformed, is approximately an ellipse; so the ratio between perimeter and area must be, within a certain tolerance:

$$\frac{Area}{FilledArea} = \frac{\sqrt{2 \cdot (a^2 + b^2)}}{ab} \pm tolerance$$

where $a = MajorAxisLength$ and $b = MinorAxisLength$. The tolerance is related to two reasons: first, image digitalization and edge detection make the ellipse perimeter a polynomial approximation. Second, the exact formula of ellipse circumference is $C = 4aE(\epsilon)$, where the function $E(\cdot)$ is the complete elliptic integral of the second kind and ϵ is the eccentricity; the formula used in this work is an approximation with about 5% of tolerance (Barnard et al., 2001).

If both these conditions are verified, the object is considered as a marker, otherwise it will be deleted. An example of the image segmentation process is shown in Figure 4.2.


(a) original image

(b) gray-scale image

(c) Canny edges

(d) marker detected

Figure 4.2: The image segmentation process. The original image in RGB components is converted in gray scale and the edges are determined by using Canny algorithm. The object are filtered in order to delete the one without holes and the one that is not an ellipse.

4.1.2 Feature description

In order to determine the correspondence between feature points in two camera of the same stereo-pair, we have to generate a complete description of the feature point in term of position, geometry and chromatic characteristics. Each marker is described from:

centroid the position of the marker's centroid expressed in image reference frame.

shape an array of four parameters that describes the geometric characteristics of the marker; these parameters are: the area of the colored marker, its eccentricity, its orientation and the major axis length.

color a scalar value that indicates the color of the marker.

stripe a scalar value that indicates the stripe which the marker belongs.

The colors of the markers are assumed to be known a priori. We can compute the mean RGB components of pixels belonging to the marker. We associate the marker to a color class by using, in sequence, the *k-means* algorithm and *nearest neighbor* algorithm. In order to minimize the miss-classification, we used only four colors, equispaced in the (a*,b*) plane of the CIE-L*a*b* color space (ISO 12640-3:2007, 2007).In our case we choose the colors: red, green, blue and yellow.

The stripes clustering is performed on the same color markers and the algorithm is based on *principal component analysis.*

4.2 Stereo-reconstruction

Once we have detected all the features in two images of the same stereo-pair, we can proceed to 3D reconstruction. This stage can be divided in two subproblems: feature matching and triangulation. What about the triangulation algorithm, we refer to the middle point approach, as described in Section 3.3.

The general approach to the feature matching for marker systems is the *minimum epipolar distance* approach. Let us consider the feature point P_L in the left image of the stereo-pair, from the calibration parameters we can compute the equation of the epipolar line on the right image. The standard algorithm associates with the point P_L the point P_R in the right image with the same color of P_L and the minimum distance to the epipolar line. When applied to our system, this approach shows two different problems:

- if the angle between the epipolar line and the stripes direction is small, the association is highly sensible to the noise in the centroid determination.
- if the epipolar line intersect a lot of stripes, the probability to have a wrong association will be very high (about 50%).

For a more robust association we can use, instead of the distance to the epipolar line, a cost function ($g(\cdot)$) of the descriptors of the feature point. With reference to the Section 4.1.2, we define the cost function as:

$$g(P_L, P_R) = K_E \cdot D_E(P_L, P_R) + K_S \cdot \|shape(P_L) - shape(P_R)\| \qquad (4.1)$$

where $D_E(P_L, P_R)$ is the distance of point P_R to the epipolar line generated from P_L, $shape(P)$ is the four elemets array of shape descriptors, and K_E and K_S are two scalar coefficients that shall be tuned for the system.

Also using the cost function, the association is still affected by miss-match. In order to have an association as robust as possible, an innovative algorithm is proposed in this work. This algorithm is based on two different steps: the determination of a correct starting point and the expansion from this point with a threshold on the disparity gradient.

4.2.1 Finding the starting stripe

Let us consider two set of feature points Σ_L and Σ_R, respectively related to left and right images. To each point $P_L \in \Sigma_L$ we associate the point $P_R \in \Sigma_R$ that has:

- the same color of the point P_L,
- the minimum value of the function $g(\cdot)$.

For each pair (P_L, P_R) we have a corresponding pair of stripes (s_L, s_R), where $P_L \in s_L$ and $P_R \in s_R$. To each stripe s_L we can associate the stripe s_R that has the maximum number of matched points, and a score that represents the percentage of points of s_R associated with the ones in s_L. The starting stripe is the pair (s_L, s_R) with the maximum score.

4.2.2 Expansion from starting stripe

Once we have the starting stripes association, we have a set of matched pairs (P_L, P_R) and, to each pair is associated a disparity value, defined as the difference of coordinates of P_L and P_R, both expressed in own image reference frame. As explained in (Hartley and Zisserman, 2004), the disparity value is directly related to the depth of the point from the cameras. If we consider that the object is a continuous surface, we can impose the continuity of the disparity map.

In practice we proceed in an iterative loop that provides:

- determination of the N feature points in the left image nearest to the ones already matched.
- to each point, we associate the feature point in the right image having:
  - the same color of the point P_L,
  - the minimum value of the function $g(\cdot)$,
  - disparity in the range $[d_0 - \Delta_d, d_0 + \Delta_d]$.

where d_0 is the disparity of the nearest point to the analyzed one, and Δ_d is a parameter of the algorithm that is related to the maximum variation of depth acceptable. If there are no points in the range, the feature point is considered without matching.

The iterative loop ends when all the points in the left image are matched to one in the right image.

4.3 Points fusion

One of the most critical point in the multiple stereo approach, is originated from the fact that the points cloud is generated from several different stereo-pairs, and so we need a final stage in order to fuse the single points clouds in a unique one.

All the existing approaches are based on the partial overlapping of the points clouds, and provide a registration of the different clouds. The most common approaches are *Iterative Closest Points* (ICP) (Trucco et al., 1999; Eggert et al., 1997), and *Bundle Adjustment* (BA) (Triggs et al., 2000), often used in sequence.

In this work we propose a method for points fusion based on the uncertainty propagation by using the jacobian matrix, a compatibility analysis by using Mahalanobis distance and a Bayesian data fusion.

4.3.1 Uncertainty analysis

In the triangulation algorithm described above, the triangulated point $\tilde{\mathbf{P}}_m$ is computed as a function of:

- the projection of 3D point on the image plane of each camera (${}^{I}\mathbf{p}_i$, 2 components each)
- the intrinsic calibration parameters of each camera ($\mathbf{K}_i$, 4 components each)
- the extrinsic calibration parameters of each camera (${}^{W}\mathbf{R}_i$ and ${}^{W}\mathbf{o}_i$, 6 components each)

And so the coordinates of point $\tilde{\mathbf{P}}_m$ are a function of 24 parameters

$$\tilde{\mathbf{P}}_m = f\underbrace{\left({}^{I}\mathbf{p}_1, {}^{I}\mathbf{p}_2, \mathbf{K}_1, \mathbf{K}_2, {}^{W}\mathbf{R}_1, {}^{W}\mathbf{R}_2, {}^{W}\mathbf{o}_1, {}^{W}\mathbf{o}_2\right)}_{24\,parameters} \tag{4.2}$$

The rotation matrix is a 3×3 matrix and so it has 9 elements but, as we know from the rigid body transformation rules (Ma et al., 2003), the matrix has only three degrees of freedom. Each rotation matrix can be conveniently expressed by a set of three *Euler angles* (α_i, β_i and γ_i) defining rotation around three different axis (Goldstein et al., 2001).

In the sections below we discuss the evaluation of the uncertainty in each of these parameters, considering the calibration process and the feature detection process. After that we discuss the propagation of these uncertainty in order to evaluate the covariance matrix of 3D point position.

Calibration uncertainty

The parameters characterizing the camera model are estimated in a camera calibration stage, the procedure used is similar to that proposed by Tsai (Horn, 2000). The proposed procedure use a planar target which translates orthogonal to itself, generating a three-dimensional grid of calibration points. At a first step, the parameters are evaluated by a pseudo-inverse solution of a least-squares problem employing points on the calibration volume and image points. A second step provides an iterative optimization in order to minimize errors between acquired image points and the projections of the 3D calibration points on the image plane with the estimated parameters.

Before the calibration algorithm can be applied, optical radial distortion are estimated and adjusted by rectifying distorted images. Radial distortion coefficients are estimated by compensation of the curvature induced by radial distortion on the calibration grid (Devernay and Faugeras, 2001).

Camera parameters uncertainties are evaluated propagating by the uncertainties of the 3D calibration points and those of image points (Chen et al., 2008; Horn, 2000). A Monte Carlo simulation is used.

The reasons of the deviation between measured image points and the projection of 3D calibration points are various:

- simplification in camera model;
- digital camera resolution;
- dimensional accuracy of the calibration grid;
- geometrical and dimensional accuracy of grid translation.

In particular, if the motion of the grid to generate the calibration volume is not perfectly orthogonal to the optical axis of the camera, a bias is induced in the uncertainty distribution of the grid points, so that the uncertainty becomes non-symmetric in the image plane. Two more parameters are therefore introduced to characterize the horizontal and vertical deviations from orthogonality, α_R and β_C are the angles between translation direction and, respectively, grid rows and columns. In an ideal grid motion the two parameters shall be 90°.

Summarizing, the calibration routine consists of the following four steps:

1. Estimation and adjustment of the optical radial distortions.
2. First estimation of the parameters α_R and β_C, defining the imperfection of the calibration target motion. These values are achieved by minimizing iteratively the deviation between the measured image coordinates of calibration points and those reconstructed projecting volume points. The principal point is assumed to lie in the middle

point of the image. With this assumption, once the systematic deviation from orthogonality have been compensated, extrinsic and intrinsic parameters can be derived from a pseudo inverse solution (Horn, 2000).

3. Final iterative optimization of all camera parameters (including principal point) is performed, iteratively minimizing the deviation between measured image points and those reconstructed projecting the 3D calibration points. This step supplies the final estimation of intrinsic and extrinsic parameters. Standard deviation of the residuals after the projection is combined with the resolution uncertainty, and is used to evaluate the uncertainty associated with the image points used in the next step.

4. Lastly, through a Monte Carlo simulation, the uncertainties of the image points (as evaluated in the previous step) and the 3D calibration points are propagated, in order to evaluate the uncertainty of the calibration parameters.

Steps 2 and 3 usually require usually less than 10 iterations, while the Mote Carlo simulation step usually require about 10^5 iterations.

Feature uncertainty

The point $\mathbf{P}$ in 3D is defined as the centroid of a circular marker; for this reason, determination of its projection $\mathbf{p}_i$ in the image plane of the i-th camera is always affected by uncertainty. First of all, digitalization and successive binarization of the image deforms the circular shape into a polygonal shape, and the centroid of these two shapes is not the same. Second, the marker, which was originally a circle, is deformed in order to adhere to the target surface; as first approximation, the deformed marker can be express as an ellipse. Third, due to perspective effects, an ellipse which is not perpendicular to the optical axis of the camera is projected on the CCD as an ovoid.

A simplified model of the perspective geometry identifies each marker projection as an ellipse; this ellipse can be fitted by the covariance matrix of the distribution of the pixels recognized as markers. The projected marker can then be compared with the corresponding covariance ellipse and the 2D distance of the two boundaries (Δ_b) can be computed as a function of angle α. This distribution $\Delta_b(\alpha) = b_{real}(\alpha) - f_{fit}(\alpha)$ is a map $\mathbb{R} \longrightarrow \mathbb{R}^2$. Figure 4.3 shows an example of a badly segmented elliptical marker and the fitted one, and the related distribution Δ_b.

Two parameters Δ_C and C_b, which express the “difference” between the projected ovoid and the estimated ellipse can be computed; $\Delta_C = [x_{\Delta_C}, y_{\Delta_C}]^T \in \mathbb{R}^2$ is the displacement between the centroid of the segmented

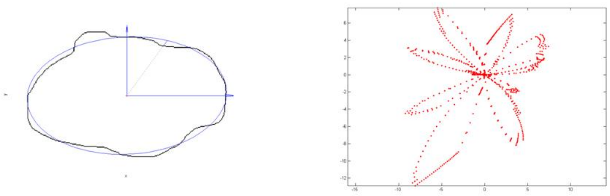


Figure 4.3: Example of badly segmented elliptical marker. On the left, the edge of the marker and the fitted ellipse; on the right, the distribution Δ_b.

marker and the fitted ellipse, and $C_b \in \mathbb{R}^{2\times 2}$ is the covariance matrix of distribution Δ_b.

The uncertainty of the centroid of the segmented marker is represented by a covariance matrix which is a function of these two parameters:

$$\mathbf{U}_{meas} = f\left(\Delta_C, C_b\right) = a \cdot \begin{bmatrix} x_{\Delta_C} & 0 \\ 0 & y_{\Delta_C} \end{bmatrix} + b \cdot C_b \tag{4.3}$$

The larger the difference between the projected ovoid and the estimated ellipse, the larger the uncertainty associated with the computed centroid. In this function, parameters a and b are evaluated by a calibration procedure, which uses a grid of circular photolithographic markers. This grid is moved in a set of known positions and orientations, and the computed centroid of the segmented marker is compared with the projected reference on the CCD. In order to have a large set of views, two kinds of grids are used: the first is a planar surface and the second the lateral surface of a cylinder.

4.3.2 Uncertainty propagation

The uncertainty evaluation for triangulated point $\tilde{\mathbf{P}}_m$ of each stereo pair becomes an uncertainty propagation problem, which employs the functional model between input and output quantities, as express in Equation 4.2.

Several uncertainty propagation methods are known. Each of them is based on a theory (i.e. probability, possibility or evidence theory), which can express uncertainty by a corresponding suitable means (i.e. probability density functions, fuzzy variables or random-fuzzy variables). According to the GUM (BIPM et al., 1993), in this work, the uncertainty is analyzed according to the probability theory and is expressed by probability density functions (*PDFs*).

In order to calculate the propagated uncertainty of triangulated position $\tilde{\mathbf{P}}_m$, taking into account the contributions of all uncertainty sources, the method based on the formula expressed in the GUM is used. This method is selected instead, for example, of the Monte Carlo propagation approach, in order to increase computing speed and to allow real-time computing implementation. The propagation formula uses the sensitivity coefficients obtained from linearization of the mathematical model; this method is based on thee hypothesis that a probability distribution, assumed or experimentally determined, can be associated with every uncertainty source considered, and that a corresponding standar uncertainty can be obtained from the probability distribution.

The GUM proposes a formula for the calculation of the uncertainty to be associated with output quantities $\tilde{\mathbf{P}}_m$, obtainable as an indirect measurement of all input quantities:

$$\mathbf{U}_{out} = \mathbf{c} \cdot \mathbf{U}_{in} \cdot \mathbf{c}^T \tag{4.4}$$

where $\mathbf{U}_{in} \in \mathbb{R}^{24 \times 24}$ is the covariance matrix associated with the input quantities, which are 24 in this application; $\mathbf{U}_{out} \in \mathbb{R}^{3 \times 3}$ is the covariance matrix associated with the output quantities, which are the three components of $\tilde{\mathbf{P}}_m$ in this application; and $\mathbf{c} \in \mathbb{R}^{3 \times 24}$ is the matrix of the sensitivity coefficients achievable from partial derivatives of $f(\cdot)$ with respect to input variables:

$$\mathbf{c}_{i,j} = \frac{\partial f_i}{\partial input_j} \tag{4.5}$$

In this application, the following assumptions are made:

1. The two components of the projected point ${}^{I}\mathbf{p}_i$ of each camera are assumed to be cross-correlated among themselves and not correlated with any other input quantity.

2. The intrinsic calibration parameters of each camera are assumed to be cross-correlated among themselves and not correlated with the corresponding parameters of the other cameras or any other input quantity.

3. The extrinsic calibration parameters of each camera are assumed to be cross-correlated among themselves and not correlated with the corresponding parameters of the other cameras or any other input quantity.

These assumptions allows us to build the 24 × 24 covariance matrix of scalar input quantities, putting six reduced dimension covariance matrices along the diagonal $\mathbf{U}_{in}$ and assigning zero values to all other elements. The matrix becomes:

$$\mathbf{U}_{in} = \begin{bmatrix} \mathbf{U}_{meas,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{U}_{meas,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{U}_{int,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{U}_{int,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{U}_{ext,1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{U}_{ext,2} \end{bmatrix} \tag{4.6}$$

where $\mathbf{U}_{meas,i} \in \mathbb{R}^{2\times 2}$ is the covariance matrix associated with measurement of the projected point ${}^{I}\mathbf{p}_i$ in the i-th camera; $\mathbf{U}_{int,i} \in \mathbb{R}^{4\times 4}$ is the covariance matrix associated with the intrinsic calibration parameters of the i-th camera; and $\mathbf{U}_{ext,i} \in \mathbb{R}^{6\times 6}$ is the covariance matrix associated with the extrinsic calibration parameters of the i-th camera.

The propagation model between input and output quantities described in this work is not very simple, but have the advantage of being explicit. Thus, it is possible to compute explicitly the sensitive coefficients as symbolic expressions, and it is not necessary to evaluate them numerically, as often happens with complex applications.

4.4 Compatibility analysis

As we have seen in Section 4.3.1, in non-ideal conditions, the stereo systems at different positions provide different measurements of the same feature point; in this work the feature points are centroids of colored spots. Each measurement comes with its uncertainty, and a fusion process can combine them in a single best-estimated one with the associated fused uncertainty. Before points measured from different stereo systems can be fused, it is necessary to state whether they are associated with the same feature or, statistically speaking, whether they belong to the same distribution. A compatibility analysis of the measured points is therefore performed.

A compatibility test on two points $\mathbf{P}_1$ and $\mathbf{P}_2$, with covariances $\mathbf{C}_1$ and $\mathbf{C}_2$, is based on the consideration that the difference $\mathbf{P}_1 - \mathbf{P}_2$ is distributed with zero mean and covariance $\mathbf{C} = \mathbf{C}_1 + \mathbf{C}_2$.

We can define the *Mahalanobis Distance* (MD) (Duda et al., 2000) as a statistical distance described by the equation:

$$MD^2 = (\mathbf{P}_1 - \mathbf{P}_2)^T (\mathbf{C}_1 + \mathbf{C}_2)^{-1} (\mathbf{P}_1 - \mathbf{P}_2) \tag{4.7}$$

Intuitively, the Mahalanobis distance is the distance between two points, divided by the width of the covariance ellipsoid in the direction of the points connection. If we consider two points with spherical uncertainty, the MD is exactly the same as the euclidean distance; but if we consider two points with generic covariance matrices, as more are the ellipsoids aligned with the line joining the two points, as less is the MD. In Figure 4.4 are shown two points with the same euclidean distance and uncertainty but MD completely different, because of the orientation of the covariance ellipsoids.

On the Gaussian assumption, the MD has a χ^2 distribution with a degree of freedom ν equal to the dimension of vectors $\mathbf{P}_i$. Once a confidence level $\alpha\prime$ has been chosen, it is stated that the two points are compatible if:

$$MD^2 \leq \chi^2(\nu, \alpha\prime) \tag{4.8}$$

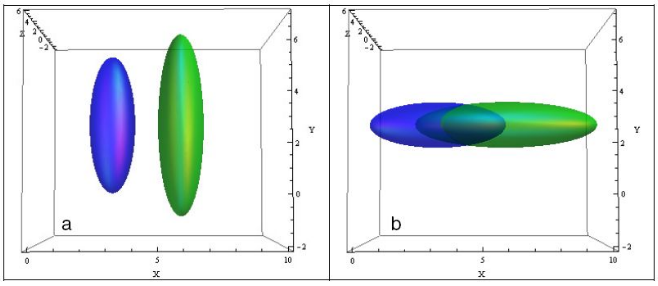


Figure 4.4: An example of different Mahalanobis Distance between the same couple of points, with covariance ellipsoids of equal axis length but different orientations. In the first case (a) the MD is equal to 4.5, in the second one (b) is 0.36.

Let $\mathbf{P}_{i,j}$ be the i-th 3D point measured by the j-th stereo-pair with covariance $\mathbf{C}_{i,j}$. The point fusion algorithm is made up of the following steps: from measured points sets Σ_m and Σ_n of stereo-pairs m and n respectively, each point $\mathbf{P}_{i,m} \in \Sigma_m$ is associated with the point $\mathbf{P}_{j,n} \in \Sigma_n$ having the minimum MD; if the compatibility test is passed, the association is accepted and the associated couple of points is fused, yelding the best estimate:

$$\mathbf{P}^*_{k,mn} = \mathbf{C}_{j,n}(\mathbf{C}_{i,m} + \mathbf{C}_{j,n})^{-1}\mathbf{P}_{i,m} + \mathbf{C}_{i,m}(\mathbf{C}_{i,m} + \mathbf{C}_{j,n})^{-1}\mathbf{P}_{j,n} \tag{4.9}$$

and its covariance matrix:

$$\mathbf{C}^*_{k,mn} = \mathbf{C}_{j,n}(\mathbf{C}_{i,m} + \mathbf{C}_{j,n})^{-1}\mathbf{C}_{i,m} \tag{4.10}$$

otherwise $\mathbf{P}_{i,m}$ and its covariance matrix $\mathbf{C}_{i,m}$ are kept as best estimate of the feature; the process between all these best estimates is iterated (including points not associated between the two sets), and a new set Σ_p is obtained.

Ambiguous cases may occur, when a point of set Σ_n is compatible with two or more points of set Σ_m. In this case, the point of Σ_n is eliminated. For this reason, threshold $\alpha\prime$ must be tuned in order both to keep cases of ambiguity low and not to lose useful information.

Notice that the best estimated fused point is computed by using the Bayes theorem, therefore the covariance ellipsoid of the fused point is smaller than the smallest of those starting. An example of the covariance ellipsoids of a pair of points and the fused one is shown in Figure 4.5

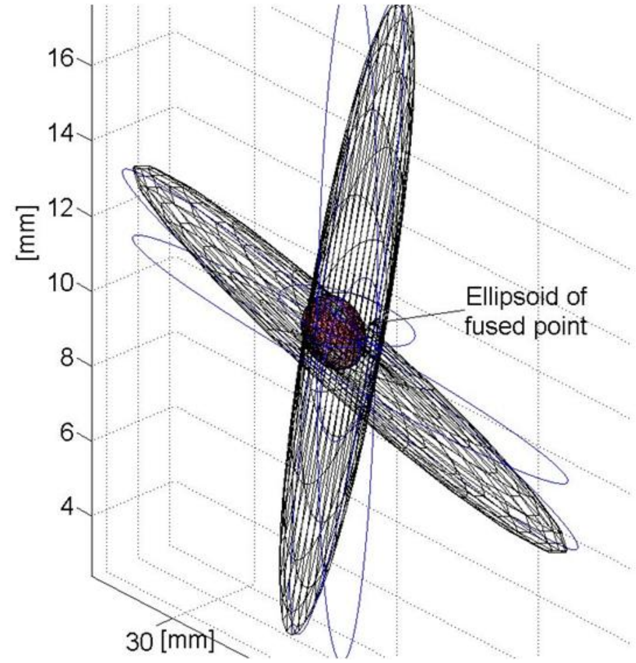


Figure 4.5: An example of covariance ellipsoids of two corresponding points acquired by two stereo-pairs, and the ellipsoid of the fused point.

Chapter 5

Algorithm for *Motion Capture*

With the algorithm described in Chapter 4 we have a system able to capture a certain number of images from different cameras and, from this images, to reconstruct the 3D position of a certain number of feature points in the field of view. In this case we acquire a single image for each camera.

If we acquire not only a single image, but a sequence of images, we can use this time-variant information in order to characterize better the viewed objects. In particular we are interested in the segmentation of the different objects in the field of view and in modeling the joints between these objects.

The final aim of the here presented system is to obtain a complete description of the articulated body in 3D and its motion properties automatically from a set of images given a known pattern. In particular, our interest is on the analysis of human motion (i.e. arms or legs).

As graphically explained in Figure 5.1, the proposed algorithm is a sequence of four steps. What about image acquisition and 3D reconstruction steps we refer to Chapter 4, here we present only a brief reminder in Section 5.1. The next Section 5.2 shows how the pairwise 3D matching is done exploiting the particular structure of the pattern. Section 5.3 presents the segmentation algorithm based on the motion of the object shape, we describe here three different algorithms: GPCA, RANSAC ans LSA. Finally Section 5.4 describes the articulated joint position computation.

5.1 Image acquisition and 3D reconstruction

The image acquisition system, by using the 3D static reconstruction algorithm described in Chapter 4, is able to reconstruct the position of a set of points belonging to a generic surface located in a given working space.

An example of a subset of acquired images is shown in Figure 5.2, together with the pattern used for the acquisitions (Figure 5.3).

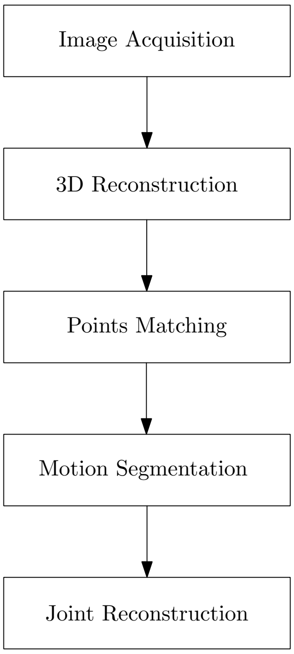
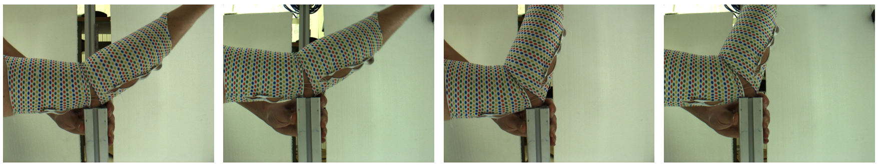


Figure 5.1: The flow chart of our approach to the motion capture problem.



(a) frame 1 left (b) frame 1 right (c) frame 8 left (d) frame 8 right

Figure 5.2: An example of a set of images acquired from the stereo-pairs.

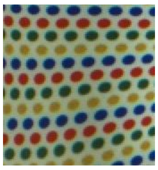


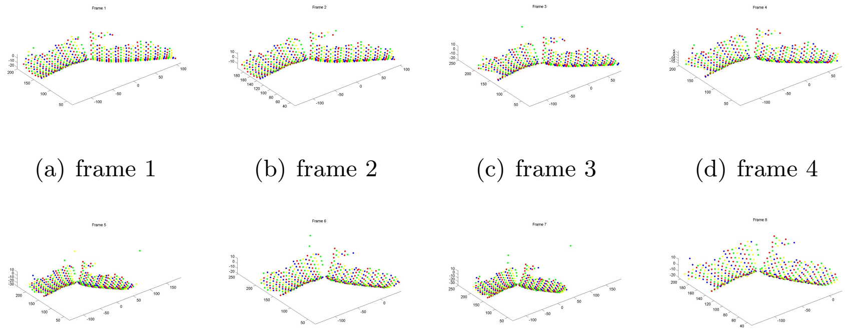
Figure 5.3: The pattern used for the 3d reconstruction.

The output of the 3D reconstruction stage for a single frame is a matrix M of size $n \times 5$, where n is the number of reconstructed points. The first three columns of M are the metric coordinates x, y and z of the point. The fourth column is a scalar that indicate the color of the marker, while the last one is a scalar that represents the uncertainty of the reconstructed point.

The matrix M at a single frame can be written as:

$$M = \begin{bmatrix} x_1 & y_1 & z_1 & col_1 & unc_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & col_n & unc_n \end{bmatrix}$$

Figure 5.4 shows an example of the 3D reconstruction for a eight frames long image sequence where the captured non-rigid motion is an arm bending as presented in Figure 5.4 from a single stereo-pair view.



(a) frame 1 (b) frame 2 (c) frame 3 (d) frame 4

(e) frame 5 (f) frame 6 (g) frame 7 (h) frame 8

Figure 5.4: An example of a set of frames used for the motion segmentation and joint reconstruction.

5.2 Trajectory matrix

The previous 3D reconstruction stage provides a set of unordered 3D coordinates at each image frame. The next task is to assign at each 3D point in a given frame the corresponding 3D point in the following frame. This is a fundamental step in order to infer the global properties of the non-rigid image shape (i.e. its motion). This 3D matching stage aims to form a measurement matrix in which each column of the matrix represent a 3D trajectory of the point. This matrix is of size $3F \times P$ and it contains the position of the P features tracked throughout F frames.

We have as input two frames, with respectively n and m points, uniquely described by two matrices

$$M_1 = \begin{bmatrix} x_1 & y_1 & z_1 & col_1 & unc_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & col_n & unc_n \end{bmatrix}$$

$$M_2 = \begin{bmatrix} x_1 & y_1 & z_1 & col_1 & unc_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m & y_m & z_m & col_m & unc_m \end{bmatrix}$$

Each row of the matrices represents a point, along the columns there are the coordinates of the point (x, y e z), the color (col) and the uncertainty (unc).

The output of the frame-by-frame points matching algorithm is a vector

$$P^2 = \begin{bmatrix} P_1^2 \\ \vdots \\ P_n^2 \end{bmatrix}$$

with the same number of rows of the matrix M_1, each row contains the index of the point in the second frame matched with the one in the first frame. If a point of the first frame has no given assignment in the second frame, the value in P^2 will be NaN.

5.2.1 Matching using Nearest Neighbor

The simplest algorithm we can use is a revisited version of the classical nearest neighbor (NN) approach to account for the different color assigned to each 3D point in both frames. The algorithm is composed by the steps below:

1. compute the metric distance matrix between each pair of points of the same color in the two frames

   $$D = \begin{bmatrix} d_1^1 & \dots & d_1^m \\ \vdots & \ddots & \vdots \\ d_n^1 & \dots & d_n^m \end{bmatrix}$$

   the pairs of point of different colors the correspondent value is NaN

2. compute the minimum distance for each point of frame 1, NaN values are ignored, we obtain a $n \times 2$ matrix in which the rows are the points and the columns are the minimum distance and the index of the nearest point

   $$D_{min} = \begin{bmatrix} d_1^{min} & ind_1^{min} \\ \vdots & \vdots \\ d_n^{min} & ind_n^{min} \end{bmatrix}$$

3. if the minimum distance between two points is lower than a threshold and the association is unique, the association is considered valid, otherwise it will be deleted.

The threshold is automatically computed from the mean distance, the mean of the first column of the matrix D_{min}, multiplied by a coefficient.

This algorithm gives reasonable results under the hypothesis that the movement of the feature between two successive frames is small with respect to the distance between the features in a single frame. This means that the motion of the bodies shall be slow with respect to the frame rate and the features spatial density.

5.2.2 Matching using NN and Procrustes analysis

We also propose a novel algorithm for the frame-by-frame point matching that combines NN and *Procrustes Analysis* (PA) theory. The algorithm is composed by the three distinctive stages.

1. **Stripe sorting.** As a result of the previous reconstruction stage, each 3D point in M has assigned a given color. Thus, given the pattern repetitive structure, it is possible to associated the points with the same color to a set of stripes. Each stripe is sorted along the principal directions of the 3D shape at the given frame.

2. **Stripe matching.** In this stage we match each stripe in the first frame to a stripe in the second frame. This association is made using a NN approach on the centroid of each stripe using again the color as a discriminative feature.

3. **Match 3D points in each stripe.** For two matched stripes, we select first the stripe containing less 3D points. Then we sequentially assign these points to the 3D points of the other stripe and we register the two sets using PA (Kanatani, 1996); see Figure 5.5 for a graphical explanation. We selected the assignment which results in the minimum 3D error.

Stage 1 and 2 are based on the observation that a NN over the centroid of the stripe is more robust to deformation and more computationally efficient then performing a NN on each 3D point. Especially for the second step, if the deforming body can be considered locally rigid on the stripe, the rigid registration by PA give low 3D residual if the matching is correct.

The proposed algorithm is very robust for short movements with respect to the stripe-to-stripe distance. If the displacement between the centroids of the same stripe in two following frames is comparable to the stripe-by-stripe distance, this method is no longer robust. In this case we can use a similar algorithm, that we can call *Local Procrustes Analysis* (LPA), in which we consider not only the associated stripe for the PA, but also the n-nearest. This method is more robust than the previous one in the case of large displacements. Unfortunately this modification introduce more sensibility to deformations.

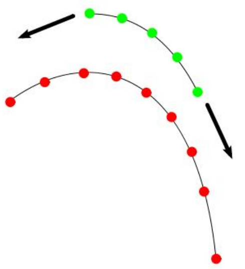


Figure 5.5: A graphical explanation of the Procrustes Analysis applied to the point matching. Dots represent the 3D points lying over a stripe. In red a set of points for a candidate line to be matched at frame t. In green a selected line at frame $t + 1$which contains less points. The green points *slides* over the line with red dots and for each association a PA is computed. The matched points are the one which give the minimum 3D error after registration.

Once we have a frame-by-frame matching array for each pair of successive frames, we can build a trajectory matrix taking into account only the features tracked in all the frames. The trajectories of the full tracked points in the example case are shown in Figure 5.6.

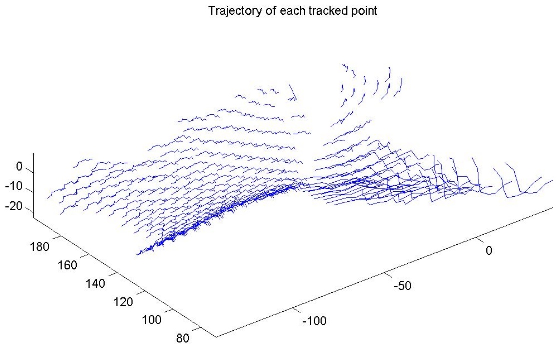


Figure 5.6: An example of trajectory of points in a sequence of eight frames.

In this algorithm we consider only the points tracked in all the frames, the markers tracked only in a few frames could be considered with a dedicated missing data algorithm.

The result trajectory matrix is:

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1P} \\ \vdots & \ddots & \vdots \\ w_{F1} & \cdots & w_{FP} \end{bmatrix} \tag{5.1}$$

where $w_{ij} = [x_{ij}, y_{ij}, z_{ij}]^T$ is the 3D position of the j-th point at the i-th frame.

5.3 Motion Segmentation

Once the assignment problem is solved, the matrix W stores the correct temporal information of the 3D trajectories of the non-rigid body. In order to perform an analysis on the motion of body articulations we need to segment the clouds of points which are assigned to relevant rigid motion. In the experimental case here presented this means to assign each 3D trajectory point in W to two clusters of points lying on the forearm or on the arm. Notably, such problem span a vast literature in the Computer Vision where it is generally termed as the motion segmentation problem. In the following we are evaluating the results obtained by a subset of these methods applied and modified for the 3D segmentation problem. In particular, we assess the performances of three algorithms: Generalized Principal Component Analysis (GPCA) (Vidal et al., 2005), Subspace RANSAC (Fischler and Bolles, 1987; Tron and Vidal, 2007) and Local Subspace Affinity (LSA) (Yan and Pollefeys, 2006b).

These algorithms obtains reasonable results for 2D motion segmentation tasks with the LSA approach obtaining the best results in general purpose databases (Tron and Vidal, 2007). In the following, we evaluate their quality in the case of bodies which have a certain degree of non-rigidity and soft tissue artifacts. In general, we expect decreasing performance in two different regions:

border zone where the marker's movement could be affected by the muscle tension. Regions are marked with blue in Figure 5.7.

joint zone where the two bodies are not well separated because of the geometrical conformation of the natural joint (elbow). The region is marked with red in Figure 5.7.

5.3.1 GPCA algorithm

The GPCA (Vidal et al., 2005) method was introduced with the purpose of segmenting data lying on multiple subspaces. This is also the case for 3D

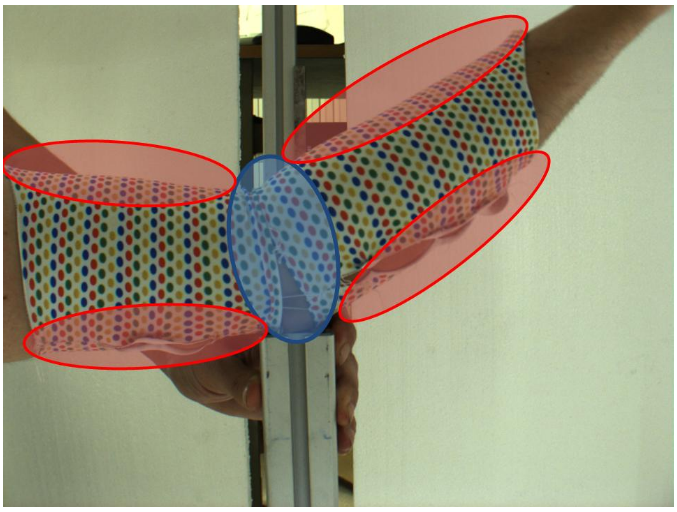


Figure 5.7: The expected critical zone in the segmentation stage: the border zone (blue) and the joint zone (red).

shapes moving and articulating since their trajectories lies on different subspaces. The method is based on the consideration that trajectories of rigid motion generate subspaces at most of dimension 4. The GPCA algorithm first project the trajectories onto a five dimensional space using PowerFactorization, then GPCA is used to obtain the segmentation in the new lower dimension space.

This method was originally developed for rigid bodies in a 2D data collection, for instance if we have only one camera; in this work we tried to apply the algorithm to the case of non-rigid bodies segmentation in a 3D data collection. As explained by the authors, the main drawback of GPCA is that the performances degraded when the number of objects increases, moreover, the method does not assume the presence of outliers.

Figure 5.8 shows the segmentation results using GPCA over the arm movement 3D data. The segmentation error is about 25% in this test showing several outliers far from the joint and thus from the expected regions. This unexpected result may be a consequence of the non-rigid motion of the body parts.

5.3.2 RANSAC algorithm

The method is a popular and effective tool for robust statistical analysis (Fischler and Bolles, 1987). It is based on the selection of the best model which fit the inlier data. In order to estimate the putative models, candidates set of points are chosen randomly and then the residual given the fitted model is stored. After several random sampling, the model which fits best the inliers is chosen.

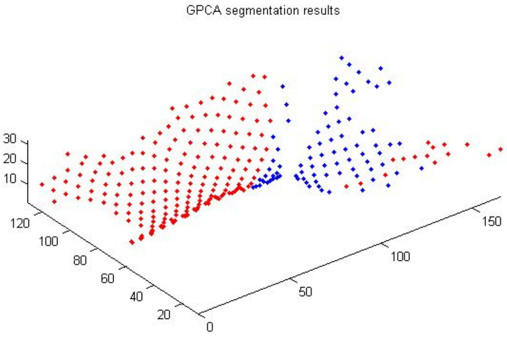


Figure 5.8: An example of motion segmentation using the Generalized Principal Component Analysis (GPCA).

In our case, this algorithm is the worst performing of the three obtaining a segmentation error of approximately 50% of the given points for this dataset, as it is possible to see in Figure 5.9. In this figure we can observe that most of the errors are at the border zone, where we expect errors because the movement of the markers could be affected by the muscle tension.

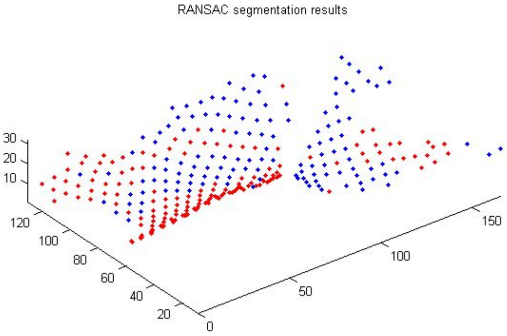


Figure 5.9: An example of motion segmentation using the RANSAC.

This method gives errors in both the expected regions, joint and border, but in general we have several errors also in similar critical region as for the GPCA algorithm.

5.3.3 LSA algorithm

The LSA approach (Yan and Pollefeys, 2006b) uses spectral analysis in order to define the data clusters which refer to different motion subspaces. It is based on local subspace fitting in the surrounding of each trajectory followed by spectral clustering.

The Local Subspace Affinity algorithm exploits the fact that different trajectories lie in a mixture of linear manifolds of different dimensions in order to deal with different type of motion: rigid, independent, articulated and non-rigid. The general idea is to estimate the local linear manifold for each trajectory, and compute an affinity matrix based on some measure of the distance between each pair of manifolds. Once the affinity matrix is built, any clustering algorithm could be applied in order to group the trajectories and hence segment the motion.

The LSA algorithm can be divided into five main steps: rank estimation, data transformation, subspace estimation, affinity matrix and clustering. The algorithm flow is summarized in Figure 5.13.

Rank estimation

From the tracked features, a full trajectory matrix $W_{3F \times P}$, where F is the number of frames and P the number of tracked point features, is built. The first step of the algorithm consists in estimating the rank of W by using Model Selection (MS) techniques (Kanatani, 2001). The rank (r) is estimated as:

$$r = argmin_r \frac{\lambda_{r+1}^2}{\sum_{k=1}^{r} \lambda_k^2} + kr \tag{5.2}$$

being λ_i the i-th singular value of W, and k a parameter that should depend on the noise. The higher the noise level is, the larger k should be used.

Data transformation

Given the trajectory matrix W and its estimated rank r it is possible to perform a data transformation. The idea is to consider each of its P columns as a vector in $\mathbb{R}^{3F}$ and to project them onto the $\mathbb{R}^r$ unit sphere. This data transformation provides a dimensionality reduction, a normalization of the data and a preparation for next step: the local subspace estimation. Figure 5.10 shows an example of trajectories that belong to two different subspaces of dimension 2 projected into an $\mathbb{R}^3$ unit sphere. The white dots are trajectories which belong to one motion while black dots are trajectories which belong to another motion. Due to noise not all the dots of the same color lie exactly on the same subspace (one of the circles in the image).

In order to perform this transformation, SVD is applied to the matrix W which is decomposed into $U_{3F \times 3F}$, $D_{3F \times P}$ and $V_{P \times P}$ as:

$$W_{3F \times P} = U_{3F \times 3F} \cdot D_{3F \times P} \cdot V_{P \times P}^T \tag{5.3}$$

Dimensionality reduction is achieved by considering only the first r rows of V. Hence, each column of the matrix $V_{r \times P}$ is normalized to project them onto the unit sphere.

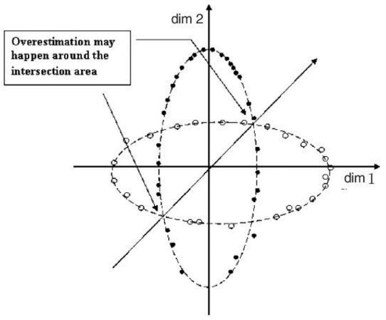


Figure 5.10: Example of trajectories that belong to two different subspaces of dimension 2 projected into an $\mathbb{R}^3$ unit sphere. The two circles are the 2 subspaces. The white dots are trajectories which belong to one motion while black dots are trajectories which belong to another motion.

Subspace estimation

As can be seen in Figure 5.10, in the transformed space most trajectories and their closest neighbors lie on the same subspace. Hence, the underlying subspace of a trajectory α can be estimated by local samples from itself and its n nearest neighbors, being $n + 1 \geq d$, where d is the highest dimension of the linear subspace generated by the cluster.

If the type of motion is known, n can be tuned knowing that; for example, for rigid motion $d \leq 4$ while for articulated motion $d \leq 7$. If the motion is not known the highest dimension should be considered, for example $d = 7$ and those $n \geq 6$. As a measure of the distance between two trajectories α_1 and α_2, the angle $arccos(\alpha_1^T \alpha_2) \in [0, \pi]$ is used. It should be noted that the computation of the nearest neighbors would not be possible without the data transformation step, because it is not the spatial vicinity that is considered here but it is the vicinity in terms of subspace distance.

Once the n nearest neighbors are identified, the bases of the subspace of the trajectory α can be computed by SVD. From W, the sub-matrix W_α containing only the trajectory α and its n nearest neighbors is extracted. The rank of W_α is estimated again using the model selection. Knowing the rank of W_α and using SVD the bases of the subspace $S(\alpha)$ can be computed. The subspace estimation is performed for each trajectory, so that at the end of this step the bases of the estimated subspace of all the trajectories are available.

When estimating the subspace of trajectories that are close to the intersection between two subspaces the sampled nearest neighbors could belong either to "correct" subspace or to the others. The proposers of the method call this phenomena *overestimation.* Even thought overestimation leads to an error in the subspace estimation, they say that this has a minor effect on the overall segmentation. The reasons are that trajectories that are close to the intersection are usually small in amount compared to the total, besides, in which cluster the trajectory will be classified relies on which of the underlying subspaces is dominant for the overestimated subspace.

Affinity matrix

The affinity matrix $A_{P\times P}$ measures the similarity of each pair of the previously estimated local subspaces. The affinity of two trajectories α_1 and α_2 is defined as the similarity of their estimated local subspaces $S(\alpha_1)$ and $S(\alpha_2)$:

$$A(\alpha_1, \alpha_2) = exp\left(-\sum_{i=1}^{M} sin(\theta_i)^2\right) \tag{5.4}$$

where θ_i is the i-th principal angle (Golub and Van Loan, 1996) between the subspaces $S(\alpha_1)$ and $S(\alpha_2)$, and M is the minimum dimension between $S(\alpha_1)$ and $S(\alpha_2)$. From this definition it can be deduced that A is a symmetric matrix and its entries take positive values, with a maximum of 1. The closer to 1 is an entry $A(\alpha_1, \alpha_2)$, the more similar the local subspaces $S(\alpha_1)$ and $S(\alpha_2)$ are.

Figure 5.11 shows the affinity matrix of trajectories which belong to three different clusters; the diagonal is white as all the values are equal to one (being any subspace identical to itself) while black spots represent subspace pairs with low similarity.

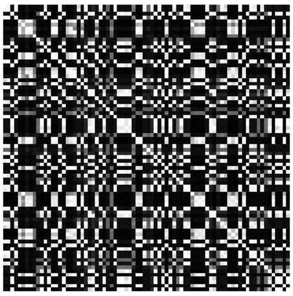


Figure 5.11: Example of affinity matrix of the trajectories of three clusters.

Clustering

Now that the affinity matrix A is computed, the idea is to group together subspaces with high degree of similarity among them. Any clustering algorithm can be applied, Yan and Pollefeys suggested to use the recursive 2-way spectral clustering technique (Shi and Malik, 1997).

The recursive 2-way spectral clustering works as follows:

- given N clusters in the scene and the affinity matrix A, segment the data into two clusters C_1 and C_2 by spectral clustering.

- while $NumberOfClusters(C_1, \ldots, C_m) < N$ (at the first step $m = 2$), compute the affinity matrix for each cluster C_i where $i = 1, \ldots, m$. Divide each C_i into two clusters, C_i^1 and C_i^2. Evaluate the Cheeger constant (Ng et al., 2001) of each pair C_i^1 and C_i^2. The Cheeger constant gives a clue of how "difficult" is to split the cluster, hence decide the subdivision of the cluster $C_{\hat{i}}$ that gives the minimum Cheeger constant. Replace $C_{\hat{i}}$ with $C_{\hat{i}}^1$ and $C_{\hat{i}}^2$.

As the spectral clustering algorithm is concerned, the normalized cut criterion (Shi and Malik, 1997) is used. This approach is related to the graph theoretic formulation of grouping. Focusing on motion segmentation, each of the previously computed local subspaces can be interpreted as the nodes V of a graph G and the affinity A defines the weights on the edges E of the graph. That is, the weight between the node V_i and V_j is defined by $A(i, j)$. The graph $G = (V, E)$ can be partitioned into two disjoint sets C_1 and C_2, where $C_1 \cup C2 = V$ and $C_1 \cap C_2 = \oslash$. The degree of dissimilarity between these two sets can be computed as total weight of the edges that have been removed. In graph theoretic language it is called a cut:

$$cut(C_1, C_2) = \sum_{u \in C_1, v \in C_2} A(u, v) \tag{5.5}$$

The idea is to find the *minimum cut* but a greedy approach would lead to partition the graph into very small sets. In order to avoid this phenomena Shi and Malik proposed the minimization of the *Normalized cut* (Ncut):

$$Ncut(C_1, C_2) = \frac{cut(C_1, C_2)}{assoc(C_1, V)} + \frac{cut(C_1, C_2)}{assoc(C_2, V)} \tag{5.6}$$

where $assoc(C_i, V) = \sum_{u \in C_i, t \in V} A(u, t)$ is the total connection from the nodes in C_i to all nodes in the graph. In this way the cut cost becomes a fraction of the total edge connections to all the nodes in the graph. Unfortunately, finding the minimum normalized cut is an NP-complete problem. However, an approximate discrete solution can be found efficiently by constraining the problem (this part is omitted for simplicity).

Let D be the degree matrix of A:

$$D(i, i) = \sum_{j=1}^{p} A(i, j) \tag{5.7}$$

where $i = 1 \ldots P$. It can be demonstrated that finding the minimum Ncut is equivalent to solve the generalized eigenvalues system:

$$(D - A)y = \lambda D y \tag{5.8}$$

The matrix $L = (D - A)$ is also called *Laplacian matrix* and it is known to be positive and semidefinite (Pothen et al., 1990). If y is relaxed to take on real values, the solution can be found by using the second smallest eigenvector. In the original formulation y would have taken on two discrete values (for example 0 or 1) so that the values of each entry would have determined the clustering. For example, if there are four features and $y = [0, 1, 0, 1]$ then one subgroup would be composed by the first and the third feature while the other by the second and the fourth feature. The only reason why this is not the optimal solution of the original problem is that y does not take on only two discrete values so a threshold need to be set. On the other hand this is what makes the problem tractable.

Figure 5.12 shows the same affinity matrix shown in Figure 5.11 rearranged after the spectral cluster algorithm. In the rearranged affinity matrix it is easy to see the block diagonal structure where each of the three clusters is represented by a bright square which signifies high affinity between the trajectories clustered together. The rearrangement of Figure 5.12 had no misclassification, hence the black stripes that cross the white squares are not errors but trajectories whose spaces are not particularly similar to the rest of the cluster.

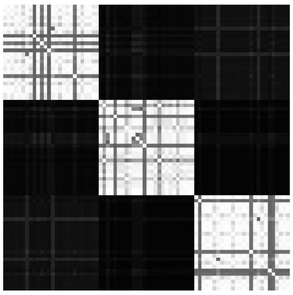


Figure 5.12: Example of affinity matrix rearranged after the spectral clustering. This is the same affinity matrix shown in Figure 5.11.

In Figure 5.13 the LSA algorithm is summarized. It starts from the trajectory matrix W built from P features tracked from F frames. The first step is the rank estimation r of W which is obtained using a model selection technique. In the second step the data are projected into a unit sphere of dimension r exploiting SVD and the rank just computed. The third step consists in estimating the subspace bases of each trajectory by sampling the n nearest neighbors of each trajectory in the sphere space and using SVD for the bases computation. In the fourth step the affinity matrix is built computing the principal angles between every pair of subspaces. The last step consists in clustering the affinity matrix, the result is a block structured affinity matrix which correspond to a segmentation of the features.

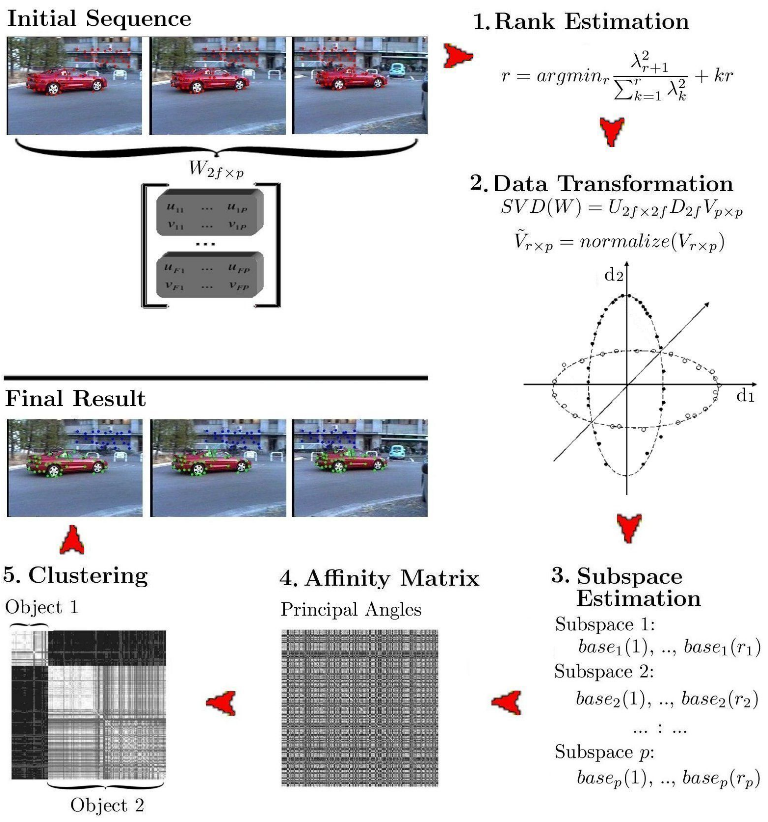


Figure 5.13: LSA flow. starting from a sequence of tracked features (red dots) through the five steps which end with the final result where the features belonging to the two objects are segmented (blue and green dots).

Figure 5.14 shows that the LSA algorithm is the best performing between the tested three with a total segmentation error of about 8%;

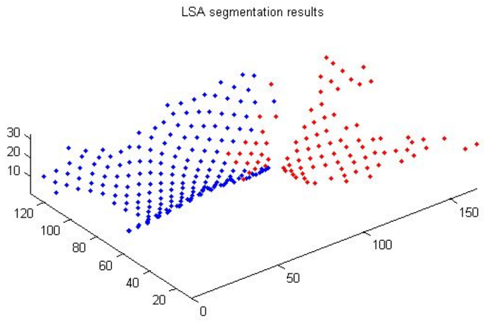


Figure 5.14: An example of motion segmentation using the Local Subspace Affinity (LSA).

The algorithm correctly estimate the points in the border zone, but we have some (expected) errors in the joint zone. but the interesting thing is that the mistakes out of the critical expected regions are very few. This is probably related to the fact that the LSA algorithm is more robust to the noise possibly introduced by soft-tissue artifacts then the other two considered approaches.

Because of the results shown in these preliminary tests, we propose to use LSA algorithm to segment the bodies.

5.4 Joint Reconstruction

Now that the points have been assigned to the respective body parts, it is now possible to compute the position and properties of the joint. The theoretical property used to perform the computation of the joint is that the subspaces computed from the trajectories of two body parts intersects. Such common intersection can be used to identify the joint position and properties as noticed by (Yan and Pollefeys, 2008; Tresadern and Reid, 2005) for image data and by (Fayad et al., 2009) for 3D data. We use the computational tool developed in the latter work to perform the estimation of the joint values.

This method is composed of three steps:

1. we refine a factorization method for Structure from Motion (SfM) by applying a variation of the weighted PCT, based on least-squares optimization; so that we have a first rigid approximation for each segment of the non-rigid bodies.

2. we use a quadratic model for non-rigid bodies, initialized by the first estimate of the rigid segment, to compute a more accurate rigid component of the non-rigid segments.

3. we combine the techniques for articulated SfM and our model for the non-rigid segments, to provide a final 3D articulated model of the human body.

5.4.1 Factorization method

Factorization methods for Structure from Motion are a family of image based algorithms that model moving objects as a product of two factors: *motion* and *shape*. The shape parameters are defined as the 3D geometric properties of the object; the motion parameters are defined as the time-varying parameters of the motion (e.g. rotations and translations of the rigid body) that the shape performs in a metric space.

The factorization method proposed by Fayad (Fayad, 2008) assumes a set of P 3D feature points being tracked over F frames. The method relies on the key fact that 3D trajectories of points belonging to the same body share the same global properties.

From the Section 5.2, we have a matrix W that represents the trajectories of all the feature points as:

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1P} \\ \vdots & \ddots & \vdots \\ w_{F1} & \cdots & w_{FP} \end{bmatrix} \tag{5.9}$$

where $w_{ij} \in \mathbb{R}^3$ is a 3 elements column vector that contains the 3D coordinates of point j at frame i. Each 3D point w_{ij} can be written, in homogeneous coordinates, as:

$$w_{ij} = [\mathbf{R}_i|\mathbf{t}_i] \begin{bmatrix} s_j \\ 1 \end{bmatrix} \tag{5.10}$$

where $s_j = [x_j, y_j, z_j]^T$ are the coordinates of the point j on a local reference frame, and $\mathbf{M}_i$ and $\mathbf{t}_i$ are respectively the rotation matrix and the translation vector that describe s_j with respect to a global reference frame. Stacking these equations for all the F frames and P points, we have:

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_F \end{bmatrix} = \begin{bmatrix} M_1 \\ \vdots \\ M_F \end{bmatrix} \begin{bmatrix} S_1 & \cdots & S_p \end{bmatrix} + \begin{bmatrix} T_1 \\ \vdots \\ T_F \end{bmatrix} = MS + T \tag{5.11}$$

where $T_i = [t_i, 1_P^T]$, with 1_P^T being a vector of P elements with all entries equal to 1. The translational component t_i can be computed as the coordinates of the centroid of the point cloud at each frame W_i. Thus, it can be easily eliminated by registering, at each frame, the point cloud to the origin i.e at each frame we subtract to the coordinates of every point the mean of the point cloud coordinates. In this scenario, it frequently occurs that, instead of W, we consider a registered form of this matrix i.e. we use a matrix $\tilde{W}$ such that:

$$\tilde{W} = W - T = MS \tag{5.12}$$

In this equation M is a $< 3F \times r >$ matrix and S a $< r \times P >$ matrix, where r is the rank of the matrix $\tilde{W}$ and it depends by the type of shape considered.

Rigid bodies

A rigid body moving rigidly brings the dimensionality of the bilinear models to either $r \leq 3$, if we consider the registered trajectory matrix, or $r \leq 4$ if we consider the translation vector too. Given this rank constraint, and remembering the equation 5.12, we can compute a factorization of $\tilde{W}$ by performing a *Singular Value Decomposition* (SVD) giving:

$$\tilde{W} = MS = U_r \Sigma_r V_r^T \tag{5.13}$$

where U_r is a $3F \times r$ orthogonal matrix, Σ_r a $r \times r$ diagonal matrix and V_r a $P \times r$ orthogonal matrix. This initial decomposition via SVD can provide a first affine fit of the motion and shape components M and S such that:

$$\tilde{M} = U_r \Sigma_r^{1/2} \qquad \text{and} \qquad \tilde{S} = \Sigma_r^{1/2} V_r^T \tag{5.14}$$

The initial factorization proposed in equation 5.14 do not guarantee that $\tilde{M}$ is in fact a collection of F 3×3 rotation matrices. Since this transformation is valid up to an affine transformation i.e. $\tilde{W} = \tilde{M} Q Q^{-1} \tilde{S}$ we seek a specific transformation Q which enforces the metric properties of M. This can be achieved by imposing orthogonality constraints on $\tilde{M}_i Q$, which is done by solving the set of linear equations for all the F frames:

$$\begin{cases} m_{ik}^T H m_{ik} = 1 \\ m_{ik}^T H m_{il} = 0, \ \forall l \neq k \end{cases} \tag{5.15}$$

with $k, l = 1, 2, 3$, m_{ik} and m_{il} are respectively the k-th and l-th row of matrix $\tilde{M}_i$, and $H = QQ^T$ is symmetric matrix (as Q is upper triangular). Q can thus be recovered from H by using Cholesky decomposition. We update the factorization in equation 5.14 to:

$$M = \tilde{M} Q \qquad \text{and} \qquad S = Q^{-1} \tilde{S} \tag{5.16}$$

When the scene is composed of N rigid objects moving independently, the same considerations are valid. The model is simply expanded for each of the different independent objects, with S showing a block diagonal structure:

$$\tilde{W} = \begin{bmatrix} M_1 & \cdots & M_N \end{bmatrix} \begin{bmatrix} S_1 & & \\ & \ddots & \\ & & S_N \end{bmatrix} \tag{5.17}$$

In this case the rank condition becomes $r \leq 3N$.

Articulated bodies

However if the rigid objects are linked by joints their motions are not independent and there is a loss in the degrees of freedom of the system. This constraint on the movement manifests itself in the measurement matrix W as a decrease in rank (Tresadern and Reid, 2005; Yan and Pollefeys, 2006a). For the sake of simplicity we will only consider systems of two rigid bodies linked by a joint, two types of joints are here considered: the *universal joint* and the *hinge joint.*

Universal joint

By universal (*spherical*) joint we mean a kind of joint in which each body is at a fixed distance to the joint center, being the relative position of the bodies constrained, but their rotations remaining independent. At each frame the shapes connected by a joint satisfy the following relation:

$$\mathbf{R}_1 d_1 + t_1 = -\mathbf{R}_2 d_2 + t_2 \tag{5.18}$$

where t_1 and t_2 are the 3D shape centroids of the two objects, $\mathbf{R}_1$ and $\mathbf{R}_2$ the 3×3 rotation matrices and d_1 and d_2 the 3D displacement vectors of each shape from the central joint.

Thus we can write t_2 as a function of t_1, and we are now able to factorize the trajectory matrix as:

$$W = [W_1|W_2] = [R_1 \; R_2 \; t_1] \left[\begin{array}{cc} S_1 & d_1 \\ 0 & S_2 + d_2 \\ 1 & 1 \end{array} \right] \tag{5.19}$$

where W_1 and W_2 are respectively the measurement matrices for the first and second body.

By using the equation 5.18 the joint parameters d_1 and d_2 are easily computed once we have found the motion parameters R_1, R_2, t_1 and t_2.

After the registration of each body to the origin of the global reference frame, the registered trajectory matrix $\tilde{W}$ is a $3F \times (P_1 + P_2)$ matrix with rank 6, and so:

$$\tilde{W} = [R_1|R_2] \left[\begin{array}{cc} S_1 & 0 \\ 0 & S_2 \end{array} \right] \tag{5.20}$$

Now we can perform the truncated SVD with $k = 6$.

$$\tilde{W} = U_k \Sigma_k V_k^T = [U_1|U_2]_{3F \times 6} [V_1|V_2]_{6 \times (P_1 + P_2)} \tag{5.21}$$

However the factorization is not final as $[V_1|V_2]$ is a dense matrix. If we define an operator $Nl(\cdot)$ that returns the left null-space of its argument, we can define a 6×6 transformation matrix T_U such that:

$$T_U = \left[\begin{array}{c} Nl(V_2) \\ Nl(V_1) \end{array} \right] \tag{5.22}$$

We can now recover S by pre-multiplying it by T_U:

$$S = \left[\begin{array}{c} Nl(V_2) \\ Nl(V_1) \end{array} \right] [V_1|V_2] = \left[\begin{array}{cc} Nl(V_2)V_1 & Nl(V_2)V_2 \\ Nl(V_1)V_1 & Nl(V_1)V_2 \end{array} \right] = \left[\begin{array}{cc} S_1 & 0 \\ 0 & S_2 \end{array} \right] \tag{5.23}$$

As we must keep the original data unaltered, we have to post-multiply $[U_1|U_2]$ by T_U^{-1}:

$$M = [U_1|U_2] \begin{bmatrix} Nl(V_2) \\ Nl(V_1) \end{bmatrix}^{-1} = [M_1|M_2] \tag{5.24}$$

Hinge Joint

In a hinge joint, two bodies can rotate around an axis such that the distance to that rotation axis is constant. Therefore their rotation matrices R_1 and R_2 are not completely independent.

From the geometry of the joint, we can see that every vector belonging to any of the two bodies, that is parallel to the joint axis, must remain so throughout the movement. Without loss of generality, we can choose a local reference frame with x-axis coincident with the axis of rotation of the joint. Let $e_x = [1\ 0\ 0]^T$ be the x-axis unit vector. Applying a general 3×3 rotation matrix $R = [c_1\ c_2\ c_3]$ to e_x will result in $c_1 \cdot e_x$, the only column of the rotation matrix that affects vectors parallel to the x-axis is the first one. Therefore, to comply with the joint constraints, the first column of R_1 must be equal to the first column of R_2. We can now define the rotation matrices as $R_1 = [c_1\ c_2\ c_3]$ and $R_2 = [c_1\ c_4\ c_5]$. As all the points belonging to the rotation axis must fulfill both movement conditions, thus, when considering registered data, $\tilde{W}$ will then be given by:

$$\tilde{W} = [c_1\, c_2\, c_3\, c_4\, c_5] \begin{bmatrix} x_1^1 & \cdots & x_{P_1}^{(1)} & x_1^{(2)} & \cdots & x_{P_2}^{(2)} \\ y_1^1 & \cdots & y_{P_1}^{(1)} & 0 & \cdots & 0 \\ z_1^1 & \cdots & z_{P_1}^{(1)} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & y_1^{(2)} & \cdots & y_{P_2}^{(2)} \\ 0 & \cdots & 0 & z_1^{(2)} & \cdots & z_{P_2}^{(2)} \end{bmatrix} \tag{5.25}$$

Once again we use the truncated SVD of $\tilde{W}$ as the first step on the parameter estimation. In this case we use $k = 5$ giving:

$$\tilde{W} = U_k \Sigma_k V_k^T = [U_1|U_2]_{3F\times 5}[V_1|V_2]_{5\times(P_1+P_2)} \tag{5.26}$$

As we have seen when we spoke about universal joints, this matrix $[V_1|V_2]$ is a dense matrix, but what we need is to compute a matrix S with the structure defined in equation 5.25. Let T_H be a transformation matrix such that:

$$T_H = \begin{bmatrix} \mathbf{b}^T \\ Nl(V_2) \\ Nl(V_1) \end{bmatrix} \tag{5.27}$$

where $Nl(\cdot)$ is the operator that returns the left null-space of its argument (as defined previously), and $\mathbf{b}^T = [1\ 0\ 0\ 0\ 0\ 0]$. By pre-multiplying $[V_1|V_2]$ with

T_H we leave the first row intact and we zero-out some entries in order to get the desired structure of S. Again, we need to post-multiply $[c_1\ c_2\ c_3\ c_4\ c_5]$ with T_H^{-1} to keep the original data unaltered.

Weighted factorization

The algorithms described in Section 5.4.1 solve the problem when the observed bodies are rigid. When dealing with non-rigid bodies they can still be used as a coarse rigid approximation of the data.

When using SVD to estimate the motion and shape parameters, the resulting shape will be the one that minimizes the error in a least-squares sense over all the frames. Nonetheless this might not be the best representation of the rigid component of the non-rigid shape. Factorizing with the previous algorithms can be seen as averaging the shape throughout the frames, resulting in an attenuation of the deformations. The algorithm proposed by Fayad (Fayad, 2008) is an approach that uses a weighted SVD in order to penalize the contribution of the points which deform most. By doing so we will attenuate the contribution of the deformations, obtaining a more accurate rigid representation of the body.

In order to do that, he proceed in three steps[1]:

1. compute the rank-3 approximation of W and factorize into M and S using the method described in Section 5.4.1.

2. for each point j, compute the *weight matrix* C_j as the covariance of the deviation between the real trajectory of the point and the approximated one by using rigid factorization.

3. recompute M and S by using the matrices C_j as weight.

Figure 5.15 shows the points cloud and the estimated rotational joint, the position of the elbow. The first approximation of the human elbow is an axial joint, this is a good model when we consider a low number of feature points; in this case we have over 300 feature points near the elbow, and so the deformation of the skin surface introduce secondary motions. For this reason we model the elbow joint as a generic rotational joint.

The first possible reason is that the outliers after the segmentation stage can produce an instability in the factorization step of the algorithm. For this reason we have tried to manually delete the outliers, the result i better than in the previous case but not so much. So there are some other reasons.

One critical point can be the quite planarity of the reconstructed points, in this case the configuration would degenerate and so the computation

[1] This is only a brief description, for more details please see the reference

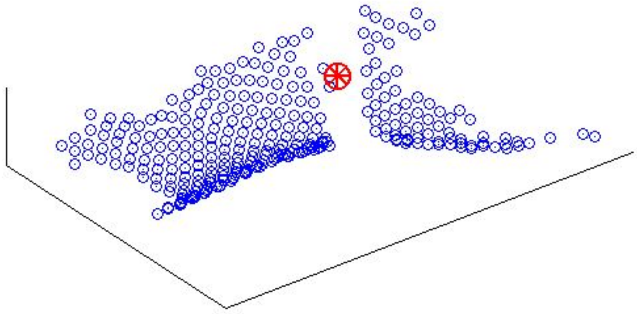


Figure 5.15: An example of joint estimation using the Fayad's algorithm.

of some internal steps of the algorithm can be affected from matrix rank deficiency. One more critical point in this case is the co-planarity of the points and the motion.

Part III

Experimental section

Chapter 6

Experimental section

In order to test the proposed algorithms, in this work we developed a physical prototype of 3D scanner based on multiple stereo cameras. In this section we present a complete description of the prototype with details about the used hardware. We also test the algorithms by two kinds of experiments: in the first one we analyze the results of the static reconstruction, in the second one we did a preliminary test about motion capture application.

6.1 Experimental setup

The developed system, shown in Figure 6.1, is composed by a set of 12 cameras, connected in a configuration of 6 stereo-pairs, and a workstation, used for camera management and image processing. The stereo-pairs are located around a workspace of dimension 400mm×200mm×200mm at a distance of about 400mm. A frame of aluminum profiles ensures stable positioning of each stereo-pair with respect to the others; also the thermal deformation were taken into account in the geometric design in order to minimize them.

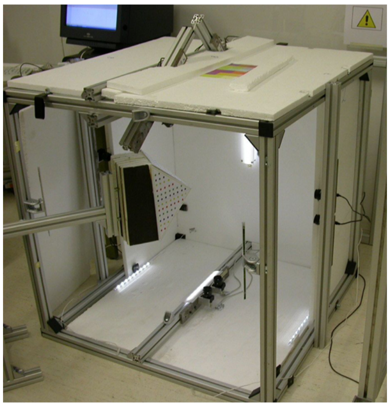


Figure 6.1: The experimental setup developed for the algorithms test.

Each stereo-pair is composed from two Point Grey Chamaleon USB cameras (Point Grey Research, Inc., 2009), with CMOS RGB sensor with maximal resolution of 1280×960 pixels. The two cameras are stuck on an iron bar, and then connected to the structure. The cameras are all triggered in order to ensure that all the images are acquired simultaneously. The lenses are standard C-mount lenses with focal length 6mm and f-number 1.4; this ensure a field of view of about 400mm×300mm at a distance of about 400mm. With reference to Figure 6.2, the baseline (b) is set at about 140mm, in order to have a compromise between accuracy in triangulation and common marker detection (Brown et al., 2003). The angle between the optical axis of two cameras is computed in order to have the maximal overlapping between the two fields of view at a distance (d) of 400mm; in this case the angle (α) is about 15°.

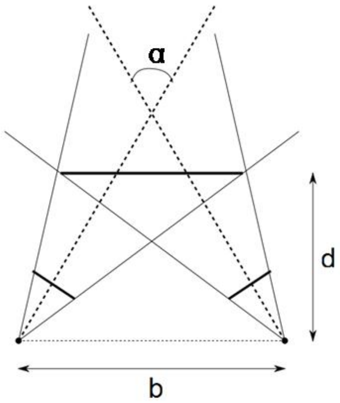


Figure 6.2: The angle between cameras in a stereo-pair. In the developed prototype $\alpha \cong 15°$.

A system of LED lightning and diffusive polystyrene panels is used in order to have in the workspace the light as most diffuse as possible.

We wrote a software for camera management and synchronous image acquisition. The software provides the following tasks:

- recognizes all the cameras connected to the PC by ID-number
- switch on and initializes all the cameras
- set camera parameters as described in a configuration file
- acquires synchronous images from all the connected cameras
- save the images in folders and files, according to ID-number of each camera and a look-up table.

Using this physical prototype we developed a set of experiments in order to verify the proposed algorithms. In the following two sections we present the experiments for static and dynamic case.

6.2 Experiments of static reconstruction

What about the static 3D reconstruction we have done two different kinds of experiments: one for the verification of shape reconstruction from a single stereo-pair and one for the verification of the data fusion algorithm in terms of uncertainty.

6.2.1 Single stereo-pair verification

In order to test and evaluate the accuracy in 3D reconstruction from a single stereo-pair, we provide an experiment with a plane, inclined with respect to the stereo-pair. In Figure 6.3 are shown the acquired images of the inclined plane.

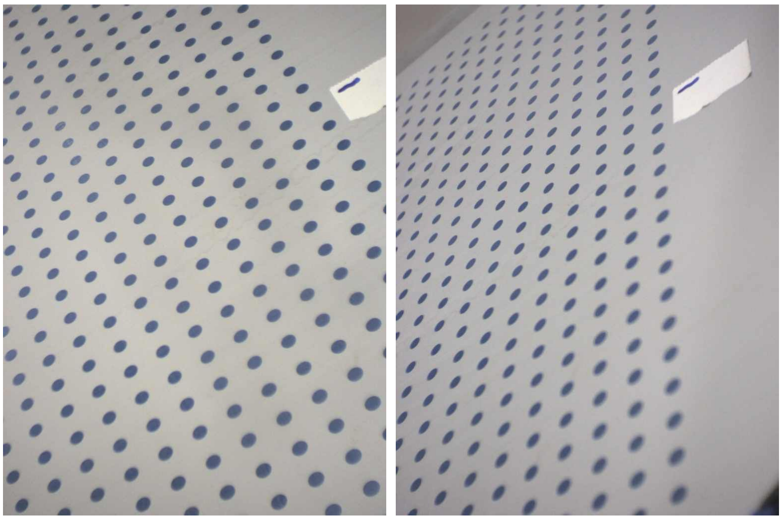


Figure 6.3: The acquired images in the inclined plane experiment.

We segment all the circular markers in the images and reconstruct the 3D position of each point by using the algorithm described in Chapter 4. In this case the markers are all of the same color but the low density of feature points and the orientation of the stripes make the algorithm adapt. The 3D reconstruction of the plane is shown in Figure 6.4.

From the 3D reconstruction we compute the distance of each point from the best fitting plane. The out of plane distance of each point and the histogram of this distribution are shown in Figure 6.5.

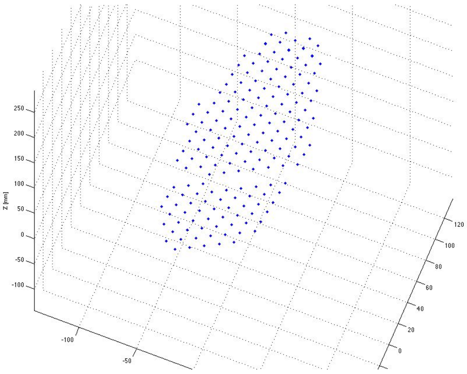


Figure 6.4: The 3D reconstruction of the inclined plane experiment.

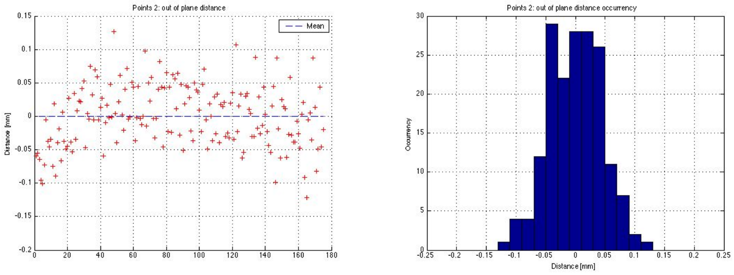


Figure 6.5: The acquired images in the inclined plane experiment.

In this experiment we can see that the standard deviation of the points from the best fitting plane is about 0.04mm (40μm), and it is a good result for our main goal.

6.2.2 Data fusion verification

As described in a first-part experiment in (De Cecco et al., 2009a), an inclined can provided with colored markers on its lateral surface and positioned by a Cartesian robot, is acquired by two stereo-pairs, which are angularly spaced apart of nearly 90° as illustrated in Figure 6.6. Starting from an initial position (A) the can is translated along a straight trajectory to a final position (B) and the markers on its surface are acquired both in the initial and final position.

The acquired colored markers yield two sets of points (Σ_1, Σ_2) for each stereo-pair and these two sets are reconstructed in both positions A e B. The reconstructed 3D sets of Σ_1, Σ_2 in position A and B are depicted in Figure 6.7 with their corresponding covariance ellipsoids; for each position a compatibility test based on Mahalanobis distance is made. The points that passed the compatibility tests have overlapping ellipsoids, as can be seen in Figure 6.8 for two corresponding points (1 marker) one belonging to Σ_1 and

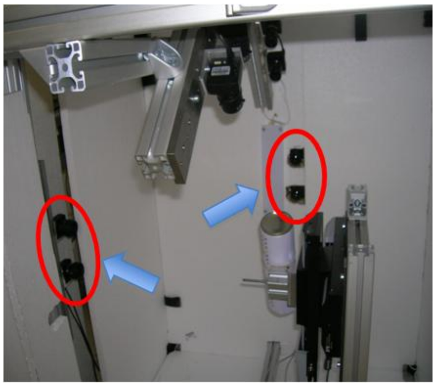


Figure 6.6: The inclined-can experiment setup. It is shown a can with markers, the two stereo-pairs and the positioning robot.

the other one to Σ_2.

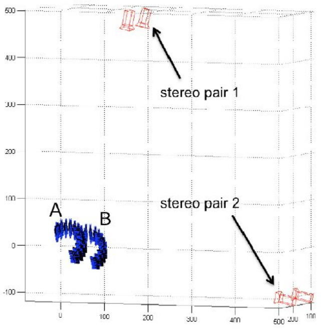


Figure 6.7: The markers on the can reconstructed from 2 stereo-pairs.

For all corresponding points it is possible to fuse their covariance ellipsoids as explained in Section 4.4, in order to obtain a fused point and its covariance ellipsoid as shown in Figure 6.8 for one marker. It is clear that the uncertainty associated with the fused point is significantly reduced with reference to uncertainties obtained from a single stereo pair. This useful result is particularly true for the two considered stereo pairs, since they are angularly spaced apart of 90°.

The resulting uncertainty for all the markers of position A is shown in Figure 6.9 where the smaller ellipsoids in the central part belong to the fused points that are the compatible ones between Σ_1 and Σ_2.

After computing the covariance of each marker, stating compatibility and fusing points in both positions A and B, we have measured the translation of the can from A to B by computing the mean displacement between corresponding points in the two positions. The point matching is computed

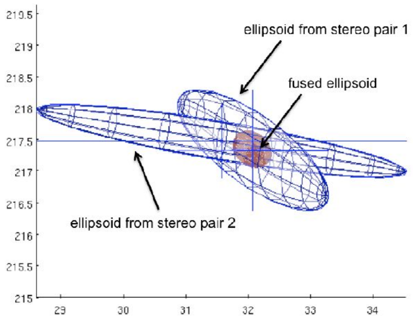


Figure 6.8: Covariance elliposids of two corresponding points (obtained from the same marker) acquired by two stereo-pairs.

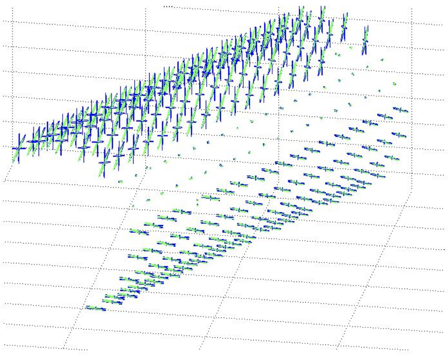


Figure 6.9: Covariance ellipsoids of all the markers in position A.

using an ICP algorithm that also employs the color information for a more robust result. The reference displacement superimposed by the Cartesian robot is of 38.000mm with 1μm of spherical overall accuracy. The measured mean displacement is of 37.95mm. Its uncertainty is computed with the squared root of the maximum eigenvalue of the covariance of the displacements set (the blue arrows in Figure 6.10) multiplied by the coverage factor for 95.5% confidence level and the resulting value is of 0.14mm. This value is fully compatible with the entity of the uncertainty of the points estimated by the propagation and fusion method showing that the initial parameters calibration is good.

6.3 Experiments of *motion capture*

For the evaluation of the motion capture framework we developed two kinds of experiments. In the first stage we improved a rigid bodies articulated

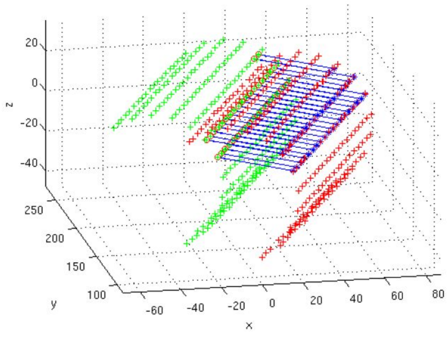
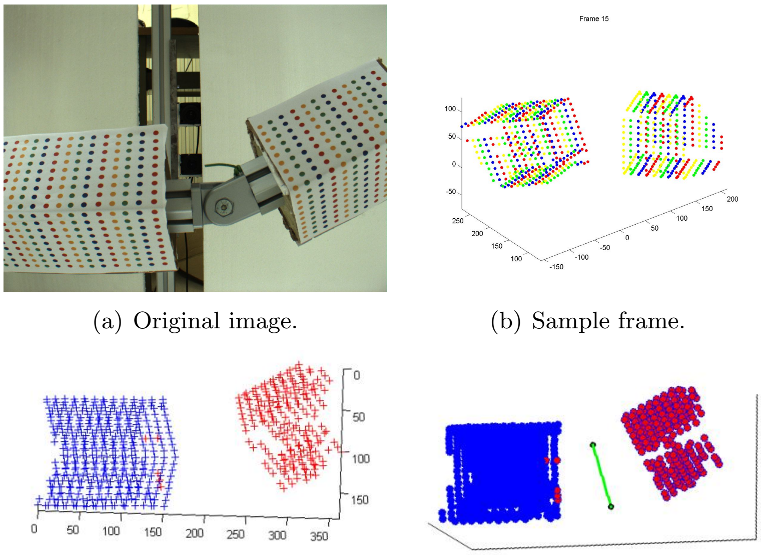


Figure 6.10: Blue vectors shown the estimated displacements of matching points of A (green markers) and B (red markers).

arm, in the second one we used a real human arm.

6.3.1 Rigid bodies

In the first experiment two parallelepipeds are connected with an axial joint in order to create a human arm mock-up. The figure 6.11 shows a sample of the acquired images, a reconstructed frames, the segmentation results using the LSA algorithm and the estimated axial joint.



(a) Original image.

(b) Sample frame.

(c) LSA segmentation.

(d) Estimated Joint.

Figure 6.11: The first experiment: two rigid bodies with axial joint.

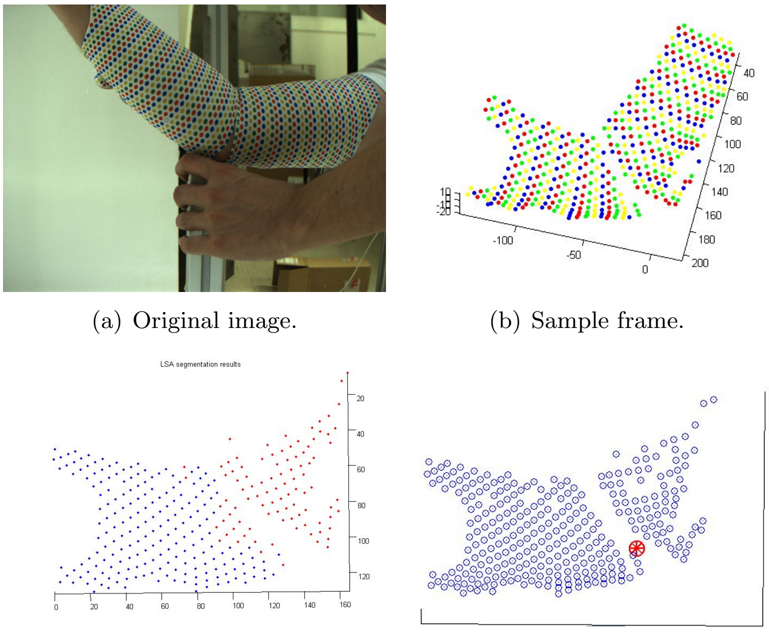
The sequence is composed from 15 frames acquired from 2 stereo-pairs, which are angularly spaced apart of nearly 180°. A pattern, similar to the one shown in Figure 5.3, is superimposed to the parallelepipeds. Using the algorithm for static reconstruction described in Chapter 4, we provide the 3D reconstruction of each frame in the sequence. The matching between frames and trajectory matrix generation is performed by using the novel algorithm that combine NN and PA, separately for the clouds acquired from the two stereo-pairs.

For the motion segmentation step we perform the LSA algorithm; as shown in Figure 6.11, this method gives a segmentation error of only 1%.

Also the joint parameters estimation is performed by using the Fayad code, as presented in Section 5.4. In this case we know a priori the kind of joint (hinge) and so we can compute the real axis of rotation.

6.3.2 Deformable bodies

In the second experiment we have acquired a sequence of a real human arm performing a bending movement. The figure 6.11 shows a sample of the acquired images, a reconstructed frames, the segmentation results using and the estimated universal joint.



(a) Original image.

(b) Sample frame.

(c) LSA segmentation.

(d) Estimated Joint.

Figure 6.12: The second experiment: a real human arm.

In this case we used a sequence of 8 frames acquired from a single stereo-pair, and we reconstruct the 3D position of markers in each frame by using the static reconstruction algorithm. The matching between frames is performed by using the novel algorithm that combine NN and PA.

In this experiments the total segmentation error using LSA algorithm is of about 8%. The first approximation of the human elbow doing a bending movement is an axial joint; nevertheless we have two conditions that disturb this approximation: the high number of feature points and the skin-bones internal movement and muscolar deformations. For this reason we model the elbow joint as a generic rotational joint (universal).

In this configuration, the joint reconstruction has two critical points:

- the outliers after the segmentation stage can produce an instability in the factorization step of the algorithm.
- the co-planarity of the points and the motion. If the point cloud is quite planar the configuration would degenerate and so the computation of some internal steps of the algorithm can be affected from matrix rank deficiency.

Chapter 7

Conclusions

In this work we presented a set of algorithms for 3D shapes reconstruction and motion segmentation for the application to multiple stereo-camera system and deformable bodies. In particular we were interested in the application to human body analysis.

In the first part of the work, we presented a complete framework for 3D static scanning, based on a multiple stereo vision approach. The framework provides the 3D reconstruction of a set of markers belonging to the surface of a target body; the 3D position of each marker is computed from each stereo-pair by using the triangulation algorithm. Later the position of points viewed from more than one pair are fused performing a compatibility analysis based on uncertainty ellipsoids of each point.

An extension to the motion analysis is presented in the second part of the work. We describe here a complete framework for the points matching among a sequence of frames, the segmentation of the bodies based on the motion analysis, and the identification of the joint parameters. This algorithm is oriented to the application to a collection of 3D data of non-rigid bodies.

A prototype of 3D scanner was developed at the Mechatronics Laboratory at the University of Trento and the presented algorithm were tested in an experimental section that comprises the analysis of static reconstruction, of planar and cylindrical surfaces, and motion analysis, of rigid and non-rigid jointed bodies.

This work represents a solid basis for the realization of a 3D scanner based on vision technologies. All the algorithm developed are implemented as modular software, this will intervene and improve on individual parts. In particular some points to improve are:

- replace the colored circular markers with natural features, this can be done modifying only the marker detection and feature matching algorithms, and developing a new method for the estimation of uncertainty in feature detection.

- replace the multiple stereo approach with the multiple camera one, this can be done modifying the triangulation algorithm and, slightly, the uncertainty propagation one. Notice that in this case there is no longer needed of compatibility analysis and points fusion stages.

Most of the work has been supported by Delta R&S Company and part of the algorithms was implemented in their products.

The work here presented resulted in two congress paper (De Cecco et al., 2009a; De Cecco et al., 2009c), a journal paper (De Cecco et al., 2009b) and a submitted congress paper.

Bibliography

Amat, J., Frigola, M., and Casals, A. (2002). Selection of the best stereo pair in a multi-camera configuration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002)*, volume 4, pages 3342–3346.

Barnard, R. W., Pearce, K., and Schovanec, L. (2001). Inequalities for the perimeter of an ellipse. *Journal of Mathematical Analysis and Applications*, 260(2):295–306.

BIPM, IEC, IFCC, ISO, IUPAP, and OIML (1993). *Guide to the Expression of Uncertainty in Measurement.* ISO - International Organization for Standardization.

Born, M. and Wolf, E. (1999). *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light.* Cambridge University Press, 7th edition.

Boujou (2009). Boujou. `http://www.vicon.com/boujou/`.

Brown, M. Z., Burschka, D., and Hager, G. D. (2003). Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008.

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Celenk, M. and Bachnak, R. (1990). Multiple stereo vision for 3d object reconstruction. In *IEEE International Conference on Systems Engineering*, pages 555–558.

Chen, J., Ding, Z., and Yuan, F. (2008). Theoretical uncertainty evaluation of stereo reconstruction. In *The 2nd International Conference on Bioinformatics and Biomedical Engineering (ICBBE 2008)*, pages 2378–2381.

Corazza, S., Mündermann, L., and Andriacchi, T. (2007). A framework for the functional identification of joint centers using markerless motion capture, validation for the hip joint. *Journal of Biomechanics*, 40(15):3510–3515.

De Cecco, M., Baglivo, L., Parzianello, G., Lunardelli, M., Setti, F., and Pertile, M. (2009a). Uncertainty analysis for multi-stereo 3d shape estimation. In *IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement*, pages 22–27, Bucharest, Romania.

De Cecco, M., Pertile, M., Baglivo, L., Lunardelli, M., Setti, F., and Tavernini, M. (2009b). A unified framework for uncertainty, compatibility analysis and data fusion for multi-stereo 3d shape estimation. *IEEE Transactions on Instrumentation Measurements.* Accepted for publication.

De Cecco, M., Pertile, M., Baglivo, L., Parzianello, G., Lunardelli, M., Setti, F., and Selmo, A. (2009c). Multi-stereo compatibility analysis for 3d shape estimation. In *XIX IMEKO World Congress, Fundamental and Applied Metrology*, pages 1909–1914, Lisbon, Portugal.

Devernay, F. and Faugeras, O. (2001). Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured enviroments. *Machine Vision Applications*, 13(1):14–24.

Duda, R. O., Hart, P. E., and Stork, D. (2000). *Pattern Classification.* Wiley-Interscience Publication.

Eggert, D. W., Lorusso, A., and Fisher, R. B. (1997). Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision Application*, 9(5-6):272–290.

Fayad, J., Del Bue, A., Agapito, L., and Aguiar, P. (2009). Human body modeling using quadratic deformations. In *7th EUROMECH Solid Mechanics Conference, Lisbon, Portugal.*

Fayad, J. R. K. (2008). Articulated three-dimensional human modeling from motion capture systems. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Fischler, M. and Bolles, R. (1987). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In Fischler, M. A. and Firschein, O., editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 726–740. Los Altos, CA.

Forsyth, D. A. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall.

Goldstein, H., Poole, C. P., and Safko, J. L. (2001). *Classical Mechanics*. Addison Wesley, 3rd edition.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD (USA), 3rd edition.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Hernandez Esteban, C. and Schmitt, F. (2002). Multi-stereo 3d object reconstruction. In *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*, pages 159–166.

Horn, B. K. P. (2000). Tsai's camera calibration method revisited. `http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf`.

Ioannidis, D., Tzovaras, D., Damousis, I. G., Argyropoulos, S., and Moustakas, K. (2007). Gait recognition using compact feature extraction transforms and depth information. *IEEE Transactions on Information Forensics and Security*, 2:623–630.

ISO 12640-3:2007 (2007). Graphic technology. pre-press digital data exchange - part 3: Cielab standard color image data (cielab/scid).

Kanatani, K. (1996). *Statistical optimization for geometric computation: theory and practice*. Elsevier Science Inc. New York, NY, USA.

Kanatani, K. (2001). Motion segmentation by subspace separation and model selection. In *Proceedings of 8-th International Conference on Computer Vision*, pages 586–591.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2003). *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag.

Mayer, R. (1999). *Scientific Canadian : invention and innovation from Canada's National Research Council*. Raincoast Books, Vancouver.

Nedevschi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., and Pocol, C. (2004). Spatial grouping of 3d points from multiple stereovision sensors. In *IEEE International Conference on Networking, Sensing and Control*, volume 2, pages 874–879.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press.

OrganicMotion (2009). Organic motion. `http://www.organicmotion.com/`.

Point Grey Research, Inc. (2009). Chamaleon datasheet. `http://www.ptgey.com/products/chamaleon/Chamaleon_datasheet.pdf`.

Pothen, A., Simon, H. D., and Liou, K.-P. (1990). Partitioning sparse matrices with eigenvectors of graphs. *Society for Industrial and Applied Mathematics*, 11(3):430–452.

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905.

Spitz, S. N. (1999). *Dimensional inspection planning for coordinate measuring machines*. PhD thesis, University of Southern California, Los Angeles, CA, USA. Adviser - Requicha, Aristides A.

Tresadern, P. and Reid, I. (2005). Articulated structure from motion by factorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1110–1115, San Diego, California.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms (ICCV 1999)*, pages 298–372. Springer-Verlag.

Tron, R. and Vidal, R. (2007). A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE conference on computer vision and pattern recognition*, volume 4.

Trucco, E., Fusiello, A., and Roberto, V. (1999). Robust motion and correspondence of noisy 3-d point sets with missing data. *Pattern Recognition Letters*, 20(9):889–898.

Vicon (2009). Vicon. `http://www.vicon.com/`.

Vidal, R., Ma, Y., and Sastry, S. (2005). Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959.

Yan, J. and Pollefeys, M. (2006a). Articulated motion segmentation using ransac with priors. In *ICCV Workshop on Dynamical Vision*, pages 75–85.

Yan, J. and Pollefeys, M. (2006b). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *9th European Conference on Computer Vision (ECCV 2006)*, Graz, Austria.

Yan, J. and Pollefeys, M. (2008). A factorization-based approach for articulated non-rigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5).

Young, D. (1994). Stereoscopic vision and perspective projection. `http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision5.html`.

Zhang, S. and Yau, S.-T. (2006). High-resolution real-time 3d absolute coordinate measurement based on a phase-shifting method. *Optics Express*, 14(7):2644–2649.