Doctoral Thesis

Surface measurement and tracking of human body parts from multi station video sequences

Author(s):
D’Apuzzo, Nicola

Publication Date:
2003

Permanent Link:
https://doi.org/10.3929/ethz-a-004620239

Rights / License:
In Copyright - Non-Commercial Use Permitted
Surface Measurement and Tracking of Human Body Parts from Multi Station Video Sequences

Nicola D’Apuzzo

Zurich, October 2003
This publication is an edited version of:

Diss. ETH No. 15271

Surface Measurement and Tracking of Human Body Parts from Multi Station Video Sequences

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of
Doctor of Technical Sciences

presented by
NICOLA MICHELE D’APUZZO
Dipl. Masch. Ing. ETH
born July 11, 1970
citizen of Bellinzona, TI

accepted on the recommendation of
Prof. Dr. Armin Grün, examiner
ETH Zurich, Switzerland
Prof. Dr. Henrik Haggrén, co-examiner
Helsinki University of Technology, Espoo, Finland
Prof. Dr. Hans-Peter Meinzer, co-examiner
German Cancer Research Center, Heidelberg, Germany

Zurich 2003

IGP Mitteilung Nr. 81

Surface Measurement and Tracking of Human Body Parts from Multi Station Video Sequences
Nicola D’Apuzzo

Copyright © 2003

Institut für Geodäsie und Photogrammetrie
Eidgenössische Technische Hochschule Zürich
ETH Hönggerberg
CH-8093 Zürich

Alle Rechte vorbehalten

ISSN 0252-9335
ISBN 3-906467-44-9
Ce que peut la vertu d'un homme ne se doit pas mesurer par ses efforts, mais par son ordinaire.

*The strength of a man’s virtue must not be measured by his efforts, but by his ordinary doings.*

Blaise Pascal (1623-1662)
Abstract

This work pertains to surface measurement and surface tracking of human body parts using video sequences acquired by multiple cameras. Traditionally, and even today, research and commercial applications were concentrated either on the measurement and modeling of the human face and body or on the capture of movement of the whole body and facial expressions. In this work, a method is presented to treat both the surface measurement aspects and the surface tracking aspects in a unique process. The proposed method can be applied to measure either the surface of a static human body part or the moving surface of a dynamic event. In the latter case, the 3-D data gained can be of two different types: surface measurement of the part of interest in the form of a 3-D point cloud for each recorded time step or surface tracking in form of a vector field of 3-D trajectories.

The process is composed of seven steps: (1) calibration of the system, i.e., establishing the internal and external orientation of the cameras and the parameters modeling the lens distortions; (2) acquisition of multi-image sets and/or multi-image sequences; (3) matching process, i.e., establishing correspondences in the multi-images; (4) computation of the 3-D point cloud for each matched multi-image set; (5) surface tracking in the multi-image sequences; (6) establishing a 3-D vector field of trajectories (position, velocity, acceleration); and (7) tracking key-points in the vector field of trajectories.

In the case of surface measurement, only the first four steps are required. The accurate measurement of a human body part starts with an adequate acquisition of the required data. In the case of static surface measurement are used multi-images (multiple images acquired from different positions in the space describing the same scene), while in the case of dynamic surface measurement and surface tracking are used multi-image sequences (multi-images acquired during a time interval). For the acquisition of multi-images different systems can be applied with differing levels of quality depending on the cameras used.

The orientation and calibration processes establish the position and orientation of the camera sensors in 3-D space, the parameters describing the internal geometry of the imaging device, and the parameters modeling the distortions caused by the optical system. A thorough determination of all the parameters is required for accurate measurement using photogrammetric techniques.
The goal of the automatic matching process is the determination of a dense set of corresponding points in the multi-images on the part of surface of interest. The process uses a stereo matcher based on least squares matching techniques. The automatic matching process begins by defining several seed points. Starting from them, a dense and robust set of corresponding points covering the entire interested region is generated. The 3-D coordinates of the matched points are then computed by forward ray intersection using the results of the calibration process. The strategy is customized to the characteristics of the surface of the human body. Moreover, it is designed to reduce the required processing time to a minimum.

The basic idea of the multi-image tracking process is tracking corresponding points in the multi-images through the sequence and computing their 3-D trajectories. Velocities and accelerations are also computed at each time step. The process is based on least squares matching techniques. These are applied to determine the spatial correspondences between the images acquired simultaneously from different views, as well as to determine the temporal correspondences between subsequent frames.

The proposed process can be used to track well defined points on the human body surface. Trajectories of single points, however, are not sufficient to understand and record the motion and movement of a human or the changes of the surface of human body parts. Accordingly, the tracking process is extended to simultaneously track a dense set of points belonging to a common surface. In this case, the result of the tracking process can be considered to be a 3-D vector field of trajectories.

To solve additional problems caused by occlusions, lack of texture, loss of tracked points and the appearance of new points, the concept of key-points is introduced. The key-points are 3-D regions, defined in the vector field of trajectories, whose size can vary and whose position is defined by their center of gravity. The key-points are interactively defined in a graphical user interface and tracked simply; the position in the next time step is computed as the mean value of the displacements of all the trajectories contained inside the 3-D region.

Graphical user interfaces were developed and implemented for all of the processes employed, including the multi-image acquisition, the calibration and orientation procedures, the automatic matching process and the visualization of the results as 3-D point clouds and 3-D vector field of trajectories.

To demonstrate the multiple functionality of the method, three different applications are presented: high accuracy measurement of human faces using five CCD cameras, measurement of a blood vessel branching casting fixed in a rotating frame using three CCD cameras and full body motion capture without markers using video sequences acquired by two or three synchronized CCD cameras.
Riassunto

Questo lavoro tratta la misurazione e il tracking di superfici di parti del corpo umano utilizzando videosequenze registrate con più telecamere. Finora, la ricerca e le applicazioni commerciali in questo settore si sono concentrate unicamente sulla misurazione e modellizzazione del viso e del corpo umano e sulla registrazione del movimento del corpo umano o di espressioni del viso. In questo lavoro, invece, viene presentato un metodo che integra in un unico processo le procedure sia di misurazione di superfici sia di registrazione del movimento del corpo umano. Infatti, il metodo proposto può venire utilizzato per la misurazione di superfici statiche di parti del corpo umano, come pure per la misurazione di superfici in movimento. In quest’ultimo caso, i dati tridimensionali ottenuti possono produrre una misurazione della superficie della parte interessata sotto forma di nuvola di punti tridimensionali per ogni passo temporale oppure un tracking della superficie sotto forma di campo vettoriale di traiettorie tridimensionali.

L’intero processo è composto dai seguenti passaggi: (1) calibrazione del sistema, determinazione dell’orientamento interno ed esterno delle telecamere, inclusi i parametri di modellizzazione delle distorsioni ottiche; (2) acquisizione di multi-immagini e/o sequenze di multi-immagini; (3) processo di matching, determinazione di corrispondenze nelle multi-immagini; (4) calcolo delle nuvole di punti tridimensionali per ogni gruppo di multi-immagini sottoposte al processo di matching; (5) tracking di superfici nelle sequenze di multi-immagini; (6) determinazione del campo vettoriale tridimensionale di traiettorie (posizione, velocità, accelerazione); (7) tracking di punti chiave nel campo vettoriale di traiettorie. Nel caso della misurazione di superfici, sono richiesti solamente i primi quattro passaggi.

La misurazione accurata di parti del corpo umano inizia con un’adeguata acquisizione di dati. Nel caso di misurazioni statiche di superfici vengono utilizzate delle multi-immagini (immagini multiple che descrivono la stessa scena da posizioni differenti), mentre nel caso di misurazioni dinamiche o di tracking di superfici vengono utilizzate delle sequenze di multi-immagini (multi-immagini acquisite durante un intervallo di tempo). Per l’acquisizione di multi-immagini vari sistemi possono essere utilizzati con differenti livelli di qualità a dipendenza delle telecamere impiegate.

I processi di orientamento e di calibrazione hanno lo scopo di determinare la posizione e l’orientamento dei sensori, stabilire i parametri che descrivono la geometria interna delle telecamere e ottenere i parametri necessari per la modellizzazione delle distorsioni.
generate dai sistemi ottici. La determinazione precisa di tutti i parametri è necessaria per ottenere un’accurata misurazione.

L’obiettivo del processo automatico di matching consiste nella determinazione di punti corrispondenti nelle multi-immagini nella parte di superficie interessata. Il processo automatico di matching utilizza uno stereo-matcher costruito con tecniche ai minimi quadrati. La strategia sviluppata per il processo automatico di matching consiste nella definizione di alcuni punti base, dai quali viene determinato automaticamente un denso gruppo di punti corrispondenti coprendo l’intera parte interessata. Le coordinate tridimensionali dei punti corrispondenti sono successivamente calcolate tramite intersezione dei raggi utilizzando i risultati del processo di calibrazione. La strategia è stata sviluppata interamente considerando le caratteristiche della superficie del corpo umano ed è stata implementata in modo da ridurre al minimo il tempo di elaborazione. L’idea base del processo di tracking consiste nel seguire punti corrispondenti nelle multi-immagini durante la sequenza completa e nel calcolare poi le relative traiettorie tridimensionali. Velocità e accelerazione possono pure venire calcolate per ogni passo temporale. Il processo è costruito su tecniche di matching ai minimi quadrati. Queste vengono utilizzate per la determinazione delle corrispondenze spaziali fra immagini acquisite contemporaneamente da diversi punti di vista come pure per la determinazione di corrispondenze temporali fra immagini consecutive.

Il processo proposto può essere utilizzato per il tracking di punti ben definiti sul corpo umano. Le traiettorie tridimensionali di singoli punti non sono sufficienti per comprendere e descrivere il movimento di una persona o i cambiamenti della superficie di una parte del corpo umano. Per questo motivo, il processo di tracking viene esteso per seguire simultaneamente un denso gruppo di punti appartenenti a una superficie comune e il risultato può venire considerato come un campo vettoriale di traiettorie tridimensionali.

Per risolvere problemi causati da occlusioni, mancanza di texture, perdita di punti e apparizioni di nuovi punti, vengono introdotti i punti chiave. I punti chiave sono regioni tridimensionali di dimensioni variabili definite nel campo vettoriale di traiettorie e la loro posizione è definita dal centro di gravità. Essi vengono scelti e posizionati interattivamente in una interfaccia grafica ed il processo di tracking avviene in maniera semplice: la posizione nel prossimo passo temporale è calcolata come il valore medio dello spostamento di tutte le traiettorie contenute nella regione tridimensionale. Interfaccie grafiche sono state sviluppate per tutti i processi, compresi l’acquisizione delle multi-immagini, la calibrazione e l’orientamento, il processo automatico di matching e la visualizzazione dei risultati come nuvola di punti tridimensionali e/o campo vettoriale di traiettorie tridimensionali.

Per dimostrare le funzionalità multiple del metodo proposto, sono presentate tre differenti applicazioni: misurazione ad alta precisione del viso umano tramite cinque telecamere CCD, misurazione di un calco arteriale tramite tre telecamere CCD e registrazione del movimento del corpo umano senza l’utilizzo di punti marcanti, tramite due o tre telecamere CCD sincronizzate.
Contents

Abstract v
Riassunto vii

1 Introduction 1
1.1 Surface measurement of human body parts 1
1.2 Human face modeling 2
1.3 Full body modeling 3
1.4 Motion capture 5
1.5 Considerations 6
1.6 Contents 7

2 Data acquisition 9
2.1 CCD cameras, video signal and frame grabber 9
  2.1.1 CCD cameras and video signal 9
  2.1.2 Frame grabbers 11
2.2 Multi image acquisition systems 12
  2.2.1 Synchronized machine vision progressive scan CCD cameras 13
  2.2.2 Synchronized machine vision interlaced CCD cameras 13
  2.2.3 Digital video camcorders 14
  2.2.4 Multiple digital still cameras 16
  2.2.5 Single moving digital still camera or digital video camcorder 16
  2.2.6 Digital web cameras 17
  2.2.7 Considerations 18

3 System calibration 19
3.1 Object space-to-image space mathematical model 19
3.2 Calibration methods 22
3.3 Bundle calibration 24
  3.3.1 Spatial resection 24
CONTENTS

4 Matching process 27
  4.1 Stereo matcher 27
    4.1.1 Least squares matching 27
    4.1.2 Geometrical constraints 29
    4.1.3 Software implementation 31
    4.1.4 Quality evaluation 35
    4.1.5 Options 36
  4.2 Automatic matching process 36
    4.2.1 Seed point definition 36
    4.2.2 Matching strategy 40
    4.2.3 Options 44
  4.3 Filtering 50
    4.3.1 Regularization of the grid 50
    4.3.2 Neighborhood filtering 51

5 Surface measurement 55
  5.1 3–D point cloud 55
    5.1.1 Forward ray intersection 55
    5.1.2 Filtering 57
  5.2 Visualisation and modeling 58
    5.2.1 3–D point cloud visualisation 58
    5.2.2 Modeling 59
  5.3 Considerations 61
  5.4 Application 1: Human face modeling 61
    5.4.1 System setup 62
    5.4.2 Surface measurement 63
    5.4.3 Modeling and photo realistic visualisation 64
    5.4.4 Measurement without artificial texture projection 66
  5.5 Application 2: Measurement of blood vessel branching casting 66
    5.5.1 System setup and calibration 67
    5.5.2 Measurement and modeling 69

6 Validation 71
  6.1 Setup 71
  6.2 Surface measurement 73
  6.3 Surface comparison 75
    6.3.1 Alignment 76
    6.3.2 Distance reference data to measured data 76
  6.4 Comparison results 77
    6.4.1 Distance error 78
    6.4.2 Comparison of surfaces 80
  6.5 Considerations 80
This work pertains to the surface measurement and surface tracking of human body parts.
In the past and even presently, the research and commercial applications were concentrated either on the measurement and modeling of the human face and body or on the capture of the movement of the whole body and facial expressions.
In this chapter, the existing techniques for the surface measurement of human body parts, for human face modeling, for full body modeling and for motion capture are described. At the end, the methods proposed in this work are described shortly.

1.1 SURFACE MEASUREMENT OF HUMAN BODY PARTS

In recent years, the measurement of the surface of the human body has gained importance in medical applications. The relevant disciplines are orthopedics (Commean et al., 1994; Ono, 1995; Hackenberg et al., 2000), orthodontics (Motegi et al., 1996; Höflinger, 1996), physiology (Youmei, 1994; Tukuisis et al., 2001), plastic surgery (Mao et al., 2000; Okada, 2001), dermatology (Nebel, 2000; Frankowski et al., 2001) and biomechanics (Ronsky et al., 1999; Savatier et al., 2001). The common characteristic of the existing techniques applied in medical applications is the required high accuracy.

To measure the surface of human body parts, four different optical methods can be used: laser scanning, silhouette extraction, structured light and photogrammetric approaches. Laser scanning (Brunsman et al., 1997; Okada, 2001; Tukuisis et al., 2001) and structured light methods (Maas, 1992; Youmei, 1994; Wolf, 1996; Frankowski et al., 2000; Hackenberg et al., 2000; Sitnik and Kujawnska, 2000) are widely used to measure objects in a macroscopic scale and are also applied for the measurement of the human body surface in a microscopic scale (Frankowski et al., 2001). They have common acquisition and processing characteristics. The precision of the measurement and the simplicity of use and the wide range of software packages available for the processing, editing and modeling, have made them the most widely used systems for surface measurement. However, depending on the size of the surface to measure, the acquisition time can range from seconds to half minute, during which the subject must remain stationary. This fact makes it difficult to obtain stable measurement results, because a human always moves slightly during the data acquisition (Marshall and Gilby, 2001). Moreover, these techniques cannot be used to measure dynamically moving surfaces.
1 INTRODUCTION

In the case of silhouette extraction based methods (Zheng, 1994; Matsumoto et al., 1999; Savatier et al., 2001), multiple cameras, a single moving camera or a single camera combined with a rotating platform are used to acquire a set of images around the object. The images are processed to extract silhouettes. Intersecting the surfaces generated by silhouettes and projection centers of each image, a coarse 3–D model of the imaged body part can be determined. The detail of the model increases with the number of acquired images. However, the accuracy of this method is limited. Systems using multiple cameras can acquire images simultaneously, allowing the recording of dynamic events (e.g., containing movements).

Photogrammetric techniques (Frobin and Hierholzer, 1983; Banda et al., 1992; Gaebel et al., 1992; Mitchell, 1992; Minakuchi et al., 1995; Boersma, 2000; Nebel et al., 2001; D’Apuzzo, 2002c) offer instead accurate surface measurement, even of dynamic processes (Dorffner, 1996; Maas, 1997), by the simultaneous acquisition of images from different directions. A thorough calibration of the camera system and an accurate establishment of image correspondences are required to achieve high accuracy.

1.2 HUMAN FACE MODELING

Modeling and measurements of the human face have wide applications including medical purposes (Banda et al., 1992; Gaebel and Kacoschke, 1996; Koch et al., 1996; Motegi et al., 1996; Thomas and Newton, 1996; D’Apuzzo, 1998; Okada, 2001), computer animation (Pighin et al., 1998; Blanz and Vetter, 1999; Lee and Magnenat-Thalmann, 2000; Liu et al., 2000; Marschner et al., 2000; Sitnik and Kujawinska, 2000; Ju and Siebert, 2001b), video surveillance (C.N.N., 2001), lip reading systems (Minakuchi et al., 1995), video teleconferencing and virtual reality (De Carlo et al., 1998; Borghese and Ferrari, 2000; Fua, 2000; Shan et al., 2001). The issues that must be considered to model the face of a real person are: how realistic and accurate the obtained shape is, how long it takes to get a result, how simple the equipment is and how much the equipment costs.

The different approaches to enable the reconstruction of a human face can be classified depending on the requirements. For animation, virtual reality and teleconferencing, the photorealistic aspect is essential. In contrast, high accuracy is required for medical applications. Two different groups of methods can also be distinguished based on their data source: the first using range digitizers and the second using only images. To date, the most popular measurement technique is laser scanning (Motegi et al., 1996; Hasegawa et al., 1999; Marschner et al., 2000; Okada, 2001); for example, the head scanner of Cyberware™ (Cyberware, 2003). However, these scanners are expensive and the data are usually noisy, requiring manual editing. Another solution is offered by the structured light range digitizers (Proesmans and Van Gool, 1996; Wolf, 1996; Sitnik and Kujawinska, 2000) which are usually composed of a stripe projector and one or more CCD cameras. These can be used for face reconstruction with relatively inexpensive equipment as compared to laser scanners. The accuracy of both systems is satisfactory for static objects; however, the acquisition time required for a face scan can range from seconds to half a minute depending on the used system. Thus, a person must remain stationary during the measurement. Not only does this place a burden on the subject, but it is also difficult to obtain stable measurement results. In fact, even when the acquisition time is short, the person moves slightly
1.3 FULL BODY MODELING

involuntarily.

A different approach to face modeling uses images as source data. Various image-based techniques have been developed. They can be distinguished by the type of source data used, e.g., a single image, two orthogonal images, a set of images, video sequences, multiple images acquired simultaneously.

Parametric face modeling techniques (Blanz and Vetter, 1999) start from a single image to generate a complete 3-D model of the face. Exploiting the statistics of a large data set of 3-D face scans, the face model is built by applying pattern classification methods. The results are impressively realistic, however the reconstructed shape is an approximation of the real face and can not be used for accurate measurements.

Computer face models can also be created from two orthogonal views (Ip and Yin, 1996). Manual intervention is required for the modeling process by selecting feature points in the images. It is a simplified method to produce realistic models of human faces. However, also in this case, the shape obtained represent an approximation of the real face. To solve this problem, some solutions (Lee and Magnenat-Thalmann, 2000) work in combination with range data acquired by laser scanners.

Another image-based method consists of automatically extracting the contour of the head from a set of images acquired around the person (Zheng, 1994; Matsumoto et al., 1999). The data are then combined to form a volumetric model of the head. The set of images can be acquired by moving a single camera around the head or having the camera fixed and the face turning. The systems are fast and completely automatic, however the accuracy of the methods is low.

Video sequences based methods (Fua and Miccio, 1997; Pighin et al., 1998; Fua, 1999, 2000; Liu et al., 2000; Shan et al., 2001) use photogrammetric techniques to recover 3-D data from the images acquired by video cameras. A generic 3-D face model is then deformed to fit the recovered data, which is usually noisy. These techniques are fully automatic but may perform poorly on faces with unusual features or other significant deviations from the normal.

High accuracy measurement of real human faces can be achieved by photogrammetric solutions which combine a thorough calibration process with the use of synchronized CCD cameras to acquire simultaneously multi-images (Gruen and Baltsavias, 1989; Banda et al., 1992; Minakuchi et al., 1995; D’Apuzzo, 1998; Borghese and Ferrari, 2000; D’Apuzzo, 2002b). To increase the reliability and robustness of the results some techniques use the projection of an artificial texture on the face (Banda et al., 1992; D’Apuzzo, 2002b). The high accuracy potential of this approach can however result in a time expensive processing.

1.3 FULL BODY MODELING

In the last few years, the demand for 3-D models of human bodies has dramatically increased. The applications include medicine, biometry and manufacturing of objects to be fitted to a specific person or group of persons. However, currently the fields where 3-D models of humans are mostly used are virtual reality, the movie industry, pure animation and computer games (Anil and Gabor, 2001). This can be explained by the high costs of the required hardware and software available to date.

Two issues have also to be considered: the portability of the equipment, its simplicity of use and the achieved accuracy and resolution of the obtained 3-D models. For
virtual reality, the movie industry, pure animation and computer game applications (Balder et al., 1999; Badler, 2000; Siebert and Ju, 2000; Ju and Siebert, 2001a), where the shape of the human body is first defined and then animated, only an approximate measurement is required. On the other hand, a precise measurement of the body surface is required in medical applications (Bhatia et al., 1994) and manufacturing applications, for example, in the space and aircraft industry for the design of seats and suits (McKenna, 1999) or more generally in the clothes or car industries (Jones et al., 1993; Certain and Stuetzle, 1999; Bradtmiller and Gross, 1999). For these purposes, methods have been developed to extract biometric information from range data (Dekker et al., 1999). Recently, anthropometric databases have been defined (Paquet and Rioux, 1999; Robinette and Daanen, 1999); besides the shape information, they contain also other records of the person, which can be used for commercial or research purposes.

The currently used approaches for building such models are laser scanning, structured light methods, silhouette extraction based methods and photogrammetry. Laser scanners are quite standard in human body modeling, because of their simplicity of use, the acquired expertise (Brunsman et al., 1997) and the related market of modeling software (Burnsides, 1997). Structured light methods are well known and used for industrial measurement to capture the shape of parts of objects with high accuracy (Wolf, 1996) and have already been applied to build full body scanners (Bhatia et al., 1994; Horiguchi, 1998; Siebert and Marshall, 2000). The acquisition time of both laser scanning and structured light systems ranges from a couple of seconds to half minute. In the case of human body modeling, this can pose problems with accuracy caused by the need for a person to remain immobile for several seconds (Daanen et al., 1997; Marshall et al., 2001).

The silhouette extraction based methods can also be employed for full body measurements (Narayanan et al., 1998; Delamarre and Faugeras, 1999; Ertaud et al., 1999; Hilton and Gentils, 1999). Multiple cameras or a single moving camera are used to acquire a set of images around a person. A coarse 3-D model is determined by intersecting the surfaces generated by silhouette of the person and projection center of each acquired image. The system based on multiple cameras can also measure dynamic events (e.g., a person performing some movement); however, high accuracy cannot be achieved. By combining the extracted silhouettes with stereo data (Vedula et al., 1998), more robust results can be obtained.

Photogrammetric solutions have already been used successfully for the measurement of human body parts (D’Apuzzo, 2002c) and can be extended to the measurement of the entire body surface. Multiple synchronized cameras acquire images of a person simultaneously from different directions. The photogrammetric method can measure the surface of the human body very precisely if the quality of the acquired image is sufficient. In this case, the unconscious movements of the subject do not affect the measurement because of the simultaneous acquisition of all the images; moreover, it is not requested to stay immobile and (voluntary) movements can also be performed. The main advantages of this method over the others are the high accuracy potential, the possibility to measure moving persons and the lower cost of the required equipment. The disadvantage is the thorough processing. Combined solutions fit complex human body models to data extracted photogrammetrically from video sequences (Fua et al., 1998; Plänkers, 2001a).
1.4 MOTION CAPTURE

Motion capture systems are mainly used for two applications: in computer animation to increase the level of realism digitizing the desired movements performed by an actor and in biomechanics to precisely measure the movement of joints. The motion capture systems can be divided into three major groups: magnetic, optical and mechanical systems. Different characteristics can be taken into account to classify them, e.g., accuracy, processing time, method used, costs and portability of the system.

The magnetic systems (e.g., Ascension™, Polhemus™) use electromagnetic sensors connected to a computer unit which can process the data and produce 3–D data in real time. The major advantage of these systems is the direct access to the 3–D data without complex processing. For this reason they are very popular in the animation community. Wireless systems have also been developed to solve the disadvantage of restricted freedom of movement caused by the cabling.

Optical systems (e.g., Motion Analysis™, Vicon™, Qualisys™) are mostly based on photogrammetric methods where the trajectories of signalized target points on the body are measured very accurately (Tsuruoka et al., 1995; Boulic et al., 1997, 1998). They offer complete freedom of movement and interaction of different actors is also possible. In the last years, several improvements have been introduced, such as the use of smart cameras and CMOS sensors to achieve real-time 3–D data acquisition (Richards, 1999; Kadaba and Stine, 2000).

Electro-Mechanical systems (e.g., Analogus™) have recently appeared in the market: in this systems the person moving has to wear a special suit with integrated electro-mechanical sensors that register the motion of the different articulations. This method also has the advantage of real-time data transfer from the sensors to the computer with minimal processing. Moreover, it is less expensive.

Motorized video theodolites in combination with a digital video camera have also been used for human motion analysis (Anai and Chikatsu, 1999, 2000).

Motion capture can also be achieved by image-based methods. They can be divided into single camera and multiple cameras systems. Single camera systems use sequences of images acquired by a single camera. To gain three-dimensional information from single video clips, knowledge of human motion must be used. Some systems learn from provided sample training data and apply statistical methods to get the performed 3–D motion (Mahoney, 2000; Song et al., 2000; Rosales and Sclaroff, 2000). Other systems perform the tracking of defined human body models with constraints by sophisticated filtering processes (Cham and Rehg, 1999; Segawa and Totsuka, 1999; Deutscher et al., 2000).

Multiple camera systems use sequences of images acquired simultaneously by two or more cameras. Some systems assume very simple 3–D human models (e.g., articulated objects made of cylinders) whose characteristic sizes and joint angles are determined by comparing the projections of the model into the different images with the extracted silhouettes of the moving person (Delamarre and Faugeras, 1999; Cheung et al., 2000) or the extracted edges (Kinzell and Behring, 1995; Gravila and Davis, 1996). Other systems use image based tracking algorithms to track three-dimensionally the surface of the human body (D’Apuzzo, 2000) or different body parts (Ohno and Yamamoto, 1999). Mathematical models of human motion can also be used to track directly in 3–D data, which can be trajectories of known key points (Iwai et al., 1999) or dense disparity maps (Jojic et al., 1999). More sophisticated methods fit generic human
1 INTRODUCTION

body models to extracted 3–D data from video sequences (Fua et al., 2000; Herda et al., 2000; Plänkers, 2001b; Fua et al., 2002).

1.5 CONSIDERATIONS

The four fields described above correspond with the recent research and commercial applications for surface and motion measurement regarding the human body. In this work, a method is presented to treat both the surface measurement and the surface tracking aspects in a unique process. The proposed method can be applied to measure either the surface of a static human body part or the moving surface of a dynamic event. The entire process is composed of seven steps (four, in case of surface measurement):

1. calibration of the system: establishment of the internal and external orientation of the cameras and the parameters modeling the lens distortion;
2. acquisition of multiple images and/or multiple image sequences;
3. matching process: establishment of correspondences in the multiple images (at least for the first set);
4. surface measurement: computation of the 3–D point cloud for each matched multiple image set;
5. surface tracking in the multiple image sequences;
6. establishment of a 3–D vectorfield of trajectories (position, velocity, acceleration);
7. tracking key-points in the vectorfield of trajectories.

To compare the proposed method for surface measurement (first four steps) with other systems, five aspects can be considered: the accuracy of the measurement, the time required to acquire the data, the ease of use, the total processing time and the hardware costs. Figure 1.1 shows the five aspects for three surface measurement methods: multi image photogrammetry (the proposed method), structured light digitizer and laser scanning.

The accuracy of the surface measurement system is one of the most important and significant aspect. The greatest accuracy potential is held by structured light digitizers, followed by multi-image photogrammetry and laser scanner. However, when measuring human body parts, the effective accuracy of the measurement depends strongly on the required acquisition time. In fact, even if stabilized and immobilized, humans move unconsciously (e.g., breathing, muscle contractions). Therefore, only a short acquisition time can assure the accuracy potential of the used system. Increasing the time required for data acquisition will affect negatively the accuracy of the measurement. About this aspect, the proposed method is certainly more adequate than structured light digitizers or laser scanners, since multi-images can be acquired simultaneously by using synchronized cameras. Moreover, the simultaneous acquisition of the images allows the measurement of moving surfaces and even the recording of dynamic events (surface tracking).

Two negative aspects of the proposed method compared to the others are the total processing time and the ease of use. In fact, multi image photogrammetry implies
Fig. 1.1 Benchmarking of the three surface measurement techniques: multi image photogrammetry, structured light digitizer, laser scanning. Accuracy of the measurement, acquisition time, ease of use, processing time and hardware costs are considered. The farther from the center of the graph, the better it is (note the different directions of the axes).

different processing steps (system calibration, matching, computation of 3-D coordinates) whereas the other systems deliver the results quite fast. Similar is the aspect about the easiness of use: multi image photogrammetry requires the setting up of the camera system, its calibration and the image acquisition whereas for, e.g., laser scanners, the scanning device has simply to be placed and the measurement process can begin. These aspects are however less relevant.

The last considered aspect is the cost of the hardware which is in case of multi image photogrammetry limited to the cameras, lenses and frame grabber. A large variety of hardware exists with large costs differences depending of the quality, performance and characteristics. However, structured light digitizers and laser scanners always require more expensive hardware.

Considering simultaneously all aspects, the method proposed in this work result more adequate for the measurement of human body parts.

1.6 CONTENTS

The system setup and the procedure of acquiring the data are described in Chapter 2. The calibration process is presented in Chapter 3. Chapter 4 explains the matching process. Chapter 5 describes how the surface of the human body is computed. Two applications are presented: human face modeling and the measurement of a blood vessel branching. Chapter 6 presents the validation of the proposed method for surface measurement. The process of tracking (single points, surface and key-points) is explained in Chapter 7. At the end of the chapter a method where full body motion capture can be achieved using the process is presented. A short application of tracking facial expressions using image sequences acquired by a single video camera is also presented to show the flexibility of the proposed method. Chapter 8 gives the conclusions of this work. The Appendix A describes the implemented graphical user interface for the developed software and Appendix B a newly implemented acquisition system. Appendix C gives some examples of parameter sets for the surface matching and tracking processes.
Chapter 2

Data Acquisition

This chapter presents the background of CCD sensors, video signal and frame grabbers, followed by the description of the different camera systems that can be used for the acquisition of multi-images and/or multi-image sequences.

2.1 CCD CAMERAS, VIDEO SIGNAL AND FRAME GRABBERS

2.1.1 CCD Cameras and Video Signal

Cameras used in the field of digital image processing have two major components: the image acquisition unit and the image output unit. Currently, the acquisition unit is based dominantly on CCD (Charge Coupled Device) chips, which consist of separated light sensitive elements, each of which represents one pixel. A CCD pixel transforms light into a charge, which during readout is transformed into a voltage. CCD chips are small in size and weight, have high dynamic and high linearity. Moreover, due to mass production for the consumer market, they are relatively inexpensive. These reasons made them the most popular acquisition units. For machine vision applications, the sensor chips require additional features such as sensor flatness and regularity of the pixel position.

One of the basic parameters of CCD chip is the number of effective pixels or pixel elements. A typical CCD chip used in video cameras, has 768 pixels in the horizontal direction (columns) and 576 pixels in the vertical direction (rows). If larger format of the images is required, more expensive CCD arrays having more pixels may be used. The output unit of a camera generates a video signal which is suitable for succeeding image processing devices. In standard monochrome cameras, the acquired image is transformed into a video signal conforming to one of the internationally accepted standards. For Europe this is CCIR (Comité Consultatif International des Radiocommunications) and for the United States it is RS-170 which was defined by the EIA (Electronics Industries Association). For color cameras, the additional standards (PAL for Europe and NTSC for the United states) are modified version of the original monochrome standards.

These standards are based on the constraints of traditional tube cameras and screens and seem antiquated in view of modern CCD chips and flat screen displays. Nevertheless, they are still used in most current camera devices. Therefore, it is interesting to describe how they were originally generated in traditional devices. In the tube cameras, the beam of electrons has to be controlled to scan the light sensitive area of
the pick-up tube; the same applies for the beam which generates the image on tube screens. The scan starts in the top left corner (see figure 2.1 top); encountering the end of the line (after 52 ms for CCIR) the deactivated ray runs back (retrace, 12 ms for CCIR) to the beginning of the third (not the second) line. In this manner, the first scan covers the field of odd lines while the second scan works on the field of even lines. This method (called interlacing) diminishes the flicker of monitors but leads to problems in image processing. A complete scan (called frame) consists of 625 lines and lasts 1/25th of a second for CCIR. During the retrace time, the horizontal sync pulse (hsync) is added to the video signal to indicate the beginning of the next line. Figure 2.1 shows how a standard video signal is built.

The vertical sync pulse (vsync) triggers the beginning of a new field. The vsync is a sequence of several pulses spread over 50 lines for CCIR. Therefore these lines are not usable for the standard video signal, which leads to 575 visible lines for CCIR. The synchronization of standard video signal transmission is exclusively based on the horizontal and vertical sync pulses (hsync, vsync).

Most machine vision video cameras can be synchronized externally so that the hsyncs and vsyncs of all of the cameras can be put in sync with each other. The technical term for this process is external synchronization or genlock. The simplest method is to build a synchronization chain in which the video output from one camera is used as the synchronization input of the next one. However, when the synchronization must be very accurate, an extra sync generator should be used; this option is usually offered by most machine vision frame grabbers.

Another remand from the video standards is the interlace technique which is a curse for metrology. The camera first scans the odd lines of the image and secondly the even lines. If a moving object is recorded, it would be in different positions between the first and second scans, resulting in the even and the odd lines being horizontally disarranged, causing a saw pattern effect. The simplest solution to this problem is to use only the odd lines, however the resolution in vertical direction would be halved. If the full resolution is required, special non-interlaced cameras have to be used, called progressive scan cameras, which are able to acquire an image in one pass. Obviously they are more expensive and require special frame grabbers.
2.1 CCD CAMERAS, VIDEO SIGNAL AND FRAME GRABBER

2.1.2 Frame Grabbers

Frame grabbers read the analog video signal of a camera and transform it into a digital image sequence. Figure 2.2 shows the basic structure of a frame grabber. First, the horizontal and vertical synchronization pulses are separated from the incoming video signal (sync separation), indicating the beginning of new lines and the beginning of new fields. This procedure can take some time (usually a couple of frames), it can however be avoided by synchronizing externally the video sources so that the hsyncs and vsyncs of all of the cameras are in synchronization with the frame grabber (this operation is called genlock). Once stable synchronization of the lines and frames has been achieved, the next part of the process of acquiring an image concerns the generation of the pixels themselves. In accordance to the video standards, the sample and hold unit digitizes 767 pixels per line (for CCIR) collecting them in an image buffer which stores at least one complete frame and is used if the bandwidth of the connection to the PC (bus) is too small to transport without loss the digitized video data stream in the main memory of the computer.

Different types of frame grabbers exist (see figure 2.3). For mobile applications, frame grabbers are available on PCMCIA cards (e.g., DFG/VPP from ImagingSource) or Video-to-FireWire converters (e.g., DFG/1394-1 from ImagingSource). The quality of the digitized images is not as high as the machine vision frame grabbers but they can be used for demonstration purposes. Low-cost machine vision frame grabbers (e.g., Matrox Meteor-II) with on board between-storage are able to acquire image sequences with full frame rate (25 Hz for CCIR) and have usually multiple composite video inputs. Frame grabbers with three independent RGB inputs can be generally used to acquire up to three monochrome video signals at the full frame rate. Frame grabbers with multiple (usually two or three) RGB inputs (e.g., Matrox Meteor-II Multi Channel) can be used to acquire images synchronously from multiple color cameras or multiple monochrome standard CCD cameras (up to six in case of two multiple RGB inputs and up to nine for three multiple RGB inputs). These frame grabbers are however more expensive (mid-cost class).

Special frame grabbers (e.g., Matrox Meteor-II/Camera Link/1394/Digital) are required for higher resolution, for higher frame rate, for use with progressive scan cameras or for use with digital cameras. The hardware costs are substantially higher.
2 DATA ACQUISITION

Fig. 2.3 Different types of frame grabbers: (a) PCMCIA frame grabber (DFG/VPP) with two composite video inputs; (b) Video-to-IEEE-1394 converter (DFG/1394-1) with two composite video inputs; (c) Low-cost machine vision frame grabber (Matrox Meteor-II) with 4 composite video inputs; (d) Multi-channel frame grabber (Matrox Meteor-II Multi Channel with two RGB inputs.

2.2 MULTI IMAGE ACQUISITION SYSTEMS

In this work, the term *multi-image* refers to multiple images acquired from different positions in the space describing the same scene and *multi-image sequence* refers to multi-images acquired during a time interval.

In this work are treated only cameras with standard video format (24 bit color or 8 bit black and white, 756x578 or 640x480 pixels) and frame rate (25 or 30 Hz). Other cameras with special characteristics (e.g., high speed, high resolution, high sensitivity) are not considered here because of the elevated costs.

To record a scene with movement, the multiple images have to be acquired simultaneously. The precision of the synchronization of the multiple imaging devices plays an essential role for the accuracy potential of the measurement achieved using the images.

On the other hand, to record static scenes, the multi-images can be acquired at different times without a loss of accuracy. However, for applications involving recording people, the human body cannot be considered as a static object, because a person always moves slightly unconsciously due to, for example, breathing or muscle contraction. Therefore, for surface measurement of human body parts, it is always recommended to precisely synchronize the multiple cameras.

Various methods can be used to acquire multi-image sequences. Although different camera systems have similar resolution and quantization, different levels of quality can be achieved depending on the system. They are listed below in order of decreasing accuracy potential:

- synchronized machine vision progressive scan CCD cameras,
- synchronized machine vision interlaced CCD cameras,
- digital video camcorders, synchronized using a clapper.

For surface measurement applications without tracking, only multi-images are required. In this case, also the following systems can be used for the acquisition (listed with decreasing accuracy potential):

- multiple digital still cameras,
- a single digital still camera or video camcorder, acquiring in different position,
- multiple digital web cameras.

In the following sections, the characteristics of the listed systems are described in more detail.
2.2 MULTI IMAGE ACQUISITION SYSTEMS

2.2.1 Synchronized Machine Vision Progressive Scan CCD Cameras

In term of quality and performance, the best imaging tool for the acquisition of multi-image video sequences are synchronized progressive scan CCD cameras. They differ from the more common and less expensive interlaced CCD cameras (described in section 2.2.2) in the way that a full frame is acquired simultaneously offering full resolution of the sensor also with moving objects. The cameras are more expensive and special frame grabbers are required.

![Progressive scan CCD camera multi-image acquisition system. Left: setup of the system, the cameras are synchronized together and connected to a frame grabber which digitizes the images and transfers them to a PC. Right: progressive scan CCD camera (Sony XC-55)(bottom) equipped with machine vision c-mount lens (top).](image)

Figure 2.4 shows the setup of an acquisition system built up with progressive scan CCD cameras. The precise electronic synchronization guarantees the simultaneous acquisition of multi-images. Depending on the number of cameras used, one or more frame grabbers are required to digitize the multi-image sequences. They are stored in the memory of a PC first and then moved to the disks.

Recently developed progressive scan cameras (e.g., Dragonfly of Point Grey Research) use the IEEE-1394 port. No frame grabber is in this case required. See the Appendix B for the description of a multi-image acquisition system based on such cameras.

2.2.2 Synchronized Machine Vision Interlaced CCD Cameras

Machine vision interlaced CCD cameras offer a less expensive but still high quality solution. The main advantage of this type of camera over the progressive scan camera is the standard video output (CCIR/EIA). In this case, low-cost frame grabbers can be used. They are usually provided with separate RGB inputs that allow the connection of three synchronized CCD cameras. This results in the least expensive high quality multi-camera acquisition system. If more than three cameras are required, multi-channel frame grabbers or multiple frame grabbers have to be used.

The setup of a multi-image acquisition system based on machine vision CCD cameras (figure 2.5) is the same as for progressive scan CCD cameras (section 2.2.1). The cameras are synchronized together and connected to a frame grabber. The image sequences are first stored in the memory of a PC and then moved to the disks.
A disadvantage of using this type of CCD cameras is the interlace effect which is caused by the fact that odd and even lines are acquired in different times. This results in saw pattern effects when movement is present, as shown in figure 2.6. A solution for solving this problem is to use only the odd lines, duplicating them in the images and deleting the even lines. The result is shown on the right of figure 2.6, the saw pattern disappears at the cost of the resolution which is halved in the vertical direction.

**Fig. 2.5** Multi-image machine vision CCD camera acquisition system. Left: setup of the system, the cameras are synchronized together and connected to a frame grabber which digitizes the images and transfers them to a PC. Right: interlaced machine vision CCD camera (Sony XC-77 with 35mm lens).

**Fig. 2.6** Interlace effect. Left: the hand is moving to the right, the interlace effect can be seen as a saw pattern. Right: result after resolution reduction: the even lines are removed and the odd lines are duplicated.

### 2.2.3 Digital Video Camcorders

Multiple digital video camcorders can also be used for the acquisition of multi-image video sequences. This is a less expensive solution because no frame grabber is required. The digital video camcorders (for example the Sony digital HandyCam DCR-VX700E, figure 2.7 right) store the image sequence digitally on mini DV tapes, in DV format with a size of 720x576 pixels and 24 bit color resolution. The DV format is a Sony proprietary digital video and audio recording standard. A single image in this
2.2 MULTI IMAGE ACQUISITION SYSTEMS

compressed format has a size of about 140kB. The image sequences stored on the mini DV tape can be transferred without loss of quality to a PC by FireWire connection and converted into common image formats.

Figure 2.7 shows the setup of a multi-image acquisition system based on multiple digital video camcorders. No frame grabber is required. The multiple cameras can be synchronized using a *clapper* (figure 2.7 left) or something generating a "clap" sound. A "clap" sound has a clear defined start peak which can be recognized in the audio signal of the different video sequences (see figure 2.8). The sequences are then manually synchronized by aligning the audio signals. A maximum error of half frame (corresponding to 1/50 of a second, for CCIR) can be expected. Depending on the applications (e.g., slow movement), this may be acceptable or not.

Fig. 2.7 Multi-image acquisition system based on multiple digital video camcorders (Sony DCR-VX700E). Left: setup of the system, a *clapper* is used for synchronization purposes.

Fig. 2.8 Synchronization with the audio signal. Right: audio signal of two acquired sequences, the beginning of the "clap" sound is clearly defined. Left: the two sequences are synchronized aligning the audio signals, the maximum error is half a frame (0.04 seconds for CCIR).
2.2.4 Multiple Digital Still Cameras

Digital still cameras decrease in price from year to year. An interesting feature of some of them is the possibility to control the camera from a PC, allowing the acquisition of images from multiple cameras in short times. The acquired images are stored on memory cards and can be transferred to a PC afterward. For example, figure 2.9 (right) shows a Kodak DC215 which can acquire images with 1152x864 or 640x480 pixels. The use of larger image formats is not treated in this work.

![Multiple digital still camera acquisition system. Left: setup of the system. Right: a digital still camera (Kodak DC215), bottom: back view, LCD monitor.](image)

With this acquisition system, it is obviously not possible to acquire multi-image sequences but only multi-images. Moreover, there is a short delay (less than 1 second) between the acquisition of the different images. Therefore, the recording of movement is not possible and only static surface measurement processes can be achieved using the images acquired with this system. Since humans move slightly unconsciously during the acquisition, the accuracy potential of this system decreases with the increment of the time delay between the acquisition from the different cameras.

2.2.5 Single Moving Digital Still Camera or Digital Video Camcorder

A less expensive solution is offered by using a single digital still camera or a single video camcorder and acquiring the images by moving the camera at different positions (see figure 2.10). Although the images can be acquired relatively fast (e.g., less than ten seconds for five multi-images of a face), this method can be used only for the measurement of immobile or fixed human body parts.

A second disadvantage of this method is the requirement of more complex orientation and calibration procedures. To utilize the method implemented for this work and described in section 3.3, reference points with known 3-D coordinates have to be imaged together with the human body part (reference points on figure 2.10).
2.2 MULTI IMAGE ACQUISITION SYSTEMS

Fig. 2.10 A digital camcorder or a digital still camera is moved around the human body part to acquire multi-images. Note the reference points required by the implemented method for the establishment of the external orientation of the camera at different positions.

2.2.6 Digital Web Cameras

The least expensive solution for the acquisition of multi-images is offered by FireWire web cameras. Their cost is in fact today under 100 US$ per camera and no frame grabber is required. The digital web cameras are in fact connected to the PC by the IEEE-1394 port (also called FireWire or i-link). The advantage of this type of connection is that only a single input port on the PC is required; the multiple cameras can be connected serially to each other (see figure 2.11).

Fig. 2.11 Multi-image acquisition system using web cameras. Setup of the system: the FireWire digital camera (Unibrain Fire-i) (right) are connected together serially through the IEEE-1394 cable (right bottom), only one camera has to be connected to a (portable) PC.

For example, the Unibrain Fire-i camera (see figure 2.11, right) has digital output and several formats (color uncompressed, color compressed and grayscale uncompressed), several resolutions (640x480, 320x240 and 160x120 pixels) and several frame rates (7.5, 16 and 25 Hz) can be chosen. The serial connection allows the acquisition of
multi-images even with a portable computer (see figure 2.12); PCMCIA cards are readily available, moreover some portable computers (e.g., Sony VAIO and Apple i-Book) are provided with an integrated FireWire input.

![Three FireWire digital cameras (Unibrain Fire-i) connected simultaneously to a portable computer.](image)

Although the image quality is low and the noise is high, digital web cameras are a convenient and inexpensive solution for demonstration purposes (figure 2.12 shows three web cameras connected to a laptop).

### 2.2.7 Considerations

The simultaneity of the acquisition of the multi-images is essential for an accurate measurement of moving human body parts. A precise synchronization of the multiple acquisition units is available with only the two described machine vision acquisition systems (sections 2.2.1, 2.2.2). Systems based on multiple digital still cameras (section 2.2.4) and systems using digital web cameras (section 2.2.6) can be set up to acquire multi-images simultaneously. However, the cameras cannot be electronically synchronized with each other; the maximum time delay which can occur between the acquisition of the different images is therefore half a frame (1/50 of a second for CCIR). The same synchronization accuracy can be achieved with systems using multiple video camcorders (section 2.2.3), aligning the audio signals of the different sequences. Systems based on a single camera (section 2.2.5) can evidently not simultaneously acquire multi-images. In some cases, e.g., when high accuracy of the measurements is not required or if the human body part can be immobilized, the longer time delay can however be acceptable.
Chapter 3

System Calibration

As system calibration is defined the simultaneous calibration and orientation of all the components involved in the acquisition system. Camera calibration refers to the determination of the parameters describing the internal geometry of the individual imaging devices and other parameters modeling the systematic errors caused by the optical system and other sources. Camera orientation includes the determination of the parameters of exterior orientation to define the camera station and camera axis in the 3-D space. A thorough determination of all the parameters is required for an accurate measurement.

In the next sections, the mathematical model for the projection of the object space onto the digital image coordinate system will be described first. The different methods used in this work to calibrate the imaging systems are then presented and finally the implemented single camera bundle calibration model is described.

3.1 OBJECT SPACE-TO-IMAGE SPACE MATHEMATICAL MODEL

In this section the mathematical model for the projection of a point in the object space onto the focal plane of an imaging device, as shown in figure 3.1, is described along with the transformation into image coordinates (image space).

The object space-to-focal plane projection can be expressed by the collinearity condition, i.e., a point in the object space, its projection on a plane and the projection center have to lie on a straight line (dotted line in the figure 3.1). Mathematically it can be described by the following equation:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \lambda \mathbf{R} \begin{bmatrix}
x' - x_p \\
y' - y_p \\
-c
\end{bmatrix} + \begin{bmatrix}
X_0 \\
Y_0 \\
Z_0
\end{bmatrix}
\] (3.1)

where
- \(X, Y, Z\) object space coordinates of point P,
- \(x', y'\) coordinates of point \(P'\) on the focal plane,
- \(x_p, y_p\) coordinates of the principal point \(H'\) on the focal plane,
- \(c\) camera constant,
- \(\mathbf{R}\) rotation matrix between sensor and object space coordinate systems,
- \(X_0, Y_0, Z_0\) object space coordinates of projection center \(O'\),
- \(r_{xx}\) coefficients of rotation matrix \(\mathbf{R}\),
- \(\lambda\) scale factor for each imaging ray.
In equation 3.1 the interior orientation is defined by the principal point $x_p, y_p$ and the camera constant $c$ while the exterior orientation is defined by the object coordinate of the projection center $O'X_0, Y_0, Z_0$ and the rotation matrix $R(\omega, \varphi, \kappa)$ describing the rotation from the object coordinate system $Oxyz$ to the sensor coordinate system $O'x'y'z'$ where $\omega, \varphi, \kappa$ are the Euler angles around the $x, y, z$ axes:

$$
R = \begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33}
\end{bmatrix}
$$

$$
= \begin{bmatrix}
\cos \omega \cos \kappa & -\cos \varphi \sin \kappa & \sin \varphi \\
\cos \omega \sin \kappa + \sin \omega \sin \varphi \cos \kappa & \cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa & -\sin \omega \cos \varphi \\
\sin \omega \sin \kappa - \cos \omega \sin \varphi \cos \kappa & \sin \omega \cos \kappa + \cos \omega \sin \varphi \sin \kappa & \cos \omega \cos \varphi
\end{bmatrix}.
$$

The three components in equation 3.1 are reduced to two by canceling out the scale factor $\lambda$, resulting in the collinearity equations:

$$
dx' = x_p - c \cdot \frac{r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0)}{r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)}
$$

$$
dy' = y_p - c \cdot \frac{r_{12} \cdot (X - X_0) + r_{22} \cdot (Y - Y_0) + r_{32} \cdot (Z - Z_0)}{r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)}
$$

(3.3)
3.1 OBJECT SPACE-TO-IMAGE SPACE MATHEMATICAL MODEL

To model the distortions occurring in the focal plane, a simplified version of the parameter set introduced by Brown (1971) is used. Two additional parameters modeling the shearing and the differential scaling in x are introduced in the model as described by Beyer (1992). Thus the coordinates on the focal plane are:

\[
\begin{align*}
    x' &= \tilde{x}' + dx' = \tilde{x}' - sc \cdot \tilde{x}' + sh \cdot \tilde{y}' + dx' \\
    y' &= \tilde{y}' + dy' = \tilde{y}' + sh \cdot \tilde{x}' + dy'
\end{align*}
\] (3.4)

where

\[
\begin{align*}
    x', y' & \quad \text{coordinates in focal plane [mm], with distortion,} \\
    \tilde{x}', \tilde{y}' & \quad \text{coordinates in focal plane [mm], without distortion, according to eq. 3.3,} \\
    dx', dy' & \quad \text{distortion terms,} \\
    d\tilde{x}', d\tilde{y}' & \quad \text{symmetric radial and decentering lens distortion terms,} \\
    sc & \quad \text{scale factor in x,} \\
    sh & \quad \text{shear factor;}
\end{align*}
\]

The last equation for the description of the mathematical model is the transformation between metric coordinate system on the focal plane and pixel image coordinate system (see figure 3.2). This can be described as an affine transformation (equation 3.6).

\[
\begin{align*}
    x' &= \left( u - \frac{n_x}{2} \right) \cdot \mu_x \\
    y' &= -\left( v - \frac{n_y}{2} \right) \cdot \mu_y
\end{align*}
\] (3.6)

where

\[
\begin{align*}
    u, v & \quad \text{coordinate in the image [pixel],} \\
    n_x, n_y & \quad \text{image size [pixel],} \\
    \mu_x, \mu_y & \quad \text{pixel size [mm].}
\end{align*}
\]

The equation 3.6 can be inverted to obtain the transformation metric to pixel coordinate system:

\[
\begin{align*}
    u &= \frac{x'}{\mu_x} + \frac{n_x}{2} \\
    v &= -\frac{y'}{\mu_y} + \frac{n_y}{2}
\end{align*}
\] (3.7)
3 SYSTEM CALIBRATION

3.2 CALIBRATION METHODS

To orient and calibrate camera systems, various methods can be used. However, two characteristics of multiple camera systems have relevant importance for choosing adequate and appropriate calibration and orientation procedures: (a) the multiple cameras have usually either a fix position or they are displaced all together without changing their relative positions; (b) the multiple cameras have to be calibrated and oriented very often (e.g., at every acquisition session) because of the need of small adjustments (e.g., focus and iris, and even position and direction). For these reasons, a simultaneous calibration and orientation (system calibration) of the multiple cameras is more appropriate. The two problems can be solved jointly with a spatial point array of unknown coordinates plus additional scales or with a suitable array of control points. Both methods were applied in this work.

In the first case, the spatial point array was achieved by moving through the object space a reference bar with two retroreflective target points during the acquisition of the multi-image sequence (reference bar method). The required additional scales are provided by the known distance between the two points on the bar. The image coordinates of the two target points are automatically measured and tracked along the sequence with a least squares matching based process (see figure 3.3). As described by Maas (1998), the multiple camera system can then be calibrated by self-calibrating bundle adjustment (Gruen and Beyer, 2001) with the additional information of the known distance between the two points on the bar at every recorded locations. The main advantage of the method is the only requirement of the reference bar, the disadvantage of the method is a thorough processing of the data to calibrate the system.

The second case can be solved using a 3-D reference field with signalized points whose coordinates in space are known. The calibration procedure is in this case simpler and full automation can be achieved using coded target points (see figure 3.4 and 3.5 left), that can fully automatically be recognized and measured in the images (Niederoest, 1996).
3.2 CALIBRATION METHODS

Fig. 3.3  Automatically tracked and measured image coordinates of the two points on the reference bar.

Fig. 3.4  3–D calibration frames with coded targets.

Fig. 3.5  Left: coded target #100. Right: template images used for the measurement of the targets with LSM.

The result of the calibration process are the exterior orientation parameters of the cameras (position $X_0, Y_0, Z_0$ and rotations $\omega, \varphi, \kappa$), the interior orientation parameters of the cameras (camera constant $c$ and principle point $x_p, y_p$), parameters for the radial and decentering distortion of the lenses and optic systems ($k_1, k_2, k_3, p_1, p_2$) and two additional parameters modeling effects as differential scaling and shearing ($sc, sh$). A thorough determination of all these parameters is required to achieve high accuracy in the measurement. Four parameters defining the sensor characteristics (sensor size $n_x, n_y$ and pixel size $\mu_x, \mu_y$) have to be known in advance.

The implemented software for the calibration process (see section A.2.2) is based on
the bundle calibration method using a single image of a reference field. The next section will briefly describe the mathematical procedure for calibrating the cameras using this method. For the details regarding the self-calibrating bundle adjustment method used for the calibration with the reference bar method, the reader is referred to the literature (Maas, 1998).

3.3 BUNDLE CALIBRATION

3.3.1 Spatial Resection

Assumed is a set of known control points object space coordinates \(X_i, Y_i, Z_i\), their measured image coordinates \(x'_i, y'_i\) and approximations for the interior and exterior orientation. The resulting set of collinearity equations (equations 3.3 and 3.4) can be considered as observation equations relating the observation \(x'_i, y'_i\) to the parameters \(x_0, y_0, z_0, \omega, \varphi, \kappa, c, x_p, y_p, sc, sh, k_1, k_2, k_3, p_1, p_2\), leading to the set of observation equations:

\[
x'_i = f_x(x_0, y_0, z_0, \omega, \varphi, \kappa, c, x_p, y_p, sc, sh, k_1, k_2, k_3, p_1, p_2, X_i, Y_i, Z_i)
y'_i = f_y(x_0, y_0, z_0, \omega, \varphi, \kappa, c, x_p, y_p, sc, sh, k_1, k_2, k_3, p_1, p_2, X_i, Y_i, Z_i)
\]  

(3.8)

After linearization of the equations 3.8 and the introduction of a true error vector \(e\), the system of equations becomes:

\[
l - e = Ax
\]  

(3.9)

with the following unknown vector \(x\), design matrix \(A\) and observation vector \(l\):

\[
x = (dx_0, dy_0, d\omega, d\varphi, dk, dc, dx_p, dy_p, ds, ds, dh, dk_1, dk_2, dk_3, dp_1, dp_2)^T
\]

\[
A = \begin{bmatrix}
\left(\frac{\partial f_x}{\partial x_0}\right)_1 & \left(\frac{\partial f_x}{\partial y_0}\right)_1 & \left(\frac{\partial f_x}{\partial \omega}\right)_1 & \left(\frac{\partial f_x}{\partial \varphi}\right)_1 & \ldots & \left(\frac{\partial f_x}{\partial p_2}\right)_1 \\
\left(\frac{\partial f_y}{\partial x_0}\right)_1 & \left(\frac{\partial f_y}{\partial y_0}\right)_1 & \left(\frac{\partial f_y}{\partial \omega}\right)_1 & \left(\frac{\partial f_y}{\partial \varphi}\right)_1 & \ldots & \left(\frac{\partial f_y}{\partial p_2}\right)_1 \\
\left(\frac{\partial f_x}{\partial x_0}\right)_2 & \left(\frac{\partial f_x}{\partial y_0}\right)_2 & \left(\frac{\partial f_x}{\partial \omega}\right)_2 & \left(\frac{\partial f_x}{\partial \varphi}\right)_2 & \ldots & \left(\frac{\partial f_x}{\partial p_2}\right)_2 \\
\left(\frac{\partial f_y}{\partial x_0}\right)_2 & \left(\frac{\partial f_y}{\partial y_0}\right)_2 & \left(\frac{\partial f_y}{\partial \omega}\right)_2 & \left(\frac{\partial f_y}{\partial \varphi}\right)_2 & \ldots & \left(\frac{\partial f_y}{\partial p_2}\right)_2 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\left(\frac{\partial f_x}{\partial x_0}\right)_n & \left(\frac{\partial f_x}{\partial y_0}\right)_n & \left(\frac{\partial f_x}{\partial \omega}\right)_n & \left(\frac{\partial f_x}{\partial \varphi}\right)_n & \ldots & \left(\frac{\partial f_x}{\partial p_2}\right)_n \\
\left(\frac{\partial f_y}{\partial x_0}\right)_n & \left(\frac{\partial f_y}{\partial y_0}\right)_n & \left(\frac{\partial f_y}{\partial \omega}\right)_n & \left(\frac{\partial f_y}{\partial \varphi}\right)_n & \ldots & \left(\frac{\partial f_y}{\partial p_2}\right)_n 
\end{bmatrix}
\]

\[
l = (x'_1 - \tilde{x}'_1, y'_1 - \tilde{y}'_1, x'_2 - \tilde{x}'_2, y'_2 - \tilde{y}'_2, \ldots, x'_n - \tilde{x}'_n, y'_n - \tilde{y}'_n)^T
\]  

(3.10)
3.3 BUNDLE CALIBRATION

where

\[ dx_0, dy_0, \ldots \] changes of the unknown parameters from the given approximations,

\[ n \] number of control points,

\[ x'_i, y'_i \] observed (i.e., measured) image coordinates of the control point \( i \),

\[ \hat{x}_i', \hat{y}_i' \] the estimated image coordinates of the control point \( i \), i.e., the control point, is backprojected onto the image according to equations 3.3 and 3.4, using the estimated parameters.

For the estimation of \( \hat{x} \) the Gauss-Markov model of least squares is used and leads to:

\[
\begin{align*}
\hat{x} &= \left( A^T PA \right)^{-1} A^T P l \\
\nu &= A \hat{x} - l \\
\tilde{\sigma}_0^2 &= \frac{\nu^T P \nu}{2n - u} \\
Q &= \tilde{\sigma}_0^2 \cdot \left( A^T PA \right)^{-1} \\
\sigma_i^2 &= Q_{ii} \text{ variance of the single unknowns;}
\end{align*}
\]

where

\[ u = 16 \quad \text{the number of unknowns,} \]

\[ P \quad \text{the weight coefficient matrix.} \]

The weight matrix \( P \) has usually diagonal form. Adding an identity part of the unknowns \( dc, dx_p, dy_p, dsc, dsh, dk_1, dk_2, dp_1, dp_2 \) to the equation system 3.10, single unknown parameters can be excluded from the adjustment process by assigning a very large value of the respective weight. The new design matrix \( \tilde{A} \) and observation vector \( l \) become:

\[
\begin{align*}
\tilde{A} &= \begin{bmatrix}
\tilde{A} & 0 \\
1 & \ddots \\
0 & 1
\end{bmatrix} \\
l &= \left( \tilde{l}, dc', dx'_p, dy'_p, dsc', dsh', dk_1', dk_2', dp_1', dp_2' \right)^T
\end{align*}
\]

where

\[ \tilde{A} \quad \text{design matrix according to equation 3.10,} \]

\[ \tilde{l} \quad \text{observation vector according to equation 3.10,} \]

\[ dc', dx'_p, \ldots \] observed changes of the unknown parameters.

The partial derivatives of \( x_0, y_0, z_0, \omega, \varphi, \kappa, c \) are computed numerically in the iteration process according to equation 3.13 (the derivatives for the other parameters are obtained in an analogue form):

\[
\left( \frac{\partial f_s}{\partial x_0} \right)_i = \frac{x'_i - x'_i}{dm}
\]
3 SYSTEM CALIBRATION

where

\[ x^i \] observed (i.e., measured) image coordinates of the control point \( i \),

\[ x'd_i \] backprojected control point \( i \) by \( x_0 = x_0 + dm \).

\[ dm \] increment.

The other derivatives are determined algebraically as follows:

\[
\begin{align*}
\frac{\partial f_x}{\partial x_p} & = 1 - sc & \frac{\partial f_y}{\partial x_p} & = sh \\
\frac{\partial f_x}{\partial y_p} & = s_h & \frac{\partial f_y}{\partial y_p} & = 1 \\
\frac{\partial f_x}{\partial s_c} & = -x'_i & \frac{\partial f_y}{\partial s_c} & = 0 \\
\frac{\partial f_x}{\partial y_i} & = y'_i & \frac{\partial f_y}{\partial y_i} & = x'_i \\
\frac{\partial f_x}{\partial \theta_1} & = x'_i \cdot r_i^2 & \frac{\partial f_y}{\partial \theta_1} & = y'_i \cdot r_i^2 \\
\frac{\partial f_x}{\partial \theta_2} & = x'_i \cdot r_i^4 & \frac{\partial f_y}{\partial \theta_2} & = y'_i \cdot r_i^4 \\
\frac{\partial f_x}{\partial \theta_3} & = x'_i \cdot r_i^6 & \frac{\partial f_y}{\partial \theta_3} & = y'_i \cdot r_i^6 \\
\frac{\partial f_x}{\partial \theta_4} & = 2x'_i^2 + r_i^2 & \frac{\partial f_y}{\partial \theta_4} & = 2x'_i \cdot y'_i \\
\frac{\partial f_x}{\partial \theta_5} & = 2x'_i \cdot y'_i & \frac{\partial f_y}{\partial \theta_5} & = 2y'_i^2 + r_i^2
\end{align*}
\]

(3.14)

where

\[ r_i = \sqrt{x'_i^2 + y'_i^2} \cdot \]

The system is solved iteratively computing the estimated unknown vector \( \hat{x} \) according to equation 3.11, actualizing the design matrix \( A \) and the observation vector \( l \) and computing again the estimated unknown vector, until the changes of all the unknown parameters are smaller than a threshold.
This chapter describes the matching process in all its components. First, in section 4.1 is described the stereo matcher which determines the position of corresponding points in an image pair. The automatic multi-image matching process based on the stereo matcher is presented and described in detail in section 4.2; the result is a dense set of corresponding points in the multi-images covering the entire interested surface parts. Finally, section 4.3 describes the optional filtering processes that can be applied to the matching results.

4.1 STEREO MATCHER

The stereo matcher is based on the adaptive least squares method (Gruen, 1985) with the additional geometrical constraint of forcing the matched point to lie on the epipolar line. The following sections describe the mathematical models for the least squares matching process and the geometrical constraint, software implementation and the evaluation of the quality of the results.

4.1.1 Least Squares Matching

The least squares estimation model is described briefly here; for a complete detailed description the reader is referred to the reference (Gruen, 1985). Assumed are two image windows (called image patches) given as discrete functions $f(u,v), g(u,v)$, wherein $f$ is the template, $g$ is the search image window and $u,v$ are the image coordinates. Since the two image windows are not identical, the comparison of them results in:

$$f(u,v) - e(u,v) = g(u,v)$$

(4.1)

wherein $e(u,v)$ is the true error vector. The idea of least squares matching (LSM) is as follows: equation 4.1 is treated as a nonlinear observation equation which models the vector of observation $f(u,v)$ with a function $g(u,v)$, whose location in the search image needs to be estimated. In order to account for the different viewing angles, image shaping parameters in form of an affine transformation (equation 4.2) must also be estimated. Additionally, a radiometric correction factor is introduced to correct different lighting conditions.
The affine transformation is applied with respect to an initial position \( u_0, v_0 \):

\[
\begin{align*}
u &= a_0 + a_1 \cdot u_0 + a_2 \cdot v_0 \\
v &= b_0 + b_1 \cdot u_0 + b_2 \cdot v_0.
\end{align*}
\] (4.2)

After linearization of the function \( g(u, v) \), equation 4.1 becomes:

\[
f(u, v) - e(u, v) = g(u_0, v_0) + \frac{\partial g(u_0, v_0)}{\partial u} \cdot du + \frac{\partial g(u_0, v_0)}{\partial v} \cdot dv + rs. \tag{4.3}
\]

With the simplified notation:

\[
g_u = \frac{\partial g(u_0, v_0)}{\partial u}, \quad g_v = \frac{\partial g(u_0, v_0)}{\partial v}
\]

and adding the differentiation of equation 4.2, equation 4.3 results in:

\[
f(u, v) - e(u, v) = g(u_0, v_0) + g_u da_0 + g_v da_1 + g_u dv_0 da_2 +
+ g_v db_0 + g_v u_0 db_1 + g_v v_0 db_2 + rs. \tag{4.4}
\]

Combining the parameters of equation 4.4 in the parameter vector \( x \)

\[
x^T = (da_0, da_1, db_0, db_1, db_2, rs) \tag{4.5}
\]

and discretizing the equation 4.3, i.e., having a number of \( n \) pixels in the image windows, the coefficients in the design matrix \( A \) and the vector difference \( l \) become:

\[
\begin{align*}A &= \begin{bmatrix}
g_u & 1_g & 1_u & 1_{gu} & 1_v & 1_{gv} & 1_{gu} & 1_{gv} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
n_g u & n_g u & n_u & n_{gu} & n_v & n_{gv} & n_{gu} & n_{gv} & n_v
\end{bmatrix},
\end{align*}
\]

\[
l^T = (1_f - 1_g, \ldots, n_f - n_g)
\] (4.6)

where

\( i_u, i_v \) local coordinates of the pixel \( i \) in the (not transformed) image window,

\( i_{g_u}, i_{g_v} \) discretized horizontal and vertical gradients at the pixel \( i \),

\( i_f, i_g \) grey values of the pixel \( i \) in the template and in the search image window.

For the search image, the value of \( i_g \) is bilinearly resampled at the affinely transformed position (equation 4.2) according to:

\[
g(x, y) = \left( sgn \cdot g(x, y_1) + \frac{y - y_1}{y_2 - y_1} \right) \cdot \left( g(x, y_2) - g(x, y_1) \right) \tag{4.7}
\]

where

\( x, y \) position to be resampled,

\( x_1, y_1, x_2, y_2 \) the support points for resampling,

\( g(x, y_1), g(x, y_2) \) the linear (to \( x \)) resampled grey values, from equation 4.8,

\( sgn \) \( sign (g(x, y_2) - g(x, y_1)) \).
Linear resampling in x is computed as:

\[
g(x) = \left( sgn \cdot g(x_1) + \frac{x - x_1}{x_2 - x_1} \right) \cdot (g(x_2) - g(x_1)) \tag{4.8}
\]

where

\[
x \quad \text{position to be resampled},
\]

\[
x_1, x_2 \quad \text{the support points for resampling},
\]

\[
sgn \quad \text{sgn} \left( g(x_2) - g(x_1) \right).
\]

Introducing the weight coefficient matrix \( P \) as:

\[
P = \begin{bmatrix}
p_{da_0} & 0 \\
p_{da_1} & p_{da_2} \\
p_{db_0} & p_{db_1} \\
p_{db_2} & p_{rs}
\end{bmatrix}, \tag{4.9}
\]

the Gauss-Markov least squares estimation model leads to:

\[
\hat{x} = (A^T P A)^{-1} A^T P l \quad \text{solution vector},
\]

\[
v = A \hat{x} - l \quad \text{residual vector},
\]

\[
\hat{\sigma}_0^2 = \frac{1}{n-7} v^T P v \quad \text{variance factor}, \tag{4.10}
\]

\[
Q = \hat{\sigma}_0^2 \cdot (A^T P A)^{-1} \quad \text{covariance matrix},
\]

\[
\hat{\sigma}_i^2 = Q_{ii} \quad \text{variance factor of the single parameters}.
\]

For the usual cases, the weight coefficient matrix \( P \) is set as identity.

### 4.1.2 Geometrical Constraints

To increase the robustness and accuracy of the matching process, geometrical constraints can be introduced. When multi-images are acquired simultaneously, they display the same object space viewed from different directions. Figure 4.1 shows graphically the case of an image pair. Given a point \( P = (x_1, y_1) \) on the image plane 1, the corresponding point \( P' \) on the image plane 2 must lie on the epipolar line. This can be determined as the intersection of the image plane 2 with the plane formed by \( O_1, \hat{O}_1, \hat{P}, \hat{B} \), called epipolar plane, where \( O_1, O_2 \) are the projection centers of the two imaging devices and \( \hat{B} \) is the basis vector between them. Therefore, the three vectors \( \hat{O}_1 \hat{P}, \hat{B}, \hat{O}_2 \hat{P} \) have to lie on the same plane; their coplanarity condition can be mathematically expressed as:

\[
\left( \hat{O}_1 \hat{P} \times \hat{O}_2 \hat{P} \right) \cdot \hat{B} = 0. \tag{4.11}
\]

After the required transformations, equation 4.11 results in the following equation expressed in the coordinate system of image plane 2:

\[
y_2 = k \cdot x_2 + h \tag{4.12}
\]
Fig. 4.1 Construction of the epipolar line from an image pair.

with:

\[ k = \frac{zm_2 \cdot by_2 - ym_2 \cdot bz_2}{zm_2 \cdot bx_2 - xm_2 \cdot bz_2} \]
\[ h = \left( \frac{bx_2 \cdot (zm_2 \cdot yp_2 - ym_2 \cdot c_2)}{zm_2 \cdot bx_2 - xm_2 \cdot bz_2} \right) + \left( \frac{by_2 \cdot (xm_2 \cdot c_2 - zm_2 \cdot xp_2)}{zm_2 \cdot bx_2 - xm_2 \cdot bz_2} \right) \]  
\[ + \left( \frac{bz_2 \cdot (ym_2 \cdot xp_2 - zm_2 \cdot yp_2)}{zm_2 \cdot bx_2 - zm_2 \cdot bz_2} \right) \]  

(4.13)

where:

\[ \overline{B_2'} = \begin{bmatrix} bx_2 \\ by_2 \\ bz_2 \end{bmatrix} = R_2 \cdot \begin{bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{bmatrix} \]

\[ \overline{O_1 P_2'} = \begin{bmatrix} xm_2 \\ ym_2 \\ zm_2 \end{bmatrix} = R_2 \cdot \overline{O_1 P_0'} = R_2 \cdot R_1^T \cdot \overline{O_1 P_1'} = \]  

(4.14)

and:

- \( c_2 \): camera constant for image 2,
- \( xp_2, yp_2 \): principal point for image 2,
- \( \overline{B_2'} \): the basis vector expressed in the coordinates system \( O_2 \),
- \( \overline{O_1 P_i'} \): the vector \( \overline{O_1 P} \) expressed in the three coordinate systems \( O, O_1, O_2 \),
- \( R_1 \): rotation matrix of coordinate system \( O_1 \) from \( O \),
- \( R_2 \): rotation matrix of coordinate system \( O_2 \) from \( O \).
Different methods can be used to integrate the epipolar geometrical constraint in the least squares matching process. Since for this work, the processing time has to be limited, the geometrical constraint is chosen to be expressed as a single additional observation:

\[ da_0 - c_e \cdot db_0 = 0, \]  

(4.15)
describing the fact that the position in the search image must move along the epipolar line, i.e., with a defined slope \( c_e \), called here *epipolar constant*. The next section describes how the value of \( c_e \) is determined from the epipolar line of equation 4.12. The new design matrix \( A \) and the vector difference \( l \) become:

\[
A = \begin{bmatrix}
1 \ g_u & 1 \ g_v \cdot 1 \ u & 1 \ g_v \cdot 1 \ v & 1 \ g_v & 1 \ g_v \cdot 1 \ v & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 \ g_u & 1 \ g_v \cdot 1 \ u & 1 \ g_v \cdot 1 \ v & -c_e & 0 & 0 & 0 & 1
\end{bmatrix}
\]  

(4.16)

\[
l^T = (1 \ f - 1 \ g, \ldots, n \ f - n \ g, 0) .
\]

A line in weight coefficient matrix \( P \) must also be introduced

\[
P = \begin{bmatrix}
p_{d0} & \cdots & p_{rs} \\
\vdots & \ddots & \vdots \\
& & p_{ce}
\end{bmatrix}
\]  

(4.17)

The value of \( p_e \) can be chosen to increase or decrease the effect of the epipolar constraint to the solution.

### 4.1.3 Software Implementation

The stereo matcher has to be flexible, since the matching process has to be used for different purposes, e.g., matching corresponding points in multi-images, signalized point measurements, tracking processes. For this reason, it is implemented with and without epipolar constraint.

The following inputs are required to begin the process: the image patch size, the position in the template image and an approximative position in the search image \((u_0, v_0)\) of equation 4.2. Then, the process continues iteratively computing the estimation of the changes of the affine transformation parameters \( da_0, da_1, db_0, db_1, db_2 \) and the radiometric correction factor \( rs \) according to equation 4.10, updating the design matrix \( A \) and the observation vector \( l \) according to equations 4.6 and 4.16 and estimating again the solution vector \( \hat{x} \), until the change of the variance factor \((\frac{1}{\sigma_0^2} \) of equation 4.10) is smaller than a preset threshold (usually 2%).

The given approximative position in the search image must be sufficiently close to the correct position to allow a successful convergence of the process. Figure 4.2 shows an example: if the starting location is too distant from the correct point, the process cannot converge correctly.
It is possible that the matching process may converge to false results producing mismatches. Therefore, it is recommended to define indicators for the quality of the matching result for evaluation purposes. This is discussed more in detail in section 4.1.4.

In case of given orientation and calibration information, the geometrical constraint can be used to increase the robustness of the matching process. In this case, some operations have to be performed before starting the least squares process. The epipolar constrained LSM is implemented in this work such that the location of the point in the search image moves with a defined slope (epipolar constant of equation 4.15), assuming the point is lying on the epipolar line. Therefore, it is required that the starting position is on the epipolar line before executing the LSM process. Moreover, the epipolar line has to be corrected for the distortion caused by the lens. In fact, the epipolar line is curved in the image instead of completely straight. This effect has to be corrected before applying the implemented LSM. To accelerate the matching process it is suitable to determine the curvature of the epipolar line only around the interested point and not in the entire image. For this reason, only a short segment of the epipolar line will be computed in the image coordinate system; this will be called the epipolar segment. A detailed description of the process to determine it will follow. Practically, the complete matching procedure is composed of four steps: (1) compute the epipolar line in the metric coordinate system in the search image, (2) compute the epipolar segment in the image coordinate system around the selection, (3) project the selected point on the epipolar segment and (4) apply the geometrically constrained LSM at that position.

The flowchart of figure 4.3 gives the overview of the four steps necessary to start the epipolar constrained LSM.

**Step 1.** Computation of the epipolar line. First, the point selected in the template image \((u_t, v_t)\) has to be transformed from the image coordinate system to the metric coordinate system according to equation 3.6. Then, the obtained metric template image coordinates \(x_t', y_t'\) have to be corrected from the lens distortion. A direct inversion of the equation 3.4 is not possible; the problem has to be solved iteratively subtracting from the image coordinates \(x_t', y_t'\) (rather than adding) the distortion \(\Delta x', \Delta y'\).
4.1 STEREO MATCHER

Fig. 4.3 Flowchart: start the LSM process with epipolar constraint.

The epipolar line, i.e., the \( k \) and \( h \) values, in the search image is then determined according to equations 4.13 and 4.14.

**Step 2.** Computation of the segment of the epipolar line around the selection in the search image. The selected point in the search image \((u_s, v_s)\) is first transformed in the metric coordinate system \((x_s', y_s')\) according to equation 3.6, the lens distortion is then corrected according to the algorithm 4.18. The epipolar line is then intersected with a window around the selected point \((x_s', y_s')\), resulting in two points defining the epipolar segment \((x_{e1}', y_{e1}'), (x_{e2}', y_{e2}')\). The distortion caused by the lens according to equations 3.4 and 3.5 has to be added and the two points are then transformed into the image coordinate system \((u_{e1}, v_{e1}), (u_{e2}, v_{e2})\) according to equation 3.7.

computed at the corrected position; algorithmically:

\[
\begin{align*}
x_{\text{correct}}' &:= x'_t \\
y_{\text{correct}}' &:= y'_t \\
\text{repeat} & \\
\deltax', \deltay' &:= f(x_{\text{correct}}', y_{\text{correct}}') \quad // \text{according to eqs. 3.4 and 3.5} \\
\text{end.}
\end{align*}
\]
Step 3. The selected point is projected onto the epipolar segment. It is required since the implemented geometrically constrained least squares matching algorithm assumes the selected point to be on the epipolar line. The selected point \((u_s, v_s)\) is orthogonally projected on the epipolar segment as:

\[
\begin{align*}
    u_{es} &= \frac{b - d}{c - a} \\
    v_{es} &= a \cdot u_{es} + b
\end{align*}
\]

where
\[
\begin{align*}
    a &= \frac{v_{e2} - v_{e1}}{u_{e2} - u_{e1}} \\
    b &= v_{e1} - a \cdot u_{e1} \\
    c &= \frac{v_{e1} - v_{e2}}{u_{e2} - u_{e1}} \\
    d &= v_{e1} - a \cdot u_{e2}.
\end{align*}
\] (4.19)

Step 4. The geometrically constrained LSM is applied at the position \(u_{es}, v_{es}\) using as value for the epipolar constant the slope of the epipolar segment as:

\[
    c_e = \frac{u_{e2} - u_{e1}}{v_{e2} - v_{e1}}.
\] (4.20)

Figure 4.4 shows an example of the geometrical constrained LSM process; the template image with the selected template point (black cross) and the image patch (black box) is displayed on the left; in the center, is shown the search image with the epipolar line drawn in white and the given approximative selection (black cross) and starting image patch (black box); on the right, is shown the result of the geometrically constrained LSM, i.e., the affinely transformed image patch drawn as black box and the epipolar segment drawn in white. For convenience, the size of the window for the intersection with the epipolar line is equal to the image patch size.

Fig. 4.4  LSM process with epipolar constraint. Left and center: in black the selection in the template and search images, in white the epipolar line. Right: result in the search image after LSM, in black the affinely transformed image patch, in white the epipolar segment.

The stereo matching process is implemented only for an image pair, i.e., an image is used as template and the other as search image. In the case of multi-images, one image is used as template and the others as search images. The stereo matcher is then applied to each search image independently using the same template image patch (see figure 4.5). A complete multi–image matching process taking in account all the images in a single step (Gruen and Baltsavias, 1988; Baltsavias, 1991) could be implemented. This would produce more robust and accurate results (e.g., mismatches as presented in figure 4.6 would not be possible). In the section regarding future works (see section 8.2.2) it is indeed suggested to define and implement a fast geometrical constrained multi–image matching process. Anyhow, the results achieved by the simpler implemented multi–image matching process were enough robust and accurate. Moreover,
4.1 STEREO MATCHER

The processing time required by the proposed method to determine a dense set of corresponding points was within the desired limits. Therefore, no further investigation was performed to define and implement more complex and accurate matching algorithms.

**Fig. 4.5** Multi-image geometrical constrained LSM is achieved using an image as template and applying the stereo matching process for each search image independently using the same template image patch. The black boxes represent the patches selected in the template image (left) and the affine transformed in the search images, the epipolar lines are drawn in white.

4.1.4 Quality Evaluation

To evaluate the quality of the results of the matching, different indicators are used: the resulting a posteriori standard deviation of unit weight of the least squares adjustment ($\sigma_0$ from equation 4.10), the resulted standard deviation of shifts in x and y directions ($\sigma_x = \sigma_y = \sigma_{b_0}$ from equation 4.10), the displacement from the start position in x and y direction ($a_0, b_0$) and in the case of geometrically constrained LSM, the distance to the epipolar line. Thresholds for these values can be defined for different cases, according to the texture and the type of the images. Quality tests are necessary to evaluate the results of the matching process.

The advantage of using multi-images is shown in figure 4.6. Shown on the left is the template image with the template point, in the center and right, the matching process resulting in two different search images. Additionally, in each image are drawn also the two epipolar lines. The matching process in the two images converges with acceptable value for the single control checks ($\sigma_0, \sigma_x, \sigma_y, a_0, b_0$, distance to epipolar line); however, it can be observed that the matched points do not lie in the intersection of both epipolar lines. Checking the distance to all the epipolar lines is indeed a straightforward and fast way to find mismatches.

**Fig. 4.6** Mismatch of the geometrical constrained LSM. Left: template image, center and right: two search images. In the search image 1 the LSM converged to a false position. The mismatch can be recognized because the results do not lie on the intersection of the two epipolar lines.

A complete multi-image matching process taking in account all the images in a single step (Gruen and Baltsavias, 1988; Baltsavias, 1991) and producing a simultaneous solution in the multi-images would not allow the kind of mismatches shown in figure 4.6. Indeed, the implementation of this feature is proposed for the further development of the process (see section 8.2.2).
4 MATCHING PROCESS

4.1.5 Options

The options that have to be chosen for the stereo matching process are the image patch size in x and y direction, which of the affine transformation parameters are used, the use of the radiometric correction factor, the use of the epipolar constraint and the weight of the epipolar constraint in the weight matrix.

4.2 AUTOMATIC MATCHING PROCESS

The role of the automatic matching process is the establishment of a dense and robust set of corresponding points in entire regions of the images. The method described here is entirely developed regarding the mainly smooth characteristic of the surface of human body parts. Moreover, the process strategy is especially developed to minimize the required processing time. The following sections describe in detail the methodology and the functionality of the process.

4.2.1 Seed Point Definition

The automatic matching process that produces a dense and robust set of corresponding points, starts from few seed points. The location, the number and the distribution of the seed points play an essential role for the time required by the automatic matching process to cover entire regions of the images. In the ideal case of a perfectly smooth surface without discontinuities and with a good level of texture, the number of defined seed points does not affect the results neither the required processing time. A single or more seed points would not produce substantial differences in the results. However, the imaged human body part can present some discontinuities (e.g., nose, overlapping parts) and some difficult regions with low texture (e.g., eyebrows). In this case, the seed points serve to define regions that are locally smooth and continuous. These regions are treated separately, allowing a fast coverage of the entire image. The process will be clearly explained in section 4.2.2. Depending on the case, the ideal distribution of seed points has to be chosen. Figure 4.7 shows three examples of well distributed seed points.

The cues for the definition of the seed points in the region to be matched are:

- usually, the seed points should be distributed regularly throughout the region of interest;
- in case of elongated shapes (e.g., legs, arms) seed points are preferable along its long axis (e.g., points 3-6, 12-15 and 16-20 on the central image or points 10-15 and 16-20 on the right image of figure 4.7);
- seed points should be placed symmetrically on both sides of problematic areas such as very dark or very shiny zones (e.g., teeth, eyebrows) (e.g., points 4 and 5 on the left image of figure 4.7), steep surfaces (e.g., nose, under the chin) (e.g., points 13, 14 or 15, 12 on the left image of figure 4.7) or discontinuities in the surface (e.g., overlapping body parts) (e.g., points 4, 8 of the central image of figure 4.7. The problematic area will, in that way, lie at the border of the local regions defined by the seed points (see figure 4.18).

Seed points can be generated in three different modes: fully automatically, semi-automatically (complete mode: defining them only in one image or partial mode
4.2 AUTOMATIC MATCHING PROCESS

4.2.1.1 Manual Mode. This is used for special cases where the automated modes could fail. In this mode, the seed points have to be selected manually (with an approximation of few pixels) in each image. LSM is then applied to find the exact positions. Figure 4.8 shows an example: on the top without using epipolar constraint, on the bottom using epipolar constraint. In the latter case, the drawn epipolar line facilitates the user. Additionally, the user has also to decide if the quality of the matching result is sufficient. The manual mode is a time consuming procedure, but in some special cases (e.g., a fine repeated pattern in the image) it is required for the convergence of LSM to the exact position.

4.2.1.2 Semi-Automated Mode. This is a faster way to define seed points. The principle is to define the position of the seed point only in the template image and to automatically establish the corresponding points in the search images. Both partial and complete semi-automated modes are possible. In the partial version, the user has
4 MATCHING PROCESS

to select also a rough approximative location in the two search images, whereas in the *complete* version this is not required. Figures 4.9 and 4.10 illustrate the procedure, with and without epipolar constraint.

**Fig. 4.9** Semi automated seed point definition. *Complete* version: the only selection made by the user is in the template image. Top: without epipolar constraint, bottom: with epipolar constraint.

**Fig. 4.10** Semi automated seed point definition. *Partial* version: the user select also an approximative location in the two search images, the searching region is smaller. Top: without epipolar constraint, bottom: with epipolar constraint.

When calibration and orientation information is available, the corresponding point in the search images is automatically determined by searching the best matching results along the epipolar line in a restricted region (the white boxes in figures 4.9 and 4.10 bottom). For the *partial* version (figure 4.10 bottom) the search region is smaller, reducing the number of matches that have to be performed and therefore reducing
4.2 AUTOMATIC MATCHING PROCESS

consistently the required computation time. For the complete version of the semi-automatic mode (figure 4.9 bottom), the search region has to be considerably larger; for convenience it is centered around the same position as for the template image (image coordinates $i, j$ in figure 4.9).

When the calibration and orientation information cannot be used or is unavailable, a more complex strategy is used: the search area is in this case scanned along a path (dotted white curve in the top of figures 4.9 and 4.10) to find the position with the best cross correlation value; once this position is found (white cross in the top of figures 4.9 and 4.10), LSM is applied at that location to find the corresponding points.

The results are evaluated in both cases (partial and complete versions) automatically, checking the thresholds described in section 4.1.4. If the epipolar constraint is used, the distances to both epipolar lines in the search images are also checked. The correctness of the seed points is essential for the automatic matching process, as no mismatches are allowed. The quality thresholds are therefore set high.

The semi-automated mode is the most convenient one for normal cases of static surface measurement: it is in fact fast but leaves the choice of the seed points to the user.

4.2.1.3 Fully Automatic Mode. This applies the Foerstner interest operator (Foerstner and Guelch, 1987) on the template image to automatically determine locations where the matching process may perform robustly. The corresponding points in the other images are then established with the same procedure as in the complete version of the semi-automated mode.

The template image is first divided into regular regions whose size can be chosen. For each region, interest points are established with the Foerstner operator; the points with the best values are chosen as candidates. The complete version of the semi-automated seed points definition procedure is then applied to all the candidates and the matching results are accepted as seed points if the quality is sufficient.

The full automatic procedure works well in case of good texture of the human body part surface. However, some regions without seed points may remain. Figure 4.11 shows a comparison between seed points defined semi-automatically (left) and the result of the full automatic mode (right).

It is evident that the fully automatic mode generates seed points that are regularly distributed in the image without any knowledge of the human body part to be measured. On the contrary, in the semi-automated mode, the user can choose to place the seed

Fig. 4.11 Example of semi-automatically selected seed points (left) and fully automatically generated seed points (right).
points in key locations decreasing the time required by the automatic matching process. The full automatic mode is therefore useful in cases where the number of multi-image sets to be processed may be very large, e.g., for dynamic surface measurements or for tracking from multi-image video sequences. In these cases, the required manual intervention has to be minimal.

4.2.2 Matching Strategy

The matching strategy defines how the entire region of interest in the image is covered starting from the seed points. A dense set of corresponding points is the result expected at the end of the process. An additional goal of the implemented matching strategy is to minimize the entire processing time. The time required to cover the entire image can vary greatly depending on the distribution and the number of the defined seed points. A very simple method to dramatically reduce the processing time is the definition of a region of interest (ROI) in the template image. It can be given as a sequence of points or manually defined in the implemented graphical user interface (see section A.2.3). The matching process will then be limited to the ROI.

After the definition of the seed points, the template image is divided into polygonal regions by Voronoi tessellation (see figure 4.13 left). The Voronoi tessellation divides the space between individual seed points into polygonal regions such that the boundaries surrounding each seed point enclose an area that is closer to that seed point than to any other neighboring points. For the software implementation, discretization is used to increase the speed of the tessellation process. The image is divided into a regular grid and the closest seed point is found for each element. Then all of the pixels included in the element are set to belong to that seed point region. The flowchart in figure 4.12 describes the process.

Starting from the seed points, the automatic matching process produces a dense set of corresponding points in each polygonal region by sequential horizontal and vertical shifts. The figure 4.13 shows graphically the process and the flowcharts in figures 4.14 and 4.15 describe it in detail.
4.2 AUTOMATIC MATCHING PROCESS

**Fig. 4.13** Search strategy for the matching process. Left: Voronoi tessellation. Right: starting from the seed points, each region is covered by sequential horizontal and vertical shifts.

**Fig. 4.14** Flowchart of the search strategy for the matching process, for the dotted bounding boxes see figure 4.15.

The process works adaptively at each shift, changing some parameters (e.g., smaller shift, bigger size of the patch) if the quality of the matching result is not satisfactory. Several indicators are used to define the quality (see section 4.1.4): a posteriori standard deviation of the least squares adjustment, standard deviation in x and y directions, displacement from the start position in x and y direction and distance to the epipolar lines. Thresholds for these values are defined at the begin of the process according to the texture and the type of the images.
The key parameter for the matching strategy is the pixel shift. It may have different values in the x and y directions; but they are usually equal. It defines the amount of pixels to shift, in the template and search images, from the seed point and from the previously matched points. Optionally, it is possible to perform the matching process by shifting half or quarter of given pixel shift value (option shift 1/2 1/4 in figure 4.15). In this case, only the matched points at the full shift locations would be stored. This option is useful if the amount of stored data has to be limited without compromising the performance and accuracy of the matching process.

If quality of the results of the matching is not sufficient, the matching may be repeated using a bigger patch (option LSM special in figure 4.15) and/or a smaller shift value (option small shift 1/2 1/4 1/8 in figure 4.15).
4.2 AUTOMATIC MATCHING PROCESS

When either the boundaries of the polygonal region are reached or when the matching fails, the process continues from the central position to the next upper or lower shift level (see figure 4.18, left). The coverage of the entire image is achieved by repeating the process for all the polygonal seed points regions. However, it is possible that at the end of the process, holes of unanalyzed areas occur in the set of matched points (see figure 4.16).

**Fig. 4.16** Result of the automatic matching process. An area of unmatched points remains. Left: template image, right: search image. The bigger points are the seed points, the white stepped lines in the template image are the boundaries of the polygonal region determined by the Voronoi tessellation.

The causes of the presence of holes can have different origins, such as the encountering of discontinuities in the surface or lack of texture (see figure 4.18). Therefore, it is highly probable that the holes can be reduced by searching from other directions. Indeed, the process close the gaps attempts to match the missing points by searching from all directions around the holes. It is composed of three steps: (1) finding the areas where matched points are missing, (2) attempting to match missing points, (3) continuing to match until no new points are matched.

To find the area where matched points are missing (1) the image is divided into a regular grid whose size can be chosen (see figure 4.17, center).

**Fig. 4.17** Close the gaps. Left: a gap in the set of matched points. Center: the area is divided into a regular grid (dashed lines) to find the regions where points are missing. Right: shifting from matched points to match the missing points.

The number of matched points is counted for each grid element and if it is less than a set percentage the total number of matched points that can fit in the grid element (e.g., 80%), then the grid element is classified as a region with missing points. Grid elements containing no points are not considered. The size of the grid elements and the threshold percentage are parameters that may be selected by the user.

The grid elements with missing points are then processed in step (2). The matching process is applied starting from all of the matched points contained in the grid element and shifting in each direction (left, right, up, down) (see figure 4.17 right).
In the third phase of the process, step (3), a matching loop is performed starting from all the new matched points and shifting in each direction (left, right, up, down), until no new points are matched. This last phase of the process is intended to close definitively all of the gaps.

The flowchart in figure 4.19 describes the process close the gaps in detail. The shifting procedures are the same as described above and the same options can be selected (1 1/2 1/4 1/8 shifts, bigger patch, smaller shift).

The close the gaps process is a time consuming process that can strongly affect the speed of the entire matching process. The number and size of the gaps in the data has therefore to be limited. The location and number of the seed points play an essential role. In case of discontinuities (see figure 4.18), two seeds points have to be placed symmetrically to them. The boundary of the polygonal regions would then coincide with the difficult area. In this way, the size of the hole would be smaller and, therefore, the time required by the entire matching process shorter.

**Fig. 4.18** Matching process. The presence of a discontinuity in the surface produces a hole in the set of the matched points (left). The boundary of the polygonal regions will coincide with the discontinuity when the location of the seed points is symmetrical to the discontinuity (right).

### 4.2.3 Options

Different options and parameters can be chosen for the automatic matching process. As explained in section 4.2.1 different methods can be used to define the seed points. At least a single seed point is required to start the process.

A relevant option that can be chosen for the matching process is whether or not to use the epipolar constraint. The complete matching process (definition of seed points and automatic matching) can in fact be performed without orientation and calibration information if desired. This functionality is useful, for example, when the orientation is not accurate enough or unknown. In these special cases, the least squares matching algorithm is not geometrically constrained. Obviously, the robustness of the results of the process decreases. However, the quality of the set of matched points may be satisfactory.

The complete set of parameters to be set for the automatic matching process is shown in figures 4.20 and 4.21 which shows three option windows of the graphical user interface of the developed software (see section A.2.3).
4.2 AUTOMATIC MATCHING PROCESS

4.2.3.1 Preprocessing. Figure 4.20 shows the parameters required for preprocessing. On the left, for the optional Wallis filtering process (top) and for the full automatic generation of seed points (bottom); on the right for the semi-automated and manual definition of seed points. Detailed description about the parameters regarding Wallis filter can be found in (Wallis, 1976).

Fig. 4.19 Flowchart of the process close the gaps.
The required parameters for the **full automatic definition of the seed points** are the following (their names refers to figure 4.20):

- **grid size**: size of the regular grid elements for dividing the template image [pixel]. This parameter control the number of the generated seed points.
- **search size**: size of the search region in the search images [pixel], i.e., the size of the white box in figure 4.9.
- **search step**: pixel step along the epipolar line, respectively along the search path in the search region.
- **max sigma 0**: threshold of $\hat{\sigma}_0$ (see equation 4.10) to accept the matching result.
- **max sigma x,y**: thresholds of $\hat{\sigma}_x = \hat{\sigma}_{a0}$ and $\hat{\sigma}_y = \hat{\sigma}_{b0}$ (see equation 4.10) to accept the matching result.
- **epipolar constraint**: on/off.
- **max epi dist**: threshold of distance [pixel] of the matched point to the two epipolar lines to accept the matching result.

The last three parameters are required by the Foerstner interest operator. For detailed information, the reader is referred to the reference (Foerstner and Guelch, 1987). The missing LSM parameters are defined in the next parameter set.

Figure 4.20, right, shows the parameters for the **manual and semi-automated seed point definition** divided into three groups: LSM parameters, epipolar parameters and automation.

---

**Fig. 4.20** Left: preprocessing options, Wallis filtering and parameters for the full automatic definition of seed points. Right: parameters for the semi-automated and manual definition of seed points.
For the **LSM** process the following parameters have to be chosen:

- **used trafo paras:** used affine transformation parameters \((a_0, a_1, a_2, b_0, b_1, b_2)\) and radiometric correction factor \((rs)\). Usually, all the parameters are used.

- **patch size:** image patch size in pixels. The ideal value for this parameter has to be found by testing LSM manually on the images. It depends strongly on the texture in the images and on the image format.

The parameters for the **epipolar constraint** are:

- **use epipolar constraint:** use or not use the epipolar constraint.

- **weight in P:** weight \((p_e)\) for the epipolar constraint in the weight coefficient matrix \(P\) (equation 4.17), defined as a multiple of the number \(n\) of pixels inside the image patch \((n, 2n, 4n, 8n)\) or as value.

- **patch size/2:** half window size to intersect with the epipolar line to find the epipolar segment, i.e., \(\Delta x, \Delta y\) in flowchart of figure 4.3. This parameter is usually set as \(lsm\ psize/2\), i.e., half of the LSM patch size (e.g., as in figure 4.4); \(lsm\ psize\) is adequate for small LSM patch sizes (7, 9 pixels); an arbitrary value can also be given.

The last set of parameters are for the **semi-automated mode**:

- **use automation:** use or not use the semi-automated mode.

- **search:** searching mode, along the epipolar line \((along\ epi\ line)\) or inside a square region along a path \((in\ box)\).

- **search size:** size of the search region in the search images, i.e., the size of the white box in figure 4.9.

- **search step:** pixel step along the epipolar line, resp. along the search path in the search region.

- **max sigma 0:** threshold of \(\sigma_0\) (see eq. 4.10) to accept the matching result.

- **max sigma x,y:** thresholds of \(\sigma_x = \sigma_{a0}\) and \(\sigma_y = \sigma_{b0}\) (see eq. 4.10) to accept the matching result.

- **max epi dist:** threshold of distance (in pixels) of the matched point to the two epipolar lines to accept the matching result.

The values for the thresholds of \(\sigma_0\), \(\sigma_x\) and \(\sigma_y\) have to be found by testing manually LSM on the images. The same applies for the two parameters regarding the automated LSM \((search\ size,\ search\ step)\); they strongly depend on the images. No rules or advises can be given regarding these parameters. Some examples of parameter sets can be found in appendix C.

### 4.2.3.2 Automatic Matching Process

The parameters required for the automatic matching process are displayed in figure 4.21. They can be divided into four groups: search strategy, LSM parameters, epipolar parameters and parameters for the automatic matching process.
The parameters defining the search strategy are:

- **search shift**: entire, half, one quarter or one eighth of the pixel shift value which is going to be used for the matching and close-the-gaps processes (shift in flowcharts of figures 4.15 and 4.19). This strategy options can be used for reducing the amount of stored data without affecting the matching results. With, e.g., the option search shift/2 and pixel shift as 2 pixels, the matching process will match each pixel in the template image but it will store only every two pixels.

- **no search shift**: entire, half, one quarter or one eighth of the pixel shift value in the case of an unsuccessful match (small shift in flowcharts of figures 4.15 and 4.19). It must be smaller than the search shift; if it is equal to the search shift then the small shift strategy is not used.

- **bigger patch**: use or not use this strategy: a bigger patch for LSM is used in case of an unsuccessful match (see LSM special in flowcharts of figures 4.15 and 4.19).

- **close the gaps**: use or not use this strategy (flowchart of figure 4.19).

The parameters used for the **LSM process** \((a_0, a_1, a_2, b_0, b_1, b_2, r, s)\) can be individually switched on or off; however, they are usually all used.
For the **epipolar constraint**, the following options can be selected:

- **use epipolar constraint**: use or not use epipolar constraint.
- **weight in P**: weight \((p_k)\) for the epipolar constraint in the weight coefficient matrix \(P\) (equation 4.17), defined as a multiple of the number \(n\) of pixels inside the image patch \((n, 2n, 4n, 8n)\) or as \(value\).
- **epi patch size**: window size to intersect with the epipolar line to find the epipolar segment, i.e., \(\Delta x, \Delta y\) in flowchart of figure 4.3. This parameter is usually set as \(lsm\ psize/2\), i.e., half of the LSM patch size (e.g., as in figure 4.4); \(lsm\ psize\) is adequate for small LSM patch sizes (7, 9 pixels); it can also be defined as equal to the parameter \(max_{x,y}\), or double of it or as an arbitrary \(value\).

The rest of the parameters for the **automatic matching process** are the following:

- **max sigma 0**: threshold of \(\hat{\sigma}_0\) (see eq. 4.10) to accept the matching result.
- **max sigma x,y**: thresholds of \(\hat{\sigma}_x = \hat{\sigma}_{a0}\) and \(\hat{\sigma}_y = \hat{\sigma}_{b0}\) (see eq. 4.10) to accept the matching result.
- **max x,y**: thresholds of the displacement of the result of the matching process from the starting position (the two resulting parameters \(a_0\) and \(b_0\)). It is usually set as the effective pixel shift performed by the matching process (e.g., half of the value of the parameter \(pixel\ shift\) if the strategy \(search\ shift/2\) is used).
- **max x,y small**: same as above for the cases of \(small\ shift\) (see flowcharts of figures 4.15 and 4.19)(e.g., one quarter of the value of the parameter \(pixel\ shift\) if the strategy \(no\ search\ shift/4\) is used).
- **patch size x,y**: LSM patch size.
- **bigger patch size x,y**: LSM patch size in case of \(LSM\ special\) (see flowcharts of figures 4.15 and 4.19), i.e., if the strategy \(bigger\ patch\) is used.
- **pixel shift**: value of pixel shift.
- **voronoi x,y**: sizes of grid elements for the computation of the Voronoi tessellation. This parameter is not relevant, it depends on the image format (examples can be found in the appendix C).
- **grid x,y**: size of the grid elements for the \(close\ the\ gaps\) process (see flowchart of figure 4.19). This parameter depends on the image format (examples can be found in the appendix C).

As explained previously, some parameters have to be determined by manually testing the LSM process on the images. These are the thresholds for the quality control (\(max\ sigma 0, max\ sigma x,y\)) and the image patch sizes for the matching process (\(patch\ size x,y\) and \(bigger\ patch\ size x,y\)). No rules or advises can be given regarding these parameters. Some examples of parameter sets can be found in appendix C.

The most important parameter for the automatic matching process is the effective pixel shift. It affects the amount of matched data as well as the time required by the matching process. Anyhow, different values of this parameter do not influence relevantly the accuracy of the performed measurement (see chapter 6).
4 MATCHING PROCESS

4.3 FILTERING

The quality controls of the matching process described in section 4.1.4 serve to minimize the number of mismatches. Anyway, errors may be expected in the set of corresponding points produced. Filters can be applied to the set of matched points before computing their 3-D coordinates, reducing therefore the number of possible blunders.

4.3.1 Regularization of the Grid

Since the seed points do not have to be snapped onto a regular grid, the matching process has the undesired effect of producing an irregular grid of points in the template image (see figure 4.23, left). Before the application of any filter, the set of matched points has to be uniformed to a regular grid (center of figure 4.23). This is easily achieved by matching all the points shifted to the regular grid. The flowchart of figure 4.22 describes the process.

![Flowchart of the process regularization of grid.](image)

First, a regular grid is generated in the region of interest (selected manually) of the template image. The nearest matched point \((mx_k, my_k)\) is searched for each element \((i)\) of the regular grid. In the search image, the matched point \((sx_k, sy_k)\) is shifted by the distance of the template point to the regular element \((dx = x_i - mx_k, dy = y_i - my_k)\). LSM is then computed again and the point is stored only if the result pass the quality check, as described in section 4.1.4.
4.3 FILTERING

4.3.2 Neighborhood Filtering

For the definition of the filter, the smooth characteristic of the surface of the human body is taken into account. As can be seen in figure 4.24, the transformed image patches of neighbor points belonging to a common smooth surface have similar shapes.

A neighborhood filter can indeed be defined to check the local uniformity of the shape of the transformed image patches in the set of matched points. To characterize the shape of the transformed patches, the angles of the two major axes to the horizontal and vertical axes and the size of a (1,1) transformed patch are used (see figure 4.25).

As illustrated in figure 4.25, left, the two angles can be computed as:

\[
\begin{align*}
\alpha_1 &= \arctan \left( \frac{b_1}{a_1} \right) \\
\alpha_2 &= \arctan \left( \frac{b_2}{a_2} \right)
\end{align*}
\]  

(4.21)

where \( a_1, a_2, b_1, b_2 \) are the affine parameters of the transformed patch.
Fig. 4.25 Neighborhood filter: defining the three values which characterize the shape of the transformed patch. Left: angles $\alpha_1, \alpha_2$; right: help values $(A, B, a, b)$ for the definition of the size.

As shown in figure 4.25, right, the size of the $(1,1)$ transformed patch can be computed as:

$$\text{size} = A \cdot B - a \cdot B - A \cdot b$$  \hspace{1cm} (4.22)

with:

$$A = 2 \cdot |a_1 + a_2|$$
$$B = 2 \cdot |b_1 + b_2|$$
$$a = 2 \cdot |a_2|$$
$$b = 2 \cdot |b_1|$$  \hspace{1cm} (4.23)

where $a_1, a_2, b_1, b_2$ are the affine parameters of the transformed patch.

The three shape characteristic values of the transformed patch ($\alpha_1, \alpha_2$ and the size of the $(1,1)$ transformed patch) are computed for each matched point. These values are compared to the mean values of the points matched in the neighborhood (defined with a mask, e.g., $3x3$ points). If the differences are larger than the thresholds, then the matched point is rejected. The flowchart of figure 4.26 shows the process.
4.3 FILTERING

The options to be chosen for the filtering process, shown in figure 4.27, are the mask size defining the neighborhood, the thresholds for the difference of angles (in degrees) and the threshold for the difference of the sizes (in percentages).

![Fig. 4.27 Options for the neighborhood filter.](image)
This chapter describes the process for the measurement of the surface of human body parts. In section 5.1.1 it will be first described how the 3–D point cloud is computed from the matched points. Section 5.2 will then treat the visualization and modeling aspects. Finally two different applications are presented (human face modeling and measurement of blood vessel branching casting) to prove the functionality of the proposed method.

5.1 3–D POINT CLOUD

This section describes how the 3–D coordinates of the matched points are computed and presents a simple 3–D filter.

5.1.1 Forward Ray Intersection

Figure 5.1 shows a typical situation: three images with known interior and exterior orientation are displayed with the image coordinate of the same object point (black circle) in each of them; the rays projection center–image coordinate are also displayed.

In case of a perfect calibration of the cameras and a perfect matching of the corresponding points in the images, the three rays would intersect in a common point, i.e., the object point. This is usually not the case and the three rays do not intersect precisely. However, a fictitious intersection point can be found by minimizing the distances to all the rays, this process is called forward ray intersection. The same mathematical model of section 3.1 describing the projection of the object space onto the sensor plane of an imaging device is used here. It is modeled by the collinearity
The forward intersection problem can be solved by least squares estimation. In this case the observations are the image coordinates of corresponding points \((x'_i, y'_i)\) in the \(n\) images of different views and the unknowns are the object space coordinates \((X, Y, Z)\) of the observed points.

The derivatives of the uncorrected terms \((\tilde{x}', \tilde{y}')\) of the collinearity equation with respect to \(X, Y, Z\) are (Luhmann, 2000):

\[
\begin{align*}
\left( \frac{\partial \tilde{x}'}{\partial X} \right)_i &= -\frac{c_i}{N_i^2} \cdot (N_i r_{11i} - Z x_i r_{13i}) \\
\left( \frac{\partial \tilde{y}'}{\partial X} \right)_i &= -\frac{c_i}{N_i^2} \cdot (N_i r_{12i} - Z y_i r_{13i}) \\
\left( \frac{\partial \tilde{x}'}{\partial Y} \right)_i &= -\frac{c_i}{N_i^2} \cdot (N_i r_{21i} - Z x_i r_{23i}) \\
\left( \frac{\partial \tilde{y}'}{\partial Y} \right)_i &= -\frac{c_i}{N_i^2} \cdot (N_i r_{22i} - Z y_i r_{23i}) \\
\left( \frac{\partial \tilde{x}'}{\partial Z} \right)_i &= -\frac{c_i}{N_i^2} \cdot (N_i r_{31i} - Z x_i r_{33i}) \\
\left( \frac{\partial \tilde{y}'}{\partial Z} \right)_i &= -\frac{c_i}{N_i^2} \cdot (N_i r_{32i} - Z y_i r_{33i})
\end{align*}
\]

where:

\[
\begin{align*}
Z x_i &= r_{11i} \cdot (X - x_{0i}) + r_{21i} \cdot (Y - y_{0i}) + r_{31i} \cdot (Z - z_{0i}) \\
Z y_i &= r_{12i} \cdot (X - x_{0i}) + r_{22i} \cdot (Y - y_{0i}) + r_{32i} \cdot (Z - z_{0i}) \\
N_i &= r_{13i} \cdot (X - x_{0i}) + r_{23i} \cdot (Y - y_{0i}) + r_{33i} \cdot (Z - z_{0i}).
\end{align*}
\]

The derivatives of the correcting term \(d\tilde{x}'\), \(d\tilde{y}'\) are very small and can be therefore omitted, thus:

\[
\begin{align*}
\left( \frac{\partial \tilde{x}'}{\partial X} \right)_i &\approx \left( \frac{\partial \tilde{x}}{\partial X} \right)_i \\
\left( \frac{\partial \tilde{x}'}{\partial Y} \right)_i &\approx \left( \frac{\partial \tilde{x}}{\partial Y} \right)_i \\
\left( \frac{\partial \tilde{x}'}{\partial Z} \right)_i &\approx \left( \frac{\partial \tilde{x}}{\partial Z} \right)_i \\
\left( \frac{\partial \tilde{y}'}{\partial X} \right)_i &\approx \left( \frac{\partial \tilde{y}}{\partial X} \right)_i \\
\left( \frac{\partial \tilde{y}'}{\partial Y} \right)_i &\approx \left( \frac{\partial \tilde{y}}{\partial Y} \right)_i \\
\left( \frac{\partial \tilde{y}'}{\partial Z} \right)_i &\approx \left( \frac{\partial \tilde{y}}{\partial Z} \right)_i
\end{align*}
\]

The unknown vector \(x\), the design matrix \(A\) and the observation vector \(l\) result therefore in:

\[
\begin{align*}
x &= (dX, dY, dZ)^T \\
A &= \begin{bmatrix}
\left( \frac{\partial \tilde{x}'}{\partial X} \right)_1 & \left( \frac{\partial \tilde{y}'}{\partial X} \right)_1 & \left( \frac{\partial \tilde{z}'}{\partial X} \right)_1 \\
\vdots & \vdots & \vdots \\
\left( \frac{\partial \tilde{x}'}{\partial X} \right)_n & \left( \frac{\partial \tilde{y}'}{\partial X} \right)_n & \left( \frac{\partial \tilde{z}'}{\partial X} \right)_n
\end{bmatrix} \\
l &= (x'_1 - \tilde{x}'_1, y'_1 - \tilde{y}'_1, \ldots, x'_n - \tilde{x}'_n, y'_n - \tilde{y}'_n)^T
\end{align*}
\]
5.1 3–D POINT CLOUD

where

\[ dX, dY, dZ \] changes of the unknown object coordinates from the estimations,
\[ x'_i, y'_i \] observed (i.e., measured) image coordinates of the point in image \( i \),
\[ \hat{x}_i, \hat{y}_i \] the estimated image coordinates, computed by backprojecting the estimated 3–D point onto the image \( i \) according to equation 5.1,
\( n \) number of images.

For the estimation of \( x \), the Gauss-Markov model of least squares is used:

\[
\begin{align*}
\hat{x} &= (A^T A)^{-1} A^T l \quad \text{solution vector,} \\
v &= A\hat{x} - l \quad \text{residual vector,} \\
\sigma_0^2 &= \frac{n^T v}{2n - 3} \quad \text{variance factor,} \\
Q &= \sigma_0^2 \cdot (A^T A)^{-1} \quad \text{covariance matrix,} \\
\sigma_i^2 &= Q_{ii} \quad \text{variance factor of the single unknowns.}
\end{align*}
\]

The system is solved iteratively, i.e., the estimated unknown vector \( \hat{x} \) is computed according to equation 5.6, the design matrix \( A \) and the observation vector \( l \) are updated and the estimated unknown vector \( \hat{x} \) is computed again, this until the changes of the unknowns are smaller than a threshold (usually 0.5%).

The precision of the estimated parameters \( \sigma_i \) resulting from equation 5.6 expresses the theoretical precision of the computed 3–D coordinates in the three axis directions.

5.1.2 Filtering

To reduce the amount of data, a simple 3–D filter may be applied. The object space is divided into voxels of variable dimensions and the points contained in each voxel are reduced to its center of gravity (figure 5.2).

The data resulting after this filtering process have a density more uniform and the amount of data is reduced. However, the accuracy of the measurement will also decrease. Figure 5.3 shows an example of the result after applying the filtering process (with a voxel size of 2x2x2 mm) to a selection of points measured on a human face. This simple filter does not take into account the fact that the points can lie on a curved surface. It can therefore work properly only if the 3–D point cloud is sufficient dense and the voxel size sufficient small, so that each voxel would contain points that could be assumed to lie on a plane; in this case, the center of gravity would lie on the same plane. If strong noise is present in the data or if the point cloud is not dense enough, more complex Gaussian filters (Borghese and Ferrari, 2000) have to be applied to the data (see, e.g., section 5.4.2).
5 SURFACE MEASUREMENT

Fig. 5.3 Simple 3–D filter: left: original 3–D data (25,636 points), in light grey the selected points; center: selected points before filtering (3,495 points), right: selected points after filtering with a voxel size of 2x2x2 mm (1,231 points).

5.2 VISUALISATION AND MODELING

This section describes the methods implemented in this work to visualize efficiently 3–D point clouds and some basic modeling procedures used for presentation purposes.

5.2.1 3–D Point Cloud Visualisation

The 3–D information in form of a point cloud alone is not sufficient for visualisation purposes. Some radiometric information is required in this case. For this reason, the 3–D point could is backprojected onto a calibrated and oriented image according to the collinearity equation (equations 3.3, 3.4 and 3.5). The radiometric information is then read directly from the image, as greyscale value or RGB color information depending on the type of the images. Eventually, if required, bilinear resampling can be applied to the image (according to equations 4.7 and 4.8).

A 3–D viewer with some basic editing functions was developed for an efficient visualisation of the point cloud. The detailed description of the viewer will follow in section A.2.5. Figure 5.4 shows two examples, on the left the result of a full body laser scan without radiometric information and on the right the result of the surface measurement process described in this chapter with radiometric information.

Some basic editing functions were added to the viewer: single points can be selected getting information about the point number, its 3–D coordinates and the color information (RGB values) if available; multiple points can be selected as rectangular box or as a (clockwise) hand drawn region; the selected points can then be deleted from the data set or stored as separate data set. This function allows a simple and easy removal of outliers or unuseful points from the data (e.g., points from the background or points not useful for modeling purposes). Of basic importance is also the functionality that allows to get the 3–D coordinates of selected points. These can indeed be used as control points to determine the orientation of uncalibrated texture images (see next section).
5.2 VISUALISATION AND MODELING

Fig. 5.4 3–D point viewer. Left: a full body 3–D point cloud (determined with laser scanning), right: 3–D point cloud with radiometric information; each point has a color value.

5.2.2 Modeling

Basic surface modeling procedures, such as 2 1/2–D triangulation, Gaussian smoothing and texture mapping with a single image, were applied to achieve a nice visualisation of the surface measurement results. In house existing software and free software was used for the triangulation and smoothing tasks. A simple software was developed to solve basic texture mapping problems. To achieve better qualitative modeling results, commercial software can be used.

To generate a triangulated surface from the 3–D point cloud, 2 1/2–D Delauney triangulation is used (IGP software DTMZ). Unfortunately, this method can process only 3-D data with 2 1/2–D characteristic, i.e., the surface defined by the 3–D point cloud can be projected onto a plane without having overlapping areas. For this reason only landscape-like surface models can be generated (e.g., face masks). In case of required 3–D surface triangulation, commercial software (e.g., Geomagic Wrap TM) can be used.

After the computation of a triangulated surface, the next step for the modeling process is the texture mapping. The standard format used nowadays for 3–D models is the Virtual Reality Modeling Language (VRML). Free software can be found for the visualization of VRML files. This format allows two different modes for texture mapping definition: texture mapping draping the entire texture image over the 3–D model or defining the texture for each triangle of the model.

In the first case, five parameters have to be defined: two translations and two scaling factors both in horizontal and vertical directions and a rotation. The entire image is then draped over the 3–D model as shown in figure 5.5. The disadvantages of this mode are the manual determination of the five parameters and the approximative result.
In the second texture mapping mode allowed by the VRML format, each triangle of the 3–D model has a texture information. This is achieved by assigning to each vertex a texture coordinate. Texture coordinates differ from the normal image coordinates in the way that the origin is in the bottom left corner of the image and their value ranges from 0 to 1. The advantage of this definition is the possibility to change the format of the texture image without modifying the 3–D model file. Figure 5.6 shows an example.

The figure shows also how the texture coordinates can be computed: the vertices of the 3–D model are backprojected onto the texture image according to the collinearity equation (equations 3.3, 3.4 and 3.5) and transformed in texture coordinates.

In case of an uncalibrated and unoriented texture image the following procedure can be used: few control points are selected manually on the 3–D model (using, e.g., the 3–D point cloud viewer described in section A.2.5) and on the texture image (see figure 5.7). The defined control points are then used to orient the texture image by bundle calibration (section 3.3). Since only few control points are selected, only the external orientation (position and angles) and the focal length can be determined with the calibration and orientation procedures. The results are however sufficiently accurate for texture mapping purposes.
5.3 CONSIDERATIONS

Fig. 5.7 Calibrating and orienting texture images: few control points are selected manually in the 3-D model and in the uncalibrated texture image. External orientation and focal length are then approximately determined by bundle calibration.

5.3 CONSIDERATIONS

The described surface measurement process is flexible and can be applied with different acquisition setups and for different purposes. It was successfully employed for face modeling using images acquired by five CCD cameras (D’Apuzzo, 2002b) (see also section 5.4), for human body modeling using three CCD cameras acquiring video sequences (D’Apuzzo et al., 2000) (see also section 7.7.2), for human body modeling using uncalibrated images acquired by a digital camera (Remondino, 2003), and for the modeling of blood vessels, where three CCD cameras acquired in eight positions twenty-four images (D’Apuzzo, 2001b) (see also section 5.5). In the next section, two of these applications are briefly presented to show the functionality of the proposed method.

The surface measurement procedure is a stand-alone process but it also constitutes part of the surface tracking process (described in chapter 7). In this case, the surface measurement process is performed for each frame of the acquired sequence.

5.4 APPLICATION 1: HUMAN FACE MODELING

The extensively employed methods to produce three dimensional computer models of the human face are laser scanning and coded light based triangulation approaches. The advantage of the presented method over these two techniques is the acquisition of the source data in a fraction of second, allowing the measurement of human faces with higher accuracy and even the possibility to measure dynamic events. Moreover, the developed software can be run on a normal home PC reducing the costs of the hardware.

In the past there was a collaboration for a project intended to evaluate the anatomical changes occurring with maxillo-facial and plastic surgery (Koch et al., 1996; D’Apuzzo, 1998). The goal of this actual work is the development of a portable, cheap and accurate system for the measurement and modeling of the human face (D’Apuzzo, 2002a).
5 SURFACE MEASUREMENT

5.4.1 System Setup

Figure 5.8 shows the setup of the used image acquisition system. It consists of five CCD cameras arranged in front of the subject. In case of required high accuracy, texture in form of random pattern can be projected simultaneously from two directions onto the face. The cameras are connected to a frame grabber which digitizes the images acquired by the five cameras at the resolution of 768x576 pixels with 8 bits quantization. A color image of the face without random pattern projection is acquired by an additional color video camera placed in front of the subject. It will be used for the realization of a photorealistic visualisation.

![Fig. 5.8 Setup of cameras and projectors.](image)

For calibration and orientation purposes, a calibration field with coded targets was designed specially for the face measurement process (see figure 5.9), so that all the target points could be viewed by the five cameras without occlusions. The camera system is then oriented and calibrated by bundle calibration method, as explained in the section 3.3.

![Fig. 5.9 Calibration field with coded targets. Left: construction design, right: an image.](image)

Since the natural texture of the human skin is relatively uniform, the projection of an artificial texture onto the face is required to perform robustly the matching process. A random pattern (see figure 5.10) is preferred over regular patterns to avoid possible mismatches. In fact, regular pattern consist of repeating musters and the automatic matching process may easily produce erroneous correspondences. The resolution of
the projected random pattern has to be adjusted to the image format. It has to result in the acquired images in structures with the size of few pixels (see figure 5.11). The use of two projectors enables a focused texture even on the lateral sides of the face. Some results achieved without the projection of an artificial texture are presented in section 5.4.4. Figure 5.11 shows the result of the random pattern projection on a face and figure 5.12 shows the five images acquired by the cameras.

![Projected random pattern](image1)

**Fig. 5.10** Projected random pattern.

![Result of the random pattern projection on a face](image2)

**Fig. 5.11** Result of the random pattern projection on a face; right: a detail, in the acquired image the structures of the random pattern have the size of few pixels.

### 5.4.2 Surface Measurement

Since the human face is a steep surface and both sides of the face are not visible by the same camera, the five acquired images are used as two separate set of triplets, one for each side of the face (see figure 5.12). They are processed independently and at the end, the results are merged into a single data set.

![The five images acquired by the cameras](image3)

**Fig. 5.12** The five images acquired by the cameras. For the processing two triplets are used, one for each side of the face.
The required intervention of the operator for the matching process is reduced to the semi-automated definition of about ten seed points (see section 4.2.1) and to the selection of a contour of the region to be measured (see figure 5.13). The operation can be performed in a couple of minutes, then the process will continue completely automatically.

![Fig. 5.13 Seed points and contour for the two sides of the face.](image)

Figure 5.14 shows the two sets of matched corresponding points established by the matching process on the two halves of the face. On a Pentium III 600 MHz machine, about 25'000 points are matched on half of the face in about 10 minutes.

![Fig. 5.14 Matched corresponding points in the five images of figure 5.12.](image)

The computation of their 3–D coordinates is achieved by forward ray intersection (see section 5.1.1) and lasts a couple of seconds. The achieved precision ($\sigma_i$ of equation 5.6) of the 3–D points is about 0.4 mm in the sagittal direction and about 0.2 mm in the lateral direction. As can be seen in figure 5.15 left, the point cloud is very dense (45’000 points) and some outliers are present. In the center line of the face, the region of overlap of the two joined data set can be observed. However, no relevant discrepancies of the two data set is present. Gaussian filters (Borghese and Ferrari, 2000) are applied to the 3–D point cloud and afterward thinned to reduce the number of points (see figure 5.15 right).

### 5.4.3 Modeling and Photo Realistic Visualisation

The last step of data processing is the generation of a triangulated surface from the cleaned point cloud and the application of color texture. A meshed surface is generated from the 3–D point cloud by 2.5–D Delauney triangulation.
5.4 APPLICATION 1: HUMAN FACE MODELING

Fig. 5.15 Left: Measured 3–D point cloud, 45'000 points. Right: after filtering and thinning, 10'000 points.

To achieve a photorealistic visualization, the natural texture acquired by the color video camera is draped over the model of the face. Figure 5.16 shows the surface model, the texture image and two views of the resulting face model with texture. Figure 5.17 shows two other examples of face models.

Fig. 5.16 Photorealistic visualisation. Left: shaded surface model, texture image. Right: face model with texture.

Fig. 5.17 Photorealistic visualisation. Two other examples of face models.
5.4.4 Measurement Without Artificial Texture Projection

The measurement of human faces can be performed also without projecting an artificial texture. In this case, the matching process will perform less robustly. An example is shown in figures 5.18 and 5.19. The acquisition system presented in appendix B was used to acquire five images of a human face (figure 5.18). The five images were used as two independent triplets to perform the matching process, as explained in section 5.4.2. The final result, after forward ray intersection is a point cloud of totally 17'000 points. As can be see in figure 5.19, blunders are present and some regions could not be measured. Indeed, the result is less robust and precise than the case with random pattern projection; the achieved theoretical precision ($\hat{\sigma}_f$ of equation 5.6) is about 2.5 mm in the sagittal direction and about 0.5 mm in the lateral direction.

**Fig. 5.18** Five images acquired by the cameras without the projection of an artificial texture.

**Fig. 5.19** Result of the surface measurement process. 3–D point cloud with radiometric information (each point has a color value).

5.5 APPLICATION 2: MEASUREMENT OF BLOOD VESSEL BRANCHING CASTING

A pilot project was executed in cooperation with the department of radiology of the university hospital Zurich. The aim was to evaluate the accuracy levels of the current techniques used for the visualization and measurement of major blood vessels in the human body. Magnetic resonance (MR), computer tomography (CT) and digital subtraction (DS) angiographies were considered in this work. In the radiology literature, the comparison of different methods can be found. The scientific publications treat the detection of diseases using visualisation tools (Sommer et al., 1996; Brandt-Zawadzki and Heiserman, 1997; Kelekis et al., 1999; Skutta et al., 1999) or describe methods
5.5 APPLICATION 2: MEASUREMENT OF BLOOD VESSEL BRANCHING CASTING

for the measurement of blood vessels with CT techniques (Rubin et al., 1998), but a quantitative assessment is still missing. The goal of this pilot project was to establish if it was possible to determine the potential accuracy of the three techniques (CTA, MRA, DSA) for the measurement of blood vessels and compare them using unbiased data as reference. The abdominal aorta and its main branches (iliaca communis and iliaca externa) were chosen for this test (see figure 5.20).

![Fig. 5.20 Aorta abdominalis and its main branches: iliaca communis, iliaca externa, iliaca interna (Benninghoff and Gotter, 1961).](image)

MR-, CT-, and DS- angiographies were first performed on a corpse. Then, a casting of the aorta abdominalis and its main branches (iliaca communis, iliaca externa) was prepared. The idea of the project was to measure the casting with optical methods and use the results as unbiased reference for a quantitative comparison between the three techniques.

5.5.1 System Setup and Calibration

The main characteristics of the aortal cast are: (1) the thin (0.5-1.5 cm), elongated (40 cm long) and branched shape, (2) its flexibility and (3) the complete lack of natural texture. For these characteristics the measurement task was rather difficult.

At first an adequate frame was constructed to immobilize the casting and sixteen white spheres were fixed on metal lines for the determination of the external orientation of the images. The rotation of the entire frame containing the blood vessel branching was possible without moving the signalized spheres. The design of the frame allowed an easy replacement of the casting with a new one. Figure 5.21 shows an image of the blood vessel branching casting fixed in the frame.

![Fig. 5.21 Left: blood vessel branching casting fixed in the frame. Right: configuration of the frame.](image)
Three CCD cameras with 768x576 pixel resolution were used for the image acquisition. Two slide projectors placed at both sides of the cameras projected a random pattern texture. To achieve a complete 360° imaging of the object, the frame was turned in eight positions. At each position, a triplet of images was acquired, resulting a total of twenty four images (figure 5.22).

![Setup of the acquisition system: three CCD cameras and two slide projectors (left). The frame is turned in eight positions (center), this result in height triplets of images around the object (right).](image)

Figure 5.23 shows a triplet of images for the first position and the images taken by the central camera at the height positions of the frame. As it can be seen, the shape of the casting was very thin and elongated. Its branched form, together with the frame construction, caused relatively wide occlusions (i.e., in the turns number 3 or 7). A large number of images was therefore required to assure the complete coverage of the object.

![One triplet of images (top) and images from the central camera for the height turns (bottom).](image)
5.5 APPLICATION 2: MEASUREMENT OF BLOOD VESSEL BRANCHING CASTING

To calibrate the three CCD cameras, a 3-D reference field with coded target points was used (figure 3.4 left). The reference field was placed in the object space and the three CCD cameras acquired a triplet of images. The images were used to perform the calibration (internal orientation and determination of the lens distortion of the three CCD cameras) and to determine the external orientation for the first position. Once the calibration process was concluded, the reference field was replaced by the frame containing the blood vessel branching casting. The frame was turned eight times of about 45 degrees around its axis. This rotation can be considered equivalent to a displacement of the three cameras around the object (figure 5.22). To determine the external orientation at the different positions of the frame, the signalized spheres (figure 5.21) were used as control points. Their image coordinates were measured semi automatically by centroid operators and finally bundle adjustment led to a mean accuracy for the external orientation (position of the cameras) of about 0.8 mm in the three axis directions.

5.5.2 Measurement and Modeling

The matching process is performed independently for each triplet acquired by each turn of the frame. As the blood vessel branching casting had a complete lack of natural texture, the projection of an artificial texture was required to perform successfully the matching process. For this purpose, two slide projectors projected a random pattern from two directions (figure 5.22). Figure 5.24 shows the effects achieved.

As the object was wide, the projected texture was unfocused in some regions of the surface. This undesired effect reduced the accuracy potential of the matching process. Moreover, the shape of the branching was critical: the main branching (iliaca externa) was about 17 pixels thick in the image. For the matching process, a patch size of 11x11 pixels was chosen, therefore only a thin part along the center of the branching could be matched. Occlusions caused by the fixation frame, the signalized spheres and the branching itself, also disturbed the automated matching procedure. Combining all these effects, the matching process required many manual interventions. However, good matching results were achieved and dense sets of matched points could be determined for all the eight turns.

The 3-D coordinates of the matched points were computed with a mean accuracy of 0.2 mm by forward ray intersection using the known calibration and orientation.
data. This process was performed independently for the eight data sets and the results were merged together into a cloud of about 13’000 points (figure 5.25, left). A 3–D filter (see section 5.1.2) was applied to reduce the amount of redundant data (overlap between data sets), to get a more uniform density of the point cloud and to reduce the noise. Figure 5.25 shows the point cloud before (left, 13’223 points) and after the filtering process using two different reduction levels (in the center 8’040 points and right 2’537 points).

Fig. 5.25 Original 3–D point cloud (left) and two level of filtered data (center, right).

In this work, the modeling process was intended only for visualisation purposes. The generation of a complete and unique digital surface model was a difficult task. Indeed, the 3–D point cloud was very dense in some regions and poor in others. The areas where the object was very thin caused the mainly problems. For example, the two iliacas interna were not fully measured, because of their extreme thinness and therefore these two parts were excluded from the modeling process.

In order to generate a surface model starting from unorganized 3–D point cloud, a commercial software (Wrap of Geomagic™) was used. Figure 5.26 shows the results of the modeling process. It works fine in the region of the aorta abdominalis and the iliaca communis, but rather poor at the end of the iliacas externa, where the blood vessels were very thin.

Fig. 5.26 DSM of the blood vessel branching for visualisation purposes.
This section treats the validation of the proposed surface measurement method. A solid reference object was measured as described in section 5.4 and the results were compared with the measurement achieved by a coordinate measurement machine. Section 6.1 describes the setup for the validation process, i.e., the acquisition system, the reference object and the reference data. Section 6.2 describes the measurement process and section 6.3 the surface comparison. Finally, section 6.4 presents the comparison results and section 6.5 some considerations.

6.1 SETUP

This section describes the setup of the validation process. A multiple camera system as described in appendix B and by D’Apuzzo and Kochi (2003) was used. Additionally, a slide projector was used to project a random pattern onto the surface, as described in section 5.4. For the validation process only three of the five CCD cameras were used. Figure 6.1 shows the image acquisition system.

Fig. 6.1 Acquisition system used for validation (see appendix B): three progressive scan CCD cameras (model DragonFly) with a 680x480 pixel sensor and IEEE-1394 port. A slide projector is used for the projection of artificial texture onto the surface.
Fig. 6.2 Reference field used for calibration and orientation of the camera system.

Fig. 6.3 Dummy head used as reference object. Right: with random pattern projection.

Fig. 6.4 Reference data: 2448 points in 37 profiles, measured by a coordinate measurement machine (*Carl Zeiss UMC 550S*) with an accuracy of 5 μm.
The camera system was calibrated and oriented as described by D’Apuzzo and Kochi (2003) using a reference field with signalized points of known 3-D coordinates (see figure 6.2). The accuracy of the calibration and orientation processes can be expressed according to equation 3.11 and results in this case in \( \sigma_0 = 0.5 \ \mu m \). For the used cameras, it corresponds to \( 1/15^{th} \) pixel.

A dummy head was used as reference object (see figure 6.3). This was firstly measured by a coordinate measurement machine (Carl Zeiss UMC 550S) with an object space accuracy of 5 \( \mu m \) (precision given by the constructor). The measurement results in profiles as shown in figure 6.4. A total number of 2448 points in 37 profiles were measured on a half side of the head.

6.2 SURFACE MEASUREMENT

Image 6.5 shows the three acquired images used for the surface measurement process. The full functionalities of the matching process proposed in chapter 4 were used. Figure 6.6 shows the template image with the defined contour and seed points. Figure 6.7 shows the result of the matching process (matching at every four pixels) and the result after grid regularization (see section 4.3.1).

Fig. 6.5 Triplet of the dummy head with random pattern projection for surface measurement.

Fig. 6.6 Matching process: template image with defined contour and seed points.
Six different matching resolutions were used: matching each pixel in the images, matching every two pixels, three pixels, four pixels, six pixels and eight pixels. The forward ray intersection process (see section 5.1.1) results in the point clouds shown in figure 6.8. As can be seen in the figures, the results are consistent and few blunders were present (e.g., at the top of the head, on the tip of the nose). This will then be confirmed by the quantitative analysis of the results (see section 6.4).

The theoretical precision of the computed 3–D point can be expressed as the Helmert’s point error $\sigma$ according to:

$$\sigma^2 = tr(Q) = \sigma_x^2 + \sigma_y^2 + \sigma_z^2$$  \hspace{1cm} (6.1)

where $Q$ is the covariance matrix and $\sigma_x, \sigma_y, \sigma_z$ are the variance factors of the three coordinates according to equation 5.6. These three values express the theoretical precisions in the three axes directions, whereas $\sigma$ expresses an average point error independent from the orientation of the coordinate system. The results for the six cases are shown in table 6.1. The total number of matched points is listed together with the average theoretical precision $\bar{\sigma}$, its standard deviation and its maximal value. The next three columns list the same values ignoring blunders with a theoretical precision larger than 1.5 mm ($\sigma > 1.5 \text{ mm}$); the last column of the table list the number of blunders.

<table>
<thead>
<tr>
<th>Surface</th>
<th>Surface points</th>
<th>$\bar{\sigma}$</th>
<th>$\text{std.dev.}$</th>
<th>$\text{max } \sigma$</th>
<th>$\bar{\sigma}_{&lt;1.5}$</th>
<th>$\text{std.dev.}$</th>
<th>$\text{max } \sigma$</th>
<th>- pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 pixel</td>
<td>92216</td>
<td>0.21</td>
<td>0.28</td>
<td>9.12</td>
<td>0.20</td>
<td>0.10</td>
<td>1.50</td>
<td>476</td>
</tr>
<tr>
<td>2 pixels</td>
<td>22944</td>
<td>0.19</td>
<td>0.09</td>
<td>2.88</td>
<td>0.19</td>
<td>0.07</td>
<td>1.41</td>
<td>19</td>
</tr>
<tr>
<td>3 pixels</td>
<td>10152</td>
<td>0.19</td>
<td>0.08</td>
<td>2.43</td>
<td>0.19</td>
<td>0.07</td>
<td>1.26</td>
<td>5</td>
</tr>
<tr>
<td>4 pixels</td>
<td>5722</td>
<td>0.19</td>
<td>0.08</td>
<td>1.76</td>
<td>0.19</td>
<td>0.07</td>
<td>1.39</td>
<td>4</td>
</tr>
<tr>
<td>6 pixels</td>
<td>2532</td>
<td>0.20</td>
<td>0.10</td>
<td>2.88</td>
<td>0.19</td>
<td>0.07</td>
<td>1.35</td>
<td>2</td>
</tr>
<tr>
<td>8 pixels</td>
<td>1428</td>
<td>0.19</td>
<td>0.07</td>
<td>1.24</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
6.3 SURFACE COMPARISON

To compare the measured surface with the reference data two tasks have to be performed. Firstly the two surface data have to be aligned and secondly the distance between the two surfaces has to be defined and determined. Additional problems to be considered for the comparison are: (a) the reference data (profiles) and the measured data (point cloud) have different structures, (b) they have different resolutions, (c) they are defined in different coordinate systems, (d) no common points, planes, or axes can be determined.

Fig. 6.8 Surface measurement of the reference object: point clouds resulted matching each pixel, every two, three, four, six and eight pixels.
6.3.1 Alignment

The adequate solution to align two point clouds with different resolutions and no common points is provided by the ICP (Iterative Closest Point) algorithm proposed by Besl and McKay (1992). The commercial software *Geomagic Studio™* was used for this process. The average error resulting from the alignment process is 0.19 mm. Figure 6.9 shows the results.

![Fig. 6.9 Alignment result, average error is 0.19 mm. Left: visualisation of aligned reference data as profiles and measured data as point cloud. Right: snapshot of the commercial software *Geomagic Studio™*.](image)

6.3.2 Distance Reference Data to Measured Data

The determination of the distance between the reference data and the measured data is not a trivial task because their resolution is different and no common points, planes or axes exist. Since the measured point cloud (99216 points) has a much higher density than the reference data (2448 points), it was decided to compute for each point of the reference data the distance to the measured point cloud. Figure 6.10 shows the two data set. The figure 6.11 shows the detail of the dotted box of figure 6.10 and explains how the distance of the reference data to the measured data is computed.

The distance of a point in the reference data to the point cloud is computed as follows: (a) determine the three closest points in the point cloud, (b) compute the distance of the reference point to the plane defined by the three points found in the measured data. The determination of the distance of a point to a plane introduces an additional error, i.e., the approximation of a curved surface by a set of triangular facets, as shown in figure 6.12. The error added by this factor ($e1$ or $e2$ in figure 6.12) decreases with the increment of the density of the measured point cloud. For this reason it is more appropriate to compare the reference data with the measured point cloud with highest density. Therefore, the 92216 point cloud achieved by matching each pixel was used for the comparison. The distance in the space of two adjacent points ($l1$ or $l2$ in figure 6.12) is about 0.5 mm for this data. Assuming for the reference object a minimal curvature radius of 5 mm ($r$ in figure 6.12), the maximal error of the distance to the real surface is $e = r \cdot (1 - \cos \sin^{-1} \frac{1}{r}) = 0.01 \text{ mm}$. The error added by this factor can therefore be considered insignificant.
6.4 COMPARISON RESULTS

Fig. 6.10 Reference data as profiles and measured data as point cloud. Right: detail (no common points exist and the density of the two data set is different).

Fig. 6.11 Determination of the distance of the reference data to the measured data. Left: for each point in the reference data, the three closest point in the measured data are determined. Right: the distance $d$ is computed as distance to the plane formed by these three points.

Fig. 6.12 The error caused by approximating the real surface with facets decrease with the increment of the density of the measured point cloud: $e_1 < e_2$.

6.4 COMPARISON RESULTS

This section will present and discuss the results of the comparison of the reference data and the measured data as well as the comparison of the six different measured point clouds.
6.4.1 Distance Error

The distance of the reference data to the measured point cloud is used as distance error. Table 6.2 summarizes the comparison of the reference data to the point cloud resulted by matching each pixel of the template image (92216 points). Values with sign and absolute values are given. The average absolute error of 0.2 mm can be considered the value describing the precision of the measurement system. Figures 6.13 and 6.14 show graphically the error distribution.

Table 6.2 Distance reference data to measured data (values with sign and absolute values). Average [mm], standard deviation [mm], minimal and maximal values [mm].

<table>
<thead>
<tr>
<th></th>
<th>average</th>
<th>std.dev.</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>-0.02</td>
<td>0.27</td>
<td>-1.13</td>
<td>1.29</td>
</tr>
<tr>
<td>absolute error</td>
<td>0.20</td>
<td>0.18</td>
<td>0.00</td>
<td>1.29</td>
</tr>
</tbody>
</table>

Fig. 6.13 Distance reference data to measured data: error distribution

Fig. 6.14 Distance reference data to measured data: absolute error distribution.
Additionally, the commercial software *Geomagic Qualify*™ was used to compare the surface with the reference data. Table 6.3 summarizes the results. They confirm the achievements of the computation of the single distances.

<table>
<thead>
<tr>
<th></th>
<th>average</th>
<th>std.dev.</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>absolute error</td>
<td>0.19</td>
<td>0.20</td>
<td>0.00</td>
<td>1.36</td>
</tr>
</tbody>
</table>

Unfortunately, no further numerical values can be obtained from the software. However, a graphical representation of the error distribution can be plotted as shown in the figure 6.15. As can be seen, the larger errors appear in the expected difficult areas, as for example steep surfaces, regions with reflexions and regions where the projected texture was unfocused. These results confirm the assumptions made in chapter 4.

![Fig. 6.15 Distance of the measured point cloud to the reference data, from software Geomagic Qualify™.](image)
6.4.2 Comparison of Surfaces

A further interesting analysis of the measured data is the comparison between surfaces resulted by matching with different point densities. The commercial software *Geomagic Qualify*™ was used for this task. The point cloud with highest density (92216 points) was used as reference data. The results of the comparison are listed in the table 6.4. The average distance of the surfaces remains in all cases below 0.1 mm. Thus, it can be concluded that no relevant accuracy decrease can be observed by matching with different point densities.

<table>
<thead>
<tr>
<th>Surface</th>
<th>points</th>
<th>average distance</th>
<th>std. dev.</th>
<th>maximal distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 pixels</td>
<td>22944</td>
<td>0.09</td>
<td>0.11</td>
<td>0.99</td>
</tr>
<tr>
<td>3 pixels</td>
<td>10152</td>
<td>0.09</td>
<td>0.11</td>
<td>1.00</td>
</tr>
<tr>
<td>4 pixels</td>
<td>5722</td>
<td>0.09</td>
<td>0.11</td>
<td>1.00</td>
</tr>
<tr>
<td>6 pixels</td>
<td>2532</td>
<td>0.10</td>
<td>0.13</td>
<td>1.48</td>
</tr>
<tr>
<td>8 pixels</td>
<td>1428</td>
<td>0.10</td>
<td>0.14</td>
<td>1.45</td>
</tr>
</tbody>
</table>

6.5 CONSIDERATIONS

The first important consideration to underline is the full consistency of all the validation steps. In fact, the average error of the alignment process (0.19 mm), the theoretical precision of the forward ray intersection (0.20 mm) and the average absolute distance of the reference data to the measured surface (0.20 mm) results in the identical value of 0.2 mm. This value can therefore be considered as real accuracy of the measurement process achieved with the proposed acquisition system. Different accuracy values can however be expected depending on the used acquisition system (different cameras, different image format). The accuracy of the calibration and orientation process also influence the achieved measurement accuracy.

On the other hand, it has to be considered that a dummy head was used for the validation process. The measurement of the human skin presents in fact more difficulties caused by, e.g., its roughness, hairs, reflections. Anyhow, the validation process also proved that the theoretical precision resulting from the forward ray intersection process is consistent with the achieved accuracy of the measurements and can therefore be used as a quality indicator.
In section 7.1 the multi-image tracking process is introduced and the method for tracking single points is described in detail. In section 7.2 the process is extended for tracking entire surface parts. Section 7.3 describes the special case of tracking keypoints in the vector field of trajectories. The software implementation is discussed in section 7.4. The options and parameters to be set are described in section 7.5. Finally, section 7.7 presents results achieved by the proposed method for full body motion capture.

7.1 MULTI-IMAGE TRACKING PROCESS

7.1.1 Tracking in Multi-image Space

The basic idea of the multi-image tracking process is to track corresponding points in the multi-images through the sequence. Therefore, at the end of the process it is possible to compute their 3-D trajectories. The process is based on least squares matching techniques (LSM). The least squares matching algorithm introduced in section 4.1 is used to establish the spatial correspondences between the images of the different views (spatial LSM) and the temporal correspondences between subsequent frames (temporal LSM). Figure 7.1 graphically depicts the principle.

![Fig. 7.1 LSM tracking: temporal and spatial correspondences are established with LSM.](image)

The process starts from matched corresponding points in the multi-images and continues with the tracking loop: (1) predicting the position in the next frame, (2) searching the position with the best cross-correlation value and (3) establishing the point in the next frames using least squares matching (temporal LSM). These three steps are
computed independently for the multi-images. The figure 7.2 shows the process graphically.

The position of the tracked point at time $i+1$ is linearly predicted from the two previous times $i-1$ and $i$ (step 1). A search box is defined around this predicted position in the frame at time $i+1$ and is scanned for the position which has the best value of cross-correlation between the frame at time $i$ and the frame at time $i+1$ (step 2). The least squares matching algorithm is applied at that position and if the result passes the quality control (same as explained in section 4.1.4, without geometrical constraints) it can be considered the exact position of the tracked point in the new frame (step 3). Figure 7.3 shows how the tracking in image space starts. At the beginning of the process, it is not possible to predict the position of the point in the next time step. In this case, the search box has larger sizes and it is centered in the position of the actual time step (image coordinates $(x_i, y_i)$ in figure 7.3). Once the prediction is possible, the search box is smaller and centered in the predicted position (circle in frame $i+2$ in figure 7.3).

This process is performed independently for the different images. To test the individual results, spatial LSM is computed at the positions resulting from the temporal LSMs and if no significant differences occur between the two matches, the point is
7.1 MULTI-IMAGE TRACKING PROCESS

considered tracked and the process continues to the next time step. If instead the differences are too large, the step (2) of the process is repeated by searching the value of best cross-correlation in a larger region around the predicted position. LSM is then applied again (step 3) and if the result is rejected again, the tracking process stops. The flowchart in figure 7.4 illustrates all the operations of the multi-image tracking process in detail.

Fig. 7.4 Flowchart of the multi-image tracking process.
7 SURFACE TRACKING

7.1.2 3–D Trajectories

The result of the multi-image tracking process applied to a single point are its coordinates in the multi-images through the sequences, thus the 3–D coordinates of the point for each time step can be computed by forward ray intersection (see section 5.1.1), finally resulting in its 3–D trajectory. Its velocity and acceleration can also be determined for each time step. Figure 7.5 shows an example of several 3–D trajectories of points on the surface of a person walking.

![Fig. 7.5](image)

*Fig. 7.5* Measured 3–D point cloud of the first frame and 3–D trajectories of single points tracked on a walking person (sequence of figure 7.17).

The proposed process can be used to track well defined points on a human body surface (e.g., signalized target points). However, trajectories of single points are not sufficient to understand and record the motion and movement of a human or the changes of a human body surface part. The next two sections propose a method to extract more complex information from the image sequence.

7.2 TRACKING SURFACE PARTS

*Tracking surface parts* means tracking simultaneously a dense set of points belonging to a common surface. In this case, the result of the tracking process can be considered to be a dense vector field of trajectories.

7.2.1 Extended Algorithm

The requirement for the surface tracking process is completing the matching process on the surface parts of interest for at least the first multi-image set of the sequence. However, it is recommended to perform the process with a defined frequency, e.g., each two multi-image set, for the entire length of the sequence. This can be performed automatically with the process described in section 4.2. The tracking process depicted in section 7.1 is then applied to all the points measured on the surface of the first frames. With this approach, a new problem must be considered: during the sequence, some surface parts can get lost by occlusions and new surface parts can appear. For these reasons, a new functionality has to be integrated in the tracking process: before passing to the next time step, the local density of the data resulted from the tracking process is compared to a threshold. This operation is executed with a defined frequency, e.g., each two frames. In the regions of low density, new points are imported from the
7.2 TRACKING SURFACE PARTS

Previously computed data, i.e., surface matching process for the multi-image sets. In this way, newly appearing surface parts or lost points are integrated in the tracking process. Figure 7.6 shows the extended flowchart for the surface tracking process. For the determination of the local density of the matched data, the template image is divided into a regular grid whose size can be chosen. The number of matched points is counted for each grid element and if it is lower than a threshold (e.g., 80%), then the grid element is classified as a region with low density.

**Fig. 7.6** Flowchart: surface tracking (dotted box “tracking at time i”, from figure 7.4).

Figure 7.7 shows an example of the result of the surface tracking process applied to a sequence of a walking person acquired by two cameras (sequence of figure 7.17). The tracked points are displayed as trajectories from the previous frame. Figure 7.8 shows a detail of the tracked points, in the frame of time i+1 the newly included points can be recognized as single dots, as these points do not have any tracking information yet.

**Fig. 7.7** Example of surface tracking. Left: image pairs acquired at time i and i+1; right: tracked points displayed as trajectory from the previous frame.

85
7 SURFACE TRACKING

Fig. 7.8 Example of surface tracking. Detail from figure 7.7: in the frame at time \( i+1 \) (right) the new included points can be recognized as single dots.

7.2.2 Filtering

Tracking entire surface parts and not only good signalized single points can produce undesirable effects. Especially in case of poor texture of the surface to be measured, the tracking process may produce false trajectories. An example is shown in figure 7.9, which displays the computed 3–D trajectories for a part of the arm of a person walking (sequence of figure 7.17).

Fig. 7.9 Surface tracking: the false trajectories do not follow the uniform movement.

The false trajectories can easily be recognized because they do not follow the common movement of the majority. The vector field of trajectories can indeed be checked for consistency and local uniformity of the movement. Two filters are applied to the results to delete the false trajectories entirely or to truncate the false part of them.

7.2.2.1 Velocity and Acceleration Filter. The first filter removes large errors using velocity and acceleration thresholds (figure 7.10 left). The filter exploits the physical limitations of humans regarding velocities and accelerations of performed movements. Depending on the situation (e.g., walking, running), the two thresholds
of maximal velocity and acceleration can be defined at the beginning of the process and remain constant during the sequence.

7.2.2.2 Uniformity Filter. The second filter checks for the local uniformity of the movement, both in space and time (see figure 7.10 right). To check this property, the space is divided into regular voxels and local mean values of the velocity vectors are computed in each voxel for each time step. The single trajectories are compared to the local mean values of the velocity vector and truncated or removed if the differences in magnitude and direction are too large. The level of difference is computed as percentage of difference of the magnitude and as angle to the local mean velocity vector. Figure 7.11 shows graphically the uniformity filtering process and the flowchart in figure 7.12 explains it in detail.

7.2.2.3 Problems. After filtering the trajectories, problems may still remain. An ideal trajectory starts from the begin of the sequence and lasts until the end. However, in some cases, depending on the quality of the image sequence, on the type of surface to be tracked and on the movement performed, the result can be a set of broken trajectories with varying lengths (see figure 7.13). This effect is particularly strong when measuring the motion of a whole human body and is caused by occlusions, lack of texture, lost of tracked points and the appearance of new points. In order to solve this problem, the concept of key-point is introduced in the next section.
7.3 TRACKING KEY-POINTS

7.3.1 Definition

The key-point is a 3–D region defined in the vectorfield of trajectories, whose size can vary and whose position is defined by its center of gravity. The key-points are interactively defined in a graphical user interface; they can be placed at any time step.
and moved in the 3–D space (see figure 7.14). This will be described in detail in section A.2.7.

Fig. 7.14 Interactive definition of key-points. Left: definition of the size, right: moving the key-point in the 3–D space.

The key-points exploit the large amount of data resulted from the tracking process and extract a general description of the performed movement: a set of 3–D trajectories of selected key-points results from a dense vectorfield of trajectories.

7.3.2 Tracking

The key-points are tracked in a simple way: the position in the next time step is established by computing the mean value of the displacement of all the trajectories inside the region which define the key-point (see figure 7.15). The tracking process is explained in detail by the flowchart in figure 7.16.

Fig. 7.15 Tracking key-points.

It starts by counting the trajectories inside the 3–D region of the key-point and by computing their average displacement in space to the next time step. This is added to the actual 3–D position of the key-point to determine the position in the next time step. This process is repeated until no trajectories are found in the 3–D region of the key-point.
7 SURFACE TRACKING

7.3.3 Considerations
Since the key-points are defined interactively in the graphical user interface, they cannot be placed precisely. Therefore, they can be used to give a qualitative 3–D description of the movement performed by a person and not to measure it precisely. The size of the region to be tracked has an important role: if a small size is chosen, the key-point can be considered with a good approximation to lie on the tracked surface, in this case representing fictitious markers on the human body. If, on the other hand, larger sizes are chosen, the key-point represents a more approximative position of, e.g., joints of the human body. Two examples are given in section 7.7.3.

7.4 SOFTWARE IMPLEMENTATION
The amount of data to take into consideration is one of the problems for the implementation of the multi-image tracking process. A multi-image sequence is in fact composed of \( n \cdot k \) images, where \( n \) is the number of used cameras and \( k \) the length of the acquired sequence, i.e., the number of acquired multi-image sets. The resulting data may also become very large. In fact, the tracked trajectories are composed of dense sets of corresponding points in each multi-image set of the sequence. For the implementation of the tracking process it is therefore preferable to read in the memory of the PC only the required images (i.e., the multi-images of the previous, the actual and the next time steps) and to store the results directly to a file while keeping in the memory only the used data (i.e., the tracked points in the previous two frames). For this reason, ring buffers (with the length of three) were used to store the multi-images and the information about the tracked points. Ring buffers are circular array structures that permit a continuous increment (also negative) of the index still having a fixed size (in this case, three). Therefore, during the entire tracking process, always the same amount of memory will be required.

![Flowchart of the key-point tracking process.](image)
A second problem to solve is the way to store the results of the tracking process. Since the process proceeds forward with the time, the tracked trajectories are firstly stored in the following format:

```plaintext
frame number
  point number  x1 y1  x2 y2  x3 y3
  point number  x1 y1  x2 y2  x3 y3
  point number  ... 
  ... 
-1
frame number
  point number  x1 y1  x2 y2  x3 y3
  ... 
-1
frame number 
  ... 
-1
```

However, this format is not adequate for post processing such as the computation of velocities and acceleration, filtering, visualization and tracking key-points. The trajectory file is therefore converted into the following more suitable form:

```plaintext
point number
  frame number  x1 y1  x2 y2  x3 y3
  frame number  x1 y1  x2 y2  x3 y3
  frame number  ... 
  ... 
-1
point number
  frame number  x1 y1  x2 y2  x3 y3
  ... 
-1
point number 
  ... 
-1
```

### 7.5 OPTIONS

Different options and parameters have to be set for the tracking process and for post processing. The tracking process parameters can be divided into four groups: the input data, the LSM parameters, the tracking parameters and the filtering parameters.

#### 7.5.1 Input Data

The required input data are the multi-image sequences, the orientation and calibration information of the cameras and, in case of surface tracking, the set of matched points for the chosen multi-image sets or at least for the first multi-image set.
7 SURFACE TRACKING

7.5.2 Least Squares Matching

The parameters regarding the least squares matching are the same used in the automatic matching process of section 4.2.3:

- \textit{used LSM parameters}: used affine transformation parameters \((a_0, a_1, a_2, b_0, b_1, b_2)\) and radiometric correction factor \((r s)\). Usually, all the parameters are used.
- \textit{patch size x,y}: image patch sizes in pixels.
- \textit{max sigma 0}: threshold of \(\hat{\sigma}_0\) (see eq. 4.10) to accept the matching result.
- \textit{max sigma x,y}: thresholds of \(\hat{\sigma}_a\) and \(\hat{\sigma}_b\) (see eq. 4.10) to accept the matching result.
- \textit{max x,y}: thresholds of the displacement of the result of the matching process from the starting position (resulting parameters \(a_0\) and \(b_0\)).
- \textit{epipolar constrain}: on/off.

As explained in section 4.2.3, some parameters have to be determined case by case. These are the thresholds for the quality control \((\text{max sigma 0, max sigma x,y, max x,y})\) and the image patch sizes for the matching process \((\text{patch size x,y})\). No rules or advises can be given regarding these parameters. They strongly depends on the texture in the image and on its format. Their ideal values have to be determined by manually testing LSM on the images. Some examples of parameter sets can be found in appendix C.

7.5.3 Tracking Process

The specific parameters for the tracking process are the following:

- \textit{patch size x,y}: image patch sizes in pixels for the cross-correlation process used to find an approximation before applying LSM (step 2 of algorithm of section 7.1.1).
- \textit{first search size x,y}: sizes of the search region in the images of the next time step at the first tracking frame (figure 7.3, at frame \(i+1\) and \(\text{box=LARGE}\) in flowchart of figure 7.4).
- \textit{search size x,y}: sizes of the search region in the images of the next time step (figure 7.3, at frame \(i+2\) and \(\text{box=small}\) in flowchart of figure 7.4).
- \textit{large search size x,y}: sizes of the search region in the images of the next time step at the second try of the tracking process, in case of an unsuccessful match \((\text{box=large}\) in flowchart of figure 7.4).
- \textit{search step x,y}: pixel steps along the search path in the search region. It is not necessary to perform the cross correlation at every pixels along the search path; moreover, the speed of the entire process can be reduced.
- \textit{max diff}: threshold of percentage of differences of the \textit{spatial} and \textit{temporal} LSM to accept the tracking result.
- \textit{tessel size x,y}: sizes of the grid elements in the image for the computation of the local density for the introduction of new points into the tracking process.
7.6 CONSIDERATIONS

- density threshold: local density threshold in the grid elements to introduce new points in the tracking process. Depends on the tessel size and on the pixel shift of the surface matching process (e.g., if tessel size=10, pixel shift=2, then the total maximal number of points matched in one tessel is 25; the density thresholds can then be for example set as 20 which is equivalent to 80%).

Examples of parameter sets for the tracking process can be found in appendix C.

7.5.4 Filtering Process

7.5.4.1 Velocity and Acceleration Filter. The following parameters have to be set:

- min frame: minimal length of a trajectory (in frames). Shorter trajectories are removed. Usually, all the trajectories shorter than 3 frames should be removed.
- max velocity: threshold of maximal velocity allowed. Trajectories with larger velocities are truncated at that time.
- max acceleration: threshold of maximal acceleration allowed. Trajectories with larger accelerations are truncated at that time.

The two thresholds can usually be determined considering the average and the maximal values occurred in the unfiltered data.

7.5.4.2 Uniformity Filter. The following parameters have to be set:

- voxel size: size of cubic voxels [mm] used for dividing the object space into local regions.
- min density: minimal density in each voxel to be processed. If the number of trajectories in a voxel is too small, there is no sense to check the uniformity of the movement.
- min velocity: minimal actual velocity to check difference with the mean value. If the actual trajectory is stationary there is no sense to check the differences with the mean value.
- max difference magnitude: threshold for the maximal percentage of the difference of the magnitudes of the velocity vector and the local mean value. The trajectories exceeding the thresholds are truncated at that time.
- max angle: threshold for the maximal angle between the actual velocity vector and the local mean velocity vector [deg]. The trajectories with larger values are truncated at that time.

7.6 CONSIDERATIONS

The presented surface tracking process is flexible and can be used under different conditions for different applications. In the next section it is applied to capture the motion of humans without using markers (D’Apuzzo, 2001a). The process of tracking was also applied successfully in two dimensional mode for tracking facial expressions (a short description of the results is given by D’Apuzzo et al. (2000) or in section 7.8).
and for special applications (D’Apuzzo and Willneff, 2001).
The discussion of the results achieved by the tracking process has only qualitative
nature in this work. A quantitative analysis was not possible because reference data
for the comparison of the results was missing. For future studies of this work, it is
strongly recommended to acquire test sequences both with the multi-image acquisi-
tion system and with a precise optical motion capture system available commercially
(see section 1.4). It would be of great interest to compare the accurate motion capture
measurements with the results achieved by the method proposed in this work as 3–D
trajectories of key-points.
A second limitation of the description of the results achieved by the proposed method
is the very small number of tested datasets. In fact, only two multi-image sequences
where used in this work for testing purposes. Anyhow, it was possible to show the
flexibility of the proposed method, i.e., the possibility to use different image acquisition
systems (two, three cameras) recording the human body entirely or partially and
resulting in different types of 3–D data.

7.7 APPLICATION: FULL BODY MOTION CAPTURE

In this section, the proposed tracking process is applied to extract, without using mark-
ers, full body 3–D motion information from multi-image video sequences acquired
with synchronized CCD cameras. This work was part of a project aimed at develop-
ing an highly automated system to model most realistically human bodies from video
sequences (Fua et al., 1998; D’Apuzzo and Plänkers, 1999; Plänkers et al., 1999;
D’Apuzzo et al., 1999, 2000; D’Apuzzo, 2001a; Fua et al., 2002).
Two different image sequences are processed. Their characteristics and the achieved
results are presented in the next sections. The two multi-image sequences are acquired
with different camera systems. The former represent a person walking parallel to the
camera system and will be called walking sequence. The latter represent a more com-
plex movement where only the upper part of the body is acquired and will be called
baseball sequence.

7.7.1 Image Acquisition System

Two different acquisition system are used for the two different sequences. For the
walking sequence, two synchronized (interlaced) CCD cameras are used. A sequence
of image pairs is acquired with a frame grabber and the images are stored with 768x576
pixels resolution at 8 bit quantization. Since the CCD cameras are interlaced (i.e., a
full frame is split into two fields which are recorded and read out consecutively), only
the odd lines of the images are processed, avoiding the saw pattern effect (see section
2.2.2). This results in the reduction of the image format in the vertical direction by
50 percent (i.e., the effective format of the images is 768x288 pixels).
Figure 7.17, shows some frames of the walking sequence. As can be seen in the figure,
no markers are signalized on the person and no special conditions are required. The
person wears indeed normal clothes and in the background even other persons are
moving.
The baseball multi-image sequence is acquired by three synchronized progressive
scan CCD cameras in a triangular arrangement. A sequence of triplets is stored with
640x480 pixels resolution at 8 bit quantization (bottom, left, right).
7.7 APPLICATION: FULL BODY MOTION CAPTURE

Figure 7.18 shows some frames of the sequence; the person performs a complex movement of the upper part of the body (resembling vaguely a baseball pitch). In order to increase the contrast, the images of this sequence are enhanced using Wallis filtering (Wallis, 1976). Figure 7.19 shows a frame of the sequence before (left) and after (right) Wallis filtering.

7.7.2 Surface Measurement

The surface of the visible parts of the human body are measured every second time step of the sequence in the way explained in chapters 4 and 5. The amount of multi-image sets to be processed is very large, therefore, the fully automatic mode is used for the definition of the seed points (section 4.2.1). The manual intervention of the operator is reduced to the definition of the contour of the region to measure. This procedure
could be skipped, measuring the entire images. However, the time required to process
the image-sequence will increase.

Fig. 7.20  Walking sequence. Left: image pair. Center: matched points. Right: measured
3–D point cloud.

Fig. 7.21  Matching process on the baseball sequence. Top: image triplet. Bottom: matched
points.

The result of the surface measurement process are 3–D point clouds each two time
steps. This data will then be used in the successive surface tracking process. Figure
7.20 shows an example of the matching results achieved on an image pair of the
walking sequence and the computed 3–D point cloud. Figure 7.21 shows a triplet
of the baseball sequence with the matched points and figure 7.22 the computed 3–D
point cloud.

Fig. 7.22  Baseball sequence: Measured 3–D point cloud.
Table 7.1 summarizes some data regarding the two sequences and the matching results and table 7.2 give some hints about the achieved precision of the measurement. Since the images of the walking sequence are acquired by only two cameras, the theoretical precision of the computed 3–D point clouds cannot be derived.

Table 7.1 Data regarding sequences and surface matching. Image format [pixel], number of used images, number of processed frames, value of pixel shift for the matching process [pixel], average number of matched points.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>image format</th>
<th>images</th>
<th>frames</th>
<th>pixel shift</th>
<th>matched points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>768x288</td>
<td>2</td>
<td>55</td>
<td>2</td>
<td>5804</td>
</tr>
<tr>
<td>Baseball</td>
<td>640x480</td>
<td>3</td>
<td>71</td>
<td>2</td>
<td>10052</td>
</tr>
</tbody>
</table>

Table 7.2 Theoretical precision (Helmert’s point error according to eq. 6.1) of the measured point clouds (first frame of the sequence). Total number of matched points, average theoretical precision $\bar{\sigma}$ [mm], standard deviation [mm] and maximal value [mm]; same values ignoring points with $\sigma > 4.0$ mm; number of ignored points.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>points</th>
<th>$\bar{\sigma}$</th>
<th>std.dev.</th>
<th>max $\sigma$</th>
<th>$\bar{\sigma}_{&lt;4.0}$</th>
<th>std.dev.</th>
<th>max $\sigma$</th>
<th>- pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseball</td>
<td>12489</td>
<td>2.6</td>
<td>1.9</td>
<td>10.0</td>
<td>1.9</td>
<td>1.0</td>
<td>4.0</td>
<td>2387</td>
</tr>
</tbody>
</table>

The results achieved on full body are less accurate and more noisy than the results presented for the face measurement application (section 5.4). The reasons are the reduced resolution, because the entire human body has to be imaged and the texture, because the projection of an artificial texture is not allowed by the tracking process. The clothes worn by the person in the walking sequence make the surface measurement task even more difficult. However, the goals of the entire process have in this case more qualitative than quantitative nature. Only the approximative extraction of 3–D movements performed by the person is required.

7.7.3 Tracking Process

The completely automatic surface tracking process applied to the walking sequence provides totally about 32’000 trajectories for 55 frames. The mean number of trajectories per time step is about 6’000 and the average length of the trajectories is 10 frames. Due to the poor natural texture (mostly on the legs) and self-occlusion the surface tracking process does not produce a dense set of long trajectories. Figure 7.23 shows the tracked trajectories (displayed as displacement to the previous frame) for the frame number 39.

The results of the surface tracking process applied to the baseball sequence are about 120’000 trajectories during 71 frames, with an average length of 9 frames and an average of 10’000 trajectories per frame. The larger format of the images of this sequence compared to the previous, is the reason of the larger number of trajectories per frame. The more complex movement and the poorer natural texture are the main factors for the shorter average length of the computed trajectories. Figure 7.24 shows the tracked trajectories for the frame number 115.
7 SURFACE TRACKING

Fig. 7.23  Walking sequence. Surface tracking results (frame 39).

Fig. 7.24  Baseball sequence. Surface tracking results (frame 115).

Table 7.3 summarizes some data regarding the results of the tracking process on the two sequences.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>frames</th>
<th>trajectories</th>
<th>per frame</th>
<th>length</th>
<th>after filtering</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>55</td>
<td>32008</td>
<td>5804</td>
<td>9.8</td>
<td>22908</td>
<td>12.9</td>
</tr>
<tr>
<td>Baseball</td>
<td>71</td>
<td>121292</td>
<td>10052</td>
<td>8.9</td>
<td>70063</td>
<td>14.7</td>
</tr>
</tbody>
</table>

7.7.4 Key-point Tracking

The process of key-points tracking (see section 7.3) is required to extract general 3-D information of the performed movements of both sequences. The key-points are therefore placed in such a way that their trajectories can describe the walking process for the walking sequence and the complex upper body movement performed in the baseball sequence. For the walking sequence, a small set of key-points is placed near the joints: feet, knees, hips, hands, elbows, shoulders, neck, head and bust (see figure 7.25).

For the baseball sequence, a small set of key-points is placed near the joints: hands, elbows, shoulders, sides of neck; three are placed on the bottom of the upper body part; three on the face (nose and eyes) and four on the head (see figure 7.26).
The final result of the tracking process for the walking sequence is shown in figures 7.27 and 7.28. The size of the region defining the key-points is chosen small enough to assume them lying on the surface of the human body. The results achieved by the tracking process have therefore the same nature of the results of a conventional motion capture system, i.e., trajectories of key points on the body, which can then be used to reconstruct the motion performed by the person. Offset values of the key-points on the body could be defined, to get the joints of the human body.

**Fig. 7.25** Walking sequence. Key-points tracked on the human body.

**Fig. 7.26** Baseball sequence. Tracked key-points.

**Fig. 7.27** Tracking key-points: some frames of the walking sequence and frontal view of the position of the tracked key-points.
Because of the complexity of the performed movement of the baseball sequence, the size of the region defining the key-point is chosen in this case large enough to guarantee robustness. The key-points represent here an approximation for the joint trajectories. The final result of the tracking process is shown in figure 7.29 together with the corresponding frames. Two views of the 3–D trajectories of the key-points are displayed in figure 7.30 (note that the connections between key-points are intended only for a better understanding of the movement).

As can be seen, e.g., in the frame number 130 of figure 7.29, some key-points cannot be tracked during the entire sequence because of strong occlusions caused by the complex movement. In this case, the lost key-points have to be defined again in the following time steps. However, the new defined key-points will not be placed exactly at the same position. Anyhow, as explained above, the 3–D trajectories of the key-points serve in this case as an approximative description of the performed movement (as shown in figure 7.30) and therefore the results are acceptable.
7.8 APPLICATION: TRACKING IN 2-D

The proposed tracking process is a flexible tool and can also be used in 2-D mode. In this case, the sequence of a single camera, e.g., a camcorder, is processed. The use of a single image sequence can obviously not produce 3-D data but for some cases 3-D information is not required (D’Apuzzo and Willneff, 2001).

The Figure 7.31 shows a simple example of tracking facial expressions, where some points (white crosses on the first image) are tracked through the sequence acquired by a video camcorder. The single point tracking algorithm of section 7.1 was applied using as input the images of a single camera. The tracking was performed in this case only in the image space.

This example underlines the flexibility of the proposed tracking process, which can produce in this case a simple animation by tracking points on the face without using markers.
Chapter 8

Conclusions

8.1 SUMMARY OF THE RESULTS

A system and the methods for automated measurement and tracking of the surface of human body parts from multi-image sequences has been presented. The proposed method can be applied to measure either the surface of a static human body part or the moving surface of a dynamic event. In the latter case, the gained 3–D data can be of two different types: surface measurement of the interested part in form of a 3–D point cloud for each recorded time step or surface tracking in form of a vector field of 3–D trajectories (position, velocity and acceleration).

The entire process is composed of seven steps (four, in case of only surface measurement):

1. calibration of the system: establishment of the internal and external orientation of the cameras and the parameters modeling the lens distortion;
2. acquisition of multi-image sets and/or multi-image sequences;
3. matching process: establishment of correspondences in the multi-images (at least in the first set);
4. surface measurement: computation of the 3–D point cloud for each matched multi-image set;
5. surface tracking in the multi-image sequences;
6. establishment of a 3–D vectorfield of trajectories (position, velocity, acceleration);
7. tracking key-points in the vectorfield of trajectories.

To demonstrate the multiple functionality of the method, three different applications were presented: high accuracy measurement of human faces using five CCD cameras, measurement of a blood vessel branching casting fixed in a rotating frame using three CCD cameras and full body motion capture without markers using video sequences acquired by two or three synchronized CCD cameras.

8.1.1 Data Acquisition and Calibration

The accurate measurement of a human body part starts with an adequate acquisition of the required data, i.e., multi-images (multiple images acquired from different positions in the space describing the same scene) in case of static surface measurement.
or multi-image sequences (multi-images acquired during a time interval) in case of
dynamic surface measurement or surface tracking. In the latter case, the multi-images
have to be obviously acquired simultaneously. Nevertheless, also in case of static
surface measurement of human body parts, it is recommended to synchronize precisely
the multiple cameras because the human body cannot be considered as a static
object; in fact, a person always moves slightly unconsciously (e.g., breathing, muscle
contractions). The precision of the synchronization of the multiple imaging devices
plays therefore an essential role for the accuracy potential of the measurement.
Different systems can be used to acquire multi-image sequences with different levels
of quality depending on the used cameras. With decreasing accuracy potential, these
are: multiple synchronized machine vision progressive scan CCD cameras, multiple
synchronized machine vision interlaced CCD cameras or multiple digital video cam-
corders synchronized using a clapper. For surface measurement purposes without
tracking, only multi-images are required. In this case also the following systems can
be used for the acquisition (with decreasing accuracy potential): multiple digital still
cameras, a single digital still camera acquiring in different position or multiple digital
web cameras.
A precise synchronization of the multiple imaging units is possible only by the two
machine vision acquisition systems. Systems based on multiple digital still cameras
and systems using digital web cameras can be set up to acquire multi-images simulta-
neously. However, the cameras cannot be electronically synchronized to each other;
the maximum time delay which can occur between the acquisition of the different
images is therefore half a frame (i.e., 0.04 seconds). The same synchronization ac-
curacy can be achieved with systems using multiple video camcorders, aligning the
audio signals of the different sequences. The systems based on a single camera can
evidently not acquire simultaneously multi-images, in some cases the larger time de-
lay can however be acceptable (e.g., when high accuracy of the measurements is not
required and if the human body part can be immobilized).
The lens and optical system of the camera also play a role for the quality of the acquired
images. Machine vision lenses are preferable because of the smaller distortions. How-
ever, consumer market products (e.g., video camcorders, digital still cameras) can also
be used for measurement purposes without a great loss of accuracy (D’Apuzzo and
Maas, 1999), correcting the larger distortions with a thorough calibration procedure.
The tasks of the orientation and calibration processes are the establishment of the
position and orientation of the camera sensors in 3–D space as well as the parameters
describing the internal geometry of the imaging device and some parameters mod-
eling the distortion caused by the optical system. A thorough determination of all
the parameters is a requirement for the accurate measurement using photogrammetric
techniques. To calibrate the multi-image camera system, various methods can be used.
For simplicity, only two methods were applied in this work: the reference bar method
and the reference field method. In the first case, a reference bar with two retroreflec-
tive target points is moved through the object space while the camera system acquires
multi-image sets. The multiple camera system can then be calibrated by self cali-
brating bundle adjustment with the only required information of the known distance
between the two points on the bar and their automatically measured image coordinates
along the sequence. The main advantage of the method is the only requirement of the
reference bar, the disadvantage of the method is a thorough processing of the data to
calibrate the system. The reference field calibration methods use instead a 3–D field
8.1 SUMMARY OF THE RESULTS

with signalized points whose coordinates in space are known. The bundle calibration method is in this case used and full automation can be achieved using coded target points. A graphical user interface for the latter calibration procedure (code recognition and calibration) was implemented to facilitate the user and to reduce dramatically the time spent on this task. In fact, the most time-expensive problem that has to be solved for the implemented calibration method is the determination of the approximate external orientation; it is especially difficult to assume the correct values for the direction angles of the camera. An effective and fast solution to solve this problem is implemented in a smart way in the graphical user interface: the external orientation of the camera is indeed modified by the user interactively translating, rotating and moving in and out the backprojected 3-D coordinate of the control points (and not the camera itself), without taking care about the external orientation of the camera (position and direction). Using the implemented graphical user interface, the time required to calibrate an image decreases to a couple of minutes (without having any information about the external orientation of the camera). The proposed solution is intuitive and can be used without any difficulties by non-expert user too (Schneeberger et al., 2002).

8.1.2 Matching Process and Surface Measurement

The goal of the implemented automatic matching process is the determination of a dense set of corresponding points in the multi-images covering the entire interested surface part. Moreover, the defined matching strategy was developed especially to reduce to a minimum the required processing time.

The automatic matching process is based on a stereo matcher which determines the position of corresponding points in an image pair. In case of multi-images, the stereo matcher is applied independently to image pairs using the same template point in an unique template image. The results are correspondences in multi-images. Evidently this method is less robust than a complete multi-image matching process taking into account all the images in a single step (Gruen and Baltsavias, 1988; Baltsavias, 1991). Anyhow, the results achieved by the simpler implemented multi-image matching process were enough accurate and no further time was spent for the investigation and implementation of more complex and accurate matching algorithms.

The stereo matcher is built on the adaptive least squares method (Gruen, 1985) with the additional geometrical constraint of the matched point to lie on the epipolar line. Different methods could be used to integrate the epipolar geometry in the least squares matching process. Since for this work, the processing time has to be limited to a minimum, the geometrical constraint is expressed as a single additional observation describing the fact that the position in the search image has to move along the epipolar line.

The quality of the single matching result is evaluated using different indicators as the resulted a posteriori standard deviation of the least squares adjustment, the resulting standard deviation in x and y directions, the displacement from the start position in x and y direction and the distance to the epipolar line.

The developed strategy for the automatic matching process consists of the definition of few seed points and, starting from them, the establishment of a dense and robust set of corresponding points covering the entire interested region. The strategy is entirely developed regarding to the smooth characteristic of the surface of the human body.
However, the surface can present some discontinuities that can be treated separating the different continuous regions by placing the seed points in the correct positions. The same applies in case of overlapping parts. The seed points can be generated in three different modes: fully automatically, semi-automatically defining them only in one image or defined manually in each image. For static surface measurement using images with good texture, the semi-automatic mode is the preferable and faster than the others. On the other hand, the full automatic mode is adequate for the processing of long sequences, because of the large number of multi-image sets. In special cases where the automatic modes could fail as, e.g. with images with a fine repeated pattern, the time-expensive manual mode has to be used. Once the seed points are defined, the image is divided into polygonal regions according to which of the seed points is closest. Starting from the seed points, the automatic matching process produces then a dense set of corresponding points in each polygonal region by sequential horizontal and vertical shifts. The time required to cover the entire image depends strongly on the distribution, the location and the number of the defined seed points. The processing time decreases by incrementing the number of seed points. However, the definition of a larger number of seed points requires more time. An ideal compromise has to be chosen. A very simple method to reduce additionally the processing time is the definition of a region of interest in the template image. It can be given as a sequence of points or defined interactively in the implemented graphical user interface. Since the automatic matching process has to be used for different purposes (e.g., matching corresponding points in multi-images, target point measurement, tracking process), various options and parameters have to be set. A relevant option that can be chosen is whether or not to use the epipolar constraint. The complete matching process (definition of seed points and automatic matching) can in fact be performed without orientation and calibration information if desired. This functionality is useful, for example, when the orientation is not accurate enough or unknown. In these special cases, the least squares matching algorithm is not geometrically constrained. Obviously, the robustness of the results of the process decreases. However, the quality of the set of matched points may be satisfactory. Problems that could not be solved in this work and that still need to be investigated (see suggestions for further work, section 8.2.2) reside in the difficulty to establish image correspondences in dark areas such as eyebrows or areas with lack of texture. The hairs and beard still pose unsolved problems too. The automatic matching process results in fact in these areas very noisy. A graphical user interface for the automatic matching process was implemented to ease the user and to decrease the total time required to perform the measurement of human body parts. All the required processes can be performed in the GUI: the definition of seed points can be achieved in the different modes (manual, semi-automated, full semi-automated, full automatic) and a ROI can be selected manually as a contour. Additionally, the automatic matching process can be visualized in real-time, allowing an analysis of the implemented search strategy. It also helps the user to get an idea of where the more problematic regions for the matching process are, gaining know-how about where to put the seed points and what their ideal number is. The regularization of grid, the neighborhood filter and the computation of the 3-D coordinates of the matched points are implemented as post-processing procedures.
8.1 SUMMARY OF THE RESULTS

An important tool is also the implemented 3-D point cloud viewer/editor. Large amounts of points can be displayed with color information (greyscale or RGB) and full navigation (translation, rotation, zoom) is possible. Additionally, stereo visualisation (with red/green glasses) and some basic editing functions are also implemented. Single points can be selected to get information about the point number, the 3-D coordinates and, if available, the color information (as RGB values). Multiple points can be selected in rectangular areas or in a (clockwise) hand drawn region. The selected points can then be deleted or stored as separate data set. This offer a simple and very fast opportunity to remove outliers or unuseful points from the data (e.g., background points or points not interesting for the modeling purposes).

Basic modeling procedures were implemented only to achieve a better visualisation of the surface measurement results. Therefore, detailed modeling aspects are not analyzed nor described in this work. For better qualitative modeling results, commercial software can be used.

A triangulated surface is generated from the 3-D point cloud by 2 1/2-D Delauney triangulation method (IGP software DTMZ). The disadvantage of this method is the impossibility to process fully 3-D data. For this reason, only landscape-like surface models can be generated (e.g., face masks). For texture mapping, basic modeling tools were implemented for the generation of VRML files with texture image coordinates for each vertex of the model.

The presented surface measurement process is flexible and can be applied with different acquisition setups and for different purposes. In fact, it was successfully employed for face modeling using images acquired by five CCD cameras (D’Apuzzo, 2002b), for human body modeling using three CCD cameras acquiring video sequences (D’Apuzzo et al., 2000), for human body modeling using uncalibrated images acquired by a digital camera (Remondino, 2003) and for the modeling of blood vessels, where three CCD cameras acquired in eight positions twenty-four images (D’Apuzzo, 2001b). Moreover, the surface measurement procedure constitutes also part of the surface tracking process. In this case, the surface measurement process can be performed for each multi-image of the acquired sequence.

The advantage of the presented method over the extensively employed methods to measure the surface of the human body such as laser scanning and coded light based triangulation approaches, is the acquisition of the source data in a fraction of second, allowing a surface measurement with higher accuracy and even the possibility to measure dynamic events like movements or changes of the surface.

8.1.3 Tracking Process

The main idea is to track corresponding points in the multi-images through the sequence and compute their 3-D trajectories. The results of the tracking process are the coordinates of a point in the multi-images through the sequence, thus the 3-D trajectory is determined by computing the 3-D coordinates of the point at each time step by forward ray intersection. Velocities and accelerations are also computed. The process is based on least squares matching techniques: the spatial correspondences between the images of the different views and the temporal correspondences between subsequent frames are determined using the least squares matching algorithm. The same problems and possible improvements for the matching process described in sections 8.1.2 and 8.2.2 apply also for the implemented tracking process. Additionally,
the impossibility to project an artificial texture onto the surface to track, increases the
noise in the set of matched and tracked points. The reason is the insufficient texture
of the human skin.
The proposed process can be used to track well defined points on the human body
surface. Trajectories of single points are however not sufficient to understand and
record the motion and movement of a human or the changes of the surface of human
body parts. For this reason the tracking process was extended to track simultaneously
a dense set of points belonging to a common surface. In this case, the result of the
tracking process can be considered as a vector field of trajectories. The process of
tracking entire surface parts and not only good signalized single points, can however
produce false trajectories especially in case of poor texture of the surface to be mea-
ured. Nevertheless, these can easily be recognized because they do not follow the
common movement of the majority. Filters for checking the local uniformity of the
movement in the vector field of trajectories were therefore implemented.

To solve additional problems caused by occlusions, lack of texture, loss of tracked
points and appearance of new points, the concept of key-points was introduced. The
key-points are 3–D regions defined in the vectorfield of trajectories, whose size can
vary and whose position is defined by its center of gravity. They are interactively de-
defined in a graphical user interface and tracked in a simple way: the position in the next
time step is computed as the mean value of the displacements of all the trajectories
contained in side the 3–D region. The size of the region defining the key-point has an
important role: if a small size is chosen, the key-point can be considered with a good
approximation to lie on the tracked surface, representing in this case fictitious mark-
ers on the human body. If, on the other hand, larger sizes are chosen, the key-point
represents a more approximative position of, e.g., joints of the human body.

Graphical user interfaces were developed and implemented for the dynamic visual-
ization of the tracking results, two-dimensionally together with the image sequences
as well as three-dimensionally displaying the computed 3–D trajectories in different
view modes. All the functionalities for the definition, tracking and visualization of
key-points are implemented in the latter graphical user interface.

The presented surface tracking process is flexible and can be used under different
conditions for different applications. The work was part of a project aimed at de-
veloping an highly automated system to model most realistically human bodies from
video sequences (Fua et al., 1998; D’Apuzzo and Plöckers, 1999; Plöckers et al.,
1999; D’Apuzzo et al., 1999, 2000; Fua et al., 2002). The process of tracking was
also applied successfully in two dimensional mode for tracking of facial expressions
(brief description in section 7.8) and for special applications (D’Apuzzo and Willneff,
2001).

8.2 SUGGESTIONS FOR FURTHER RESEARCH

8.2.1 Data Acquisition and Calibration

Concerning the image acquisition systems, further research can be directed toward
two different interesting goals: high quality or low cost. For an high quality acqui-
sition system the number of cameras has to be extended, allowing a more complete
imaging of the interested human body part (from the front, sideways and form the
back). Machine vision progressive scan cameras are the best choice for multi-image
acquisition systems. The larger amount of data (in case of six, nine or even twelve cameras) has to be considered. Probably multiple frame grabbers or multiple acquisition systems have to be used and synchronized together to solve the task.

In the other direction, for demonstration and educational purposes, it would be very interesting to develop a (very) cheap multi-image acquisition system using, e.g., web cameras connected to a portable PC. A precise synchronization of the cameras is not possible and therefore high accuracy of the measurement cannot be achieved. Still, the great demonstration potential of a portable measurement system would be very attractive.

Concerning the two calibration methods applied in this work (reference field and reference bar), full automation is still a field for further investigation and implementation. For the reference field calibration methods, it could be achieved using in a first step a reduced number of coded targets for the determination of approximate values for the external orientation and continue then with the full calibration procedure. For the reference bar method, the achievement of full automation is challenging. The automatic recognition and measurement of the target points on the bar in all the acquired images is not trivial in case of an extension of the number of used cameras. Additionally, an automation in the calibration procedure has also to be implemented, because the used software package (SGAP of IGP) is very user unfriendly and long time is actually spent for the preparation of the required input files. Moreover, more steps are required to achieve the final calibration of the camera system.

For the calibration and orientation of multi-image acquisition systems, other methods, such as the point field method or the Australis™ approach, could be investigated.

8.2.2 Matching Process and Surface Measurement

The implementation of a fast multi-image matching process in an unique least squares process could be an interesting task for further research. The main problem to solve is the definition of a robust and fast method to remove the irrelevant images from the set of images used in the process. Since the cameras are usually placed concentric to the human body part, it is in fact probable that regions of the interested part are not imaged by all the cameras. A smart strategy has therefore to be defined to select the relevant images before entering the least squares procedure.

The use of generic full body models or generic surface models of the interested parts could be integrated in the matching strategy to limit in the images the regions to process. The implementation of an automatic silhouette extraction could also be useful for this task.

A second interesting aspect that could be further investigated is the integration of neighborhood conditions. In this work, a neighborhood filter was implemented to detect false matching results. Furthermore, it could be interesting to use the smooth characteristic of the human body surface to integrate the neighborhood in the least squares process allowing the matching of multiple neighboring points in an unique process (Gruen and Zhang, 2002), making the results more robust and increasing the success rate in the regions with lack of texture. Moreover, new techniques for evaluating the matching results have to be investigated.

It would also be very interesting to compare the results achieved by the proposed method with commercially available measurement systems like laser scanners and coded light approaches. Methods for the comparison of the different data have to
be investigated. Moreover, the acquisition of the required data for the three different surface measurement systems should be performed simultaneously.

8.2.3 Tracking Process

The possible improvements of the result of the tracking process mainly depend on the matching process. The same suggestions described in the previous section can therefore be applied also for the tracking process.

What could be additionally investigated is the implementation of the independently used least squares matching processes (temporal LSM and spatial LSM) in an unique least squares procedure. Moreover, smart strategies could be integrated in the process to adaptively change parameters of the least squares matching depending on the results achieved.

In case of full body tracking or tracking of articulations, to increase the level of automation of the entire process, an automatic extraction of the silhouette of the human body or the contour of human body parts has to be implemented. A student work (Nef, 2001) investigated methods that could be used to solve this problem and presented preliminary implementations. Lot of work has however still to be done in this field to solve problems as shadow and noise. A thorough illumination of the subject and laboratory conditions, such as an uniform background, could limit the shadow effects as well as the noise caused by other persons or moving objects present in the scene.

Additional cues that could be extracted from the vectorfield of trajectories resulting from the improved tracking process are the extraction of rotation axes and angles of the different articulations. The definition of the key-points could be extended for this purpose.

For the extraction of the more elaborated data, it is strongly recommended to increase the number of the used cameras (to six, nine or twelve), so that also very complex movements can be acquired from the front as well as sideways and from the back. This will pose additional problems for the tracking process which has also to be generalized for the use of a variable number of cameras.

For future studies of this work we also recommend a comparison of the results with reference data. Test sequences should be acquired simultaneously both with the multi-image acquisition system and with a precise optical motion capture system available commercially. It would be of great interest to compare the accurate motion capture measurements with the results achieved by the newly developed method.
A.1 MOTIVATIONS

The development and the implementation of a graphical user interface for the surface measurement and surface tracking processes have different intentions:

- **user friendliness**: to make everything easier for the user;
- **preprocessing time**: to reduce the time required for the preprocessing of the data;
- **visualisation**: for visualisation purposes (matched points, point cloud, trajectories);
- **analysis**: to analyze the defined processes in real-time.

A.2 OPERATING SYSTEM AND IMPLEMENTED GUI

A.2.1 Operating System

All the software was developed under Unix/Linux environment, from the calibration procedure through the matching process and the 3–D point cloud editing, till the visualisation of the tracking results and the definition of the key points. The code was implemented in ANSI C for a general portability.

A user friendly graphical interface (GUI) was implemented using the GTK/GDK libraries. All the software can be compiled for Linux and Unix operating systems. Nowadays, GDK/GTK libraries exists also for Microsoft Windows operating system, it is therefore possible to compile the software also under this environment by performing some changes in the code.

A.2.2 GUI for the Calibration Process

A graphical user interface for the calibration procedures (code recognition and calibration) was implemented to facilitate the user and to reduce the time spent on these
tasks. Figure A.1 shows the implemented graphical user interface and the figure A.2 describes the functions of the different menus and tool buttons.

**Fig. A.1** Graphical user interface of the code recognition and calibration processes.

**Fig. A.2** Menu line and tool buttons of the graphical user interface of the code recognition and calibration processes.

**Fig. A.3** Parameters for the automatic recognition and measurement of the coded targets in the image.
A.2 OPERATING SYSTEM AND IMPLEMENTED GUI

Fig. A.4 Result of the code recognition and target measurement processes.

The calibration process starts with the automatic recognition and measurement of the coded targets in the image (Niederoest, 1996). Figure A.3 shows the parameters to be set for the process and the figure A.4 shows an example. For the details about the process and the parameters, the reader is referred to the publication (Niederoest, 1996).

Fig. A.5 Smart change of the orientation of the camera.
The most time expensive problem that the user has to solve for the calibration process applied in this work, is the determination of the approximative external orientation of the camera: position of projection center \((x_0, y_0, z_0)\) and direction \((\omega, \varphi, \kappa)\). What cause the biggest troubles, is the difficulty to assume the correct values for the direction angles. An effective and fast solution to find an approximation of the external orientation is implemented in a smart way in the graphical user interface. The external orientation of the camera is indeed changed by the user interactively by translating, rotating and moving in and out the backprojected 3–D coordinates of the control points (and not the camera itself) (see figure A.5), without taking care about the values of the position and direction of the camera (for the details about how the movement of the camera is computed, see section A.2.5).

If wished, the user can also change directly the position \((x_0, y_0, z_0)\) and the direction \((\omega, \varphi, \kappa)\) of the camera (see figure A.6).
Before executing the calibration process of section 3.3, an approximation of the correct orientation of the camera has to be determined. This is the case when the control points backprojected onto the image have more or less the same disposition as the measured image coordinates (see figure A.7).

**Fig. A.7** Starting the calibration process (the backprojected control points have more or less the same position as the measured image coordinates) and used parameters for the calibration.

Using the implemented graphical user interface, the process of calibrating an image takes a couple of minutes (even without having any information about the orientation of the camera).

The geometrical correction of the images is implemented as additional feature. An undistorted image can be generated by using the distortion parameters \((k_1, k_2, k_3, p_1, p_2, sc, sh)\) determined during the calibration process. The flowchart of figure A.8 describes the process in detail.

**Fig. A.8** Flowchart: correcting the distortion of an image.
A new image with the same resolution is generated. Each pixel in the new undistorted image \((i, j) = (\tilde{u}, \tilde{v})\) is firstly transformed from image coordinate to metric coordinates \((\tilde{x}, \tilde{y})\) (according to equation 3.6). The distortion terms are then added to the obtained metric image coordinates resulting in \((x, y) = (\tilde{x}, \tilde{y}) + (\tilde{d}x, \tilde{d}y)\) (according to equations 3.4 and 3.5). These are then transformed into image coordinates \((u, v)\) (according to equation 3.7), which represent the distorted image coordinates. The gray value of the new image at the pixel \((i, j)\) is therefore computed as bilinear resampling (according to equation 4.7) of the original image at the position \((u, v)\); if \((u, v)\) is outside the image limits, the new pixel is set black.

### A.2.3 GUI for the Automatic Matching Process

The graphical user interface for the automatic matching process is the core program for the surface measurement process. All the processes described in section 4.2 can be performed in the GUI. As pre-processing, Wallis filter contrast enhancement can be applied on the images before starting the matching process. The definition of seed points can be performed in the different modes (manual, semi-automated and fully automatic) and a ROI can be selected manually as a contour.

![Fig. A.9 Menu line and tool buttons of the graphical user interface for the automatic matching process.](image)

The automatic matching process can be visualized in real time. This functionality plays an essential role for the analysis of the implemented search strategy. It also helps the user to get an idea of where the more problematic regions for the matching process are, gaining know-how about where to put the seed points and what their ideal number is.

The regularization of grid (see section 4.3.1), the neighborhood filter (see section 4.3.2) and the computation of the 3-D coordinates of the matched points (see section 5.1) are implemented as post-processing procedures. Figure A.10 and figure A.11 show three moments of the processing: the definition of seed points, the real time visualization of the matching process and the end result with the computed 3-D point cloud. While the figure A.9 explains the functions of the menu line and the various tool buttons.
A.2 OPERATING SYSTEM AND IMPLEMENTED GUI

Fig. A.10 Graphical user interface of the matching process. Left: definition of seed point. Right: real time visualisation of the matching process.

Fig. A.11 Graphical user interface of the matching process after finishing the matching and computed 3–D point cloud.
A.2.4 Matching Result Viewer

The graphical user interface of the automatic matching process provides also a visualization of the results. However, an additional tool was implemented to allow a more accurate analysis of the data. The set of matched points can be displayed alone or over the used images, zooming and selection functions are implemented. Figure A.12 shows an example while the figure A.13 explains the functions of the tool buttons.

![Fig. A.12 Matching result viewer.](image)

![Fig. A.13 Tool buttons of the matching results viewer.](image)
The additional feature of this tool is the possibility to visualize the matched points with different colors according to the resulting values of the parameters of the affine transformation \((a_1, a_2, b_1, b_2)\). This functionality is interesting to analyze the local uniformity of the transformed patches in the search images.

### A.2.5 Point Cloud Viewer/Editor

The implemented 3–D point cloud viewer/editor is completely based on the collinearity equation (equation 5.1), i.e., the points displayed on the graphical user interface are the result of the backprojection of the 3–D point cloud onto a virtual image plane. The actual external orientation of the virtual camera as position \(X_0, Y_0, Z_0\) and rotation angles \(\omega, \varphi, \kappa\) and a very large camera constant \(f\) (to avoid large perspective effects) are displayed on the bottom text bar of the viewer (the figure A.14). The color information (greyscale or RGB) can also be displayed (figure A.14 right).

![Graphical user interface of the 3–D point cloud viewer/editor. Left: simple view. Right: color information of the 3–D point cloud is also displayed.](image)

The figure A.15 explains the functions of the tool buttons.

![Toolbar of the 3–D point cloud viewer/editor.](image)

Figure A.16 describes how the initial position of the virtual camera (that can be also recalled with the \textit{go home} button of the toolbar) is determined. It is oriented along the \(z\)-axis (i.e., all rotation angles are equal to 0), pointing to the center of the 3–D point cloud. The distance of the virtual camera to the 3–D point cloud is chosen very large to
A GRAPHICAL USER INTERFACE

avoid strong perspective effects (a defined factor \( PERSP \) multiplied with the extension of the point cloud in \( z \) direction \( \Delta z \)); the camera constant \( f \) is then determined so that all the 3–D points can be displayed in the maximal size of the viewer window.

![Diagram](image)

**Fig. A.16** Definition of the initial position of the virtual camera for the 3–D point viewer.

The navigation possibilities are: pan in vertical and horizontal direction, zooming in and out and rotate around the center of the point cloud. The figure A.17 explains how the virtual camera position and orientation change, while the 3–D point cloud stays fix.

![Diagram](image)

**Fig. A.17** Navigation in the 3–D point viewer: zoom in and out, panning horizontally and vertically, rotate 1 and 2. The virtual camera changes position and orientation, while the 3–D point cloud stays fixed.

For **zooming in and out**, the direction of the virtual camera remains unchanged, while its position moves along the view direction according to:

\[
\begin{align*}
\Delta X_0 &= r_{13} \cdot \Delta \text{zoom} \\
\Delta Y_0 &= r_{23} \cdot \Delta \text{zoom} \\
\Delta Z_0 &= r_{33} \cdot \Delta \text{zoom}
\end{align*}
\]  

(A.1)
where
\[ \Delta X_0, \Delta Y_0, \Delta Z_0 \] position change of the projection center of the virtual camera,
\[ r_{ij} \] elements of the rotation matrix \( \mathbf{R} \) defining the orientation of the virtual camera,
\[ \Delta \text{zoom} \] displacement along the view direction of the virtual camera defined interactively by the user.

For **panning horizontally and/or vertically**, the direction of the virtual camera also remains unchanged and the position of the camera moves in the image plane according to:
\[
\begin{align*}
\Delta X_0 &= r_{11} \cdot \Delta \text{hor} + r_{12} \cdot \Delta \text{vert} \\
\Delta Y_0 &= r_{21} \cdot \Delta \text{hor} + r_{22} \cdot \Delta \text{vert} \\
\Delta Z_0 &= r_{31} \cdot \Delta \text{hor} + r_{32} \cdot \Delta \text{vert}
\end{align*}
\] (A.2)

where
\[ \Delta X_0, \Delta Y_0, \Delta Z_0 \] position change of the projection center of the virtual camera,
\[ r_{ij} \] elements of the rotation matrix \( \mathbf{R} \) defining the orientation of the virtual camera,
\[ \Delta \text{hor}, \Delta \text{vert} \] horizontal and vertical displacements in the image plane of the virtual camera defined interactively by the user.

For the **rotations 1 and 2**, the virtual camera rotates around the center of the point cloud in two directions, the horizontal and vertical directions of the starting location of the virtual image plane (see figure A.17). The position of the projection center and the direction of the virtual camera are computed according to:
\[
\begin{align*}
\Delta X_0 &= 2 \cdot d \cdot (r_{11} \cdot \sin \frac{\alpha_1}{2} \cdot \cos \frac{\alpha_1}{2} - r_{13} \cdot \sin^2 \frac{\alpha_1}{2} - r_{12} \cdot \sin \frac{\alpha_2}{2} \cdot \cos \frac{\alpha_2}{2} - r_{13} \cdot \sin^2 \frac{\alpha_2}{2}) \\
\Delta Y_0 &= 2 \cdot d \cdot (r_{21} \cdot \sin \frac{\alpha_1}{2} \cdot \cos \frac{\alpha_1}{2} - r_{23} \cdot \sin^2 \frac{\alpha_1}{2} - r_{22} \cdot \sin \frac{\alpha_2}{2} \cdot \cos \frac{\alpha_2}{2} - r_{23} \cdot \sin^2 \frac{\alpha_2}{2}) \\
\Delta Z_0 &= 2 \cdot d \cdot (r_{31} \cdot \sin \frac{\alpha_1}{2} \cdot \cos \frac{\alpha_1}{2} - r_{33} \cdot \sin^2 \frac{\alpha_1}{2} - r_{32} \cdot \sin \frac{\alpha_2}{2} \cdot \cos \frac{\alpha_2}{2} - r_{33} \cdot \sin^2 \frac{\alpha_2}{2}) \\
r_{11}'' &= r_{11} \cdot \cos \alpha_1 - r_{13} \cdot \sin \alpha_1 \\
r_{12}'' &= r_{12} \cdot \cos \alpha_2 + r_{13}' \cdot \sin \alpha_2 \\
r_{13}'' &= -r_{12} \cdot \sin \alpha_2 + r_{13}' \cdot \cos \alpha_2 \\
r_{21}'' &= r_{21} \cdot \cos \alpha_1 - r_{23} \cdot \sin \alpha_1 \\
r_{22}'' &= r_{22} \cdot \cos \alpha_2 + r_{23}' \cdot \sin \alpha_2 \\
r_{23}'' &= -r_{22} \cdot \sin \alpha_2 + r_{23}' \cdot \cos \alpha_2 \\
r_{31}'' &= r_{31} \cdot \cos \alpha_1 - r_{33} \cdot \sin \alpha_1 \\
r_{32}'' &= r_{32} \cdot \sin \alpha_2 + r_{33}' \cdot \cos \alpha_2 \\
r_{33}'' &= -r_{32} \cdot \sin \alpha_2 + r_{33}' \cdot \cos \alpha_2 \\
\end{align*}
\] (A.3)
with:
\[
\begin{align*}
r_{13}' &= r_{11} \cdot \sin \alpha_1 + r_{13} \cdot \cos \alpha_1 \\
r_{23}' &= r_{21} \cdot \sin \alpha_1 + r_{23} \cdot \cos \alpha_1 \\
r_{33}' &= r_{31} \cdot \sin \alpha_1 + r_{33} \cdot \cos \alpha_1
\end{align*}
\]
where

\[ \Delta X_0, \Delta Y_0, \Delta Z_0 \] position change of the projection center of the virtual camera,

\[ r_{ij} \] elements of the rotation matrix \( R \) defining the orientation of the virtual camera before the rotation,

\[ r_{ij}'' \] elements of the rotation matrix \( R \) defining the orientation of the virtual camera after the rotation,

\[ d \] distance of the projection center of the virtual camera to the center of the 3–D point cloud,

\[ \alpha_1, \alpha_2 \] rotations angles in horizontal and vertical directions defined interactively by the user.

Stereo visualisation (with red/green glasses) is also implemented in the GUI (see figure A.19). Figure A.18 describe how it is achieved: two virtual cameras with a default parallax of \((\alpha_{\text{stereo}} = 0.1 \text{ [rad]})\) are defined according to equations A.3 with the two angles \( \alpha_1 = \alpha_{\text{stereo}}, \alpha_2 = 0 \). By need, the parallax can be interactively changed by the user. The points backprojected on the two different virtual cameras are drawn in red for the first camera and in green for the second camera (displaced with \( \alpha_{\text{stereo}} \)). Using red/green glasses, the 3–D point cloud can therefore be viewed in stereo.

![Diagram of stereo visualisation](image)

**Fig. A.18** Stereo visualisation is achieved by backprojecting the 3–D points onto two virtual cameras with red and green colors (green is for the right eye, red for the left).

Some basic editing functions were implemented in the viewer. Single points can be selected to get information about the point number, the 3–D coordinates and, if available, the color information (as RGB values). Multiple points can be selected in rectangular areas or in a (clockwise) hand drawn region. The selected points can then be deleted or stored as separate data set. This offer a simple but very fast opportunity to remove outliers or unuseful points from the data (e.g., background points or points not interesting for modeling purposes).

Figure A.20 describes the strategy for the selection of points inside a clockwise hand drawn area, which is given in the form of a sequence of points \((x_i, y_i)\). Firstly the contour is interpolated so that the displacement in y direction is always 1 pixel (or 0 in case of horizontal parts). For all the elements of the contour, is searched the nearest point on the right with the same y, if the contour is moving up and respectively for the nearest point on the left with the same y, if the contour is moving down.
A.2 OPERATING SYSTEM AND IMPLEMENTED GUI

Fig. A.19  Stereo visualisation: left normal view, right: stereo view (red and green points).

Fig. A.20  Hand drawn clockwise contour. The inside of the region is determined searching on the contour for the nearest point with the same y, on the right if the contour is going up and respectively on the left if it is going down; all the points on the horizontal line between the two found points are inside the region.

Fig. A.21  Hand drawn point selection (the selected points are drawn in grey). Left: selection, right: after turning the 3–D point cloud.
All the points lying on the segment between the two points are then defined inside the hand drawn region. At the end, an image mask defining the points inside the region (value 1 if the pixel is inside the hand drawn region and value 0 if it is outside) is obtained. For the selection of 3–D points, these are first backprojected onto the image. Their image coordinates are then compared with the image mask and the points having value 1 are selected. In this way, complex 3–D point sets can be selected easily. Figure A.21 shows an example of a hand drawn selection.

A.2.6 2–D Trajectories Viewer

The 2–D trajectory viewer is analogous to the matching result viewer described in section A.2.4.

Fig. A.22 Visualization of trajectories. Top: the trajectories are displayed as points on the actual frame; bottom: without the images.
Matched and tracked points can be visualized over the multi-image sequences or alone. The sequence can be browsed and played forward or backward with variable speed. Different display modes exist for the trajectories: they can be displayed as points on the actual frame (*actual points*), as complete trajectories (*all trajs*), as complete trajectories with crosses on the actual frame (*all trajs + actual crosses*) (these two last mode are suitable only for a small number of trajectories) or as points with tail (*trajs with tail*) with variable length. Figures A.22 and A.23 show four screen shots. In the first figure on the left the trajectories are displayed as points on the actual frame, on the top right without the images. In the latter, with a tail the length of one frame, left together with the images and right alone. The figure A.24 explains the functions of the tool buttons.
A.2.7 3–D Trajectories Viewer / Key-Points Editor

The 3–D trajectories viewer combines the functionalities of the 2–D trajectory (section A.2.6) and the 3–D point cloud (section A.2.5) viewers. The trajectories are in this case displayed in 3–D and a real time navigation (moving the virtual camera) is possible while playing the sequence.

The 3–D trajectories and the trajectories of the key-points can be displayed together in different modes as explained in the previous section: points in the actual frame, complete trajectories, complete trajectories with crosses on the actual frame or as trajectories with tails. Figure A.25 shows two different combined visualizations: on the left, some selected trajectories are displayed entirely while all the trajectories are displayed as points in the actual frame; on the right, the defined key-points connections and the trajectories of the tracked key-points are displayed entirely while all the trajectories are displayed as points in the actual frame. The figure A.26 explains the functions of the various tool buttons.
All the functionalities for the definition, tracking and visualization of key-points are implemented in the graphical user interface. The key-points are defined interactively in the vector field of trajectories. A key-point can be easily placed over an existing tracked point at any frame of the sequence, its position in space can then be manually adjusted by moving the key-points along the three axes (figure A.27 right). The size of the region to be tracked (defined as a cube) is also chosen interactively (see figure A.27 left).

For a nice visualization and a better understanding of the movement, connections between the tracked key-points can be defined interactively (note that they do not have any physical meaning). Figure A.28 shows an example of visualization of tracked key-points; their trajectories are displayed entirely and the connections are drawn for the actual frame, all the other trajectories are hidden. As can be seen in the figure, the connections make the understanding of the movement easier.
Fig. A.28  Key-point trajectory visualization, two views of the complete 3–D trajectories of the tracked key-points and visualization of the connections at the beginning of the sequence.
IEEE-1394 Camera System

B.1 MOTIVATIONS

A multi-image acquisition system based on five IEEE-1394 progressive scan CCD cameras was developed by myself at the Imaging Laboratory of the R&D Center of TOPCON Corporation in Tokyo, under the project "3-D Human Face Feature Extraction" (D’Apuzzo and Kochi, 2003).

The used cameras are provided with the IEEE-1394 port. Therefore no frame grabber is required for the acquisition of images. They can be read out accessing the IEEE-1394 port.

This appendix describes shortly the camera system and the implemented GUI for controlling the cameras and for acquiring the images.

B.2 MULTI-IMAGE ACQUISITION SYSTEM

The outline of the acquisition system is based upon the face measurement camera system described in section 5.4.

The cameras used in this case are progressive scan CCD cameras provided with a 1/3" sensor of 640x480 pixels and IEEE-1394 port. Through the IEEE-1394 interface, the computer communicates with the camera, allowing digital transmission of images as well as software control of some camera parameters. Figure B.1 shows the camera Dragonfly of Point Grey Research™ from the front and from the back.

Different frame rates can be set from 3.75 Hz up to 30 Hz. Other features, such as brightness, exposure, gain, shutter and white balance can be controlled through the IEEE-1394 port. The Dragonfly comes equipped with the ability to perform multiple camera synchronization. When more than one Dragonfly camera is present on the IEEE 1394 bus, the cameras automatically synchronize their acquisition time to within 20µs. This feature is particularly important for multi-image acquisition systems.

A c-mount lens with 16mm focal lens was chosen for the camera setup. In this way the complete system can be relatively compact in size (about 1 m wide). The lens are provided with variable iris and the focus can be fixed.
B IEEE-1394 CAMERA SYSTEM

Fig. B.1  Point Grey Research camera, model Dragonfly, with c-mount 16 mm lens. Left: frontal view, right: back view.

Figure B.2 shows the camera setup. The entire system is mounted on a tripod that can move vertically and rotate around its vertical axis. The five cameras can rotate around two axes. Additionally, for the fine tuning of the system, the two extern cameras (number 1 and 5) can be displaced sideways and to the front or back. The central camera (number 3) can be moved to the front and back too. Three cameras (number 1, 3 and 5) are mounted over sockets to have a difference in the heights.

Fig. B.2  Camera setup. The entire system can move up and down and turn around the tripod axis. All the cameras can be rotated around two axes. Additionally, for fine tuning, cameras 1 and 5 can be moved sideways and to the front or back. Camera 3 can also be moved to the front or back.

B.3 IMPLEMENTED GUI

The camera control, the multi-images display and the multi-image acquisition are implemented in the same graphical user interface of the 3-D face feature extraction software. In that way, the user can control all the required processes within an unique software. The images acquired by the five cameras can be viewed in real-time, as shown in figure B.3. Figure B.4 describes the functionalities of different buttons on the toolbar.
Fig. B.3 Graphical user interface of the 3–D face feature extraction software. Live camera display mode.

As described in the previous section, some features of the cameras can be controlled by software through the IEEE-1394 port. This was implemented in a way that the five cameras have exactly the same features. Figure B.5 shows the camera setting window where information (camera name, port number, node number) about the cameras on the IEEE-1394 bus is displayed and where the user can select the different features (exposure, shutter, gain, brightness).
B.4 CALIBRATION AND ORIENTATION

A reference field with signalized points was especially designed for the five camera acquisition system to allow a full automatic calibration procedure. The 3-D coordinates of the signalized points were measured photogrammetrically by PI-3000 system (Kochi et al., 2003) of TOPCON with a precision of 0.2 mm.

The calibration and orientation processes are integrated in the graphical user interface software and the unique task required for the user is to place the calibration field at the indicated position. The signalized points are then fully automatically recognized and measured in the different images by least squares matching. The establishment of the external and internal orientation as well as parameters modeling the lens distortion is performed by bundle adjustment. Figure B.6 shows the calibration field and the full automatic image coordinate measurement process as can be viewed in the graphical user interface.

Fig. B.5 IEEE-1395 camera information and camera settings.

Fig. B.6 Left: calibration field. Right: full automatic measurement of signalized point.
Appendix C

Parameter Sets

This appendix contains some examples of parameter sets for the matching and tracking processes.

C.1 MATCHING PROCESS

Table C.1 lists parameter sets of the matching process for some examples; the names refer to the following figures where are displayed the template images with the used seed points and the contour defining the ROI.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Max</th>
<th>Naoko</th>
<th>Naomi</th>
<th>Dummy</th>
<th>Walking</th>
<th>Baseball</th>
</tr>
</thead>
<tbody>
<tr>
<td>search shift</td>
<td>1/4</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>no search shift</td>
<td>1/8</td>
<td>1/4</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>bigger patch</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>close the gaps</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>used LSM par.</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
</tr>
<tr>
<td>use epipolar constr.</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>epi weight in P</td>
<td>4n</td>
<td>4n</td>
<td>4n</td>
<td>4n</td>
<td>n</td>
<td>2n</td>
</tr>
<tr>
<td>epi patch size</td>
<td>lsm/2</td>
<td>lsm/2</td>
<td>lsm/2</td>
<td>lsm/2</td>
<td>lsm/2</td>
<td>lsm/2</td>
</tr>
<tr>
<td>max sigma 0</td>
<td>20.0</td>
<td>20.0</td>
<td>25.0</td>
<td>20.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>max sigma x,y</td>
<td>0.18</td>
<td>0.18</td>
<td>0.20</td>
<td>0.15</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>max x,y [pixels]</td>
<td>2.0</td>
<td>2.0</td>
<td>1.5</td>
<td>1.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>max x,y small [pixels]</td>
<td>1.0</td>
<td>1.0</td>
<td>0.7</td>
<td>0.5</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>patch size x,y [pixels]</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>bigger p. size x,y [pixels]</td>
<td>21</td>
<td>17</td>
<td>21</td>
<td>21</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>pixel shift [pixels]</td>
<td>10.0</td>
<td>2.0</td>
<td>3.0</td>
<td>1.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>voronoi x,y [pixels]</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>grid size x,y [pixels]</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
Fig. C.1 Max: face measurement with random pattern projection, image format 768x576 pixels, for demonstration purposes (shift 10 pixels). Template image with contour defining the ROI and seed points.

Fig. C.2 Naoko; face measurement with random pattern projection, image format 768x576 pixels, usual face measurement (shift 2 pixels). Template image with contour defining the ROI and seed points.
C.1 MATCHING PROCESS

Fig. C.3  Naomi; face measurement without random pattern projection, image format 640x480 pixels, higher thresholds. Template image with contour defining the ROI and seed points.

Fig. C.4  Dummy; dummy head measurement with random pattern projection, image format 640x480 pixels, fine measurement (shift 1 pixel). Template image with contour defining the ROI and seed points.

Fig. C.5  Walking; full body dynamic measurement, image format 768x288 pixels, interlaced CCD camera. Template image with contour defining the ROI and seed points.
C.2 TRACKING PROCESS

An example of parameter set for the tracking process is listed below. It refers to the walking and baseball sequences presented in section 7.7.

Table C.2  Example of full parameter set for the tracking process. The parameter names refer to section 7.5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Walking</th>
<th>Baseball</th>
</tr>
</thead>
<tbody>
<tr>
<td>used LSM parameters</td>
<td>all</td>
<td>all</td>
</tr>
<tr>
<td>patch size x,y [pixels]</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>max sigma 0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>max sigma x,y</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>max x,y [pixels]</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>cross correlation patch size [pixels]</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>first search size x,y [pixels]</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>search size x,y [pixels]</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>large search size x,y [pixels]</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>search step x,y [pixels]</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>max diff [%]</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>tessel size x,y [pixels]</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>density threshold</td>
<td>23</td>
<td>20</td>
</tr>
</tbody>
</table>
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCEs


REFERENCES


Index

Affine transformation, 27, 31, 36, 47, 51, 119
Automatic matching process, 36, 116
Backprojection, 60
Calibration, 19, 22, 69
  bundle, 24, 60, 62
  reference bar, 22
  reference field, 22, 62
Camcorder, 14, 62, 101
CCD camera, 9, 62, 68
  interlaced, 13, 94
  progressive scan, 10, 13, 94, 129
Close the gaps, 43, 48–49
Coded light, 61, 107
Coded target, 22, 62, 69, 113
Collinearity equation, 20, 56, 58
Coplanarity condition, 29
Corresponding points, 40, 56, 64
Covariance matrix, 25, 29, 57
Cross-correlation, 39, 81–82
Data acquisition, 9
Delauney triangulation, 59, 64
Design matrix, 24–25, 28, 31, 56
Digital still camera, 16
Digital video camcorder, 16
Distortion terms
  decentering, 21
  radial, 21
Editing, 58
Epipolar
  constant, 31–32, 34
  constraint, 31–32, 36–37, 47, 49
  line, 29, 32, 47, 49
  plane, 29
  segment, 32–33, 47, 49
Face modeling, 61
Filtering, 50, 57, 86
  neighborhood, 51, 116
  regularization of grid, 50, 116
FireWire, 15, 17
Foerstner interest operator, 39, 46
Forward intersection, 64, 69, 84
Frame grabber, 11
Gaussian filters, 64
Gauss-Markov model, 25, 29, 57
Genlock, 11
Geometrical correction, 115
GUI, 111
IEEE-1394, 13, 17, 129
I-link, 17
Image coordinate system, 32
Image patch, 27, 31, 34, 36, 51
Interlacing, 10, 14
Key-point, 88, 98, 127
  tracking, 88, 98
Laser scanning, 61, 107
Least squares estimation, 27, 56
Least squares matching, 27, 81–82
  spatial, 81–82
  temporal, 81–82
Lens distortion, 21
Linear prediction, 82
Local uniformity, 51
  filtering, 87
Matching process, 64, 69, 84
Matching strategy, 40
Metric coordinate system, 32
Modeling, 59
Motion capture, 94
Multi-images, 12, 35
Object coordinate system, 20
Observation equation, 24, 27, 31
Observation vector, 24–25, 31, 56
Operating System, 111
Orientation
  exterior, 20, 23, 69, 114
  interior, 20, 23
Patch size, 47, 49
Point cloud, 55, 119
INDEX

Point viewer, 58
Principal point, 19
Projection center, 19, 29
Radiometric correction, 27, 31, 36, 47
Random pattern, 62, 68–69
Region of interest, 40, 116
Resampling
  bilinear, 28, 58
  linear, 29
Residual vector, 25, 29, 57
Result check, 35
Ring buffer, 90
Rotation matrix, 19–20
Search box, 82
Search image, 27, 31, 34, 37
Search region, 46
Search strategy, 48, 116
Seed point definition, 116
  automatic, 39, 46, 95
  manual, 37, 46
  semi-automatic, 37, 46–47, 64
Seed points, 36
Self calibrating bundle adjustment, 22
Sensor coordinate system, 20
Solution vector, 25, 29, 57
Spatial resection, 24
Stereo matcher, 27, 34
Surface measurement, 55, 63, 95
Surface tracking, 84
Synchronization, 10–11
  clapper, 15
  genlock, 10
Template image, 27, 31, 34–35, 37
Texture coordinate, 60
Texture mapping, 59, 65
Tracking loop, 81
Tracking process, 81, 84, 97
Trajectory, 84, 86, 124
  vector field, 84, 86, 88
Transformation
  affine, 27, 31, 36, 47, 51, 119
  metric to pixel, 21
  pixel to metric, 21
Validation, 71
Variance factor, 25, 29, 57
Vector difference, 31
Video signal, 9
  CCIR, 9
  EIA, 9
Virtual camera, 119
Visualisation, 64
  point cloud, 58
Voronoi tessellation, 40, 49
VRML, 59
Wallis filter, 45, 95, 116
Web camera, 17
Weight coefficient matrix, 29, 31
Acknowledgments

I would like to thank Prof. Dr. Armin Gruen at the Institute of Geodesy and Photogrammetry of the ETH Zurich for giving me the opportunity to work on interesting projects and for providing me with excellent research environment.
I thank Professor Hans-Gerd Maas who gave me the C-routines from which I started my project and for important advice during the course of my research.
I thank Dr. Pascal Fua and Dr. Ralf Plaenkers from the Laboratoire d’Infographie at the EPF Lausanne for providing me with multi-image sequences.
I thank Mr. Nobuo Kochi of the Imaging Laboratory of TOPCON Corporation in Tokyo for allowing me to continue working on my thesis during my stay in Japan.
I thank Prof. Dr. Hans-Peter Meinzer of the German Cancer Research Center in Heidelberg and Prof. Dr. Henrik Haggrén of the Institute of Photogrammetry and Remote Sensing at the Helsinki University of Technology in Espoo for being coadvisors for this work.
I thank my friends who kindly posed as models for the face measurement studies and all my colleagues and friends in the group of Photogrammetry and Remote Sensing at the ETH Zurich for their assistance in various technical fields and for their friendships. Special thank are due to my brother Massimo and my sister-in-law Chantal for their support, friendship and for correcting my English during the long writing process.
Finally, I thank from heart my family and my close friends who gave my life richness in so many ways that I can not list them all here.

N. D’A.