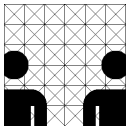
Diploma Thesis
in Informatics

Hand Pose Reconstruction using a Three-Camera Stereo Vision System

Technical Aspects of Multimodal Systems
Department Informatics
University of Hamburg

submitted by
Eugen Richter
August 2011



supervised by
Prof. Dr. Jianwei Zhang
Prof. Dr. Leonie Dreschler-Fischer

Abstract

Sequences of hand model configurations, representing articulated hand motion, can be used for many purposes. The sequence can be used as a data set for a learning process, to analyze a grasp or a manipulation task and also for remote operation or human-computer interaction purposes. Because of the high amount of degrees of freedom and the complexity involved in mapping a visual external representation of the hand onto the internal articulated structure, model reconstruction of an articulated hand based on visual information represents a complex task.

This thesis describes an approach to reconstruction of hand model configurations based on visual information from three camera views. The approach builds upon a low-cost three-camera motion capturing setup using a glove equipped with differently colored marker objects. The approach is described by a workflow, consisting of various methods that are used in order to solve image processing tasks, obtain three-dimensional information based on multiple view geometry and to reconstruct a hand model configuration using anthropometric and kinematic constraints of the hand. The presented result describes the reconstruction of a hand model as a kinematic chain that is represented by Denavit-Hartenberg parameter sets.

Kurzfassung

Als Repräsentation von Handbewegungen können Sequenzen von Gelenkkonfigurationen eines Handmodells vielseitig verwendet werden. Die entsprechenden Datensätze können eine Grundlage für Lernprozesse bereitstellen, der Analyse von Greif- oder Manipulationsaufgaben dienen und auch für Fernsteuerungsaufgaben oder die Mensch-Computer Interaktion verwendet werden. Aufgrund der hohen Anzahl an Freiheitsgraden und der einhergehenden Komplexität der Abbildung einer visuellen externen Repräsentation der Hand auf die interne Gelenkstruktur, stellt die Aufgabe der Rekonstruktion eines Modells basierend auf visueller Information ein komplexes Problem dar.

Diese Diplomarbeit beschreibt einen Ansatz zur Rekonstruktion von Gelenkkonfigurationen eines Handmodells, basierend auf visueller Information aus drei Kamera-Ansichten. Der Ansatz verwendet einen kostengünstigen experimentellen Aufbau für die Bewegungserfassung, im Zusammenhang mit einem Handschuh, ausgestattet mit unterschiedlich gefärbten Markerobjekten. Der Ansatz wird durch einen Ablauf beschrieben, bei dem unterschiedliche Methoden verwendet werden, um Aufgaben der Bildverarbeitung zu bewältigen, um drei-dimensionale Information ausgehend von der Geometrie mehrerer Ansichten zu erhalten und um Gelenkkonfigurationen des Handmodells zu rekonstruieren, unter der Verwendung von anthropometrischen und kinematischen Einschränkungen der Hand. Das präsentierte Ergebnis beschreibt die Rekonstruktion eines Handmodells in der Form einer kinematischen Kette, die durch Denavit-Hartenberg Parametersätze repräsentiert wird.

Contents

Introduction

1

Over the recent years the field of robotics has seen an enormous increase in research activity. Although it is not possible to generalize the direction of the research being conducted, a lot of the research shows a clear trend towards the area of mobile service robots.

The ultimate goal is to create a mobile service robot system that can act autonomously to support us in a wide variety of tasks. In order to manage this, it is most important to solve the challenging tasks regarding the interaction of the robot with its environment and the human being. An efficient and safe solution in this area poses the most important requirement for successful operation.

To allow for the best possible operation in a man-made environment, a lot of research has been dedicated to the humanoid form. But it is insufficient to only imitate the human appearance, the imitation must incorporate the best possible approximation of human abilities and behavior.

Most of the abilities we as human beings consider simple without giving it much thought - e.g. walking or running, interpretation of scenes we see, recognition and grasping of objects - we acquire through learning processes during the phase of growing up. And although at the current stage of the research in the field of robotics many of the abilities have been realized in some form, those realizations are usually constrained to a high degree. This seems to be unavoidable especially when relying on a single modality as the source of information, e.g. using vision trying to analyze the stability of a grasp. Human abilities are based on the use of various modalities in connection with each other. The complexity of collective interpretation of multimodal information, has so far restricted many approaches from being universally applicable.

In the design phase of a mobile service robot it is usually not very difficult to establish channels for the major modalities, vision and hearing. But it is impossible to process the channel information without additional knowledge. Apart from that, every modality is constrained by the current state of technology. Touch sensors for example, have an insufficient approximation of the feel of the human skin. Video cameras show a varying susceptibility to noise and suffer from distortion introduced by the lenses.

It is apparent that the topics related to the area of interaction of the robot with its environment and the human being, will dominate the research on mobile service robots for years to come. The european HANDLE project [HAN11], under which this thesis was written, focuses on the human ability to grasp and manipulate objects with the hands. It aims to build an understanding of the ability by observing the human being and learning based on the acquired data, in order to be able to replicate the ability using an anthropomorphic artificial hand. Direct observation represents one of the basic tools available to research, generating data specific to the observed behavior or exercised ability. This thesis serves the purpose of reconstruction and description of hand configuration sequences through direct observation of human hand motion via a multi-camera stereo vision setup.

Although this thesis was written with the field of service robotics in mind, the results are not restricted to it and could also be used in other areas, such as biomechanical research, hand prosthetics, human-computer interaction or virtual reality.

1.1 Motivation

Currently the tasks we perform with our hands, still pose a very complex problem with regards to the analysis and replication. Much of it can be ascribed to the high dimensionality of the problem. The human hand possesses 27 degrees of freedom. Although the kinematics of the articulated structure is constrained to a relatively high degree, the configuration search space remains very large. Moreover, the acquisition of data as a representation of the performed task is non-trivial.

Motion capture techniques

Motion capture represents the fundamental method for gathering data that describes human motion sequences. It has been widely adapted within the motion picture industry, in order to produce natural motion of animated human characters. According to Bray [Bra03], existing motion capture systems can be categorized as electromagnetic, electromechanical and optical.

Electromagnetic systems operate by measuring voltages resulting from relative magnetic flux in the orthogonal coils of the transmitter and the receiving sensors. This allows to determine the position and orientation of each sensor attached to the subject. While this system is insensitive to occlusion, it requires the subject to wear wired sensors, which often restricts the range of motion. The system also suffers from interference introduced by metallic objects within the working volume.

Electromechanical systems operate by direct measurement of joint angles, requiring the subject to wear an exoskeletal sensor system. While this type of system is also insensitive to occlusion, it often restricts the range of motion due to rigidity and unwieldiness of the construction.

Optical motion capture systems operate by recording and processing images of the subject, equipped with marker objects. The marker objects can either be passive (e.g. retro-reflective or colored), or active (e.g. LEDs). While passive markers are least restrictive in terms of weight and range of motion, use of passive markers imposes many constraints on the environment of the working volume. Moeslund and Granum [MG01] give a short overview of typical assumptions made by optical motion capture systems. Computational effort of the image processing stage can vary, depending on how well the constraints of the working volume are satisfied. Passive markers of the same color (e.g. retro-reflective) complicate the identification/labeling process. Pulsating active markers can be used to alleviate the problem. Optical motion capture systems in general can not handle occlusion of marker objects. The possibility of marker object occlusion usually is counteracted with an increase in the amount of camera devices.

Another type of optical motion capture that enjoys popularity, due to its non-invasiveness, is markerless motion capture. This approach is based upon the extraction of features in combination with attributes specific to parts of the human body (e.g. skin tone). In order to find the most fitting model within the high-dimensional search space of hand configurations, computationally expensive inference algorithms are required.

1.1.1 Objective of the thesis

Current available commercial motion capture systems mostly use dedicated special purpose hardware for acquisition and processing of the data. Therefore they are generally prohibitively expensive, even with a small configuration.

Due to constant progress in camera technology and the increasing interest in using cameras for interactive gaming, current consumer-grade cameras offer a readily available and relatively inexpensive means of recording images. Therefore it is of interest to investigate the possibility and limits of using an inexpensive optical setup for motion capture purposes.

Motivated by the topics of the HANDLE project, which the TAMS group participates in, this thesis pursues the objective of investigation and development of a solution allowing to capture the motion of an articulated hand using an inexpensive three-camera stereo vision setup with off-the-shelf components.

Based on acquired image data, the results produced by the developed solution should represent a description of the sequence of hand configurations according to a defined model of the hand. For visual interpretation and validation of the results, the developed solution should provide a visualization module.

The hand model used by the developed solution, should offer a close relation to the kinematic structure of the ShadowHand C5 [Sha08], an anthropomorphic artificial

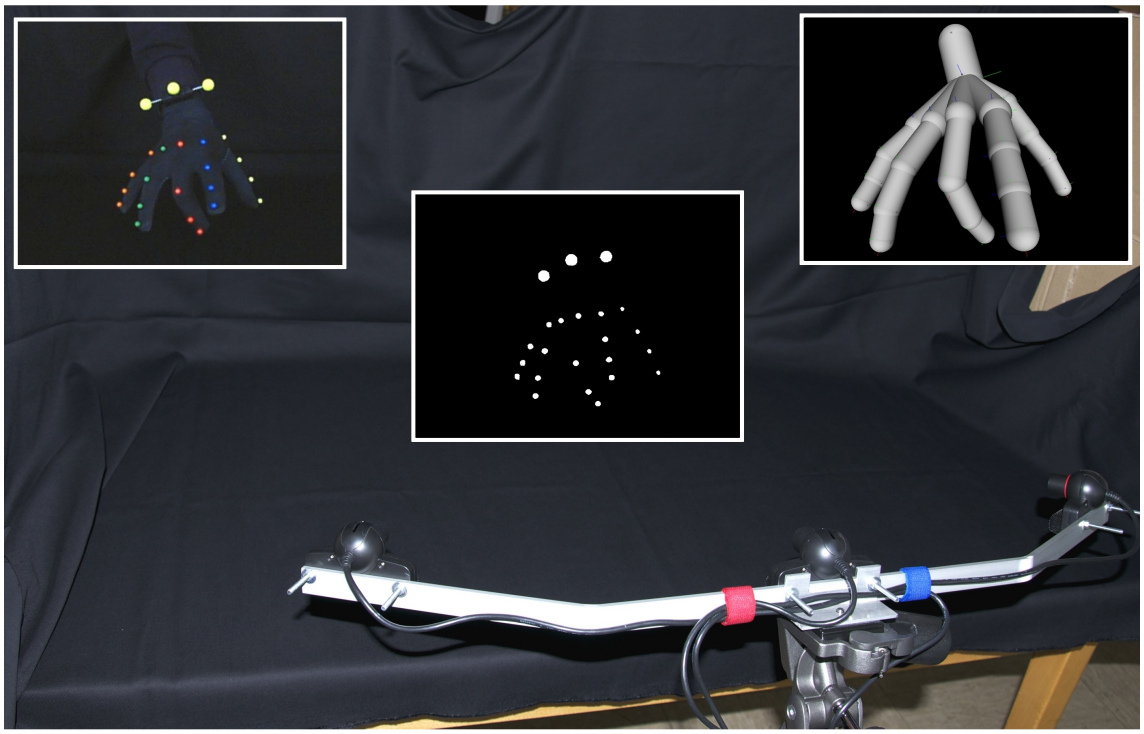


Figure 1.1: General outline of the approach (left to right): **1.** Motion recording with the three-camera setup - **2.** Image processing and stereo reconstruction - **3.** Hand pose reconstruction and visualization.

hand used within the HANDLE project by the TAMS group. This would allow the transfer of reconstructed hand model configurations onto the artificial hand for testing and validation purposes.

In order to be able to share the data with the participating partners of the HANDLE project, the results produced by the developed solution should be stored and annotated in compliance with the guidelines [HAN09] set for the project.

1.2 Related work

Pose reconstruction of the human body based on visual information is a very actively researched topic. A survey done by Aggarwal and Cai [AC97] followed by a more recent survey by Moeslund and Granum [MG01] presented an overview of existing vision-based approaches, putting forward a taxonomy and a general outline of the framework for pose reconstruction of the human body. In [EBN+07] Erol et al. presented a review of vision-based approaches specifically to the problem of hand pose reconstruction. Within the review problems specific to the hand, such as

high dimensionality and self-occlusion, are discussed, followed by categorization of existing approaches and outline of associated methods.

A multitude of research has been done in reference to hand pose reconstruction. While some approaches used a stereo vision setup with two or three cameras, many focused on reconstruction based on a single view, in order to reduce the computational effort associated by image processing. Although the use of marker objects is well established, many recent approaches focused on the features of the hand itself. Self-occlusion was often approached with statistical methods in combination with inverse kinematics and kinematic constraints. The approaches outlined in the following are a few examples of related research.

A single camera in combination with a glove equipped with eight color markers, was used by Lathuiliere and Herve [LH00] in order to reconstruct a hand model described by 26 degrees of freedom. Following three-dimensional reconstruction of the palm based on known fixed marker distances, an inverse kinematics model of the hand was used, in order to determine finger configurations. Self-occlusion was modeled based on fingertip visibility tables and a heuristic was defined, in order to perform the reconstruction with occlusions. Experimental evaluation showed that their method was able to operate at a frame rate of 15 frames per second, while producing an average error between 1.0 and 9.0 degrees for the joint angles.

Also using a single camera, Kwon et al. [KZD01] approached reconstruction of a partial hand pose based on a glove equipped with 12 equal markers for the palm and four fingers excluding the thumb. The POSIT (Pose from Orthography and Scaling with ITerations [DD95]) algorithm was used for three-dimensional reconstruction of the hand pose. Experiments with a cardboard hand form showed an average error of up to 5 degrees for the reconstructed joint angles.

Rehg and Kanade [RK94] described a markerless approach using a two-camera stereo setup. Edges were used as features to derive image coordinates for the tip of each finger, as well as the finger joints based on determination of the orientations of each finger link. Through estimation of feature locations based on short-time history, accuracy was improved. Range of motion was constrained to exclude self-occlusion. A description of the reconstructed hand pose by Denavit-Hartenberg parameter sets was produced.

A similar approach was presented by Chen et al. in [CFAT07]. Here a three-camera stereo setup was used to reduce self-occlusion. Based on extraction of shape features of the hand, a heuristic method was used for assignment of the detected features. A combination of established kinematic constraints and an inverse kinematics model was used to reconstruct a hand configuration. Visual evaluation of experimental results showed that the method provided relatively accurate reconstruction.

In [SMC01] Stenger et al. proposed a model-based single-view approach that utilized the geometry of a three-dimensional hand model built from truncated quadrics, in order to determine a close fitting match between generated two-dimensional profiles

of the synthetic model and features extracted from the acquired image. An unscented Kalman filter was used to minimize the geometric error. The approach produced a successful reconstruction of a model with seven variable degrees of freedom while the remaining 20 were assumed as fixed.

Ueda et al. [UMIO01] described another model-based approach using multiple silhouette images of the hand acquired from multiple views. Based on the integration of the silhouette images into an octree-based voxel model of the hand pose, the hand configuration was obtained by fitting a three-dimensional hand model to the voxel model. The method was evaluated based on computation of the angle error for the metacarpophalangeal joint of the index finger throughout multiple fitting iterations. Depending on the level of the octree, the experimental results showed an error between 1.0 and 2.0 degrees.

A data-driven approach for markerless reconstruction of the articulated hand pose using a single view, was presented by Romero et al. [RKK09]. A histogram of oriented gradients, as a representation of the hand in the image, was used to query a database composed of 100000 synthetic hand poses, conducting a nearest-neighbor search to find a fitting hand pose.

A quite similar approach was published by Wang and Popović [WP09]. The presented data-driven approach used a single view, with a database of natural hand poses of a hand model with 26 degrees of freedom. The database was constructed from a set of 18000 hand configurations recorded with a CyberGlove II. Using a distance metric between two configurations a low-dispersion set of configurations was selected, describing a non-redundant set of natural hand poses. Using the determined set of configurations, a synthetic hand model was rendered at various three-dimensional orientations to cover the range of natural hand poses followed by scaling, in order to produce tiny rasterized images. The subject was required to wear a glove with a color pattern, designed specifically to simplify the hand pose reconstruction problem. The colored glove is shown in figure 1.2.

The camera images were transformed into normalized tiny images and used to query the database. A nearest-neighbor search was performed on the database to retrieve a most fitting hand configuration. To improve accuracy of the result, an inverse kinematics model was used to minimize the error between the rasterized images. The method was able to reconstruct individual articulation of the fingers at a frame rate of 10 frames per second. The method was evaluated with different hand poses, based on the reprojection error between the query image and the result from the database. The method produced a reprojection error of up to 5 pixels, with jitter (given as the root mean square of the pose distance) as high as 35mm. Visual evaluation of the depth ambiguity showed an average error between 5cm and 10cm.

While some of the approaches using a single camera, were able to perform in real-time for a determined maximum frame rate, non of the methods were able to offer real-time performance at 30 frames per second or higher. Although the approach

Figure 1.2: Colored glove (left) used in the approach by Wang and Popović [WP09] and the reconstructed three-dimensional model (right).

described in this thesis does not offer real-time performance, it investigates the possibility of using a frame rate of 60 frames per second for image acquisition with a three-camera stereo vision setup, in order to reduce motion between subsequent images and thus improve stability of the results in offline processing. Moreover, a glove with 23 colored markers is used, in order to allow determination of joint positions without the need for an inverse kinematics model.

Another approach, similar to the approach described in this thesis, was published by Cerveri et al. [CDML+07]. The described method used a six-camera stereo vision setup to reconstruct a hand pose for the purpose of biomechanical analysis of hand motion. A commercially available stereo-vision setup with dedicated hardware was used, along with a set of 24 retro-reflective markers, in order to identify the joints of the articulated hand structure and the reference frame of the wrist joint. The hand and the reconstructed three-dimensional model are shown in figure 1.3.

A hand model with 27 degrees of freedom was used. Starting with a static posture at the beginning of the motion capture process, a static distance based marker labeling algorithm was used to assign the reconstructed marker objects to the corresponding joints. With the wrist joint as the root of the model, the thumb was labeled first, followed by the metacarpophalangeal (MCP) joints of the other fingers based on their closeness to the wrist joint and increasing distance from the metacarpophalangeal joint of the thumb. The remaining markers were labeled according to increasing distance from the corresponding MCP joint. Following the labeling procedure, anthropometric measures along with the axes and centers of rotation specific to the subject's hand were determined. A set of equations for computation of rotation about the determined axes of rotation was defined and used to determine the hierarchical kinematic chain of the hand model.

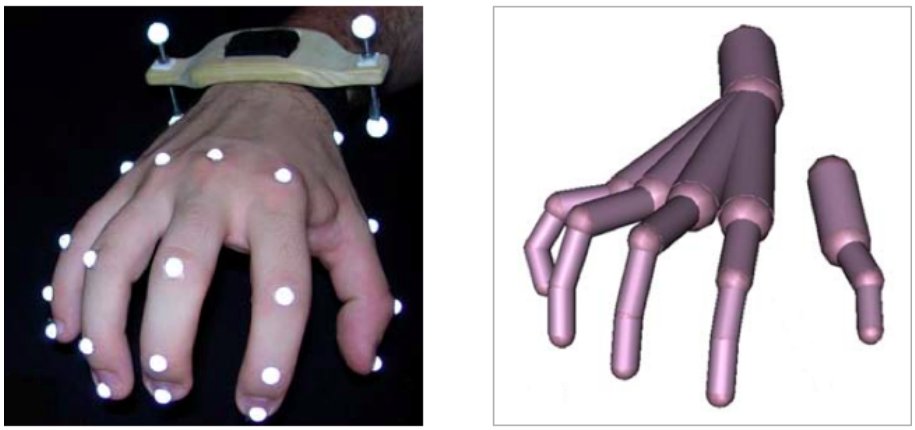


Figure 1.3: The hand equipped with retroreflective marker objects (left) used in the approach by Cerveri et al. [CDML+07] and the reconstructed three-dimensional model (right).

In order to increase performance a tracking method was used. Based on a short-time history (two consecutive images) for the trajectory of each marker object, the reconstructed marker objects were labeled dynamically using proximity thresholds to compare a predicted position with the reconstructed position. Failing to do so, the static labeling procedure was applied. In order to increase the robustness to self-occlusion, geometrical constraints were used.

Evaluation of the approach with multiple motion sequences, showed a root mean square error (predicted/reconstructed position of thumb marker objects) between 0.5mm and 2.85mm. Testing with the same motion sequences at 60 frames per second showed robustness of the tracking method, with a low percentage of lost markers: up to 4.47% for the thumb markers and up to 11.42% for the index finger. Acquisition at 120 frames per second showed significant improvement. Actual real-time performance was determined at about 50 frames per second.

While the approach presented in this thesis is quite similar to the approach described in [CDML+07], it has two important differences. One important point is the use of inexpensive off-the-shelf components, in order to acquire and process the image data. Moreover, the marker objects used to identify the joints of the articulated hand structure are color coded, so that each finger is represented by one single color. This approach simplifies the labeling procedure of the reconstructed marker objects.

1.3 Outline of the thesis

The current chapter presented the motivation for this thesis and its objective. Related works have been discussed and a distinction made in relation to the approach described in this thesis.

The following chapter addresses the image processing workflow, building the foundation for the approach described in this thesis. It discusses the formation of digital images and the inherent problem of noise, followed by methods for noise reduction. The motivation behind the choice of the HSV color model for the specification and identification of color information is presented. Following the outline of the role of image segmentation, the employed semi-automated approach to multiple thresholding based on color calibration samples is explained. The purpose and effects of morphological filtering are investigated, followed by shape extraction using a contour search algorithm. Concluding the image processing workflow, the concept of moments as shape descriptors is addressed, in order to enable determination of two-dimensional centroid coordinates of every shape representing a marker object projection.

Chapter three describes the stereo vision workflow that is used to perform three-dimensional reconstruction of the marker objects. Starting with the ideal pinhole camera model, the importance of the process of camera calibration is discussed and the intrinsic and extrinsic camera parameters are presented. Following that, the camera calibration method used to obtain the calibration parameters of the cameras used in the experimental setup, is presented in detail. The necessary steps for the transition to a stereo vision system are explained, followed by a distinction between systems with a standard and general stereo geometry. Epipolar geometry, the geometry of a general stereo vision system is described next, followed by the distinction between the fundamental matrix and the essential matrix in relation to the epipolar geometry, completed by the outline of two common methods for estimation of both types. The problem of search for correspondence is presented, along with the purpose and effects of image rectification, followed by a description of the rectification method used in this thesis. Triangulation, the reconstruction of a three-dimensional representation of a marker object based on its corresponding projections is described next, along with a discussion of existing methods. The application of the workflow to the experimental three-camera setup concludes the chapter.

The fourth chapter describes the approach to hand pose reconstruction based on the obtained three-dimensional marker object data. The articulated structure of the human hand is presented, together with its anthropometric attributes and kinematic constraints. The definition of the kinematic structure of the hand model used in this thesis follows, along with the associated assumptions. A distance-based approach is presented for the purpose of assignment of marker objects to joints according to

their order in the articulated structure of the model. The idea behind the Denavit-Hartenberg parameters is presented next, followed by the approach to reconstruction of the kinematic chain of the model based on assigned marker object sets.

Chapter five describes the components of the experimental setup that was used in this thesis. The architecture of the implemented software application is presented, along with the rationale behind it.

The sixth chapter presents and discusses the experimental results that were obtained using the approach presented in this thesis.

The concluding seventh chapter summarizes the work done in this thesis and discusses ideas for improvement and further work.

Image processing workflow

2

The domain of image processing consists of a wealth of methods to achieve various tasks. The task of image enhancement for example, may require the use of methods for noise reduction, while the task of image restoration may depend on a mathematical model of image degradation. Many fields, such as the field of artificial intelligence, rely heavily on the methods of image processing. To approach the task of object recognition, one of the fundamental problems to solve is the problem of image segmentation - separation of the objects of interest from the rest of the image. The quality of the segmented image dictates the degree of possible differentiation between object shapes based on their various features. The result is a primitive description of the image.

In order to allow identification of the hand joints, a glove was used, equipped with differently colored spherical marker objects that were attached at the approximated positions of the corresponding joints. A set of image processing steps has been implemented as part of a software application that allowed to determine the two-dimensional positions of the marker objects in the recorded images, while differentiating the marker objects by color.

In contrast to the task of three-dimensional reconstruction based on stereo vision (see chapter 3), where a general sequence of steps is necessary to obtain a result, there does not exist a general sequence of steps in image processing. It is up to the viewer to decide, whether a certain image processing workflow provides acceptable results.

This chapter will present the image processing workflow, which was implemented within a software application as part of this thesis. Starting with an overview of the process of digital image formation, background information on the main types of image noise and a few methods for noise reduction will be provided. The choice of a suitable color model will be presented, followed by the process of image segmentation and the threshold determination method used in this thesis. In the final section the set of methods will be presented that was used for the extraction of two-dimensional coordinates of the marker objects, based on shape analysis.

2.1 Digital image formation

The foundation of digital image formation is based on the projection of rays of light onto a photosensitive image sensor which represents a digital image plane. The image sensor consists of an array of photodiode cells that contain a photoactive region and represent the image pixels. Upon contact with a cell of the image sensor, the stream of photons transported by a ray of light entering a camera, is converted into an electric charge. The charge produced in each image sensor cell is proportional to the intensity of light at that specific cell.

Charge-coupled device (CCD) image sensors and *complementary metal-oxide semiconductor (CMOS)* image sensors, both constructed from silicon, are two main types of image sensors that are widely used in various types of digital cameras. The cameras used in the experimental setup (see chapter 5) are equipped with a CMOS-sensor.

A CCD-sensor works by transporting the charges of each cell that were accumulated during the exposure interval, to a charge amplifier. A commonly taken approach is to shift the charges through the adjacent cells of a column of the image sensor array, down into a readout register that delivers the charges to the charge amplifier. Therefore the image sensor is read row by row. The charge amplifier converts the charges into voltages. The result is an analog representation of a light intensity image acquired by the image sensor.

In contrast to the CCD-sensor, each one of the image sensor cells of a CMOS-sensor, has a dedicated charge amplifier. Therefore the cells are not coupled and each cell is directly addressable. This benefits the readout performance of the CMOS-sensor, which is generally higher compared to the CCD-sensor. In addition to the advantage of being less expensive to manufacture, another advantage of the CMOS-sensor is provided by the CMOS manufacturing process itself, which allows to integrate additional circuitry, e.g. for image processing, directly onto the image sensor chip. Although the sensor thus becomes more complex, it does require less external logic compared to the CCD-sensor.

To obtain a discrete representation, the voltages produced by charge amplifiers are sampled. A following quantization of the sampled values maps the sampled values to a finite set of numeric values representing a specific digital interpretation, e.g. an 8-bit scale of grey values. A CMOS-sensor usually provides the analog-digital conversion directly on-chip, thus producing a digital output, in contrast to a CCD-sensor, which requires external components to achieve this result.

2.1.1 Color filter array

The image sensor generally delivers a measure of light intensity only, with the digital representation being a grayscale image. To enable the measurement of color specific

intensities, a color filter needs to be used. The *Bayer color filter array* is the most common and widely used color filter array. Figure 2.1 shows the principle of the Bayer color filter array.

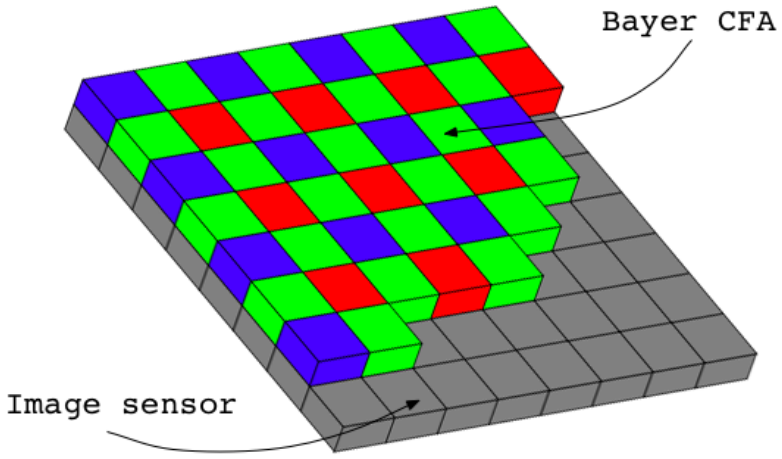


Figure 2.1: The Bayer color filter array (RGB). The filter array separates the spectrum of visible light into wavelength intervals centered around red, green and blue. Each wavelength interval is passed through to a single pixel according to the given pattern, with green being passed to twice as many pixels as red or blue.

A color filter works by allowing only a specific wavelength interval of the spectrum of visible light, to pass. The Bayer color filter array separates the spectrum of visible light into intervals centered around red, green and blue (RGB), and passes it on to the image sensor cells according to the pattern. The pattern of the filter is inspired by the physiology of the human eye, that is more sensitive to green than either red or blue, with the green component carrying most of the luminance information [MHC04]. Thus half of the space of the color filter array is dedicated to green. The pattern does not allow a direct determination of the complete color information of a single image pixel. The color information must be computed using a demosaicing algorithm, such as the one proposed by Malvar et al. in [MHC04].

The performance of image sensors is affected by a variety of factors, such as the temperature of the operating environment for example. As a result of non-ideal operating conditions the resulting images are affected by noise.

2.2 Image noise

Image noise is an undesirable byproduct of the process of image acquisition, which complicates the analysis of recorded images. Noise in digital images arises mainly

during the image acquisition itself. According to Healey and Kondepudy [HK94] and Liu et al. [LFSK06] the image is mainly affected by the following types of noise: *fixed-pattern noise*, *dark-current noise*, *shot noise*, *amplifier noise* and *quantization noise*. Figure 2.2 shows a few examples of the different types of image noise.

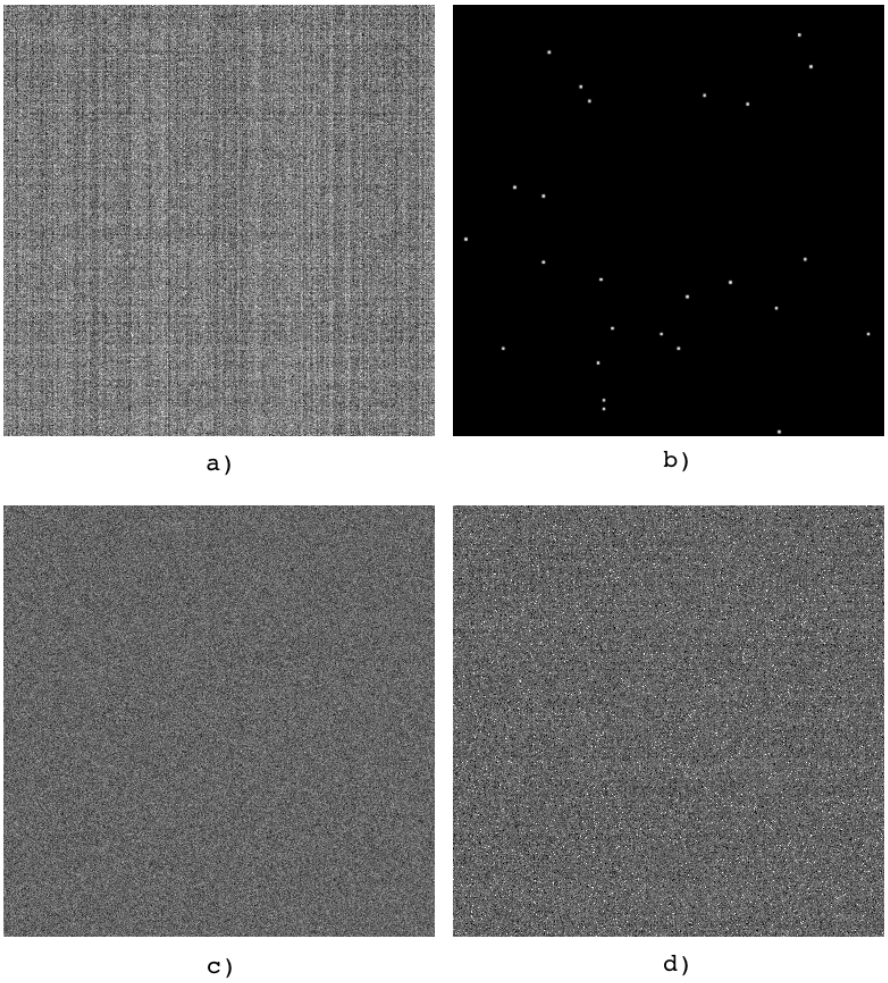


Figure 2.2: Examples of image noise. **a)** Fixed-pattern noise, that represents a constant non-uniformity in the image. **b)** Dark-current noise, a result of thermal agitation, produces a charge without exposure. **c)** Shot noise, a result of random charge fluctuations due to the randomness of photon arrival time. **d)** Amplifier noise, introduced by charge amplifiers during readout.

Fixed-pattern noise is represented by a pattern of brighter and darker pixels, which occurs in sequences of images that have been taken under the exact same lighting conditions. The fixed pattern therefore represents a non-uniformity that is constant over time and appears in each recorded image.

Dark-current noise affects the recorded images due to thermal agitation (thermal noise) and longer exposure times. Randomly generated electrons in the depletion region within the image sensor cells produce a small electric current. The current is generated even when the image sensor is not exposed to light, hence it is called the dark current. Dark current noise follows a Gaussian distribution.

Shot noise is caused by random charge fluctuations, representing the uncertainty of the amount of electrons stored in an image sensor cell, due to the randomness of the arrival time of photons on the surface of the image sensor. Shot noise follows a Gaussian distribution. According to [HK94] shot noise is a fundamental limitation and cannot be eliminated.

Amplifier noise is introduced by charge amplifiers, which generate read noise mainly due to thermal agitation. The amplifier noise is an additive form of noise that follows a Gaussian distribution.

Quantization noise occurs during the conversion of the analog voltage signal into a digital representation. Mapping of intervals of sampled values to the same value inside the range of the digital representation, affects the image with uniformly distributed noise.

Another type of image noise is *impulsive noise*, known as *salt-and-pepper noise*. It affects the image with randomly distributed white and black pixels. Impulsive noise appears due to malfunction of the image sensor. A false saturation produces a white pixel and a failed response produces a black pixel.

2.2.1 Noise reduction

Not all methods of noise reduction require image processing. Cooling of the camera device is one possible method for reduction of image noise due to thermal agitation. No general recommendation can be given as to what methods must be used and in what combination, to achieve the best possible result. The choice remains firmly linked to the conditions under which the images are recorded and the associated purpose. The methods that have been evaluated in the course of this thesis will be outlined in the following.

Dark-frame subtraction is one well known method to reduce image noise that is often used for images taken with a long exposure time. A series of images are taken with the camera shutter closed or lens covered, hence the term dark frame. With the image sensor in the dark, only the noise produced by the image sensor is recorded. Averaging over the series of dark frames reveals the fixed-pattern noise that is constant in all frames and thus can be removed from a recorded image by subtraction. The image sensor that is used by the cameras in the experimental setup, incorporates circuitry that uses a proprietary method to remove fixed-pattern noise. Due to short exposure times as a result of recording images at high frame rates of

up to 60 frames per second, the analysis of dark frames taken with the lens covered, revealed almost no noise, thererofe the decision was made to skip this step.

A variety of different image filters exist, which possess a noise reducing effect. The *Gaussian filter* and the *median filter* are two popular methods in the field of image processing. Both filter methods have been evaluated for the purpose of this thesis and will be outlined in the following.

The Gaussian filter method is based on the use of a convolution matrix. The convolution matrix, usually of the form $n \times n$ (e.g. 5×5), represents a discrete approximation of the filter function. The components of the matrix represent positive and negative weights, which are applied to the image pixel values found under the matrix during a convolution process. The matrix contains a center point that is usually located at the center of the matrix. Through simple modification of the convolution matrix, a variety of filter functions with different properties can be defined.

The convolution matrix of a Gaussian filter consists of positive weights only - with the highest weight applied to the center point - and produces the weighted average of the pixel under the center point and its neighborhood as a result. Due to this circumstance the Gaussian filter smooths the image and the associated effect is called Gaussian blur.

The Gaussian filter is a linear and isotropic filter. It corresponds to a two-dimensional Gaussian function given by

$$G_\sigma(x, y) = e^{-\frac{r^2}{2\sigma^2}} = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

where σ is the standard deviation that represents the width of the bell-shaped function and r is the distance from the center.

The main disadvantage of a linear filter is that the smoothing of the image that reduces image noise, simultaneously reduces the image quality. A linear filter affects all image structures, such as object boundaries, edges and points, by displacing or eliminating them. The effect is amplified through the presence of outliers, as is for example the case with impulsive noise. Therefore the use of a Gaussian filter for image noise reduction is limited and the Gaussian filter is not very often used for this purpose in the field of computer vision. Kuwahara et al. proposed an improved method in [KHEK76], known as the *Kuwahara filter*, that smoothes the image while preserving the edge structures.

The median filter method operates on the image using a filter matrix, which resembles the convolution matrix used by the Gaussian filter in reference to the form (usually $n \times n$). The median filter replaces the value of the image pixel located under the center point, at its defined location in the filter matrix, by the median pixel value of all sorted pixel values in the neighborhood region defined by the matrix. Let $I(x, y)$ be a two-dimensional image function and $F(u, v)$ a filter region, then the filtered image $I'(x, y)$ is given by:

$$I'(x,y) = median\{I(x+u, y+v) \mid (u,v) \in F\} \tag{2.2}$$

The median filter shows robustness to outliers in the filter region. According to [BB08] (ch.6), the smoothing of the image introduced by the median filter eliminates very small structures, smaller than half the size of the filter matrix, while leaving bigger structures mostly intact. Furthermore the median filter tends to preserve the positions of those structures while causing only minimal blurring of regional boundaries. An example of the application of a median filter is shown in figure 2.3. A repeated application of the median filter has the effect of producing an image with nearly uniform regions, which proves to be a very useful effect pertaining to image segmentation. These properties make the median filter a very popular method in computer vision applications.

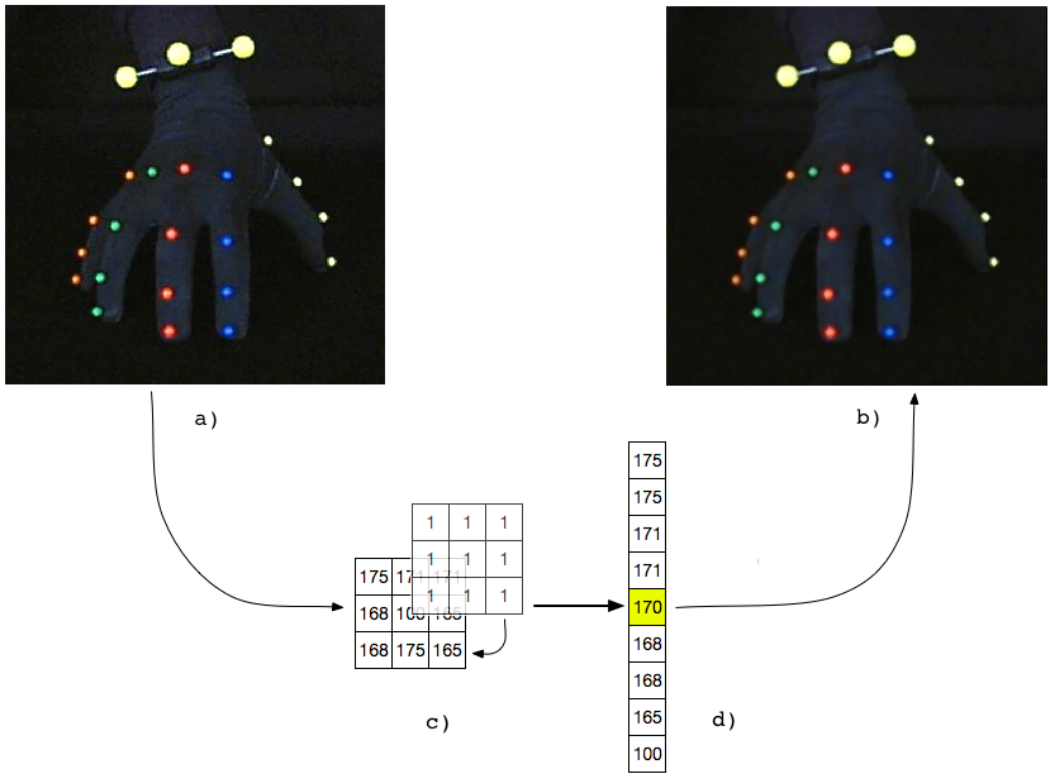


Figure 2.3: a) shows an extract of a recorded image in its unprocessed form. **b)** shows the result of the application of two iterations of a 3×3 median filter, as defined by the filter matrix in figure **c)**. **d)** illustrates the selection of the median value under the filter region, replacing an outlier.

The OpenCV framework [Wil10] provides implementations of both filter methods that were used for evaluation. Multiple iterations, 3×3 and 5×5 convolution and

filter matrices, as well as a combination of both filter methods have been evaluated based on the visual quality of the result produced in the segmentation step. Based on the obtained results, two iterations of the median filter with a 3×3 filter matrix were chosen. In comparison to the Gaussian filter, the chosen filtering combination provided a more uniform color region representing the marker object, thus leading to better segmentation results. Measurement of execution time over a series of 500 images yielded a slightly better performance of the median filter.

2.3 Color models

An important choice to make in processing of color images, is the choice of a color space and model. The predominantly used color space for storage and presentation of digital color images, is the RGB color space. The RGB color space specifies a color based on the additive mixture of values of the primary colors red, green and blue. While the RGB color space, is very well suited for storage and display of color information on electronic devices, its specification of color does not resemble the human color perception and therefore makes the specification or identification of a specific color not very intuitive.

One possible solution to this problem is provided by the *HSV color model*, introduced by Smith in [Smi78]. Formally belonging to the RGB color space, the HSV color model is defined as a geometric transformation of the RGB color space that represents a different encoding of color information. The HSV color model enables the specification of a color based on three main properties of human color perception: *hue (H)*, *saturation (S)* and *value (V)*. The geometry of the HSV color model can be described in form of a hexcone or cylinder. An illustration of the geometry of the HSV color model is shown in figure 2.4.

The hexcone of the HSV color model is created through vertical alignment of the RGB cube's diagonal that represents the grey values, followed by flattening of the cube, so that the primary colors become coplanar with the highest grey value (white). The hexcone of the HSV color model addresses a color by following properties:

- **Hue (H):** The circle on top of the hexcone represents the hue with a range of 0° - 359°. The primary colors of the RGB cube are located as follows: **R**ed = 0°, **Y**ellow = 60°, **G**reen = 120°, **C**yan = 180°, **B**lue = 240°, **M**agenta = 300°.
- **Saturation (S):** The distance from the center of the circle represents the saturation of the color as a percentage, with a value at the center representing a completely unsaturated color ($S = 0\%$) and a value at the perimeter representing a fully saturated color ($S = 100\%$).
- **Value (V):** The perpendicular to the circle represents the value of the color given as a percentage. The value property often is called brightness (B),

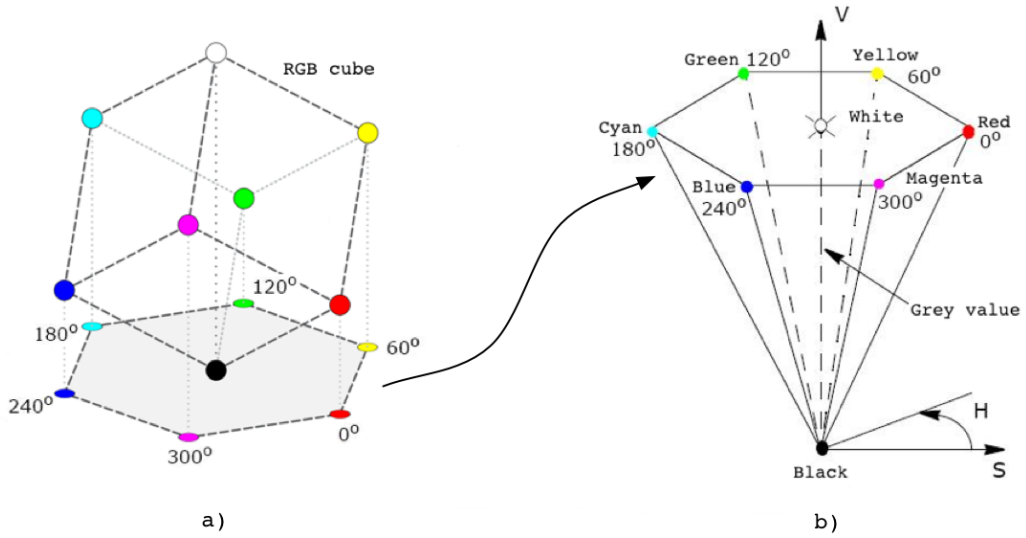


Figure 2.4: Illustration of the HSV hexcone. The transition from **a)** to **b)** illustrates the geometric transformation of the RGB cube. The circle on top of the hexcone represents the hue (H), with a range of 0° - 359°. The distance from the center of the circle represents the saturation (S) of the color, with a range of 0% - 100%. The perpendicular to the circle represents the value (V) of the color, also with a range of 0% - 100%.

thereby defining HSB, a synonym to HSV. A total absence of value/brightness (black: $V = 0\%$) is found at the bottom of the cone, while $V = 100\%$ is located at the center of the circle.

The conversion between the RGB color information and its HSV representation according to [Smi78] and the implementation within the OpenCV framework, is given by:

$$V = max(R, G, B) \tag{2.3}$$

$$S = \begin{cases} \frac{V - min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} 60 * \frac{(G-B)}{S} & \text{if } V = R \\ 60 * (2 + \frac{(B-R)}{S}) & \text{if } V = G \\ 60 * (4 + \frac{(R-G)}{S}) & \text{if } V = B \end{cases}$$

For the purpose of this thesis the HSV color model was chosen for color related processing of the acquired color images. It conveniently allows to specify the different colors of the marker objects, through simple identification of the associated hue range

of each marker color, based on color samples that were used for calibration purposes. The identification of the hue range of the marker colors effectively identifies the marker objects in the image and therefore allows to separate the marker objects from the rest of the image. The process of image segmentation will be discussed in the following section.

The cameras used in the experimental setup produce images with color information encoded according to the *Y'CbCr color model*. The Y'CbCr color model was defined in [ITU11] as a standard for encoding of color information in digital component video. Y'CbCr does not represent a color space, like the HSV color model it is another definition of an encoding scheme for color information that is based on the RGB color space.

The Y'CbCr color model is a scaled and offset digital counterpart of the *YUV color model*. Y' represents the light intensity of an image pixel, known as *luma*, whereas Cb and Cr represent the blue and red difference components, called *chroma*. The blue and red difference components are defined as $Cb = B' - Y'$ and $Cr = R' - Y'$, where Y', B' and R' represent gamma-corrected values.

Although the cameras work with raw RGB data internally, the images are converted into a Y'CbCr format to reduce the amount of data to be transferred to the host computer. This is a very common choice for consumer grade cameras that offer a frame rate of 30 frames per second and higher.

The Y'CbCr color model offers various format definitions for specification and storage of color information of a single image. Two widely used formats are Y'CbCr 4:2:2 and Y'CbCr 4:2:0, with the first one being used by the cameras in the experimental setup. Both formats are based on subsampling of the chroma information. Because the human visual system is more sensitive to rapid changes of intensity than to rapid changes of color, it is possible to reduce the color information through subsampling, thus reducing the amount of data, without significant loss in image quality.

The Y'CbCr 4:2:2 format makes use of horizontal subsampling. It therefore contains luma component information at double the sampling rate of the blue/red chroma components, i.e. the color information of two horizontally adjacent pixels is represented by different luma values but the same chroma values. Compared to RGB, the 4:2:2 format allows to reduce the amount of data by a third. The Y'CbCr 4:2:0 format uses horizontal and vertical subsampling. It samples the blue and red chroma components at half the sampling rate of the luma component in the horizontal and vertical directions. Therefore one single sample of both chroma components together with four luma component samples, specify the color of four (2×2) adjacent image pixels. This allows to halve the data size compared to RGB. This format is widely adapted by many video coding standards, such as MPEG-2 and MPEG-4.

The conversion of color information from Y'CbCr to RGB according to [Jac07] (ch.3), is given by:

$$
\begin{aligned}
R &= Y' + 1.371(Cr - 128) \\
G &= Y' - 0.698(Cr - 128) - 0.336(Cb - 128) \\
B &= Y' + 1.732(Cb - 128)
\end{aligned}
\tag{2.4}
$$

These equations produce RGB values in the range of 16 - 235 (Studio RGB). Therefore an adapted form of the equations, as presented in [Jac07], was used in the implementation of a software application as part of this thesis, to ensure the use of the full RGB range:

$$
\begin{aligned}
R &= 1.164(Y' - 16) + 1.596(Cr - 128) \\
G &= 1.164(Y' - 16) - 0.813(Cr - 128) - 0.391(Cb - 128) \\
B &= 1.164(Y' - 16) + 2.018(Cb - 128)
\end{aligned}
\tag{2.5}
$$

It is important to note that the resulting RGB values may exceed the 0 - 255 range, due to noisy values for luma and chroma. In that case the results need to be clipped. Furthermore the in-camera conversion from RGB to Y'CbCr, followed by a reversed conversion on the host computer, leads to slight degradation of color information due to non-integer factors used in the conversion.

2.4 Image segmentation

The steps of image acquisition and noise reduction are followed by the process of image segmentation, i.e. the differentiation between the objects of interest and the rest of the image. Applied to the recorded images, the segmentation implicitly defines the marker objects as objects of the foreground and the rest of the image as background. For better control of the outcome, the image sequences were recorded in a controlled environment with a black background and mostly constant lighting conditions.

One of the simplest techniques for the separation of foreground objects and the background is *image differencing*, i.e. the calculation of a segmented image by pixel-wise subtraction of two subsequent images. Assuming that the background is static, moving objects are considered as foreground objects. This method has the advantage of being the least computationally expensive method for background subtraction. An extension building upon this method, is based on "learning" of the background through calculation of the average and its standard deviation for every

image pixel, over a sequence of multiple images. Toyama et al. provide a good overview of background subtraction techniques in [TKBM99].

While both methods are particularly useful for detection of motion between two images, they are not applicable to the problem of this thesis. An evaluation has shown that both methods produced an image, which either contained a more or less complete form of the hand, with completely indiscernible marker objects, or contained no objects of interest at all, in the case where the hand did not move. Apart from that, both methods completely ignored the color information provided by the marker objects.

2.4.1 Multiple thresholding

Another fundamental method for image segmentation is image thresholding. It is comparable to the background subtraction methods with regard to simplicity of implementation and computational effort. Image tresholding allows to perform a differentiation between the foreground objects and background of an image based on either a single threshold, known as *global thresholding*, or multiple thresholds, known as *multiple thresholding*. The latter variant effectively enables to handle multiple classes of objects. The result is a binary image, with 0 (black) usually representing a background pixel and 1 (white) representing a foreground pixel. Let $I(x, y)$ be a two-dimensional image and T_l and T_u the lower and upper threshold for a single class of objects. The resulting binary image $I'(x, y)$ is given by:

$$I'(x,y) = \begin{cases} 1 & \text{if } T_l \leq I(x,y) \leq T_u \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

Depending on the type of encoding of the image data, thresholding can be performed based on different visual properties of an object. Working with images that store color information encoded according to the HSV color model, allows to perform thresholding based on the attributes of human color perception, hue, saturation and value (brightness). Consequently the multiple thresholding method was chosen as part of the image processing workflow.

2.4.2 Threshold determination

In order to allow for an easier determination of the thresholds for every color of the marker objects, a semi-automated approach was implemented. In order to obtain information about the colors of the marker objects, calibration sample images were recorded before recording any hand motion sequence, using color sample plates. Only a region that was fully dominated by the color sample was considered for further computation. To achieve a coarse approximation of variation of hue, saturation and

value due to the spherical form of the marker objects, the color sample plates were held at five different orientations.

Histograms were created for the hue, saturation and value channels of every sample image. The lower and upper thresholds for the hue of every marker color were chosen as the lowest and highest bound determined from the five histograms. In the same fashion a lower and upper threshold was determined for saturation and value of every marker color. Due to the controlled environment (see section 5.1.4) the marker objects were the only instances of their specific color in the image, therefore only the lower thresholds for saturation and value have been used.

A manual inspection of the variation of hue, saturation and value for every marker color followed by a comparison with the determined thresholds, yielded an average difference of $\pm 10°$ for the hue and an average difference of max. $\pm 10\%$ for the saturation. Therefore corresponding offsets were added to the determined thresholds. Although an average variation of $\pm 15\%$ was determined for the value of a color, the addition of an offset to the determined threshold, did not fully account for the effect of shadows due to occlusion of the illumination source.

Five different colors that the camera allowed to separate based on the hue range, were chosen for the marker objects attached to the fingers and the wrist reference object. A single color was assigned to a set of four marker objects that approximate the positions of the joints and fingertip of a single finger. The three markers on the wrist reference object were assigned the same color as the thumb, but can be separated with ease due to a bigger size. To fully make use of this color coding scheme, the implemented segmentation procedure produced five separate binary images that contained regions, representing the marker objects of a specific finger. The design of the glove and the wrist fixture used in this thesis are presented in section 5.1.3.

The thresholding operation is highly sensitive to noise. When unfiltered, the noise will lead to oversegmentation. Therefore the application of a smoothing filter, such as a Gaussian filter or a median filter described in the previous section, leads to more consistent segmentation results.

As determined in section 2.2.1, two iterations of the median filter produced an acceptable result, without oversmoothing the image. But because the median filter does not perform very well when a high amount of Gaussian noise is present, the segmented images produced by multiple thresholding might contain segmented noise.

The evaluation of the segmented images produced in the experiments of this thesis, revealed that small amounts of noise persisted along the contour of the segmented regions, producing small defects in the curvature and small holes in the region itself. This effect was amplified by misdetermination of thresholds for the value (brightness) of a color, due to shadows caused by occlusion. To improve the result of the segmentation stage, morphological filters were evaluated.

2.4.3 Morphological filters

Morphological filters are based on set theory. The basic definition of morphological filters considers binary images, which represent two sets, the set of white and the set of black pixels. The fundamental tool of a morphological filter is called a *structuring element*. The structuring element represents a sub-image, which is used to inspect the image for properties defined by the form of that structuring element. The structuring element is required to be a rectangular array, similar to a filter matrix, to allow application to an image. The structuring element contains a center point that is usually located at the center of the array, but the location is problem dependent in general. Figure 2.5 shows an example of a structuring element used for a morphological filter.

The application of a morphological filter to an image, determines whether an image pixel located under the center point of the structuring element should be assigned to the set of white pixels or to the set of black pixels. The distinction depends on the type of the operation. The two fundamental operations, which many morphological filters are based on, are known as *erosion* and *dilation*.

Erosion is the morphological operation that corresponds to the concept of "shrinking" of regions in an image. The erosion operation $\ominus$ is defined as

$$A \ominus S = \{p \in A \mid S_p \subseteq A\} \tag{2.7}$$

where A is a set in a binary image, S is a structuring element and p an image pixel represented by its coordinates. Therefore the resulting set $A \ominus S$ contains a pixel p if and only if, the structuring element with its center point positioned at p is fully contained in A, i.e. is a subset of A.

Dilation can be considered as the counterpart of the erosion operation, as the dilation "grows" the regions in an image. Erosion and dilation are dual operations. In general the effects of the erosion operation cannot be reversed by a subsequent dilation. The dilation operation $\oplus$ is defined as:

$$A \oplus S = \{q \in S \mid S_p \cap A \neq \emptyset\} \tag{2.8}$$

The dilation operation defines the resulting set $A \oplus S$ to contain a pixel q, if at least one pixel of the structuring element S with its center point positioned at p is contained within A, i.e. the cross-section of A and S_p is not an empty set. In simple terms, if the condition is met, the form of the structuring element is replicated, eventually adding new pixels to the set.

Two basic morphological filters based on a combination of erosion and dilation are called the *opening filter* and the *closing filter*. The opening filter applies erosion to an image followed by a dilation of the result, as given by:

$$A \circ S = (A \ominus S) \oplus S \tag{2.9}$$

The closing filter applies the operations in reverse, as given by:

$$A \bullet S = (A \oplus S) \ominus S \tag{2.10}$$

The erosion step of the opening filter has the effect of removing structures from a region that are smaller than the structuring element. The subsequent dilation smoothes the contours of a region. Application of the closing filter to the image results in reparation of small holes and fissures in the contour of the region.

One can generally assume that segmented regions identifying objects of interest in an image, have a much higher connectivity in comparison to noise. Therefore dilation of foreground regions using a suitable structuring element should lead to further removal of noise.

Due to its ability for small repairs of the contour, a closing filter using a structuring element displayed in figure 2.5, was chosen and evaluated as part of the processing workflow.

The evaluation of the results after the application of the closing filter showed that multiple iterations of morphological filtering should be applied with caution. While removing small specks of segmented noise and refining the contour of a region, more than one or two iterations of the closing filter showed a tendency to fuse segmented regions of closely located marker objects. This effect is caused by the dilation operation. Due to the duality of erosion and dilation, the dilation of the foreground results in the erosion of the background. The use of a smaller structuring element (a 3×3 cross-shape) did alleviate the problem to a small extent.

2.5 Feature extraction

Once the recorded images have been segmented into regions, with the foreground regions representing the shape of the marker objects, it is possible to proceed with the next step of finding a suitable form of representation for these regions of pixels.

A region can be identified either by its internal representation or its external representation. The internal representation is useful when the focus lies on characteristics, such as color or texture of a region. The external representation allows to identify a region based on its shape. Gonzales and Woods provide a broad overview in [GW07] (ch.11).

The result of this step should lead to the representation of the marker objects by two-dimensional coordinates at the center of gravity of the specific region. With binary images representing the foundation for this process, the decision was made to focus on the shape characteristics of the regions.

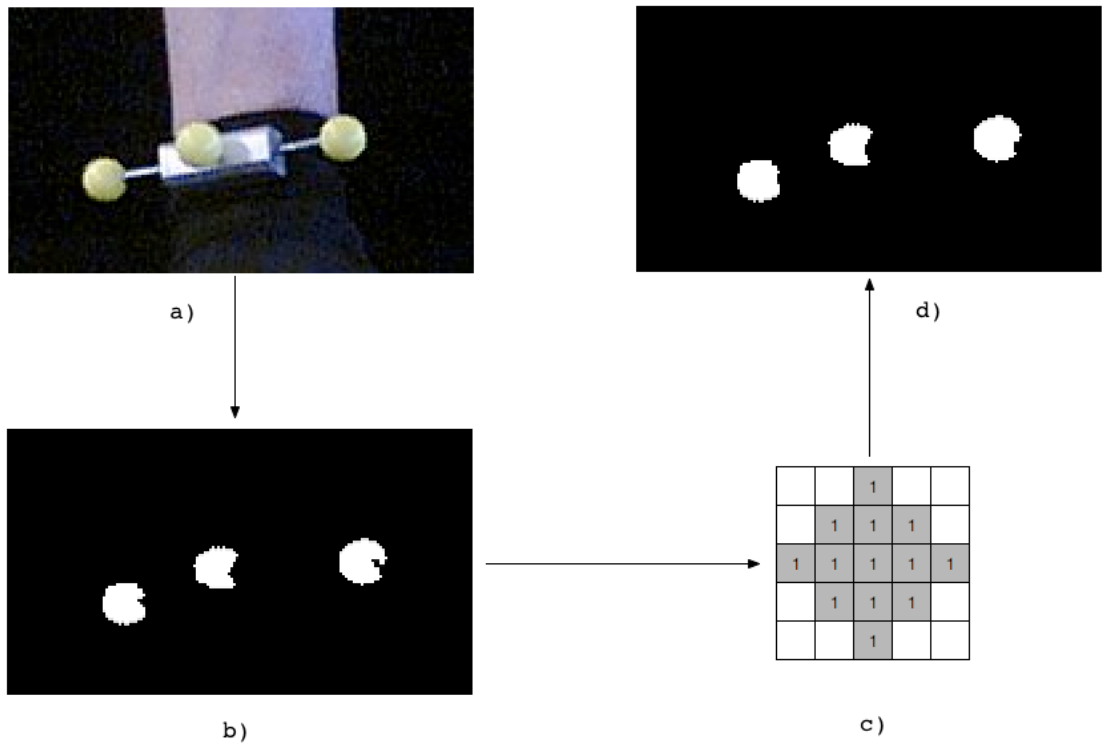


Figure 2.5: Illustration of the effect of a closing filter. **a)** shows a part of the original recorded image. **b)** shows the result of the image segmentation by multiple thresholding. **c)** describes the 5×5 structuring element, which was used with the closing filter, approximating the shape of a circle. **d)** displays the result of the closing operation. The left and right shapes show repaired contours.

2.5.1 Contour finding

A region can be represented by its contour. In the case of a binary image the contour of a region describes the edge between a region that is a foreground object and the background. Multiple methods exist for the purpose of contour finding in binary images, one of the earliest being by Moore [Moo68]. Another method was proposed by Danielsson in [Dan82], which also allows to perform contour finding in non-binary images. Two methods were proposed by Suzuki and Abe [SA85], with one retrieving only the outermost contours of a regions and the second one also retrieving inner contours, due to holes in regions. Haig and Attikiouzel [HA89] proposed a method that improved on the shortcomings of finding of inner contours, exhibited by the previous methods.

For the purpose of this thesis the retrieval of only the outermost contours of a region is of interest, as this leads to a sufficient representation of the shape of a marker object. A side-effect of this approach is robustness to holes in regions due

to unremoved noise.

A contour finding algorithm provided by the OpenCV framework that is based on the algorithm proposed by Suzuki and Abe [SA85], was used in the implementation of the image processing workflow within the software application. The algorithm will be outlined in the following.

The algorithm requires that black pixels represent the background and white pixels represent the foreground regions. Therefore the background pixels are called *0-pixels* and foreground pixels are called *1-pixels*. The algorithm further defines a 1-pixel, which has a 0-pixel in its 4- or 8-neighborhood - depending on the chosen convention - as a border point. The algorithm processes the image according to the following scheme:

Algorithm 1 Find outermost contours in segmented image

1. Initialize the variable $LNBD$ with zero, representing the value of the last encountered non-zero pixel.
2. Execute a raster scan of the image looking for a border point. Update $LNBD$ upon encounter of a non-zero valued pixel.
3. If a border point has been found, check if the following conditions are satisfied:
   - Every pixel $(x, 1) \dots (x, y-1)$ is a 0-pixel.
   - If another border point (x, h) of an outer border with $h < y$ was recently detected, $(x, h+1)$ belongs to the background.
4. If conditions are satisfied, follow the border starting with the border point (e.g. [Moo68]), while updating $LNBD$. Mark each pixel (x, y) of the followed border as follows:
   - If $(x, y+1)$ is a 0-pixel, mark (x, y) with a value of -2.
   - Else mark (x, y) with a value of 2.
5. Store ordered sequence of found border pixels, representing the contour of a region.
6. Resume at 1. until the right bottom corner of the image is reached.

The set of contours that represent the shape of a marker object based on its perimeter, establishes the foundation for the step of shape analysis, in order to obtain the two-dimensional image coordinates of every marker object.

2.5.2 Shape analysis by moments

Computation of moments provides a basic tool for analysis of two-dimensional shapes. Many attributes of a shape can be derived from its moments, such as the area of the shape, its center of gravity (centroid), or its orientation. In [TC88], Teh and Chin present an overview of the different types of moments and their properties. The topic of shape analysis by moments has been approached by many methods. In [Hu62] Hu contributed the definition of moment invariants, based on the geometric moments. The defined moment invariants allow to derive shape features that are invariant to geometric transformations. For this reason it is possible to use moment invariants for the purpose of shape matching. In [Leu91] Leu proposed a method to compute geometric moments of a shape, without the need to involve every pixel, using its boundary only. Jiang and Bunke [JB91] proposed a similar method, which can also be applied to other than the geometric moments.

Only the geometric moments are considered in this thesis, as they are sufficient to obtain the necessary attributes of a marker object shape, in form of its centroid and eccentricity.

Moments are generally classified by their order. The sum $p + q$ of the indices p and q of a moment $m_{p,q}$ represents the order of the moment. Let $f(x, y)$ be a two-dimensional function, then a $(p+q)$-order spacial geometric moment of that function is defined as:

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q \tag{2.11}$$

In the case of a discrete function represented by a two-dimensional image $I(x, y)$, the spacial geometric moment $m_{p,q}$ is given by:

$$m_{p,q} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) x^p y^q \tag{2.12}$$

A central geometric moment corresponds to the spatial geometric moment, reduced by the center of gravity. It therefore expresses the moment in reference to the centroid of the shape and is invariant to translation. The $(p + q)$-order central geometric moment $\mu_{p,q}$ is given by

$$\mu_{p,q} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y)(x - \overline{x})^p (y - \overline{y})^q \tag{2.13}$$

with $\overline{x}$ and $\overline{y}$ representing the coordinates of the shape's centroid, defined as:

$$\overline{x} = \frac{m_{1,0}}{m_{0,0}} \tag{2.14}$$

$$\overline{y} = \frac{m_{0,1}}{m_{0,0}} \tag{2.15}$$

In addition to the coordinates of the shape's centroid, the *eccentricity* of the shape was determined, based on the central geometric moments of the 2nd-order. The eccentricity of a shape describes its deviation from the shape of a circle. In the case of elliptical shape, which is often the shape that most of the spherical marker objects are represented by after segmentation, the eccentricity is defined as the ratio between the distance of both focus points and the major axis of the ellipse. According to [BB08] the eccentricity ϵ of the shape is determined by:

$$\epsilon = \frac{\mu_{2,0} + \mu_{0,2} + \sqrt{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2}}{\mu_{2,0} + \mu_{0,2} - \sqrt{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2}} \tag{2.16}$$

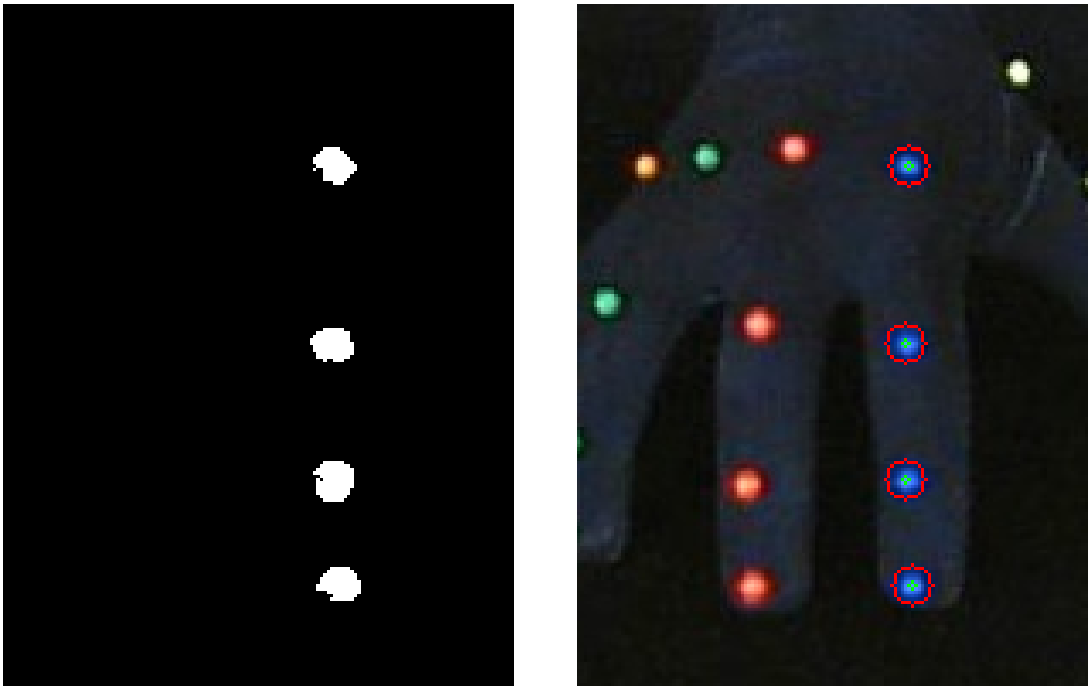
The range of ϵ is $[1, \infty)$, where $\epsilon = 1$ represents a circle, and $\epsilon > 1$ represents an elongated object.

Unfiltered noise, variation in lighting, shadows due to occlusion of the illumination source and occlusion of a marker object by another marker object of the same color, can lead to severe degradation of the shape resulting from the segmentation step. The computed eccentricity attribute allowed to establish a constraint, in order to discard highly non-circular shapes, as these would lead to a high amount of error regarding the position of the marker object.

A visual evaluation of results produced from real image data revealed, that the centroid coordinates provided a good approximation of the center of a marker object, as long as the eccentricity stayed low and the shape showed an elliptical distribution of pixels. Based on visual examination of segmentation results of 50 images, an upper limit of 3 was chosen for the eccentricity attribute. Figure 2.6 shows a result that was achieved using the moment-based approach.

2.6 Conclusion

In this chapter several image processing steps and associated methods have been presented. Following the discussion of various types of image noise, the median filter was chosen for the purpose of noise reduction. The HSV color model was chosen, in order to be able to identify the marker object colors based on the attributes of human color perception. A semi-automated procedure for image segmentation



a **b**

Figure 2.6: Figure **a)** shows a segmentation result based on multiple thresholding. The shapes correspond to the blue marker objects. Figure **b)** shows the computed approximation of the center of the blue marker objects using the moment-based approach. The centroids are colored in green. To allow an easier visual evaluation, the perimeters (red) have been added. The eccentricity attribute for every shape shown in **a)** did not exceed a value of 2.

based on multiple thresholding, using color sample plates for the purpose of initial threshold determination, has been presented. The positive and negative effects of morphological filtering, for the purpose of shape-related refinement of the segmentation result, have been discussed. Spacial and central geometric moments have been chosen as descriptors, in order to retrieve a representation of the marker objects based on their shapes with sub-pixel accuracy, beneficial to the accuracy of results presented in sections 6.2 and 6.3.

The results of this chapter, in form of two-dimensional coordinates that represent the centroids of the marker objects, form the dataset, which is required for the purpose of reconstruction of a three-dimensional representation. The next chapter will discuss the sequence of steps that are necessary, in order to obtain three-dimensional positions of marker objects, using a stereo vision system with three camera views. An evaluation of the overall performance of the image processing workflow in combination with the three-dimensional reconstruction is presented in sections 6.1 and 6.2.

Stereo vision workflow

3

Due to the ubiquity of a wide range of camera devices, which can be attached to a computer, it is no longer a difficult task to make a computer "see". But many tasks, we as human beings perceive as trivial, require the application of a complex processing workflow to the image data, acquired from a vision system attached to a computer.

A single camera device provides a two-dimensional view of a scene. The domain of image processing provides a wealth of different methods, which allow to extract scene specific information from a recorded image. Due to the projective nature of image formation, the information obtained from a recorded image represents flat information. Without any knowledge about the geometry of the scene, the reconstruction of a three-dimensional representation corresponding to two-dimensional information extracted from a single view, does not lead to a useful result, as the reprojection of a two-dimensional point is a ray in three-dimensional space.

To successfully obtain three-dimensional information of a viewed scene, more than one camera device is necessary. With at least two views, it becomes possible to obtain the three-dimensional location of a point in the viewed scene, based on its two-dimensional projections onto the image planes of the cameras. A binocular camera setup approximates the visual stereo system of the human being, which acquires retinal images of the viewed scene using the visual cortex to obtain a perception of depth.

The problem of stereo vision has been covered extensively, with a wealth of existing methods for every step of the process of three-dimensional reconstruction. This chapter will discuss the fundamental geometric concepts that represent key elements in the reconstruction process, based on a stereo vision system using two-dimensional point data, extracted through processing of the recorded images, as previously described in chapter 2. A sequence of steps will be presented that are necessary, in order to obtain three-dimensional positions of the marker objects attached to the glove. Methods that were used in this thesis will be presented - starting with the fundamental process of calibration of a single camera, followed by the epipolar geometry of a stereo system and the process of image rectification and concluding with the process of triangulation.

3.1 The camera model

To describe the geometry of projective image formation it is necessary to define a camera model used in the process. The *pinhole camera model* is the simplest camera model available and is well-suited for describing the fundamentals of projective geometry.

The pinhole camera model consists of a *center of projection* c (the pinhole) and an image plane. The image plane is located behind the center of projection at a distance f, known as the *focal length*. The image plane is perpendicular to the optical axis going through the center of projection. The projective geometry of the pinhole camera model is completely determined by the choice of a center of projection and the position of an image plane. Upon entering the camera, a ray of light from any particular point $P_w \in \mathbb{R}^3$ intersects the image plane, leading to a projected representation $p_c \in \mathbb{R}^2$. Figure 3.1 illustrates the projective geometry of the rearranged pinhole camera model.

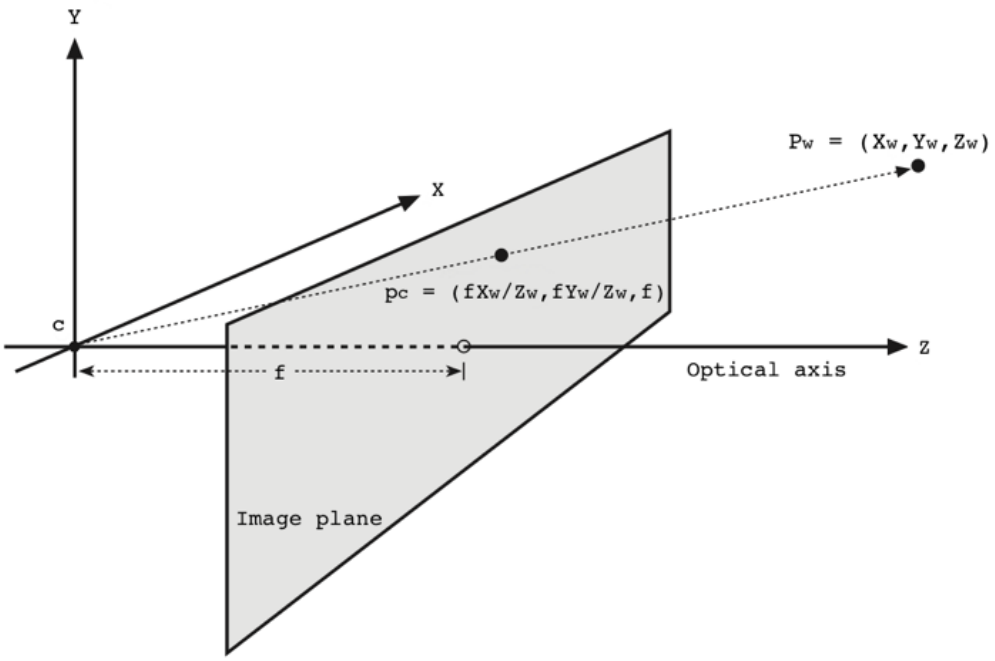


Figure 3.1: Illustration of the rearranged pinhole camera model. The center of projection and the image plane determine the projective geometry of the model. The projection of point P_w intersects the image plane, located at distance f from the center of projection, resulting in the projected point p_c. [Adapted from [BK08]]

Given the three-dimensional representation of a point $P_w = (X_w, Y_w, Z_w)^T$, the projected representation p_c is given by:

$$P_w = (X_w, Y_w, Z_w)^T \mapsto p_c = (f\frac{X}{Z}, f\frac{Y}{Z}, f)^T \tag{3.1}$$

The opening at the center of projection is known as aperture. It is assumed to be infinitely small. Due to this circumstance, the resulting image is always in focus. The pinhole camera does not account for a lens - it describes a rectilinear system, therefore the resulting image is free from any distortion.

The projection of $P_w \in \mathbb{R}^3$ into $\mathbb{R}^2$ leads to loss of depth information, as all projected points obtain the same depth coordinate. To approach the task of three-dimensional reconstruction, one of the fundamental problems that needs to be solved, is the determination of the internal and external camera geometry, represented by the intrinsic and extrinsic camera parameters. This is described as the process of camera calibration.

3.2 Camera calibration

The basic idea of the process of camera calibration is to derive the projection equations between known three-dimensional coordinates of a set of points and their respective two-dimensional projections. Based on these equations it is possible to solve for the camera parameters. Knowing the camera parameters it becomes possible to reverse the projective transformation and obtain the three-dimensional representation of a point, based on its two-dimensional projection. The accuracy of the three-dimensional reconstruction directly depends on the correctness and stability of the calibration, the calibration process therefore plays a key role in computer vision.

Many approaches to camera calibration exist. The two approaches most often cited in literature that are also widely used in practice are the ones by Tsai [Tsa87] and Zhang [Zha00]. While Tsai's approach uses a set of points with known world coordinates, Zhang's approach is based upon the knowledge of the geometry of the pattern used on a calibration object. Working with a calibration object all that is necessary is the knowledge about the geometry of the calibration pattern. It is necessary to provide at least two different images of the calibration object to be able to compute the camera parameters.

In the following this section will outline the approach described in [Tsa87], whereas the approach proposed in [Zha00] will be looked at in more detail, as it is the approach used in this thesis.

3.2.1 Camera parameters

In order to establish the projective transformation between the three-dimensional space and the two-dimensional image plane it is necessary to determine two sets of

parameters - the *intrinsic* and the *extrinsic* parameters. The camera parameters obtained for the three-camera experimental setup used in this thesis are presented in section 6.2.1.

Intrinsic parameters

The intrinsic parameters describe the mapping of optical rays onto the image plane resulting in two-dimensional pixel coordinates. Therefore these parameters reflect the internal camera geometry including the lens. The intrinsic parameters are represented by the following 3×3 matrix:

$$A = \begin{bmatrix} \alpha_x & \gamma & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

The matrix A is called the *camera calibration matrix*. The parameters $\alpha_x = fm_x$ and $\alpha_y = fm_y$ describe the focal length of the camera in terms of pixel dimensions, with m_x and m_y representing the number of pixels per unit of distance in the X and Y direction. γ describes the skew between the axes of the coordinate frame. The skew can usually be assumed to be zero for most of the cameras. c_x and c_y represent the principal point position as the offset from the origin of the image plane. Furthermore α_x/α_y represents the aspect ratio of the image.

For a camera with a fixed optical system the set of intrinsic parameters remains identical for every image recorded. In the case of a zoom lens (varifocal/parfocal) the focal length and the principal point can vary.

Extrinsic parameters

The extrinsic parameters relate the orientation and position of the camera coordinate frame to a global world coordinate frame. The respective transformations are given by a 3×3 rotation matrix and a 3×1 translation vector. It is possible to combine the rotation and translation into a single transformation step of the form

$$p'_c = [R \mid t] P_w \tag{3.3}$$

where p'_c represents the projected point in the coordinate frame of the camera. The extrinsic parameters combined with the set of intrinsic parameters given by the camera calibration matrix, form the *camera projection matrix*:

$$M = A[R \mid t] \tag{3.4}$$

The 3×4 camera projection matrix M describes the projective transformation of a specific camera. Given the camera projection matrix it is possible to calculate the two-dimensional projection p_c in pixel coordinates of a point P_w in metric world coordinates by the following equation:

$$p_c = MP_w \tag{3.5}$$

3.2.2 Distortion coefficients

Any lens used in a camera introduces non-linear distortions into the projection resulting in a deviation from the projection described by the ideal pinhole camera model. A wide-angle lens or fish-eye lens are good examples. Distortions belong to the category of optical aberrations. While distortions affect the image geometry, they do not affect the quality of the image. The process of camera calibration includes the estimation of the distortion coefficients based on a distortion model to facilitate distortion removal. Formally the distortion coefficients belong to the set of intrinsic camera parameters. The estimation will be looked at in more detail in the following sections.

There are two main forms of lens distortion, *radial* and *tangential* distortion. Figure 3.2 illustrates both of these forms of distortion.

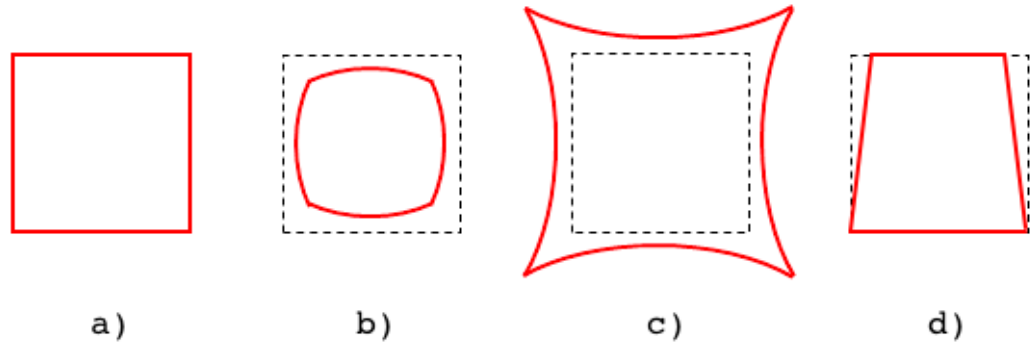
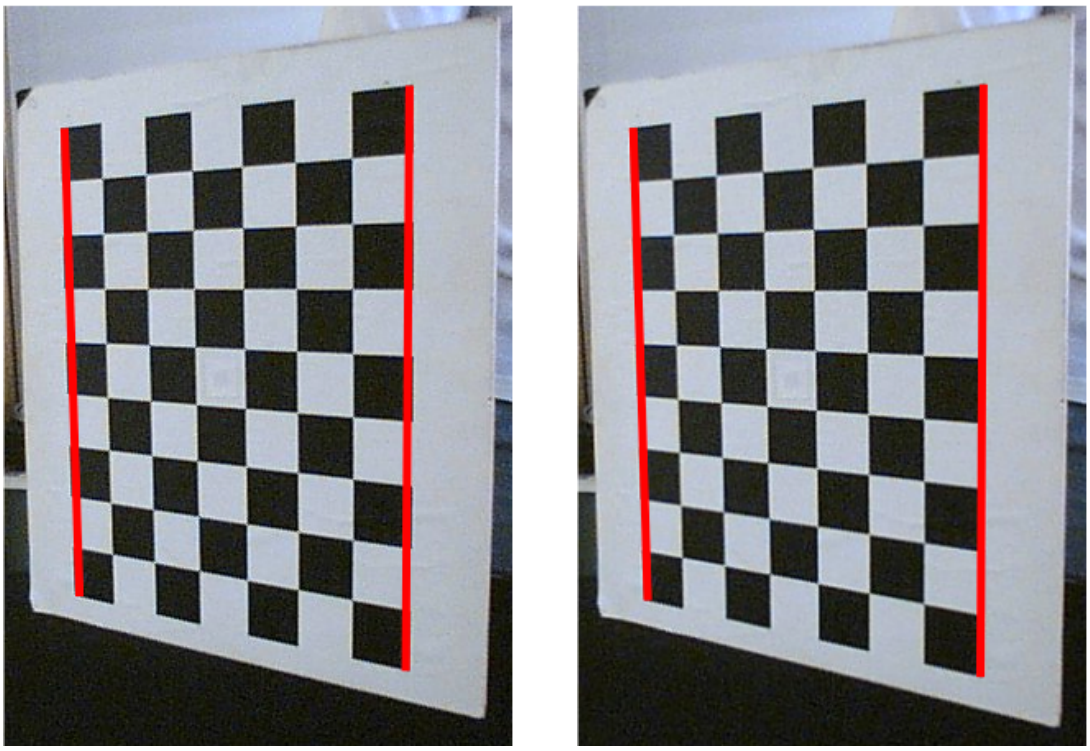


Figure 3.2: Illustration of the two main forms of lens distortion. Figure **a)** shows the undistorted image form. Figures **b)** and **c)** show the negative barrel distortion and the positive pincushion distortion, as two main forms of radial distortion. Figure **d)** shows the tangential distortion.

The radial form of distortion results in displacement of image points from their correct positions along the radial axis from the principal point of the image plane. Radial distortion is generally caused by the flawed radial curvature of the lens. Tangential distortion results in displacement of the image points along the tangent of a circle centered at the principal point. The tangential form of distortion is a

result of imperfect centering of the lens components and lens/imager alignment. In contrast to radial distortion, the effects of tangential distortion in practice are negligible and therefore disregarded by most calibration methods.

The radial distortion occurs in two distinctive forms: the *barrel distortion* and the *pincushion distortion*. The barrel distortion represents a negative displacement - decrease in image magnification - of image point positions with increasing distance from the principal point. Figure 3.3 a) shows the amount of barrel distortion introduced by the lens of the camera used in the experimental setup. The pincushion distortion represents a positive displacement - increase in image magnification - of image point positions with increasing distance.



a **b**

Figure 3.3: A small amount of barrel distortion, introduced by the lens of the cameras used in the experimental setup, can be seen in figure **a)**. Note the edges of the black squares showing along the red lines. Figure **b)** shows the result after distortion removal.

3.2.3 Calibration method by Tsai

The calibration method proposed by Tsai in [Tsa87] is based on the use of a set of non-coplanar points. The accurate three-dimensional coordinates of the used set of points must be known before the calibration. It is necessary to use at least seven points to enable the parameter estimation. An increasing number of points will facilitate a more precise estimation of the parameter sets through minimization of the error.

The calibration method consists of four steps, the succession of which represents the inspection of the image formation process. This is done in order to establish the equations for the estimation of the parameter sets. The steps will be outlined in the following.

The first step considers the transformation from the three-dimensional world coordinate system into the three-dimensional coordinate frame of the camera. The transformation is given by:

$$p'_c = R\,P_w + t \tag{3.6}$$

$P_w = (X_w, Y_w, Z_w)^T$ and $p'_c = (x, y, z)^T$ represent the three-dimensional points in the world coordinate frame and the camera coordinate frame. The rotation matrix R and the translation vector t can be estimated, given the set of non-coplanar three-dimensional points along with their projections.

The second step looks upon the mapping of the three-dimensional point P_w onto the two-dimensional image plane with the help of the perspective projection, as it is described for the pinhole camera model. The resulting ideal image coordinates X_u and Y_u of the projection are given by:

$$X_u = f\,\frac{x}{z} \tag{3.7}$$

$$Y_u = f\,\frac{y}{z} \tag{3.8}$$

Given the set of non-coplanar points and their projections, this step yields the focal length of the camera.

The third step handles the task of estimating the distortion. In addition to the intrinsic and extrinsic parameters, Tsai's calibration method estimates two coefficients of the radial distortion. Tsai uses a simplified form of the distortion model proposed in [Bro71], which describes a polynomial approximation. Based on the finding that the tangential distortion is negligible in practice, only the radial form is modeled. Tsai further states that the radial distortion is dominated by the first term and that

a more complex model could cause numerical instability. The distortion is defined by the following relation:

$$X_u = X_d + D_x \tag{3.9}$$

$$Y_u = Y_d + D_y \tag{3.10}$$

X_d and Y_d represent the distorted image coordinates and D_x and D_y represent factors given by the distortion model as follows:

$$D_x = X_d \left(k_1 \, r^2 + k_2 \, r^4 + \ldots\right) \tag{3.11}$$

$$D_y = Y_d \left(k_1 \, r^2 + k_2 \, r^4 + \ldots\right) \tag{3.12}$$

$$r = \sqrt{X_d^2 + Y_d^2} \tag{3.13}$$

The fourth step of the calibration method examines the mapping of the projection of P_c from metric image coordinates to normalized pixel coordinates.

$$X_f = s_x \, d_x^{'-1} \, X_d + C_x \tag{3.14}$$

$$Y_f = d_y^{-1} \, Y_d + C_y \tag{3.15}$$

C_x and C_y represent the position of the principal point in pixels. d_x and d_y represent the metric distance between two adjacent sensor pixels in the $X-$ and $Y-$direction. The derivative of d_x^{-1} is given by: $d_x^{'-1} = d_x \, N_{cx}/N_{fx}$, with N_{cx} being the number of pixels in the $X-$direction and N_{fx} being the number of sampled pixels in a line. s_x represents a scaling factor to accommodate the uncertainty introduced by a possible deviation in the process of image acquisition.

The result of the outlined sequential examination of the image formation process, enables the estimation of the set of extrinsic parameters followed by the estimation of the set of intrinsic parameters.

3.2.4 Calibration method by Zhang

The calibration method proposed by Zhang in [Zha00] is based on the use of a planar calibration object with a checkerboard pattern, the geometry of which is known. The checkerboard pattern is used to extract a set of calibration points. Figure 3.4 shows

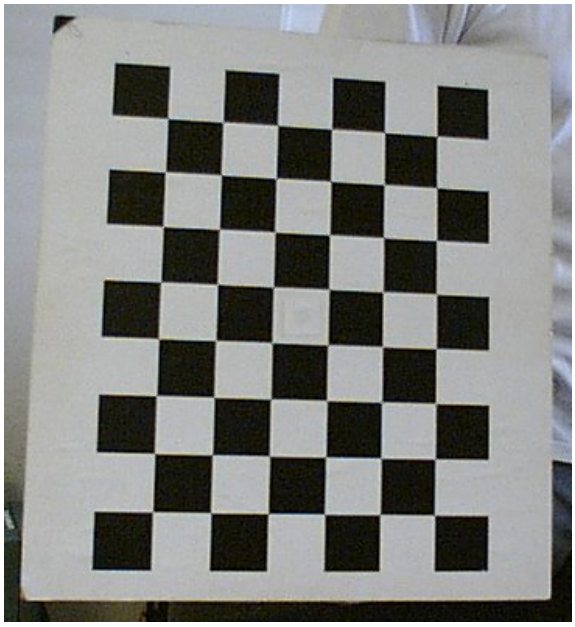


Figure 3.4: The planar checkerboard calibration pattern that was used to calibrate the experimental camera setup.

the calibration object used in the calibration process of the experimental camera setup of this thesis.

At least two views of the calibration object at a different orientation are necessary to enable the estimation of the set of intrinsic parameters including the distortion coefficients, as well as the set of extrinsic parameters. Practice shows that far more than two images are needed due to inaccuracy of point extraction caused by noise. An increasing amount of images of the calibration object at different orientations results in an increase of accuracy in the estimation of the intrinsic and extrinsic parameters.

Zhang's calibration method is often cited in literature and widely used in practice due to its robustness and flexibility. A very popular version of this method has been implemented by Jean-Yves Bouguet in the form of the Camera Calibration Toolbox for Matlab® [Bou10]. The Camera Calibration Toolbox was used to obtain the camera calibration parameters of the experimental three-camera stereo vision setup described in section 5.1.

The initial step of the calibration method consists of the estimation of a homography between the plane of the calibration object and its image. To estimate the homography, without loss of generality, the assumption is made that the plane of the calibration object is positioned on $Z = 0$ of the world coordinate system. Therefore the projected point $p_i = (u, v, 1)^T$ of a point $P_w = (X, Y, Z, 1)^T$ in the calibration object plane, based on the perspective projection of the camera, is given by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[r_1\ r_2\ r_3\ t] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A[r_1\ r_2\ t] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{3.16}$$

r_1, r_2 and r_3 are the columns of the rotation matrix R and t is the translation vector. The set of intrinsic parameters is given by the matrix A:

$$A = \begin{bmatrix} \alpha & \gamma & v_0 \\ 0 & \beta & u_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

Based on (3.16) the homography H as a 3×3 matrix, is given by:

$$H = A[r_1\ r_2\ t] \tag{3.18}$$

Therefore based on (3.18) the relation of point P_w and its image p_i is given by:

$$p_i = HP_w \tag{3.19}$$

In practice the extracted image points can not be assumed to be absolutely accurate due to noise. Thus the extracted points do not satisfy (3.19) and the homography estimation results in a non-linear least-squares problem.

The notation of the homography as $H = [h_1\ h_2\ h_3]$ followed by a substitution in (3.18) results in:

$$[h_1\ h_2\ h_3] = A[r_1\ r_2\ t] \tag{3.20}$$

With the knowledge that the rotational axes r_1 and r_2 are orthonormal, it is possible to derive two constraints for the intrinsic parameters:

$$h_1^T A^{-T} A^{-1} h_2 = 0 \tag{3.21}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \tag{3.22}$$

Through the proposed closed-form solution it is now possible to effectively estimate the intrinsic parameters:

$$B = A^{-T}A^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \tag{3.23}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta_2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta_2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \tag{3.24}$$

The intrinsic parameters can be extracted from B as follows:

$$v_0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \tag{3.25}$$

$$\lambda = B_{33} - (B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23}))/B_{11} \tag{3.26}$$

$$\alpha = \sqrt{\lambda/B_{11}} \tag{3.27}$$

$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{3.28}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda \tag{3.29}$$

$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda \tag{3.30}$$

Once the intrinsic parameters are known, the extrinsic parameters can be calculated based on (3.20):

$$r_1 = \lambda\, A^{-1}\, h_1 \tag{3.31}$$

$$r_2 = \lambda\, A^{-1}\, h_2 \tag{3.32}$$

$$r_3 = r_1 \times r_2 \tag{3.33}$$

$$t = \lambda\, A^{-1}\, h_3 \tag{3.34}$$

Distortion estimation

As described in the previous section 3.2.2 any lens introduces non-linear distortions into the projection resulting in a deviation from the projection described by the ideal pinhole camera model. Thus far the calibration method has estimated the

intrinsic and extrinsic parameters of the ideal pinhole camera model. To allow the application of the perspective projection of the pinhole model it is necessary to remove the distortion beforehand.

As previously described the tangential distortion is negligible in practice. Zhang follows the findings of [Tsa87], assuming that the radial distortion is dominated by the first term and a more elaborate modeling could cause numerical instability. Zhang's calibration method therefore builds upon the distortion model presented in [Tsa87] and only considers the radial distortion estimating the first two coefficients. The claim about the tangential distortion being negligible is validated by the calibration results obtained for the experimental setup, which are presented in section 6.2.1.

With the assumption that (x, y) are the ideal undistorted image coordinates of a projected point according to the pinhole camera model, and (x', y') are the observed distorted image coordinates of the projection, the distortion is given by following model:

$$x' = x + x\,(k_1\,(x^2 + y^2) + k_2\,(x^2 + y^2)^2) \tag{3.35}$$

$$y' = y + y\,(k_1\,(x^2 + y^2) + k_2\,(x^2 + y^2)^2) \tag{3.36}$$

k_1 and k_2 are the radial distortion coefficients. The center of the radial distortion is located at the principal point. It is further assumed that (u, v) are the ideal image pixel coordinates of the projection and (u', v') are the distorted image pixel coordinates. Based on $u' = u_0 + \alpha\,x' + \gamma\,y'$, $v' = v_0 + \beta\,y'$ and the assumption of a negligible skew, $\gamma = 0$, the distortion model for pixel coordinates is given by:

$$u' = u + (u - u_0)\,(k_1\,(x^2 + y^2) + k_2\,(x^2 + y^2)^2) \tag{3.37}$$

$$v' = v + (v - v_0)\,(k_1\,(x^2 + y^2) + k_2\,(x^2 + y^2)^2) \tag{3.38}$$

Given the distortion model, and the previously estimated intrinsic parameters, which will give the ideal pixel coordinates, it is now possible to solve for the distortion coefficients.

To achieve a faster convergence Zhang proposes the estimation of the complete set of parameters through a maximum likelihood estimation of the following equation:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||p_{ij} - p'(A, k_1, k_2, R_i, t_i, P_j)||^2 \tag{3.39}$$

$p'(A, k_1, k_2, R_i, t_i, P_j)$ represents the distorted projection of point P_j. The solution of this non-linear minimization problem is conducted by using the Levenberg-Marquardt algorithm [Mor78]. The Levenberg-Marquardt algorithm, like any other

iterative minimization algorithm requires an initial guess to initialize the minimization. According to [Zha00] the estimated intrinsic and extrinsic parameters can be used as the initial guess. It is also possible to initialize the distortion coefficients with 0.

3.3 Transition to stereo vision

Up to this point the calibration process of a single camera has been described, yielding the set of intrinsic parameters including the distortion coefficients and the extrinsic parameters in reference to the calibration object.

For a stereo vision system to be usable, it is necessary to determine the relative position and orientation of the coordinate frame of one camera with respect to the other, represented by a euclidean transformation. The calibration parameters obtained for both stereo pairs of the experimental setup used in this thesis are presented in section 6.2.

Placing the calibration object at different orientations in both views of the stereo vision setup simultaneously, allows to obtain corresponding sets of points in both views in reference to the calibration object. Based on these sets of corresponding points, it is possible to estimate an algebraic representation of the geometry of a given stereo vision system, represented by either the so-called fundamental matrix, or the essential matrix. While it is possible to derive the euclidean transformation from one camera coordinate frame to the coordinate frame of the other camera from the fundamental matrix, the essential matrix directly describes the transformation given by a rotation and translation.

The following sections will present two stereo geometry systems that describe the geometry of a binocular setup. Following that description, the role of the fundamental matrix and the essential matrix in the description of the geometry of a stereo vision system will be explained. The eight-point algorithm, an approach to the estimation of either one of the matrices, will be described.

3.3.1 Standard stereo geometry

The standard stereo geometry describes the idealized configuration of two cameras such that the following requirements are satisfied:

- Identical internal geometry of both cameras (intrinsic parameters)
- Coplanarity and row-alignment of both image planes

Coplanar and row-aligned image planes imply the parallel alignment of the optical axes of both cameras, and their perpendicularity to the baseline of the stereo vision

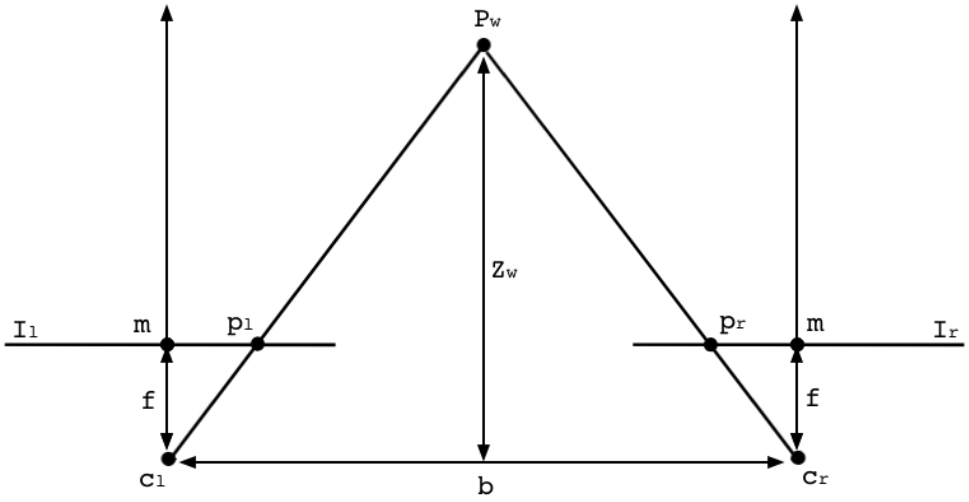


Figure 3.5: Basic schematic of two cameras in a standard stereo geometry configuration. Both cameras have the same intrinsic parameters. The image planes I_l and I_r are coplanar and row-aligned. The optical axes are parallel to each other and perpendicular to the baseline b.

system. Figure 3.5 illustrates the configuration described by the standard stereo geometry.

As shown in the illustration the baseline b represents the horizontal translation between the center of projection c_l of the left camera and the center of projection c_r of the right camera. f is the focal length and $m = (m_x, m_y)^T$ represents the principal point. Given that the images are row-aligned, the projection $p_l = (x_l, y_l)^T$ of a three-dimensional point $P_w = (X_w, Y_w, Z_w)^T$ in the left image with the horizontal coordinate x_l, has a corresponding projection $p_r = (x_r, y_r)^T$ of P_w in the right image with the horizontal coordinate x_r, located in the exact same row. Due to this circumstance, the search for corresponding points in a standard stereo geometry system is a one-dimensional problem. Furthermore two corresponding projections p_l and p_r are constrained by $x_l \geq x_r$.

The difference $d = x_l - x_r$ is called disparity. Given the disparity of two corresponding projections p_l and p_r, it is possible to reconstruct the three-dimensional representation of $P_w = (X_w, Y_w, Z_w)^T$. Assuming the left camera as the frame of reference the three-dimensional coordinates of P_w are given by:

$$X_w = \frac{(x_l - m_x)b}{d} \tag{3.40}$$

$$Y_w = \frac{(y_l - m_y)b}{d} \tag{3.41}$$

$$Z_w = \frac{fb}{d} \tag{3.42}$$

The depth behaves inversely proportional to the disparity. A small disparity signifies a higher depth with coarse depth resolution, whereas a higher disparity signifies a point closer to the cameras with fine depth resolution.

Due to various tolerances an actual physical stereo vision system with a standard stereo geometry configuration is impossible to build. It is therefore always necessary to obtain the configuration of the system through calibration. The following section will discuss the geometry of a general stereo vision system. Nevertheless it is desirable to approximate the standard stereo geometry configuration to simplify the search for corresponding points. This is possible through image rectification and will be discussed in section 3.4.

3.3.2 Epipolar geometry

The configuration of a general stereo vision system does not satisfy any of the constraints of the standard stereo geometry. Usually the cameras are configured in a convergent configuration. They are aligned towards a specific scene or object and therefore provide a maximized area of stereo overlap allowing to fully capture the specific scene or object.

Apart from the different internal geometry, the relation between the cameras is given by an euclidean transformation, represented by a 3×3 rotation matrix R and a 3×1 translation vector. With the left camera as the frame of reference, the relation between two projections is given by:

$$p'_r = Rp'_l + t \tag{3.43}$$

p'_l and p'_r represent the projections of the three-dimensional point P_w in camera coordinates.

Due to the rotated orientation of the image planes, the search for a corresponding projection p_r of a point P_w in the right image based on its projection p_l in the left image, becomes a two-dimensional problem.

The geometry of a general stereo vision system is described by the *epipolar geometry*. Figure 3.6 shows an illustration of the epipolar geometry describing a general stereo

vision system. The configuration of the experimental stereo vision setup is presented in section 5.1.2.

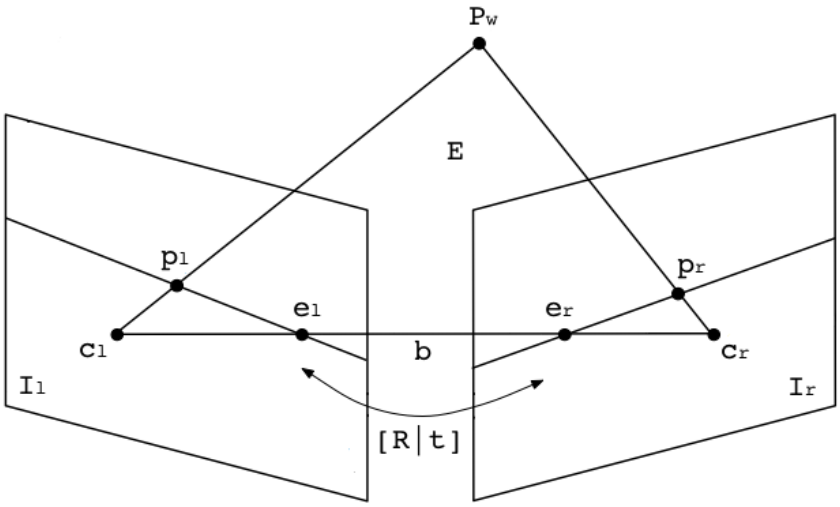


Figure 3.6: Illustration of two cameras in a general stereo geometry configuration, described by epipolar geometry. The cameras are aligned towards a specific scene or object. The optical axes of the cameras intersect in a point P_w. Point P_w and the epipoles e_l and e_r span the epipolar plane E. The projections of P_w given by p_l and p_r lie along the epipolar lines l_l and l_r. [Adapted from [Sch05]]

The rotation of the image planes towards each other results in the intersection of both image planes with the baseline b. The resulting points of intersection are called the epipoles e_l and e_r. They can be viewed as the projection of the center of projection of the other camera. Every three-dimensional point P_w in view of both cameras, together with the epipoles e_l and e_r spans an epipolar plane E. The spanned epipolar plane intersects both images resulting in corresponding epipolar lines l_l and l_r.

All points lying in the epipolar plane have a projection in both image planes along the respective epipolar line. This gives rise to the epipolar constraint:

- Given a projection p_l of a point P_w in the image of the left camera, the corresponding projection p_r in the right image must lie along the corresponding epipolar line l_r.

Therefore in reference to the search of corresponding points p_l and p_r the epipolar geometry allows to reduce the search space in to a line.

Fundamental matrix

The fundamental matrix represents a complete description of the epipolar geometry of a stereo vision system in an algebraic form. The fundamental matrix, proposed by Luong and Faugeras in [LF96], is a basic tool for the description of a relation between the cameras of a stereo vision system. The matrix fully determines the geometric relation of two cameras in projective space.

The fundamental matrix F is a singular 3×3 homogeneous matrix of rank 2 with seven degrees of freedom that satisfies the epipolar constraint, represented as:

$$p_r^T F p_l = 0 \tag{3.44}$$

p_l and p_r represent the projections in pixel coordinates of a three-dimensional point P_w in the left and right camera of the stereo vision system. Let A_l and A_r be the intrinsic matrices of the left and the right camera, and p_l' and p_r' the projections of the three-dimensional point P_w in metric camera coordinates. The relation between the projections is given by:

$$p_l' = A_l^{-1} p_l \tag{3.45}$$

$$p_r' = A_r^{-1} p_r \tag{3.46}$$

With (3.45) and the extrinsic parameters of the euclidean transformation between the cameras given by the rotation matrix R and the translation vector t, the fundamental matrix can be written as follows:

$$F = A_r^{-T} R [t]_\times A_l^{-1} \tag{3.47}$$

The fact that the fundamental matrix contains the intrinsic parameters of both cameras as well as the extrinsic parameters of the euclidean transformation between the cameras, allows to describe the epipolar geometry of a stereo vision system without any prior knowledge through estimation of the fundamental matrix. The fundamental matrix therefore allows to describe the epipolar geometry of an uncalibrated stereo vision system. Equally it is possible to derive the camera matrices for the left and right camera of the stereo vision system from the fundamental matrix.

The fundamental matrix can be estimated using a set of corresponding points. Every pair of corresponding points $p_l(x_l, y_l)$ and $p_r(x_r, y_r)$ satisfying (3.44) establishes a linear homogeneous equation of the following form:

$$\begin{aligned} x_l\, x_l\, F_{11} + x_l\, y_r\, F_{21} + x_l\, F_{31} + y_l\, x_r\, F_{12} + \\ y_l\, y_r\, F_{22} + y_l\, F_{32} + x_r\, F_{13} + y_r\, F_{23} + F_{33} = 0 \end{aligned} \tag{3.48}$$

A set of i pairs of corresponding points forms a linear homogeneous system:

$$Mf = 0 \tag{3.49}$$

M represents a $i \times 9$ matrix with i rows of the form:

$$M_i = [x_l x_l \;\; x_l y_r \;\; x_l \;\; y_l x_r \;\; y_l y_r \;\; y_l \;\; x_r \;\; y_r \;\; 1] \tag{3.50}$$

f represents a vector containing the nine entries of the fundamental matrix F:

$$f = [F_{11}\; F_{21}\; F_{31}\; F_{12}\; F_{22}\; F_{32}\; F_{13}\; F_{23}\; F_{33}] \tag{3.51}$$

Different methods exist to estimate the fundamental matrix: linear, iterative and non-linear. The *eight-point algorithm* describes a simple algorithm to a linear solution of the fundamental matrix based on a set of at least eight corresponding points. Initially introduced by Longuet-Higgins in [LH81], the eight-point algorithm was modified by Hartley in [Har97], introducing normalization of the coordinates of the corresponding point pairs to increase the stability of the solution. The eight-point algorithm will be outlined in the following.

The eight-point algorithm begins with the application of the normalization transformations N_l and N_r to the appropriate set of points. The transformation shifts the centroid of the appropriate set of points to the origin and equally scales the coordinates of every point, so that the average distance to the origin is that of $\sqrt{2}$.

The entries of F are then calculated through singular value decomposition of M given by:

$$M = UDV^T \tag{3.52}$$

The entries of F are found in the column of V corresponding to the least singular value of M. The resulting fundamental matrix generally will not be of rank 2 and will therefore not be singular. Thus the singular value decomposition of $F = U\,D\,V^T$ is computed, followed by setting the smallest singular value in the diagonal of D to 0 resulting in D', to enforce the singularity constraint. The corrected estimate of the fundamental matrix F is then given by:

$$F' = UD'V^T \tag{3.53}$$

In a last step the estimated fundamental matrix F' must be denormalized to remove the relation to normalized point coordinates. This is done with $F = N_r^{-1}\, F'\, N_l^{-1}$.

According to [Har97] the eight-point algorithm is very sensitive to noise in the provided set of corresponding points. One proposed solution to remedy this problem is the *RANSAC-algorithm (Random Sample Consensus)*.

The RANSAC-algorithm, proposed by Fischler and Bolles in [FB81] is an iterative non-linear parameter estimation algorithm based on data resampling, that is able to cope with a large amount of outliers in the provided dataset. Contrary to techniques such as least-squares optimization, which uses the full dataset, the RANSAC-algorithm only uses the amount of samples needed to provide an estimate of the set of parameters. Based on the evaluation of the estimated parameters the algorithm changes its selection of data samples, discarding outliers and selecting new samples, and performs the estimation and evaluation procedure again, until it fulfills a termination criteria. In combination with the eight-point algorithm, the RANSAC-algorithm can be used to provide an optimized estimate of the fundamental matrix based on a noisy set of pairs of corresponding points. For more information on the RANSAC-algorithm see [FB81].

Essential matrix

The concept of the essential matrix was proposed by Longuet-Higgins in [LH81], prior to the concept of the fundamental matrix. In the same way as the fundamental matrix, the essential matrix represents the epipolar geometry of a stereo vision system, establishing the epipolar constraint as:

$$p_r^{'T} E p_l^{'} = 0 \tag{3.54}$$

$p_l^{'}$ and $p_r^{'}$ represent the projections of a three-dimensional point P_w in camera coordinates. The difference between the essential matrix and the fundamental matrix is that the essential matrix requires the stereo vision system to be calibrated. Thus the relation between the essential matrix and the fundamental matrix is given by:

$$E = A_r^T F A_l \tag{3.55}$$

The representation of the fundamental matrix given by equation (3.47), allows to write the essential matrix as:

$$E = R[t]_\times \tag{3.56}$$

For a stereo vision system with intrinsic parameters of both cameras obtained through calibration, the essential matrix describes the relative position and orientation of both cameras in form of the extrinsic parameters of the stereo vision

system represented by the rotation matrix R and the translation vector t. The estimation of the essential matrix can be conducted using the same algorithms that are used for the estimation of the fundamental matrix.

As presented in this section, the epipolar geometry of a stereo vision system can be used to describe the relation between the two projections of a three-dimensional point. Although the epipolar constraint is sufficient to determine the correspondence of two projections of the same point, the search is non-trivial. The next section will discuss the process of image rectification, which allows to simplify the search for correspondence.

3.3.3 Three-camera experimental setup

In order to conduct three-dimensional reconstruction using the experimental setup with three cameras (see section 5.1.2), the setup was split into two logical stereo pairs. The middle camera was chosen to provide the frame of reference for both stereo pairs. Pairwise stereo calibration was conducted using the Matlab® Camera Calibration Toolbox [Bou10], in order to obtain the intrinsic parameters for all three cameras, as well as the euclidean transformations between both cameras of each stereo pair. Results of the pairwise stereo calibration are presented in section 6.2.1.

The perspective projection matrices of the reference camera and the other camera of a logical stereo pair were assumed as:

$$M_{ref} = A_{ref}[I \mid 0] \tag{3.57}$$

$$M_{oth} = A_{oth}[R \mid t] \tag{3.58}$$

Using these projection matrices, a rectification of the images was obtained for both logical stereo pairs, according to the image rectification process described in the following section.

3.4 Image rectification

Within a stereo vision system one of the main questions is how to establish a correspondence between the projections of a three-dimensional point onto the image planes of the left and right camera. If a point in either one of the images is known, the epipolar constraint $p_r^T F p_l = 0$ defined by the epipolar geometry described in the previous section, allows to restrict the search space for the corresponding point in the other image to the corresponding epipolar line. Nonetheless the search is non-trivial and requires the computation of the epipolar line. The rectification of both

images allows to ease the search, by transforming an arbitrarily configured stereo vision system into a standard stereo geometry system. Figure 3.7 illustrates the basic idea of image rectification. While the search space for a corresponding point is still restricted to a line, with the line being the exact same image row as in the other image, the search becomes much more efficient. Furthermore the necessity of computation of the epipolar line is removed.

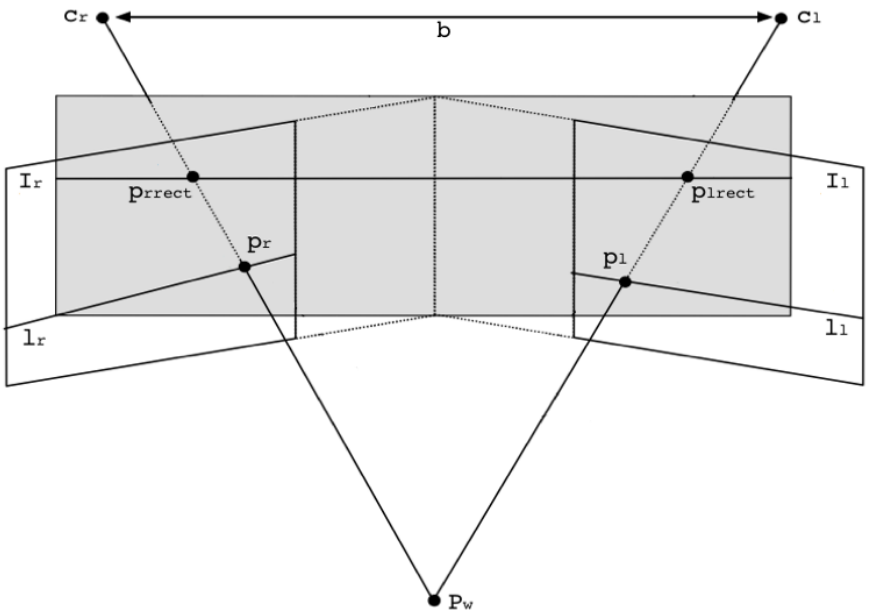


Figure 3.7: Illustration of the basic idea of image rectification. Rectification aligns epipolar lines parallel to the baseline as a result of a suitable rotation of both image planes. [Adapted from [Sch05]]

To rectify the images of a stereo vision system means to transform the images, so that coplanarity and row-alignment of the image planes are ensured. Coplanarity implies the alignment of the image planes parallel to the baseline of the stereo vision system, with the optical axes being perpendicular to the baseline. Parallel alignment of the image planes to the baseline results in parallel, row-aligned epipolar lines due to the epipoles being mapped to infinity.

Different approaches to image rectification exist, categorized by the need for knowledge of the camera parameters. In [Har99] Hartley describes an approach based on an uncalibrated system. As this thesis uses a stereo vision system for which the calibration parameters have been obtained, only the calibrated case of stereo image rectification will be considered. The method used in this thesis has been proposed by Fusiello et al. in [FTV00] as an improvement of the approach published by Ayache and Lustman in [AL91] and will be described in the following.

Let $M = A[R\,|\,t]$ be the perspective projection matrix of the camera, containing the intrinsic and extrinsic parameters. The relation between a three-dimensional point P_w and its two-dimensional projection p_i in the image is given by:

$$p_i = MP_w \tag{3.59}$$

By decomposing $M = [Q\,|\,q]$, with Q being a 3×3 matrix and q a 3×1 vector, the coordinates of the center of projection c can be obtained by:

$$c = -Q^{-1}q \tag{3.60}$$

Due to equation (3.60), the perspective projection matrix M can be written as:

$$M = [Q\,|\,-Qc] \tag{3.61}$$

Through calibration the perspective projection matrices M_l and M_r for the left and right camera have been determined. The rectification of the images can be obtained through application of a suitable rotation to the projection matrices, so that the image planes become coplanar, with the X-axis of both images becoming parallel to the baseline of the stereo vision system. Furthermore the rectification must produce a row-aligned pair of images. To ensure row-alignment it is required for the resulting new perspective projection matrices M_l' and M_r' to share the same set of intrinsic parameters. This can be done by taking the average of the corresponding entries from both sets of intrinsic parameters A_l and A_r. A proper rectification therefore approximates the standard stereo geometry configuration.

The positions of the cameras represented by the new projection matrices remain the same, while the orientations differ by suitable rotations. The new perspective projection matrices are given by:

$$M_l' = A_n'[R\,|\,-Rc_l] \tag{3.62}$$

$$M_r' = A_n'[R\,|\,-Rc_r] \tag{3.63}$$

For reasons of convenience the rotation matrix can be specified through its row-vectors representing the axes of the three-dimensional coordinate system:

$$R = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix} \tag{3.64}$$

With the constraint of the new X-axis of the image being parallel to the baseline of the stereo vision system, the first row-vector of R is given by:

$$r_1 = (c_l - c_r)/||c_l - c_r|| \tag{3.65}$$

Without the existence of any further constraints, the orientation of the new Y-axis is chosen to be orthogonal to the plane spanned by the new X-axis and an arbitrary unit vector k. The remaining Z-axis is orthogonal to the XY-plane:

$$r_2 = r_1 \times k \tag{3.66}$$

$$r_3 = r_1 \times r_2 \tag{3.67}$$

The rectification transformations for both cameras are established as the mapping between the old perspective projection matrices and the constructed new projection matrices. This collinear mapping is given by:

$$T_l = R_{nl} R_{ol}^{-1} \tag{3.68}$$

$$T_r = R_{nr} R_{or}^{-1} \tag{3.69}$$

After the performed rectification the resulting images can be used to perform an efficient search for corresponding points. Figure 3.8 shows an example of a pair of rectified images. A wealth of methods exist to approach the problem of an automated search for corresponding points in stereo images. Two well known examples are the methods by Konolige [Kon97] and Birchfield and Tomasi [BT98]. Such methods are outside of the scope of this thesis. Search for corresponding points will be conducted over a highly reduced set of points, which represent the marker object projections obtained from the images through feature extraction described in section 2.5.

Three-dimensional reconstruction can be performed, once the set of pairs of corresponding points has been determined. The process of reconstruction of three-dimensional point coordinates by triangulation will be discussed in the following section.

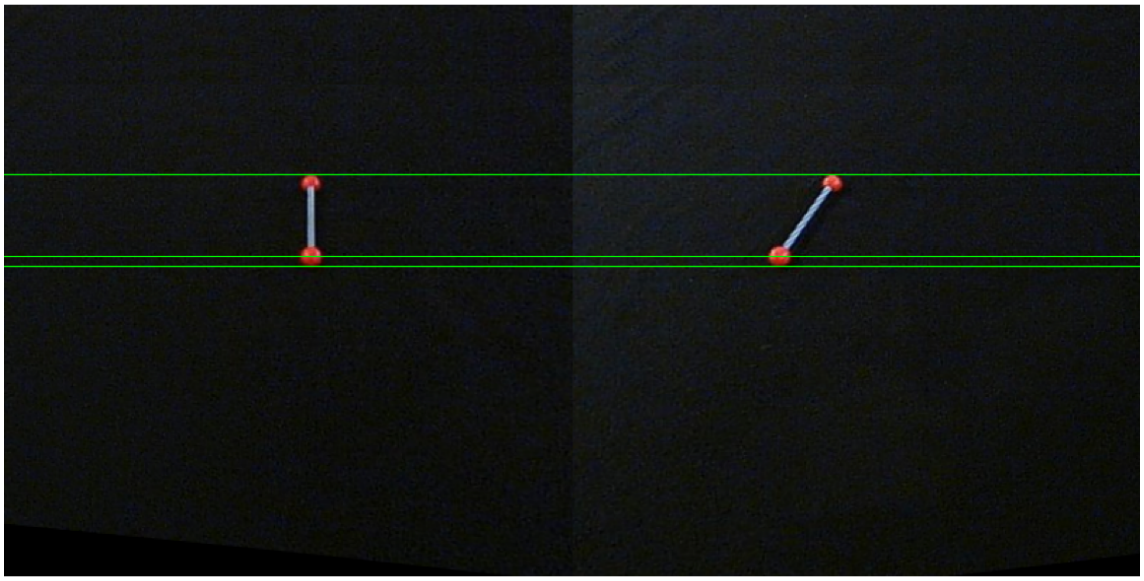


Figure 3.8: A pair of recorded images after rectification. The green lines represent the epipolar lines parallel to the X-axis and baseline as a result of mapping the epipoles to infinity. Corresponding points, e.g. the centroids of both red marker objects shown in both images, are located in the same row of both images.

3.5 Triangulation

Triangulation is one of the fundamental problems in computer vision. Given the idealized standard stereo geometry system, previously presented in section 3.3.1, the reconstruction of a three-dimensional point P_w based on both its projections p_l and p_r in the left and right image is trivial. But the constraints imposed by the definition of the standard stereo geometry can never be met by a real stereo vision system. Due to the inherent inaccuracies of the calibration process and manufacturing tolerances, no two used cameras will have the exact same internal geometry represented by the intrinsic parameters. In addition to that, due to the presence of noise a pair of corresponding points might not satisfy the epipolar constraint $p_r^T F p_l = 0$. As a result the rays from the observed projections p_l and p_r through the corresponding centers of projection c_l and c_r, will not intersect in three-dimensional space. Figure 3.9 illustrates the triangulation problem. Therefore the task of triangulation becomes one of finding the three-dimensional point that best describes the observed projections.

Various approaches to the triangulation problem exist. Alongside their own method, Hartley and Sturm [HS97], describe and evaluate other common methods. Kanatani, Sugaya and Niitsuma propose a faster method in [KSN08], that is of equal quality in comparison to the method by Hartley and Sturm. Lindstrom proposed another

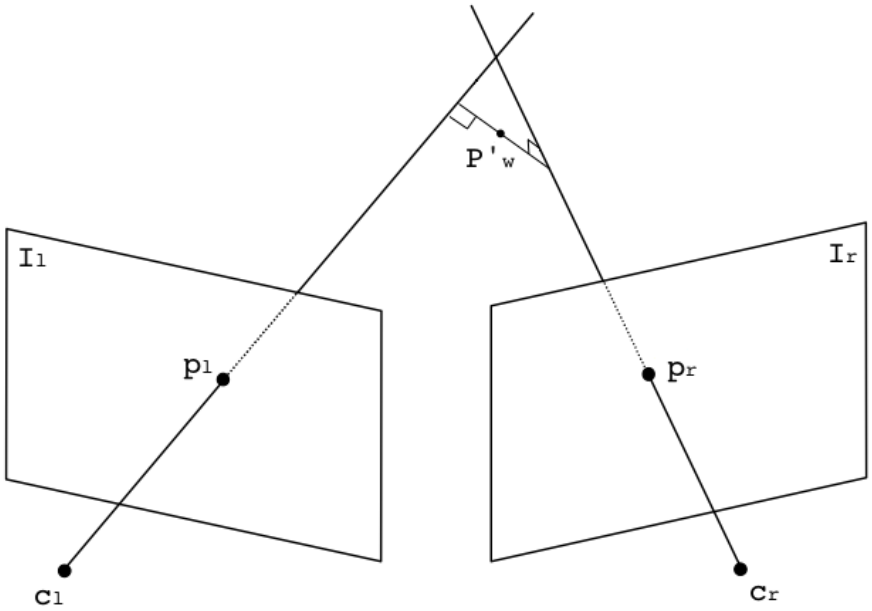


Figure 3.9: Triangulation in a stereo vision system. Due to various inaccuracies and noise the rays from the observed projections through the corresponding centers of projection might not intersect in three-dimensional space. The illustrated mid-point approach considers the center of the shortest distance between the two rays as the best approximation. [Adapted from [Sch05]]

method in [Lin10], that achieves equal results compared to mentioned methods, while providing higher numerical stability and operating up to 4x faster.

An implementation of the point correction algorithm proposed by Hartley and Sturm in [HS97] (as part of the optimal triangulation method), provided by the OpenCV framework [Wil10], was used in the implementation of the stereo reconstruction workflow within the software application as part of this thesis.

After giving a short overview over common triangulation methods (as evaluated in [HS97]), Hartley and Sturm's own optimal triangulation method will be described in the following.

One common approach to solve the triangulation problem is the linear *mid-point method*. The mid-point method suggests the usage of the mid-point located on the common perpendicular to both rays (see Figure 3.9). Let $M = [Q \mid q]$ be a decomposition of the perspective projection matrix of a camera and $c = -Q^{-1}q$ the center of projection of that camera. Any three-dimensional point P_w that results in the projection p_i is located along the ray given by:

$$r = c + sM^{-1}p_i \tag{3.70}$$

s represents a scale factor. Given a left and a right image, the rays r_l and r_r must intersect in space:

$$s_l M_l^{-1} p_l - s_r M_r^{-1} p_r = c_r - c_l \tag{3.71}$$

The values of s_l and s_r can be solved by linear least squares methods. This minimizes the squared distance between the two rays in three-dimensional space and the mid-point is therefore given by:

$$P'_w = (c_l + s_l M_l^{-1} p_l + c_r + s_r M_r^{-1} p_r)/2 \tag{3.72}$$

The mid-point method is not invariant to perspective projection and according to the evaluation in [HS97], it is not recommendable, since it delivers the highest amount of error.

Another common approach is the *linear triangulation method*. The linear triangulation method resembles the *direct linear transformation method (DLT)* proposed by Abdel-Aziz and Karara in [AAK71]. With M being the perspective projection matrix of a camera, the projection $p_i = s(x, y, 1)^T$ of P_w is given by:

$$s p_i = M P_w \tag{3.73}$$

With m_i being the i-th row vector of the projection matrix the equation can be split into:

$$sx = m_1^T P_w \tag{3.74}$$

$$sy = m_2^T P_w \tag{3.75}$$

$$s = m_3^T P_w \tag{3.76}$$

Elimination of the scale factor s leads to:

$$x m_3^T P_w = m_1^T P_w \tag{3.77}$$

$$y m_3^T P_w = m_2^T P_w \tag{3.78}$$

With two equations corresponding to the left and right image, an equation of the form $AP_w = 0$ can be established with:

$$A = \begin{bmatrix} x_l m_{3l}^T - m_{1l}^T \\ y_l m_{3l}^T - m_{2l}^T \\ x_r m_{3r}^T - m_{1r}^T \\ y_r m_{3r}^T - m_{2r}^T \end{bmatrix} \tag{3.79}$$

Due to the coordinate values of the projections in the left and right image being affected by noise, equation $AP_w = 0$ will not be satisfied. Two methods have been described in [HS97], to deal with this problem. A more detailed explanation of both methods can be found in [HZ04] (ch.12).

The *linear eigen method*, also known as the homogeneous method, is one approach to find a best solution for P_w. Under the normalization constraint $||P_w|| = 1$, the solution can be found as the unit eigenvector corresponding to the smallest eigenvector of the matrix $A^T A$.

The second (non-homogeneous) approach is the *linear least-squares method*, which finds the least-squares solution for P_w by singular value decomposition after establishing the constraint $P_w = (X_w, Y_w, Z_w, 1)^T$. The constraint reduces the set of the four equations to non-homogeneous equations. Neither the homogeneous nor the non-homogeneous method are invariant to perspective projection.

In [HS97] Hartley and Sturm proposed the *optimal triangulation method*. It is a non-iterative method that provides a provably optimal solution in the case of the projections being affected by Gaussian noise. It acts by finding the absolute minimum of an established cost-function, which represents a parametrized sum of squared distances. In contrast to the previously described methods the optimal triangulation method works in two-dimensional space and is invariant to perspective projection.

With p_l and p_r being the observed projections of P_w that do not satisfy the epipolar constraint, two corrected projections p_l' and p_r' need to be found, which satisfy the epipolar constraint, while minimizing the sum of squared distances to the observed projections:

$$d(p_l, p_l')^2 + d(p_r, p_r')^2 \tag{3.80}$$

By definition of the epipolar constraint, the two corrected projections p_l' and p_r' will be lying on two corresponding epipolar lines l_l and l_r. Thus the equation (3.80) can be rewritten as:

$$d(p_l, l_l)^2 + d(p_r, l_r)^2 \tag{3.81}$$

The minimization criterion itself (sum of squared distances) ensures the equality of both expressions. Parametrizing the epipolar pencil, the set of epipolar planes

rotated around the baseline implicitly describing the set of corresponding epipolar lines in both images, by a parameter t, allows to reformulate the minimization criterion as a function of a single variable. Equation (3.81) can be rewritten as:

$$s(t) = d(p_l, l_l(t))^2 + d(p_r, l_r(t))^2 \tag{3.82}$$

To simplify further analysis the observed projections are translated to the origin of the corresponding image by following rigid transformations:

$$T_l = \begin{bmatrix} 1 & 0 & -x_l \\ 0 & 1 & -y_l \\ 0 & 0 & 1 \end{bmatrix}, \qquad T_r = \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} \tag{3.83}$$

After the application of T_l and T_r, the epipolar geometry is represented by the new fundamental matrix $F' = T_r^{-T} F T_l^{-1}$. The left and right epipoles $e_l = (e_{1l}, e_{2l}, e_{3l})^T$ and $e_r = (e_{1r}, e_{2r}, e_{3r})^T$, can be computed so that $F'e_l = 0$ and $e_r^T F' = 0$.

For further simplification, the epipoles are normalized, so that $e_{1l}^2 + e_{2l}^2 = 1$ and $e_{1r}^2 + e_{2r}^2 = 1$. Furthermore the epipoles are rotated onto the X-axis of the corresponding image at positions $(1, 0, e_{3l})^T$ and $(1, 0, e_{3r})^T$ by application of following transformations:

$$R_l = \begin{bmatrix} e_{1l} & e_{2l} & 0 \\ -e_{2l} & e_{1l} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R_r = \begin{bmatrix} e_{1r} & e_{2r} & 0 \\ -e_{2r} & e_{1r} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.84}$$

The minimization criterion remains unaffected by these transformations. Again $F'' = R_r F R_l^T$ represents the altered epipolar geometry. Assuming $f_l = e_{3l}$, $f_r = e_{3r}$ and $a = F''_{22}$, $b = F''_{23}$, $c = F''_{32}$, $d = F''_{33}$, the cost-function can be established by following polynomial:

$$s(t) = \frac{t^2}{1 + f_l^2 t^2} + \frac{(ct + d)^2}{(at + b)^2 + f_r^2 (ct + d)^2} \tag{3.85}$$

Equating the derivative of $s(t)$ to zero results in polynomial of degree 6 given by:

$$\begin{aligned} g(t) = {} & t((at + b)^2 + f_r^2 (ct + d)^2)^2 \\ & - (ad - bc)(1 + f_l^2 t^2)^2 (at + b)(ct + d) \\ = {} & 0 \end{aligned} \tag{3.86}$$

According to [HS97] the extrema of $s(t)$ will occur at the roots of $g(t)$, of which up to 6 may exist given the degree of $g(t)$. Evaluation of $s(t)$ at each of those roots will

yield the absolute minimum t_{min} of the cost-function. Evaluation of the epipolar lines l_l and l_r according to t_{min}, followed by computation of the closest points to the image origin, provides the transformed corrected projections. After removal of the previously applied transformations by $T_l^{-1}R_l^T p'_l$ and $T_r^{-1}R_r^T p'_r$, the remaining three-dimensional reconstruction of P_w can be performed by a linear triangulation method.

3.6 Application to the experimental setup

Based on the results obtained from pairwise stereo calibration (see section 6.2.1) the perspective camera projection matrices of both logical stereo pairs, previously described in section 3.3.3, were used to obtain rectification transformations for both stereo pairs. This was implemented using methods provided by the OpenCV framework, based on the approach presented in section 3.4.

The sequence of methods described in chapter 2 was applied to the rectified images, in order to extract two-dimensional point sets representing the marker object projections with sub-pixel accuracy due to the use of moments as shape descriptors (see section 2.5.2).

The fundamental matrices of both rectified stereo pairs were derived from the perspective projection matrices of the rectified cameras by decomposition. The fundamental matrices describing the epipolar geometry of the rectified stereo pairs, were used to determine corresponding points between the obtained two-dimensional point sets, based on the epipolar constraint given by the equation (3.44).

Determined corresponding projections were optimized using the point correction algorithm of the previously presented optimal triangulation method [HS97]. Following the optimization, three-dimensional points were computed by reprojection of the two-dimensional points belonging to the image of the reference camera. Using the disparity, three-dimensional coordinates were calculated according to the set of equations presented in section 3.3.1.

In order to allow matching of the reconstructed points between both stereo pairs within the coordinate frame of the middle camera, the inverse rectification transformation was applied to the reconstructed points, transforming them into the same coordinate frame. Based on the evaluation of multiple sets of reconstructed points, distance thresholds were determined to allow merging of the reconstructed point sets.

The evaluation of the stereo vision setup presented in section 6.2, shows the amount of maximum coordinate difference obtained along the three axes for the same marker object reconstructed by both stereo pairs. Based on the results of the evaluation the distance thresholds were chosen as 6.0mm, 3.0mm and 7.0mm, for the X-, Y- and Z-axes respectively. For every pair of reconstructed points covered by the chosen

thresholds, a merged representation was computed by averaging over the X-, Y- and Z-coordinate values.

The merged set of three-dimensional points was used as the starting point for the reconstruction of the hand pose that will be presented in the following chapter.

3.6.1 Changing coordinate frames

The three-dimensional coordinates of a reconstructed point P_w obtained, describe its position relative to the coordinate frame of the reference camera of each stereo pair.

To obtain the three-dimensional coordinates of P_w relative to a world coordinate system, a transformation must be applied to P_w, consisting of a rotation and translation. The transformation is described by the set of extrinsic parameters, that relate the coordinate frame of the reference camera to a global coordinate frame.

The set of extrinsic parameters of the chosen reference camera can be determined in reference to a calibration object, as described in section 3.2.4, by estimation of a homography between the plane of the calibration object and the image plane. Therefore a suitable world coordinate frame can be chosen by positioning the calibration object accordingly.

3.7 Conclusion

In this chapter the workflow for three-dimensional reconstruction based on a stereo vision system has been presented. Starting with the fundamental process of camera calibration, the geometry of a general stereo vision system, like the one used in the experimental setup of this thesis, has been described as epipolar geometry. The fundamental matrix and the essential matrix have been presented as representations of the epipolar geometry of the uncalibrated and calibrated case. The advantages of image rectification have been discussed, together with the image rectification method, which was used in the implementation of a software application as part of this thesis. Several triangulation methods have been presented, in order to obtain a reconstruction of a point in three-dimensional space based on its corresponding projections. In conclusion the application of the stereo vision workflow in reference to the experimental setup has been described.

The methods for three-dimensional reconstruction together with the image processing workflow described in chapter 2, provide the core functionality within the visualization module of the developed software application. An evaluation of the overall performance of both implemented workflows in combination with the experimental setup is presented in sections 6.1 and 6.2. The architecture of the software application is presented in chapter 5.

Results provided by the three-dimensional reconstruction of points that represent marker objects attached to the glove, form the dataset required for the step of reconstruction of the associated articulated hand pose. The next chapter will discuss the steps and associated methods, that were used in order to obtain a model of the hand described as a kinematic chain using Denavit-Hartenberg parameter sets.

Hand pose reconstruction

4

The hand is arguably the single most dexterous tool available to the human being for interaction with the environment. The motion capability of the articulated structure, albeit constrained to a high degree, is unique and enables various grasping possibilities as well as fine-grained manipulation of objects.

Due to its versatility, various aspects of the human hand have been covered by a wealth of research. Various designs of prosthetic and robotic hands have been created, striving to create an anthropomorphic artificial hand with a reasonably comparable motion capability. Much of the research has been dedicated to creation of an understanding of grasping and manipulation abilities, in order to enable learning of the abilities and their replication with an anthropomorphic artificial hand. Because of this, a lot of research focused on finding a means of observing the human grasping and manipulation capability, in order to provide a basis for interpretation. The HANDLE project [HAN11], under which this thesis was written, aims to learn from the human being in order to advance the current state of manipulation capability of robotic platforms.

This thesis approaches the task of reconstruction and description of the hand pose based on visual observation of human hand motion, in order to perform a description of hand motion sequences suitable for further analysis.

This chapter will present the process of hand pose reconstruction, which was implemented within the software application as part of this thesis. Starting with an overview of the anatomy of the human hand, anthropometric attributes and kinematic constraints of the hand will be presented, leading to a biomechanical model. The kinematic structure of the articulated hand model that is used for reconstruction purposes, will be described and the constraints of the model outlined. The Denavit-Hartenberg convention, used for the description of the reconstructed pose, will be discussed. Concluding this chapter, the hand pose reconstruction based on the data obtained from the three-dimensional reconstruction stage will be presented.

4.1 Anatomy of the human hand

The human hand has been subjected to a lot of research in the field of biomechanics, therefore the anatomy and the kinematic structure of the hand is relatively well understood. In [ACCL79] An et al. established a general three-dimensional model of the hand based on averaged data of multiple subjects, while Buchholz and Armstrong [BA92] developed a kinematic model of the hand skeleton. Figure 4.1 illustrates the bone structure if the human hand.

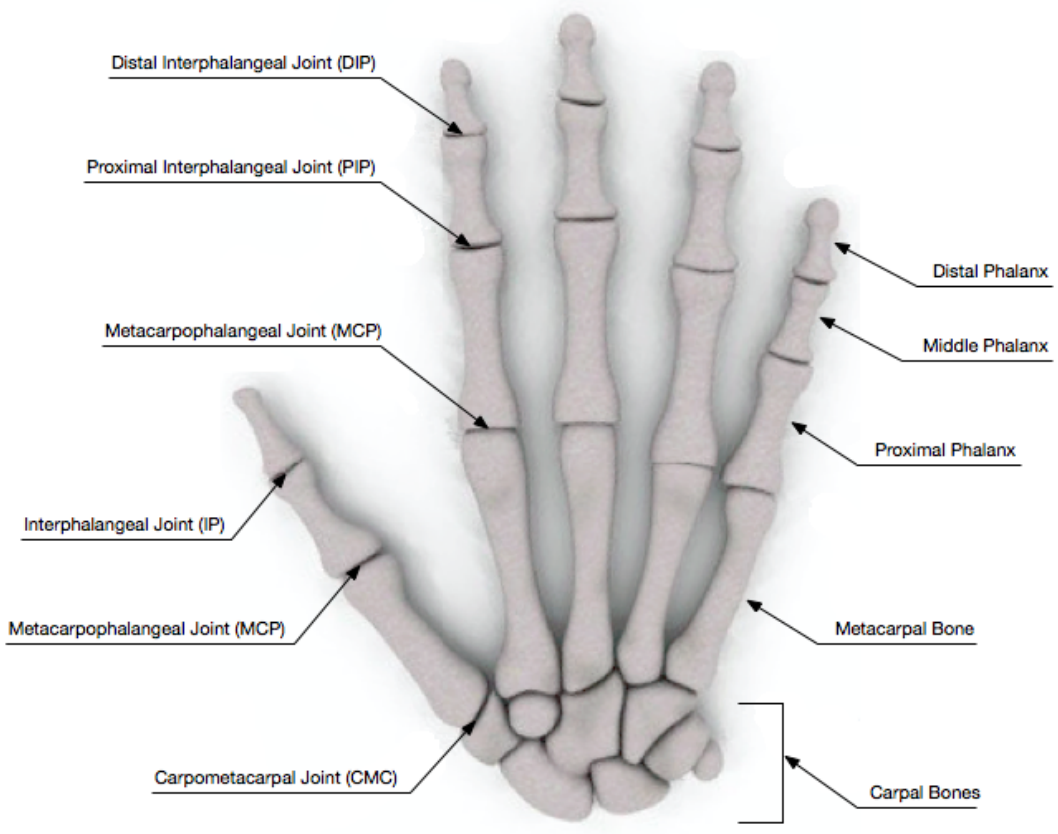


Figure 4.1: The bone structure of the human hand (dorsal view of the right hand). A total of 27 bones comprises 8 carpal bones, 5 metacarpal bones, 5 proximal phalanges, 4 middle phalanges and 5 distal phalanges [Adapted from [TP11]]

The skeletal structure of the human hand is composed of 27 bones, which are subdivided into three main groups:

- **Phalanges (fingers)**: 14 bones, with 3 bones each assigned to the index, middle, ring and little fingers and 2 bones assigned to the thumb.
- **Metacarpal bones (palm)**: 5 bones that connect the the wrist to each of the five fingers.
- **Carpal bones (wrist)**: 8 bones located between the metacarpal bones and the radius/ulna bones of the forearm.

The bones of the index, middle, ring and little fingers are the *proximal*, *middle* and *distal phalanges*, with the proximal phalanges connecting the fingers to the corresponding metacarpal bones. The thumb deviates from the other fingers, in that it does not have a middle phalanx.

In order to describe the kinematics of the bone structure, the intricacies of the carpal bone structure can be disregarded [BA92, CDML+07] and all eight bones can be considered as a single wrist joint. Therefore the complete articulation of the hand is represented by the motion of 16 joints. The joints, as shown in figure 4.1, are subdivided according to type. Table 4.1 shows the distribution of joint types within the kinematic structure of the hand.

		Fingers	
Joint	Wrist	Thumb	Index/Middle/Ring/Little
RU (Radioulnar)	×		
CMC (Carpometacarpal)		×	
MCP (Metacarpophalangeal)		×	×
IP (Interphalangeal)		×	
PIP (Proximal IP)			×
DIP (Distal IP)			×

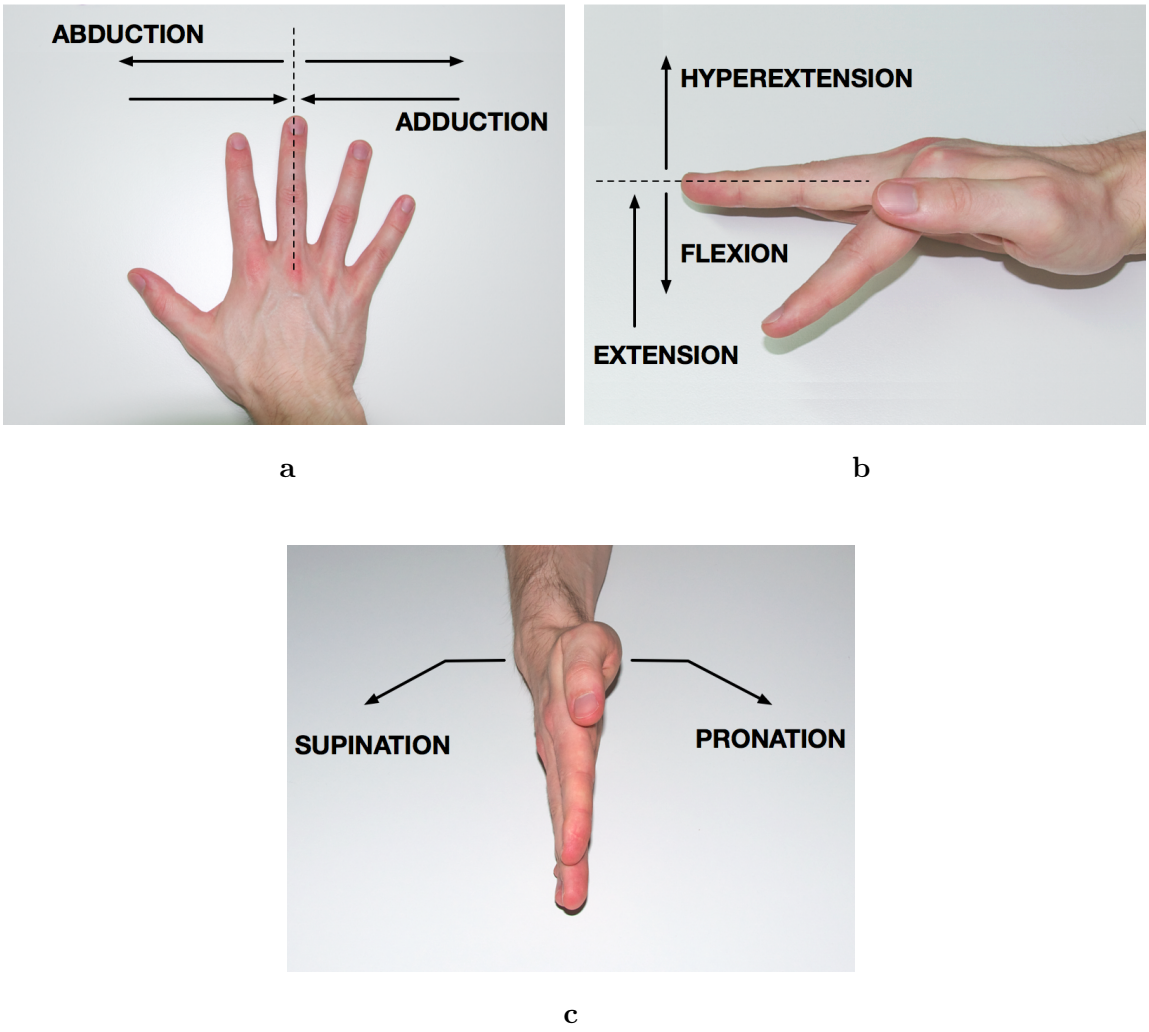
Table 4.1: Distribution of joint types within the kinematic structure of the hand.

The types of motion a joint performs in accordance with its amount of degrees of freedom (DOF), are described as follows:

- **Abduction/Adduction (AA)**: This type of motion describes the yaw rotation. Most of this type of rotation happens in the *coronal plane*, which divides the back of the hand from the front of the hand. In reference to the wrist joint this type of motion is called the *radioulnar deviation*.

- **Flexion/Extension (FE)**: This type of motion describes the pitch rotation. It usually happens in the *sagittal plane*, which is orthogonal to the coronal plane and divides the left part of the hand from the right part, along the line through the wrist joint and the MCP joint of the middle finger.
- **Pronation/Supination (PS)**: This type describes the roll rotation. For the most part this type of motion is performed in the *transverse plane*, which divides the hand from the forearm and is orthogonal to both the other planes.

The following figure 4.2 illustrates all three types of motion performed by the kinematic structure of the hand.



a **b**

c

Figure 4.2: Illustration of three types of motion performed by the hand. Figure **a)**: Abduction/Adduction, **b)**: Flexion/Extension and **c)**: Pronation/Supination.

It should be noted that the described types of motion in association with the anatomical planes are only considered valid for the index, middle, ring and little fingers as well as the wrist. No accepted standard terminology for the motion of the thumb exists. Cooney et al. [CLCL81] proposed the abduction/adduction within the sagittal plane and the flexion/extension within the coronal plane. This proposal was adopted within this thesis.

The kinematic structure of the human hand has a total of 27 degrees of freedom. Based on the amount of DOF assigned to each joint, each finger has 4 degrees of freedom, with the exception of the thumb, which has 5. The wrist has 3 degrees of freedom with reference to orientation and an additional 3 with reference to its global position. Table 4.2 describes the amount of degrees of freedom for every type of joint along with the associated type of motion.

		Type of motion		
Joint	DOF	Ab-/Adduction	Flexion/Extension	Pro-/Supination
RU	3	×	×	×
CMC	2	×	×	
MCP	2	×	×	
IP	1		×	
PIP	1		×	
DIP	1		×	

Table 4.2: Degrees of freedom and the associated types of motion.

4.1.1 Anthropometric attributes

The dimensions of the skeletal structure of the hand, in relation to the lengths of the bones and the sizes of the joints, are specific to the individual subject. Buchholz et al. [BAG92] have determined a set of coefficients for estimation of bone segment lengths based on the length of the hand (measured from the wrist to the tip of the middle finger). The set of coefficients is shown in table 4.3.

It should be noted that segment 1, shown in table 4.3, does not correspond to a single actual bone segment. It describes the distance between the wrist joint and the MCP and CMC joints of the fingers. The table shows that the length of each bone segment decreases from the wrist to the tip of a finger. Moreover, Littler [Lit73] has determined that the bone lengths of a finger closely follow the Fibonacci sequence (fingertip to wrist).

Segment	Thumb	Index finger	Middle finger	Ring finger	Little finger
1	0.118 ±0.005	0.463 ±0.003	0.446 ±0.003	0.421 ±0.004	0.414 ±0.003
2	0.251 ±0.004	0.245 ±0.001	0.266 ±0.003	0.244 ±0.003	0.204 ±0.002
3	0.196 ±0.003	0.143 ±0.003	0.170 ±0.003	0.165 ±0.002	0.117 ±0.002
4	0.158 ±0.002	0.097 ±0.002	0.108 ±0.003	0.107 ±0.004	0.093 ±0.003

Table 4.3: Set of coefficients, determined by Buchholz et al. [BAG92], for the estimation of bone segment lengths, based on the length of the subject's hand. Shown are the mean coefficient values and the associated standard error.

4.1.2 Kinematic constraints

Although the kinematic structure of the human hand is highly articulated, it is constrained to a high degree. Constraints imposed on the range of motion of the hand's kinematic structure define the set of natural hand configurations.

According to Lin et al. [LWH00] constraints of the hand's kinematic structure can be classified into the following three types:

- **Static constraints**: Motion constraints due to the anatomy of the hand.
- **Dynamic constraints**: Limitation of the motion due to motion coupling.
- **Natural motion constraints**: Limitations imposed by cognitive processes.

Static constraints

Static constraints describe the general limitations of the articulated kinematic structure regarding the range of motion of each individual joint. Although the range of motion of each specific joint depends on many factors, the static constraints shown in table 4.4 have been widely accepted [YGFS78, LWH00, CFSU+08] for the joints of the index, middle, ring and little fingers.

A variety of different static constraints for the thumb have been presented in literature [CLCL81, SKH+98, CM06], mainly because of the lack of a standard neutral position. The values shown in table 4.5 have been determined by Smutz et al. [SKH+98].

Table 4.6 shows static constraints for the wrist [YMFG78, VLB80, MW11]. Pronation/supination is not considered.

Joint	Abduction	Adduction	Flexion	Extension
MCP	15.0°	15.0°	90.0°	0.0°
PIP			110.0°	0.0°
DIP			90.0°	0.0°

Table 4.4: Static constraints: range of motion of the joints of the index, middle, ring and little fingers. Hyperextension is not considered.

Joint	Abduction	Adduction	Flexion	Extension
CMC	20.0°	20.0°	20.0°	25.0°
MCP	15.0°	15.0°	60.0°	0.0°
IP			60.0°	0.0°

Table 4.5: Static constraints: range of motion of the joints of the thumb. Hyperextension is not considered.

Dynamic constraints

The most widely accepted dynamic constraint [RK94, LWH00, TP11] describes an intra-finger dependency between the PIP and DIP joints of the index, middle, ring and little fingers:

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP} \tag{4.1}$$

Another dynamic constraint established by Kuch and Huang [KH94], describes a dependency between the flexion of the PIP joint and the flexion of the MCP joint:

$$\theta_{MCP} = \frac{1}{2}\theta_{PIP} \tag{4.2}$$

Flexion of the middle finger's MCP joint that follows a flexion of the index finger's MCP joint, is one further example of a dynamic constraint. An equal constraint exists between the middle finger and the ring finger.

Natural motion constraints

The type of natural motion constraints has yet to be researched. Examples of natural motion constraints, such as the way most human beings form a fist [LWH00], or close the hand with only the index finger extended to point at something, reveal that natural motion constrains have no relation to static or dynamic constraints.

Joint	Radial deviation	Ulnar deviation	Flexion	Extension
RU	30.0°	30.0°	70.0°	55.0°

Table 4.6: Static constraints: range of motion of the wrist.

Aside from the mentioned constraints, many other constraints exist that can not be expressed in a general closed form. Many proposed articulated hand models, with [RK94, CDML+07, CFSU+08] being only a few examples, have shown that the existence of such constraints does not largely affect the reconstruction of the hand pose and that accurate results are possible.

4.2 Articulated hand model

The articulation of the kinematic structure of the hand model used in this thesis is described by 26 degrees of freedom. Apart from the 3 DOF related to the global position of the wrist joint, flexion/extension and radioulnar deviation are considered. The CMC and MCP joints are modeled with two degrees of freedom in reference to the flexion/extension and abduction/adduction of the joint. All remaining joints are modeled with a single DOF, corresponding to flexion/extension of the joint. Figure 4.3 illustrates the kinematic structure of the articulated hand model used in this thesis.

The 20 colored marker objects attached to the glove used for the purpose of motion capture, allow to estimate the positions of every joint of all five fingers as well as the tip of the finger. Another 3 markers attached to a wrist fixture allow to estimate the orientation and to determine the position of the wrist joint. The design of the glove and the wrist fixture is presented in section 5.1.3.

4.2.1 Hand model constraints

Given the set of three-dimensional marker object positions from the stereo reconstruction stage (see chapter 3), it is necessary to impose a few constraints on the model, in order to be able to assign the marker objects to corresponding joints.

The three-camera stereo vision setup allows a greater range of global and local hand motion without self-occlusion, compared to a binocular setup. Nevertheless the set of occlusion-free reconstructable hand configurations, which are possible with the experimental setup used in this thesis, is relatively small.

In order to enable successful assignment of the marker objects to their corresponding joints, the following two constraints have been established:

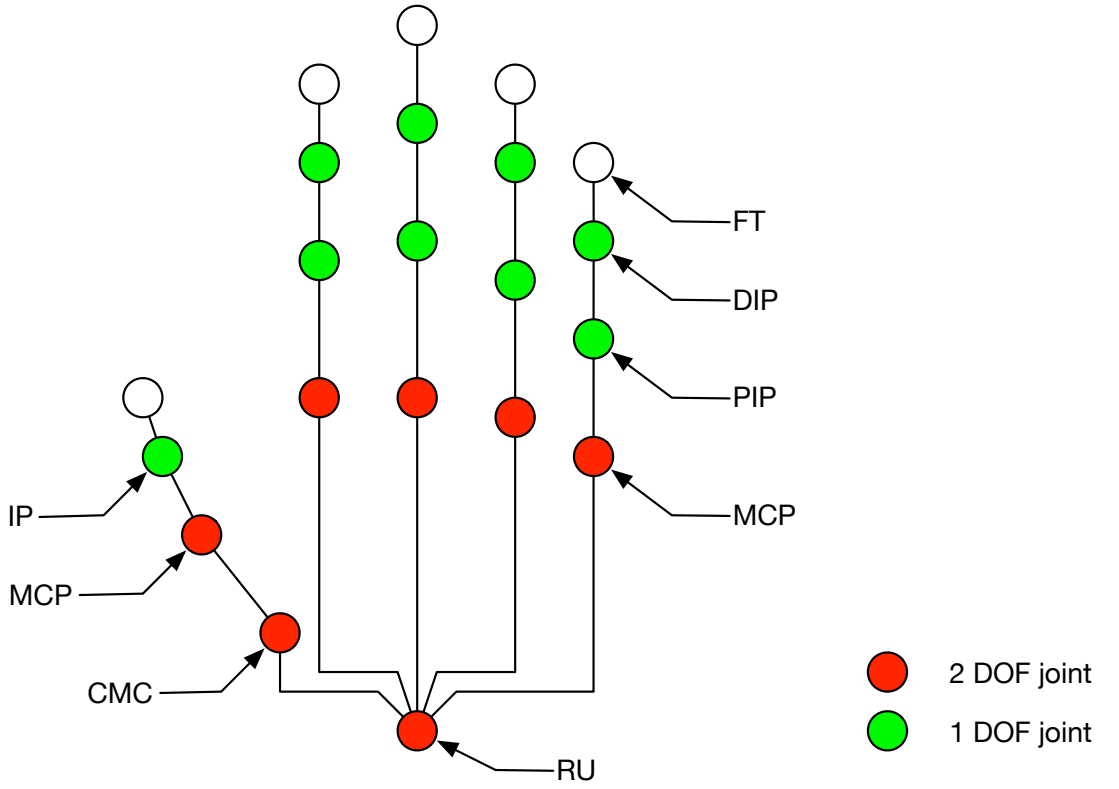


Figure 4.3: Illustration of the kinematic structure of the articulated hand model used in this thesis (dorsal view). The hand model has a total of 26 degrees of freedom. 3 degrees of freedom in reference to the global position of the hand are not shown in the illustration.

1. The subject's hand should not show any malformation related to its dimensions. A general applicability of the coefficients derived by Buchholz et al. [BAG92] is assumed.

2. The range of motion of the MCP joint of the thumb with regard to flexion/extension is assumed to be limited by $\theta_{MCP-FE} > 0.0°$.

The first constraint is aimed to allow the determination of a correct marker/joint assignment in case of an arbitrarily incomplete set of reconstructed marker objects, producing a degraded hand model as a result. The second constraint allows the approximate determination of the rotation axes of the thumb joints associated with flexion/extension. This in turn allows to approximate the centers of rotation of the thumb joints, as described in the following section.

Apart from that, the model is subject to the constraints presented in the previous section, albeit they are of no immediate use due to direct estimation of every associated degree of freedom.

4.3 Assignment of marker objects to joints

The first step on the way to a description of the hand configuration, is the assignment of the reconstructed marker objects to joints of the kinematic structure of the articulated hand model.

Since the marker objects and the image processing workflow were specifically designed to simplify the identification of the fingers and the wrist, it is possible to perform the assignment between marker objects and joints based on distance measurements only.

Since the wrist markers attached to the wrist fixture form a triangle, an initial wrist reference point is found as the point of the intersection of the base of the triangle with a perpendicular going through the third vertex.

Assignment within a complete set

Once the wrist reference point is determined, the set of reconstructed marker objects in respect to a single finger is processed according to the following algorithm:

1. Determine the marker object closest to the wrist reference point and assign it to the CMC joint (thumb), or the MCP joint (other).
2. Set the determined marker object as the new reference point and remove it from the set.
3. For the three remaining marker objects of the current set:
   a) Determine two marker objects with the largest distance between them and assign them to the variables m_1 and m_2.
   b) Assign the third marker object to the variable m_3.
   c) If the distance between m_3 and m_1 is larger than the distance between m_3 and m_2: Assign the marker object sequence $[m_1, m_3, m_2]$ to the remaining sequence of unassigned joints within the kinematic chain of the current finger.
   d) Otherwise assign the sequence: $[m_2, m_3, m_1]$.
4. Continue with step 1, until all marker object sets have been processed.

The described algorithm is very simple and fast, but it does not cover the case of an incomplete reconstructed set of marker objects. Therefore an even simpler version is used as a fallback, in order to be able to reconstruct degraded kinematic chains. An example of an incomplete/degraded reconstructed hand pose can be found in section A.2.

In order to correctly determine the assignment in case of a reconstructed set of marker objects that is incomplete, bone segment lengths are used. Based on the

lengths it is determined, whether the next closest marker object from the reconstructed set can be assigned to the next joint in the kinematic chain, or whether the kinematic chain can be assumed as broken.

In order to provide initial bone segment lengths, measurements of the subject's hand were taken. The following approximate bone segment lengths were determined:

	Segment			
Finger	1 (mm)	2 (mm)	3 (mm)	4 (mm)
Thumb	45.0	49.5	40.5	31.5
Index	95.0	49.0	27.0	22.0
Middle	90.5	53.0	34.0	23.5
Ring	86.0	50.0	31.5	23.0
Little	82.0	40.0	23.0	22.0

Table 4.7: Approximate bone segment lengths obtained from the subject's hand.

In reference to table 4.7 it should be noted that the first segment does not represent an actual single bone, but rather the length from the approximate position of the wrist joint to the first joint of each of the fingers. An approximate segment length from the wrist joint to the CMC joint of the thumb could not be measured directly due to the curvature of the wrist. Therefore the best possible measurement of 45.0mm was used as an initial estimate, which turned out to be reasonable according to the results presented in section 6.3.

Assignment within an incomplete set

The reconstruction of a degraded hand configuration is performed according to the following algorithm:

1. Determine the marker object closest to the current reference point and compare the distance with the previously determined segment length between the current joint and the next joint in the chain.
2. If the difference lies within 5.0mm, assign the marker object to the next joint in the kinematic chain of the current finger. Terminate otherwise.
3. Set the determined marker object as the new reference point and remove it from the set.
4. Continue with step 1, until all marker objects have been processed.

In order to consider shifting of the marker objects during articulated motion and therefore provide a higher possibility for a successful marker object assignment, the

bone segment lengths are updated with the measured lengths in the situation where a complete reconstruction of the hand configuration was possible.

4.4 Determination of joint centers

The marker objects attached to the glove allow the determination of a three-dimensional, displaced joint position. The measured position deviates from the actual joint center by at least the sum of the joint radius and the radius of the marker object sphere. Therefore it is necessary to approximate the center of rotation of every given joint, before attempting the determination of a description of the kinematic structure of the articulated hand model.

In order to perform the approximation, the joint diameters of the subject's hand were measured approximately. Table 4.8 shows the measured values for the respective joints.

		Fingers				
Joint	Wrist	Thumb	Index	Middle	Ring	Little
RU	42.0					
CMC		25.0				
MCP		23.5	25.0	27.5	25.0	22.0
IP		16.0				
PIP			16.0	17.5	16.5	15.0
DIP			12.0	13.0	12.0	10.5
FT		9.5	9.0	9.5	9.0	8.0

Table 4.8: Approximate joint diameters obtained from the subject's hand. The unit of measure is millimeters. The tip of a finger was included, in order to provide a complete overview and is not to be considered a joint.

To approximate the actual position of the wrist joint (RU), the base coordinate frame needs to be established. In order to determine the base frame, a right-hand coordinate system is established with the help of the wrist marker objects. The Y-axis is chosen to point into the dorsal direction (away from back of the hand). The X-axis is chosen as a normal to the transverse plane spanned by the wrist marker objects, pointing towards the hand. The Z-axis, as the axis of rotation, is chosen orthogonal to the XY-plane. Figure 4.5 illustrates a similarly chosen base frame, the only difference being a rotation around the X-axis by -90.0°.

Having determined the base coordinate frame according to the described convention, the actual position of the wrist joint can be approximated by a translation along

the negative Y-axis by the sum of the radius of the wrist joint and the radius of the marker object sphere.

For the purpose of determination of the centers of rotation, each joint is considered to have only one degree of freedom in reference to flexion/extension.

Joint center approximation

Starting from the wrist joint, the determination of the approximated position of the actual joint is performed according to the following general algorithm:

1. Transform the next marker object (joint or fingertip) into the current local coordinate frame.
2. Determine the rotation angle θ, as the angle between the vector pointing to the XY-position of the marker object and the X-axis of the local coordinate frame.
3. Assign the Z-coordinate of the marker object to the variable d.
4. Assign the length of the XY-position vector to the variable a.
5. If the current marker object corresponds to the wrist joint, apply the transformation. Rotate around the Z-axis with the angle θ, followed by a translation along the Z-axis by d and the X-axis by a.
6. Else, if the current joint corresponds to the CMC joint of the thumb:
   a) Correct the coordinate frame by determination of the Z-axis orthogonal to the plane spanned by the vectors pointing towards the MCP joint and the IP joint. Establish the vector pointing towards the MCP joint as the X-axis and determine the Y-axis to form a right-hand coordinate system.
   b) Perform a translation along the negative Y-axis by the sum of the radius of the CMC joint and the radius of the marker object sphere. Store the determined position as the approximation of the actual joint center. Reverse the translation.
   c) Translate along the X-axis by the length of the vector pointing towards the MCP joint.
7. Else:
   a) Perform a rotation around the Z-axis with the angle $\theta/2$.
   b) Perform a translation along the negative Y-axis by the sum of the radius of the current joint and the radius of the marker object sphere. Store the determined position as the approximation of the actual joint center. Reverse the translation.

c) Apply the remaining transformation by rotating around the Z-axis with the angle $\theta/2$, followed by a translation along the Z-axis by d and the X-axis by a.

8. Continue at 1, until all marker objects, corresponding to a single finger, have been processed.

9. Determine the approximated center of the fingertip by performing a translation along the negative Y-axis by the sum of the radius of the fingertip and the radius of the marker object sphere. Store the determined position as the approximation of the fingertip center.

Figure 4.4 visualizes the principle of the described algorithm, by an example of the approximation of the center of rotation for all joints in the kinematic chain of the index finger, including the fingertip.

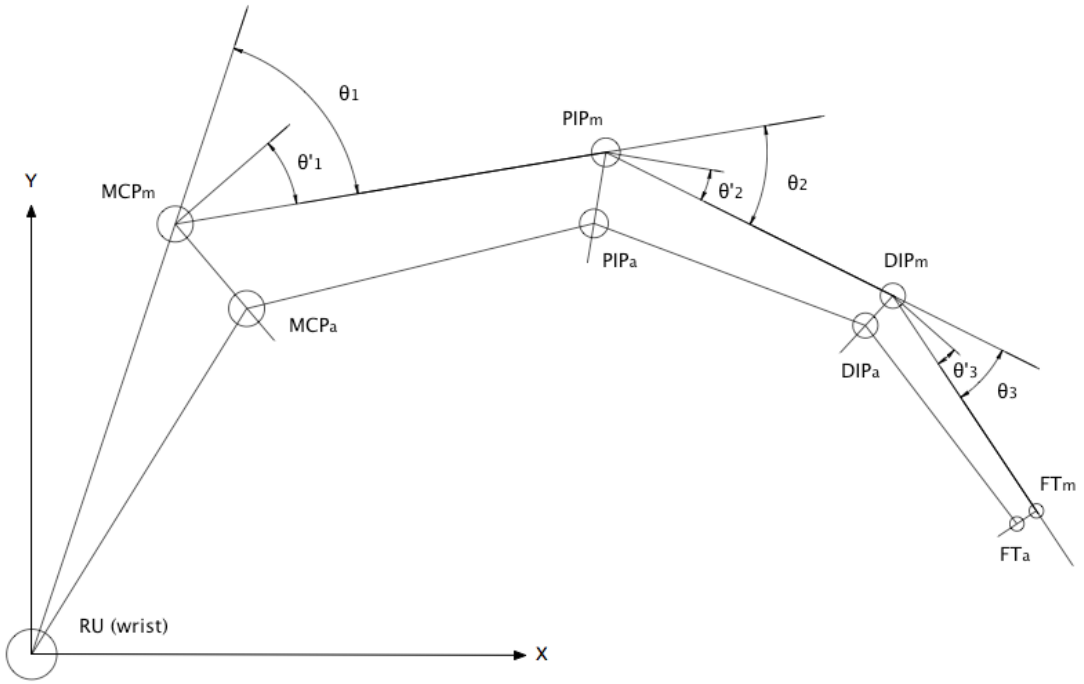


Figure 4.4: Illustration of the approximation of the center of rotation of every joint in the kinematic chain of the index finger, according to the described algorithm. θ_i describes the determined rotation angle around the rotation axis of the joint in reference to flexion/extension. Execution of only half the rotation, given by θ_i', establishes a coordinate frame that allows to determine the approximation of the actual joint position by a simple translation.

Although the described approach does disregard the fact that the wrist joint (RU) [YMFG78, AY79] and the CMC joint [HBM+92, CDML+07] do not have intersecting rotation axes and therefore no single center of rotation, the assumption of a single center of rotation is considered to provide suitable results [RG91, RK94, CDML+07].

4.5 Denavit-Hartenberg convention

The Denavit-Hartenberg (DH) convention [DH55] is a very common method for description of the forward kinematics of robotic systems. It provides a straightforward way of describing the kinematic chain of serial-link manipulators. It therefore allows to determine the position and orientation of the end-effector for a given configuration of the manipulator's joints.

The DH convention describes the kinematic chain, by describing the transformations between coordinate frames of the serial-link manipulator. In order to do this, a set of four DH parameters is used to describe the coordinate frame of joint $i+1$ in relation to the coordinate frame of the current joint i. In order for that to be possible, two constraints must be satisfied [SHV05]:

1. The X-axis of the subsequent coordinate frame must be perpendicular to the Z-axis of the current coordinate frame.
2. The X-axis of the subsequent frame must have a point of intersection with the Z-axis of the current frame.

The following are the four DH parameters that specify the complete transformation based on the established constraints:

- θ: The rotation angle around the Z-axis of the current joint's coordinate frame.
- d: The link offset, a translation along the Z-axis.
- a: The effective length of the link attached to the current joint, a translation along the X-axis.
- α: The rotation around the X-axis describing the link twist.

Each of the parameters specifies a single transformation step, given by the following homogeneous transformation matrices:

$$Rot_{z,\theta} = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$Trans_{z,d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

$$Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha & 0 \\ 0 & sin\alpha & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

Varying combinations of the transformation steps can be found in the literature. For the purpose of this thesis, the combination presented in [SHV05] was adopted. It describes the complete homogeneous transformation from coordinate frame i to coordinate frame $i+1$ as follows:

$$T = Rot_{z,\theta} * Trans_{z,d} * Trans_{x,a} * Rot_{x,\alpha} \tag{4.7}$$

$$= \begin{bmatrix} cos\theta & -sin\theta * cos\alpha & sin\theta * sin\alpha & a * cos\theta \\ sin\theta & cos\theta * cos\alpha & -cos\theta * sin\alpha & a * sin\theta \\ 0 & sin\alpha & cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In order to obtain a DH description of a serial-link manipulator, the first step is to establish the base coordinate frame and the joint axes. For the purpose of this thesis only rotary joints will be considered, as the kinematic structure of the hand model can be fully modeled using this type of joint. While the base coordinate frame can be chosen arbitrarily, the DH convention requires for the Z-axis of a joint to be established as the rotation axis of the joint.

The X-axis of each joint is required to be established as either along the common normal that intersects the rotation axes of both joints, or in case the rotation axes intersect, as the normal to the plane spanned by the rotation axes. Should the rotation axes be parallel, then the X-axis can be chosen arbitrarily.

The intersection of the either the common normal or the plane normal with the Z-axis of the coordinate frame $i+1$, determines its origin. The remaining Y-axis is chosen to complete a right-hand coordinate system.

Having established the coordinate frames of each joint, a DH description using the set of parameters θ, d, a and α, can be obtained for each coordinate frame transformation.

It should be noted that the Denavit-Hartenberg convention does not consider the actual positions of the joints, as the convention does not require for the origin of the coordinate frame to be placed within the center of the joint. This does not affect the end result, which is the position and orientation of the end-effector.

In order to obtain a DH description of a hand configuration based on the presented hand model the following constraints were established:

1. Every joint of the kinematic structure of the hand model is either a hinge joint with 1 DOF, or a combination of two hinge joints with orthogonal intersecting rotation axes in the case of a 2 DOF joint.
2. The rotation axes of two hinge joints connected by a bone segment lie on parallel planes.

Particularly due to the second constraint, no single pair of subsequent joints connected by a bone segment have skewed axes in reference to each other. This allows to consequently place the origin of the coordinate frame of each joint within its center. Apart from simplifying the visualization of the hand model, based on the obtained description, this also allows to deviate from the standard description procedure described above, in that it is not necessary to determine the rotation axes of the joints upfront.

4.6 Denavit-Hartenberg hand model description

Once the marker object positions have been adjusted to approximate the center of rotation of the corresponding joints, a description of the hand configuration according to the Denavit-Hartenberg convention, can be obtained.

It is generally adequate to consider the fingers as planar systems [VB06, LK95, LH00] and therefore assume the DH parameters d_i and a_i as fixed, for every transformation between coordinate frames within these planar systems. Nevertheless it is important to note that these fixed parameters can not be uphold throughout a recorded motion sequence and will vary. This behavior is mainly due to noise in the image processing stage. Moreover, subsequent flexion and extension of the fingers might cause a shift of marker object positions.

It is also possible to consider the metacarpal bones of the index, middle, ring and little fingers as coplanar [RK94], describing the palm as a rigid body. This assumption is limiting regarding the range of natural hand motion. Touching the tip of the thumb with the tip of the little finger or grasping of spherical objects, results in flexion of multiple metacarpal bone segments. To approach this problem, the pitch

rotation (flexion/extension) of the wrist is computed in reference to the CMC joint of the thumb and the MCP joints of the index, middle, ring and little finger. To reflect this in the model, the degree of freedom associated with the pitch rotation of the joint is modeled by a segmented hinge joint.

In order to obtain a DH description of the model, a four-stage approach is used:

1. Determine the base frame with the help of the wrist marker objects.
2. Determine the DH parameters associated with the yaw rotation (radioulnar deviation) of the wrist.
3. For the thumb:
   a) Determine the set of DH parameters in regard to the transformation from the wrist to the CMC joint.
   b) Determine the DH parameter sets associated with both DOF (AA and FE) of the CMC and the MCP joint.
   c) Conclude with the DH parameters related to the single DOF (FE) of the IP joint.
4. For the remaining fingers:
   a) Determine the set of DH parameters associated with the pitch rotation (FE) of the wrist in respect to the MCP joint of the finger.
   b) Determine the DH parameter sets associated with both DOF (AA and FE) of the MCP joint.
   c) Conclude with the parameter sets related to the single DOF (FE) of the PIP joint and the DIP joint.

4.6.1 Determination of the base coordinate frame

In order to determine the base frame, a right-hand coordinate system is established with the help of the wrist marker objects. The Z-axis, as the axis of rotation, is chosen to point into the dorsal direction (away from the back of the hand). The Y-axis is chosen in such a way that the X-axis finalizing the right hand coordinate system, as a normal to the transverse plane spanned by the wrist marker objects, will point towards the hand. Figure 4.5 illustrates the described process.

4.6.2 Determination of DH parameters

The approach to determination of DH parameters is based upon the standard Denavit-Hartenberg convention. The DH parameters are determined in reference

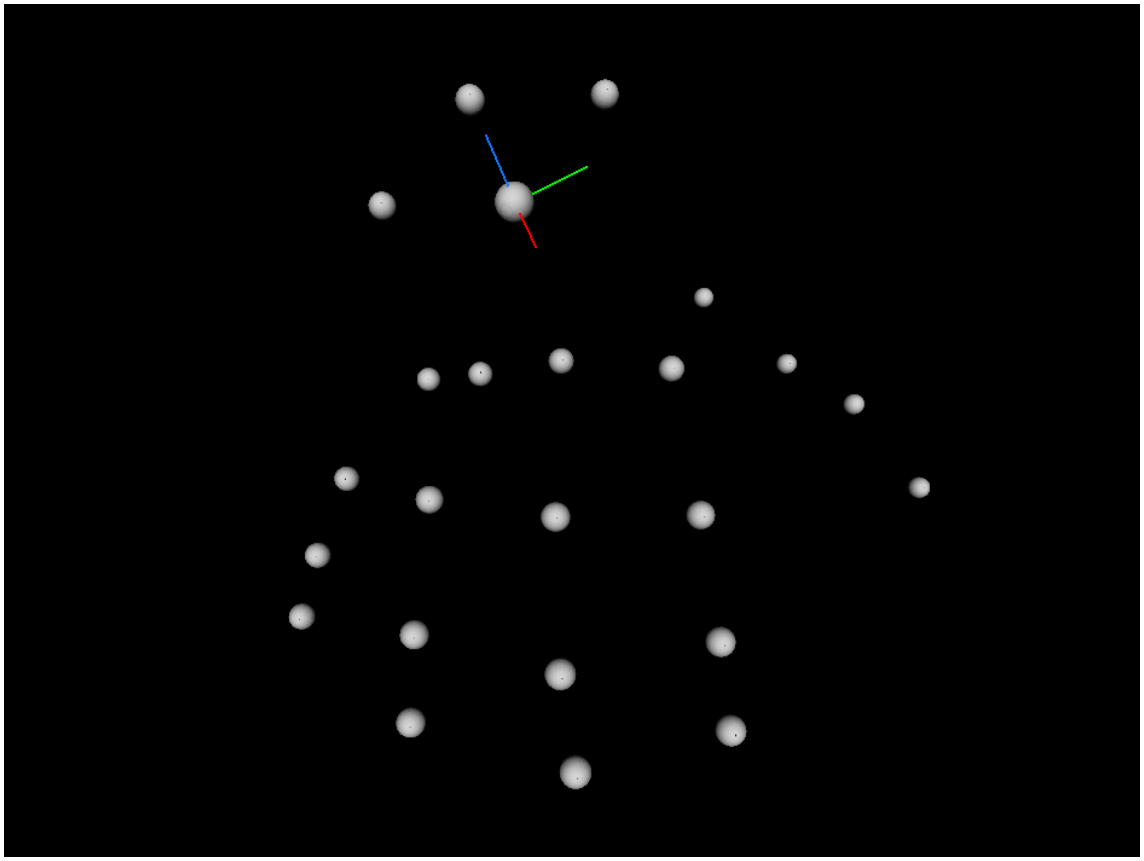


Figure 4.5: Illustration of the reconstructed marker objects. The three wrist marker objects span the transverse plane. The right-handed base coordinate frame is chosen with the X-axis (red) pointing towards the hand, the Z-axis (blue) pointing into the dorsal direction of the coronal plane and the Y-axis orthogonal to the XZ-plane. The position of the wrist joint has been adjusted to approximate the center of rotation.

to the next marker object along the kinematic chain of each finger, representing either a joint or the tip of the finger. Thus the parameters describe the transformation to the coordinate frame of the next joint in relation to that of the current joint.

The parameters associated with the first DOF of the wrist are determined using the MCP joint of the middle finger as the point of reference. Following the transformation of the reference point into the base coordinate frame, the yaw rotation represented by θ is determined using basic trigonometry. Due to the assumption of orthogonal intersecting rotation axes of both DOF of the wrist, the twist parameter is determined as $\alpha = 90.0°$. The translation parameters d and a are equal to zero.

In order to determine the DH parameters of the kinematic chain of each of the five fingers, the following general algorithm is used:

1. Transform the next marker object (joint or fingertip) into the current local coordinate frame.
2. Determine the rotation angle θ, as the angle between the vector pointing to the XY-position of the marker object and the X-axis of the local coordinate frame.
3. Assign the Z-coordinate of the marker object to d.
4. Assign the length of the XY-position vector to a.
5. Determine the twist angle α, according to following convention:
   - If the current finger is the thumb:
     - When describing the second DOF of the wrist joint, assign $\alpha = 0.0°$.
     - When describing the first DOF of a 2 DOF joint, assign $\alpha = -90.0°$.
     - When describing the second DOF of a joint in reference to another 2 DOF joint that is next in the chain, assign $\alpha = 90.0°$.
     - Otherwise assign $\alpha = 0.0°$.
   - Else:
     - When describing the first DOF of a 2 DOF joint, assign $\alpha = 90°$.
     - When describing the second DOF of a joint in reference to another 2 DOF joint that is next in the chain, assign $\alpha = -90°$.
     - Otherwise assign $\alpha = 0.0°$.
6. Apply the determined transformation and continue with step 1, until all marker objects assigned to a kinematic chain have been processed.

It should be noted that in order to determine the DH parameters for both degrees of freedom of a 2 DOF joint, the same marker object, which is next in the kinematic chain, is used twice. Moreover, due to the non-existent link between the two logical hinge joints that represent both DOF, the parameters d and a for the translation along the Z-axis and X-axis respectively, are equal to zero.

4.6.3 General DH hand configuration description

The following tables 4.9 - 4.11 describe the general Denavit-Hartenberg description of a hand configuration according to the articulated hand model used in this thesis. The data is based on the assumption that the base coordinate frame has been determined according to steps described above. Actual examples of a model description according to the DH convention can be found in section A.2.

Table 4.9 describes the general DH parameters of the wrist joint. The 2 DOF wrist joint is modeled by two logical hinge joints with intersecting axes of rotation. Since the second hinge joint is segmented, the DH parameters related to each segment have been moved to the tables that describe the parameters of the kinematic chains of each of the fingers.

Link	θ	d	a	α
1	θ_1	0.0	0.0	90.0°

Table 4.9: General DH parameters of the wrist joint.

Link	θ	d	a	α
2	θ_2	d_2	a_2	0.0°
3	θ_3	0.0	0.0	-90.0°
4	θ_4	0.0	a_4	90.0°
5	θ_5	0.0	0.0	-90.0°
6	θ_6	0.0	a_6	0.0°
7	θ_7	d_7	a_7	0.0°

Table 4.10: General DH parameters of the thumb.

Since the index, middle, ring and little finger are planar systems, the general DH description of the kinematic chains is identical and is shown in table 4.11.

Link	θ	d	a	α
2	θ_2	d_2	a_2	-90.0°
3	θ_3	0.0	0.0	90.0°
4	θ_4	0.0	a_4	0.0°
5	θ_5	d_5	a_5	0.0°
6	θ_6	d_6	a_6	0.0°

Table 4.11: General DH parameters of the index, middle, ring and little finger.

4.7 Conclusion

In this chapter the process of hand pose reconstruction was presented, based on reconstructed marker object positions provided by the stereo vision workflow. Following the general discussion of the anatomy of the human hand and the associated constraints, the chosen articulated hand model was presented. The process of assignment of marker objects to corresponding joints was described, followed by the process of determination of joint centers based on the measured positions of the assigned marker objects. The Denavit-Hartenberg convention was presented next, followed by constraints specific to the articulated hand model. In conclusion the process of determination of a Denavit-Hartenberg description of the articulated hand model was presented and general DH parameter sets related to the hand model were established.

The approach described in this chapter allows to obtain a Denavit-Hartenberg description of hand configuration sequences, based on the chosen articulated hand model. Results obtained from recorded motion sequences are presented in section 6.3. Examples of DH descriptions of the hand model are presented in section A.2.

Experimental system

5

Due to many motion capture systems being prohibitively expensive and providing proprietary solutions with regard to the hardware as well as the software, investigation of what could be achieved using readily available, inexpensive off-the-shelf hardware and open source software, posed a very intriguing task.

This chapter will present an overview over the experimental system. Following a presentation of hardware components used for the experimental setup and a discussion of the rationale behind the choice, the architecture of the developed software application will be presented.

5.1 Hardware components

In order to achieve the objectives of this thesis, an optical motion capture setup with passive marker objects was used. Constant progress pertaining to camera technology and the growing interest for use of this technology within the area of interactive gaming, led to products such as the PlayStation® Eye (PSEye) or the recent Microsoft Kinect™.

In order to achieve fine-grained segmentation of the recorded hand motion, one important criterium for the choice of the camera device was the maximum attainable image acquisition rate. The PSEye camera shown in figure 5.1, offered a maximum rate of 60 frames per second, which was twice the frequency of the Microsoft Kinect™at the exact same resolution while costing only a fifth of the price - at the time of the writing of this thesis, the PSEye camera was available at a price point of 20 Euro.

Further evaluation of the PSEye camera revealed a very usable and stable image quality with a relatively low amount of noise, even under low-light conditions. Due to the OmniVision OV7221 CMOS VGA sensor chip used within the camera, most of the noise is taken care of in-camera, reducing the need for much of additional noise reduction to be implemented within the software application.

Moreover the USB-connected camera was very well supported out-of-the-box under

Figure 5.1: The PlayStation® Eye camera used in the experimental setup. The camera offers a maximum frame rate of 60Hz at a resolution of 640 × 480 pixels. The factory fixed-focus zoom lens offers two levels of magnification with a field of view of either 56° or 75°. The upper segment of the camera contains an array of four microphones.

the Linux operating system. The Video4Linux2 application programming interface provided a simple and straightforward interface for communication with the camera.

Based on the points mentioned, the PSEye camera was chosen for the experimental setup. The following list of specifications offers a short summary over the main capabilities and advantages of the camera:

- Image resolution: 640 × 480 pixel (VGA)
- Maximum frame rate: 60Hz @ VGA / 120Hz @ QVGA
- Fixed-focus lens with a maximum field of view of 75°
- In-camera adaptive noise reduction
- Frame synchronization capability
- Support for third-party lenses (M/CS)
- Stable Linux driver available
- Low price point

5.1.1 Performance considerations

One specific problem when using USB-connected video devices, is posed by the USB bus itself. Video devices operated at high frame rates can easily saturate the bus. Although the PSEye camera acquires raw RGB data internally, it performs a conversion into the Y'CbCr color model using chroma subsampling (4:2:2) to reduce the amount of data to be transferred to the host system.

Sampling of the chroma components at half the rate of the luma component (Y'CbCr 4:2:2), allows to reduce the amount of data by a third. At 60Hz and the VGA resolution this amounts to approximately 35MByte per second (≈ 281MBit/sec.), without the overhead of the USB transmission protocol. A USB 2.0 bus offering 480MBit/sec of maximum transfer rate is therefore unable to handle more than a single PSEye camera device at 60Hz.

All experiments were conducted on a Dell OptiPlex 980 computer workstation with an Intel i5-750 quad-core (2.66Ghz) processor, 8GByte of RAM and an NVIDIA® Quadro® NVS 295 graphics card, running the Linux operating system.

The computer workstation only offered two separate USB busses. Therefore a separate USB 2.0 controller needed to be purchased to allow a third PSEye camera device to be operated.

Synchronized operation

One of the most important aspects of simultaneous image acquisition with a stereo vision setup, is synchronized operation of all camera devices. Without synchronicity results of stereo reconstruction would either be affected by a high amount of error, or no results would be produced at all.

Although the OmniVision sensor chip offers a capability for frame synchronization, it turned out not to be easily accessible within the PSEye camera devices.

A visual evaluation, based on recorded images of a timer (example shown in section 6.1) and the evaluation of image timestamps, revealed only a slight deviation between the three cameras. The deviation was 6ms at most, which is less than half of the interval between the frames, with the cameras operated at 60 frames per second. Moreover it was found that usually only a single camera deviated from the other two that worked synchronously. As the effect of the measured deviation remained unnoticeable in the results, no further measures were taken.

5.1.2 Three-camera stereo vision setup

Regarding an optical motion capture system, the range of natural hand motions which can be recorded without self-occlusion is generally dictated by the amount of camera devices as well as their position and orientation. Usage of only two cameras was found very limiting, as it allowed only a slightly curved hand pose to be recorded without occlusion. Therefore a three-camera setup (shown in figure 5.2) was used, with the cameras aligned in a convergent configuration. Section 6.2.1 presents the results of the calibration obtained for the experimental setup.

While a setup of the three cameras with their image planes approximating coplanarity would have provided a higher depth resolution, this was not considered a

Figure 5.2: The three-camera stereo vision setup. Three PlayStation® Eye cameras were mounted in a convergent configuration. The red marking signified the first stereo pair, while the blue marking signified the second stereo pair. The length of the baselines was determined as 331.0562mm and 331.7844mm, for the first and second pairs respectively.

crucial property due to the relatively small size of the chosen marker objects. The convergent configuration was chosen in order to obtain a better view of the hand, simultaneously providing better visual feedback throughout the recording of a hand motion sequence. Subsequent rectification of the obtained images allowed to obtain higher depth resolution, albeit only pairwise.

The reason for configuring the cameras with their optical axes approaching coplanarity, is purely that of convenience regarding the camera mount. Mounting the cameras in a triangle configuration might have offered an advantage with regard to the range of occlusion-free hand motion and could be investigated as part of further work.

The use of a fourth camera device was evaluated, but proved problematic. Apart from the obvious requirement of additional calibration and another USB controller, the fourth camera caused instability during simultaneous image acquisition, leading to non-deterministic drop of the frame rates.

5.1.3 Glove design

Many commercially available optical motion capture systems use near-infrared illumination in combination with retroreflective passive markers. Although the use of such a solution was evaluated, it required the PSEye cameras to be outfitted with a third-party lens in order to exchange the infrared blocking filter with an infrared

pass filter. It therefore required to deviate from the "off-the-shelf" and "inexpensive" aspects, as the lens, filter and mount were approximately four times the cost of one camera device. The issue was complicated further by the cost and availability of retroreflective markers.

Apart from the type of the marker objects, attachment also posed a non-trivial task. A simple cotton glove was used in order to attach the marker objects at the respective joint positions with metric screws. The glove offered enough stretching capability for the markers objects to remain at their approximate positions throughout flexion of the fingers. Figure 5.3 shows the glove design used in the experiments.

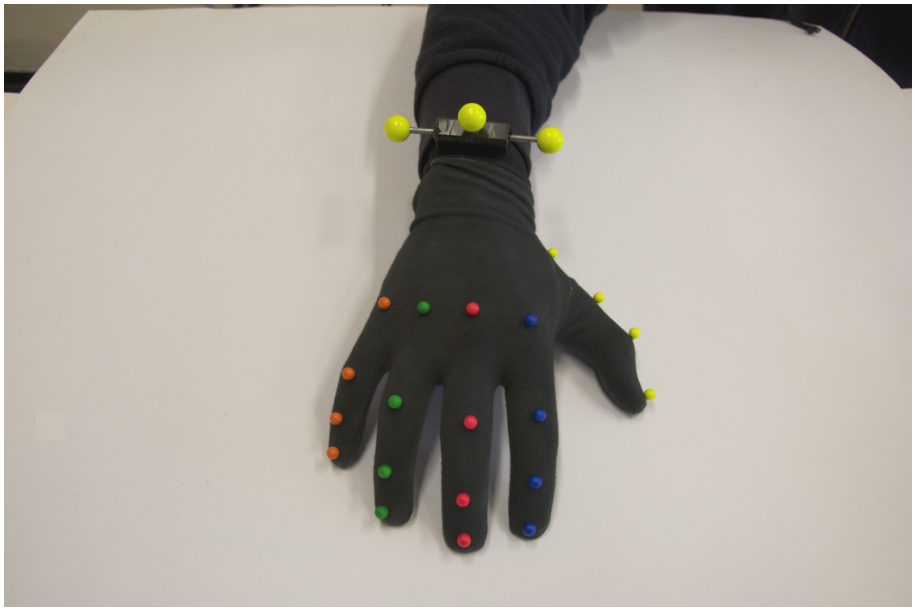


Figure 5.3: The glove design using colored, passive marker objects to approximate the joint positions. Different colors were used to simplify the identification of the joints in the kinematic chain of each finger.

Spherical marker objects were used in order to obtain rotational invariance regarding their visual representation. Moreover, the spherical form allowed to filter segmentation noise by calculation of the eccentricities of shapes obtained from the image (see section 2.5.2). The marker objects were coated with different colors in order to utilize the color information and simplify the identification of separate fingers, thereby greatly simplifying the marker/joint assignment task described in section 4.3. Five readily available colors were found to be separable within the acquired images based on their hue. Therefore the same color was used for the identification of the thumb and the wrist. Bigger marker objects were used for the wrist fixture in order to allow distinction by size. 15mm marker objects were used for the wrist fixture and 6mm marker objects for each of the fingers.

Color sample plates shown in figure 5.4 were used to calibrate initial estimates for

the thresholds of each color, by taking five snapshots of the color plates held at different orientations.

Figure 5.4: The color sample plates. The plates were used for the purpose of calibration of initial estimates for the thresholds of each color.

5.1.4 Experimental environment

All experiments were carried out within the experimental environment shown in figure 5.5. A black, non-reflecting cloth was used to control the background. While attempts were made to control the lighting by directing the illumination source against walls, in order to generate ambient lighting, good results were obtained with ambient daylight diffused with a white cloth.

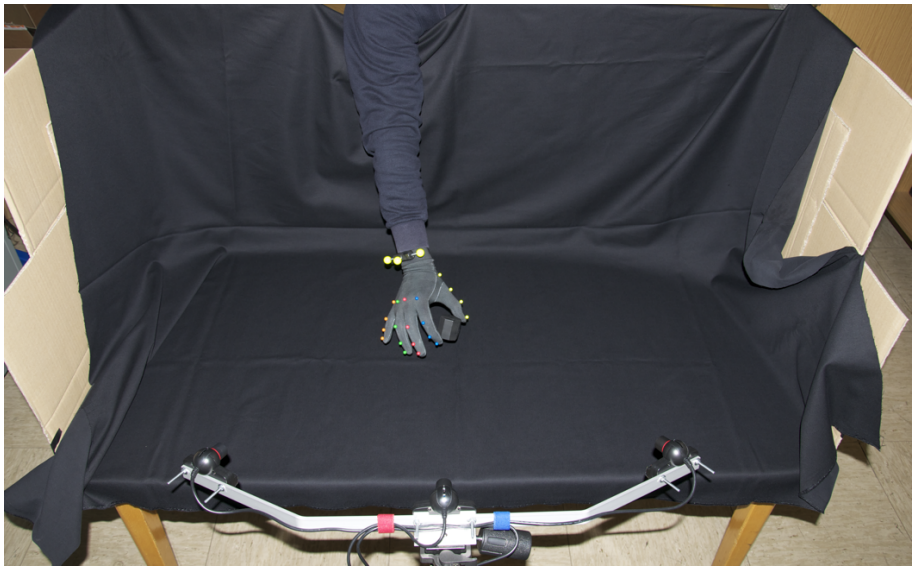


Figure 5.5: The experimental environment. The background was controlled with a black cloth. In order to make use of the ambient daylight, the environment was positioned in front of windows covered with a white cloth, in order to diffuse the light.

5.2 Software architecture

Development of the software application was carried out using C++ as the programming language. A number of factors contributed to that choice, such as availability of computer vision libraries, portability and performance. Furthermore the open source aspect was considered important.

The OpenCV framework [Wil10] was chosen as the computer vision library based on its set of features and its portability. The framework was used to implement most of the image processing workflow as well as the stereo vision workflow.

The Qt framework [Nok11] was chosen for the purpose of implementing the user interface. Apart from the part concerned with the user interface, the Qt framework offers a wealth of additional functionality, such as networking or threading. Since the open source Qt framework is available on all the major desktop platforms, it allows to develop a cross-platform solution with a single code base. Although the Qt framework is written in C++, it offers bindings to other programming languages, such as Java or Python. In combination with the Qt framework the OpenGL® specification [Khr04] was used for visualization purposes.

The complete developed application consists of a total of 41 files with approximately 5000 actual code lines, describing 19 classes which represent four logically separated modules. The complete source code is available on the accompanying CD.

5.2.1 Modular design

Due to infeasibility of a real-time solution with an image acquisition rate of 60 frames per second, the core functionalities of image acquisition and image processing were realized within two different modules. Figure 5.6 shows the main application window.

The main application window offers the choice between four user interface modules designed to fulfill the task of

1. Acquisition of images of the calibration object for the purpose of single and pairwise camera calibration.
2. Calibration of thresholds for color segmentation based on the color sample plates.
3. Recording of hand motion sequences (first core functionality).
4. Processing and visualization of the recorded hand motion sequences (second core functionality).

Figure 5.6: The main application window of **HANDle**, the developed software application. It offers the choice between four user interface modules, each reflecting a main step within the bottom-up approach to obtaining a description of the hand configuration.

5.2.2 Design requirements

At the beginning of the development phase the following requirements were established for the software application:

- *Performance*: Application must operate at the highest image acquisition rate possible, in order to reduce movement between subsequent images.
- *Usability*: Application must provide visual feedback for every active camera source by displaying the image stream.
- *Data output*: Every recorded image must be stored on disk and annotated with a timestamp. Calibration and configuration data as well as the results must be stored in a format, which is in compliance with the HANDLE project.

Performance considerations

It was determined impossible to offer a performance anywhere near real-time at high frame rates, when using the three-camera stereo vision setup with a general purpose workstation. This was mainly due to the high computational effort associated with the image processing workflow. Therefore the decision was made to split the recording and processing of the motion sequence.

Nevertheless it was possible to utilize the highest frame rate offered by the cameras, due to direct use of the Video4Linux2 (V4L2) application programming interface (API). Although the OpenCV framework offers functionality to acquire images, which in fact itself is using the V4L2 API, an evaluation of its image acquisition pipeline compared to the direct approach revealed a slower performance. Through direct use of the V4L2 API it was possible to obtain 60 frames per second for all three cameras of the stereo vision setup, simultaneously.

Using the V4L2 API directly, simplified the development of a user interface (see figure 5.7) for control over the settings provided by the camera driver and also offered direct access to the Y'CbCr data stream delivered by the cameras. In contrast to the advantages, the direct use of the V4L2 API led to loss of portability, as this API is only available on the Linux operating system. Therefore a clean interface between the V4L2 related code and the rest of the application code was designed, to simplify porting to another platform.

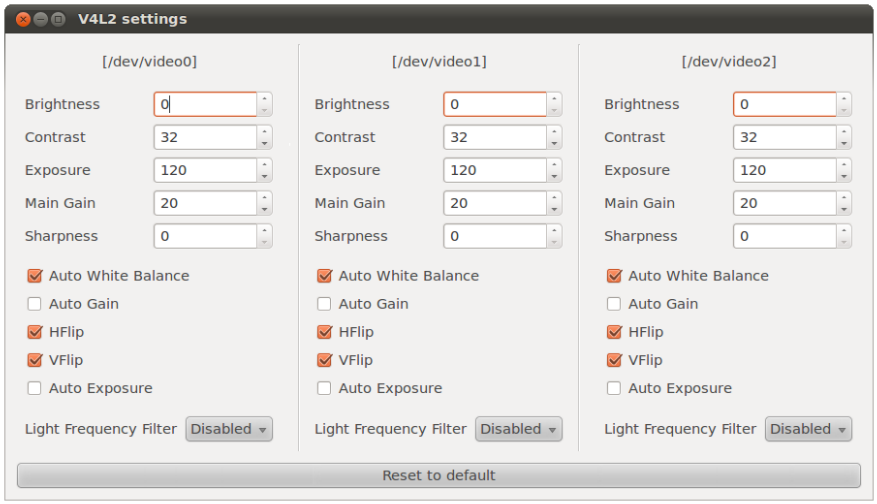


Figure 5.7: The configuration interface for camera related settings. The settings are provided by the camera driver and addressed via the V4L2 API. Apart from allowing to configure basic settings, such as brightness and contrast, the configuration interface allows to disable automatic adjustments, which usually are a major cause for instability in the image processing stage.

5.2.3 Camera calibration

The developed application provides two user interface modules concerned with calibration. The first module, shown in figure 5.8, allows to record pairwise images of the calibration object, which in this thesis was a planar object with a checkerboard pattern, in order to calibrate the cameras individually and as a stereo pair.

Although the OpenCV framework does offer methods to perform single and pairwise camera calibration, evaluation of those methods revealed them not to offer consistent stability of the results. Two out of five experiments resulted in high errors regarding the distortion coefficients, leading to an even more distorted image after undistortion.

Therefore the choice was made not to include this functionality into the software application and instead to use the semi-automated method provided by the widely

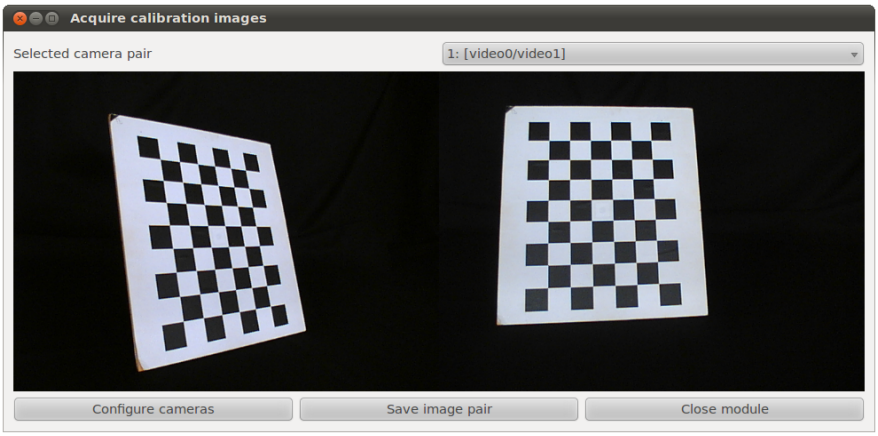


Figure 5.8: The user interface of the camera calibration module. The module allows to record calibration images of the calibration object for both of the stereo pairs. The single and pairwise camera calibration itself is carried out using the Matlab® Camera Calibration Toolbox [Bou10].

used Matlab® Camera Calibration Toolbox [Bou10]. Although the toolbox provided stable results, as can be seen in the following chapter, the calibration was much more time-consuming due to manual selection of corner points in each recorded image. The calibration results obtained for the experimental setup are presented in section 6.2.1.

The OpenCV framework offers a convenient and straightforward way to specify $m \times n$ matrices outside of the source code, via XML files as shown in listing 5.1.

```
<?xml version="1.0"?>
<opencv_storage>
<matrix_name type_id="opencv-matrix">
    <rows>3</rows>
    <cols>3</cols>
    <dt>d</dt>
    <data>
        m_11 m_12 m_13
        m_21 m_22 m_23
        m_31 m_32 m_33
    </data>
</matrix_name>
</opencv_storage>
```

Listing 5.1: OpenCV XML file format for specification of matrices.

Therefore OpenCV compatible XML files were used to store the results obtained

for the sets of intrinsic parameters, distortion coefficients, as well as the euclidean transformation between the cameras of a stereo pair.

5.2.4 Threshold calibration

The second user interface module allows to determine thresholds for the colors of the marker objects attached to the glove and wrist fixture. In order to achieve that, the module expects five sample images of the color sample plates to be taken, holding the plates at different orientations to approximate the color variation across the spherical form of the marker objects. An example can be seen in figure 5.9. It further requires the colored area of the sample plate to fully cover the region of interest (ROI), a 50 × 50 pixel large area in the middle of the image.

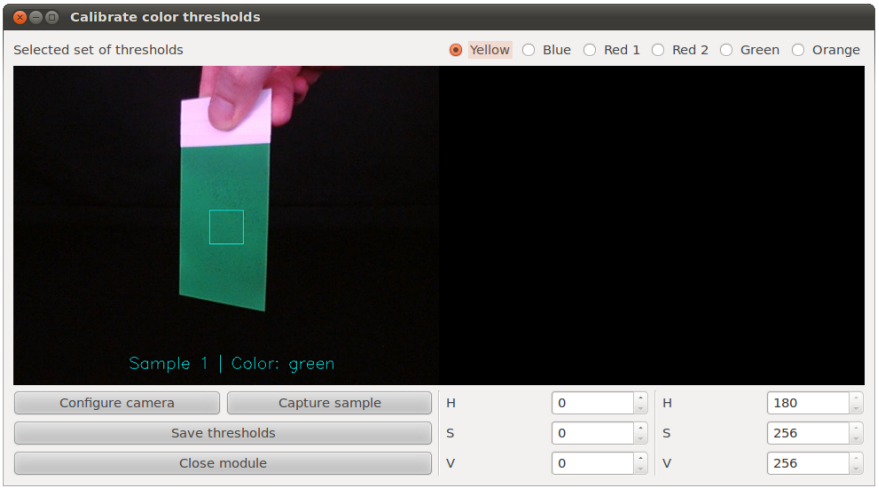


Figure 5.9: The user interface of the threshold calibration module. The module requires an initial set of five images per color based on the color sample plates held at different orientations. Processing of the images yields initial estimates for hue, saturation and value thresholds. Subsequent manual adjustment of the thresholds is done based on visual feedback of a processed feed from a single camera. The results are stored as an XML configuration file.

Following a conversion of the region of interest into the HSV color model, lower and upper thresholds are determined for the hue, saturation and value. Among the values determined for each of the five images per color, the smallest values are selected as the lower threshold and the highest values are selected as the upper threshold. The determined intervals for hue, saturation and value represent an estimate of the color variation of the set of spherical marker objects with the same color.

Although the determined thresholds allowed a general identification of the marker objects in each image, the results did not prove very stable, even after adjustment

of the thresholds by offsets, derived by manual inspection of the color variation. The effect of shadows due to occlusion of the illumination source remained mostly disregarded and separation of colors, which are closely located in the hue range, was sometimes impossible due to overlapping thresholds.

To remedy this problem the color calibration module was extended to allow manual adjustment of initially determined thresholds based on visual feedback from a processed image of a camera feed.

Results determined by the color calibration module are saved into an XML configuration file called *color_thresholds.xml*, which is used by the visualization module. An extract of the configuration file used for the conducted experiments can be found in section A.1.

5.2.5 Motion sequence recording

As addressed previously, the core functionality of recording hand motion sequences was implemented as a separate module. The module offers the possibility of recording image data from all three cameras of the setup, while simultaneously displaying the live feed of every camera for visual feedback. Figure 5.10 shows the user interface implemented for the recording module.

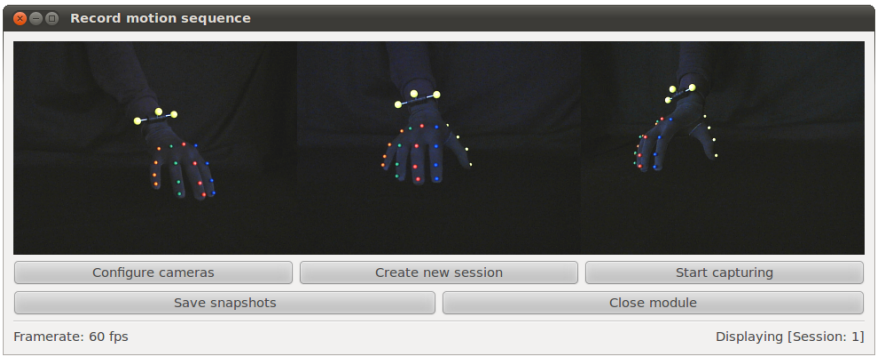


Figure 5.10: The recording module. The module allows to record from all three camera sources, while providing visual feedback by also displaying live camera images. Moreover it is possible to record multiple sessions, without having to restart the application. It is also possible to record snapshots from all three camera streams simultaneously for evaluation purposes.

In order to achieve a high performance, the image acquisition as well as the compression into the JPEG format were split off into separate threads to utilize the four processor cores available on the workstation used for the experiments. To further lighten the load imposed by the displaying operation, OpenGL®-accelerated views were used and the images rendered as textures. Despite the impossibility of achieving a real-time performance of the whole pipeline at even 15 frames per second, it

was possible to obtain a performance of the motion sequence recording module at 60 frames per second for every single camera simultaneously.

Motion sequence annotation

The recording module allows to record multiple hand motion sequences in the form of sessions. Figure 5.11 shows the general file structure of a single session on disk.

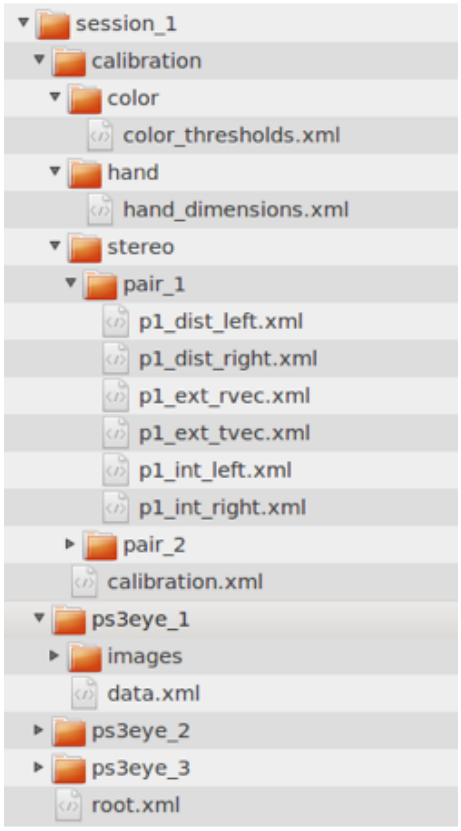


Figure 5.11: General file structure of a single session after a successful recording. The image stream of every camera source is annotated with absolute timestamps (*data.xml*). *root.xml* references the raw image data of every camera as well as the calibration data, created in the previous steps.

In order to comply with the annotation guidelines [HAN09] set for the HANDLE project, every session contains the following files:

- **root.xml**: References the cameras used to acquire the raw image data as well as the calibration data associated with the experiment.

- **data.xml**: References each one of the recorded images, annotated with an absolute timestamp.
- **calibration.xml**: References the calibration for the hand, the thresholds as well as the stereo setup.
- **hand_dimensions.xml**: Contains the measured bone lengths and joint diameters of the subject's hand as well as the offset of the wrist joint in reference to the wrist fixture.
- **color_thresholds.xml**: Contains determined thresholds for hue, saturation and value of each of the marker object colors.
- **p1_*.xml / p2_*.xml**: Contain the camera calibration parameters for both stereo pairs, stored in the previously mentioned OpenCV compatible format.

The files *root.xml*, *data.xml* and *calibration.xml* as well as the folder structure are created automatically upon finishing the recording of a hand motion sequence. The remaining files associated with the calibration need to be inserted into the folder structure manually.

5.2.6 Processing and visualization

The visualization module forms the core entity regarding the processing of recorded hand motion sequences and visualization of the results. It incorporates the methods for image processing and three-dimensional reconstruction as well as the hand pose reconstruction methods described in previous chapters.

The visualization module is also the only module that was not designed to handle live image data, albeit OpenGL®-accelerated views were used to display recorded images, in order to remain consistent.

The visualization module expects the input to be a recorded hand motion sequence in the form of a session as described in the previous section. The user interface of the module allows to navigate the set of recorded images one-by-one or using a slider, while displaying each selected set of images. For the purpose of visual validation, pairwise rectified images are also displayed.

Figure 5.12 illustrates the processing and visualization pipeline implemented within this module. The stages of the pipeline will be looked at briefly in the following.

Image rectification

Using the stereo calibration data, every set of images is initially subjected to pairwise rectification. Performance could be improved by executing the image processing steps first, followed by rectification of only the two-dimensional marker object coordinates extracted from the images. Nevertheless it was decided to rectify complete

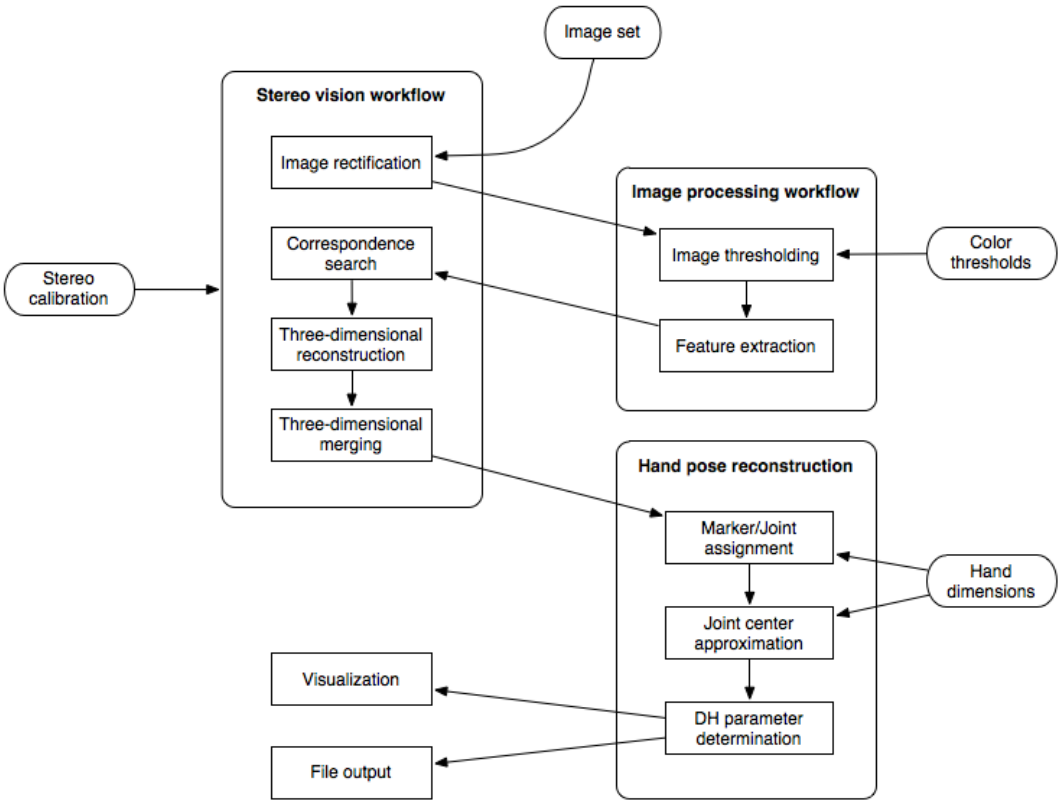


Figure 5.12: Flowchart of the processing and visualization pipeline used to process the recorded hand motion sequences. It incorporates the methods described in the previous chapters, in order to obtain a description of the hand configuration based on a set of images. In addition to visualization of the result, the obtained description is also output as an XML file.

images, in order to provide the result as visual feedback. This proved useful in identifying loss of marker object projections due to rectification warping, in cases where the hand was moved too close to the cameras, past the boundary of the working area.

Image processing

The determined color thresholds are used to threshold both pairs of rectified images, after an initial noise reduction with two iterations of the median filter. Since every single color is thresholded separately, subsequent feature extraction yields sets of two-dimensional coordinates for marker object projections of a single finger, greatly simplifying the step of marker/joint assignment.

Although the color "yellow" is used for the thumb as well as the wrist, the execution

of a simple comparison of extracted shape sizes proved sufficient to differentiate between both entities.

Segmentation sometimes produced unwanted foreground pixels due to colors, which are closely located regarding the hue circle of the HSV color model. This was the case with the colors red and orange, where pixels around the perimeter of the projection of an orange marker object were in the range of the color red. Therefore an evaluation of the extracted shapes based on their eccentricity and minimum size was implemented, leading to reliable results.

Three-dimensional reconstruction

The search for corresponding marker object projections between two rectified images was implemented using the epipolar constraint, based on the fundamental matrix for each of the rectified pairs. Although the constraint violation between two corresponding projections was mostly very small, a subsequent correction of the determined correspondences using the optimization method [HS97] provided by the OpenCV framework, was included.

The middle camera that is part of both stereo pairs, was implemented to supply the coordinate frame for three-dimensional reprojection of the points using the disparity between the correspondences. Inversion of the rectifying transformation for the middle camera was included, in order to transform points reconstructed in both pairs into the same coordinate frame. Subsequent merging of points was implemented using thresholds for differences along the X-, Y- and Z-axes.

Hand pose reconstruction

Given that the color coded marker objects identify each finger as a separate entity of the hand, the task of marker/joint assignment is greatly simplified. Distance measurements are sufficient to obtain an assignment (see chapter 4). The implemented approach uses bone link lengths of the subject's hand as an initial guard against misassignment, in the case of an incompletely reconstructed marker object set.

Although the final result, a description of the articulated hand model by Denavit-Hartenberg parameter sets, does not depend on the approximation of the joint centers and can be produced anyway, the accuracy of the data obviously does. Despite a small shift, the marker objects are subjected to during articulated motion, it is reasonable to assume the location of a joint to be under a marker object. Therefore joint center approximation was implemented using joint diameter data obtained from the subject's hand. The results were determined to be valid approximations based on visual evaluation of the hand model.

Concluding the processing part of the pipeline, determination of DH parameter sets was implemented based on geometric inspection of the kinematic chain of every finger, from the wrist joint to the fingertip.

Data output

The obtained DH description of the 26 DOF articulated hand model is utilized to visualize the results within an OpenGL® environment. Figure 5.13 shows an example of a fully reconstructed articulated hand model in a resting configuration.

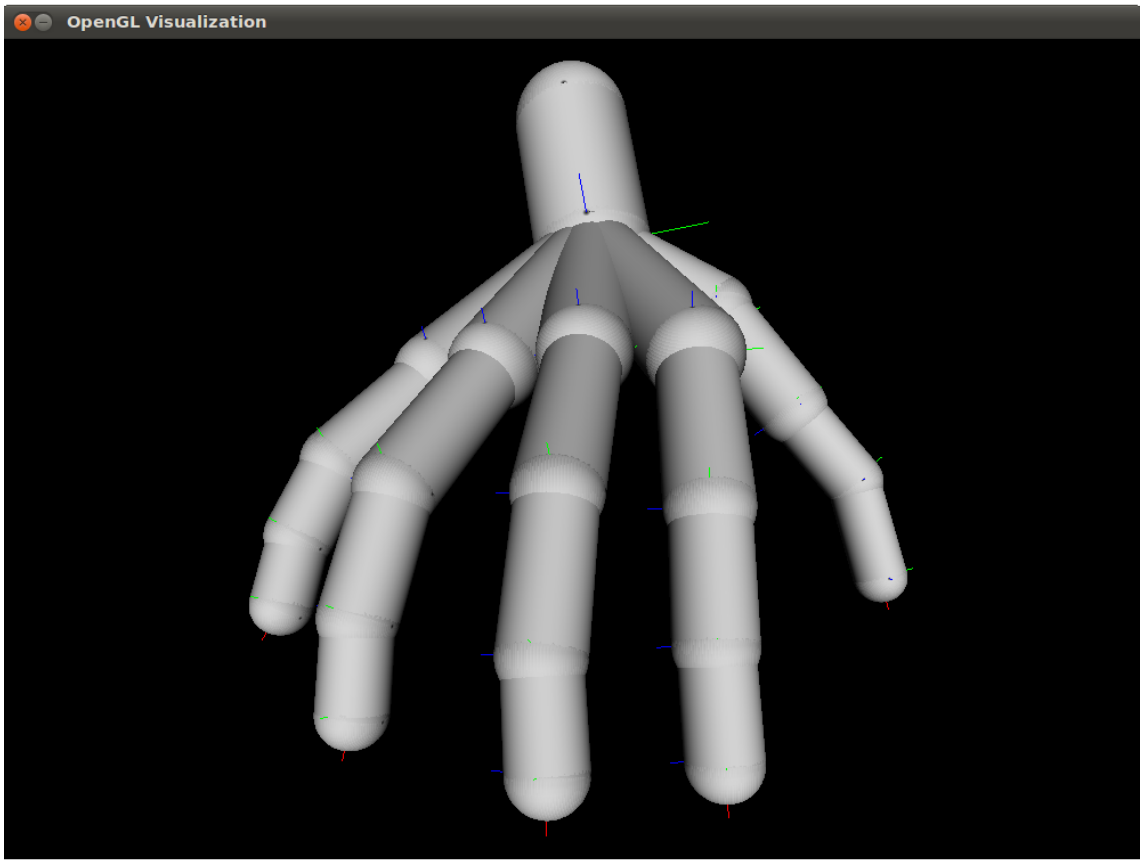


Figure 5.13: Visualization of the reconstructed hand configuration based upon determined Denavit-Hartenberg parameter sets. The figure shows the basic position that the hand must be positioned in, in order to allow the reconstruction of the full pose and the recalibration of the bone length dimensions.

For the purpose of post-processing of the obtained parameters, the description obtained from each set of images is also stored on disk. Use of the XML format was implemented, in order to store the data in a well-defined format, which can be easily parsed and also allows exchange of the obtained data within the HANDLE project.

5.3 Conclusion

This chapter presented an overview over the hardware components used in the experimental setup, with a specific focus on the low-cost aspect and the obtainable performance. Moreover the architecture of the developed software application was discussed.

The software application serves as a prototype for the demonstration of the abilities of the whole system. While the application does not focus on the aspect of real-time operation, it does focus on image acquisition at the highest frame rate achievable with the setup. Apart from that, it enables visual evaluation of the quality of the results, while also making the results available in the XML file format, to allow post-processing and data exchange within the HANDLE project.

Experimental results

6

In order to evaluate the approach described in the previous chapters as well as the experimental stereo vision setup, several experimental results were obtained that will be discussed in this chapter.

At first performance related aspects of the experimental system will be discussed in reference to the synchronicity of the image acquisition and the computational load associated with the image processing and stereo reconstruction.

The accuracy and general performance of the stereo vision setup will be evaluated next, based on a calibration object with a known length.

Concluding this chapter an evaluation of the accuracy of the reconstructed hand model will be presented, based on two recorded hand motion sequences: tip-to-tip grasping and rotation of a small object.

6.1 Performance

As mentioned in the previous chapter, in order for a stereo vision setup to be usable, the images of all cameras must be acquired synchronously. During the initial testing of the experimental setup it was found that operation of all three cameras at 60 frames per second, resulted in almost synchronous image acquisition. For that the timestamp data of all image streams was evaluated, in combination with a visual evaluation of the image stream itself, showing the recording of a counting timer. Figure 6.1 shows three subsequent image sets from the recorded image sequences.

As the figure clearly shows, the images were recorded synchronously by the first and the second camera with the third camera trailing very slightly behind. In fact it was found that although the shown synchronicity could not be uphold consistently, most of the time a pair of cameras would operate synchronously. The deviation of any camera as the third camera was found not to exceed a range of 2 - 6ms. The effect of the deviation on the stereo reconstruction process was found to be indeterminable due to general volatility of the image processing results.

Figure 6.1: Three image sets from the recorded sequences of a counting timer (milliseconds). The image sets are displayed from top to bottom, with the images of each single camera from left to right.

In order to quantify the computational load of the image processing, stereo and hand pose reconstruction stages, an evaluation was conducted over a series of 1000 image sets. The results of the evaluation are shown in tables 6.1 - 6.3.

Operation	Mean time (ms)	Std. deviation (ms)
Noise reduction	15.4990	±1.3605
Thresholding	5.8692	±1.1577
Morph. filtering	3.1690	±1.1236
Feature extraction	19.9930	±1.6591

Table 6.1: Computational load of the image processing stage (based on 1000 image sets).

As the visualization module, which the image processing and stereo reconstruction stage were implemented in, was designed unthreaded, clearly an increase in performance could be gained by parallelizing most of the image processing steps. Since image rectification is particularly costly, performance could be increased further by only rectifying the two-dimensional points gained from feature extraction, instead of the complete images.

Operation	Mean time (ms)	Std. deviation (ms)
Image rectification	20.9220	±1.4135
3D reconstruction	2.5440	±0.7602
3D point merging	0.0140	±0.1175

Table 6.2: Computational load of the stereo reconstruction stage (based on 1000 image sets).

Operation	Mean time (ms)	Std. deviation (ms)
Marker/Joint assignment	0.2430	±0.4735
Center approximation	0.0279	±0.1649
DH description	0.2022	±0.4442

Table 6.3: Computational load of the hand pose reconstruction (based on 1000 image sets).

6.2 Stereo vision setup

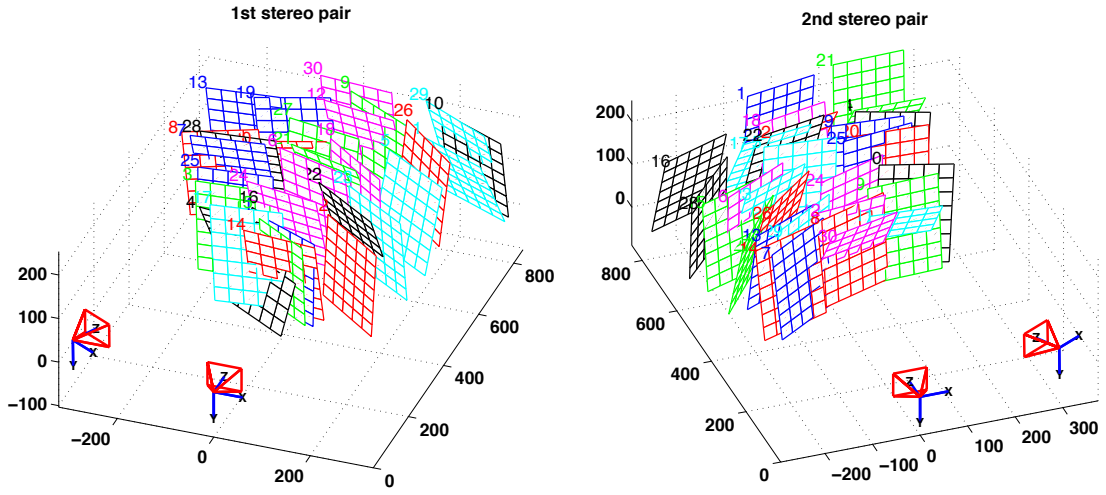
The calibration of the three-camera stereo vision system was conducted using a planar board with a checkerboard pattern. A total of two stereo pairs were calibrated (shown in figure 6.2), based on 30 recorded images of the calibration object for each pair. The middle camera was part of both pairs and was also chosen to provide the frame of reference for the complete setup.

The calibration of the stereo vision setup was obtained using the Matlab® Camera Calibration Toolbox [Bou10]. In order to obtain the intrinsic parameter sets including the distortion coefficients, each camera was calibrated separately. Table 6.4 shows the reprojection error obtained for both cameras of the first and second stereo pairs respectively.

Pair	Camera	X-axis (pixel)	Y-axis (pixel)
1	Left	±0.08437	±0.08420
1	Right	±0.09012	±0.08694
2	Left	±0.07672	±0.08130
2	Right	±0.09545	±0.09829

Table 6.4: Reprojection error for both cameras of the first and second stereo pair, obtained through calibration using the Matlab® Camera Calibration Toolbox.

Subsequent pairwise stereo calibration yielded the euclidean transformation between



a **b**

Figure 6.2: Calibration of the three-camera stereo vision system. 30 images of the calibration object were recorded, in order to obtain the camera parameters as well as the euclidean transformation for both stereo pairs shown in figures **a)** and **b)**.

both cameras of each pair. Based on the obtained translation vector between the cameras of each pair, the length of the baselines was determined as 331.0562mm and 331.7844mm, for the first and second pair respectively.

6.2.1 Stereo calibration results

The following tables 6.5 - 6.8 show the results of pairwise stereo calibration, which were obtained using the Matlab® Camera Calibration Toolbox.

First stereo pair

Parameters	Left camera	Right camera
Focal length (px)	[539.59297, 544.48877]	[542.30998, 547.22378]
Principal point (px)	[289.59646, 259.95388]	[317.74818, 237.10474]
Radial distortion	[-0.10789, 0.15179]	[-0.10705, 0.12898]
Tangential distortion	[-0.00109, -0.00116]	[-0.00267, -0.00130]

Table 6.5: Intrinsic parameters obtained for the left camera and the right camera of the first stereo pair.

Parameters	X-axis	Y-axis	Z-axis
Rotation vector (deg.)	0.02876	0.49873	0.00954
Translation vector (mm)	-321.18156	-3.65293	80.17039

Table 6.6: Euclidean transformation between the cameras of the first stereo pair. Given are the rotation vector as well as the translation vector.

Second stereo pair

Parameters	Left camera	Right camera
Focal length (px)	[542.04782, 547.44179]	[543.95326, 547.90733]
Principal point (px)	[314.79313, 237.90660]	[329.75157, 229.84456]
Radial distortion	[-0.11092, 0.15687]	[-0.11636, 0.16046]
Tangential distortion	[-0.00248, -0.00181]	[-0.00015, -0.00474]

Table 6.7: Intrinsic parameters obtained for the left camera and the right camera of the second stereo pair.

Parameters	X-axis	Y-axis	Z-axis
Rotation vector (deg.)	0.02233	0.50637	-0.00035
Translation vector (mm)	-320.54696	-1.92223	85.59715

Table 6.8: Euclidean transformation between the cameras of the second stereo pair. Given are the rotation vector as well as the translation vector.

Although Zhang's approach (described in section 3.2.4) does not consider tangential distortion, the implementation within the Camera Calibration Toolbox does. Therefore tables 6.5 and 6.7 show estimated values for two coefficients of tangential distortion. It should be noted that the coefficients of the tangential distortion are negligibly small compared to the coefficients of the radial distortion. This validates the claim about the dominance of radial distortion [Tsa87, Zha00].

The images used for the calibration as well as toolbox-specific files with the obtained results, are available on the accompanying CD.

6.2.2 Evaluation of the stereo setup

In order to quantify the quality of obtained calibration and also inevitably the quality of the image processing, a test object, shown in figure 6.3, with a known length of 100mm was used.

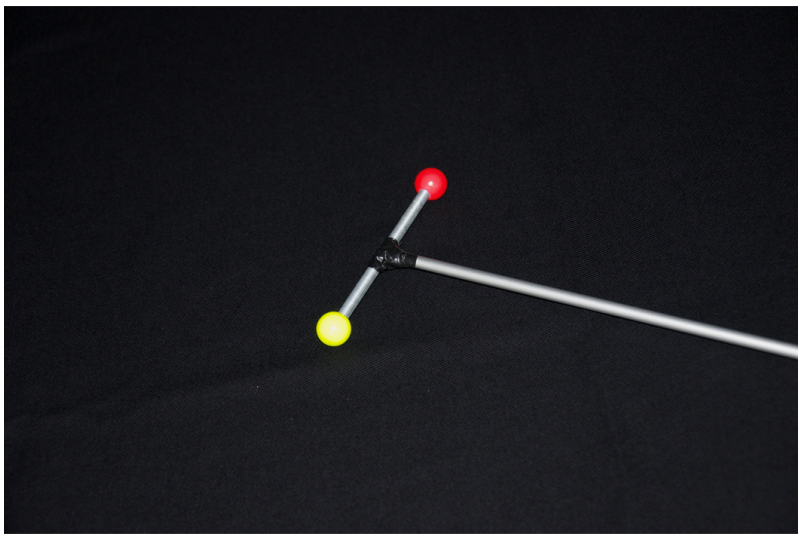


Figure 6.3: Test object with a known length of 100mm between the centers of the marker objects, used for the purpose of evaluation of the quality of stereo reconstruction.

The test object was moved towards and away from the cameras of the setup to measure the stability of the stereo reconstruction within the depth of the experimental environment. A maximum depth of approximately 850mm was measured. Length of the object was measured based on three-dimensional points of the marker objects as reconstructed from each pair separately as well as from both pairs, after merging of the points by averaging as described in section 3.6. Figure 6.4 shows the result of a single motion of the test object within the experimental environment.

Based on a full reconstruction of the test object in each of the stereo pairs, a minimum distance to the cameras was determined to be between 400 - 450mm, with no usable results beyond that point. Regarding the available maximum depth within the experimental environment, a usable three-dimensional reconstruction of both marker objects could be obtained throughout.

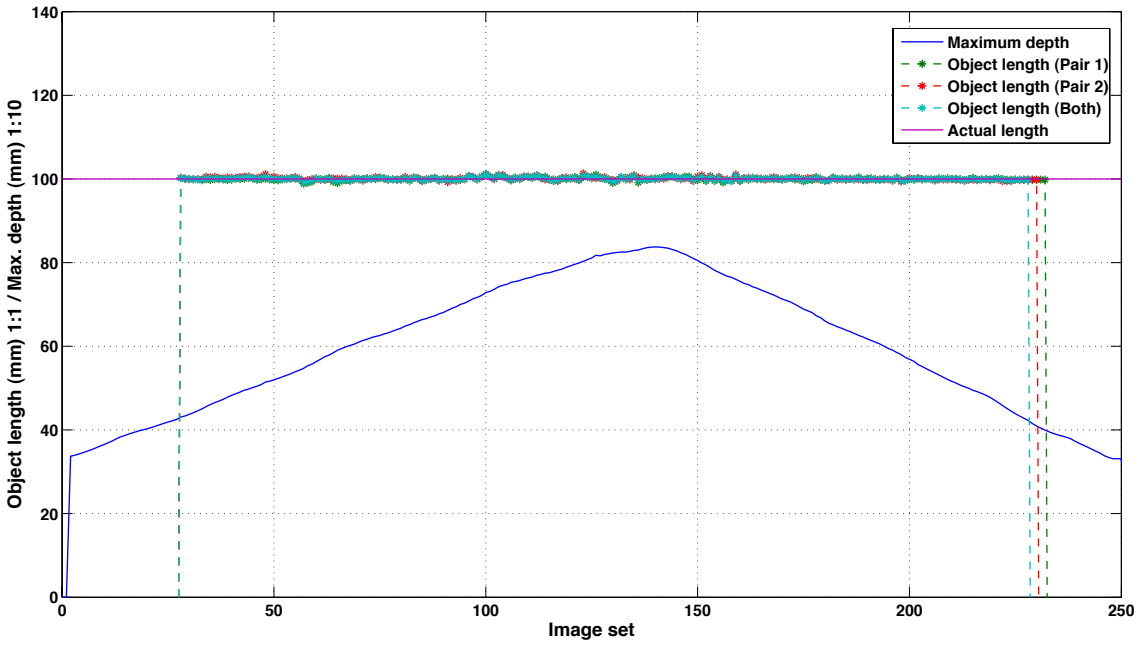


Figure 6.4: Evaluation of the calibration quality as well as the quality of the image processing using a test object with a known length of 100mm between the marker objects. Shown are the object lengths determined from the first and second pairs separately as well as after the merging.

A visual representation of the deviation of the measured object length from the ideal length is shown in figure 6.5.

Calculation of the object length based on points reconstructed from each stereo pair as well as after merging of these points, yielded a maximum deviation of less than 1.5mm. Table 6.9 quantifies the results.

Stereo pair	Mean length deviation (mm)	Std. deviation (mm)
First pair	-0.0038	±0.3438
Second pair	0.0860	±0.3557
Both pairs	0.0395	±0.3042

Table 6.9: Deviation from the actual object length, measured by the stereo vision system.

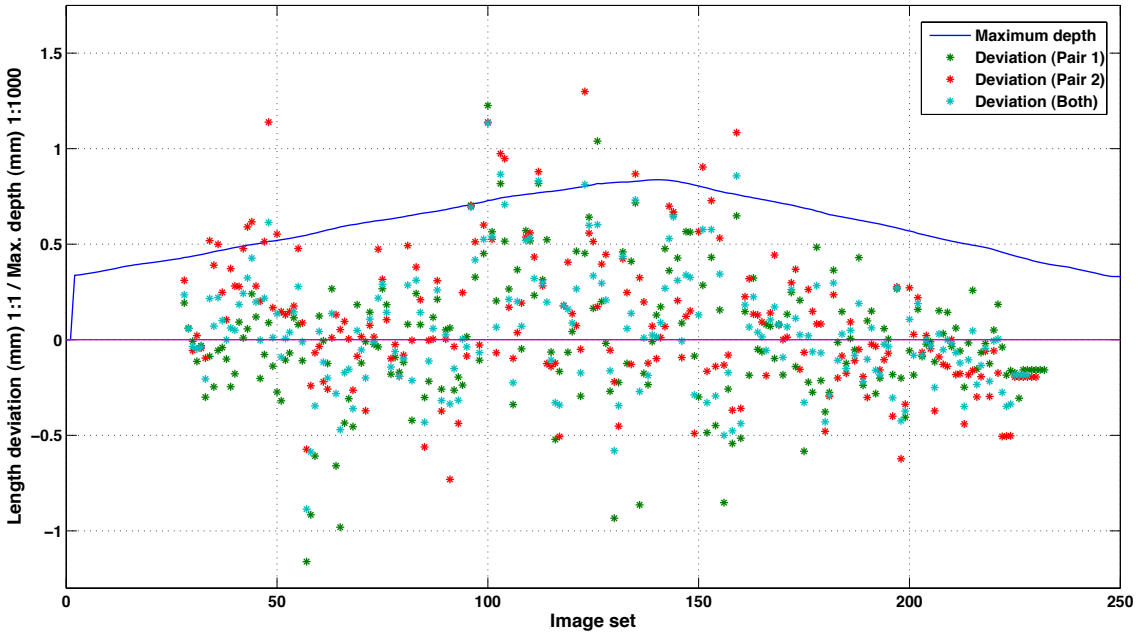


Figure 6.5: Deviation of the measured object length, measured based on reconstructed points from both stereo pairs as well as after merging of the reconstructed points regarding the same marker objects.

In addition to the deviation of the measured object length, the ability to merge pairwise three-dimensional representations of the same marker object was evaluated. Figures 6.6 and 6.7 visualize the measured coordinate differences of the yellow and red marker objects with reference to both stereo pairs.

As expected, the coordinate differences increased with increasing distance of the marker object from the cameras. Although the highest difference was consistently found along the X-axis, the coordinate difference along the Z-axis showed the highest volatility. Table 6.10 shows the mean differences and the associated standard deviation.

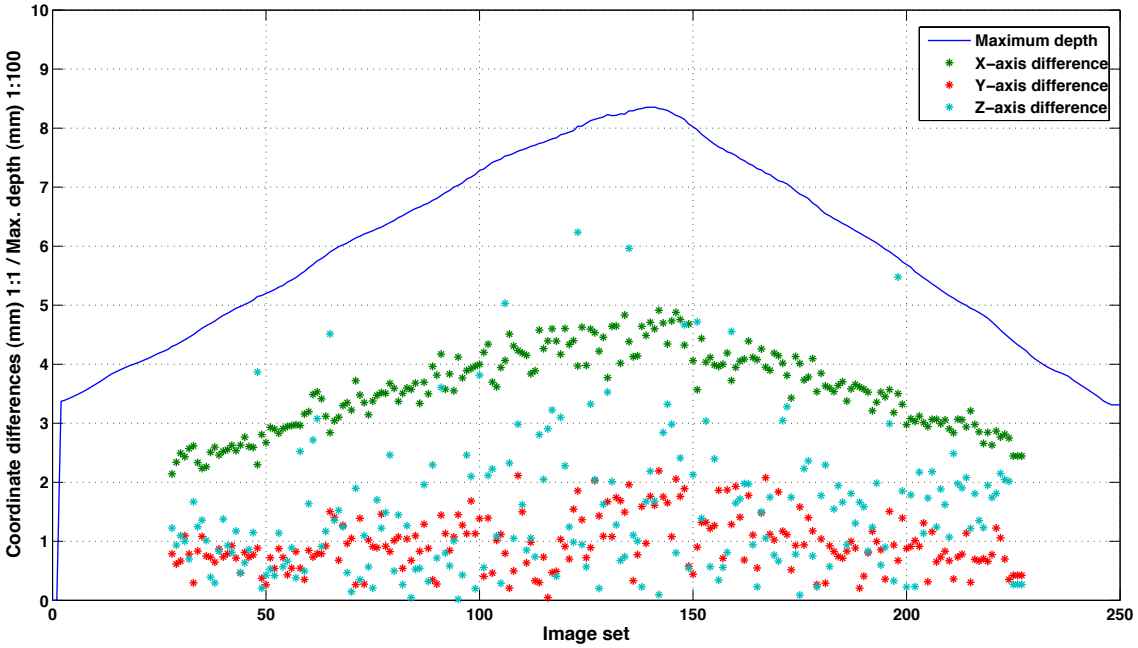


Figure 6.6: *Yellow marker object*: Coordinate differences between three-dimensional representations reconstructed from the first and second stereo pairs.

Axis	Mean difference (mm)	Std. deviation (mm)
X	3.5691	$\pm$0.6751
Y	0.9649	$\pm$0.4558
Z	1.5293	$\pm$1.1475

Table 6.10: *Yellow marker object*: Deviation of coordinates between both stereo pairs.

As expected the results for the red marker object shown in figure 6.7 and table 6.11, were very similar to the results for the yellow marker object.

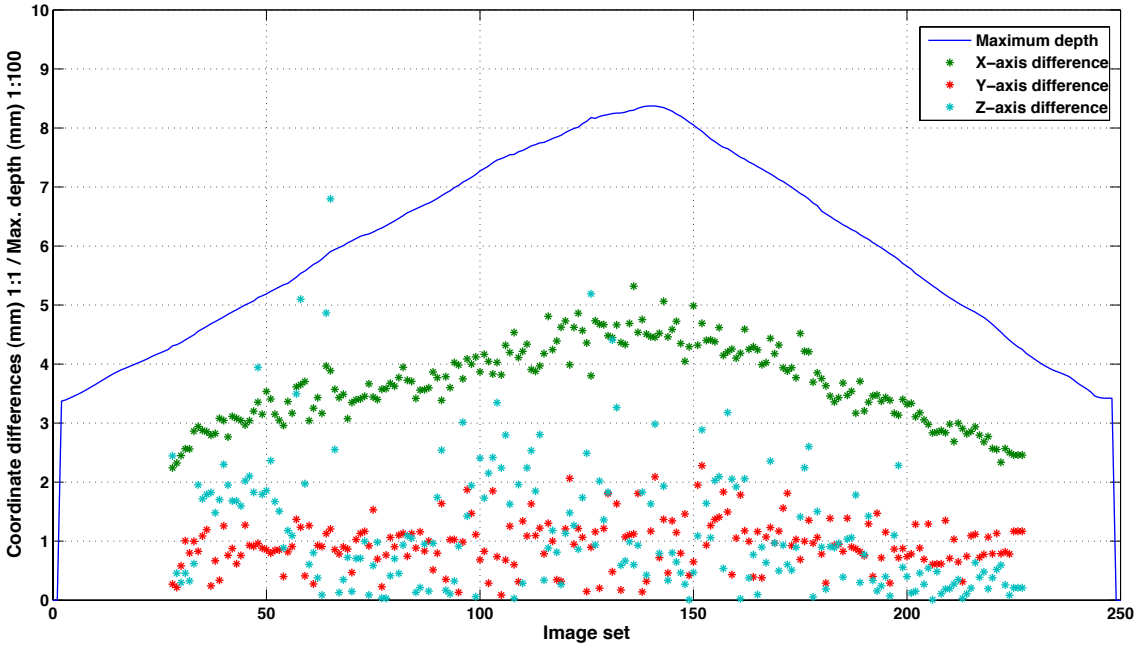


Figure 6.7: *Red marker object*: Coordinate differences between three-dimensional representations reconstructed from the first and second stereo pairs.

Axis	Mean difference (mm)	Std. deviation (mm)
X	3.6755	$\pm$0.6653
Y	0.9496	$\pm$0.4131
Z	1.1891	$\pm$1.0933

Table 6.11: *Red marker object*: Deviation of coordinates between both stereo pairs.

Based on the obtained results regarding the maximum difference along each of the three axes, the thresholds for merging of three-dimensional points reconstructed by both stereo pairs were chosen as 6.0mm, 3.0mm and 7.0mm, for the X-, Y- and Z-axes respectively. The thresholds were chosen higher than the maximum values shown in figures 6.6 and 6.7, in order to allow some room for error due to the volatility of the results produces by the image segmentation process described in section 2.4.

6.3 Hand pose reconstruction

Due to lack of ground truth data regarding the positions of all joints and the angles of their rotation, direct evaluation of the accuracy of the reconstructed hand poses was not possible.

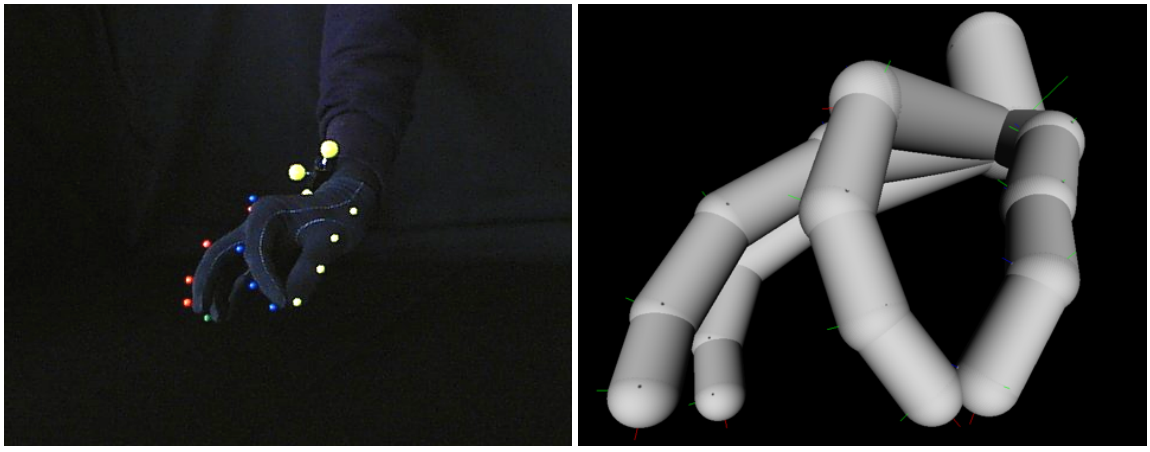
Therefore the accuracy was evaluated based on two aspects

1. Repeatability regarding the joint angles as well as the fingertip center distances
2. Deviation of the measured bone segment lengths regarding the lengths obtained from the subject's hand

Two experiments were conducted recording a tip-to-tip grasping motion of the thumb and the index finger and the motion associated with rotation of a small object.

6.3.1 Tip-to-tip grasping

Regarding the motion associated with tip-to-tip grasping with the thumb and the index finger (shown in figure 6.8), a sequence of seven grasping repetitions was recorded.



a **b**

Figure 6.8: Tip-to-tip grasping with the thumb and the index finger. Figure **a)** shows one of the recorded source images, while figure **b)** shows the result of the hand pose reconstruction.

Visual evaluation of the result of the hand pose reconstruction shown in figure 6.8 b) shows a reasonably accurate visualization. It should be noted that the visual representation does not account for the actual thickness of the fingers, considering joint and fingertip diameters of the bone structure only. Therefore the distance

between the fingertips should be considered as the distance between the *centers of the fingertips*.

Moreover, it should also be noted that in order to obtain a more consistent visualization, the spheres representing the tips of the thumb and the index finger were drawn with a diameter of 16mm, deviating from the diameters of 9.5mm and 9.0mm obtained from the subject's hand (see section 4.4). Due to this a visible merging of the spheres (with the fingertips touching) was expected. It indirectly verifies the accuracy of the joint centers obtained by the approach described in section 4.4.

In order to evaluate the repeatability of the joint angles and the fingertip center distance, only the degrees of freedom associated with the flexion/extension of the thumb and the index finger were considered, as these showed the most action. Figure 6.9 shows the joint angle trajectories produced by three repetitions of tip-to-tip grasping.

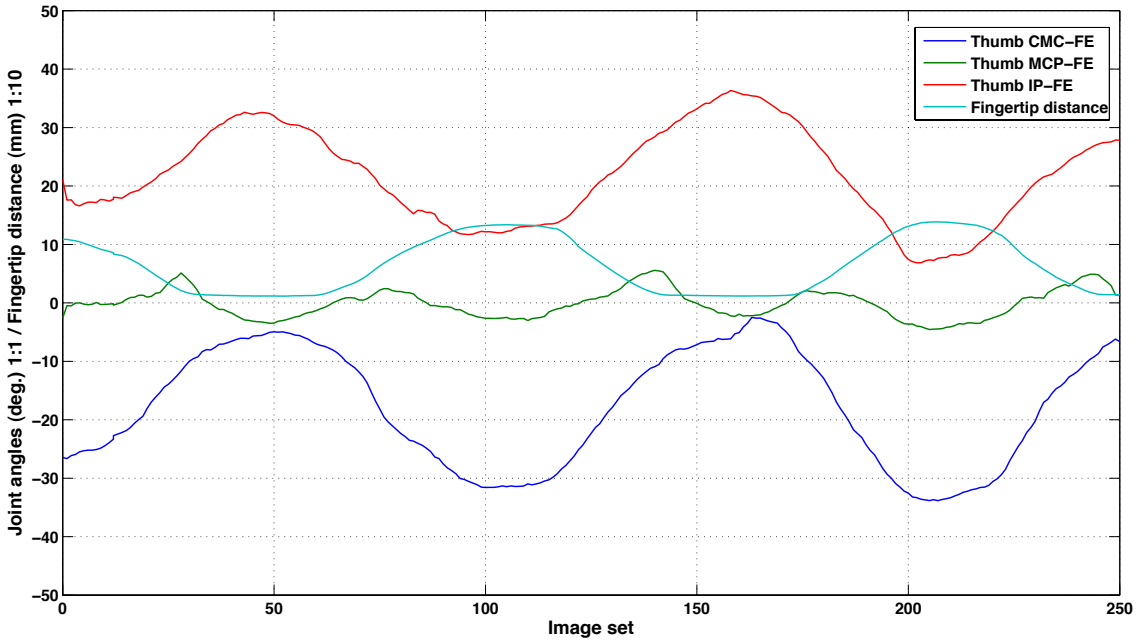


Figure 6.9: Joint angle trajectories of joints of the thumb regarding the DOF associated with the flexion/extension in combination with the fingertip center distance. For better visibility only three out of seven repetitions are shown.

The trajectories shown in the figure clearly display a motion pattern, with the occasional deviation, such as that of the thumb's interphalangeal joint during the transition from the second to the third contact of the fingertips. As expected the fingertip center distance remains stable during the contact phases.

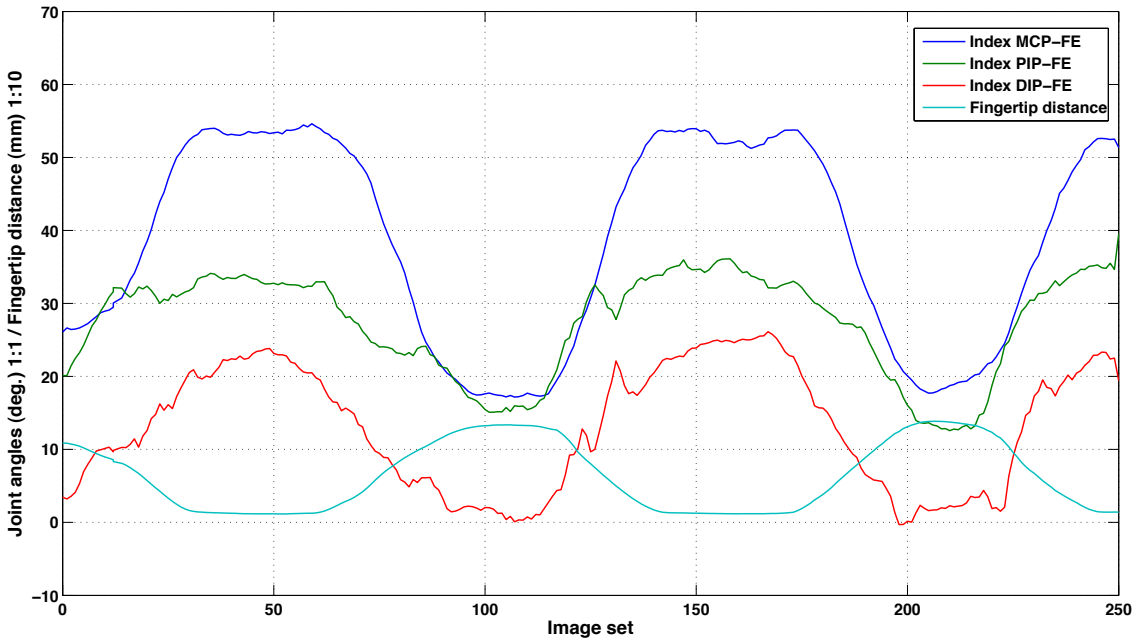


Figure 6.10: Joint angle trajectories of joints of the index finger regarding the DOF associated with the flexion/extension in combination with the fingertip center distance. For better visibility only three out of seven repetitions are shown.

The joint angle trajectories of the joints of the index finger shown in figure 6.10 also show a clear motion pattern, albeit slightly more unstable.

For each of the seven tip-to-tip grasping repetitions, a sample of all variables was taken in the middle of the fingertip contact phase. The repeatability regarding the joint angles as well as the fingertip center distance based on the seven samples taken, is shown in tables 6.12 and 6.13.

Based on the relatively small standard deviation for each joint's angle, the approach to hand pose reconstruction can be considered to provide a relatively accurate model of the articulated hand.

It should be noted that the fingertip center distance does not represent the distance between the actual fingertips, but between the fingertip bone centers approximated with the method described in section 4.4. In order to obtain the distance between the actual fingertips, the radii of both fingertip bone diameters need to be subtracted as well as the thickness of the skin and the glove fabric.

According to the measurements obtained from the subject's hand, shown in table 4.8 in section 4.4, the radii of 4.75mm and 4.5mm need to subtracted. Regarding the result in table 6.13 this leaves approximately 3.0mm to account for the flesh, skin and the glove fabric. Viewed in combination with the very small standard deviation, the

Joint	Mean angle (deg.)	Std. deviation (deg.)
Thumb CMC	-5.3598	±0.9755
Thumb MCP	-1.8755	±0.9741
Thumb IP	34.8212	±3.5307
Index MCP	52.5035	±1.3475
Index PIP	35.5558	±1.7547
Index DIP	24.8104	±1.9376

Table 6.12: Repeatability regarding the joint angles during the contact phases.

	Mean distance (mm)	Std. deviation (mm)
Fingertip center distance	12.1850	±0.4650

Table 6.13: Repeatability regarding the fingertip center distance during the contact phases.

approximation of the joint centers can therefore be considered to provide reasonably accurate results regarding the joint positions. Moreover the result attests very good repeatablity to the hand pose reconstruction.

To further evaluate the validity of the produced results the widely accepted DIP/PIP constraint

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP} \tag{6.1}$$

was considered. The result shown in table 6.14 verifies a relatively good approximation.

	Mean ratio	Std. deviation
DIP/PIP ratio	0.5723	±0.1513

Table 6.14: Evaluation of the DIP/PIP constraint over the full sequence of 750 image sets of the tip-to-tip grasping motion.

As the second aspect to judge the accuracy on, the deviations of the measured bone segment lengths compared to the lengths obtained from the subject's hand, were determined. Table 6.15 shows the obtained results.

Although the table partly shows high length deviations, the associated standard deviation is relatively small. Therefore most of the deviation can be attributed to

Segment	Thumb	Index finger	Middle finger	Ring finger	Little finger
1 (mm)	3.3327 ±0.8271	0.7340 ±0.5966	2.2213 ±0.5856	0.5486 ±0.4071	1.3869 ±0.0002
2 (mm)	9.8878 ±0.5704	5.5558 ±1.1607	3.8951 ±0.7511	2.5824 ±0.4782	0.1346 ±0.0001
3 (mm)	10.2642 ±0.8031	1.9731 ±0.5859	0.5116 ±0.3567	0.3118 ±0.2194	0.6649 ±0.0001
4 (mm)	1.7599 ±0.3179	0.4866 ±0.5595	2.4009 ±0.4870	2.6575 ±0.6049	3.5257 ±0.0004

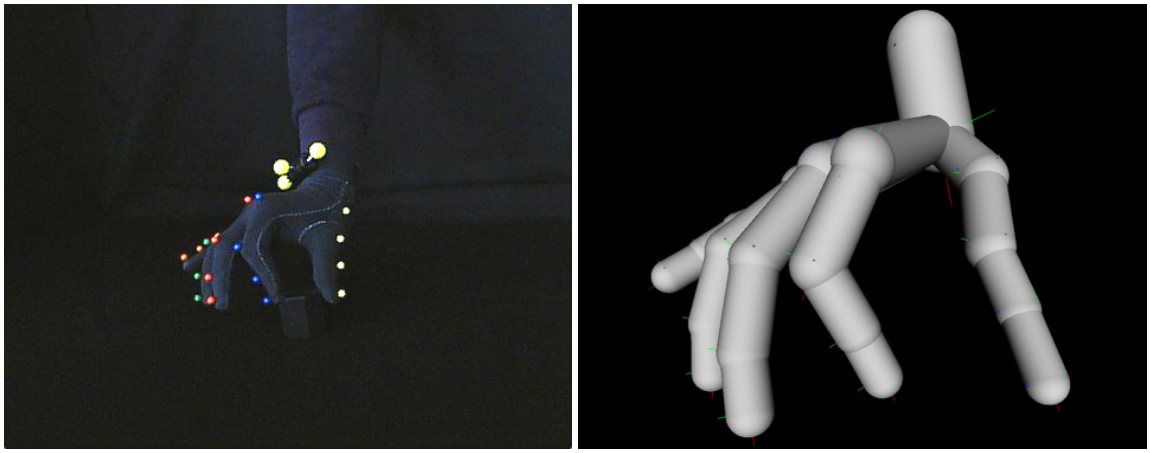
Table 6.15: Mean deviations of bone segment lengths compared to the lengths obtained from the subject's hand. Shown are the mean length deviation and the associated standard deviation. Obtained over the full sequence of the tip-to-tip grasping motion.

an incorrect placement of the marker object over the respective joint and the results can be considered a verification of the consistency of the reconstructed model.

It should be noted that the standard deviation of the bone segment lengths of the little finger in table 6.15 was determined as extremely small due to availability of only a few samples of the fully reconstructed kinematic chain because of self-occlusion.

6.3.2 Object manipulation

Manipulation (rotation) of a small object shown in figure 6.11, was the second recorded sequence for the evaluation of the hand pose reconstruction approach. Just as with the first experiment, seven repetitions were recorded.



a **b**

Figure 6.11: Object manipulation by example of rotation of a small octagonal object with the thumb and the index finger. Figure **a)** shows one of the recorded source images, while figure **b)** shows the result of the hand pose reconstruction.

In order to be able to disregard the object during image segmentation, a black object was chosen, with the unfortunate effect of being mostly invisible in the recorded images. During the recording of the object manipulation the object was touched at the corners of the octagonal plane and rotated clockwise for about 60° during each repetition. The distance along the octagonal plane of the object from corner to corner was measured at 40mm.

Similarly to the first experiment only the degrees of freedom associated with the flexion/extension of the thumb and the index finger were considered as well as the fingertip distance. Figure 6.12 shows the joint angle trajectories produced by three repetitions of object rotation.

The fingertip center distance remains stable throughout the contact phase with the object, yielding a clear pattern. Although the joint angle trajectories show a motion pattern, contrary to the results of the first experiment, sampling of all joint angle variables in the middle of the contact phase of the fingertips with the object, for the most part did not yield any acceptable repeatability. The results are shown in table 6.16.

Although the visual representation of the joint angle trajectories of the joints of the index finger, shown in figure 6.13, allows to distinguish a motion pattern, the

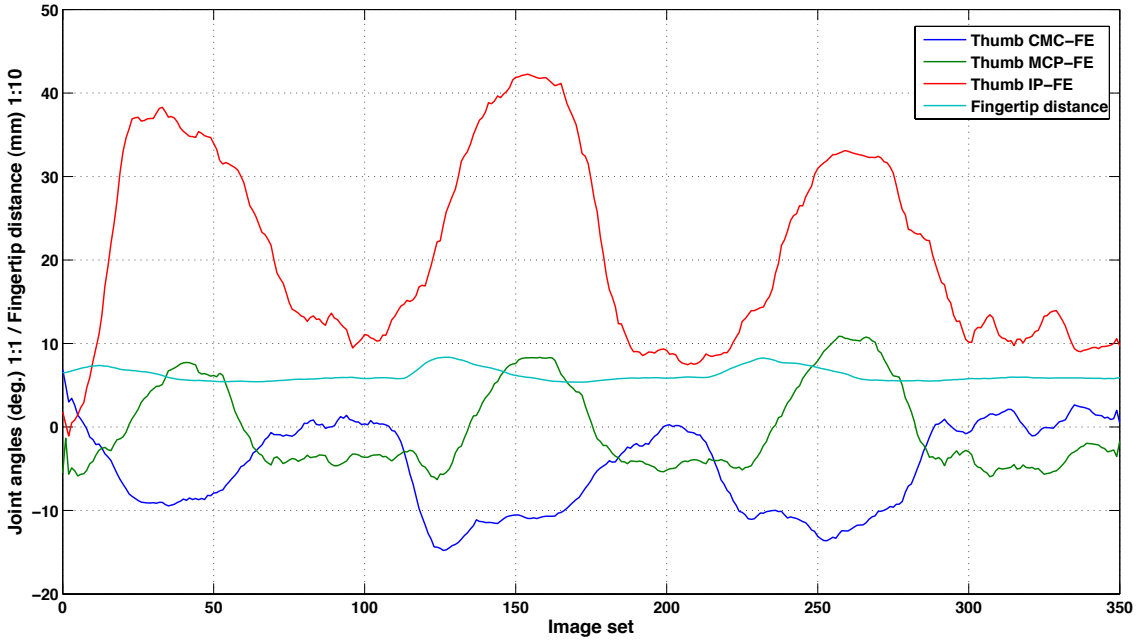


Figure 6.12: Joint angle trajectories of joints of the thumb regarding the DOF associated with the flexion/extension in combination with the fingertip center distance. For better visibility only three out of seven motion repetitions are shown.

trajectories are unstable and experience a lot of jitter. This is especially apparent regarding the distal interphalangeal joint. Similarly to the joint angle trajectories of the thumb, no acceptable repeatability could be determined.

Mean values of the samples of all variables taken in the middle of the contact phase between the fingertips and the object are shown in tables 6.16 and 6.17. Similarly to the first experiment, all seven repetitions of the motion were sampled.

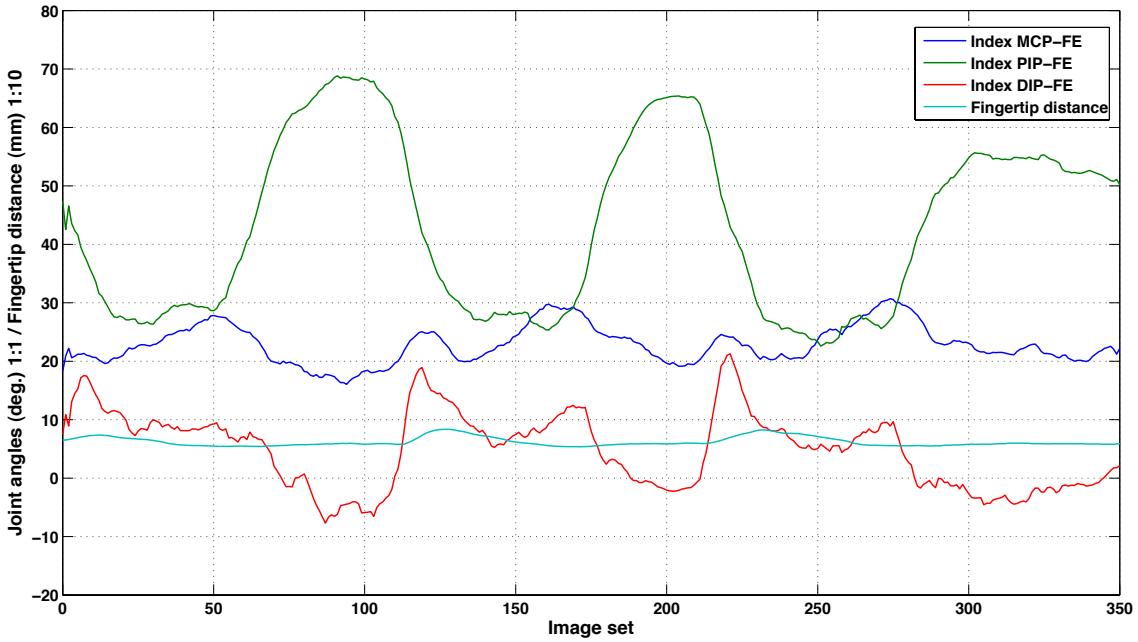


Figure 6.13: Joint angle trajectories of joints of the index finger regarding the DOF associated with the flexion/extension in combination with the fingertip center distance. For better visibility only three out of seven motion repetitions are shown.

Joint	Mean angle (deg.)	Std. deviation (deg.)
Thumb CMC	-4.5492	±3.2285
Thumb MCP	-2.8997	±3.7482
Thumb IP	26.3834	±7.1784
Index MCP	29.2728	±3.6253
Index PIP	29.5401	±9.4766
Index DIP	6.4409	±6.0966

Table 6.16: Repeatability regarding the joint angles during the contact phases with the manipulated object.

	Mean distance (mm)	Std. deviation (mm)
Fingertip center distance	54.5640	±0.8570

Table 6.17: Repeatability regarding the fingertip center distance during the contact phases with the manipulated object.

The determined standard deviation upwards of 3° for each of the joints, does not show any acceptable repeatability. The most likely explanation for this behavior can be given by the motion itself. Although repositioning of the fingertips can be done with ease, execution of each rotating motion with similar forces applied to each joint seems a nontrivial task. A good example can be given by the comparison of the first rotation to the other two, shown in figure 6.13. During the first object rotation the DIP joint experiences a more or less constant force application, while its trajectory shows spikes around the image set numbers 160 and 290.

Nevertheless the measured fingertip center distance shows an acceptable accuracy. Given the distance of 40mm introduced by the object, the remaining distance is 14.564mm, which differs from the result obtained in the first experiment by only 2.379mm. A standard deviation of less than a millimeter expresses good repeatability.

Similarly to the first experiment the DIP/PIP constraint (see equation 6.1) was used for further evaluation of the accuracy. The obtained result (see table 6.18) does not fulfill the constraint, due to most of the motion associated with the rotation of the object having been executed by rotation of the PIP joint.

	Mean ratio	Std. deviation
DIP/PIP ratio	0.2031	±0.1832

Table 6.18: Evaluation of the DIP/PIP constraint over the full sequence of 750 image sets of the recorded object manipulation.

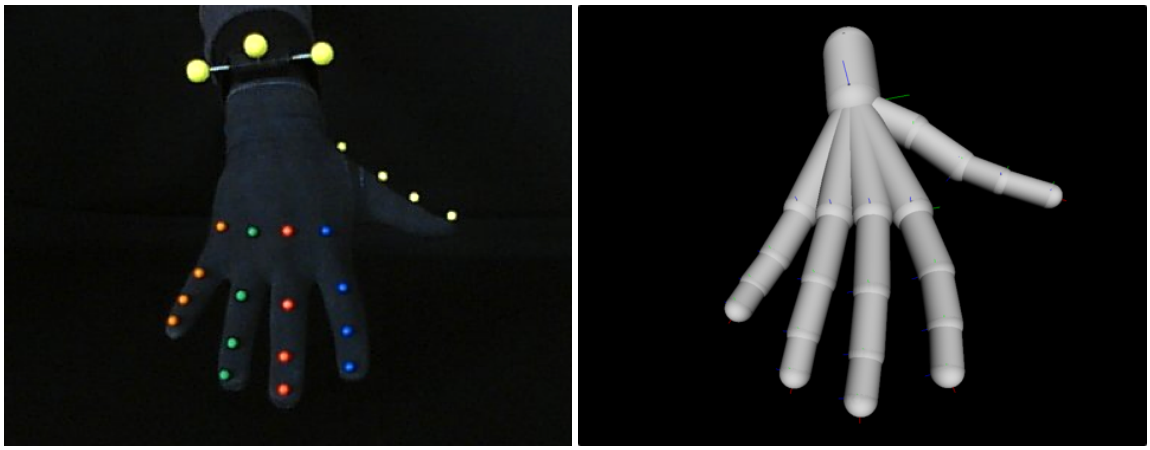
As expected, the determined deviations of measured bone segment lengths compared to the lengths obtained from the subject's hand, are relatively close to those obtained during the first experiment. The bone segment lengths of the little finger represent an exception. Due to the availability of much more samples of a fully reconstructed kinematic chain of the little finger, the mean length deviation as well as the associated standard deviation show a noticeable difference with reference to the results obtained in during the first experiment.

Segment	Thumb	Index finger	Middle finger	Ring finger	Little finger
1 (mm)	4.8215 ±1.0359	1.1989 ±0.7484	2.2797 ±0.7916	0.6426 ±0.4811	2.0245 ±0.8688
2 (mm)	9.5189 ±0.7080	5.8699 ±0.6921	3.3512 ±0.6385	3.7981 ±0.4571	3.4774 ±0.6984
3 (mm)	10.3741 ±0.7308	1.5557 ±0.4361	0.6694 ±0.3526	0.8892 ±0.3063	3.3365 ±0.7472
4 (mm)	2.1068 ±0.5275	0.4626 ±0.3743	2.7216 ±0.2466	2.9299 ±0.2544	4.1355 ±0.6094

Table 6.19: Mean deviations of bone segment lengths compared to the lengths obtained from the subject's hand. Shown are the mean length deviation and the associated standard deviation. Obtained over the full sequence of the recorded object manipulation.

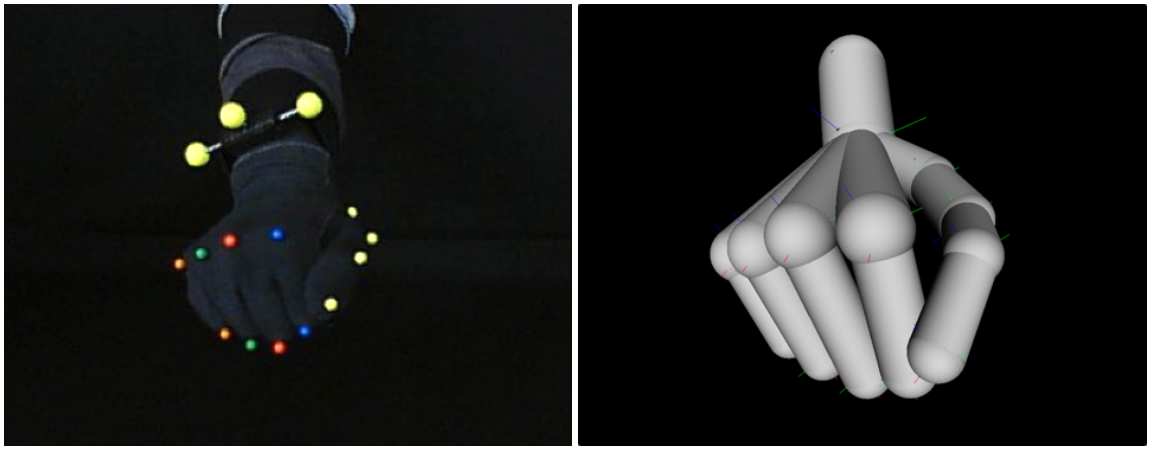
6.3.3 Further reconstruction examples

Another two examples of hand pose reconstruction were produced for the purpose of visual evaluation. Figure 6.14 shows the a completely reconstructed pose of the hand with the fingers spread. Figure 6.15 shows a hand formed into a fist and the corresponding degraded reconstruction result due to self-occlusion.



a **b**

Figure 6.14: Reconstructed hand pose of a hand with the fingers spread. Figure **a)** shows the image recorded by the middle camera of the experimental setup, while figure **b)** shows the result of the hand pose reconstruction. Note the slight curvature of the index finger from the PIP joint to the fingertip, which is due to a malformation of the subject's hand.



a **b**

Figure 6.15: Reconstructed hand pose of a hand formed into a fist. Due to self-occlusion the middle and distal phalanges of the index, middle, ring and little fingers could not be reconstructed.

6.4 Conclusion

This chapter presented and discussed the results that were obtained using the experimental system. The evaluation of the accuracy of the calibration of the stereo vision setup was presented, based on an object with a known length. The evaluation produced acceptable results with a relatively small deviation of the measured length.

The accuracy of the reconstructed hand model was evaluated based on repeatability and fingertip distances as well as the deviations of the measured bone segment lengths compared to the lengths obtained from the subject's hand. Two hand motions were recorded, tip-to-tip grasping (thumb and index finger) and object manipulation (rotation of a small object).

While the tip-to-tip grasping motion produced acceptable results across-the-board attesting good accuracy of the model, the rotation of the object did not yield acceptable repeatability results regarding the joint configurations, albeit it provided good repeatability regarding the fingertip distance. In conclusion the model can be said to provide results with an acceptable accuracy for grasping motions, while only the end-effector (fingertip) distances possess a relatively good accuracy regarding motions related to object manipulation.

Conclusion & further work

7

The approach presented in this thesis provides a framework suitable for the reconstruction of a hand model configurations based on visual information from three camera views.

Special attention was given to the aspect of the presented three-camera setup and the glove being low-cost and using widely available off-the-shelf components. The PlayStation® Eye cameras used in the experimental setup satisfied the low-cost criterium, while providing a high frame rate combined with very good image quality at low noise levels. A black cotton glove, that was equipped with differently colored spherical marker objects, was used to approximately identify the location of the hand joints.

In order to make use of the high frame rate offered by the cameras, a multi-threaded image capture approach, directly using the Video4Linux2 API, was implemented within the software application. Due to this, the solution is able to acquire images from all three sources at 60 frames per second, while storing the images using the JPEG format. Multiple image capture experiments showed, that the images were acquired almost synchronously, with one of the three cameras sometimes being 2-6 ms late. This has been verified visually, as presented in the previous chapter, as well as through evaluation of the timestamp data.

The image processing workflow was implemented using various methods provided by the OpenCV framework. A semi-automated multiple thresholding method has been implemented, using color sample plater for each of the marker object colors, in order to determine initial thresholds for the HSV color model based on histogram evaluation. The determined thresholds were used to obtain a binary image for every color used. Moments were used as shape descriptors, in order to obtain the centroid of each shape, yielding a two-dimensional representation of the projected marker objects. The eccentricity of each shape was determined in order to decide, whether the shape could be accepted as a representation for a spherical marker object.

Calibration of the three-camera experimental setup was obtained using the well-known Matlab® Camera Calibration Toolbox. The three cameras were calibrated as two stereo pairs, with the middle camera providing the frame of reference. Using the OpenCV framework rectification of the images was done prior to further processing

of the images. The results yielded by the image processing stage were used as two-dimensional point sets, in order to determine correspondence using the epipolar constraint. Three-dimensional point coordinates were determined by triangulation. In order to provide a single set of three-dimensional points in the reference frame, the points reconstructed by both pairs were merged based on bounded distances along all three coordinate axes. Merged points were created as the average of both source points.

The obtained three-dimensional points, representing the approximated positions of hand joints, were sorted according to their order in the articulated structure, based on comparison of distances, which was possible due to kinematic constraints imposed throughout the reconstruction. Anthopometric constraints, obtained from a reconstruction of a complete model, were used to ensure the correct marker-joint assignment in the case of an incomplete set of points, resulting in a degraded model. The centers of rotation were approximated using joint thickness measurements specific to the subject's hand. Starting with the wrist joint, the kinematic chain of the reconstructed hand model was determined, describing each degree of freedom using Denavit-Hartenberg parameter sets.

The approach presented in this thesis represents a compromise solution, since the accuracy and usability of the results directly depend on the quality of the setup components as well as the accuracy of the camera calibration and the image processing workflow, to a high degree.

Nevertheless the results produced by the implemented software application, represent a usable approximation of the subject's hand, that can be used for motion evaluation, evaluation of simple grasps and for the purpose of human-computer interaction.

The generated description, representing the hand as a kinematic chain using Denavit-Hartenberg parameter sets, could also be used as a basis for a mapping onto an anthropomorphic artificial hand, such as the ShadowHand C5 used within the HANDLE project.

7.1 Ideas for further work

The produced solution offers room for improvement in all areas. This chapter will discuss possible ideas for further work on different parts of the framework, such as:

- Expansion of the setup to n-views, in order to reduce the amount of marker object occlusion
- Automation of the camera calibration process, in order to gain flexibility of the camera setup
- Distribution of the computational load, in order to achieve real-time reconstruction of the model

- Development of a more robust thresholding procedure with an improved determination of thresholds for the color distribution of the spherical markers
- Using an inverse kinematics model and/or a Kalman filter, in order to enable the completion of a degraded model
- Development of a real-time remote operation solution using an anthropomorphic artificial hand

Expansion to n-views

Due to the high amount of degrees of freedom of the hand, the amount of occlusion is very high with almost any but the most basic configuration of the articulated structure. Even though the experimental setup with three views used in this thesis, offers the possibility to obtain a full configuration of the hand model, based on a constellation of the hand approaching a spherical curvature - something, that is impossible with only two views - the presented approach would greatly benefit from a higher number of additional views. The approach described in this thesis does not require a fixed number of views, therefore the experimental setup could be extended relatively easy. The only important factors to consider would be the increase in effort necessary for the calibration of the setup, as well as the increase in computational load due to processing of additional image data.

Automation of the calibration process

The process of camera calibration is very time-intensive. Therefore it is most desirable to automate the calibration process as far as possible, especially when looking at additional views. While the calibration can be done using the same method, that was used in this thesis (see section 3.2.4), the calibration of a true n-view setup would be far too time-intensive. One possible approach to this problem was proposed by Svoboda et al. in [SMP05]. The proposed approach is a self-calibration method for multiple camera setups, that requires a bright spot as a calibration object, that needs to be waved through the working volume in order to generate point sets for calibration. Less time needed for calibration of a true n-view setup would provide more flexibility as the setup would not necessarily need to be fixed.

Distribution of computational load

There is no need for the data to be processed by a central unit, up to the point of the search for corresponding point pairs within the point sets, that were extracted from the binary images. Therefore it is possible to distribute the views among multiple units, in order to increase performance.

Multiple experiments at the beginning of this thesis have shown, that an off-the-shelf computer system used for the experiments (see section 5.1) was able to reliably handle at most three cameras operated at 60 frames per second, while also displaying and writing the images to the hard disk in JPEG format. Adding the image processing workflow, drastically affected the performance, by dropping the frame rate to about 10 frames per second. Therefore in order to use additional views, the computational load would need to be distributed to some extent.

The distribution of the computational load could for example be handled by in-camera preprocessing of the images. It would also be possible to offload the image acquisition and processing to multiple workstations or dedicated embedded hardware units. Although this step would increase the performance and allow to process the images in real-time for a given frame rate, it would also amplify the problem of synchronization of the cameras. This problem could be solved by constant synchronization of the units to a single time source (e.g. via the network time protocol) and processing of the timestamp data.

Development of a robust segmentation

Although the implemented method for determination of thresholds based on the color calibration samples delivers usable results, it shows particularly weak performance in the case of shadows cast on the marker objects due to occlusion of the illumination source. The shadows lead to degradation of the segmented shape and its discarding due to its eccentricity attribute.

Better results should be possible using the implemented method in combination with region growing. A selection of seed points, that are needed in order to grow a region, would be provided by the implemented method. Since the shape of the spherical marker objects is invariant to rotation, it would be possible to combine the determined hue thresholds with the eccentricity attribute, in order to establish the region membership criterion. This would allow to identify pixels belonging to a marker object disregarding the shadows, while growing the shape to its maximum circular form.

A more robust segmentation would also allow a better separation of the colors, which consequently would allow to use more colors. The additional colors would enable tracking of colored objects in addition to the hand itself, which would highly improve the analysis and visualization of object manipulation.

Completion of a degraded model

The approach presented in this thesis provides a reconstruction of the hand pose using information of only a single set of images. Consequentially the reconstructed model is incomplete, if not all marker object positions could be determined. If the

joint chain of a finger is incomplete (e.g. missing a distal interphalangeal joint), the finger is reconstructed up to the point of the missing joint.

Working with only a single set of images, the problem of an incomplete model can not be averted, if a full finger is missing due to occlusion. The behavior in the case of a missing joint however, can be improved by the use of an inverse kinematics model of the hand. The constraints imposed by consideration of the fingers as planar rigid systems, would allow to determine the position of the missing joint and thus to complete the model.

Another possibility for improvement would be the use of a Kalman filter in combination with information accumulated over a larger series of image sets. This would allow to provide an estimation for the position of joints that were lost for a short period of time.

Real-time remote operation solution

The kinematic chain of the reconstructed hand model has a close resemblance to the kinematic chain of the ShadowHand C5, the anthropomorphic artificial hand designed by ShadowRobot. Therefore it would be possible to map the parameter sets of the reconstructed model onto the articulated structure of the artificial hand.

The mapping procedure could either be used to replay recorded hand configuration sequences with the artificial hand, or - given an improved solution able to provide a reconstruction in real-time - to provide a remote operation capability.

Bibliography

[AAK71] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Proceedings of the Symposium on Close-Range Photogrammetry*, pages 1–18, 1971.

[AC97] J. K. Aggarwal and Q. Cai. Human motion analysis: a review. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 90–102, 1997.

[ACCL79] K. N. An, E. Y. Chao, W. P. Cooney, and R. L. Linscheid. Normative model of human hand for biomechanical analysis. *Journal of Biomechanics*, 12(10):775–788, 1979.

[AL91] N. Ayache and F. Lustman. Trinocular Stereo Vision for Robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:73–85, 1991.

[AY79] J. G. Andrews and Y. Youm. A biomechanical investigation of wrist kinematics. *Journal of Biomechanics*, 12(1):83–93, 1979.

[BA92] B. Buchholz and T. J. Armstrong. A kinematic model of the human hand to evaluate its prehensile capabilities. *Journal of Biomechanics*, 25(2):149–162, 1992.

[BAG92] B. Buchholz, T. J. Armstrong, and S. A. Goldstein. Anthropometric data for describing the kinematics of the human hand. *Ergonomics*, 35(3):261–273, 1992.

[BB08] W. Burger and M. J. Burge. *Digital Image Processing: An Algorithmic Introduction Using Java*. Springer, 1st edition, 2008.

[BK08] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

[Bou10] J.-Y. Bouguet. Matlab® Camera Calibration Toolbox, 2010. http://www.vision.caltech.edu/bouguetj/calib_doc/.

[Bra03] J. Bray. Markerless Based Human Motion Capture: A Survey. Lab report, 2003.

[Bro71] D. C. Brown. Close Range Camera Calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[BT98] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *Computer Vision, 1998. Sixth International Conference on*, pages 1073–1080, 1998.

[CDML+07] P. Cerveri, E. De Momi, N. Lopomo, G. Baud-Bovy, R. Barros, and G. Ferrigno. Finger Kinematic Modeling and Real-Time Hand Motion Estimation. *Annals of Biomedical Engineering*, 35:1989–2002, 2007.

[CFAT07] W. Chen, R. Fujiki, D. Arita, and R. Taniguchi. Real-time 3D Hand Shape Estimation based on Image Feature Analysis and Inverse Kinematics. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 247–252, 2007.

[CFSU+08] S. Cobos, M. Ferre, M. A. Sanchez Uran, J. Ortego, and C. Pena. Efficient human hand kinematics for manipulation tasks. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2246–2251, 2008.

[CLCL81] W. P. Cooney, M. J. Lucca, E. Y. S. Chao, and R. L. Linscheid. The kinesiology of the thumb trapeziometacarpal joint. *The Journal of Bone and Joint Surgery*, 63(9):1371–1381, 1981.

[CM06] L. Y. Chang and Y. Matsuoka. A kinematic thumb model for the ACT hand. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1000–1005, 2006.

[Dan82] P. E. Danielsson. An Improved Segmentation and Coding Algorithm for Binary and Nonbinary Images. *IBM Journal of Research and Development*, 26(6):698–707, 1982.

[DD95] D. F. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

[DH55] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Transactions of the ASME. Journal of Applied Mechanics*, 22:215–221, 1955.

[EBN+07] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108:52–73, 2007.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[FTV00] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000.

[GW07] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2007.

[HA89] T. D. Haig and Y. Attikiouzel. An improved algorithm for border following of binary images. In *Circuit Theory and Design, 1989., European Conference on*, pages 118–122, 1989.

[HAN09] HANDLE. A Protocol for the corpus of sensed grasp and handling data. Deliverable 4, 2009.

[HAN11] HANDLE. Developmental pathway towards autonomy and dexterity in robot in-hand manipulation, August 2011. http://www.handle-project.eu/.

[Har97] R.I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997.

[Har99] Richard I. Hartley. Theory and Practice of Projective Rectification. *International Journal of Computer Vision*, 35:115–127, 1999.

[HBM+92] A. Hollister, W. L. Buford, L. M. Myers, D. J. Giurintano, and A. Novick. The axes of rotation of the thumb carpometacarpal joint. *Journal of Orthopaedic Research*, 10(3):454–460, 1992.

[HK94] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(3):267–276, 1994.

[HS97] R. I. Hartley and P. Sturm. Triangulation. In *Computer Vision and Image Understanding*, volume 68, pages 146–157, 1997.

[Hu62] M. K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[ITU11] ITU. Recommendation BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. Technical report, International Telecommunication Union, Radiocommunication Sector, 2011.

[Jac07] K. Jack. *Video Demystified: A Handbook for the Digital Engineer*. Newnes, 5th edition, 2007.

[JB91] X. Y. Jiang and H. Bunke. Simple and fast computation of moments. *Pattern Recognition*, 24(8):801–806, 1991.

[KH94] J. J. Kuch and T. S. Huang. Human computer interaction via the human hand: a hand model. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, volume 2, pages 1252–1256, 1994.

[KHEK76] M. Kuwahara, K. Hachimura, S. Ehiu, and M. Kinoshita. Processing of Ri-Angiocardiographic Images. In *Digital Processing of Biomedical Images*, pages 187–203, 1976.

[Khr04] Khronos Group. OpenGL® specification (version 2.0), 2004. http://www.opengl.org/.

[Kon97] K. Konolige. Small vision systems: hardware and implementation. In *Eighth International Symposium on Robotics Research*, pages 111–116, 1997.

[KSN08] K. Kanatani, Y. Sugaya, and H. Niitsuma. Triangulation from Two Views Revisited: Hartley-Sturm vs. Optimal Correction. In *19th British Machine Vision Conference*, volume 4, pages 173–182, 2008.

[KZD01] K. Kwon, H. Zhang, and F. Dornaika. Hand pose recovery with a single video camera. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1194–1200, 2001.

[Leu91] J. G. Leu. Computing a shape's moments from its boundary. *Pattern Recognition*, 24(10):949–957, 1991.

[LF96] Q. T. Luong and O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17:43–75, 1996.

[LFSK06] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise Estimation from a Single Image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 901–908, 2006.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[LH00] F. Lathuiliere and J.-Y. Herve. Visual tracking of hand posture with occlusion handling. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 1129–1133, 2000.

[Lin10] P. Lindstrom. Triangulation made easy. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1554–1561, 2010.

[Lit73] J. W. Littler. On The Adaptability of Man's Hand (With Reference to the Equiangular Curve). *The Hand*, 5(3):187–191, 1973.

[LK95] J. Lee and T. L. Kunii. Model-based analysis of hand posture. *Computer Graphics and Applications, IEEE*, 15(5):77–86, 1995.

[LWH00] J. Lin, Y. Wu, and T. S. Huang. Modeling the constraints of human hand motion. In *Human Motion, 2000. Proceedings. Workshop on*, pages 121–126, 2000.

[MG01] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.

[MHC04] H. S. Malvar, L. W. He, and R. Cutler. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, page iii, 2004.

[Moo68] G. A. Moore. Automatic scanning and computer process for the quantitative analysis of micrographs and equivalent subjects. In *Pictorial Pattern Recognition*, pages 275–326, 1968.

[Mor78] J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer Berlin / Heidelberg, 1978.

[MW11] B. Majors and J. Wayne. Development and Validation of a Computational Model for Investigation of Wrist Biomechanics. *Annals of Biomedical Engineering*, pages 1–9, 2011.

[Nok11] Nokia Corporation. Qt software development kit (version 1.1.1), 2011. http://qt.nokia.com/.

[RG91] H. Rijpkema and M. Girard. Computer animation of knowledge-based human grasping. *SIGGRAPH Computer Graphics*, 25:339–348, 1991.

[RK94] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *Proceedings of the third European conference on Computer Vision (Vol. II)*, pages 35–46, 1994.

[RKK09] J. Romero, H. Kjellstrom, and D. Kragic. Monocular real-time 3D articulated hand pose estimation. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 87–92, 2009.

[SA85] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision Graphics and Image Processing*, 30(1):32–46, 1985.

[Sch05] O. Schreer. *Stereoanalyse und Bildsynthese*. Springer Berlin / Heidelberg, 2005.

[Sha08] Shadow Robot Company Ltd. Shadow Dexterous Hand C5: Technical Specification, 2008. http://www.shadowrobot.com/hand/.

[SHV05] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley and Sons, 1st edition, 2005.

[SKH+98] W. P. Smutz, A. Kongsayreepong, R. E. Hughes, G. Niebur, W. P. Cooney, and K. N. An. Mechanical advantage of the thumb muscles. *Journal of Biomechanics*, 31(6):565–570, 1998.

[SMC01] B. Stenger, P.R.S. Mendonca, and R. Cipolla. Model-based 3D tracking of an articulated hand. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages 310–315, 2001.

[Smi78] A. R. Smith. Color gamut transform pairs. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '78, pages 12–19, 1978.

[SMP05] T. Svoboda, D. Martinec, and T. Pajdla. A Convenient Multi-Camera Self-Calibration for Virtual Environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, 2005.

[TC88] C. H. Teh and R. T. Chin. On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):496–513, 1988.

[TKBM99] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and Practice of Background Maintenance. In *Seventh International Conference on Computer Vision (ICCV 1999)*, pages 255–261, 1999.

[TP11] N. Thayer and S. Priya. Design and implementation of a dexterous anthropomorphic robotic typing (DART) hand. *Smart Materials and Structures*, 20(3):10–22, 2011.

[Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[UMIO01] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara. Hand pose estimation using multi-viewpoint silhouette images. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 1989–1996, 2001.

[VB06] M. Veber and T. Bajd. Assessment of human hand kinematics. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2966–2971, 2006.

[VLB80] R. G. Volz, M. Lieb, and J. Benjamin. Biomechanics of the wrist. *Clinical Orthopaedics and Related Research*, 80(149):112–117, 1980.

[Wil10] Willow Garage Inc. OpenCV framework (version 2.2), 2010. http://opencv.willowgarage.com/wiki/.

[WP09] R. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28:1–8, 2009.

[YGFS78] Y. Youm, T. E. Gillespie, A. E. Flatt, and B. L. Sprague. Kinematic investigation of normal MCP joint. *Journal of Biomechanics*, 11(3):109–118, 1978.

[YMFG78] Y. Youm, R. Y. McMurthy, A. E. Flatt, and T. E. Gillespie. Kinematics of the wrist. I. An experimental study of radial-ulnar deviation and flexion-extension. *The Journal of Bone and Joint Surgery*, 60(4):423–431, 1978.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

Appendix

A.1 XML configuration data

The following listings show extracts of the configuration files used by the visualization module of the software application. The full versions of the files are available on the accompanying CD.

Color thresholds

Listing 1 shows an extract of the XML configuration file storing the color thresholds, which were determined using the threshold calibration module of the software application. The listing shows the upper and lower thresholds determined for the colors blue (index finger) and red (middle finger).

```
<?xml version="1.0" encoding="UTF-8"?>
<color_thresholds>
        .
        .
        .
        <color name="blue">
                <thresholds count="2">
                        <threshold_1 type="lower">
                                <hue>107.0</hue>
                                <saturation>200.0</saturation>
                                <value>50.0</value>
                        </threshold_1>
                        <threshold_2 type="upper">
                                <hue>118.0</hue>
                                <saturation>256.0</saturation>
                                <value>256.0</value>
                        </threshold_2>
                </thresholds>
        </color>
        <color name="red">
                <thresholds count="4">
                        <threshold_1 type="lower">
                                <hue>173.0</hue>
                                <saturation>118.0</saturation>
                                <value>85.0</value>
                        </threshold_1>
                        <threshold_2 type="upper">
                                <hue>180.0</hue>
                                <saturation>256.0</saturation>
                                <value>256.0</value>
                        </threshold_2>
```

```
                        <threshold_3 type="lower">
                                <hue>0.0</hue>
                                <saturation>118.0</saturation>
                                <value>85.0</value>
                        </threshold_3>
                        <threshold_4 type="upper">
                                <hue>8.0</hue>
                                <saturation>256.0</saturation>
                                <value>256.0</value>
                        </threshold_4>
                </thresholds>
        </color>
        .
        .
        .
</color_thresholds>
```

Listing 1: Extract of the color_thresholds.xml configuration file. Shown are the determined thresholds for the colors blue (index finger) and red (middle finger).

Hand dimensions

Listing 2 shows an extract of the XML configuration file, which stores the hand dimensions (bone segment lengths, wrist joint offset, joint diameters) that were obtained from the subject's hand used in the experiments. The listing shows the bone segment lengths of the index finger, the wrist joint offset and diameter as well as the diameters of the joints of the thumb and the index finger.

```
<?xml version="1.0" encoding="UTF-8"?>
<hand_dimensions>
        <bone_dimensions>
                .
                .
                <index>
                        <bone name="metacarpal">
                                <length>95.0</length>
                        </bone>
                        <bone name="proximal">
                                <length>49.0</length>
                        </bone>
                        <bone name="middle">
                                <length>27.0</length>
                        </bone>
                        <bone name="distal">
                                <length>22.0</length>
                        </bone>
                </index>
                .
                .
        </bone_dimensions>
        <joint_dimensions>
                <wrist>
                        <offset>30.0</offset>
                        <joint name="ru">
                                <diameter>46.0</diameter>
                        </joint>
                </wrist>
```

```
                <thumb>
                        <joint name="cmc">
                                <diameter>25.0</diameter>
                        </joint>
                        <joint name="mcp">
                                <diameter>23.5</diameter>
                        </joint>
                        <joint name="ip">
                                <diameter>16.0</diameter>
                        </joint>
                        <fingertip>
                                <diameter>9.5</diameter>
                        </fingertip>
                </thumb>
                <index>
                        <joint name="mcp">
                                <diameter>25.0</diameter>
                        </joint>
                        <joint name="pip">
                                <diameter>16.0</diameter>
                        </joint>
                        <joint name="dip">
                                <diameter>12.0</diameter>
                        </joint>
                        <fingertip>
                                <diameter>9.0</diameter>
                        </fingertip>
                </index>
                .
                .
        </joint_dimensions>
</hand_dimensions>
```

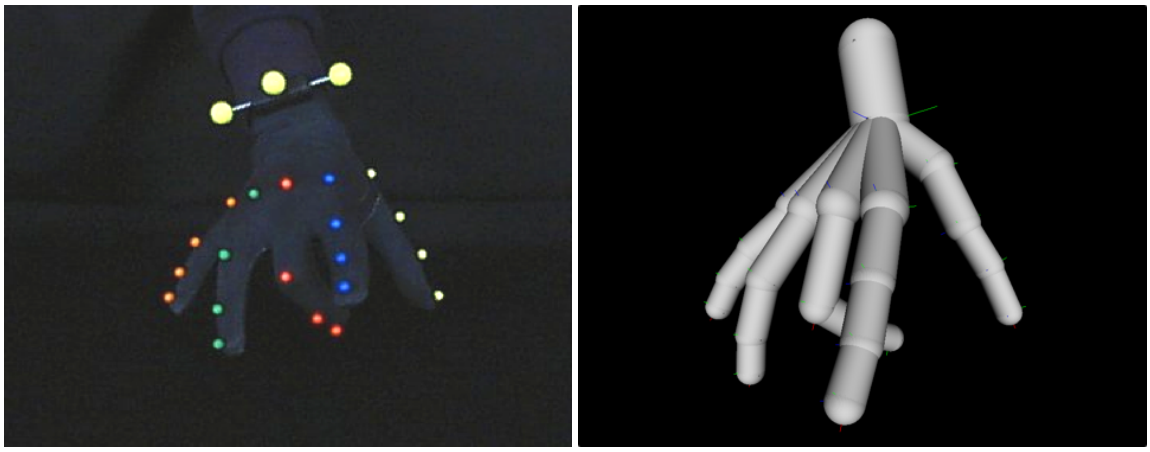
Listing 2: Extract of the hand_dimensions.xml configuration file. Shown are the bone segment lengths of the index finger, the wrist joint offset and diameter as well as the diameters of the joints of the thumb and the index finger.

A.2 Examples of obtained model descriptions

The following listings show extracts of the XML file structure used to store the obtained Denavit-Hartenberg description of a reconstructed hand pose.

The accompanying CD contains the sequences of XML files with DH descriptions obtained from four recorded sessions, located inside the *output/* folder of each session. The raw session data including the configuration and calibration files are also included.

Fully reconstructed hand pose



a **b**

Figure A.1: Example of a fully reconstructed hand pose, flexing the middle finger. Figure **a)** shows the image acquired by the middle camera of the three-camera setup, with the visualization shown in figure **b)**.

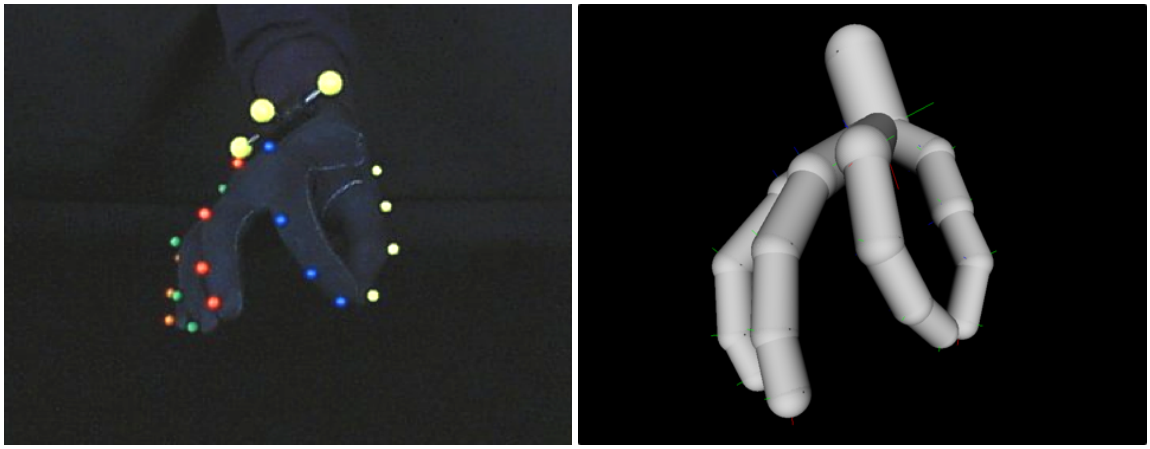
Listing 3 shows an extract of the obtained DH description for a fully reconstructed hand pose, as shown in figure A.1 b). The completeness of the model is specified using the <*model_ complete*> XML tag. 1 specifies completeness, whereas 0 signifies a degraded model. The extract includes the base frame, the first DOF of the wrist joint and the full kinematic chain of the index finger.

```
<?xml version="1.0" encoding="UTF-8"?>
<hand_configuration>
        <model_complete>1</model_complete>
        <scale_factor>0.100000</scale_factor>
        <base_frame>
                <origin>
                        <x>0.756534</x>
                        <y>21.9871</y>
                        <z>-11.9562</z>
                </origin>
                <x_axis>
                        <x>0.238565</x>
                        <y>-0.939894</y>
                        <z>0.244308</z>
                </x_axis>
                <y_axis>
                        <x>0.947434</x>
                        <y>0.28049</y>
                        <z>0.153927</z>
                </y_axis>
                <z_axis>
                        <x>-0.213201</x>
                        <y>0.194744</y>
                        <z>0.957403</z>
                </z_axis>
        </base_frame>
        <dh_parameters>
                <wrist>
                        <frame_1>
                                <theta>-4.458602</theta>
                                <d>0.000000</d>
                                <a>0.000000</a>
                                <alpha>90.000000</alpha>
                        </frame_1>
                </wrist>
                .
                .
                .
                <finger_2 name="index">
                        <frame_2>
                                <theta>48.432644</theta>
                                <d>-2.673507</d>
                                <a>9.305589</a>
                                <alpha>-90.000000</alpha>
                        </frame_2>
                        <frame_3>
                                <theta>4.389958</theta>
                                <d>0.000000</d>
                                <a>0.000000</a>
                                <alpha>90.000000</alpha>
                        </frame_3>
                        <frame_4>
                                <theta>-6.386497</theta>
                                <d>0.000000</d>
                                <a>4.309835</a>
                                <alpha>0.000000</alpha>
                        </frame_4>
                        <frame_5>
                                <theta>-5.960621</theta>
                                <d>0.184687</d>
                                <a>2.837374</a>
                                <alpha>0.000000</alpha>
                        </frame_5>
```

```
                        <frame_6>
                                <theta>3.700457</theta>
                                <d>0.338151</d>
                                <a>2.308040</a>
                                <alpha>0.000000</alpha>
                        </frame_6>
                </finger_2>
                .
                .
                .
        </dh_parameters>
</hand_configuration>
```

Listing 3: Extract of a complete DH description obtained of the hand pose shown in figure A.1. The extract shows the base frame, the first DOF of the wrist as well as the full kinematic chain of the index finger.

Degraded hand pose



a **b**

Figure A.2: Example of a degraded reconstructed hand pose, touching the fingertips of the thumb and the index finger. Due to a broken kinematic chain of the little finger caused by occlusion, no reconstruction was performed. Figure **a)** shows the image acquired by the middle camera of the three-camera setup, with the visualization shown in figure **b)**.

Listing 4 shows an extract of the obtained DH description for an incomplete reconstructed hand pose, as shown in figure A.2 b), with the little finger missing due to occlusion. The incompleteness of the model is specified by a 0 inside the *<model_complete>* XML tag. The listing shows the base frame, the first DOF of the wrist, the full kinematic chain of the index finger and the empty XML tag for the missing data of the little finger.

```
<?xmlversion="1.0" encoding="UTF-8"?>
<hand_configuration>
        <model_complete>0</model_complete>
        <scale_factor>0.100000</scale_factor>
        <base_frame>
                <origin>
                        <x>-0.988765</x>
                        <y>20.6166</y>
                        <z>-9.59429</z>
                </origin>
                <x_axis>
                        <x>0.287789</x>
                        <y>-0.905788</y>
                        <z>0.311008</z>
                </x_axis>
                <y_axis>
                        <x>0.738523</x>
                        <y>0.416652</y>
                        <z>0.53008</z>
                </y_axis>
                <z_axis>
                        <x>-0.609722</x>
                        <y>0.077135</y>
                        <z>0.788853</z>
                </z_axis>
        </base_frame>
        <dh_parameters>
                <wrist>
                        <frame_1>
                                <theta>3.522921</theta>
                                <d>0.000000</d>
                                <a>0.000000</a>
                                <alpha>90.000000</alpha>
                        </frame_1>
                </wrist>
                .
                .
                .
                <finger_2 name="index">
                        <frame_2>
                                <theta>68.654360</theta>
                                <d>-2.985919</d>
                                <a>9.160658</a>
                                <alpha>-90.000000</alpha>
                        </frame_2>
                        <frame_3>
                                <theta>-1.700529</theta>
                                <d>0.000000</d>
                                <a>0.000000</a>
                                <alpha>90.000000</alpha>
                        </frame_3>
                        <frame_4>
                                <theta>-49.628241</theta>
                                <d>0.000000</d>
                                <a>4.237178</a>
                                <alpha>0.000000</alpha>
                        </frame_4>
                        <frame_5>
                                <theta>-38.128397</theta>
                                <d>0.171973</d>
                                <a>2.926587</a>
                                <alpha>0.000000</alpha>
                        </frame_5>
```

```
                        <frame_6>
                                <theta>-22.337419</theta>
                                <d>-0.012442</d>
                                <a>2.255376</a>
                                <alpha>0.000000</alpha>
                        </frame_6>
                </finger_2>
                .
                .
                .
                <finger_5 name="little"/>
        </dh_parameters>
</hand_configuration>
```

Listing 4: Incomplete DH description obtained of the hand pose shown in A.2. The incompleteness of the model is declared by the 0 inside the *<model_complete>* tag. Shown are the base frame, the first DOF of the wrist joint, the full kinematic chain of the index finger and the empty tag signifying the missing little finger.

A.3 Recorded hand motion sequences

Four recorded sessions are available on the accompanying CD, containing following hand motion recordings:

- **session_1**: Motion of the hand within the working area of the experimental environment with flexion/extension of the wrist joint.
- **session_2**: Tip-to-tip grasping motion of the thumb and the index finger.
- **session_3**: Flexion/extension of each of the five fingers.
- **session_4**: Object manipulation by rotation of a small octagonal object using the thumb and the index finger.

Acknowledgements

I would like to thank everybody who made this thesis possible. First of all my supervisors Prof. Dr. Jianwei Zhang and Prof. Dr. Leonie Dreschler-Fischer. I would like to thank Prof. Dr. Jianwei Zhang for the opportunity to write this thesis within the TAMS group and the sign of trust by taking the part of primary supervisor. I would like to thank Prof. Dr. Leonie Dreschler-Fischer for secondary supervision and the support with ideas and advice regarding various parts of image processing and stereo vision.

Special thanks go to Dr. Norman Hendrich, for the constant support, advice and constructive feedback throughout this thesis.

I would like to thank Denis Klimentjew for ideas and advice regarding the aspects of stereo vision and providing me with the calibration object, Dr. Andreas Mäder for trusting me with root access to the workstations and Hannes Bistry for helping out with the evaluation of image acquisition performance using the GStreamer framework.

I would like to thank all members of the TAMS group for providing an enjoyable atmosphere to work in.

I would especially like to thank my parents for the moral support throughout this thesis and the unquestioning support during the years of my studies.

Erklärung

Ich, Eugen Richter, Matr.-Nr.: 5526063, versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Departments Informatik einverstanden.

(Ort, Datum) (Unterschrift)