Felix Kühnel, BSc

Robust Head and Eye Tracking in Medical Environments

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Telematik

submitted to

Graz University of Technology

Supervisor

Dr. Matthias Rüther
Institute for Computer Graphics and Vision

Graz, Austria, April 2015

Abstract

In radiation therapy, the treatment of eye tumors is particularly challenging. In order to determine the tumor location in 3D, a magnetic resonance imaging system is used to acquire slice images of the head. Subsequently, the affected region has to be radiated in a linear accelerator. During the whole treatment high precision is crucial, thus in previous solutions head and eyes were rigidly fixed, which was not only very uncomfortable, but also invasive. In a project cooperation of the Graz University of Technology and the Medical University of Graz an eye tracking system has been developed. It enables a treatment without the invasive fixation of head and eyes by using a face mask and two 2D cameras for the estimation of the eye position. The deviation of the current eye position from their initial position is used to generate commands to control the treatment, e.g. pause the radiation therapy device due to a too strong deviation.

In this Master's Thesis we discuss a head tracking system, which measures unwanted head motion of a patient and can be combined with the existing eye tracking system. For this purpose, we find a suited hardware setup using a time-of-flight camera and deal with the topic of robust head pose estimation. We implement three methods with different approaches to gain head pose information and compare them in a statistic evaluation. First, we analyze the ICP algorithm, which is based on the registration of two point clouds. Further, a template matching approach using three templates located at facial feature points is examined. The tracked positions in combination with their depth values allow us to compute the head pose. Finally, we adapt a method by Meers and Ward. It uses spherical intersections of the face to gain topographic information. In this way we are able to derive the face orientation. In a facial expression analysis, we examine the robustness of the three methods. Based on our results, the suitability of the methods can be discussed, which is a step towards the realization of the head and eye tracking system.

Keywords: *computer vision, head and eye tracking system, time-of-flight camera, robust head pose estimation, medical environments, fixation-free eye tumor treatment*

Kurzfassung

In der Strahlentheraphie ist die Behandlung von Augentumoren besonders herausfordernd. Um die Tumorposition in 3D zu bestimmen, wird ein Magnetresonanztomograph zur Aufnahme von Schnittbildern des Kopfes verwendet. Anschließend kann die betroffene Region in einem Linearbeschleuniger bestrahlt werden. Während der gesamten Behandlung ist hohe Genauigkeit ausschlaggebend, weshalb in bisherigen Lösungen Kopf und Augen ruhiggestellt wurden, was nicht nur sehr unangenehm sondern auch invasiv ist. In einem gemeinsamen Projekt der Technischen und Medizinischen Universität Graz wurde ein Augen-Trackingsystem entwickelt. Dieses ermöglicht, mit Hilfe einer Gesichtsmaske und zwei 2D Kameras zur Bestimmung der Augenposition, eine Behandlung ohne invasive Fixierung von Kopf und Augen. Die Abweichung der momentanen Augenposition von der Ursprungsposition kann verwendet werden, um Kommandos zur Steuerung der Behandlung zu erzeugen, z.B. kann das Behandlungsgerät bei einer zu großen Abweichung pausiert werden.

In dieser Masterarbeit stellen wir ein Kopf-Trackingsystem vor, das ungewollte Kopfbewegungen eines Patienten misst und mit dem bestehenden Augen-Trackingsystem kombiniert werden kann. Zu diesem Zweck suchen wir nach einem geeigneten Hardwareaufbau mit einer Time-of-Flight Kamera und besprechen, wie eine robuste Bestimmung der Kopfpose möglich ist. Wir implementieren drei Methoden mit verschiedenen Ansätzen, um Information über die Kopfpose zu gewinnen und vergleichen diese in einer statistischen Auswertung. Zu Beginn analysieren wir den ICP Algorithmus, welcher auf der Registrierung zweier Punktwolken basiert. Mittels Template Matching verfolgen wir in einem weiteren Ansatz die Bewegung dreier Templates, die sich an den Positionen markanter Gesichtsmerkmale befinden. Die getrackten Positionen erlauben es uns, gemeinsam mit ihren Tiefenwerten die Kopfpose zu berechnen. Zuletzt adaptieren wir eine Methode von Meers und Ward, welche Kugeln mit dem Gesicht schneidet, um topographische Informationen zu erhalten. Auf diese Weise kann die Ausrichtung des Gesichts abgeleitet

werden. In einer Gesichtsausdrucksanalyse untersuchen wir die Robustheit der drei Methoden. Unter Verwendung unserer Ergebnisse kann die Eignung der Methoden diskutiert werden, wodurch wir einen Schritt näher an die Verwirklichung des Kopf- und Augen-Trackingsystems rücken.

Schlagwörter: *digitale Bildverarbeitung, Kopf- und Augen-Trackingsystem, Time-of-Flight Kamera, robuste Bestimmung der Kopfpose, medizinische Umgebungen, befestigungsfreie Augentumorbehandlung*

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

The text document uploaded to TUGRAZonline is identical to the presented master's thesis dissertation.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Ort	Datum	Unterschrift

Acknowledgements

First of all I want to thank Matthias Rüther for guiding me through this Master's Thesis. He gave me the great possibility to work in many projects during my studies and kindly integrated me into his Robot Vision Group. Further, I want to thank all group members, Christian, Gernot and especially David for their valuable advice. At this point I want to mention Roman, thank you for many refreshing conversations during work. Finally, I would like to thank Inés for her never-ending energy and support during the years we shared and the time we spent outdoors walking the soles off our shoes. Last but not least, I want to thank my parents Irmtraud and Berndt for giving me the possibility to follow my interests and helping me to keep in mind what is essential in life.

Contents

List of Figures

List of Tables

1 Introduction and motivation

Contents

In radiation therapy, the treatment of eye tumors is particularly challenging and high accuracy is crucial for success. To avoid movement during the treatment, an invasive technique is commonly used to rigidly fix head and eyes to the treatment couch. In this way, only the true 3D position of the eye tumor is radiated.

In this Master's Thesis, we develop a head and eye tracking system which offers a non-invasive solution to this problem. It estimates the head pose and eye gaze direction of the patient, which can be used to generate triggering commands to control the treatment. The patient is advised to remain in a straight head position and look at a certain point, which will be remembered as the initial position. In case of deviations, the treatment devices can simply pause and continue when the initial position is reached again.

The tracking of head and eyes is generally not an easy task, since it is composed of two independent movements. The head pose is described by three rotational and three translational Degrees of Freedom (DOF). The eyes move relative to the head but their lines of vision are not parallel. We can describe each eye by three rotational *DOF*. In this work we utilize the advantage, that the patient must remain in the initial position. Thus, we can separate head and eye tracking, and perform head pose estimation first (six *DOF*). The eyes are only tracked when the head is currently in its initial position. As a consequence, 2D eye tracking by only observing the pupils' locations becomes possible.

1.1 Field of application

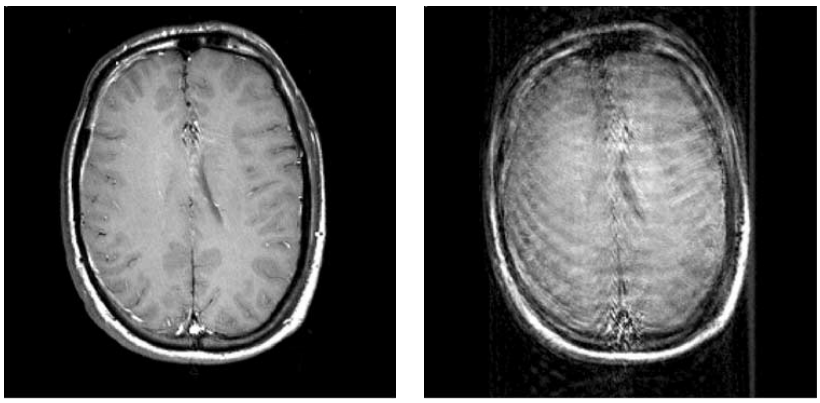
The head and eye tracking system is targeted to simplify the treatment of eye tumors in radiation therapy. First, a Magnetic Resonance Imaging (MRI) system (Figure 1.1a) acquires slice images of the head, which are then used to locate the eye tumor in 3D. In the subsequent radiation therapy the tumor cells are intended to be destroyed. The radiation therapy device (Figure 1.1b) rotates around the head and sends radiation through the tumor from many directions. In this way, surrounding healthy tissues are less exposed to radiation, while the dose in the tumor is maximized. For the success of this treatment millimeter precision is required.

(a) MRI system (©Lunghammer, TU Graz).

(b) Linear accelerator.

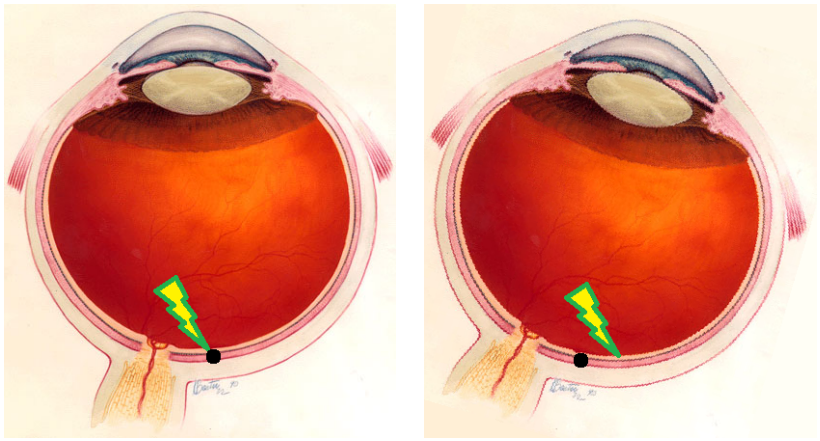
Figure 1.1: Devices for localization and treatment of tumors.

Several difficulties can arise due to head or eye movements of the patient. In Figure 1.2 we see, that movements during *MRI* cause strong artifacts in the resulting slice images, and the planning of the treatment becomes inaccurate. Let us now assume, that the head does not move during a subsequent radiation therapy session. Even if only the gaze direction deviates from the one imaged in the *MRI* system, a healthy region is radiated and the eye tumor is spared, which clearly should be prevented (Figure 1.3).



(a) Correct slice image. **(b)** Head movement.

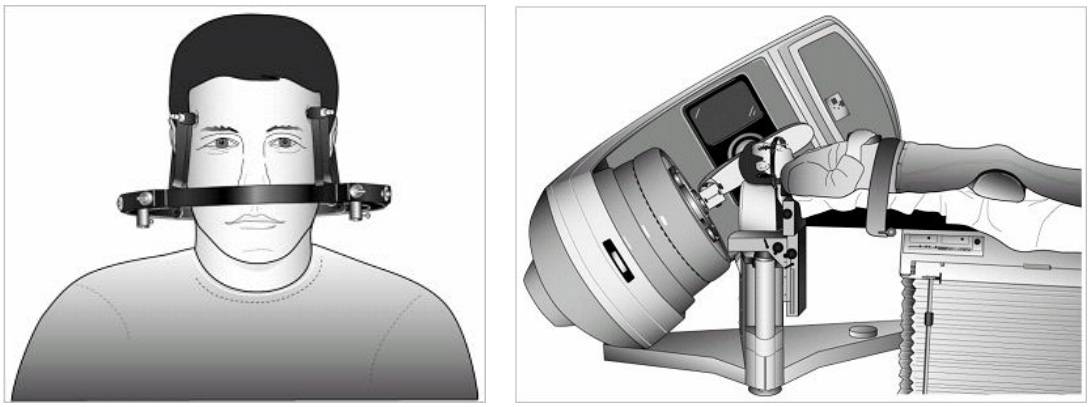
Figure 1.2: Artifacts caused by patient movement during MRI.



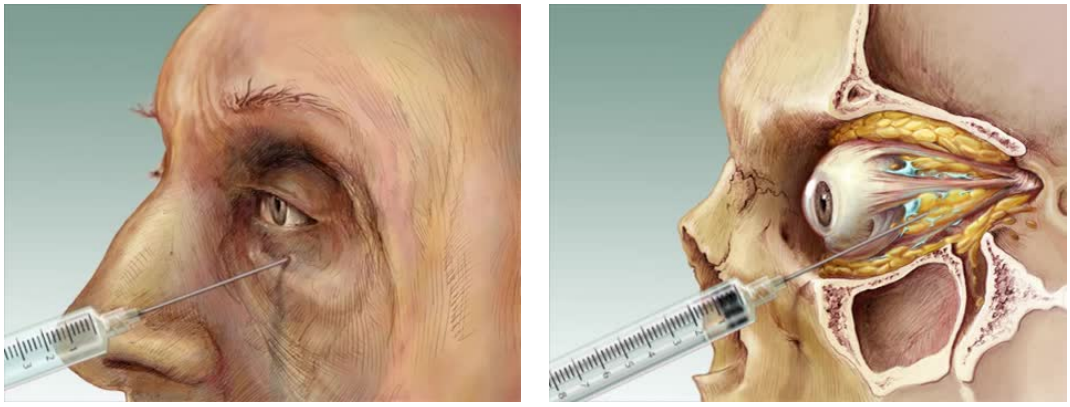
(a) Correct treatment. **(b)** Wrong treatment.

Figure 1.3: Therapy success depends on the eye gaze direction.

Therapy success depends on a fixed head and eye position. To avoid voluntary and involuntary movements, commonly head and eyes of the patient are immobilized. The scull is rigidly fixed in a frame (Figure 1.4a) which is screwed onto the treatment couch. The muscles behind the eye become anesthetized (Figure 1.4b) and tied to the frame with surgical threads.



(a) Stereotactic frame (taken from [22]).



(b) Retrobulbar anesthesia (taken from [2]).

Figure 1.4: Immobilization of head and eyes during eye tumor treatment.

In contrast, the head and eye tracking system is a non-invasive approach, which gets along completely without fixation. The patient comfort is increased and it even becomes possible to perform repeated and shorter treatments.

1.2 Related work

Head and eye tracking is still a topic of current research. A wide range of application can be found in driver surveillance, e.g. to analyze the field of view or the blind spot of the driver to assist if pedestrians, traffic signs or vehicles are overlooked, or, if there is a risk that the driver is inattentive or falls asleep. Further applications can be found in virtual reality, in human-computer interaction even for paraplegic persons, or in medicine.

This Master's Thesis focuses on head pose estimation and many different approaches exist for this task. Depending on the application and also the required accuracy, different hardware setups, sensor types and methods may be suited. We will now give a short overview on related work.

In [4], the head pose and gaze of a driver is estimated using an approach based on the Iterative Closest Point (ICP) algorithm ([5]). Only a rough estimate of the head pose using a small point cloud can be determined, otherwise the computation would be too costly. Additionally, the algorithm only converges against a local maxima and proper initialization is necessary. The Go-*ICP* algorithm from [24] is able to determine the global maximum, but is too costly because it runs through a large number of translations and rotations for initialization. All in all, the *ICP* algorithm may be too inaccurate when only a sparse point cloud is used, and otherwise too slow for real-time applications.

To speed up the pose estimation of an object, typically only a small set of feature points is used. In the case of head pose estimation, the face is not descriptive enough to extract e.g. SIFT features, because it is sparsely textured. Thus, a rather complex method is used for the robust facial feature tracker with a 2D camera presented in [20]. Rigid facial feature points represented by Gabor-filtered samples and statistical shape models for the shape of the face are both trained offline on a face database.

In [23] a head tracking system designed for the use during radiation therapy is presented. A mouthpiece has to prepared specifically for each patient and an infrared marker frame is attached, which is tracked by a infrared stereo vision system. In this way correspondence points are established very easily, which are uniquely detectable and move rigidly with the head.

In [27], a stereo-vision system is used for head tracking without other equipment. Three facial features are learned, the nose tip and the eye brows. A Kanade–Lucas–Tomasi tracker and template matching is applied to track the features in 2D. In combination with the depth values, their 3D positions and further the head pose can be computed.

In [19], the head pose of a vehicle operator is estimated using a Time of Flight (ToF) sensor. If other workers are in the proximity of the construction vehicle and are possibly overlooked because they are in the blind spot, a warning signal can be triggered. At the start of tracking it is assumed that the nose tip is closest to the camera. To estimate a coarse head pose, a 3D line is fitted through the nose bridge and a symmetry plane is calcuted. They use principal component analysis and support vector regression to compute an exact head orientation.

The authors of [14] also use a *ToF* camera and rely on the nose as a sufficient feature for face tracking. Their method is based on the assumption, that the roll angle of the head is negligible small compared to pitch and yaw. Spheres placed around the nose tip are intersected with the face. A 3D line can be fitted through the midpoints of the spherical intersection profiles, which provides the pitch and yaw angles of the head pose.

The method in [8] does not rely on a single facial feature like the nose tip, but uses the whole facial image to estimate a head pose. A regression between depth images and probabilities in the head pose space is learned, using random forests and synthetically generated training data. This makes it possible, that even parts of the face can be occluded. The focus of the paper lies on large variations of the head pose and the accuracy of the method is evaluated in areas of $15° \times 15°$.

1.3 Objective of the thesis

Regarding medical eye tumor treatment, the focus lies on the high accuracy of a head pose estimation method. This Master's Thesis concentrates on the tracking of unwanted motion. The distinction, if a patient is currently in the initial head pose recorded at the first session or not, should become possible. Such a method would be a simplification of the medical treatment and an improvement of the patient comfort.

1.4 Contribution

Here, we shortly summarize the contribution of this Master's Thesis. A foundation is given by the MedEyeTrack system, which was developed by the Graz University of Technology and the Medical University of Graz. In this system, the head is rigidly fixed using a thermoplastic face mask and each eye is independently tracked using a separate 2D camera.

Based upon the MedEyeTrack system, we now want to create a new system in which the patient can move the head freely. For that reason, we develop a head tracking system which can be combined with the existing system without affecting it. For robust head pose estimation, we first make experiments to find distinct facial features and analyze their stability under varying facial expressions. We think about how a head pose can uniquely be described and the trackability of the facial features during head rotations. We develop a prototype system based on depth measurement with *ToF* cameras, which meets the requirements of eye tumor treatment. First, the head tracking system has to be small enough to fit in the limited space of an *MRI* tube. Further, we test the correct operation of *ToF* cameras under the strong magnetic field of the *MRI* system. We do a research on algorithms which may be suited for robust head pose estimation. Then, we use MATLAB to implement three algorithms, adapt them to our head tracking system and analyze their accurateness and real-time capability. For this purpose, we find an error measure which describes the deviation of an estimated head pose from a reference pose. We acquire a series of video sequences, create manually labeled ground truth data and make a detailed statistic evaluation and experiments on the performance of our algorithms. In our future work we will port the MATLAB algorithms to *C++*. That is why we also create an interface in this Master's Thesis, which allows seamless integration of *C++* code from the head tracking system into the MedEyeTrack software, which was written in *C#*.

1.5 Organization of the thesis

Chapter 2 offers a detailed explanation on the development of the head and eye tracking system. Chapter 3 gives a theoretical background. In Chapter 4 we discuss the problem of head pose estimation and compare methods which may be suited for tracking during medical eye tumor treatment. Chapter 5 shows some experiments and compares the results of the methods. Finally, Chapter 6 gives a conclusion and an outlook to future work.

2

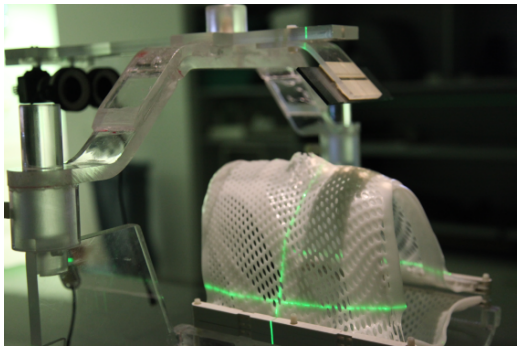
Development of a head and eye tracking system

Contents

In this chapter we describe the development of the head and eye tracking system presented in this Master's Thesis. A foundation is given by the MedEyeTrack system, which is an eye tracking system with non-invasive head fixation. It uses eye region detection and pupil center localization. In a seminar project, we followed the idea of a fixation-free head and eye tracking system. We analyzed facial features and implemented a head tracking prototype based on template matching. In this Master's Thesis we continue the work from the seminar project. We want to find a robust method for head pose estimation with sufficient accuracy for eye tumor treatment. First, we find a hardware setup suited for this purpose. Further we address, how the head tracker will be integrated into the existing MedEyeTrack system. We implement three methods for head pose estimation and compare them in a statistic evaluation.

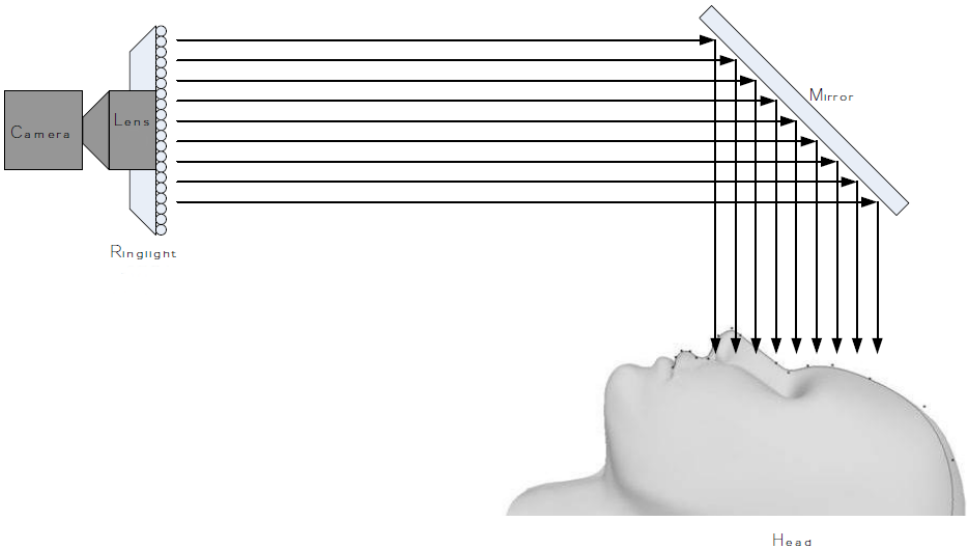
2.1 MedEyeTrack: Eye tracking with 2D cameras

In a project cooperation of the Graz University of Technology and the Medical University of Graz the MedEyeTrack system was developed (Figure 2.1). A thermoplastic face mask is used to rigidly fix the head to the system. In Figure 2.1c we see the schematic structure of the eye tracking system. A 2D camera for each eye is used to observe their movement during treatment. Due to the lack of space in the Magnetic Resonance Imaging (MRI) tube, the cameras are mounted over the chest of the patient and directed onto the eye region with the help of a mirror. Active infrared illumination is used to get independent from environmental lighting conditions.



(a) A plexiglass prototype system.

(b) The system for clinical use (taken from [15]).



(c) Schematic structure.

Figure 2.1: Hardware setup of the MedEyeTrack system.

The MedEyeTrack system can be coupled with the input of a therapy device. It includes a software to find the eyes and subsequently the pupil centers, by using a Haar-cascade feature detector and a blob detector. In this way, the current 2D positions of the pupils can be observed (see blue pluses in Figure 2.2).

At the beginning of the *MRI* session, round markers indicating the eye positions of the patient are initialized. In order to guarantee an optimal treatment, these initial positions should continuously be held during the acquisition of the slice images and also during the subsequent radiation therapy sessions. The software measures the deviation of the pupils (see diagrams), triggers the devices only inside of a defined range and pauses them otherwise. As a result, artifacts caused by movements during *MRI* are minimized and only the true tumor position gets radiated.

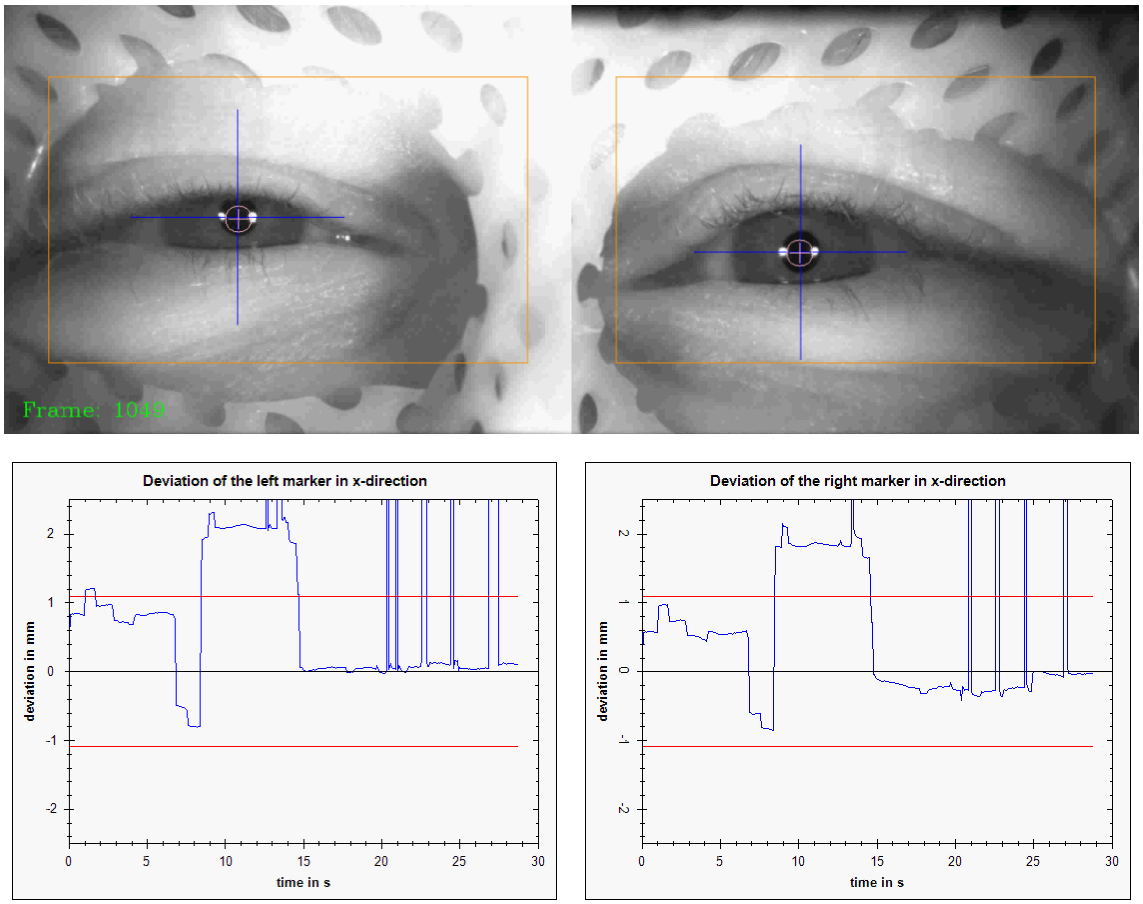
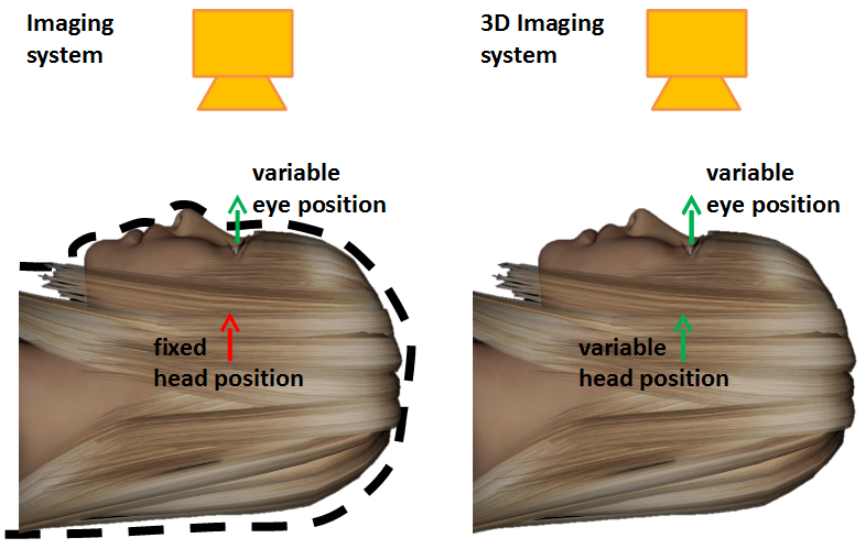


Figure 2.2: MedEyeTrack software and diagrams of the pupil misalignment.

2.2 Facial feature analysis

As discussed in Section 1.1, mostly immobilization of head and eyes of the patient is applied to prohibit any movements. With the MedEyeTrack system and its thermoplastic face mask a great alternative to an invasive fixation has been found. One obvious disadvantage of the system is, that the head of the patient has to be rigidly fixed to the system with a specifically prepared face mask. In Figure 2.3 we see a comparison of the MedEyeTrack system and a new fixation-free system based on a Time of Flight (ToF) camera, which we began to develop in our seminar project. Due to the active measuring principle, it is independent of lighting conditions and can directly extract depth information of the face. For the acquisition of video sequences from *ToF* cameras, we implemented an application with a Graphical User Interface (GUI) (see Figure 2.4). After analyzing the *ToF* sequences of several subjects with varying facial expressions we came to the conclusion, that the nose tip and the inner eye corners are stable facial features which can be used for head tracking. Further, we implemented a head tracking prototype based on template matching.



(a) MedEyeTrack system. **(b)** New system.

Figure 2.3: Comparison of the operating principles of the tracking systems.

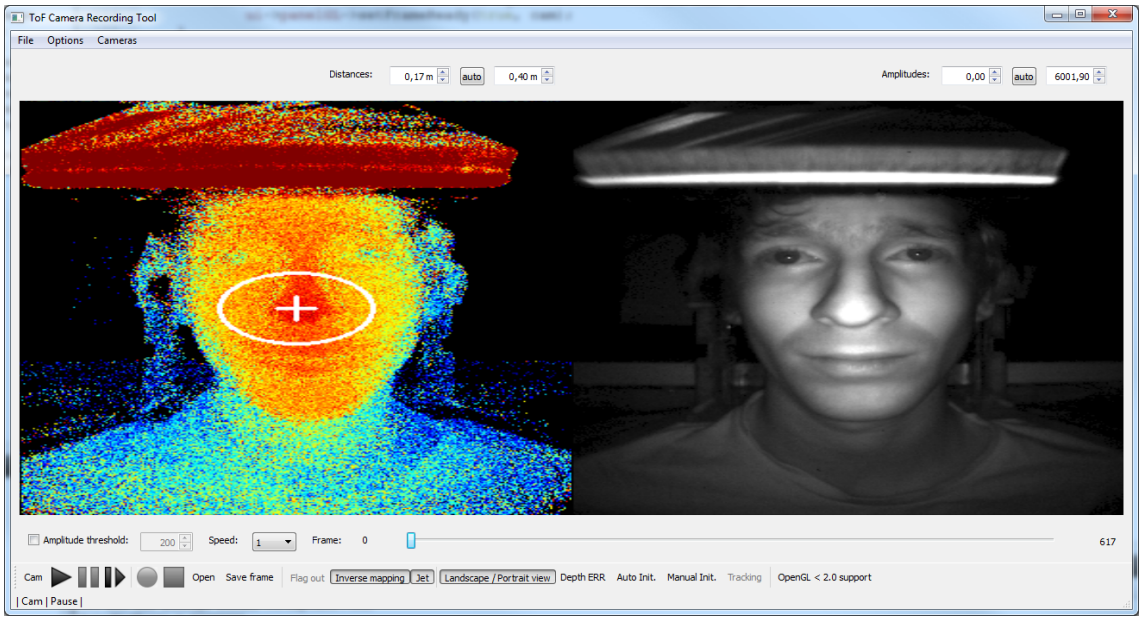


Figure 2.4: GUI application (developed in Qt).

2.3 Robust head pose estimation

2.3.1 Overview

To sum up, we want to build a head and eye tracking system, by combining the existing MedEyeTrack system with a head tracking system. The objective is to receive data of unwanted head or eye motion, which lets us control the treatment by generating triggering signals, if a certain initial position is held by the patient. In this Master's Thesis we focus on the implementation of three head pose estimation methods, which are examined for their accurateness and robustness. One of them could possibly be appropriate for the head tracking system. An overview on the development process is given in Figure 2.5.

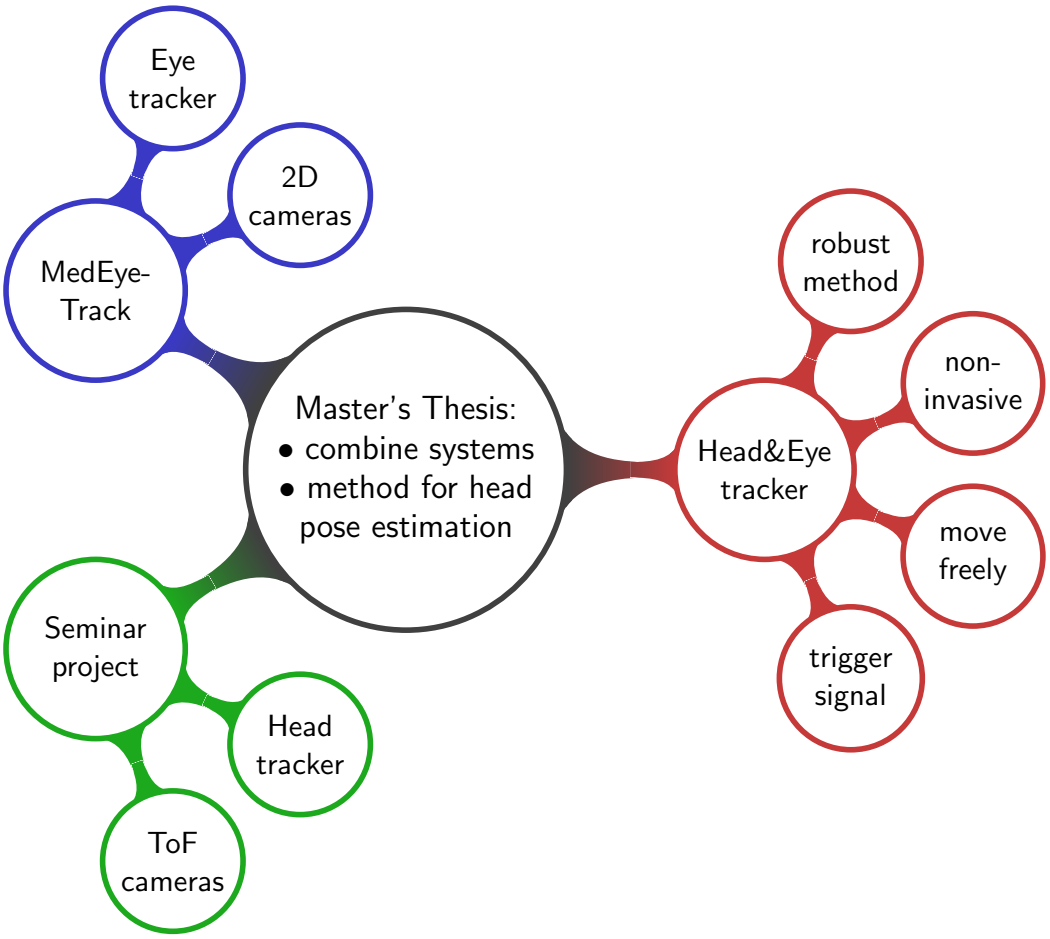


Figure 2.5: Development of the head and eye tracking system.

2.3.2 Hardware setup

Regarding hardware requirements, a compromise has to be found. There is not only not only very limited space available in the narrow *MRI* tube, but also interactions between

the strong magnetic field and the imaging sensor are possible. For one thing, the magnetic field can influence the imaging process of the sensor, for another thing, metal parts of the sensor distort the imaging process of the *MRI* system (Section 3.1.1). That's why we choose a *ToF* sensor for head tracking, which has only few metal parts. It is generally much smaller than sensors like the Microsoft Kinect™ which uses a stereo vision principle (projector and camera), but has a much lower resolution and introduces more noise. These limitations are in direct contrast to the goal of accurate head pose estimation.

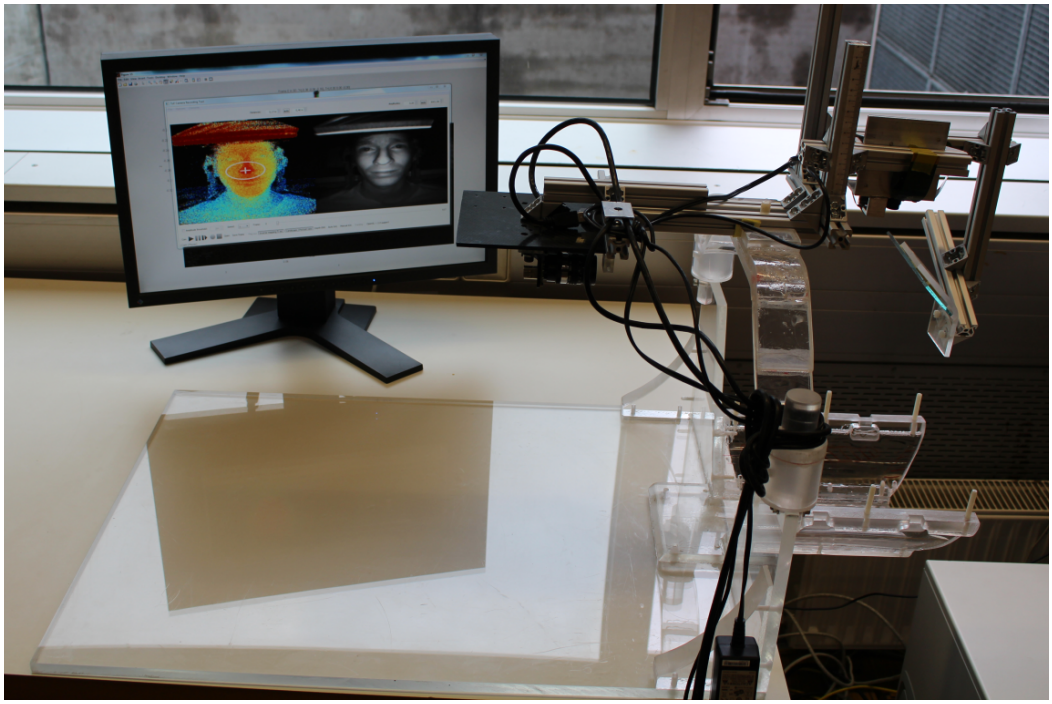
In this Master's Thesis, we build upon the plexiglass prototype of the MedEyeTrack system (Figure 2.1). For pupil tracking, the eyes of a subject have to be aligned with the mirror so that they look directly into the 2D infrared cameras (see Figure 2.6b). For head tracking, we extend this hardware setup by a *ToF* camera directed onto an approximate nose tip position of the subjects. It is oriented in portrait format and mounted very near to the face, so that the low resolution is best utilized. The head rests on the back plane of the system and is fully visible during all motions. Each subject has a different distance of the nose tip to the *ToF* camera, approximately in a range of 15 to 20cm. This is a large variation compared to the distance from the camera. In Section 3.1.2.2 the significant role of a *ToF* camera's integration time on the measuring range and hence the accuracy of the acquired data is discussed. In an experiment we determined an integration time of 400μs at a medium distance of 17.5cm to be the best compromise between the introduced noise and distortion, by comparing faces measured with the *ToF* camera to a high-resolution scan from a Structured Light (SL) scanner. The hardware setup is shown in Figure 2.6.

2.3.3 Software integration

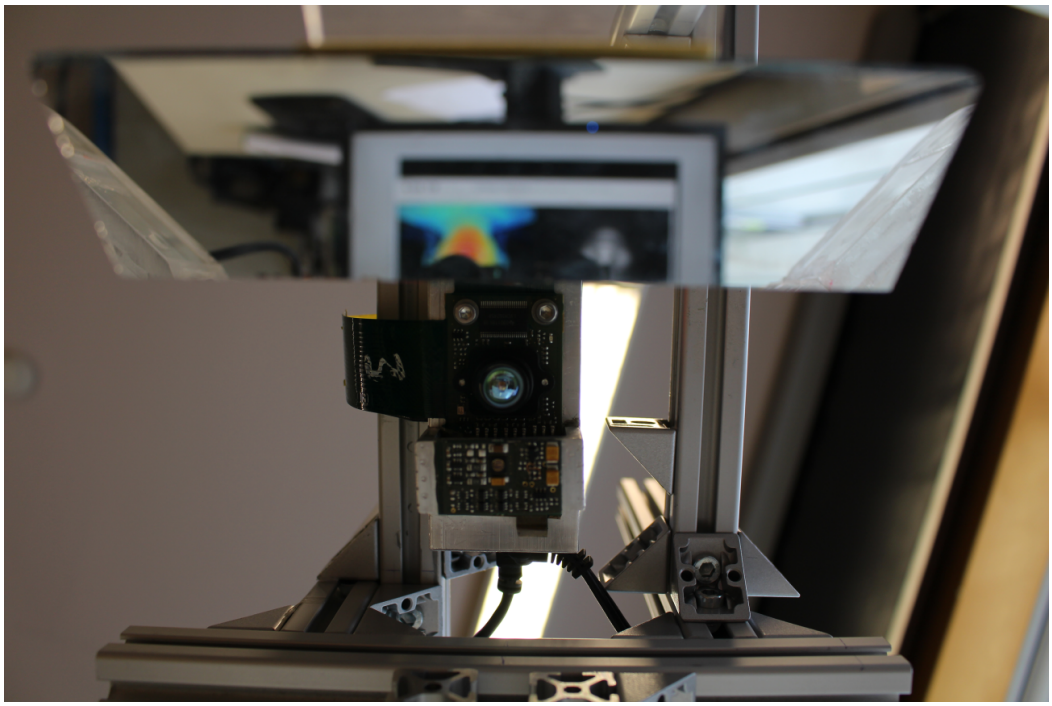
Here, we want to briefly mention some details on the integration of the head tracker into the existing MedEyeTrack system. In our seminar project we used Qt as a development environment and the OpenCV library for image processing to create a *GUI* application in *C++*. In this Master's Thesis we use MATLAB to implement prototypes for the head pose estimation methods, which are integrated into the application in our future work. In this way, the head tracker can be developed independently of the eye tracking system.

The MedEyeTrack software was implemented in *C#*. In contrast to *C++* code, which is directly translated to machine code, *C#* code runs in a runtime environment. To integrate the unmanaged *C++* code (*ToF* and OpenCV libraries and the head tracking application) into the managed *C#* code of the MedEyeTrack application, we created an interface using a wrapper library. With Microsoft Visual Studio this is possible using Visual *C++* and Dynamic Link Library (DLL) export and import keywords. First an unmanaged Win32 export *DLL* and then a managed wrapper Common Language Runtime (CLR) class library has to be created. In this way all methods of the head tracking system can simply be reused in the MedEyeTrack system with the class library as a reference.

The head tracker can notify the MedEyeTrack system when the head is in its initial pose. If the eyes are as well in their correct position, a treatment device will be triggered.



(a) Prototype system.



(b) Patient's view.

Figure 2.6: Hardware setup.

3 Theory and background

Contents

3.1 Imaging sensors

Next, we give a short overview on the properties and measurement principles of the sensor types we use in this project. But to begin with, we want to mention the possibility of interactions between the strong magnetic field of an Magnetic Resonance Imaging (MRI) device and the sensors brought into this field. To conclude this section, we explain the 2D calibration of the sensors.

3.1.1 MRI suitability of materials

Ferromagnetic materials - like for example iron, nickel or cobalt - can become very dangerous in the proximity of *MRI* systems. They can develop a force corresponding to a multiple of their own weight, are attracted to the center of the *MRI* system and do not stop at obstacles like persons, thus resulting in serious injuries or even death. Experiments showed, that also paramagnetic materials cause unwanted irradiated frequencies which lead to artifacts in the image (Figure 3.1). This has to be taken care of during the construction of a device which will be used in an *MRI* system. For the development of the eye tracking system, a plexiglass prototype has been constructed. The Time of Flight (ToF) camera has also been tested in the *MRI* system, it had no influences on the imaging process and showed correct operation under the strong magnetic field.

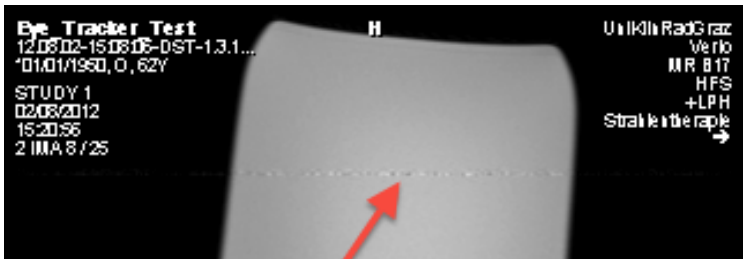


Figure 3.1: Paramagnetic materials cause artifacts in the MR image.

3.1.2 Types of sensors

3.1.2.1 Intesity-based camera

Intensity-based cameras produce a 2D image of the scene. In order to get independent from environmental lighting conditions, an infrared camera in combination with a ring light for active illumination is used. Infrared light is similar to visible light, but has no heat information. In the MedEyeTrack system a 2D camera is used for each eye region (see Figure 3.2), to allow eye tracking with high accuracy. Due to the strong magnetic field in the *MRI* system, cameras with a CMOS sensor have to be used, otherwise the electrical charges on a CCD chip get distorted and the resulting image becomes useless.

If we use a single 2D camera for the task of head pose estimation, no 3D translation can be determined. A stereo vision system with two cameras on a stereo rig and known relation between the cameras would be necessary to observe the head, which is not possible because of the very limited space in the *MRI* tube. Thus, we use a *ToF* sensor.

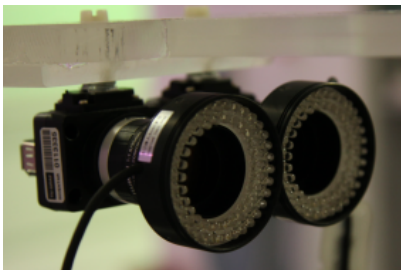


Figure 3.2: Two 2D cameras for eye tracking.

3.1.2.2 ToF sensor

ToF cameras were first proposed in [12]. Unlike 2D cameras, they use an active measuring principle (Figure 3.3) and are independent of lighting conditions. Compared to stereo vision systems, the sensor is very small because no stereo basis is necessary to measure depth information. *ToF* cameras were invented just about a decade before this work and they are still at the beginning of development, so only low resolution sensors are available at the moment.

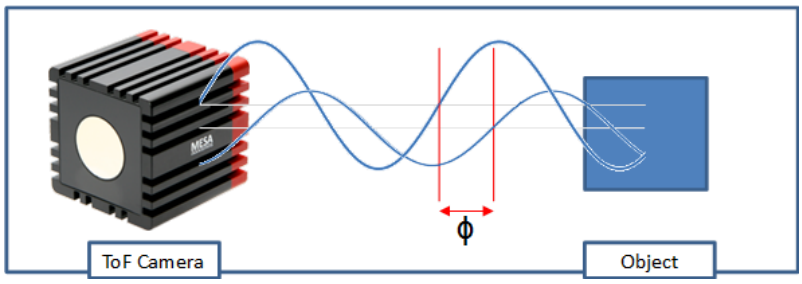


Figure 3.3: Measuring principle of a ToF sensor.

An amplitude modulated near infrared light signal is emitted, reflected at the object and projected onto the image sensor. Based on the known speed of light, the phase shift between incoming and outgoing light at each pixel can be used to measure depth information. What we get is a 2.5D image of the scene, i.e. an infrared image and the corresponding depth map.

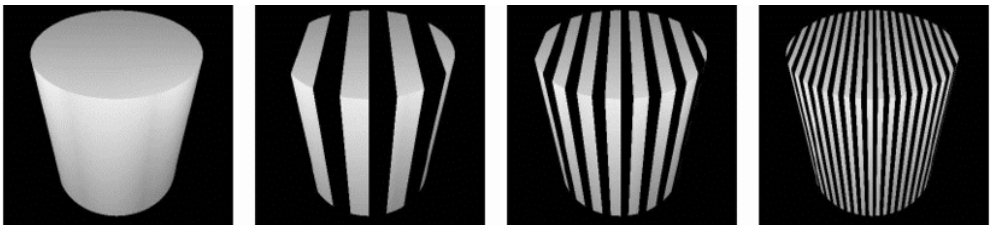
Each *ToF* camera works at a specific modulation frequency. To avoid interference between two cameras, they have to work on different modulation frequencies.

An important parameter is the integration time, which defines how much light can reach the image sensor. In [28], the effect of the integration time on the accuracy of a head tracking system and its direct influence on the distance measurement are discussed. Too distant objects cause the acquired data to be noisy and uncertain. Too near objects may cause saturation and distort the measured distances.

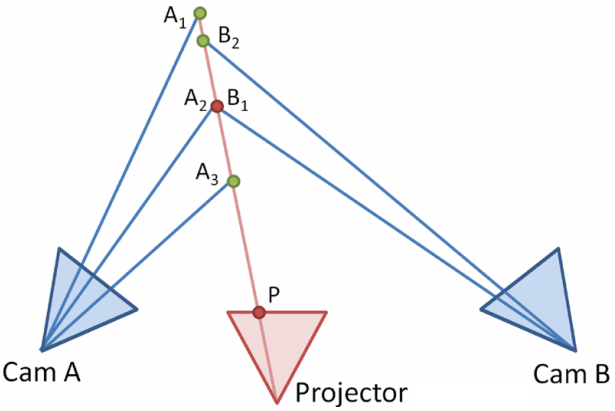
3.1.2.3 Structured light 3D scanner

The basic problem of passive stereo vision systems is to find correspondences. A 3D point can only be triangulated if a point correspondence is known. It is very hard to establish correspondences for every point of an object, especially when it is sparsely textured, like it is the case for a face. In [16], a good introduction to Structured Light (SL) systems can be found. They are similar to passive stereo vision systems, just that one of the two cameras is replaced by a projector. A temporal encoded stripe pattern is projected into the scene (Figure 3.4a), which assigns a binary code to every illuminated pixel. This active measurement principle makes it a lot easier to find correspondences, since the pixels can be uniquely distinguished from their neighbors.

Like in [25], we use a *SL* system with a projector and two cameras for reconstruction of a high resolution 3D face model. A single camera setup has the disadvantage, that only one side of the face is fully visible and for example the opposite side of the nose is occluded. In a two camera setup, there is one camera for either side of the face and the reconstruction gets much better. An overview on the active triangulation principle is given in Figure 3.4b.



(a) A temporal encoded stripe pattern is projected onto an object (taken from [16]).

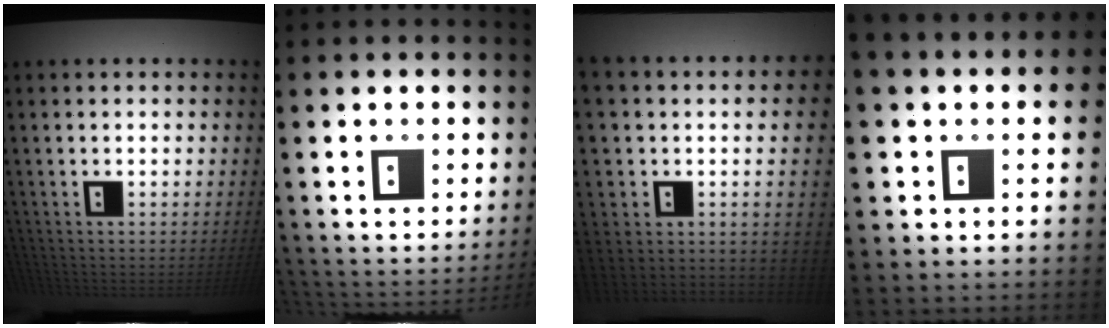


(b) Stereo camera geometry: the projector makes it easier to find correspondences in the camera views (taken from [11]).

Figure 3.4: Principle of a structured light 3D scanner.

3.1.3 2D calibration

To undo lens distortion, the sensors need to be calibrated, which is here demonstrated for the *ToF* camera. We acquire images of a target at several viewing angles and do a calibration based upon [26] and [6]. Figure 3.5 gives a comparison of the distorted and undistorted target and we can see, that the slightly distorted grid of black points and also the target edge have become straight again.



(a) Distorted calibration target. **(b)** Undistorted calibration target.

Figure 3.5: 2D calibration of the ToF camera with a control point target.

3.2 Geometric definitions

3.2.1 Definition of a coordinate system

The *ToF* camera records an infrared image and a depth map. From the depth map we can compute a 3D point cloud of the facial surface. For convenience, we choose a right-handed coordinate system suitable for the data from the *ToF* camera (see Figure 3.6a). The x- and y-axis point to the same directions as in the acquired images (y-axis facing downwards), and the z-axis looks in view direction of the *ToF* camera. This way the nose lies at the minimal z-value. A common choice for the origin is at the centroid of the head or at the top of the neck (beginning of the backbone). Similar to [18], we place it at the nose tip. This makes it easier to align 2.5D data, which does not contain information about the back of the head. In addition, the nose tip is a good choice of a common reference point for a learning-based method we want to implement in our future work.

3.2.2 Rigid-body transformations in 3D

In our seminar project we experimented with several facial features and found, that the nose tip and the inner eye corner points do not change their positions notably during facial expressions. Thus, in our methods (Section 4.3 and 4.4) we assume that the face is a rigid body. In this way we can describe all head poses by just rotation and translation between two corresponding 3D point clouds $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$ (homogeneous 3D coordinates):

$$H_{rigid} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}, \text{ with } \mathbf{R} \in (3 \times 3),\ \mathbf{t} \in (3 \times 1)$$

$$\{\mathbf{q}_i\} = H_{rigid} \cdot \{\mathbf{p}_i\} = T \cdot R \cdot \{\mathbf{p}_i\}, \text{ with } H_{rigid},\ R,\ T \in (4 \times 4)$$

A rigid-body transformation preserves distances between points. We need at least three point correspondences to compute the rigid-body transformation with its 6 Degrees of Freedom (DOF).

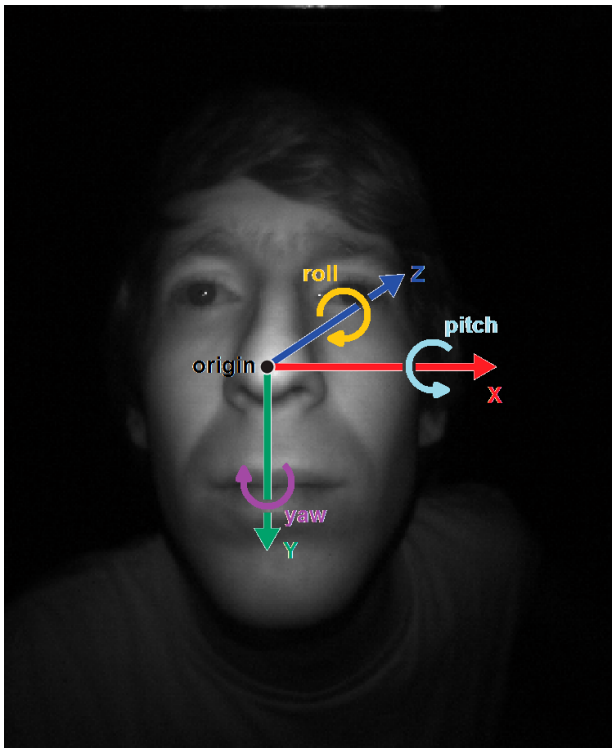
Estimation using Singular Value Decomposition (SVD): To compute an optimal rotation $\mathbf{R}$ and translation $\mathbf{t}$, which best transform the point set $\{\boldsymbol{p}_i\}$ to $\{\boldsymbol{q}_i\}$ (inhomogeneous 3D coordinates), we need to minimize the error

$$E = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{R}\boldsymbol{p}_i + \mathbf{t} - \boldsymbol{q}_i|^2.$$

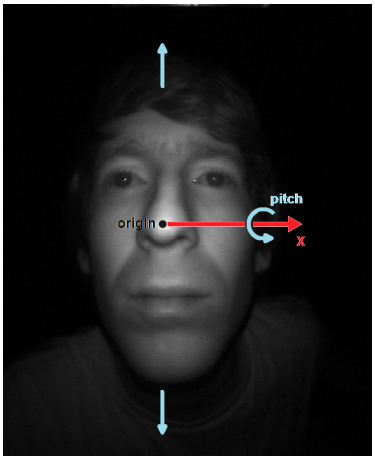
This is established by the following steps (adapted from [17]):

- center both point sets: $\widetilde{\boldsymbol{p}}_i = \boldsymbol{p}_i - \bar{\boldsymbol{p}}$ and $\widetilde{\boldsymbol{q}}_i = \boldsymbol{q}_i - \bar{\boldsymbol{q}}$
- compute a correlation matrix of the centered point sets:
  $C = \widetilde{P} \cdot \widetilde{Q}^T$ with P and $Q \in (3 \times N)$

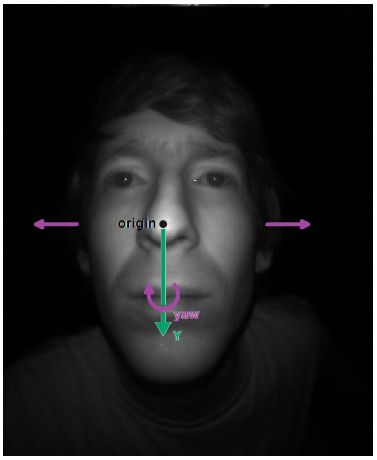
- perform an *SVD* on the correlation matrix: $C = U \cdot D \cdot V^T$
- construct the rotation matrix from its results: $\mathbf{R} = V \cdot diag(1, 1, det(V \cdot U^T)) \cdot U^T$
- compute an optimal translation: $\mathbf{t} = \bar{\boldsymbol{q}} - \mathbf{R}\bar{\boldsymbol{p}}$



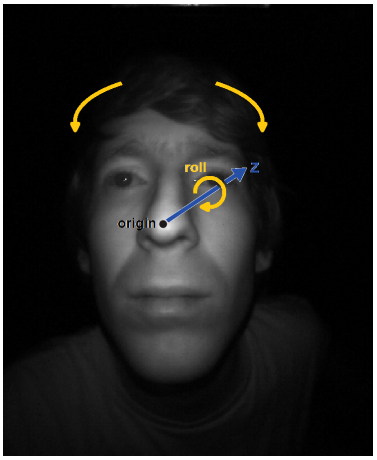
(a) The origin lies at the nose tip.



(b) Pitch: a rotation around the left-right-axis, like shaking the head yes.



(c) Yaw: a rotation around the up-down-axis, like shaking the head no.



(d) Roll: a rotation around the front-back-axis, like shaking the head maybe.

Figure 3.6: Definition of a coordinate system (right-handed, y-axis down).

3.2.3 Arbitrary 3D rotations

The following definitions are taken from [1]. Rotations about a single coordinate axis are easy to define (also have a look at Figures 3.6b to (d)):

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$R_z(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This rotations about single coordinate axes can be combined to a rotation matrix R, which allows for an arbitrary rotation of an object in 3D. One possibility would be to factor a rotation as $R = R_x(\varphi) \cdot R_y(\theta) \cdot R_z(\psi)$ (ordering xyz, with Euler angles φ, θ and ψ). Rotation matrices are not commutable, so all of the five other possible combinations xzy, yxz, yzx, zxy and zyx yield different results, when using the same angles. In our case we choose the ordering

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\varphi).$$

Worth mentioning is the fact, that the inverse of R is

$$R^{-1} = R_x(-\varphi) \cdot R_y(-\theta) \cdot R_z(-\psi) \overset{!}{\neq} R_z(-\psi) \cdot R_y(-\theta) \cdot R_x(-\varphi).$$

Contrary to possible expectations, the ordering specification has an influence on the value of the angles:

$$R^{-1} = R_z(\tilde{\psi}) \cdot R_y(\tilde{\theta}) \cdot R_x(\tilde{\varphi}),$$

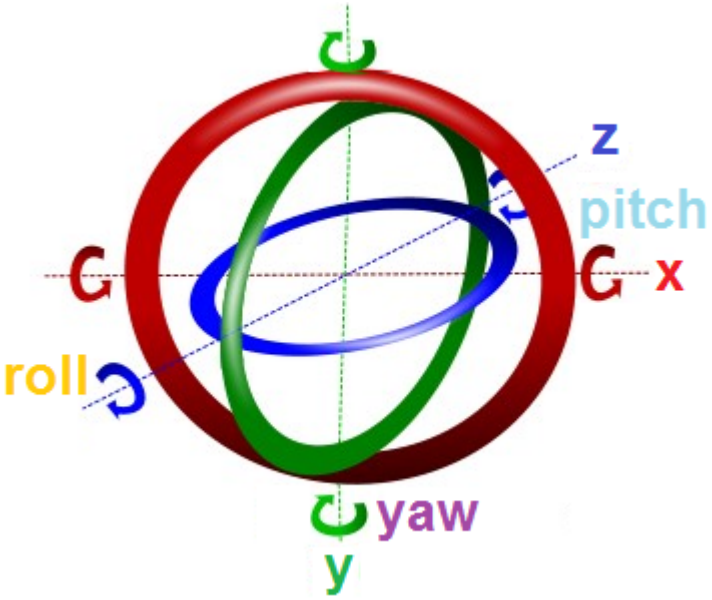
with $|\tilde{\varphi}| \neq |\varphi|$, $|\tilde{\theta}| \neq |\theta|$ and $|\tilde{\psi}| \neq |\psi|$. We get completely different Euler angles, which we illustrate by the following example:

$$R = R_z(30°) \cdot R_y(20°) \cdot R_x(10°)$$

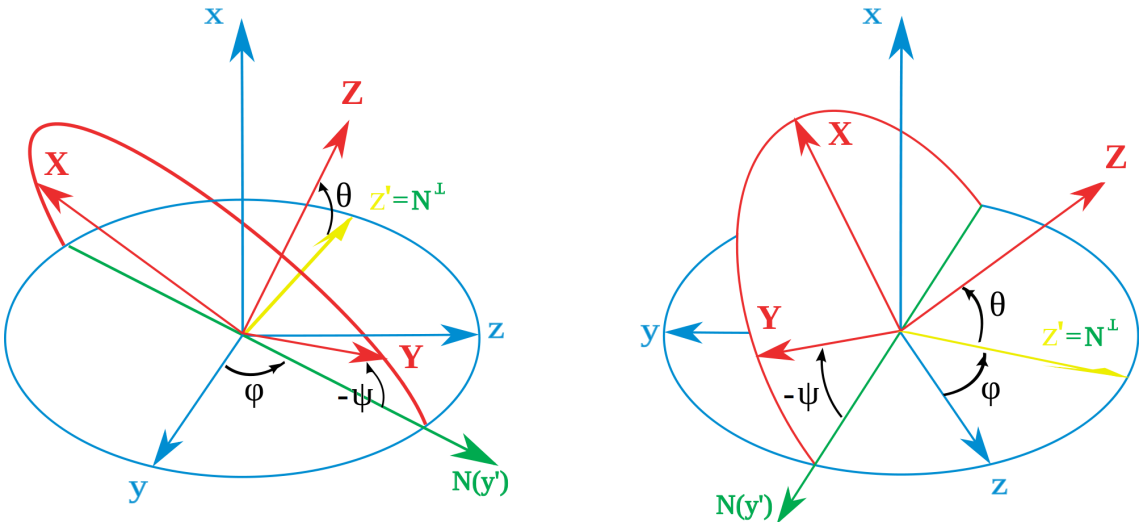
has the inverse

$$R^{-1} = R_x(-10°) \cdot R_y(-20°) \cdot R_z(-30°) = R_z(-28.45°) \cdot R_y(-22.24°) \cdot R_x(1.12°).$$

By comparing the dependence of successive rotations to a gimbal system, we also get a very descriptive explanation (Figure 3.7a). Figure 3.7b shows the succession of rotations of the coordinate frame.



(a) Gimbal system (adapted from [9]).



(b) Euler angles (adapted from Wikipedia).

Figure 3.7: Explanatory figures for successive rotations of the ordering zyx.

3.2.4 Computation of Euler angles from a 3D rotation

In [1] one can find a detailed explanation on the computation of Euler angles φ, θ and ψ from a given rotation matrix R (upper-left 3×3 submatrix of a homography H). These represent the rotational components $R_x(\varphi)$, $R_y(\theta)$ and $R_z(\psi)$ about single coordinate axes. Algorithm 1 specifies the computation of Euler angles for the ordering $R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\varphi)$:

Algorithm 1 For a rotation matrix R with the ordering $R_z(\psi) \cdot R_y(\theta) \cdot R_x(\varphi)$, compute the Euler angles φ, θ and ψ which represent the rotational components about single coordinate axes (algorithm taken from [1]).

procedure ROT2EULER($R_{3\times3}$)
  if $r_{20} < +1$ **then**
    if $r_{20} > -1$ **then**
      $\theta = \arcsin(-r_{20})$
      $\psi = \text{atan2}(r_{10}, r_{00})$
      $\varphi = \text{atan2}(r_{21}, r_{22})$
    else ▷ $r_{20} = -1$ ▷ Not a unique solution: $\varphi - \psi = \text{atan2}(-r_{12}, r_{11})$
      $\theta = +\frac{\pi}{2}$
      $\psi = -\text{atan2}(-r_{12}, r_{11})$
      $\varphi = 0$
    end if
  else ▷ $r_{20} = +1$ ▷ Not a unique solution: $\varphi + \psi = \text{atan2}(-r_{12}, r_{11})$
    $\theta = -\frac{\pi}{2}$
    $\psi = \text{atan2}(-r_{12}, r_{11})$
    $\varphi = 0$
  end if
end procedure

3.3 Registration with the ICP algorithm

With registration, we try to find a mapping between two views of the facial surface in order to minimize their distance. In our application, the first view is a data point cloud measured by a *ToF* camera and the second a high-resolution model acquired by a *SL* scanner. The result should be a rigid-body transformation, which maps the data point set to a corresponding set of model points. The problem is that we don't know which points correspond, which is complicated by changing facial expressions, sensor noise and a different sampling of the points.

An optimal solution to this problem offers the Iterative Closest Point (ICP) algorithm (see [5]), which assumes a set of closest points to be the corresponding points. It performs the following steps iteratively:

- compute a set of model points which is closest to the data point set
- compute an optimal registration of both point sets, i.e. a rigid-body transformation using *SVD* (see Section 3.2.2)
- transform the data point set by the estimated registration result
- stop, if the change of a distance error between the point sets is below a threshold

The algorithm computes an optimal registration result, but is sensitive to noise or outliers. This is due to the use of a squared error measure, which is computed by summing up the squared distances between the data and model points. This gives a strong weight to the outliers and the registered point cloud is incorrectly moved into their direction. To reduce the error caused by noise and outliers, it is important to appropriately filter the input point clouds.

Further a good initialization is necessary, so that the algorithm converges against the global and not only a local minimum.

The execution time of the algorithm depends on the sizes of the input point clouds, and is a few seconds in our case. Smaller point cloud sizes accelerate the execution, but lead to less accurate and unsatisfactory results. Thus, it generally can not be applied in real-time applications.

3.4 Template matching

In [3] a tutorial on template matching with the OpenCV library is given, here we give a short recap. Template matching is used to find areas in an image that are similar to a template image. For example, an eye template can be searched in the image of a face. To find the matching location of the template, it is slided over the image and at each pixel location a similarity measure is computed for the overlapping region. For the similarity measure we use the normalized correlation coefficient, which results in the image

$$R(x,y) = \frac{\sum_{x',y'} T'(x',y') \cdot I'(x+x',y+y')}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}},$$

with

$$T'(x',y') = T(x',y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} T(x'',y''),$$

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{w \cdot h} \cdot \sum_{x'',y''} I(x+x'',y+y''),$$

where T is the template and I the search image. This measure is based on the computation of the covariance of T and the overlapping region in I, normalized by their standard deviations. The highest score in R should then correspond to the location of the template in the search image.

3.5 Error measures

3.5.1 Mean absolute error

The Absolute Error (AE) is a scalar that tells us how close an estimated value is to the true value. We use it to compute the 3D distance error $AE = \|\hat{\mathbf{p}} - \mathbf{p}\|$ between two points, a point estimate $\hat{\mathbf{p}}$ and the true point $\mathbf{p}$ from the ground truth. The Mean Absolute Error (MAE)

$$MAE = \sum_{i=1}^{n} \frac{\|\hat{\mathbf{p}}_i - \mathbf{p}_i\|}{n}$$

is the mean distance error of all point pairs of a set $\{(\hat{\mathbf{p}}_i, \mathbf{p}_i)\}$.

3.5.2 Mean signed difference

The Signed Difference (SD) gives information about how much and in which direction an estimated value is displaced from a true value. For points, we get the displacement error $SD = \hat{\mathbf{p}} - \mathbf{p}$ between a point estimate $\hat{\mathbf{p}}$ and the true point $\mathbf{p}$ from the ground truth in three dimensions. The Mean Signed Difference (MSD)

$$MSD = \sum_{i=1}^{n} \frac{\hat{\mathbf{p}}_i - \mathbf{p}_i}{n}$$

is the mean value of all displacement errors of a set $\{(\hat{\mathbf{p}}_i, \mathbf{p}_i)\}$ of point pairs and its result is a 3D mean error vector $\mu = (\,\mu_x\ \mu_y\ \mu_z\,)^T$.

4 Head pose estimation

Contents

4.1 Problem statement

In our application, we need to assure that a treatment device is only triggered, when the patient's head is in a unique pose, namely a straight pose which is acquired during initialization. When the head leaves this initial pose, the device is paused. In a naive approach, we try to solve this task by observing the position of a single facial feature. We use the nose tip, which can always be detected at a depth minimum, because the face moves in a restricted range of only a few centimeters around the initial head pose in our application. Now we try to verify, if the head is currently in the initial pose, just by measuring the nose tip's deviation from its initial position. After short consideration this turns out to be unsatisfactory, because this naive approach can not recognize head pose changes caused by rotations around the initial nose tip position.

Head pose estimation is a hard problem, but why? The human face is a strongly deformable surface with only little texture, which is demonstrated in Figure 4.1. A minimum of three facial feature points is necessary to define a head pose uniquely. In our seminar project we had a focus on stable facial features suited for head tracking. We found that the nose tip and inner eye corner points are best suited, due to their central position and rigidness. In this chapter, we want to find a head pose estimation method meeting the

requirements of medical eye tumor treatment, under the use of the noisy data from a Time of Flight (ToF) sensor.

Figure 4.1: Variability of facial expressions.

4.2 Overview

4.2.1 Definition of a reference frame

We acquire *ToF* sequences of several subjects with an approximately straight initial head pose (Figure 4.2), using the hardware setup explained in Section 2.3.2.

Figure 4.2: Initial head pose in ToF sequences of four subjects.

In Section 3.2.1 we defined a coordinate system with its origin located at the nose tip. Now, we introduce a straight reference head pose with the coordinate frame located at the origin of the *ToF* camera. It is important to note the difference between reference and initial head pose. The reference pose at the origin is important for establishing a common frame for method comparison, while the initial pose somewhere in 3D space is the one in which the patient must remain during a therapy session. Every frame of a *ToF* sequence represents a head pose in 3D space, which is connected over a rigid-body transformation (3D rotation and translation, see Section 3.2.2) to the reference frame (see Figure 4.3).

Only if the current head pose lies within a certain threshold of the initial head pose, the MedEyeTrack system can perform pupil detection. If head and pupils are in their initial positions, a treatment device may be triggered.

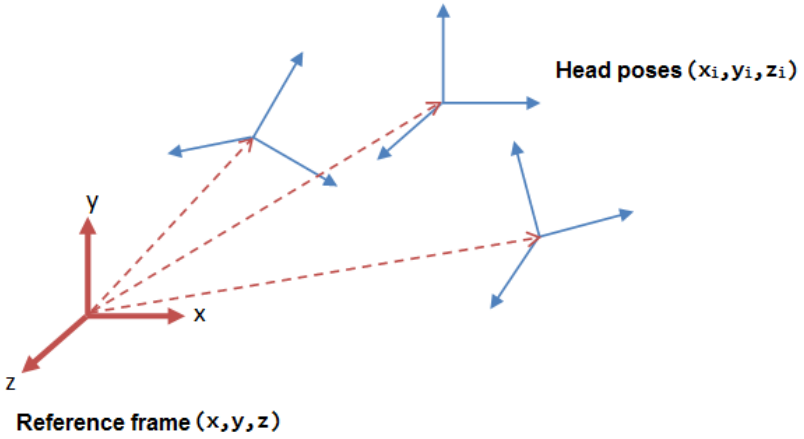


Figure 4.3: Relation of head poses and reference frame (adapted from [7]).

4.2.2 Methods

We want to find a method for head pose estimation with sufficient precision to allow the application in a medical environment. This chapter gives a detailed theoretical explanation on the three different approaches we implemented, here is a brief overview. In the following experiments and conclusion sections (Chapters 5 and 6) we analyze statistical results on the accuracy and discuss the methods' advantages and drawbacks.

Iterative Closest Point (ICP) method: The main idea of a registration-based method is the alignment of two point clouds. A regular grid is sampled on the face and the rigid-body transformation between two views can be determined. We compute the registration relative to a Structured Light (SL) reference model of the face (Section 3.1.2.3), which represents a straight head pose at the origin of the *ToF* camera. Registration-based methods are time consuming, but offer results with high accuracy. They only rely on 3D information and are thus independent of the appearance of an object. We investigate the performance on (noisy) *ToF* data and find out, if we get

reliable results which can be used as ground truth and training data for a learning-based method.

Template Matching (T3M) method: We extend the head tracking prototype from our seminar project (Section 2.2) and create a simple method based on template matching. The positions of three templates are sufficient to describe a head pose. Template matching computes the 2D similarity of an image patch, so it is dependent on the appearance of an object. We extend the common approach and additionally compute the similarity of the corresponding depth patch. We analyze the performance of the resulting 2.5D method and find out if it can compete with the other two methods.

Spherical Intersection Profile (SIP) method: The third method we implement is based on 3D face geometry analysis and is also appearance-independent. The main idea is to place spheres of varying radii at the nose tip and find the resulting intersections with the face. With these intersection profiles, which are somehow related to the contour lines of a mountain, we are able to analyze the facial surface and compute a viewing direction.

4.2.3 Preprocessing of the ToF data

Each *ToF* frame undergoes a preprocessing stage:

- undistortion of the infrared and depth image
- weak or strong median filtering and optional gaussian filtering
- Region Of Interest (ROI) detection

At the beginning, a *ToF* frame has to be undistorted. Section 3.1.3 explains the 2D calibration of *ToF* camera. The calibration data is used to undistort the infrared and depth image.

Characteristic for the *ToF* measuring principle is the rather strong noise on the depth data. It is necessary to preprocess each *ToF* frame in a filtering stage, with kernel sizes depending on the applied head pose estimation method. For the registration-based method, we avoid that the 2.5D model of the face will be deformed and only use a 7×7 kernel for median filtering. For the methods based on template matching or topographic analysis stronger median filtering (15×15 kernel) and additional gaussian filtering (7×7 kernel) is suited. For time efficiency reasons, the filter kernels are approximated by successive filtering in both dimensions ($i \times i \rightarrow i \times 1$ and $1 \times i$).

After the filtering a *ROI* can be determined. In our application, the patient is advised to remain in a straight position. Comparable to a passport photo, the face is always prominently visible during the whole *ToF* sequence and we only need to take care of a limited range of movements. Besides of the high accurracy this restricted setup offers, the patient's face can also easily be extracted via depth thresholding. The nose tip always remains at the depth minimum. An elliptical region around the nose tip forms the *ROI*.

4.3 ICP method: A registration-based approach

4.3.1 Motivation

First of all, we want to investigate a method based on registration with the *ICP* algorithm (see Section 3.3). The registration of two point clouds typically takes several seconds, so this method is not suited for the real-time case, where a model must be compared to a video sequence with several frames per second. Nevertheless, we want to investigate if it can be used for creating ground truth data with high accuracy.

4.3.2 Overview

Our method consists of the following steps:

- aquisition of high resolution models with a *SL* scanner
- alignment of the *SL* scans in a reference frame at the origin of the *ToF* camera
- estimation of the initial head pose (first frame of a *ToF* sequence)
- registration of all frames of a *ToF* sequence
- computation of the head pose deviations from their initial position

Right its use, a *ToF* frame is preprocessed according to Section 4.2.3.

4.3.3 Acquisition and alignment of SL scans

With a *SL* scanner we acquire a high resolution model for each subject:

Figure 4.4: Structured light scans of some subjects.

These high quality reconstructions of the face are then used to register each frame of the lower quality *ToF* sequences to them. In this way we get an estimate of each subject's motion in relation to a common reference frame with all nose tips at the origin (defined in Section 4.2.1, also see Figure 3.6).

For the alignment of the *SL* models we first detect the nose tips at the depth minima of the point clouds and move them to the origin. The inter-subject registration of models fails due to the huge variation of the facial topology. Instead, after a rough alignment to the xy-plane by registration, we do a horizontal symmetry correction. The model is iteratively rotated about the y-axis, according to the enclosed angle between the left and right cheekbone and the x-axis (height deviation). The heights of the cheekbones are estimated by computing the median values of small patches to the left and right of the nose tip. After that we do a vertical symmetry correction, as described in [18]. In this paper, faces are rotated about the x-axis until the nose bridge is tilted by an angle of 30°, which is assumed to be a straight gaze (see Figure 4.5). To accomplish this, we need to fit a line to the nose bridge. In order to create a vertical profile of the face (Figure 4.5a), we project all 3D points within a distance of $5mm$ onto the symmetry plane (black dots). After outliers are filtered, we resample the profile with a y-distance of $1mm$ (yellow markers). After gaussian filtering we get a smooth height profile of the face (green line). Next we fit a line to the nose bridge with the Random Sample Consesus (RANSAC) algorithm (see [10]). Figure 4.5b shows the construction of height profiles from the resampled points for several subjects. We see a lines fitted through each nose bridge. Figure 4.5c and 4.5d show the profiles before and after the alignment to a tilt angle of 30° (magenta-colored line).

4.3.4 Estimation of the initial head pose

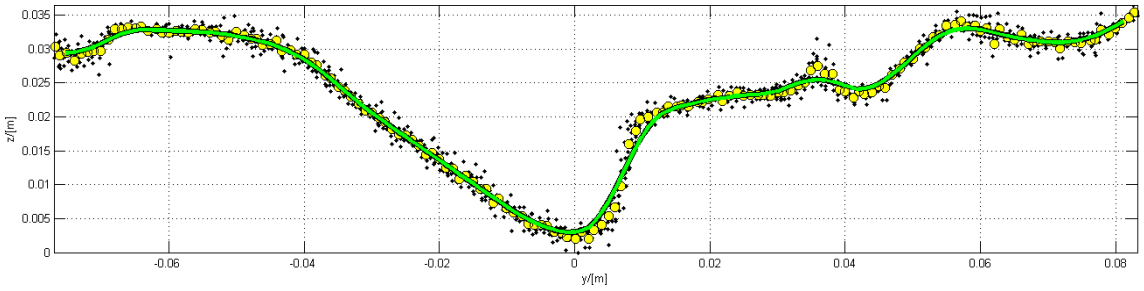
After we have aligned models of all subjects, we can register the first frame of each *ToF* sequence. We compute a point cloud from the first *ToF* frame and detect the nose tip at the depth minimum. With the nose tip as a center point, we limit the point cloud to a sphere with a radius of $8cm$. To avoid that the *ICP* algorithm only converges to a local minimum, we do a pre-alignment step. The nose tip is moved to the origin and the point cloud is registered to the xy-plane, to reach a frontal view. Then a transormation H_{ICP} is computed by registering the pre-aligned *ToF* point cloud to the *SL* point cloud. The initial head pose is the given by:

$$H_{initial} = H_{ICP} \cdot H_{preAlign}$$

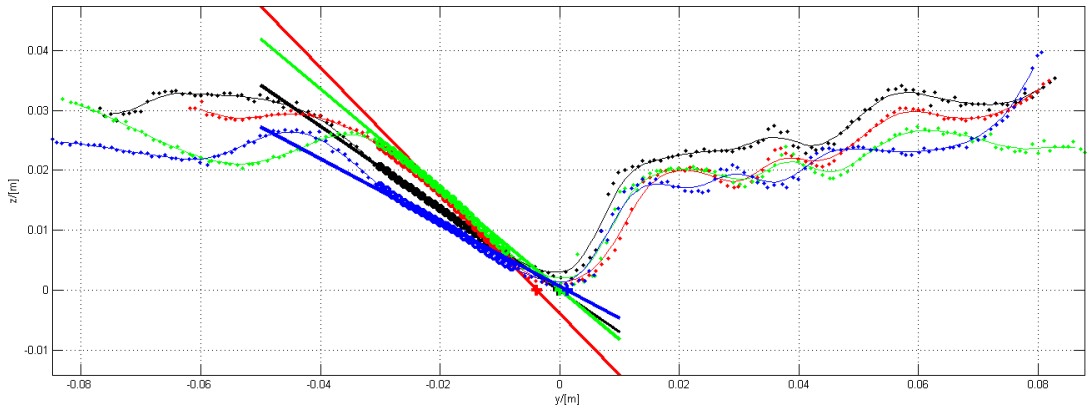
and the corresponding Euler angles can be computed (see Section 3.2.4). The initial translation is given by the upper-right 3×1 submatrix of $H_{initial}$.

4.3.5 Registration of all frames of a ToF sequence

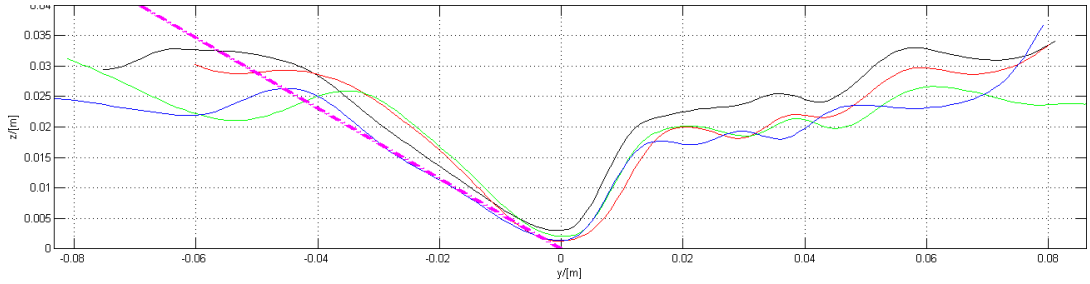
In Figure 4.3 we see, how a head pose is related to the reference frame at the nose tip. Previously, we located the point cloud of the *SL* model with a straight gaze at this position. With *ICP* registration, we want to estimate the rigid-body transformation (see 3.2.2)



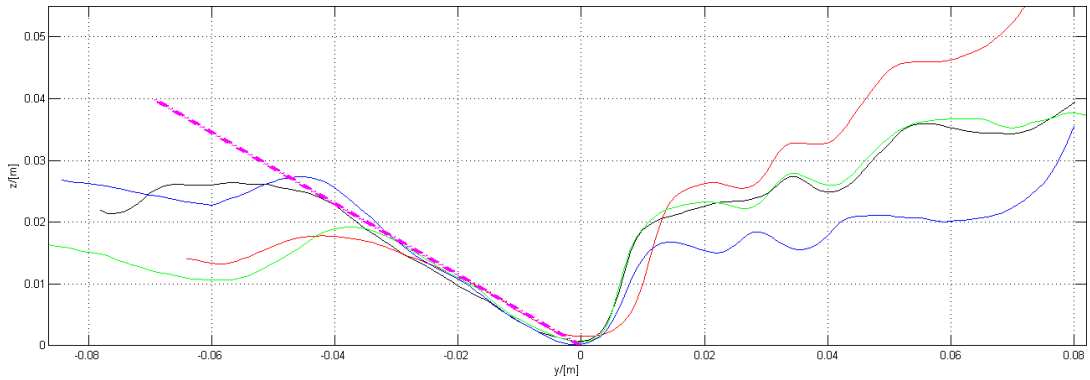
(a) Vertical profile of the face at the symmetry plane.



(b) Fitting of lines through nose bridges of several subjects.



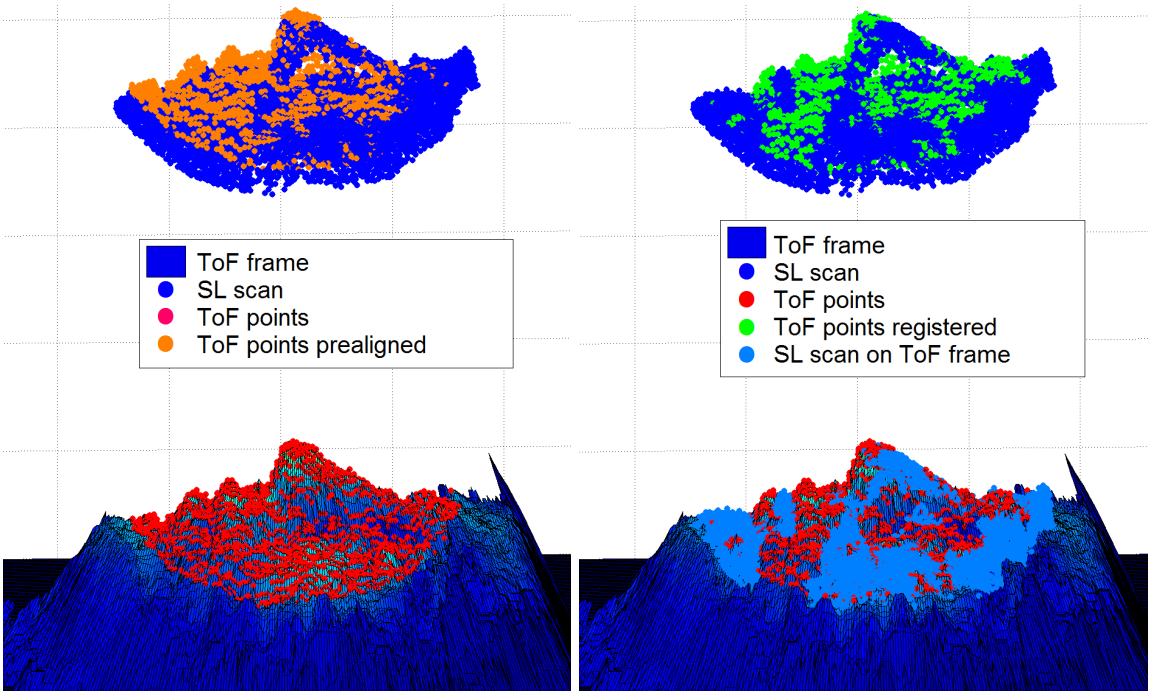
(c) Profiles before the alignment to a tilt angle of 30°.



(d) Profiles after the alignment to a tilt angle of 30°.

Figure 4.5: Straight gaze alignment: Nose bridges must be tilted by 30°.

between each head pose (given by a *ToF* point cloud in space) and the reference pose (given by the *SL* model). Based on the assumption, that a human can only make limited head movements during two subsequent *ToF* frames, we make use of the inverse rotation transform proposed in [21]. The name results from the fact, that the inverse registration of the previous frame can always transform the *SL* model nearly to the *ToF* point cloud of the current frame. Instead of computing the inverse, we directly apply the previous registration result (beginning with $H_{initial}$) to the current *ToF* point cloud and transform it to the *SL* model. When the temporal condition is met, this guarantees a small deviation of the point clouds and the convergence of the *ICP* algorithm in the global optimum. Otherwise, the algorithm would probably find only a local optimum, when the deviation of the current head pose from the reference pose is too large. A final overview on the registration process is given in Figure 4.6.



(a) Prealignment step: From a ToF frame, a point cloud (red) is extracted and prealigned (orange) with the SL scan's point cloud (blue) with the nose tip at the origin.

(b) Registration step: We register the prealigned ToF point cloud (green) and get a transformation. We can now also inversely transform the SL scan onto the ToF frame.

Figure 4.6: Registration of a ToF point cloud at the SL model (reference pose).

4.4 T3M method: Eyes and nose template matching

4.4.1 Motivation

Next, we analyze a very intuitive method for head pose estimation, based on facial feature tracking. In [27], this approach is applied with a stereo-vision camera system. The main idea is to estimate a rigid-body transformation from point correspondences between two views of the head. With template matching and depth information supplied by the *ToF* camera, we are able to compute 3D positions of facial features. We need to track three templates, which must be located at relatively stable facial feature points.

Because template matching completely relies on the appearance of an object, problems can arise, e.g. if we track the eye region and the eye lid gets closed. This would lead to a strong deviation of the template from the true facial feature point. To overcome this problem, we additionally match the corresponding depth patch. In this way, two facial feature points can better be distinguished from one another. Furthermore, the method gets more robust, because the template matching orients on the geometric structure of the face.

From the nature of template matching, increasing deviations from the true template location arise at strong rotations of the head. This is because only a 2D similarity measure is computed, though the face undergoes a 3D transformation. However, we will focus our comparison with the other methods on the target range of our application, which is only a few centimeters around the initial nose tip position.

4.4.2 Overview

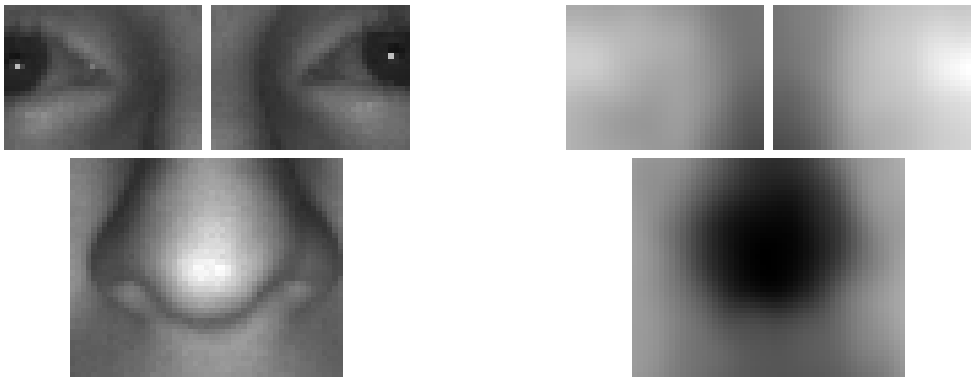
The template matching approach can be subdivided into the following parts:

- before its use, a *ToF* frame is preprocessed (Section 4.2.3)
- initialize eyes and nose templates in the first *ToF* frame
- match templates in every following frame
- in case of template loss: fallback function
- otherwise: compute a head pose

4.4.3 Template initialization

At the beginning of a *ToF* sequence, the subject looks straight into the camera. If we track three facial features, we get enough point correspondences to compute a rigid body transformation between two frames (Section 3.2.2). We empirically found, that the left and right inner eye corners and the nose tip are the most stable facial features in variation of location and appearance. We extract tracking templates at these locations in the first frame (see Figure 4.7). The inner eye corners must be selected manually and the nose tip

can be selected automatically by finding the depth minimum. To increase robustness, the templates not only consist of an infrared (4.7a) but also a depth patch (4.7b). For reliable matching sufficient detail surrounding a facial feature has to be covered. Ideally, only a rigid region should be chosen as template. As a consequence, the eyebrows are excluded from the eye templates to get more accurate results. The template size is a compromise between small and large faces and we choose an average size, which unrestrictedly fits for all subjects.



(a) Infrared patches. **(b)** Depth patches.

Figure 4.7: Eyes (37 × 27 pixels) and nose templates (51 × 41 pixels).

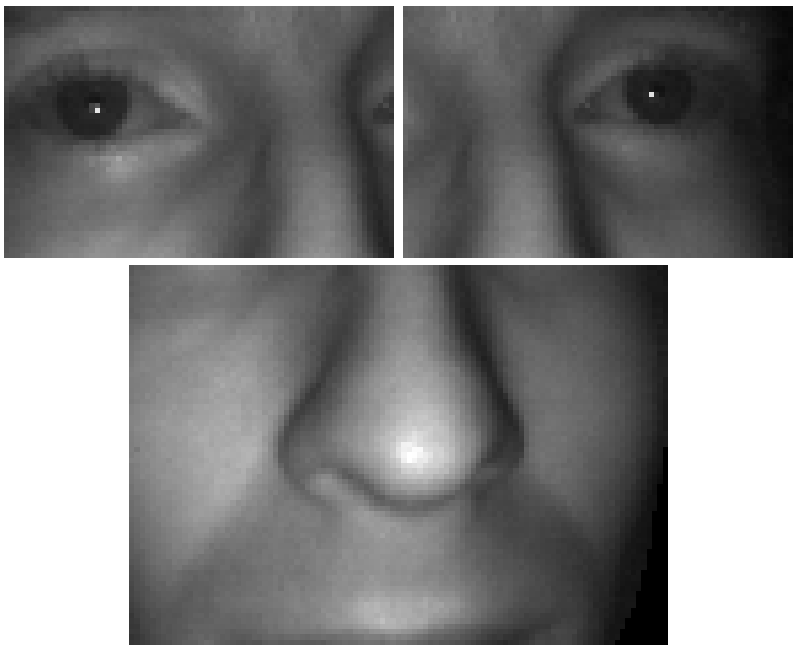
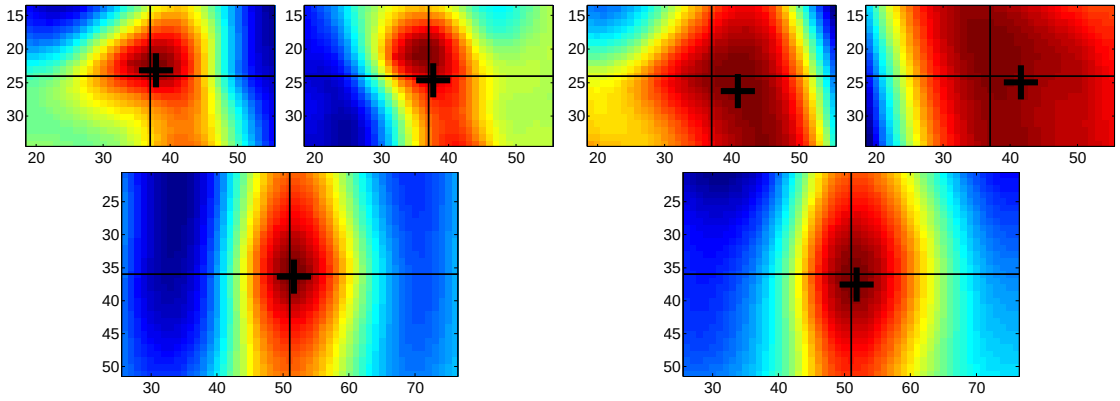
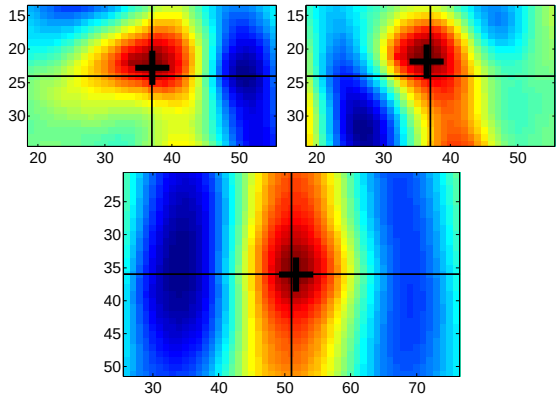


Figure 4.8: Example infrared search images (73 × 47 and 101 × 71 pixels). The result for this example frame can be found in Figure 4.10b.

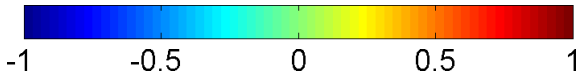


(a) Matching results for image patches.

(b) Matching results for depth patches.



(c) Combined matching results.



(d) Colorbar.

Figure 4.9: Normalized cross correlation results (the axes give the pixel positions in the search images, see Figure 4.8) of (a) infrared patches and infrared search images, (b) depth patches and depth search images and (c) both combined. The center of a correlation image is at the center of the corresponding search image. The new template positions are given by the bold markers in (c). The results are arranged like the eyes and nose templates in Figure 4.7. (d) shows a colorbar for the template matching scores in the range [-1,1], whereby 1 means maximum similarity.

4.4.4 Template matching

After the tracking templates have been initialized in the first frame, they are matched in every following frame. A limited search area around each template's old location is computed (see Figure 4.8; marked by white dotted bounding boxes in the results, see Figure 4.10), with about two times the template's extent. Besides from computational efficiency, a confusion with other prominent parts of the face is avoided in this way, e.g. inner eye corners with mouth corners. Inside each search image the OpenCV template matching

function based on the normalized correlation coefficient (see Section 3.4) is applied for both, the template's infrared and depth patch. Correlation is computed by sliding the patch over the search window. We only get valid results in a region where both completely overlap:

$$size(NCC) = size(search\_image) - size(patch) + 1.$$

In Figure 4.9 this is shown for a sample frame, in which the head pose has changed in relation to the initial frame. The matching results of the infrared (Figure 4.9a) and depth patches (Figure 4.9b), are then averaged to a common matching score (Figure 4.9c). The new template locations are estimated in subpixel accuracy, using a score dependent weighted average of all pixel positions above a threshold:

$$location_{template} = \frac{[\sum_{x,y,} x \cdot score(x,y), \sum_{x,y} y \cdot score(x,y)]}{\sum_{x,y} score(x,y)},$$

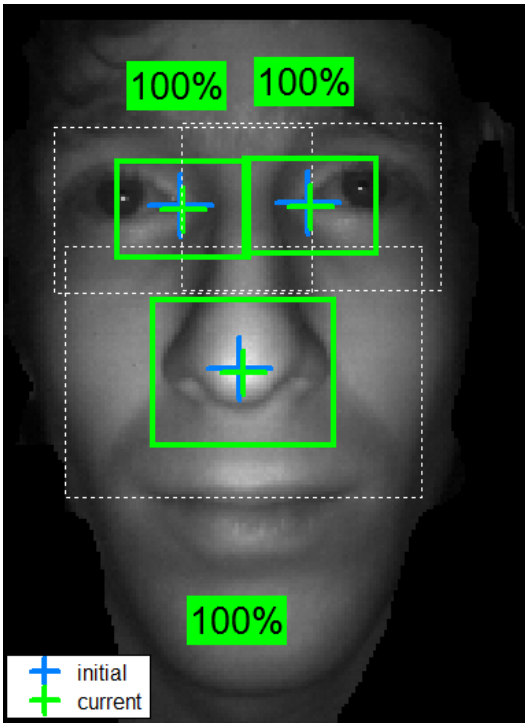
$$\forall x, y : score(x,y) > 0.8 \cdot max(score).$$

The template matching score is in the range $[-1, 1]$, whereby 1 means maximum similarity. If one of the templates reaches a score lower than 0.8 (which is equivalent to a threshold of 90%), a fallback function is executed. The template is searched at the initial template location and the search area is increased to three times of the template's extent. Only after the threshold is exceeded again, the result is accepted as correct match.

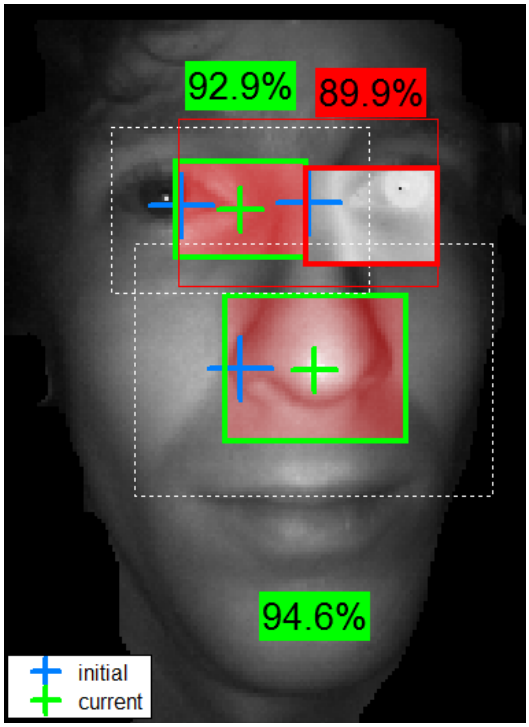
4.4.5 Head pose computation

From the 2D template locations we compute 3D feature points, if all three templates were found correctly in the current frame (no fallbacks). We use the point correspondences - the current and the initial 3D feature points - and build a linear equation system to estimate the best fitting rigid-body transformation (Section 3.2.2). In this way we get a rotation and a translation, which connect the current to the initial head pose.

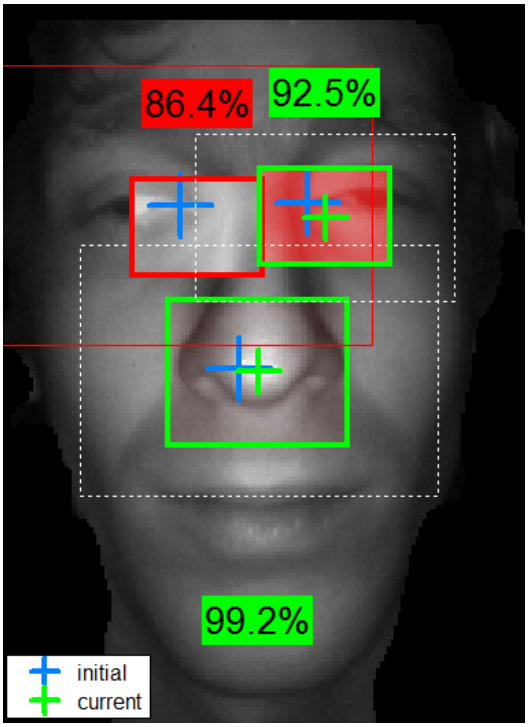
In Figure 4.10 we see template matching examples. The templates are marked by solid bounding boxes and are surrounded by their search areas (white dotted bounding boxes). If a fallback occurs, the bounding box of the affected template changes from green to red. The matching scores are given in percent. The more the scores decrease, the more the template region discolors to red. If the score drops below 90%, the region is shown in inverted grey values. The blue and the green plus markers show the initial and the current template locations.



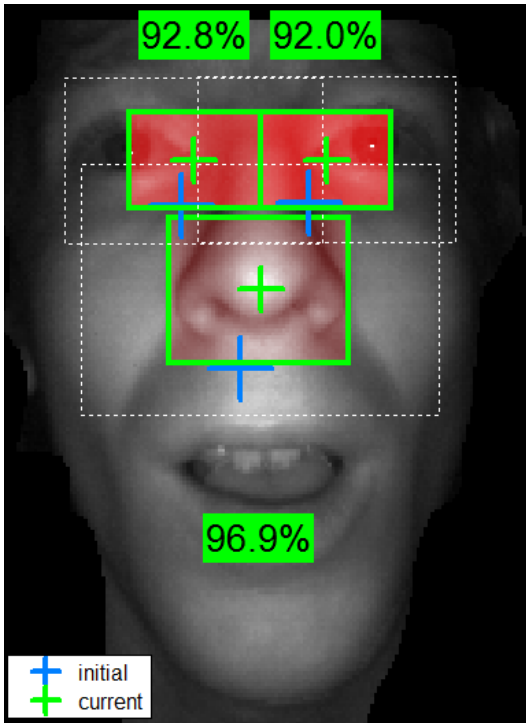
(a) Initial straight head pose.



(b) Head turned sideways.



(c) Eyes nearly closed.



(d) Looking upwards.

Figure 4.10: Example frames with annotated matching results.

4.5 SIP method: Topographic analysis of the face

4.5.1 Motivation

The sparse texturing of the face and also the high movability of facial features make it very difficult to estimate a head pose with high accurracy. Therefore, we want to try a geometric measurement approach which is appearance-independent. In an experiment we search for a stable feature of the face (see Figure 4.11). In comparison to other facial features (e.g. eye brows, mouth corners), the nose tip shows out to be relatively stable and due to its central and elevated position optimally suited for our purposes. The inner eye corners are also stable and bounded within the eye region, but are not as easy to detect as the nose tip during head rotations. We adapted a very sophisticated technique from [13] and [14], with the main idea of performing spherical intersections. These allow us to analyze the facial structure and estimate a head pose at low computational effort. We want to find out if the dependence on only a single facial feature point brings us advantages in robustness over the methods so far.

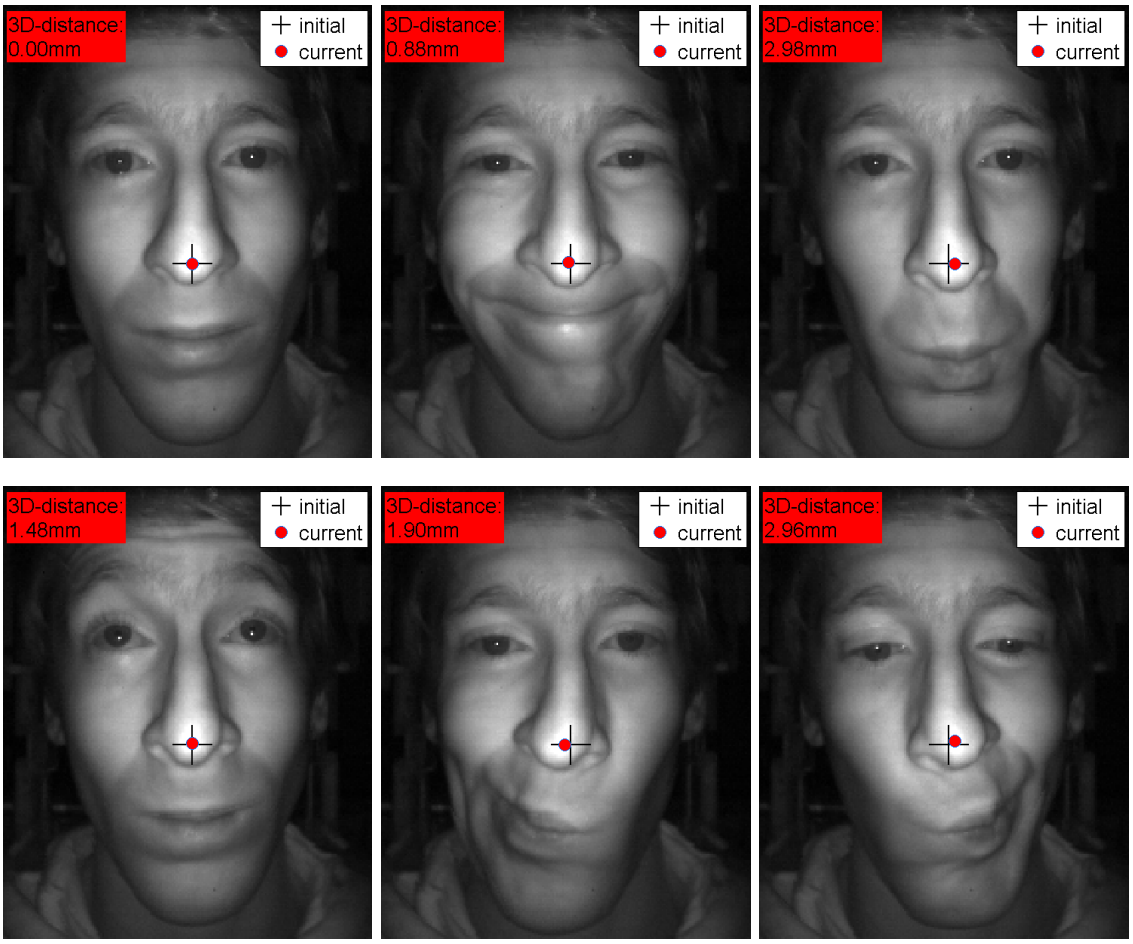


Figure 4.11: An experiment on the stability of the nose tip.

4.5.2 Overview

We can roughly divide the method into the following steps:

- preprocessing: filtering, find a *ROI* (Sections 4.2.3 and 4.5.4)
- detect the nose tip and compute *SIPs*
- estimate the midpoints of the *SIPs*
- get the orientation of the face by fitting a line
- current head pose: position of the nose tip and orientation of the face

4.5.3 Topography

Topography is the measurement of surface shape and features of the earth, e.g. the elevation of a mountain. If we take a look at Figure 4.12 we see, that the human face can be analyzed in the same way. The highest elevation is the nose tip and surrounding facial features are nearly regularly descending in height. In topography, contour lines (black lines in the figure) are used to get an impression of the terrain slope. Later we explain the concept of *SIPs* centered at the nose tip (colored lines), which are better suited for computing the face orientation (also shown in the top view of the face in Figure 4.14d).

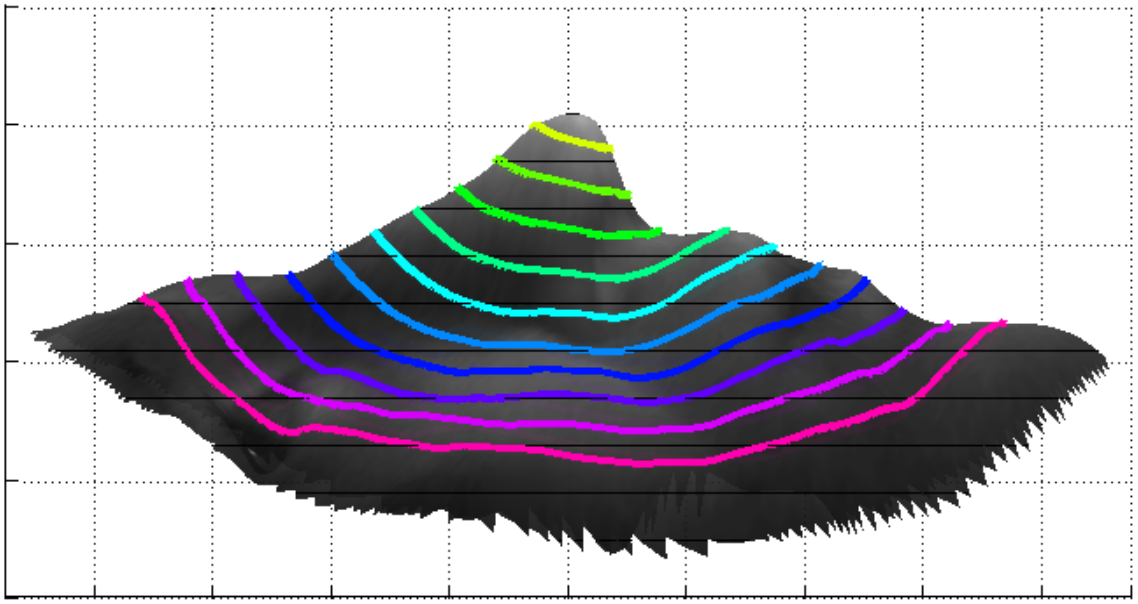
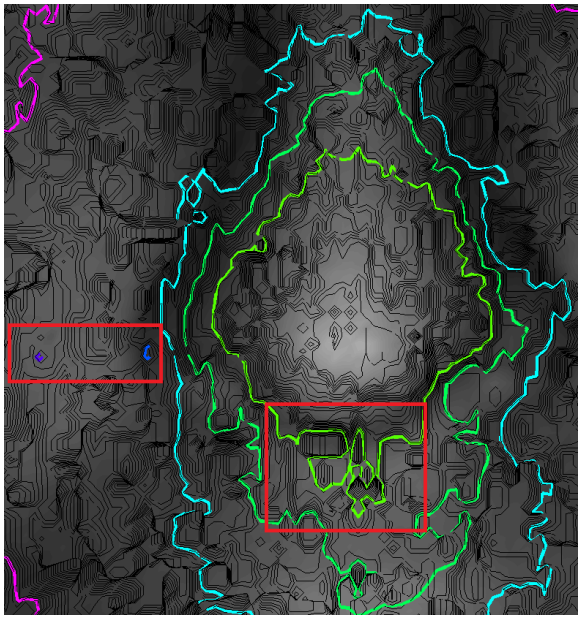


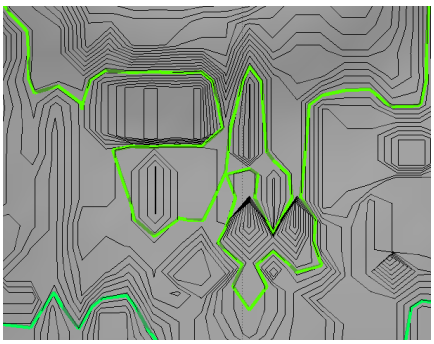
Figure 4.12: Comparison of contour lines (black) and SIPs (colored).

4.5.4 Preprocessing

At the beginning, the *ToF* depth map is median and gaussian filtered, with kernel sizes $m \times m$ and $g \times g$. The filter kernels are decomposed for time efficiency reasons (see Section 4.2.3). In Figure 4.14 the effect of different filter kernel sizes on the contour lines (black) and the results of the *SIP* algorithm (colored lines) is compared. In the top row we see the result of pure median filtering. With a small kernel size the *SIPs* get quite mazy. A bigger kernel size improves this, but still some local extrema remain, which is illustrated by the contour lines. In the bottom row we see the positive effect of additional gaussian filtering. In our system we choose the kernel sizes given in Figure 4.14a, which leads to a smooth surface of the face.



(a) Mazy profile lines.

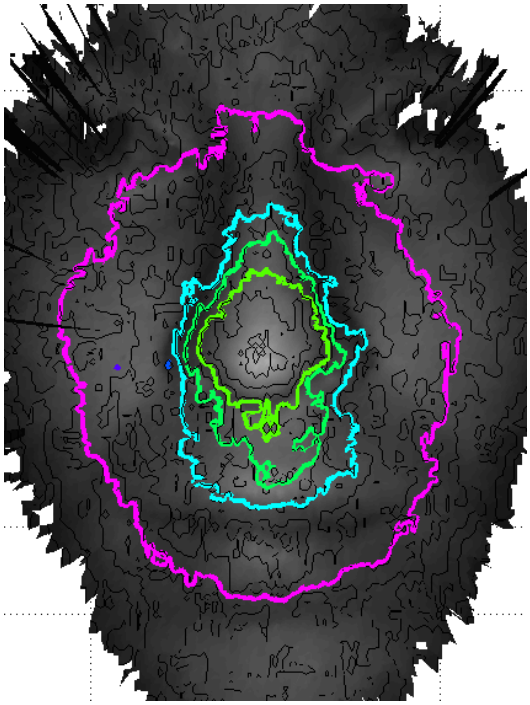


(b) Bypassing of obstacles.

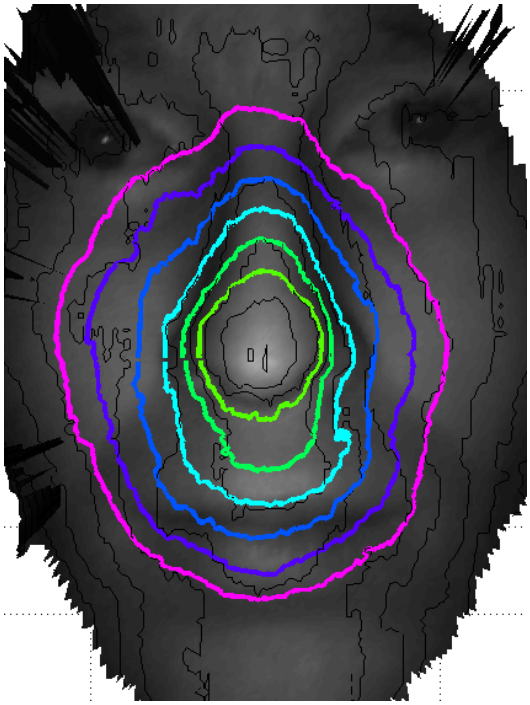
(c) Trapped in local extrema.

Figure 4.13: (a) Insufficient filtering of the depth map. (b), (c) Detail views.

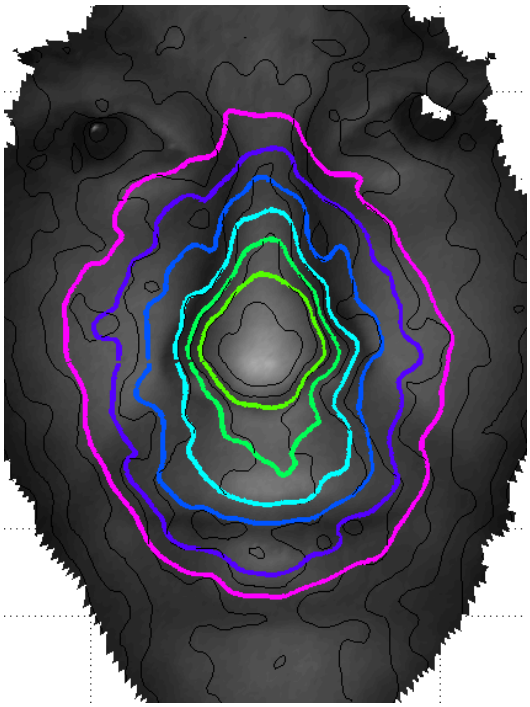
In Figure 4.13 we see an image detail from Figure 4.14a, displayed with contour lines of higher density. Further enlarged views of the image details highlighted by the red boxes are given. From this examples we recognize, that sufficient filtering of the *ToF* depth map is essential for the correct operation of the *SIP* algorithm. Otherwise, the determination of an *SIP* can be compared to finding a way through a maze and a lot of obstacles have to be bypassed in the height profile of the face (see Figure 4.13a). It gets even worse when when we start the computation in a local extrema (e.g. an outlier in the depth map), because then the resulting *SIP* is degenerated or incomplete (see Figure 4.13b). To sum up, one can say that insufficient filtering leads to a higher time consumption during the computation of the *SIPs* and will only lead to an inaccurate head pose.



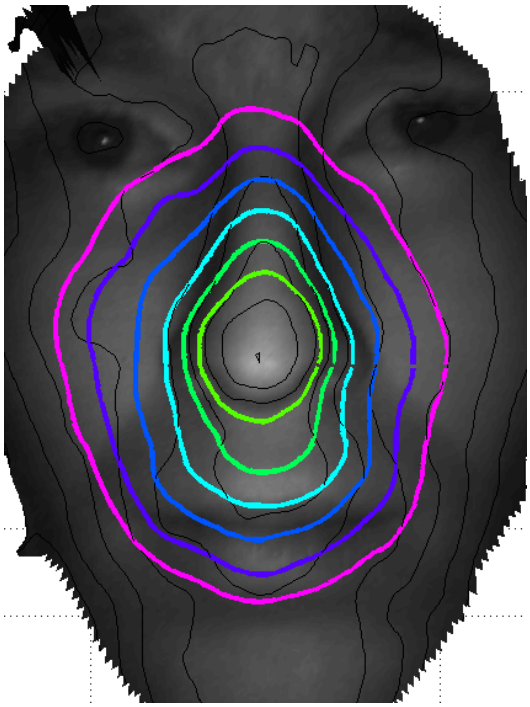
(a) $m = 5$, $g = 0$.



(b) $m = 15$, $g = 0$.



(c) $m = 5$, $g = 7$.



(d) $m = 15$, $g = 7$.

Figure 4.14: Choice of parameters for median and gaussian filtering.

4.5.5 Nose tip detection

After the facial *ROI* is determined (see Section 4.2.3), we need to find the nose tip in every new frame. This could be established by detecting the highest curvature in the face, but in our specific situation the nose tip always is at a depth minimum (near to the center of the image) and we just have to consider proximity to the camera. We can again use the fact, that a human is only able to make limited head movements during two subsequent *ToF* frames to ensure the correctness of the nose tip location.

4.5.6 Spherical intersection profiles

The core concept of this method is to find a number of *SIPs* centered at the nose tip. They are related to contour lines of a mountain, with the sole difference that the intersections are caused by spheres and not by planes (see Figs. 4.12 and 4.15). In this way we reach the advantage, that the resulting intersections become independent of the orientation of the face.

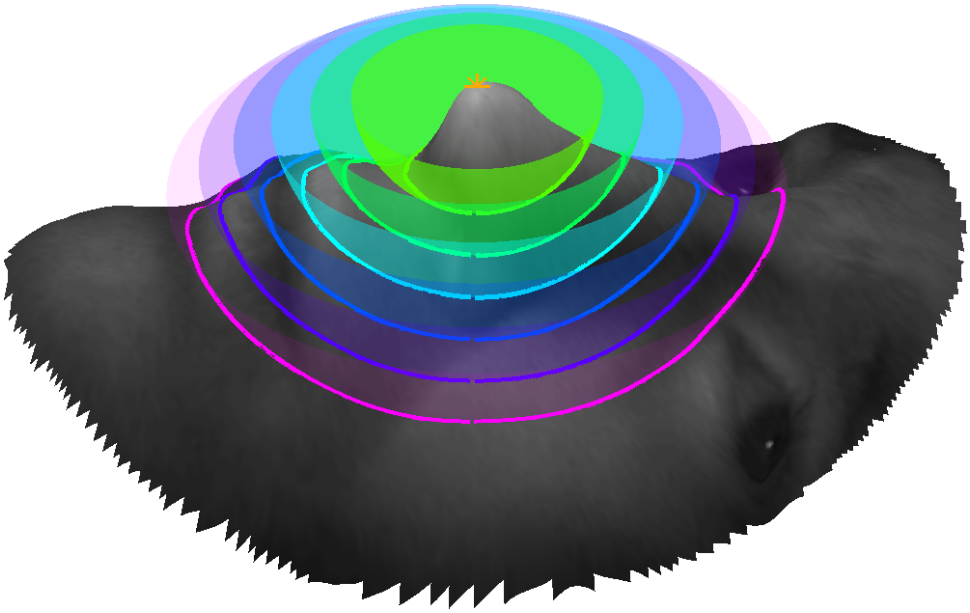
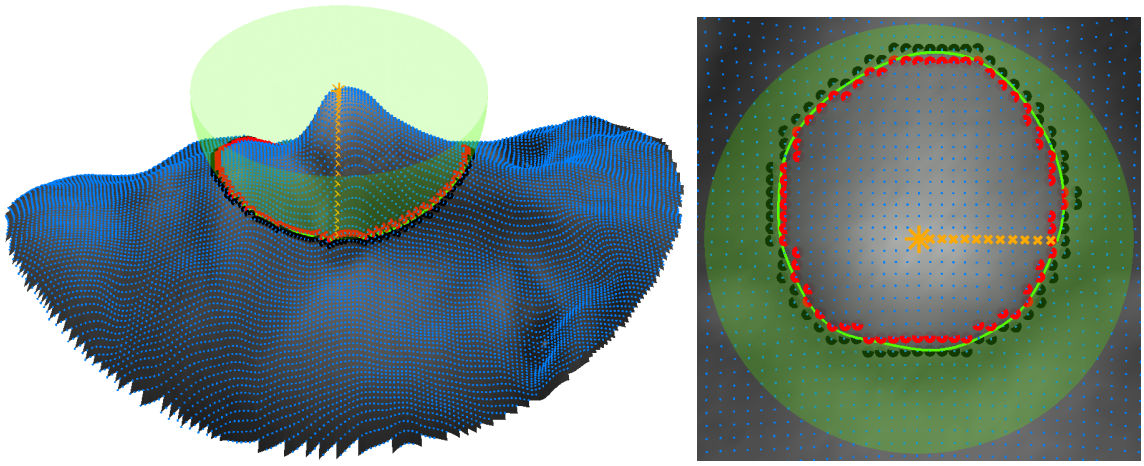


Figure 4.15: SIP construction: Multiple spheres centered at the nose tip.

Algorithm 2 gives an efficient way of computing an *SIP* based on the depth map (see Figure 4.16b). Starting from the nose tip, we advance (orange crosses) in direction of the face centroid until we reach the last point within a sphere of a specific radius r. We want to build a point set lying exactly at the sphere boundary, representing the intersection with the face mesh. Thus we interpolate between the inner (red) and outer (black) points lying nearest to r. If the face centroid was to the right of the nose tip before, we follow the sphere boundary a full round in upward (otherwise in downward) direction until the intersection profile is complete (line between red and black points).



(a) 3D view. **(b)** Depth map (top view).

Figure 4.16: SIP examples: find inner and outer points to interpolate an SIP.

Algorithm 2 Compute 3D intersection of face and sphere (radius r) at nose tip, based on 2D depth map traversal (algorithm adapted from [13]).

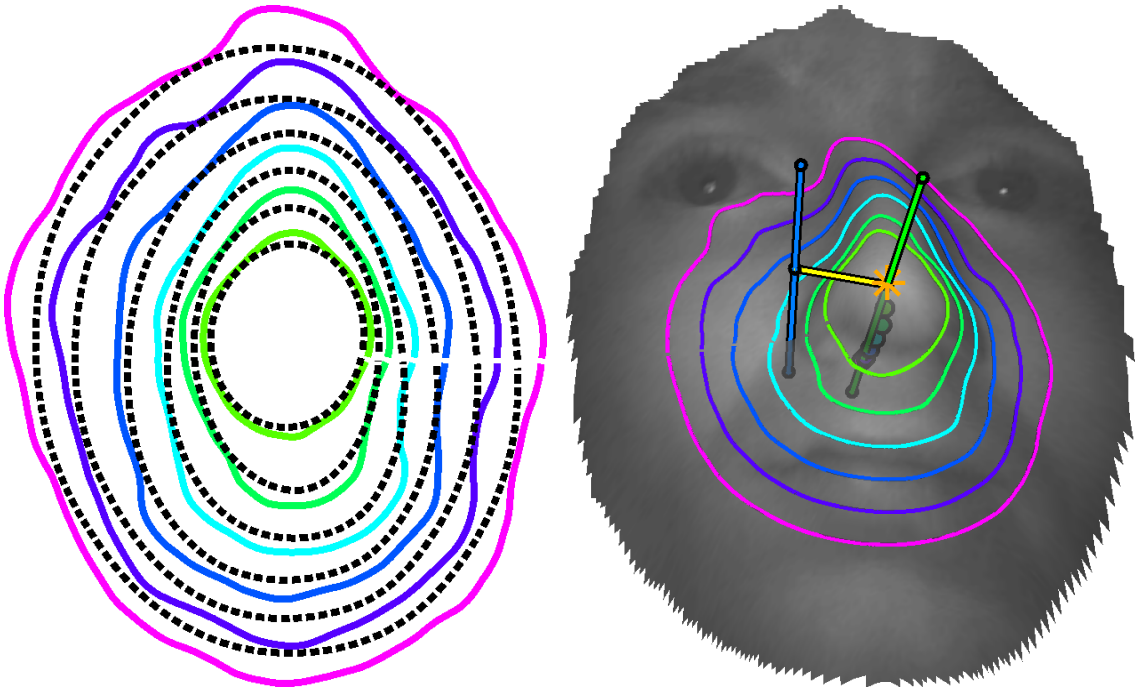
procedure SIP(*depthMap*, *ROI*, *nose*, *sphere*)
  $(i_r, i_c) \leftarrow (nose.row, nose.col)$ ▷ Start: nose tip in the depth map
  $inner.pt3D \leftarrow nose.pt3D$ ▷ 3D point from depthMap projection
  $found \leftarrow$ **false** ▷ try to find intersection
  if $faceCentroid.col < nose.col$ **then**
    $boundaryDirection \leftarrow L_{\Leftarrow}$, $direction \leftarrow D_{\Downarrow}$
  else
    $boundaryDirection \leftarrow R_{\Rightarrow}$, $direction \leftarrow U_{\Uparrow}$
  end if
  while (i_r, i_c) in *ROI* **and** $distance(inner.pt3D, nose.pt3D) < r$ **do**
    $(i_r, i_c) \leftarrow translate(i_r, i_c, boundaryDirection)$ ▷ $L_{\Leftarrow}$ or $R_{\Rightarrow}$
    $outer.pt3D \leftarrow project3D(depthMap, i_r, i_c)$
    if $r < distance(outer.pt3D, nose.pt3D)$ **then**
      $found \leftarrow$ **true** ▷ intersection found
    else
      $inner \leftarrow outer$
    end if
  end while
  if $found \neq$ **true then**
    return No intersection
  end if

22: **repeat** ▷ *direction*: $U_{\Uparrow}$, $L_{\Leftarrow}$, $D_{\Downarrow}$, or $R_{\Rightarrow}$
23: $(i_r, i_c) \leftarrow translate(inner.row, inner.col, direction)$
24: $inner_2.pt3D \leftarrow project3D(depthMap, i_r, i_c)$
25: $(i_r, i_c) \leftarrow translate(outer.row, outer.col, direction)$
26: $outer_2.pt3D \leftarrow project3D(depthMap, i_r, i_c)$
27: **if** $inner_2.pt3D$ **or** $outer_2.pt3D$ invalid **then**
28: **break**
29: **else if** $r < distance(inner_2.pt3D, nose.pt3D)$ **then** ▷ turn left
30: $outer \leftarrow inner_2$
31: **else if** $distance(outer_2.pt3D, nose.pt3D) < r$ **then** ▷ turn right
32: $inner \leftarrow outer_2$
33: **else** ▷ move straight ahead
34: $inner \leftarrow inner_2$
35: $outer \leftarrow outer_2$
36: **end if**
37: $innerDist \leftarrow distance(inner.pt3D, nose.pt3D)$
38: $outerDist \leftarrow distance(outer.pt3D, nose.pt3D)$
39: $t \leftarrow \frac{r - innerDist}{outerDist - innerDist}$ ▷ relative distance to sphere boundary
40: $interpolated.pt3D = inner.pt3D + t \cdot (outer.pt3D - inner.pt3D)$
41: add $interpolated.pt3D$ to SIP
42: $update(direction)$ ▷ move straight ahead or turn left/right
43: **until** SIP complete
44: **return** SIP ▷ spherical intersection profile for radius r
45: **end procedure**

4.5.7 Head pose computation

Now we have given a set of *SIPs*. Initially, the profile lines are running average filtered, which results in the smoothed black lines in Figure 4.17a. Algorithm 3 explains, how we can compute facial symmetry and further a midpoint per profile. The algorithm is based on the assumption, that the roll angle of the head is negligible compared to the pitch and yaw angle (also see Figure 3.6). For each point of an *SIP*, we can thus interpolate a second point on the opposite side of the profile, which is at the same height of the face. In the middle of every point pair lies a symmetry point. By averaging the symmetry points we get a midpoint per profile. The symmetry points of all *SIPs* form a facial symmetry plane. Now, it is easy to fit a 3D line through the nose tip and the midpoints per profile using the *RANSAC* algorithm, which lets us determine the orientation of the face (only pitch and yaw angle). The head pose is then given by the position of the nose tip and the face orientation we have just computed. Figure 4.17b shows an example, where a subject has

its head rotated to the left. The initial head pose is indicated by the blue axis. The green axis is fitted to the *SIP* midpoints, which is the new viewing direction.



(a) Running average filtered SIPs. **(b)** 3D view (angle to z-plane is 60°).

Figure 4.17: Head pose: fit 3D line through nose tip and SIP midpoints.

Algorithm 3 For a set of SIPs, compute facial symmetry and midpoints, which can be used to fit a 3D line (algorithm adapted from [14]).

```
1:  procedure MIDPOINTS({SIP})
2:      midpoints ← ∅
3:      for all SIP ∈ {SIP} do                          ▷ for each profile (point set)
4:          symmetryPoints ← ∅
5:          for all p ∈ SIP do
6:              found ← false                ▷ find other point in SIP at same height
7:              for all q ∈ SIP do
8:                  k ← (index(q) mod length(SIP)) + 1                  ▷ wrap around
9:                  next ← SIP[k]                                       ▷ point after q
10:                 if p ≡ q or p ≡ next then
11:                     continue
12:                 end if
13:                 if q.y < p.y < next.y or next.y < p.y < q.y then
14:                     other ← (q+next)/2                              ▷ take medium point
15:                     found ← true          ▷ height of p between q's and next's
16:                     break
17:                 end if
18:             end for
19:             if found ≡ true then
20:                 add (p+other)/2 to symmetryPoints
21:             end if
22:         end for
23:         add mean(symmetryPoints) to midpoints                           ▷ per SIP
24:     end for
25:     return midpoints                                          ▷ for 3D line fitting
26: end procedure
```

5 Experiments

Contents

In this chapter we make some experiments and compare the performances of the methods we presented in Chapter 4. For this purpose, the inner eye corner and nose tip feature points are hand-labeled in the video sequences to create ground truth data. We use an accuracy measure, which is defined as the percentage of frames lying within a certain error threshold, whereby the distance and angle errors are considered separately. In Section 5.1 this is explained in more detail.

In Section 5.2, we optimize parameters and properties within a method to increase its accuracy.

After that, we make a statistic evaluation to compare our methods with one another in Section 5.3. For this purpose, mean and standard deviation of the absolute errors of the nose tip position and the head orientation angles are computed. Additionally, we determine the quantity of how often the Template Matching (T3M) method fails due to unreliable matches.

We further make an experiment in Section 5.4, to find out how the accuracy of each method depends on facial expressions.

Because the head rests on the backplane of the system during tracking, the head rotation is strongly coupled to the 2D nose tip position. In Section 5.5, we therefore analyze the 2D-dependence of the displacement error.

5.1 Overview

5.1.1 Notation

In the following we use the superscript (f) or (f, m) to indicate that a variable belongs to a specific frame f of a video sequence or a method $m = \{\text{ICP}, \text{T3M}, \text{SIP}\}$. We use $\mathbf{p}_i$ for a 3D feature point from the ground truth, whereby the index i ranges from one to three. With $\{\mathbf{p}_i\}^{(f)}$ for example, we can specify all three feature points of the frame f. We use the hat symbol for feature points $\hat{\mathbf{p}}_i$ that have been determined using a head pose estimation method.

5.1.2 Creating ground truth data

In order to evaluate the performance of a head pose estimation method, we create 3D ground truth data. Each frame of a video sequence shows the face in a different head pose. We label the 2D feature positions of the nose tip and the inner eye corners by hand and compute the corresponding 3D points $\{\mathbf{p}_i\}^{(f)}$ from the depth map. In Section 5.1.4 we will explain, how the results of our methods can be compared with these ground truth points.

5.1.3 The relation of initial and reference head pose

In Section 4.2.1 we defined a reference frame which lies at the origin of the Time of Flight (ToF) camera. We want to relate all head poses in space to this reference head pose. In Section 4.3.3, we thus aligned a Structured Light (SL) face scan of each subject at the reference frame in a way that it shows a straight gaze. The video sequence of a subject does not start directly at the reference pose, but at an arbitrary head pose. With the Iterative Closest Point (ICP) method, the rigid transformation $H^{(1,\text{ICP})}$ can be computed (recall Section 3.2.2), which relates the initial to the reference head pose. Further, all frames can be registered directly to the reference pose in the same way:

$$T^{(f,\text{ICP})} = H^{(f,\text{ICP})}.$$

By using the initial registration $H^{(1,\text{ICP})}$, we can also transform the results of the *T3M* method, which were only computed relative to the initial head pose:

$$T^{(f,\text{T3M})} = H^{(1,\text{ICP})} \cdot H^{(f,\text{T3M})}.$$

In the Spherical Intersection Profile (SIP) method, we gain head pose information based on the facial surface, by computing the pitch and yaw angles, which are related to the z-plane. The roll angle is not taken into account. For the reference pose we defined a straight gaze based upon a nose bridge angle of 30°. In the *SIP* method, the angles are computed without taking care of the nose bridge, so they are not consistent with the

straight face definition. To fix this problem, we transform the results by the inverse of the first frame and include the initial registration:

$$T^{(f,\mathrm{SIP})} = H^{(1,\mathrm{ICP})} \cdot \left(H^{(1,\mathrm{SIP})}\right)^{-1} \cdot H^{(f,SIP)}.$$

In this way, we reach a common relation among the methods, and we are able to compare their results in the statistic evaluation in Section 5.3:

$$T^{(1,\mathrm{ICP})} = T^{(1,\mathrm{T3M})} = T^{(1,\mathrm{SIP})}.$$

For each *ToF* frame, we define the associated head pose as a rigid motion starting from the reference pose:

$$HeadPose^{(f,m)} = \left(T^{(f,m)}\right)^{-1},$$

whereby m is one of the three methods. Finally, we want to remind that the head pose is uniquely defined by six parameters: the x, y and z values of the nose tip and the Euler angles φ, θ and ψ of the face orientation which can be computed from the head pose according to Section 3.2.4.

5.1.4 Definition of distance and angle errors

Here we describe, how a head pose estimation method can be evaluated (adapted from [8]). The error of the six Degrees of Freedom (DOF) of a head pose can be easily measured, if we treat the translational and rotational part separately. Then a distance and an angle error can be defined based on the comparison of point correspondences. We use the error measures from Section 3.5.

The 3D ground truth points of the first frame of a video sequence must be transformed from the initial to the reference pose:

$$\{\tilde{\mathbf{p}}_i\}^{(1)} = H^{(1,ICP)} \cdot \{\mathbf{p}_i\}^{(1)}$$

For a frame f, we now use the result of a head pose estimation method to transform the ground truth points at the reference pose back to the current head pose and get estimated feature points:

$$\{\hat{\mathbf{p}}_i\}^{(f,m)} = HeadPose^{(f,m)} \cdot \{\tilde{\mathbf{p}}_i\}^{(1)}$$

Now we can evaluate one of our methods by computing a distance error of the nose tip feature point ($i = 1$) using the Mean Absolute Error (MAE):

$$MAE^{(m)}_{\mathrm{Nose}} = \sum_{f=1}^{n} \frac{\left\|\hat{\mathbf{p}}_1^{(f,m)} - \mathbf{p}_1^{(f)}\right\|}{n}.$$

For the angle error, we need to compare the estimated and true Euler angles, which we

get from the head pose transformation of each frame (see Section 3.2.4). From a method's head pose we get the Euler angle estimates:

$$HeadPose^{(f,m)} \longrightarrow \begin{pmatrix} \hat{\varphi} & \hat{\theta} & \hat{\psi} \end{pmatrix}.$$

For the ground truth, we have to define the true head poses first. We estimate a rigid transformation $H^{(f)}$ between the ground truth feature points of the current frame $\{\mathbf{p}_i\}^{(f)}$ and the ones of the initial frame $\{\mathbf{p}_i\}^{(1)}$. Using the initial registration, we can further transform them to the feature points at the reference pose $\{\widetilde{\mathbf{p}}_i\}^{(1)}$:

$$T^{(f)} = H^{(1,\mathrm{ICP})} \cdot H^{(f)}.$$

The inverse transformation is then the ground truth head pose of frame f, from which we can compute the true Euler angles:

$$HeadPose^{(f)} = \left(T^{(f)}\right)^{-1} \longrightarrow \begin{pmatrix} \varphi & \theta & \psi \end{pmatrix}.$$

Now we can formulate the angle error using the *MAE*:

$$MAE^{(m)}_{\mathrm{Angles}} = \sum_{f=1}^{n} \frac{\sum_{i=1}^{3} \left|\hat{\alpha}_i^{(f,m)} - \alpha_i^{(f)}\right|}{n} = \sum_{f=1}^{n} \frac{\left|\Delta\varphi^{(f,m)}\right| + \left|\Delta\theta^{(f,m)}\right| + \left|\Delta\psi^{(f,m)}\right|}{n}.$$

We use an accuracy measure for the distance error, which tells us how accurately a method works if a certain distance threshold t is tolerated. It computes the percentage of frames, for which the Absolute Error (AE) between estimated and true nose tip position lies underneath t:

$$Accuracy^{(m)}_{\mathrm{Nose}}(t) = \frac{1}{n} \cdot \sum_{f=1}^{n} \left(1\left[AE^{(f,m)}_{Nose} < t\right]\right).$$

In the same way, we compute an accuracy measure for the angle error:

$$Accuracy^{(m)}_{\mathrm{Angles}}(t) = \frac{1}{n} \cdot \sum_{f=1}^{n} \left(1\left[AE^{(f,m)}_{Angles} < t\right]\right).$$

It tells us the percentage of frames, where the *AE* between true Euler angles and estimates $\left|\Delta\varphi^{(f,m)}\right| + \left|\Delta\theta^{(f,m)}\right| + \left|\Delta\psi^{(f,m)}\right|$ is lower than a certain threshold t.

In Section 5.5 we will use the Mean Signed Difference (MSD)

$$MSD^{(m)}_{\mathrm{Nose}} = \sum_{f=1}^{n} \frac{\hat{\mathbf{p}}_1^{(f,m)} - \mathbf{p}_1^{(f)}}{n},$$

to analyze the 2D dependence of the nose tip distance error. It gives us information about the direction and amount of displacement between estimated and true nose tip position.

5.2 Intra-method optimization

In this section we want to optimize parameters and properties within a method. For the *ICP* method, we try to optimize the filtering in the preprocessing step, to increase the accuracy of the registration results. Later, we want to decide between two approaches for nose tip detection in the *SIP* method.

5.2.1 Filter parameter evaluation for the ICP method

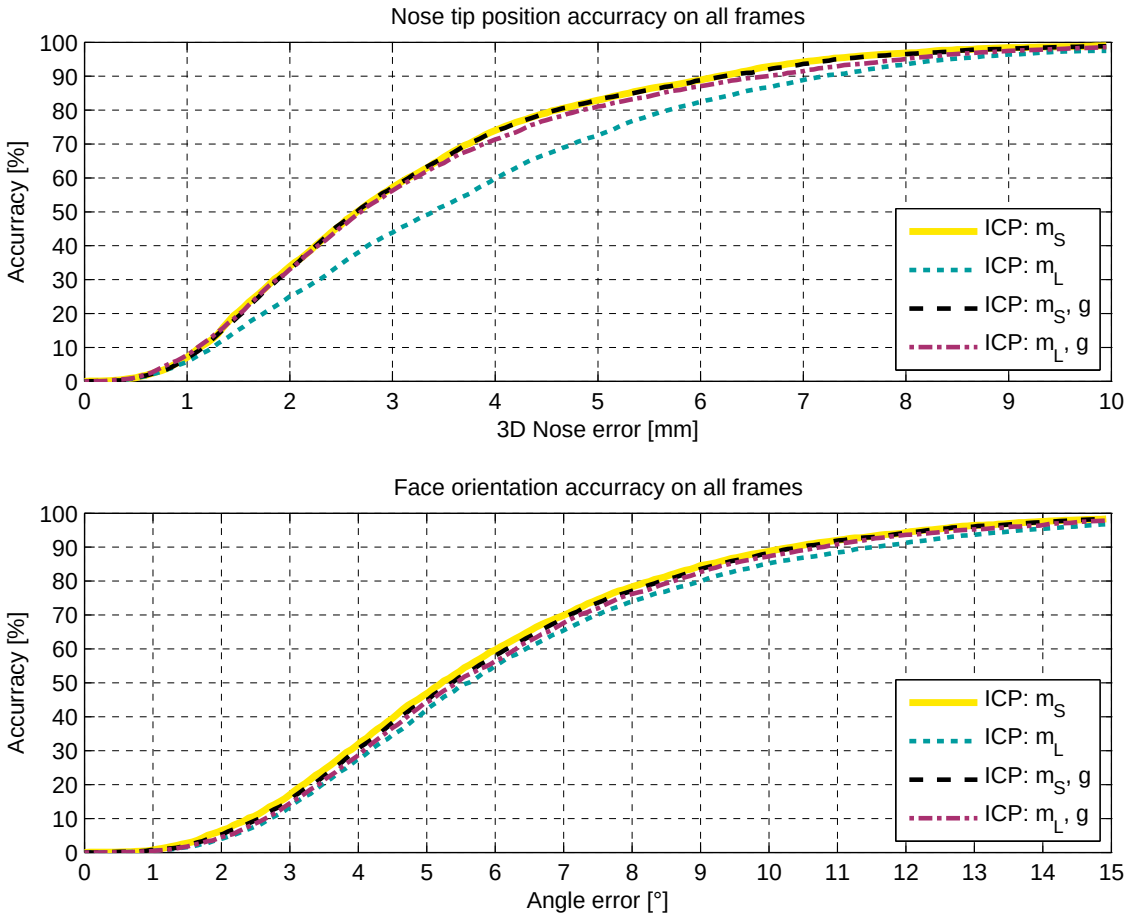


Figure 5.1: ICP method: Accuracy evaluation with various filter kernels.

The use of a *ToF* camera introduces strong noise and outliers on the measured depth data of the face. As already mentioned in Section 3.3, the *ICP* algorithm performs a least squares optimization. The sum of squared distances between data and model points gives a strong weight to outliers. In this way the result is possibly distorted, but the algorithm may also converge against the wrong minimum. Thus, appropriate filtering is essential for accurate operation of our *ICP* method. Nevertheless, it has to be a compromise between sufficient noise reduction and little deformation of the face model. As discussed in Section

4.2.3, the type of filtering depends on the target method. In this experiment we first determined, that at least a 7×7 kernel should be used for the median filter to sufficiently reduce outlying depth values. We tried four filtering possibilities: only a small (7×7) or a large (15×15) median filter kernel, or each of them in combination with Gaussian filtering. We reached the most accurate registration results by just using the small median kernel without the Gaussian filter (see Figure 5.1).

5.2.2 Nose detector evaluation for the SIP method

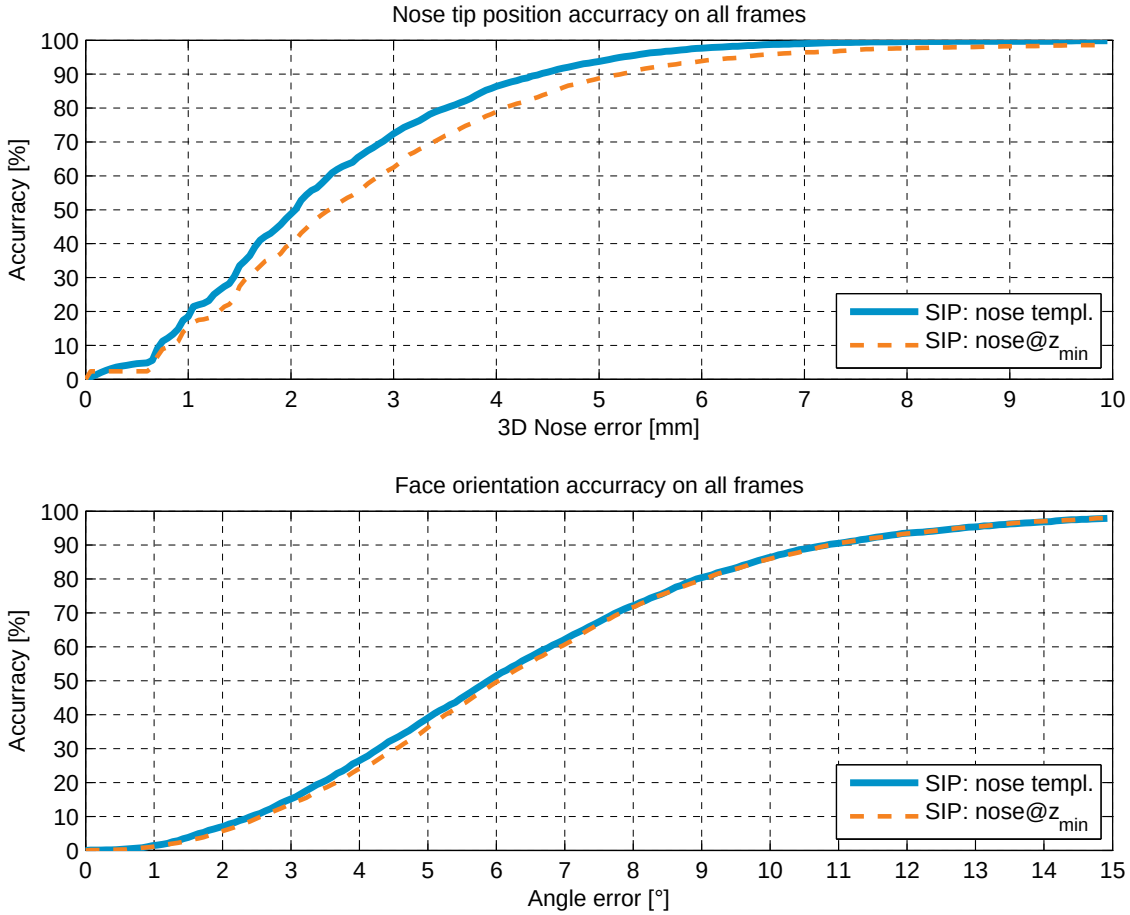


Figure 5.2: SIP method: Accuracy evaluation with different nose tip detectors.

For the *SIP* method it is important, that the position of the nose tip is detected precisely. This increases the accuracy of the subsequent head pose estimation. For this purpose, we want to compare two different approaches and use the better one in our method. In the first approach, the nose tip is assumed to remain exactly at the global depth minimum. This is not the case during head rotations, and a small error arises. The detected position is then a little bit shifted away from the true one, namely into the direction of the camera. This may also be the case in our second approach, in which

we detect the nose tip via template matching. Due to the central and exposed position of the nose tip, a nose template can be matched robustly. In Figure 5.2 we see, that the detection approach using template matching works slightly better. By observing the detection results in a video sequence, we see that in the first approach the detected position jitters around the true nose tip. This is caused by sensor noise of the *ToF* camera and the fact that the nose tip is not a perfect peak. In the second approach, the detection is stabilized by matching a region surrounding the nose tip.

5.3 Inter-method statistical evaluation

In this section, we compare our *ICP*, *SIP* and *T3M* method in a statistic evaluation. We use a set of video sequences acquired from four subjects (see Figure 4.2), with hand-labeled eye corner and nose tip feature points. While inspecting the labeled feature points we made the observation, that especially during head rotations the labeled nose tip positions deviate from the true ones. Even for a human it is particularly challenging to decide where the true position is. Thus, we decided to acquire a sequence of a single subject with small black markers at the feature points (see Figure 4.1), which make labeling much easier. We strictly distinguish unmarked and marked sequences in the evaluation (see Sections 5.3.1 and 5.3.2), which makes it possible to estimate the error which was introduced by imprecise labeling. In Section 5.3.3 the results of both types are compared briefly.

In the following, we compute the *MAE* of the nose tip position and the face orientation angles for each method. The accuracy measures are used to compare the methods in a diagram. In a table and a diagram, the mean and standard deviation of the errors are compared. We make the statistical evaluations on all frames and additionally only inside the working range of the head tracking system. Within a radius of $5mm$ from the initial nose tip position the system should work accurately, outside of this range the treatment devices will have been paused already. In Section 5.1.4, the error definitions can be looked up.

5.3.1 Unmarked sequences

In this section we use hand-labeled data of four video sequences of different subjects without marker points. In the next section we estimate the error which was introduced by imprecise labeling. In Figure 5.3 we see the method's results, on the left side the nose tip and angle accuracy is computed using all frames, on the right side only the ones within the working radius of $5mm$. For method comparison, we have a look at the accuracies at a nose tip distance error threshold of $3mm$ and an angle threshold of $5°$. For all frames, the nose tip accuracy $Accuracy^{(m)}_{\text{Nose}}(3mm)$ is at 63%, 79% and 67%, for the *ICP*, *SIP* and *T3M* method respectively. In the working range, it is 89%, 99% and 93%. The face orientation accuracy $Accuracy^{(m)}_{\text{Angles}}(5°)$ is 56%, 46% and 60% for all frames, and 77%, 63% and 90% within the working range. An overview is also given in Table 5.3.

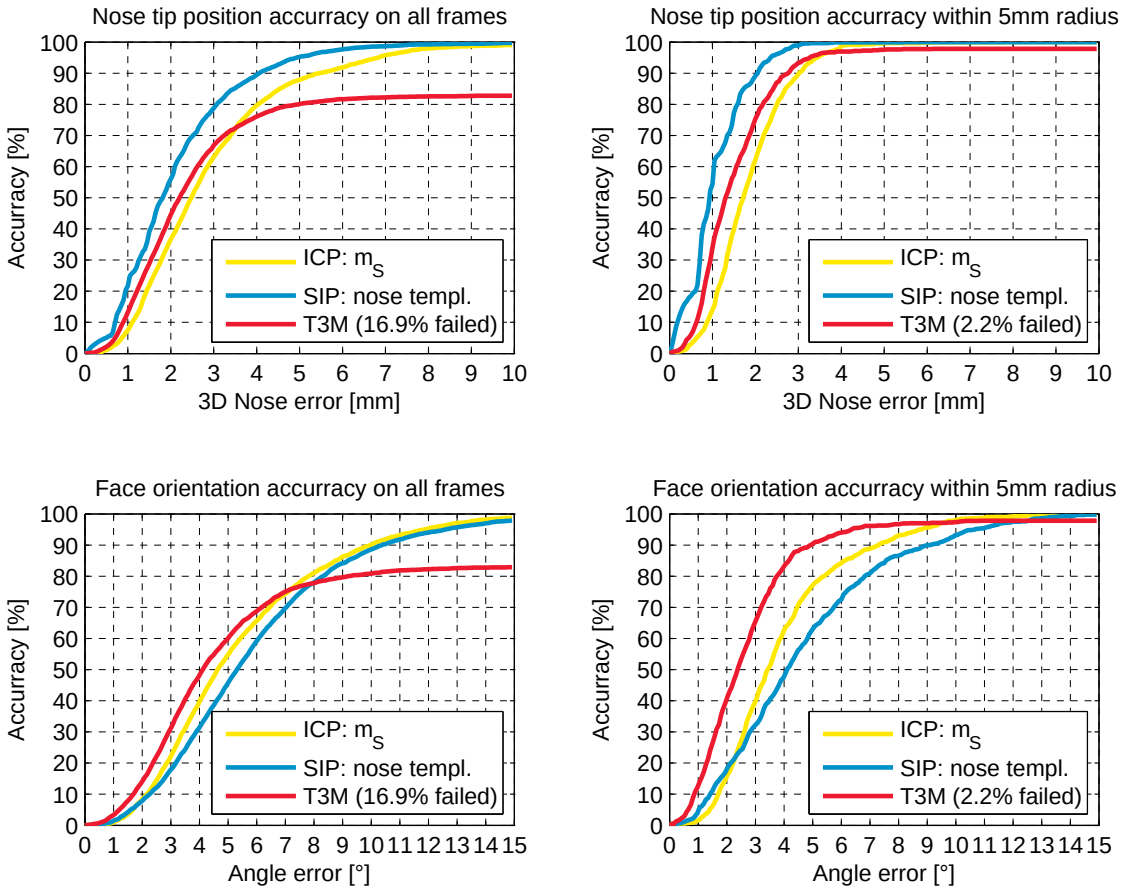


Figure 5.3: Unmarked sequences: Tracking accuracies of nose tip and face orientation.

In Table 5.1 and Figure 5.4 we see a method comparison for the mean and standard deviation of the *AEs* of nose tip position, summed angle, and pitch, yaw and roll angle separately. The mean of the *AE* is equivalent to the *MAE*. Again, we give the results for all frames and then only within the working range of $5mm$ radius of a subject's initial nose tip position. On the right side a fail percentage for the *T3M* method is given, which reduces inside the working range where the method works more robustly. We compare the tracking qualities of the *ICP*, *SIP* and *T3M* in detail in the next section, because the ground truth of the marked sequence is more reliable.

	Nose [mm]	All angles [°]	Pitch [°]	Yaw [°]	Roll [°]	Fail [%]
ICP	2.94 ± 1.94	5.48 ± 3.28	1.93 ± 1.74	1.66 ± 1.27	1.88 ± 1.64	-
SIP	2.14 ± 1.51	5.95 ± 3.44	2.26 ± 2.13	1.58 ± 1.54	2.11 ± 1.47	-
T3M	2.18 ± 1.38	4.05 ± 2.37	1.32 ± 1.09	1.12 ± 0.97	1.60 ± 1.41	16.9

(a) All frames evaluated.

	Nose [mm]	All angles [°]	Pitch [°]	Yaw [°]	Roll [°]	Fail [%]
ICP	1.85 ± 0.88	3.96 ± 2.25	1.10 ± 1.08	1.50 ± 1.14	1.36 ± 1.08	-
SIP	1.06 ± 0.69	4.70 ± 2.95	1.57 ± 1.56	1.43 ± 1.20	1.70 ± 1.33	-
T3M	1.47 ± 0.82	2.60 ± 1.61	0.97 ± 0.74	0.73 ± 0.68	0.90 ± 0.75	2.2

(b) Frames within working range of $5mm$ evaluated.

Table 5.1: Unmarked sequences: MAEs with standard deviation.

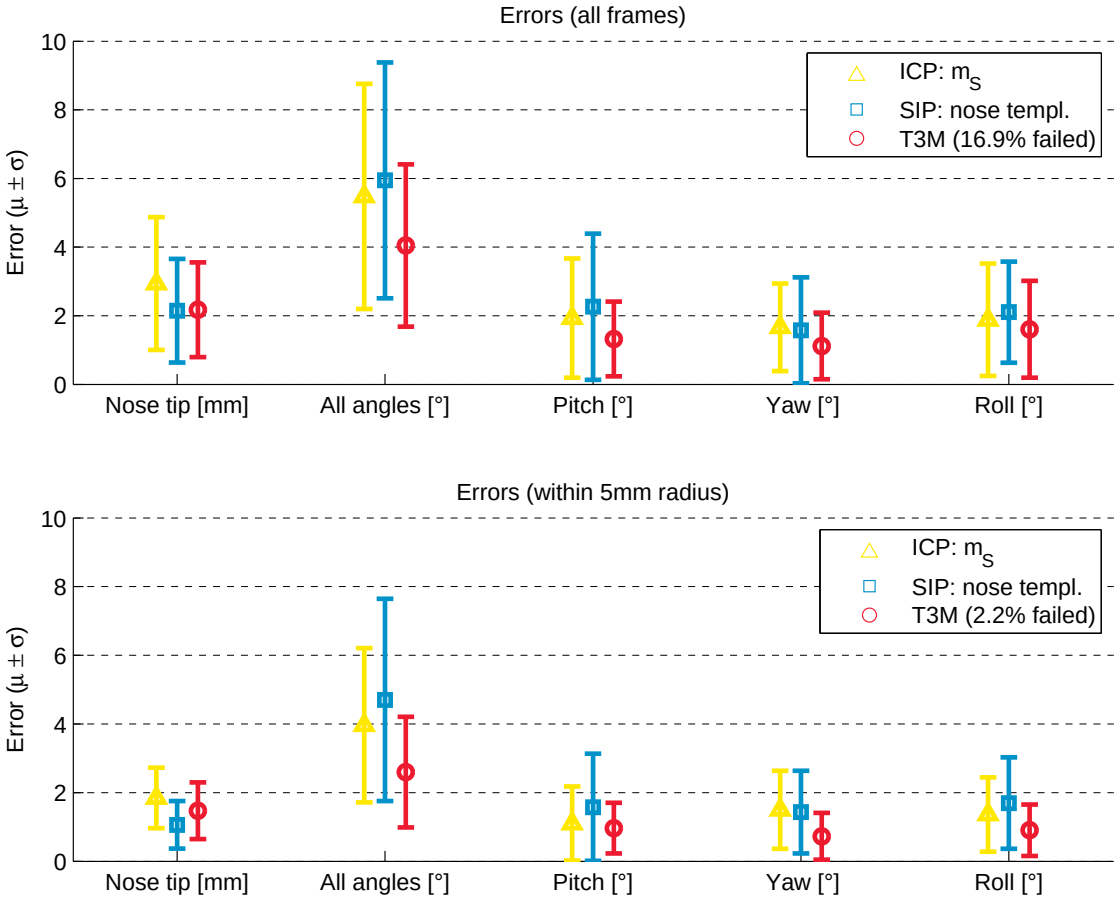


Figure 5.4: Unmarked sequences: Comparison of all method's head pose errors. In the Figure we see bar charts for the distance and angle error, which represent the translational and rotational part of the head pose error, both given as MAE with standard deviation. Below the plots a unit is given next to each label, which corresponds to the error shown on the y-axis. Besides the summed angle error, the rotational components about the single coordinate axes are given (the pitch, yaw and roll angles are intuitively explained in Figure 3.6).

5.3.2 Marked sequence

In this section we use a hand-labeled video sequences of a subject with marker points. The results can be used to estimate the error of the ground truth which was introduced by imprecise labeling in the unmarked sequences of the previous section. In Figure 5.5 we see the accuracy results for the marked sequence. The nose tip accuracy $Accuracy^{(m)}_{\text{Nose}}(3mm)$ for all frames is 68%, 91% and 74%, for the *ICP*, *SIP* and *T3M* method respectively, and 80%, 98% and 85% in the working range. The face orientation accuracy $Accuracy^{(m)}_{\text{Angles}}(5°)$ is 40%, 46% and 49% for all frames, and 44%, 45% and 76% within the working range. An overview is also given in Table 5.3.

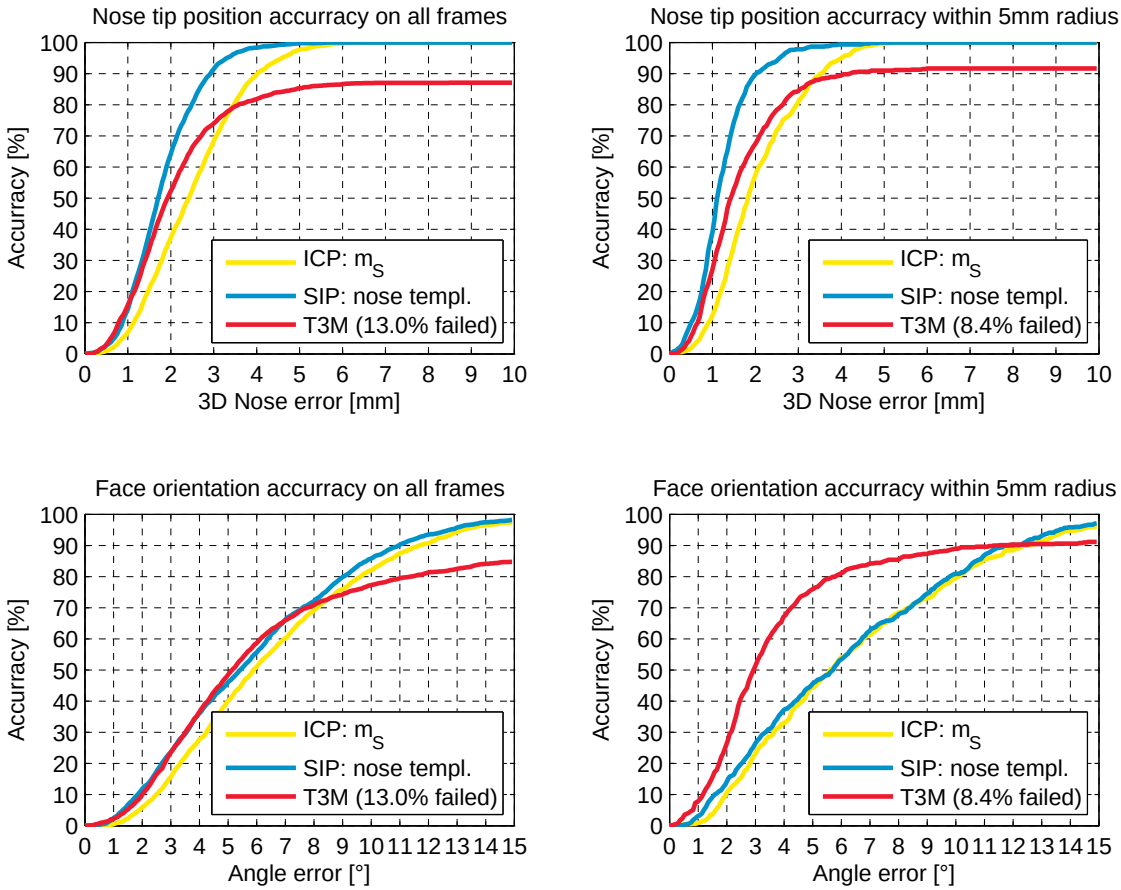


Figure 5.5: Marked sequence: Tracking accuracies of nose tip and face orientation.

From the accuracy curves we see, that the *SIP* method is the most accurate one in estimating the nose tip position. The *T3M* method is generally better than the *ICP* method, but the matching of the eye templates fails at too strong head rotations or tightly closed eyes. This can also be seen from the upper limit of the curves, which is lowered by the fail percentage. At the estimation of the face orientation, *SIP* is slightly better than *ICP*, but the *T3M* method outperforms them both inside the working range of $5mm$.

	Nose [mm]	All angles [°]	Pitch [°]	Yaw [°]	Roll [°]	Fail [%]
ICP	2.51 ± 1.13	6.61 ± 3.69	2.09 ± 1.54	2.07 ± 1.66	2.45 ± 1.95	-
SIP	1.82 ± 0.83	5.99 ± 3.60	2.04 ± 2.02	2.03 ± 1.59	1.92 ± 1.47	-
T3M	1.95 ± 1.10	5.53 ± 4.05	2.12 ± 1.71	1.13 ± 1.10	2.28 ± 2.26	13.0

(a) All frames evaluated.

	Nose [mm]	All angles [°]	Pitch [°]	Yaw [°]	Roll [°]	Fail [%]
ICP	2.04 ± 1.01	6.56 ± 4.18	1.67 ± 1.59	2.19 ± 1.79	2.69 ± 2.09	-
SIP	1.24 ± 0.68	6.28 ± 4.04	2.13 ± 2.49	2.05 ± 1.67	2.09 ± 1.58	-
T3M	1.57 ± 0.92	3.40 ± 2.55	1.23 ± 1.19	0.85 ± 0.83	1.32 ± 1.21	8.4

(b) Frames within working range of $5mm$ evaluated.

Table 5.2: Marked sequence: MAEs with standard deviation.

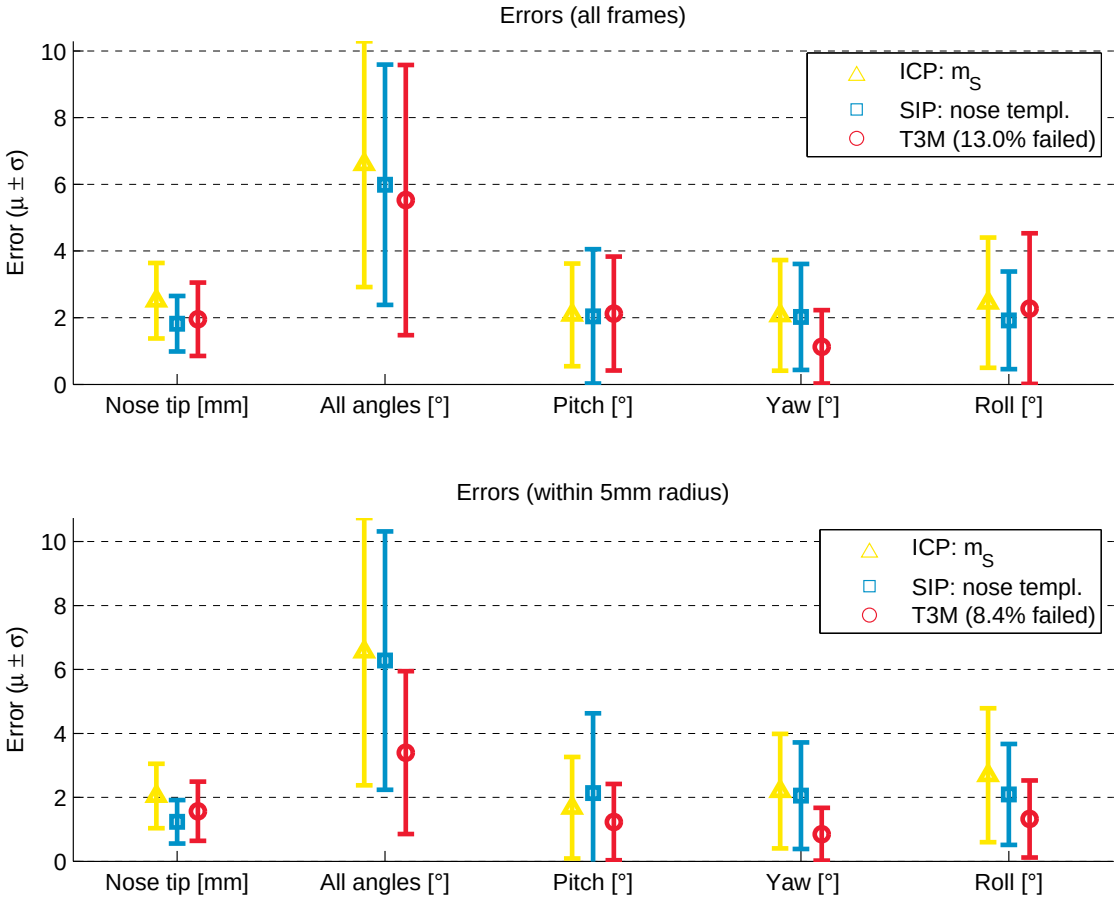


Figure 5.6: Marked sequences: Comparison of all method's head pose errors. In the Figure we see bar charts for the distance and angle error, which represent the translational and rotational part of the head pose error, both given as MAE with standard deviation. Below the plots a unit is given next to each label, which corresponds to the error shown on the y-axis. Besides the summed angle error, the rotational components about the single coordinate axes are given (the pitch, yaw and roll angles are intuitively explained in Figure 3.6).

In Table 5.2 and Figure 5.6 we see the method comparison for the *MAE* and standard deviation of the marked sequence. Especially in the figure we see a good comparison of the methods. For the nose tip, the ordering from worst to best method is *ICP*, *T3M* and *SIP*. For the angles, it is *ICP*, *SIP* and *T3M*. We want to mention again at this point, that the *T3M* method is evaluated only for the frames where all three templates could be matched successfully. For example within the working range, the method failed in 8.4%. This fail rate depends on the facial expressions present in a video sequence, which is illustrated in Table 5.6 of the next section.

5.3.3 Comparison of unmarked and marked sequences

Ideally, the results of the unmarked and marked sequences should be the same, otherwise the unmarked sequences were labeled imprecisely. By comparing the accuracies in Table 5.3, we can get an impression of the error that has been introduced. Nevertheless, it remains a rough estimation, because the results depend on how a subject moved its head during the video sequence and which facial expressions were made. From the nose tip accuracy for all frames we see, that the results except for those of the *ICP* method have become slightly better. The use of markers showed that the nose tip distance error for all frames is lower in reality. In contrast, the nose tip accuracy within the working range and also the angle accuracies for both, all frames and those within the working range, have become worse mostly. The Tables 5.1 and 5.2 which show the *MAE* and the standard deviation confirm the trend of the accuracy results. From this comparison we see, that hand-labeling without markers is not always accurate. The feature points were often assumed to be at a different location, which happened to be near the results of our head pose estimation methods. However, the marked sequence improved the quality of our statistic evaluation. The results of the marked sequence are more reliable and still far from being bad. But, to get an even more clear view on the results, we categorize the facial expressions of the marked sequence in the next section. In this way, we can evaluate the results of the frames showing a neutral face.

	Nose tip accuracy at $t = 3mm$ [%]						Angle accuracy at $t = 5°$ [%]					
	all frames			within $r = 5mm$			all frames			within $r = 5mm$		
method	ICP	SIP	T3M	ICP	SIP	T3M	ICP	SIP	T3M	ICP	SIP	T3M
unmarked	63	79	67	89	99	93	56	46	60	77	63	90
marked	68	91	74	80	98	85	40	46	49	44	45	76

Table 5.3: Accuracy comparison for the unmarked and marked sequences.

5.4 Facial expression analysis

In this experiment we manually sort the various facial expressions (recall Figure 4.1) in the marker sequence into four categories:

Neutral describes the human face when no emotion can be observed.

Eyes closed denotes that the eyes are normally closed, like during blinking or sleeping.

Mouth moved characterizes talking or laughing.

Grimaces show strong facial deformation, such as a widely opened mouth, a chin which is moved to one side or tightly closed eyes.

Now, we can evaluate the performance of the head tracking system based on the categories and get a clearer view on the results. By observing the accuracy diagrams in the Figures 5.7, 5.8 and 5.9, we get an impression of how our methods react during facial expressions.

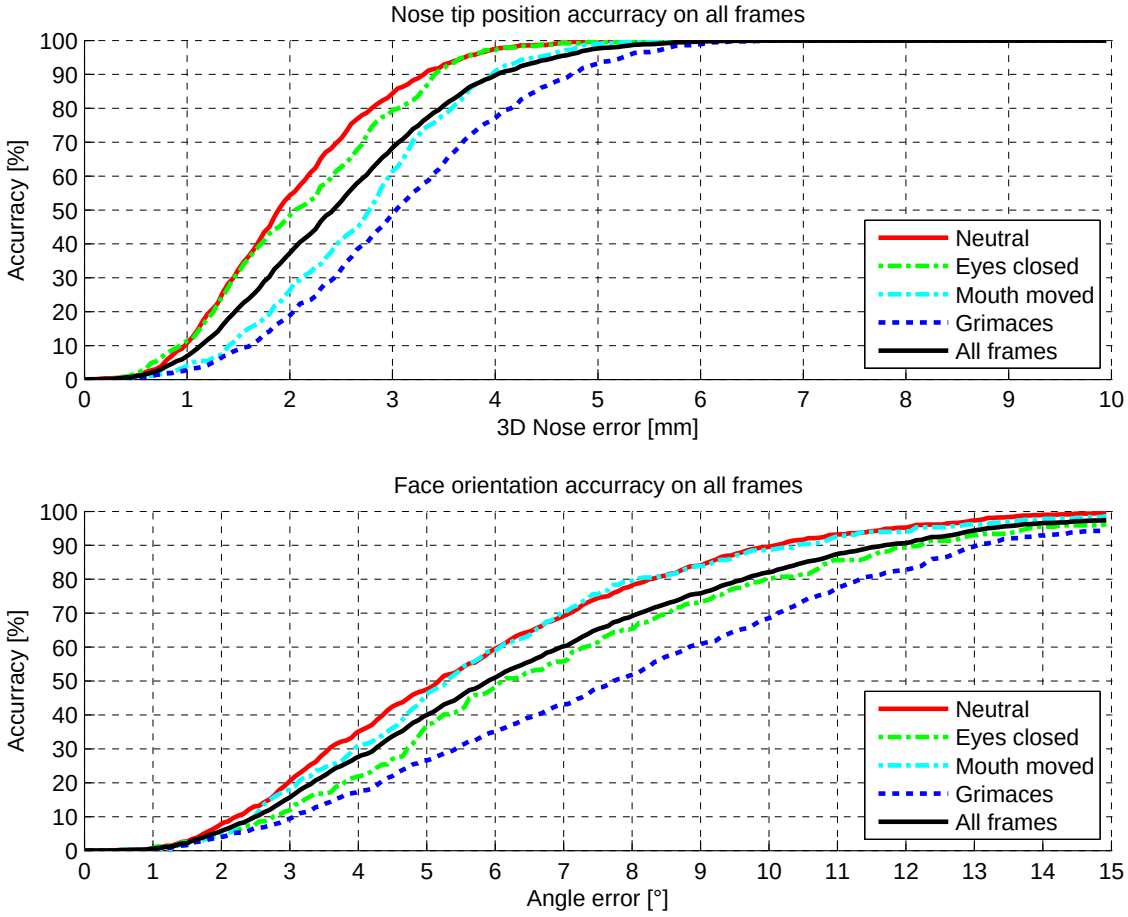


Figure 5.7: ICP method: Comparison of facial expressions.

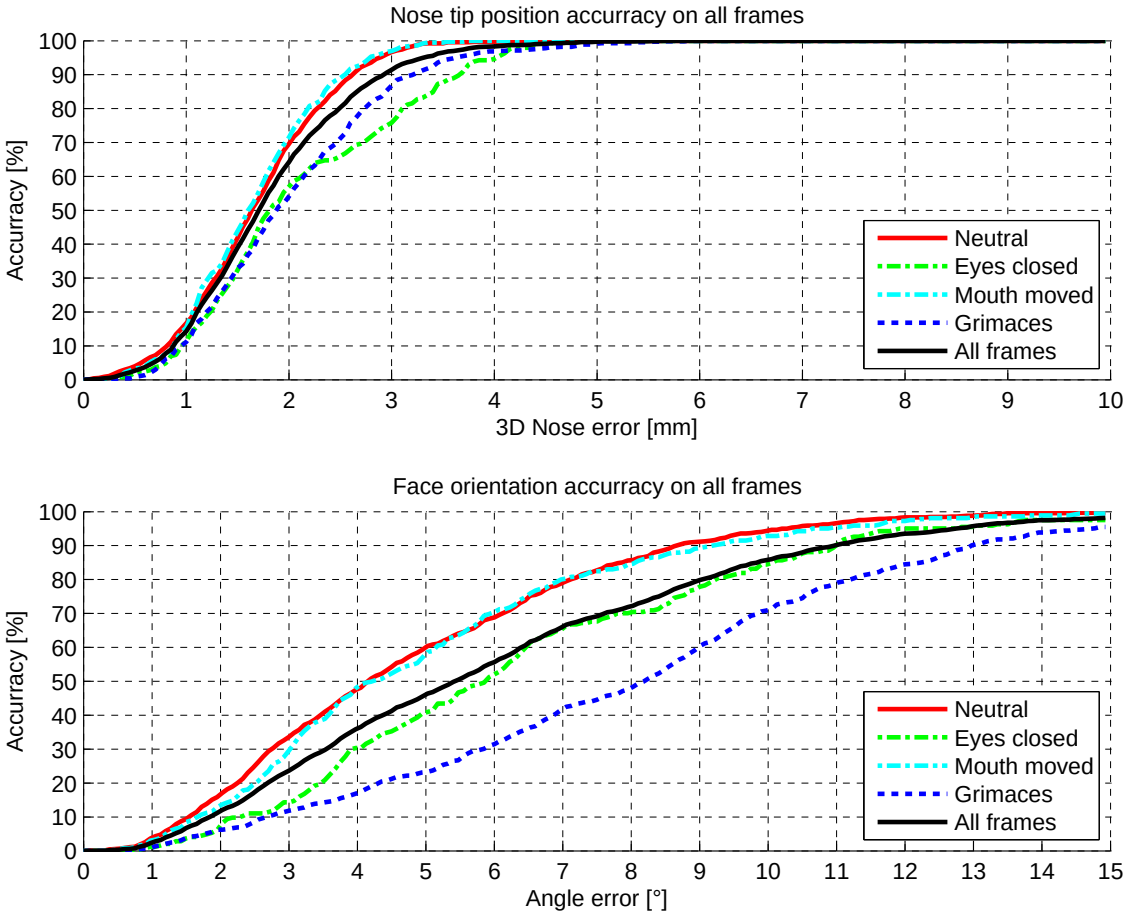


Figure 5.8: SIP method: Comparison of facial expressions.

In the *ICP* method, grimaces and mouth movement cause the largest errors, because a point cloud of the whole face is compared. For the *SIP* and *T3M* method, closing the eyes and grimaces are worst. This can also be seen in Tables 5.4 and 5.5.

In Table 5.6 we see, how the fail rate of the *T3M* method depends on facial expressions. The percentages refer to the frames present in one category, so they do not sum up to 100%. Closing the eyes like blinking or tightly during grimaces is worst for this method. Mouth movement can be handled like a neutral face, because none of the three templates includes the mouth region, which is why the template matching is not affected. The neutral face, especially within the working range, can be handled without problems.

The head tracking system should be applied during medical eye tumor treatment, where the patient shows a neutral expression and stays within the working range. In this case, the *SIP* method is best for estimating the nose tip position and the *T3M* method for the facial orientation. Further, 68% of the corresponding frames lie within $1.15 + 0.71 = 1.86mm$ of the true nose tip position and $2.45 + 1.42 = 3.87°$ of the true facial orientation $(\mu + \sigma)$ and 99% within $3.28mm$ and $6.71°$ $(\mu + 3\sigma)$ (note: diagrams show ALL frames, not $< 5mm$!).

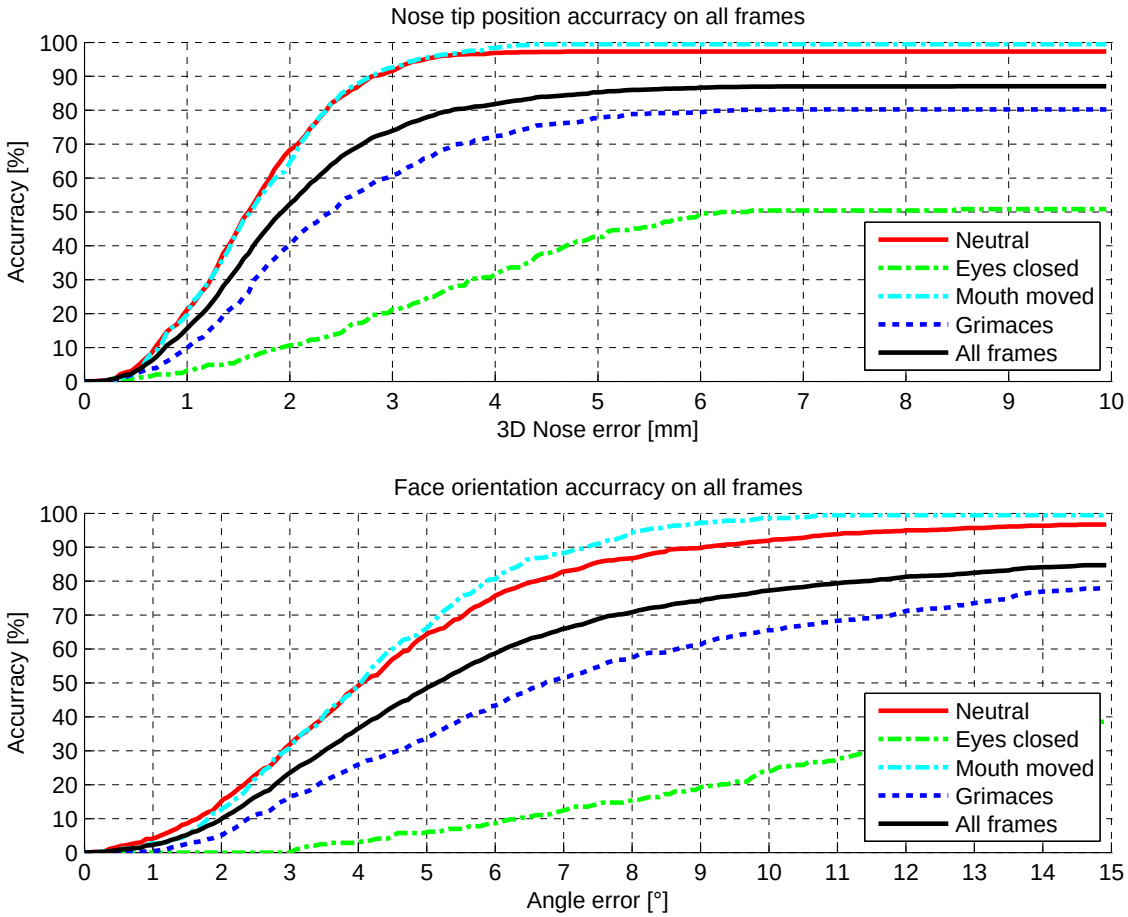


Figure 5.9: T3M method: Comparison of facial expressions.

	Neutral	Eyes closed	Mouth moved	Grimaces	All frames
ICP	2.03 ± 0.91	2.14 ± 0.98	2.71 ± 0.99	3.10 ± 1.20	2.51 ± 1.13
SIP	1.68 ± 0.71	2.11 ± 1.01	1.64 ± 0.65	2.02 ± 0.93	1.82 ± 0.83
T3M	1.65 ± 0.77	3.41 ± 1.56	1.71 ± 0.80	2.25 ± 1.21	1.95 ± 1.10

(a) All frames.

	Neutral	Eyes closed	Mouth moved	Grimaces	All frames
ICP	1.42 ± 0.53	1.42 ± 0.64	2.22 ± 0.79	2.96 ± 0.98	2.04 ± 1.01
SIP	1.15 ± 0.71	1.35 ± 0.73	1.28 ± 0.62	1.31 ± 0.67	1.24 ± 0.68
T3M	1.34 ± 0.68	2.97 ± 1.66	1.42 ± 0.73	1.88 ± 1.00	1.57 ± 0.92

(b) Within $5mm$.

Table 5.4: Facial expressions: Comparison of the nose tip error [mm].

	Neutral	Eyes closed	Mouth moved	Grimaces	All frames
ICP	5.74 ± 3.11	7.01 ± 3.83	5.92 ± 3.16	8.07 ± 4.12	6.61 ± 3.69
SIP	4.76 ± 2.89	6.33 ± 3.41	4.93 ± 2.97	8.02 ± 3.91	5.99 ± 3.60
T3M	4.54 ± 2.82	12.25 ± 7.88	4.24 ± 2.06	6.46 ± 3.86	5.53 ± 4.05

(a) All frames.

	Neutral	Eyes closed	Mouth moved	Grimaces	All frames
ICP	5.62 ± 3.62	7.27 ± 4.30	5.98 ± 3.65	8.40 ± 4.65	6.56 ± 4.18
SIP	4.74 ± 3.25	6.38 ± 3.85	5.35 ± 3.32	9.17 ± 4.14	6.28 ± 4.04
T3M	2.45 ± 1.42	7.31 ± 4.32	2.77 ± 1.38	4.84 ± 3.10	3.40 ± 2.55

(b) Within $5mm$.

Table 5.5: Facial expressions: Comparison of the angle error [°].

	Neutral	Eyes closed	Mouth moved	Grimaces	All frames
T3M	2.7	49.2	0.5	19.8	13.0

(a) All frames

	Neutral	Eyes closed	Mouth moved	Grimaces	All frames
T3M	0.0	57.7	0.0	10.9	8.4

(b) Within $5mm$.

Table 5.6: Facial expressions: Fail rate comparison for the T3M method [%].

5.5 2D displacement error analysis

In this section we want to analyze the 2D dependence (inspired by [8]) of the displacement error. For this evaluation we only use the marked sequence, so that it can be reliable said if the displacement error is random or systematic. To be able to make a meaningful statement, we have to divide the area in which the head moves into grid cells first. The grid is centered at the initial nose tip position and uses patches of the size $2 \times 2mm$. Instead of the *MAE*, which only computes an absolute error without a direction, we use the *MSD* from Section 3.5.2 (also see Section 5.1.4). It can be interpreted as the displacement between the mean position of a set of true points and the one of a set of point estimates. Applied only in a single grid cell, it gives us the approximate error of one of our head pose estimation methods at a given 2D nose tip position, which is the center of a grid cell. Because the head rests on the backplane of the system during tracking, the head rotation is strongly coupled to this position. We want to find out, if the displacement error is somehow correlated with the x- or y-coordinates. In this case, the error is not only random but systematic and could be calibrated to increase the accuracy of a method.

Now we will discuss the *MSD* diagrams for our head pose estimation methods. The blue arrows show the mean error vector in x- and y-direction at a specific nose tip position. To the right of the diagrams we see a colorbar which gives information about how many

frames were evaluated for the computation of the error vector in each of the grid cells.

In Figure 5.10 we see a diagram for the *MSD*, evaluated for the *ICP* method. It shows a dependence of the error in y-direction, but there is also a random component of the error, as we can see from the arrows pointing up and down in some rows of the grid. By comparing the ground truth feature points to the estimates in the whole video sequence, one can see a strong displacement of the estimated nose tip position during mouth movements and grimaces. This is because a complete point cloud of the face is compared in the *ICP* algorithm. The resulting error caused by varying facial expressions at the initial head pose is mostly larger than the registration error for just a neutral face during head rotations. Due to this strong dependence on facial expressions, a 2D displacement error can not be calibrated.

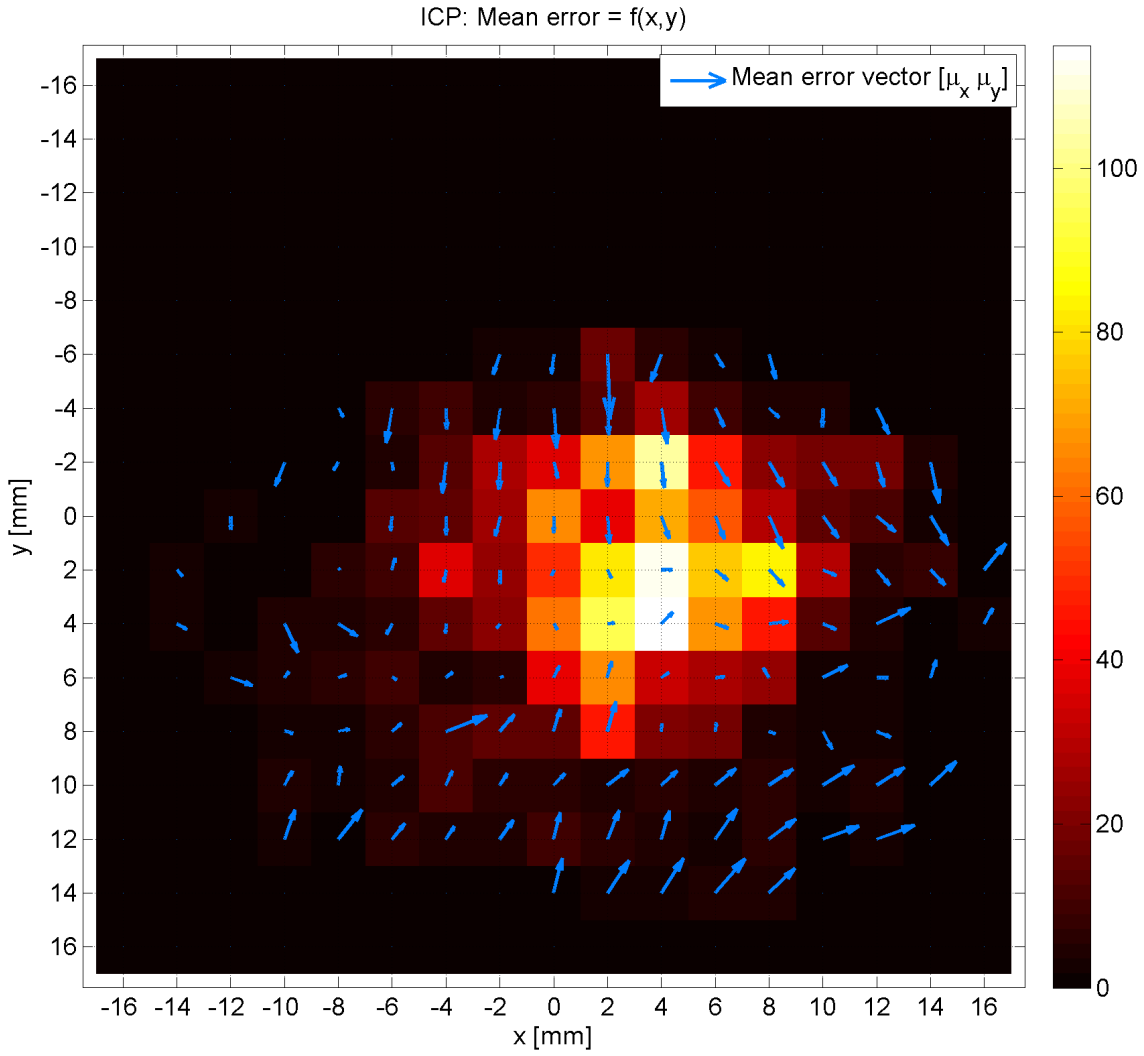


Figure 5.10: ICP method: 2D displacement analysis.

In Figure 5.11 the diagram for the *T3M* method shows a systematic error. The estimated nose tip position is shifted into the direction of the initial one at the center. The pattern is a bit turbulent, which may be caused by the rigid motion estimation for three template points in 3D. The eye templates are bounded in the eye socket, which results in a different movement than for the nose template. Further, grimaces, especially with tightly closed eyes cause the eye templates to move different from the nose template.
By comparing the diagram of the *T3M* method to one of the other methods we see, that some of the cell colors are different. This has the reason, that the template matching failed for 13% of the frames, mostly for the eye templates at grimaces or too strong head rotations. These frames were left out from the displacement error vector computation.

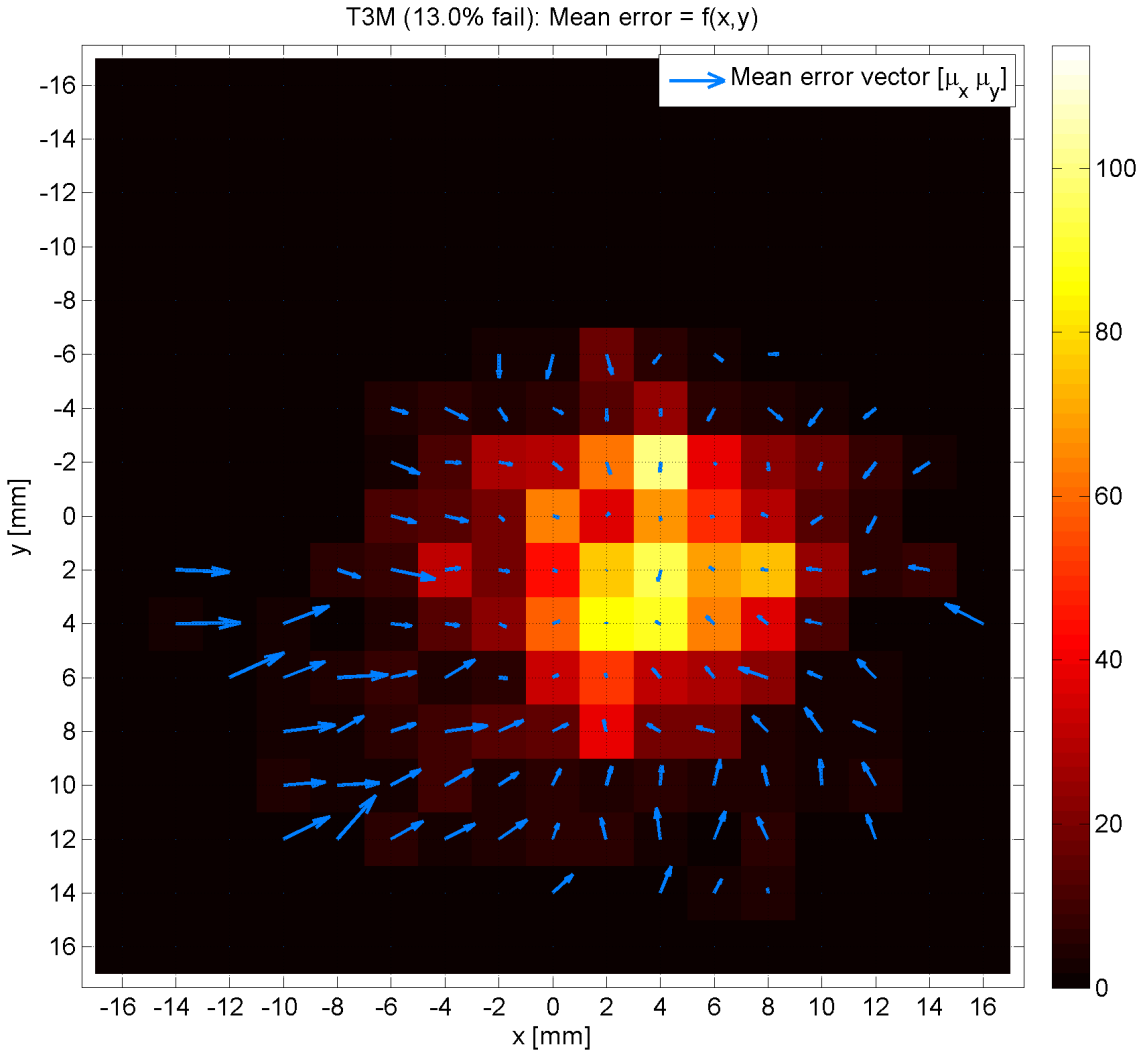


Figure 5.11: T3M method: 2D displacement analysis.

Figure 5.12 shows the displacement error for the *SIP* method. It is clearly visible, that the estimated nose tip position is shifted into the direction of the initial one at the center. Additionally, this deviation gets stronger, the greater the distance of current true nose tip position and initial one gets. This method has a very clear systematic error pattern which could be calibrated, in order to improve the accuracy for the head pose estimates.
This method shows a very clear displacement compared to the *T3M* method, because it only depends on the template position of the very exposed nose tip feature.

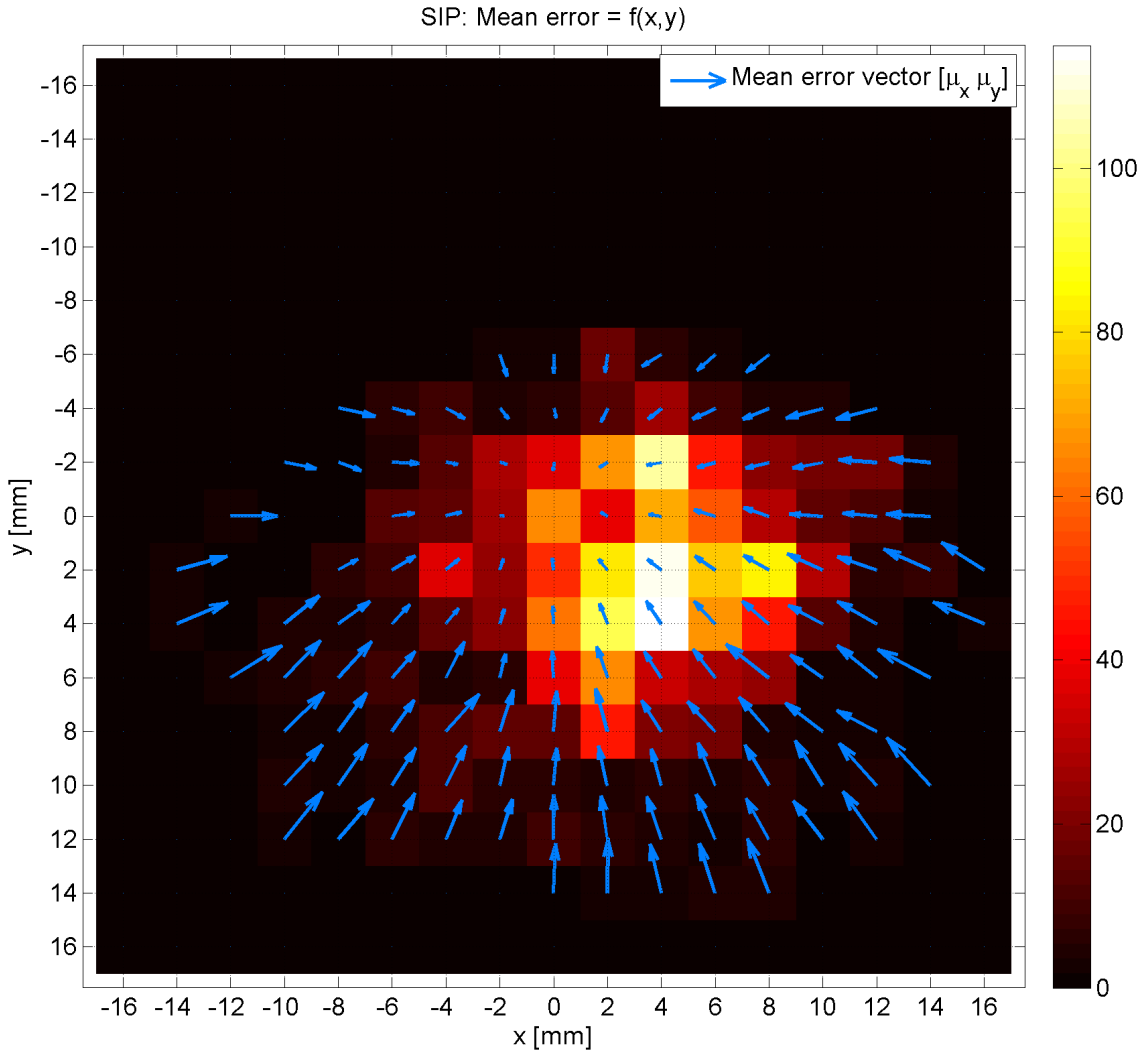


Figure 5.12: SIP method: 2D displacement analysis.

5.6 Discussion

In Section 5.2 we optimized parameters and properties within a method. For the *ICP* method, we analyzed the filtering stage and tried different filter kernels. We found, that a small median filter (7×7 kernel) without a Gauss filter is the best compromise between noise and outlier reduction and deformation of the face model.
For the *SIP* method we tried out two nose detector approaches and came to the result, that matching a nose template is more robust than detecting the nose tip just from proximity to the *ToF* camera. The nose would remain at the global depth minimum during the whole video sequence, but *ToF* noise and the actual size of the nose tip lead to a strong jitter of the estimated position, which makes this approach too imprecise.

In the inter-method statistic evaluation of Section 5.3 we compared the *ICP*, *SIP* and *T3M* method based on the *MAE* of the nose tip position and the face orientation angles. We used an accuracy measure, which computes the performance of our head pose estimation methods as a function of the *AE* of the nose tip position and the face orientation angles, respectively. *SIP* turned out to be the most accurate method to estimate the nose tip position. *T3M* is best for estimating the face orientation, but is not robust during facial expressions where the eyes are normally or tightly closed. Further, we discussed the error which is introduced from imprecise hand-labeling, by comparing sequences without markers at the nose tip and inner eye corner feature points to a sequence with markers.

In Section 5.4 we divided the various expressions of the human face into four categories to analyze how they influence our head pose estimation methods. Grimaces in which the mouth is widely opened or the chin is moved to one side bring large errors for *ICP* and *SIP*, but do not influence the *T3M* method. Normally and tightly closed eyes bring large errors for *SIP* and *T3M*, or even cause the *T3M* method to fail. In medical eye tumor treatment, the field of application of the head tracking system, the neutral facial expression within a working range of $5mm$ is the most important. Further, the eyes need to stay open for the MedEyeTrack system to work correctly. During the treatment, the patient will also not be moving the mouth like during talking or laughing. Under these special conditions, 99% of the corresponding frames lie within $3.28mm$ from the true nose tip position and $6.71°$ from the true face orientation for *SIP* and *T3M*, respectively.

In the 2D displacement error analysis of Section 5.5 we used the *MSD* to find systematic error patterns. The *ICP* method showed a more random pattern, which was caused by the stronger influence of facial expressions on the algorithm.
The *T3M* and the *SIP* method both use approaches based on 2D template matching. Template matching assumes only a 2D movement of the template. This leads to a systematic error which can be calibrated. In the *T3M* method, where the nose tip position is estimated from all three templates, the error pattern is a bit turbulent. In the *SIP* method, the nose tip estimate is clearly shifted towards the initial position at the center, because it only depends on the template position of the very exposed nose tip feature. The error also depends on the distance between true nose tip and center.

6 Conclusion

Contents

As a conclusion, we shortly summarize the chapters of this Master's Thesis and give an outlook to our future work.

6.1 A final summary

In Chapter 1 we gave an introduction to medical eye tumor treatment, which is the field of application of the head and eye tracking system. The objective of this thesis was to find a head pose estimation method for the tracking of unwanted motion. The system can be used to generate triggering commands to control the treatment devices. If the head leaves the initial position which was recorded at the first session, the devices can be paused.

In Chapter 2 we discussed, how the head and eye tracking system is developed based upon the existing MedEyeTrack system. For the head tracking system we had to develop a hardware setup which can be combined with the existing setup. A Time of Flight (ToF) camera is mounted closely to the face, directed onto the approximate nose tip position of the patient. After an analysis of facial feature points, the nose tip and the inner eye corners have been identified to be the most stable ones which are best suited for face tracking.
Further, we suggested a possibility for integrating the *C++* software of our head tracking prototype into the existing MedEyeTrack software written in *C#* via a class library. Both systems can be combined to generate triggering signals when both, head and eyes, are in their correct position for the treatment.

Chapter 3 collected theory and background information. We explained for example the sensor types which are important for this project and how the 2D calibration of a

sensor works. The main focus was directed onto geometric definitions. We discussed the coordinate system we use, how a rigid-body transformation can be estimated using Singular Value Decomposition (SVD) and in detail how 3D rotations work.

The main part of this thesis was Chapter 4, in which we discussed the problem of head pose estimation and presented three methods which can be used for this purpose.
The first method was a registration-based approach using the Iterative Closest Point (ICP) algorithm. Data point clouds are transformed to a high-resolution face model which has been generated using a Structured Light (SL) system. This well known approach can not be used in real-time applications, but we wanted to use it as a reference for comparing the performance of the other methods.
In the Template Matching (T3M) method we use templates at the nose tip and both inner eye corner feature points to track their positions in 2D. From the *ToF* depth map we get the corresponding 3D points and are able to estimate a rigid motion between two frames.
In the Spherical Intersection Profile (SIP) method, we wanted to try an approach which has been proposed by Meers and Ward. Outgoing from the nose tip position, which we tracked with a template, spheres are intersected with the face to generate profile lines. These can be used to compute head pose information from a single frame.

In Chapter 5 we made experiments to optimize and analyze the three head pose estimation methods. We found that the *ICP* method had difficulties with noise and outliers. Further, the point cloud comparison was inaccurate at facial expressions which cause a strong deformation of the face, like for example when the mouth is opened wide. The *SIP* method, which uses a single nose template, turned out to give the best estimation of the nose tip position. The *T3M* method was best for estimating the face orientation, but lacked in robustness at too strong rotations or facial expressions with closed eyes.
Then, we discussed the case of medical eye tumor treatment, where the patient shows a neutral face and is advised to remain at the initial head pose (we assumed a working range of $5mm$ around the initial nose tip position). For this case, the error of our head tracking system would lie within $3.28mm$ for the nose tip (*SIP* method) and 6.71° for the face orientation (*T3M* method) in 99% of the frames showing a neutral face we evaluated.
Finally, we analyzed the 2D dependence of the nose tip error for our head pose estimation methods. The head rests on the back of the tracking system, so the nose tip position depends on the head rotation. Especially the *SIP* method showed a very clear systematic error pattern, which can be calibrated.

6.2 Future work

In our future work we will fully develop the head tracking system, whereby a choice about the head pose estimation method has to be made. A possibility would be a mixed approach, in which the *SIP* method could be used in the cases where *T3M* fails. The *SIP* method would still give a better estimation of the face orientation than the *ICP* method. Further, the *T3M* method could use just the nose template instead of all three ones for

position estimation. As a result, the estimated nose tip position would be as accurate as in the *SIP* method.

We are eager to improve the results and lower the errors we reached in this Master's Thesis. First, we will try to calibrate the systematic error of the nose tip position. To increase accuracy further, the *ToF* camera could also be depth calibrated. At the moment we just did a 2D calibration of the *ToF* sensor, which leads to a distortion of the depth data depending on the distance to the camera. For example the point cloud comparison of the *ICP* method could be improved in this way. To analyze the influence of noise, we will also test different *ToF* sensors which could be of lower noise and maybe of higher resolution.

Further, we will record video sequences of more test subjects to make a more comprehensive statistic evaluation.

An idea we also have in mind is to try out Fanelli's method from [8], which is a learning-based method using a random-forest classifier. In this method, a classifier is trained on a large synthetic data set. Fanelli uses his method for the classification of head poses in a large range of rotations and translations where the requirements to the error are different from ours. It would be interesting to adapt this method to the application of medical eye tumor treatment, where small head motions and high accuracy have priority. A great advantage of this method would be that it copes with partial occlusions of the face. For the field of medical eye tumor treatment, this would allow the patient to wear a head coil during Magnetic Resonance Imaging (MRI) (see Figure 6.1), which enhances the image quality and makes it possible to determine the tumor location more exactly.

Figure 6.1: A head coil improves the image quality during MRI.

A

List of Acronyms

AE	Absolute Error
CLR	Common Language Runtime
DLL	Dynamic Link Library
DOF	Degrees of Freedom
GUI	Graphical User Interface
ICP	Iterative Closest Point
MAE	Mean Absolute Error
MRI	Magnetic Resonance Imaging
MSD	Mean Signed Difference
RANSAC	Random Sample Consesus
ROI	Region Of Interest
SD	Signed Difference
SIP	Spherical Intersection Profile
SL	Structured Light
SVD	Singular Value Decomposition
T3M	Template Matching
ToF	Time of Flight

Bibliography

[1] (2014). Euler angle formulas. `http://www.geometrictools.com/Documentation/EulerAngles.pdf`. (page 21, 22, 23)

[2] (2014). Retrobulbar anesthesia. `http://www.youtube.com/watch?v=F0LuOYUv6zw`. (page 4)

[3] (2014). Template matching. `http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html`. (page 24)

[4] Bar, T., Reuter, J., and Zollner, J.-M. (2012). Driver head pose and gaze estimation based on multi-template icp 3-d point cloud alignment. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1797–1802. (page 5)

[5] Besl, P. and McKay, N. D. (1992). A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256. (page 5, 23)

[6] Bouguet, J.-Y. (2014). Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc`. (page 18)

[7] Chua, H.-C. (2012). 3d graphics with OpenGL - basic theory. `http://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html`. (page 29)

[8] Fanelli, G., Gall, J., and Van Gool, L. (2011). Real time head pose estimation with random regression forests. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*. (page 5, 51, 64, 71)

[9] Fauvel, C. and Weng, N. (2012). Avoid gimbal lock for rotation/direction maya manipulators. `http://around-the-corner.typepad.com/adn/2012/08/avoid-gimbal-lock-for-rotationdirection-maya-manipulators.html`. (page 22)

[10] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395. (page 32)

[11] Garcia, R. and Zakhor, A. (2012). Consistent stereo-assisted absolute phase unwrapping methods for structured light systems. *Selected Topics in Signal Processing, IEEE Journal of*, 6(5):411–424. (page 18)

[12] Lange, R. (2000). *3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*. PhD thesis, University of Siegen. (page 16)

[13] Meers, S. and Ward, K. (2008). Head-pose tracking with a time-of-flight camera. (page 40, 45)

[14] Meers, S. and Ward, K. (2009). Face recognition using a time-of-flight camera. In *Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference on*, pages 377–382. (page 5, 40, 48)

[15] M&R Automation GmbH (2013). Eye-Tracker ermöglicht neuartige Behandlungsverfahren bei Augen-Tumoren. *Human Technology Styria, Botenstoff 01.13 (March 2013)*, pages 11 –12. (page 8)

[16] Ribo, M. and Brandner, M. (2005). State of the art on vision-based structured light systems for 3d measurements. In *Robotic Sensors: Robotic and Sensor Environments, 2005. International Workshop on*, pages 2–6. (page 17, 18)

[17] Sorkine-Hornung, O. (2014). Least-squares rigid motion using svd. `http://igl.ethz.ch/projects/ARAP/svd_rot.pdf`. (page 19)

[18] Spreeuwers, L. (2011). Fast and accurate 3d face recognition. *International Journal of Computer Vision*, 93(3):389–414. (page 19, 32)

[19] Teizer, S. J. R. J. (2011). Automated head pose estimation of vehicle operators. In *2011 Proceedings of the 28th ISARC, Seoul, Korea*, pages 880–885. (page 5)

[20] Tong, Y., Wang, Y., Zhu, Z., and Ji, Q. (2007). Robust facial feature tracking under varying face pose and facial expression. *Pattern Recogn.*, 40(11):3195–3208. (page 5)

[21] Tu, Y., Lin, H.-S., Li, T.-H., and Ouhyoung, M. (2012). Depth-based real time head pose tracking using 3d template matching. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, pages 13:1–13:4, New York, NY, USA. ACM. (page 34)

[22] University of Wisconsin Hospitals and Clinics Authority (2014). Stereotactic radiosurgery - a patient guide. `http://www.uwhealth.org/healthfacts/cancer/5218.html`. (page 4)

[23] Wiersma, R. D., Wen, Z., Sadinski, M., Farrey, K., and Yenice, K. M. (2010). Development of a frameless stereotactic radiosurgery system based on real-time 6d position monitoring and adaptive head motion compensation. *Physics in Medicine and Biology*, 55(2):389. (page 5)

[24] Yang, J., Li, H., and Jia, Y. (2013). Go-icp: Solving 3d registration efficiently and globally optimally. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1457–1464, 1457-1464. (page 5)

[25] Zhang, S. and Yau, S.-T. (2008). Three-dimensional shape measurement using a structured light system with dual cameras. *Optical Engineering*, 47(1):013604–013604–12. (page 17)

[26] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334. (page 18)

[27] Zhao, G.-Q., Chen, L., and Chen, G.-C. (2005). A simple 3d face tracking method based on depth information. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 8, pages 5022–5027 Vol. 8. (page 5, 35)

[28] Ziraknejad, N., Lawrence, P., and Romilly, D. (2012). The effect of time-of-flight camera integration time on vehicle driver head pose tracking accuracy. In *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, pages 247–254. (page 17)