

UNIVERSITY OF TWENTE
DEPARTMENT OF SIGNALS AND SYSTEMS

MASTER THESIS

Tracking of the tongue in three dimensions
using a visual recording system

Author:
T.A.G. HAGEMAN

Committee:
Prof. Dr. Ir. C.H. SLUMP
Dr. Ir. F. VAN DER HEIJDEN
Prof. Dr. A.J.M. BALM
MSc. M.J.A. VAN ALPHEN
Dr. Ir. C. SALM

DOCUMENT ID: EWI/SAS 2013-002

January 24, 2013

Summary

Oral cancer is a disease which can significantly affect one's oral abilities, including speech, food transport, chewing and swallowing. There are several treatment methods, but the choice of treatment is determined by subjective means. The Dynamic Virtual Surgery project aims to develop a system which allows the study of post-operative function loss by pre-operative simulations. For tongue cancers, this means that a good patient-specific tongue model must be constructed. The training process of such a model involves tracking the tongue shape in three dimensions during an operation, while simultaneously acquiring EMG data of the tongue.

This thesis involved the development of a system able to track the shape of the tongue in three dimensions. The chosen approach uses three cameras placed in front of the patient, converting the information of multiple cameras to a 3D-representation. Difficulties in this process include the smooth, moist and occlusion-prone environment of the tongue. Markers are stuck to specific locations on the tongue in order to define high-contrast landmarks, a process involving the use of non-toxic materials. Reproducibility of marker locations is guaranteed to a certain extent by preparing the layout beforehand on a flexible bandage which is stuck to the tongue.

The tracking algorithm works offline and involves several important processes. Template matching is used to find the marker coordinates in the recorded frames. An outlier correction algorithm is then run to correct for measurement errors. Finally, a Kalman filter is used in order to track the state of the tongue, directly transforming the 2D-measurements to a 3D-representation.

To deal with outliers, a method is proposed involving the use of a principal component (PCA) model. This model, based on the 3D marker locations, allows reduction of dimensionality and does not allow physically impossible tongue states. A method similar to RANSAC, involving hypothesis generation and -testing on multiple subsets of measured marker coordinates in the recorded frames, determines which PCA components are most likely for the current situation, which can then be used to correct the outliers.

The method proves to be working and provides results with a 3D-accuracy down to sub-millimeter level. In the case of occluded markers, accuracy drops but still remains below two millimeters.

Contents

| | |
|--|-----------|
| Summary | 3 |
| 1 Introduction | 7 |
| 1.1 Dynamic virtual surgery in oral cancer | 7 |
| 1.2 The model | 7 |
| 1.3 Scope of this thesis | 8 |
| 2 The tongue | 10 |
| 2.1 Tongue layout | 10 |
| 2.2 Constraints on tracking | 10 |
| 3 Tracking the tongue | 12 |
| 3.1 Method | 12 |
| 3.2 Video processing | 13 |
| 4 Camera analysis and -calibration | 15 |
| 4.1 Camera geometry | 15 |
| 4.2 Camera calibration | 15 |
| 4.3 Calibration using checkerboard pattern | 17 |
| 4.3.1 Method | 17 |
| 4.3.2 Results | 17 |
| 4.3.3 Discussion | 18 |
| 4.4 Calibration using cube | 18 |
| 4.4.1 Method | 19 |
| 4.4.2 Results | 19 |
| 4.4.3 Discussion | 21 |
| 4.5 Comparison of methods | 21 |
| 4.6 Conclusion | 21 |
| 5 Multi-camera analysis | 23 |
| 5.1 3D estimation of a single point | 23 |
| 5.2 3D estimation of multiple points | 24 |
| 5.3 3D estimation using a Kalman filter | 25 |
| 6 Principal Component Model | 27 |
| 6.1 Degrees of freedom | 27 |
| 6.2 Training the PCA model | 28 |
| 6.3 Training with incomplete data | 30 |
| 6.4 Training results | 30 |
| 6.5 Role of PCA within the system | 31 |
| 7 Marker layout | 34 |
| 7.1 Materials | 34 |
| 7.2 Marker color | 35 |
| 7.2.1 Intensity | 35 |
| 7.2.2 Color | 35 |
| 7.3 Final layout | 36 |
| 7.4 Facial markers | 38 |

| | | |
|-----------|---|-----------|
| 8 | Setup | 40 |
| 8.1 | Initial setup | 40 |
| 8.2 | New Setup | 40 |
| 8.2.1 | General requirements | 41 |
| 8.2.2 | Technical requirements | 42 |
| 8.2.3 | Design | 43 |
| 8.2.4 | Hardware | 43 |
| 9 | Tracking algorithm | 46 |
| 9.1 | Overview of the tracking process | 47 |
| 9.2 | Pre-processing images | 47 |
| 9.3 | Marker detection | 49 |
| 9.3.1 | Defining the templates | 50 |
| 9.4 | PCA model rotation | 50 |
| 9.5 | Outlier detection | 51 |
| 9.6 | Tongue state estimation and -prediction | 52 |
| 10 | Experiments | 56 |
| 10.1 | Qualitative experiments | 56 |
| 10.2 | Quantitative experiments | 58 |
| 10.2.1 | Static experiments | 58 |
| 10.2.2 | Dynamic experiments | 59 |
| 11 | Results | 62 |
| 11.1 | Qualitative results | 62 |
| 11.1.1 | Experiment 1 | 62 |
| 11.1.2 | Experiment 2 | 62 |
| 11.1.3 | Experiment 3 | 64 |
| 11.2 | Quantitative results | 65 |
| 11.2.1 | Static results | 65 |
| 11.2.2 | Dynamic results | 67 |
| 12 | Conclusions | 73 |
| 12.1 | Thesis overview | 73 |
| 12.2 | Camera calibration | 73 |
| 12.3 | Setup | 73 |
| 12.4 | Measurement protocol | 73 |
| 12.5 | PCA model | 74 |
| 12.6 | Qualitative experiments | 74 |
| 12.7 | Quantitative results | 74 |
| 13 | Recommendations | 75 |
| 13.1 | Setup | 75 |
| 13.2 | Measurement protocol | 75 |
| 13.3 | PCA model | 75 |
| 13.4 | Marker detection | 76 |
| | Abbreviations | 77 |
| | Bibliography | 79 |

Chapter 1

Introduction

Oral cancer is a type of head and neck cancer affecting several regions in the oral cavity and lips. The disease affects 4,6 in 100.000 persons. Oral cancer can significantly affect one's oral abilities, including speech, food transport, chewing and swallowing, and can be very painful. Oral functions are important to the quality of life and should be affected as little as possible by treatment methods.

Current treatment methods include surgery, chemotherapy, radiation therapy or a combination of these. While small tumors are often treated by radiation therapy, the standard treatment of larger ones is by means of surgery and prosthetic reconstruction, whose results are most optimal. However, there are also tumors which are declared *functional inoperable* due to their anatomic location or size. These tumors would result in a too large functional loss when being treated by surgical means, and are treated by a combination of radiotherapy and chemotherapy [1].

For best results it is crucial to determine the right method of treatment. One of the problems is to declare whether a tumor is functional inoperable or not, which is currently decided by subjective means, as no clear anatomical boundaries can be defined. Furthermore, surveys among head-neck surgeons and radiotherapists point out that there is disagreement about when a tumor can be declared functional inoperable [22][3]. Development of a system that can aid doctors in making this choice would provide an outcome.

1.1 Dynamic virtual surgery in oral cancer

The Anthonie van Leeuwenhoek Hospital in the Netherlands has started a project in collaboration with the department of Signals and Systems (SAS) of the University of Twente. The project *Dynamic virtual surgery in oral cancer* is focused on development of technology which can predict the influence of treatment methods on patients. The most important goals are:

- to accurately predict the patient-specific functional losses that might occur after a partial resection of the tongue and other oral regions;
- to predict and present the remaining functionality, including speech; and
- to provide an objective measure for defining functional inoperability.

This technology aims to construct a patient-specific model of the oral region which will allow simulation of the tongue motion and speech. After construction it can be altered by means of virtual surgery, by removing tissue, performing reconstruction and adding scar tissue. This allows the pre-operative study of virtual postoperative change of speech and tongue motion, providing a fundamental step in the decision process for different treatment alternatives, and providing a way of advising patients.

An additional non-patient-specific goal is to map the nerves across the tongue. Although it is known which nerves actuate certain muscles in the tongue, it is currently unknown how those nerves split up in the tongue and how the nerve endings are mapped across it. An additional goal is to find the general mapping of this pattern.

1.2 The model

A good oral model is very important for this project, as it should resemble the physics of a real oral region as accurately as possible. Because of inter-personal variance, such a model should be developed for each patient

specifically. Important is the relation between nerve activity and motion of oral regions, visualized in figure 1.1. Nerve activity result in muscle activity, which produces motion of oral regions resulting in desired functionality as speech and expressions. For construction of a model, it is needed to define the relation between nerve activity and shape and motion of the oral regions. The process of training such a model is also shown in figure 1.1.

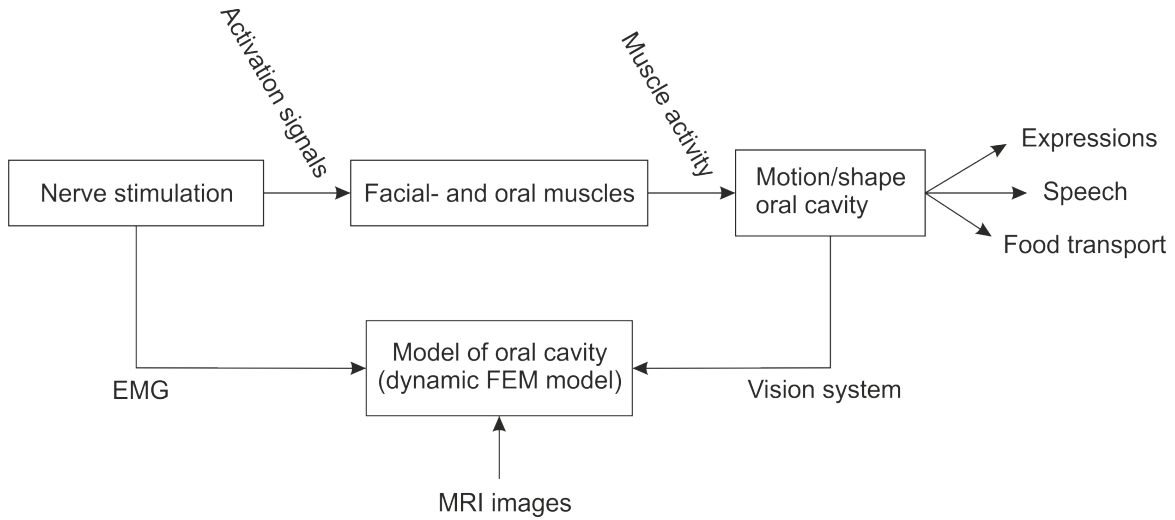


Figure 1.1: Block diagram for building an oral cavity model.

Training will be performed during narcosis, in which the surgeon will apply nerve stimulation on the patient. Nerve activity is measured on or close to specific tongue muscles by means of EMG, while the motion resulting from these excitations can be measured by a 3D imaging system. The relation between nerve activity and motion can then be modeled based on these measurements, by anatomically mapping the muscles and nerves and their relation. This information can then be combined with a biomechanic finite element model (FEM). A tongue FEM has already been constructed by M. van Alphen, who based its design onto a MRI scan[20]. The model created this way then allows the simulation of motion and its resulting functions based on virtual nerve innervations. After having trained the model, adaptations can be made by for instance removing a part of the tongue during virtual operations. Additional adaptations can be made, like the addition of scar tissue to the tongue by modifying its biomechanical properties. Then, using again similar nerve excitations, the loss of functionality can be simulated, including the loss of speech. This process is illustrated in figure 1.2.

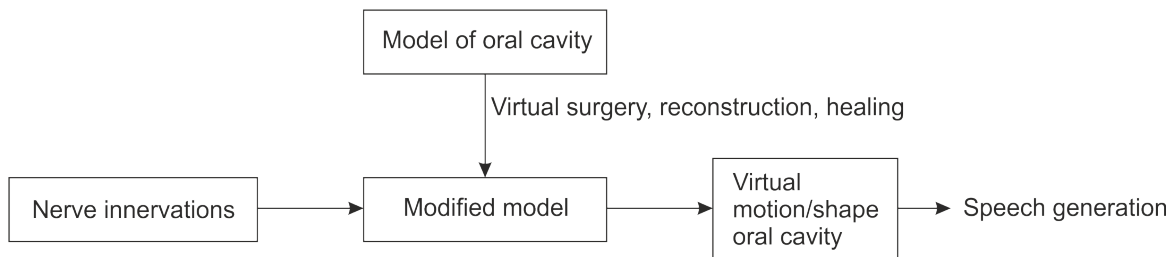


Figure 1.2: Block diagram for using the constructed model for virtual surgery.

1.3 Scope of this thesis

This thesis describes the development of a part of this project: to develop the imaging system used to record motion of the tongue in three dimensions, crucial to developing a patient-specific model. For this, a multi-camera setup is used to track marked positions of the tongue through the mouth opening. Multiple-view detection of objects makes it possible to reconstruct the three-dimensional shape. Difficulties include the narrow shape of the oral cavity, the moist environment, and the high possibility of occlusion of parts of the tongue by itself or its environment.

First a description of the tongue will be given, and the problems it imposes on tracking it. Then, the general tracking method will be proposed. In order to obtain 3D information from camera images, it is needed to derive the mathematics for camera analysis, calibration, and 3D reconstruction. This will be done in chapters 4.6 and 5. In order to deal with occlusion, a statistical model will be introduced by means of a *Principal Component*

Analysis. The next few chapters deal with the measurement protocol and the setup for use within the operating room (OR). Chapter 9 will explain the tracking process itself in detail. Finally, the method will be evaluated by performing measurements, evaluating both qualitative and quantitative results.

Chapter 2

The tongue

The tongue is an organ located in the mouth, which fulfills several functions. First of all, it is covered with taste buds which allows humans to perceive a sense of taste. A second function is transport of food in the mouth, moving it such that the jaw is capable of chewing it to bits and also taking part in the swallowing process. An additional important function is phonetic articulation, allowing one to speak. Furthermore, it also functions as a natural way of cleaning one's teeth.

2.1 Tongue layout

In contrast to many other manipulative organs of humans, the tongue does not contain bones. It consists mainly of muscles, blood vessels, and nerves. Although having no skeletal support, manipulation of the tongue is made possible due to the fact that water is effectively incompressible at physiological pressures. As muscles mainly consist of water, it means that the volume of the tongue remains roughly constant, while excitations of muscles allows one to change its shape. This makes the tongue a *muscular hydrostat*.

The tongue consists of eight muscles, divided into two groups. The intrinsic muscles, lying entirely in the tongue, allows one to change the shape. The extrinsic muscles attach the tongue to other structures and are able to reposition it. Table 2.1 gives an overview of the muscles and their function. Figure 2.1 shows how these muscles are located in and around the tongue.

Table 2.1: A list of tongue muscles and their functions.

| | Muscle | Function |
|------------------|-----------------------|--|
| Extrinsic | Genioglossus | Protusion of tongue and depression of the center |
| | Hyoglossus | Depression of tongue |
| | Styloglossus | Elevation and retraction of tongue |
| | Palatoglossus | Elevates back of tongue |
| Intrinsic | Superior longitudinal | Runs along the superior surface of the tongue. Elevates, assist retraction of, and deviates the tip of the tongue. |
| | Inferior longitudinal | Runs along the under surface of the tongue. Lines the sides of the tongue |
| | Verticalis | Located in the middle of the tongue, and joins the superior and inferior longitudinal muscles |
| | Transversus | Divides the tongue at the middle |

2.2 Constraints on tracking

Due to the tongue's properties and environmental conditions, developing a tracking method is less straightforward than for other, more external human parts. These constraints are divided up into two groups. The first one, referred to as the *visibility constraints*, deal with the fact that the visibility of the tongue is limited and can change over time. The second one, referred to as the *environmental constraints*, deal with the delicate environment conditions of the tongue, allowing only a limited selection of methods to aid the tracking. The following sources of constraints were identified:

Visibility constraints

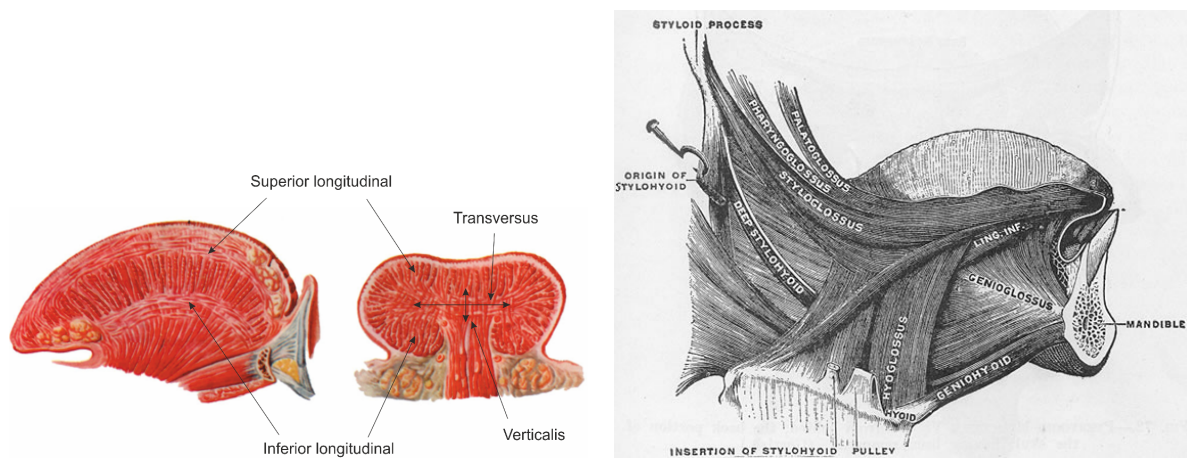


Figure 2.1: Intrinsic (left) and extrinsic (right) muscles of the tongue.

- *No fixed shape:* the tongue is a very flexible organ with no fixed shape. The use of rigid body models will not be possible.
- *Oral viewpoint:* due to the fact that the tongue is only visible through the mouth opening, only the frontal part of the tongue can be observed. The majority of the tongue will remain hidden.
- *Occlusion:* depending on the state of the tongue, regions which are visible at moment A can be occluded at moment B. Sources of occlusion include the tongue itself and other oral regions, such as the teeth.

Environmental constraints

- *Tongue surface:* the surface of the tongue, although being textured, does not offer clear landmarks which can be tracked with ease. Therefore, attaching artificial landmarks to the tongue becomes a necessity.
- *Oral climate:* the tongue is placed in a moist climate. This poses some constraints on the attachment artificial landmarks to the tongue.
- *Dangerous materials:* the tongue is a delicate organ, being the opening to the digestive system and located close to mucous membranes. This discourages the use of dangerous materials, including sharp and toxic objects.

Chapter 3

Tracking the tongue

Studying the tongue's influence in various important tasks, such as speech, is no new research topic. The role of the tongue in speech has been studied for some time now. Although it had been studied long before, Lindblom [18] was the first to propose a model in which vocal tract shapes are determined as a function of parameters such as *jaw*, *tongue-body*, *tongue-tip*, *lip height* and *width*, and *larynx height*. Maeda [10] expanded analysis of tongue motions obtained with x-ray measurements by using a linear component model, inspired by the jaw-based model of Lindblom. The complex activities of the articulatory organs were organized into a limited number of independently controllable functional blocks using a statistical analysis, similar to a principal component analysis. The state of tract shapes were then determined by the state of these blocks. Beautemps [2], inspired by this method, also used a linear component analysis to analyze a set of x-ray measurements of French sentences. Five tongue parameters (jaw height, jaw advance, tongue body, tongue dorsum and tongue tip) appeared to be enough to describe 96% of the variance of the tongue during speech. The work of Engwall [6] added an additional tongue parameter (tongue width). With a sequence of MRI images of Swedish vowels, a 3D model has been constructed. A recent work by Steiner et al. [13] describes the use of EMA sensors placed onto the tongue in order to track dedicated points in 3D. These points were then used for skeletal animation, by deforming a rig based on the measured 3D points. From these researches, it follows that tracking a set of only a few dedicated points on the tongue is enough to describe the majority of its shapes. This, however, is during speech, while the system designed during this thesis measures tongue motion with widely opened mouth, while no vowels are articulated, creating an entirely different situation.

Very few works exist on tracking the tongue shape with a camera system. Liu et al. [16] published a paper on a system using four cameras in order to construct a 3D finite element model. However, the described system works with very low frame rates (3fps) and only captures the protruded tongue, as the system is designed for medical diagnosis on images. Although not having similar applications, Liu et al. [11] have published their work of a vision system using multiple cameras to track the motion of a surgical robot. For this purpose, spherical markers applied to the robot tool were detected in the camera frames of a total of four cameras using a circular Hough transform. Reconstruction to 3D was performed using an extended Kalman filter.

In the of tracking lip tracking, many publications are based on active shape models, active appearance models and color segmentation. These methods expectantly are not very suitable due to the relative smooth texture and color of the tongue and other oral regions and due to the lack of clearly distinguishable landmarks on the tongue.

Some lip 3D reconstruction methods, such as the one used in [9] use full-field 3D shape measurement techniques. These methods project a pattern (in this specific paper a fringe pattern) onto a 3D object. The pattern as observed by the camera will be distorted as a result of relief on the 3D image. From the observed distortion, the 3D shape can be estimated. Due to the limited size of the mouth opening, as well as the occurrence of occlusion, these methods will expectantly not yield good results.

Only little has been published on marker-based tracking of the lips; this technique is primarily applied to tracking facial features. In [19] and [17], such marker-tracking systems were described, both methods using multiple-view geometry to reconstruct the 3D coordinates of markers applied onto the face.

3.1 Method

For the application of tracking the motion of the tongue, most of the methods described above are not suitable. MRI and CT are too slow and offer spatial constraints. EMA is limited by the number of sensors that can be used, but is a suitable alternative. For this report, a visual system has been chosen as it is an extension of

earlier research towards 3D modeling of the lips at this research chair. Figure 3.1 pictures a global diagram of the system that has been developed.

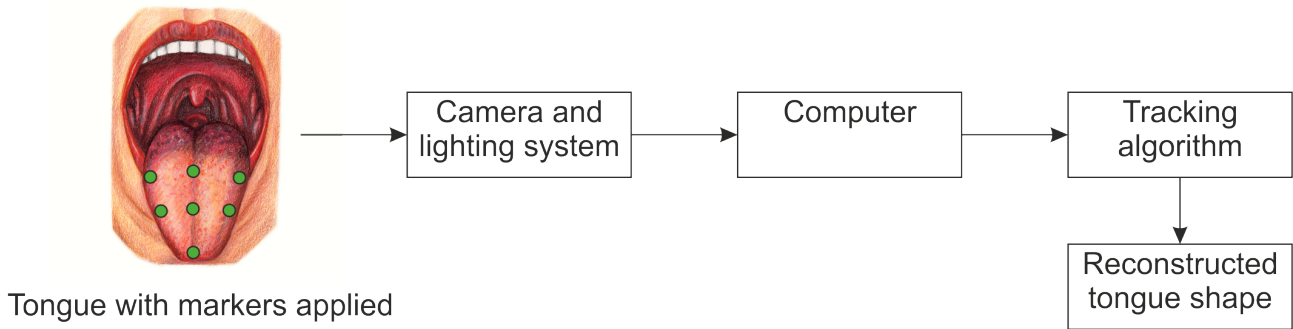


Figure 3.1: Block diagram of the vision system

As described in the previous chapter, the tongue is very flexible and has very few anatomical distinguishable landmark points. Therefore, markers are fixated onto it in order to create easy-to-track points. As follows from previous research, the tracking of few dedicated points is sufficient to describe much of the variance of the complete tongue shape. Although the parts of the tongue that can be observed by the vision system do not span the complete set of dedicated points as mentioned by the cited works on tracking the tongue (for instance, the *tongue dorsum* variable is located at the back side of the tongue and therefore not visible), the applied markers might suffice if they are sufficiently great in number. Furthermore, one has to take into account that the dedicated points mentioned in the papers describe only the variance during pronouncing vowels and not during raw tongue motion with opened mouth, creating an entirely different situation.

Measurements are performed during narcosis, when the mouth is opened as far as possible (instruments are used to take care of this), offering the cameras a good view of the tongue. A lighting system provides a sufficient amount of light in such a way that contrast between the markers and tongue is high. Multiple cameras are used so that a 3D reconstruction of the markers can be made. A number of three cameras is chosen, as this increases the chance of detecting a marker in certain critical areas, such as the sides of the tongue. As will be made clear in the remainder of the report, it is not critical for a marker to be successfully detected by at least two cameras; sometimes the successful detection of a marker in only a single camera can be sufficient. Furthermore, it will be shown that the false detection of one or several markers in each camera image may be corrected. However, the use of three cameras instead of two increases the robustness and accuracy of the system. The (synchronized) frames are sent to a computer where they are stored. A tracking algorithm can reconstruct the 3D-state of the measured markers. These calculations are performed offline.

Due to the restricted view of the tongue, it is not possible to track all surfaces of the tongue. In many cases, a good visibility of one surface requires sacrificing a good view of a secondary surface. For instance, either the top or bottom surface of the tongue can be observed well, but not both at the same time. When choosing the to-be-tracked surfaces, one should take into account that a good set of markers is able to give as much information about the shape of the tongue as possible, while being visible during most of the general tongue states. It has been decided that a good starting point is by tracking the top surface and the sides of the tongue. Especially the top surface and tongue tip are suitable due to their low chance of occlusion and expectantly high descriptive power. In addition to this, the sides of the tongue offer good points for the situation in which the tongue moves to the side.

3.2 Video processing

Now the general plan for the data acquisition is presented, an overview of the processing step can be introduced. Figure 3.2 gives an overview of the method. It starts off with a set of frames recorded by the camera system. A marker detection algorithm then detects the image coordinates of the markers. An outlier correction-algorithm is then needed to correct measurement errors possibly originating from occlusion. A principal component model (PCA model) can offer a solution. This is a statistical model of the tongue which will be further explained in chapter 6. Before being able to convert the corrected points to 3D, a camera model is needed describing the relation between points in 3D and their projection to the images as observed by the cameras. This is a result of the camera calibration step.

As will be shown further on in the report, the tongue state can also be expressed by other means than a collection of points in three dimensions. Also the PCA model variables seem to be a good method for tracking the tongue.

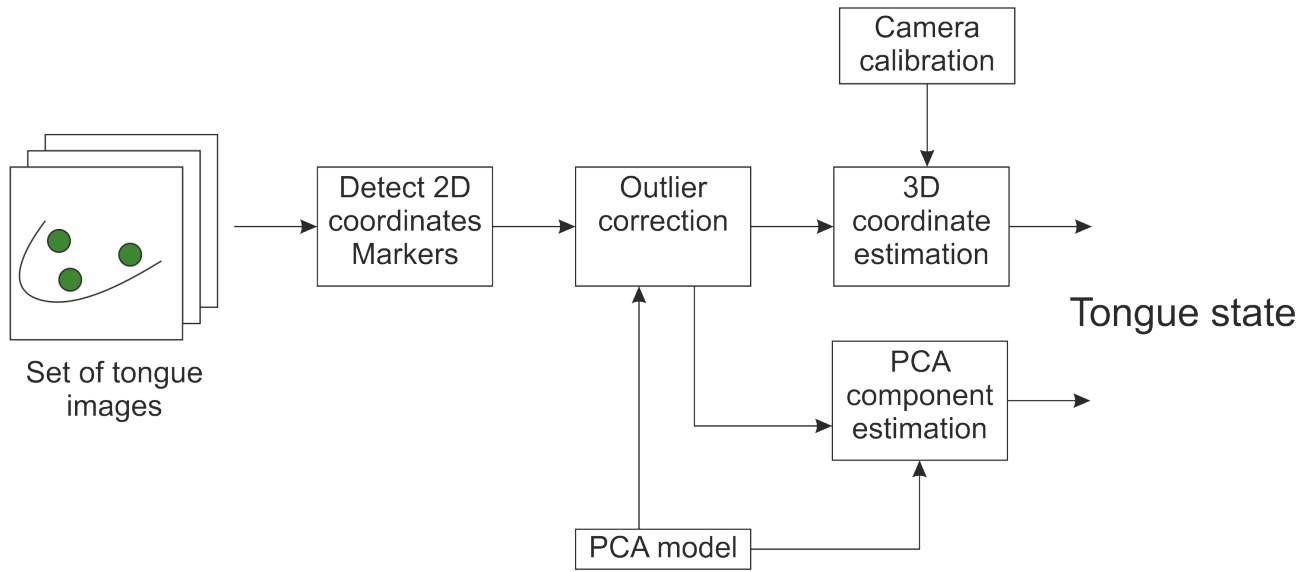


Figure 3.2: Block diagram of the processing method.

Chapter 4

Camera analysis and -calibration

Digital CCD-cameras are widely used for storing visual information. For many applications, the registration of only qualitative information is sufficient, like in the case of recording a performance of an artist. When quantitative information is desired, for instance when measuring the size of an object, one has to have information about how the camera projects 3D environment information onto a 2D image plane. The process of determining a mathematical model of such a camera is called the camera calibration process and is crucial to the application of recording and analyzing tongue motion, as this application is in need of quantitative distance measures.

4.1 Camera geometry

In figure 4.1, a general camera geometry can be seen based on the pinhole geometry. This model assumes an image plane in front of the camera center, which defines the projection plane. Graphically, projection of the 3D-world points onto the image plane can be described by the intersection of a straight line between the 3D point \mathbf{X} and the camera center and the image plane.

The camera center (or optical center) is located in the origin of the camera coordinate system \mathbf{O}_c , which is defined in such a way that its XY-plane is oriented parallel to the image plane. The z-axis of the camera coordinate system is perpendicular to the image plane and is called the *principal axis* of the camera. Its intersection with the image plane is the *principal point* \mathbf{p} . The distance between this intersection point and the camera center is the focal distance f . The camera is located in a world coordinate system, which can have a different origin \mathbf{O}_w than the camera coordinate system, as well as a different orientation. A 3D point \mathbf{X} can be projected onto the image plane. The result, denoted in image coordinates with pixels as units, can be denoted by \mathbf{x} . The image coordinate system has only 2 dimensions, and expresses the location of the projection in pixel units instead of meters. Note that the y-axis of the image plane is defined downwards, in contrast to the other coordinate systems. The model described this far does not take into account nonlinear effects, such as those caused by lens distortion. Introducing such nonlinear concepts into the model provides better accuracy, but results in a more difficult model.

4.2 Camera calibration

The objective now is to determine a mathematical description which transforms the 3D coordinate to an image coordinate. When assuming no nonlinear lens distortion, this can be described by projection P :

$$\mathbf{x} = P\mathbf{X} \tag{4.1}$$

The used vectors are generally expressed in homogeneous coordinates, making them independent of scaling:

$$\mathbf{x} = \begin{bmatrix} \alpha x \\ \alpha y \\ \alpha \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \beta X \\ \beta Y \\ \beta Z \\ \beta \end{bmatrix} \tag{4.2}$$

P is thus a 3x4 matrix. Now a distinction can be made between *internal* and *external* parameters. External parameters describe the transformation (rotation and translation) needed to express world coordinates into camera coordinates. Internal parameters describe the mapping from 3D camera coordinates to image coordinates. Equation 4.1 can then be rewritten to:

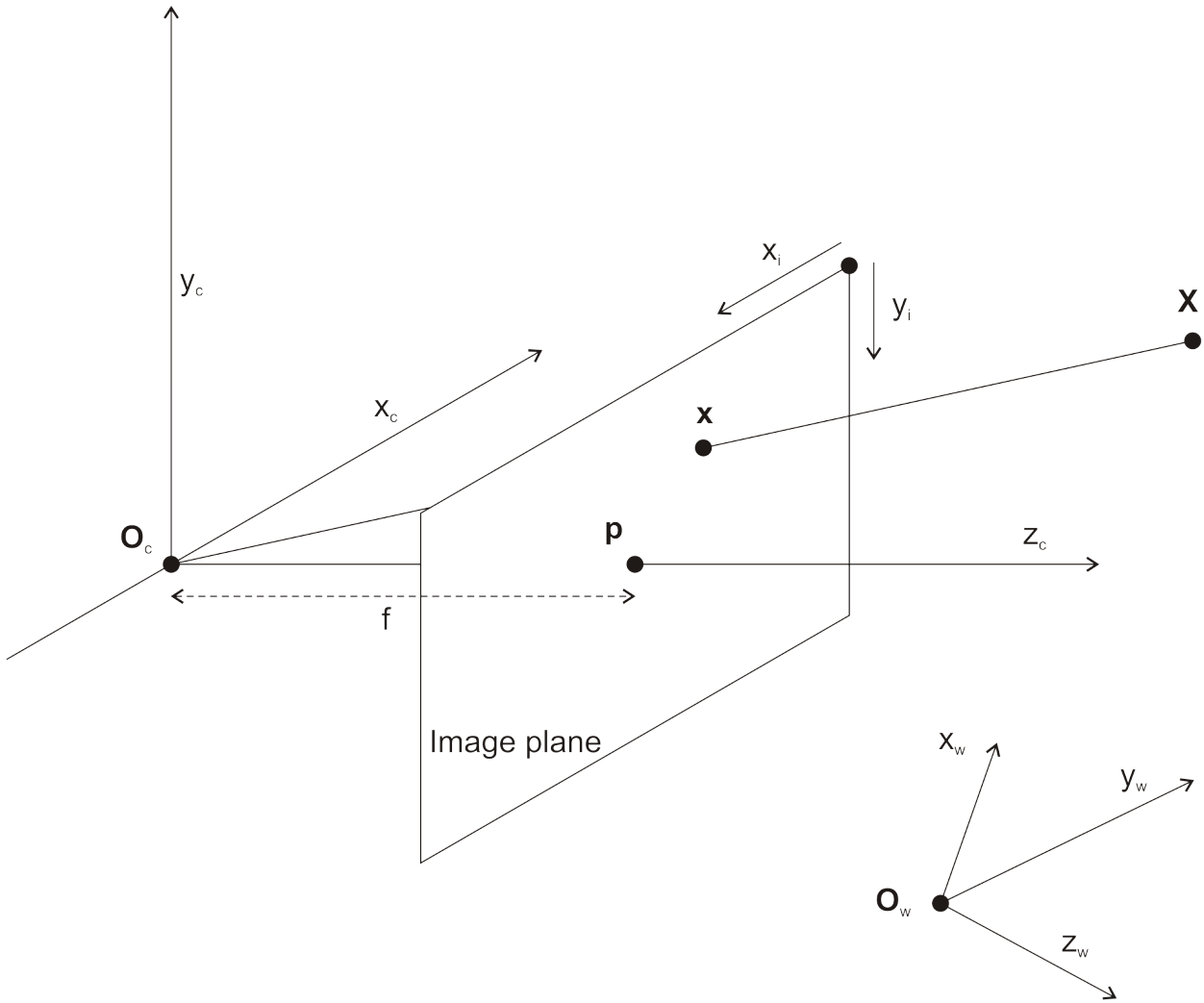


Figure 4.1: Pinhole camera geometry

$$P = P_{int}P_{ext} = K[R|\mathbf{t}] \quad (4.3)$$

Where R and \mathbf{t} respectively represent the 3x3 rotation matrix and the 3x1 translation vector. K represents the internal calibration matrix:

$$\begin{bmatrix} f & 0 & p_x \\ 0 & -f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

The resulting model is completely linear. If nonlinear lens distortion would be needed, an additional distortion step is needed which acts directly on camera coordinates:

$$\mathbf{X}_{c,dist} = f_{dist}\{\mathbf{X}_c\} = f_{dist}\{[R|T]\mathbf{X}\} \quad (4.5)$$

Where \mathbf{X}_c are the camera coordinates of the 3D-point, $\mathbf{X}_{c,dist}$ the distorted camera coordinates, and $f_{dist}\{\}$ is the nonlinear distortion model, such that:

$$\mathbf{x} = K\mathbf{X}_{c,dist} = Kf_{dist}\{[R|T]\mathbf{X}\} \quad (4.6)$$

A camera is considered calibrated if the focal distance, principal point and lens distortion parameters are known. Generally, such algorithms require point correspondences between the real world and image coordinates. In [7], an overview of several calibration methods is given, as well as their performance. These methods can be classified, such as linear versus nonlinear methods, point-based versus line-based methods and 3D versus planar point arrays. Two of these methods have been selected and tested in order to investigate if their performance is good enough for the application. These methods have been selected based on availability and the fact that they are quite distinct. The first one is known by the *Image processing toolbox for Matlab*, a free toolbox providing calibration using a series of images from planar checkerboard patterns. The method used by the toolbox needs multiple images, but is relative precise, also modeling lens distortion. The second one needs a cube and is based on the *Direct Linear Transform* algorithm, a method not modeling lens distortion. This method only needs one image from a 3D-object (in this case a cube), and offers a completely linear model.

The preference is a fully linear calibrated model over a nonlinear one, as this will save processing resources and is easier to use for Kalman tracking (which makes use of linear(ized) systems).

4.3 Calibration using checkerboard pattern

This calibration method is based on the the work of Zhang [27] and Heikkilä and Silvén [14]. A user-friendly Matlab version is available via the website of the California Institute of Technology [5]. This method requires several images of a planar checkerboard, viewed from different camera angles. A search is performed for the corners of the pattern. The known real world spacing of these corners and the measured image coordinates provide point correspondences. Lens distortion is included in the model, and is modeled as follows:

$$\mathbf{X}_{c,dist} = (1 + k_1r^2 + k_2r^4 + k_3r^6)\mathbf{X}_c + d\mathbf{X}_t \quad (4.7)$$

$$d\mathbf{X}_t = \begin{bmatrix} 2k_3xy + k_4(r^2 + 2x^2) \\ 2k_4xy + k_3(r^2 + 2y^2) \end{bmatrix} \quad (4.8)$$

Where all used variables are nonhomogeneous and $\mathbf{X}_c^T = [x, y]$.

4.3.1 Method

In order to test the performance of this calibration toolbox, a series of nine images from a planar checkerboard-pattern under various angles has been made with a resolution of 2592×3456 pixels. These were used to calibrate the internal parameters of the camera. Figure 4.2 shows one of the images used for calibration. Next to the calibration result, the calibration toolbox also gives a measure for the uncertainty of these parameters, which can be used to verify the accuracy of the estimation.

4.3.2 Results

Table 4.1 gives an overview of the calibration parameters along with the uncertainty of those parameters (standard deviation).

Especially in the case of the focal distance and focal point, the parameter uncertainty is very small. However, in the end what matters is how well the model relates to reality. For this purpose, the error between the detected

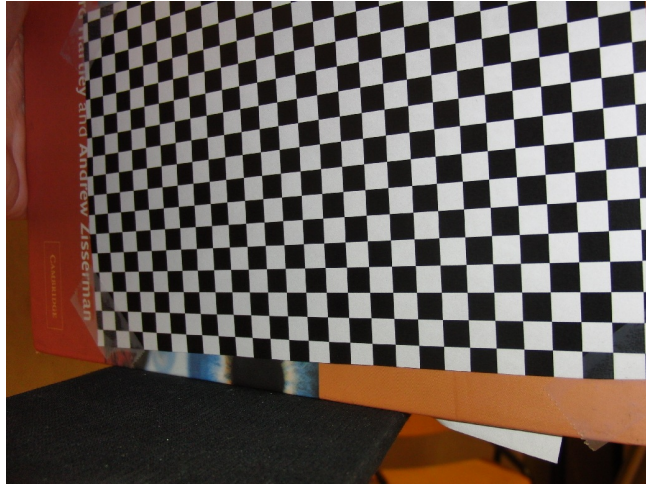


Figure 4.2: Example of checkerboard pattern for calibration purposes

Table 4.1: Calibration results by using a chessboard.

| Parameter | Value | Uncertainty σ | Relative uncertainty |
|-----------|-------------|----------------------|----------------------|
| f_1 | 3689 pixels | 1.19 pixels | 0.03% |
| f_2 | 3678 pixels | 1.16 pixels | 0.03% |
| p_x | 1871 pixels | 2.16 pixels | 0.12% |
| p_y | 1335 pixels | 1.38 pixels | 0.10% |
| k_1 | -0.15167 | 1.12E-3 | 0.74% |
| k_2 | 0.16539 | 3.44E-3 | 2.01% |
| k_3 | -0.00134 | 1.01E-4 | 7.58% |
| k_4 | 0.00478 | 1.35E-4 | 2.82% |
| k_5 | 0 | 0 | - |

crossings of the checkerboard and their projections conform the model has been studied. The mean Euclidian distance between those points is 1.264 pixels.

4.3.3 Discussion

The method delivers good calibration results with an average Euclidian error only slightly larger than 1 pixel. Although performance looks good, the method is a bit cumbersome due to the relative large amount of images that have to be taken, and due to the relative long time it takes to perform the calibration by having to select several corners in each image. Furthermore, the method provides a nonlinear calibration result, which is relative precise, but is harder and more computational expensive than linear methods. An advantage is the used calibration grid: this is fully 2D and can easily be constructed with high precision using for instance a printer.

4.4 Calibration using cube

This method is in need of only a single image from a cube, in which sufficient point correspondences can be selected (at least six correspondences are needed). The algorithm starts off with normalizing both the 3D and the image coordinates towards an average distance of respectively $\sqrt{3}$ and $\sqrt{2}$ to the center of gravity of the selected points. Before determining the camera parameters, the matrix P will be determined using the *Direct Linear Transform* algorithm [26]. For analysis, we first observe equation 4.1 again:

$$\begin{bmatrix} \alpha x_i \\ \alpha y_i \\ \alpha \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \mathbf{X} \quad (4.9)$$

Where \mathbf{p}_n^T represents the n'th row of P . Filling α into the upper two equations and rewriting yields:

$$\mathbf{0} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \end{bmatrix} \mathbf{X} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \mathbf{p}_3^T \mathbf{X} = \begin{bmatrix} -\mathbf{X}^T & \mathbf{0}^T & x_i \mathbf{X}^T \\ \mathbf{0}^T & -\mathbf{X}^T & y_i \mathbf{X}^T \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = H\mathbf{p} \quad (4.10)$$

Then, a singular value decomposition (SVD) can be applied to H :

$$H = UDV^* \quad (4.11)$$

Here, \mathbf{D} is a square, diagonal matrix with the eigenvalues of H along its diagonal. The solution corresponding to the smallest singular value is the solution \mathbf{p} . The smallest singular value corresponds with the smallest eigenvalue of H . The column of V corresponding with the specific eigenvalue is the solution to p .

After these operations, the internal and external camera parameters can be determined. By virtually drawing parallel lines and projecting these on the image plane using P , the vanishing points can be determined. From these, as described in the document *Camera calibration using cubes* [23], the focal distance and principal point can be determined, which are all needed internal parameters. The external parameters can be determined as follows:

$$P = K[R|T] \quad (4.12)$$

$$[R|T] = K^{-1}P \quad (4.13)$$

4.4.1 Method

A 3D-object with sufficient marked locations needs only a single calibration image for obtaining the camera parameters. A cube with 27 (3x3x3) markings provides more than enough of these locations. An overdetermined solution generally provides better results as the result of an error in a single point correspondence will be damped as more point correspondences are included. An image at a resolution of 2592×3456 pixels has been taken. An automatic search for the corners of the cube is not provided, and has to be performed manually (which may result in a suboptimal solution).

4.4.2 Results

The numerical results of calibration using a cube are as follows:

$$P = \begin{bmatrix} 3149.9 & -1341.2 & -2334.9 & 5.3594E5 \\ -270.17 & -3816.3 & 862.96 & 3.0323E5 \\ -0.2502 & -0.49915 & -0.82961 & 307.88 \end{bmatrix} \quad (4.14)$$

$$K = \begin{bmatrix} 3733.3 & 0 & 1812.5 \\ 0 & -3733.3 & 1291 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} -5.9138 \\ 25.245 \\ 307.88 \end{bmatrix} \quad (4.15)$$

Figure 4.3 shows the cube used for calibration, which is $10 \times 10 \times 10$ cm. It first of all shows the manually selected points. Furthermore, the known 3D-grid of the calibration cube has been projected back to the image plane, and also plotted in the image. The average error between the selected and projected points is 3.39 pixels and the standard deviation of this error is 1.72 pixels. The size of this error is acceptable, and makes the method useful for the intended application.

It is investigated how the calibration result performs in the spatial regions not covered by the point correspondences. For this reason, calibration has been performed multiple times, every time excluding a single point from the set of point correspondences. Then, using the result, the point not included in the calibration is projected to the image plane, and compared to the manually selected variant. The result of projecting each of these points in this manner can be observed in figure 4.4. The average error has risen to 4.43 pixels, with a standard deviation of 2.63 pixels. Visually seen, the error is still acceptable.

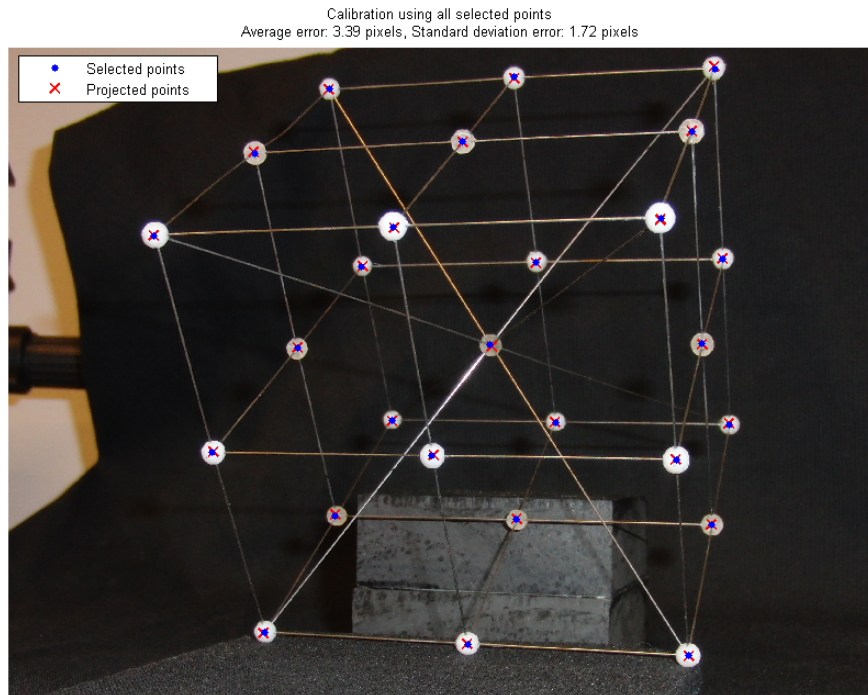


Figure 4.3: Calibration result by using a cube. Both manually selected and projected points are given.

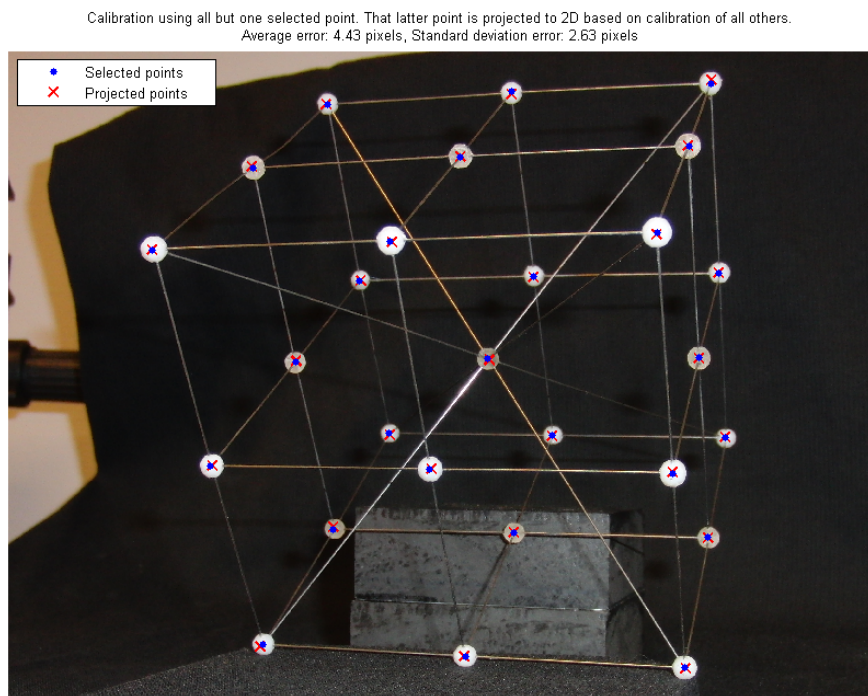


Figure 4.4: Calibration result using a cube, using all but one point for calibration, and comparing the last projected point with the manually selected one.

4.4.3 Discussion

The error of the reconstruction result is acceptable for the application. Errors are still acceptable outside of the spatial region included in the calibration process. The model is fully linear and is easy and fast to use. Furthermore, only a single image is needed for calibration. This however requires manually selection of the points in the image, which can lead to a bad calibration results if not done accurately. An additional danger of this method is that the cube is not ideal; when this deviates from its ideal shape, such as being slightly skewed, while being assumed ideal, the resulting model incorporates this non-ideality.

4.5 Comparison of methods

Numerically, it has been shown that the checkerboard calibration provides a more accurate calibration result than the cube calibration, achieving around three times more accuracy. However, this has been purely based on point correspondences of the used calibration objects. It may be that the cube, for instance, can be skewed, resulting in non-ideal calibration. In order to check if this phenomenon occurs, a calibration picture is taken from both objects with their grids aligned. Additional images are used for checkerboard calibration. Then, using both methods of calibration, the grids of both calibration objects are projected to 2D.

The result can be seen in figure 4.5. As expected, the points projected using cube calibration overlaps the cube well. It also nicely follows the shape of the checkerboard pattern within the calibration area, but it drifts away from the true pattern the further it gets from the cube, a result of local calibration. Lens distortion may be a probable cause for this phenomenon.

Calibration using the checkerboard pattern obtains the board corner coordinates very nice when including lens distortion correction, but performs bad when distortion correction is not included. Projecting the coordinates of the cube yields worse results than expected, especially in the upper region. A probable cause is the fact that most of the set of checkerboard images were taken in a different spatial region than the region in which the cube is located. Furthermore, this may be a result of the zooming nature of the lens, automatically adapting itself to different situations and therefore not providing a consistent model. Furthermore, it may be that the cube deviates from its ideal shape.

4.6 Conclusion

Two methods were investigated to calibrate a camera. One is based on a two-dimensional checkerboard pattern, requiring multiple images, and including nonlinear lens distortion correction. The second method is based on a three-dimensional cube, requiring only one image, being fully linear, but not including lens distortion correction.

Numerical results show that the checkerboard method performs better, with almost a factor three in accuracy (see table 4.2). However, both methods show enough precision to be acceptable.

Table 4.2: Average reconstruction error of the points selected in the image after calibration.

| Calibration method | Average reconstruction error (Euclidian distance) |
|---------------------|---|
| Checkerboard method | 1.26 pixels |
| Cube method | 3.39 pixels |

The process of calibration is much easier when using the cube, as this only requires a single image compared to multiple for the checkerboard method. Furthermore, processing of the images also takes longer in the latter case.

The linearity of the cube calibration makes it very easy and fast to use. However, one should take into account that the calibration result is only valid within the calibrated area. That is, the area spanned by the calibration cube. When this area grows, as a consequence the linearization result will be less accurate. This phenomenon does not occur when using the checkerboard calibration. This nonlinear model, if trained well, will yield good results everywhere. A good training will include many images taken across the complete field of view, at different angles.

The main tradeoff between these methods is accuracy but complexity versus ease of use. As accuracy will not be the limiting factor, the choice for a linear model based on cube calibration has been taken.

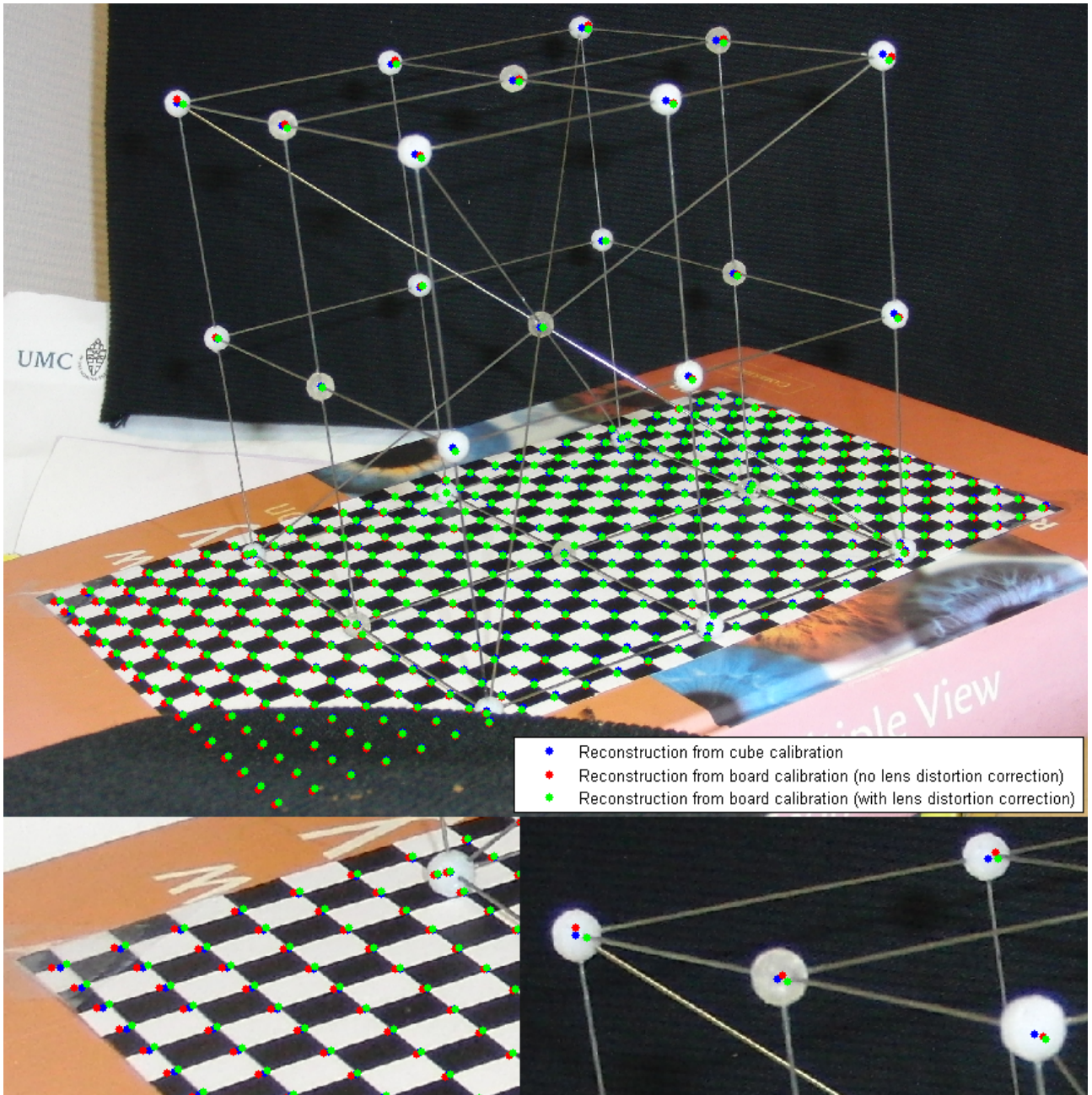


Figure 4.5: Objects reconstructed using calibration based on the other object.

Chapter 5

Multi-camera analysis

A single camera can only estimate the position of an object in two dimensions, as no depth information can be directly measured from a frame without a good model of the measured object. Estimation of the 3D position of an object is possible using multiple cameras which are displaced with respect to each other, as illustrated in figure 5.1. This figure shows two cameras with each their estimation of a 3D-point. This estimation is denoted by a line cast from the camera, surrounded by an uncertainty region in yellow. This region is diverging, as a measurement error in the recorded image is more significant at larger distances from the camera. A secondary camera has an equal estimation and uncertainty shape. When combining the information of these cameras, a much smaller uncertainty region (in orange) is the result, providing depth information about the measured object.

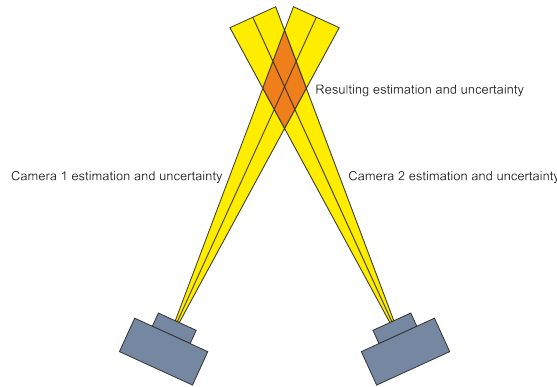


Figure 5.1: Estimation using a 3D point using two cameras, and the resulting uncertainty region.

There are many algorithms for reconstructing a 3D-point from two or more image measurements. In this chapter, a distinction is made between estimation of a single 3D point and a multiple points simultaneously. Furthermore, a distinction is made between a single 3D reconstruction and a tracking algorithm. The latter requires a stream of frames rather than a single set of images.

5.1 3D estimation of a single point

In this problem, a three-dimensional vector has to be estimated from several two-dimensional vectors. For this purpose we can use the calibration result as determined during the camera calibration phase. The linear calibration result offers a solution for solving the estimation problem. For each image taken, as defined equation 4.1, the following relation holds:

$$\begin{bmatrix} \alpha x \\ \alpha y \\ \alpha \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \beta X \\ \beta Y \\ \beta Z \\ \beta \end{bmatrix} \quad (5.1)$$

After filling in $\alpha = p_3^T \mathbf{X}$ in the upper 2 equations, where p_n^T represents the n'th row of P , and rewriting this we get:

$$\mathbf{0} = \begin{bmatrix} xp_3^T - p_1^T \\ yp_3^T - p_2^T \end{bmatrix} \begin{bmatrix} \beta X \\ \beta Y \\ \beta Z \\ \beta \end{bmatrix} \quad (5.2)$$

Then, when choosing $\beta = 1$ and bringing the fourth variable of \mathbf{X} to the left side of the equation, the following holds:

$$\begin{bmatrix} p_{14} - xp_{34} \\ p_{24} - yp_{34} \end{bmatrix} = \begin{bmatrix} xp_3^{T'} - p_1^{T'} \\ yp_3^{T'} - p_2^{T'} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5.3)$$

Note that $p_n^{T'} = [p_{n1} \ p_{n2} \ p_{n3}]$ represents the first three variables of the n 'th row of P . We now have a direct transform between the 3D world coordinates and the 2D image coordinates. One image, however, does not offer enough information to estimate depth information. This can also be seen in equation 5.3, where three variables are to be estimated from a vector containing 2 variables. Using two images, the left-hand part of the equation contains four variables such that the system will be overdetermined and thus solvable (assuming that the cameras are placed sufficiently apart). Mathematically, this can be described by simply adding more equations:

$$\begin{bmatrix} p_{14} - xp_{34} \\ p_{24} - yp_{34} \\ \overline{p}_{14} - \overline{xp}_{34} \\ \overline{p}_{24} - \overline{yp}_{34} \end{bmatrix} = \begin{bmatrix} xp_3^{T'} - p_1^{T'} \\ yp_3^{T'} - p_2^{T'} \\ \overline{xp}_3^{T'} - \overline{p}_1^{T'} \\ \overline{yp}_3^{T'} - \overline{p}_2^{T'} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5.4)$$

The parameters and variables belonging to the secondary equations are overlined. The expression in the current form is the equivalent of $\mathbf{z} = H\mathbf{X}$. A least squares estimator now is able to solve this problem for \mathbf{X} .

5.2 3D estimation of multiple points

The method described is able to estimate point-by-point. However, by extending the matrices it is possible to perform the estimation of all points in a single step. For this, it is needed to define homogeneous and non-homogeneous representations for the two- and three-dimensional vectors:

$$\mathbf{u} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_C \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_C \end{bmatrix} \quad \mathbf{u}_h = \begin{bmatrix} \text{diag}(\vec{\alpha}_1)\mathbf{x}_1 \\ \text{diag}(\vec{\alpha}_2)\mathbf{x}_2 \\ \vdots \\ \text{diag}(\vec{\alpha}_C)\mathbf{x}_C \\ \text{diag}(\vec{\alpha}_1)\mathbf{y}_1 \\ \text{diag}(\vec{\alpha}_2)\mathbf{y}_2 \\ \vdots \\ \text{diag}(\vec{\alpha}_C)\mathbf{y}_C \\ \vec{\alpha}_1 \\ \vec{\alpha}_2 \\ \vdots \\ \vec{\alpha}_C \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} \quad \mathbf{U}_h = \begin{bmatrix} \text{diag}(\vec{\beta})\mathbf{X} \\ \text{diag}(\vec{\beta})\mathbf{Y} \\ \text{diag}(\vec{\beta})\mathbf{Z} \\ \vec{\beta} \end{bmatrix} \quad (5.5)$$

As can be seen, \mathbf{u} and \mathbf{u}_h include the coordinate vectors \mathbf{x}_c and \mathbf{y}_c , where $c \in C$, c being the camera identifier and C the number of cameras. \mathbf{x}_c and \mathbf{y}_c are of length N , the number of markers. It may be clear that these represent the pixel coordinates of the N markers in the frames recorded by the cameras. $\vec{\alpha}_c$, also of length N , is the vector used to create the homogeneous representation in a similar way as done in the single-point situation. $\text{diag}(\vec{\alpha}_c)$ is used to create a matrix with the elements of $\vec{\alpha}_c$ arranged across its diagonal, in order to describe element-wise vector multiplications.

\mathbf{U} and \mathbf{U}_h are composed of the three-dimensional coordinate vectors \mathbf{X} , \mathbf{Y} and \mathbf{Z} , all of length N . $\vec{\beta}$ is the N -dimensional vector used to make a homogeneous representation.

Now, a multi-marker equivalent of equation 5.1 can be created:

$$\mathbf{u}_h = P_{ext} \mathbf{U}_h = \begin{bmatrix} p_{11}^1 \mathbf{I}_N & p_{12}^1 \mathbf{I}_N & p_{13}^1 \mathbf{I}_N & p_{14}^1 \mathbf{I}_N \\ p_{11}^2 \mathbf{I}_N & p_{12}^2 \mathbf{I}_N & p_{13}^2 \mathbf{I}_N & p_{14}^2 \mathbf{I}_N \\ \vdots & \vdots & \vdots & \vdots \\ p_{11}^C \mathbf{I}_N & p_{12}^C \mathbf{I}_N & p_{13}^C \mathbf{I}_N & p_{14}^C \mathbf{I}_N \\ p_{21}^1 \mathbf{I}_N & p_{22}^1 \mathbf{I}_N & p_{23}^1 \mathbf{I}_N & p_{24}^1 \mathbf{I}_N \\ p_{21}^2 \mathbf{I}_N & p_{22}^2 \mathbf{I}_N & p_{23}^2 \mathbf{I}_N & p_{24}^2 \mathbf{I}_N \\ \vdots & \vdots & \vdots & \vdots \\ p_{21}^C \mathbf{I}_N & p_{22}^C \mathbf{I}_N & p_{23}^C \mathbf{I}_N & p_{24}^C \mathbf{I}_N \\ p_{31}^1 \mathbf{I}_N & p_{32}^1 \mathbf{I}_N & p_{33}^1 \mathbf{I}_N & p_{34}^1 \mathbf{I}_N \\ p_{31}^2 \mathbf{I}_N & p_{32}^2 \mathbf{I}_N & p_{33}^2 \mathbf{I}_N & p_{34}^2 \mathbf{I}_N \\ \vdots & \vdots & \vdots & \vdots \\ p_{31}^C \mathbf{I}_N & p_{32}^C \mathbf{I}_N & p_{33}^C \mathbf{I}_N & p_{34}^C \mathbf{I}_N \end{bmatrix} \mathbf{U}_h = \begin{bmatrix} P_{ext_{11}} & P_{ext_{12}} & P_{ext_{13}} & P_{ext_{14}} \\ P_{ext_{21}} & P_{ext_{22}} & P_{ext_{23}} & P_{ext_{24}} \\ P_{ext_{31}} & P_{ext_{32}} & P_{ext_{33}} & P_{ext_{34}} \end{bmatrix} \mathbf{U}_h \quad (5.6)$$

Here, \mathbf{I}_N is a $N \times N$ identity matrix, and is p_{jk}^i the element on the j 'th row and k 'th column of the calibration matrix of the i 'th camera. For notation purposes, P_{ext} is divided into sub-matrices.

Now, following a similar approach as in the previous section, we arrive at the following expression:

$$\sum_{row} \left(\begin{bmatrix} P_{ext_{14}} \\ P_{ext_{24}} \end{bmatrix} - \text{diag}(\mathbf{u}) \begin{bmatrix} P_{ext_{34}} \\ P_{ext_{34}} \end{bmatrix} \right) = \left(\text{diag}(\mathbf{u}) \begin{bmatrix} P_{ext_{31}} & P_{ext_{32}} & P_{ext_{33}} \\ P_{ext_{31}} & P_{ext_{32}} & P_{ext_{33}} \end{bmatrix} - \begin{bmatrix} P_{ext_{11}} & P_{ext_{12}} & P_{ext_{13}} \\ P_{ext_{21}} & P_{ext_{22}} & P_{ext_{23}} \end{bmatrix} \right) \mathbf{U} \quad (5.7)$$

The result is somehow different because we now deal with sub-matrices instead of scalars. The operator $\sum_{row}(A)$ means a summation along the rows of matrix A . The operator $\text{diag}(\mathbf{u})$ creates a matrix with the elements of \mathbf{u} arranged along its diagonal. When comparing this to equation 5.3, similarities can be observed. Also this expression can be seen as the equivalent of $\mathbf{a} = B\mathbf{U}$, which can be solved using a least squares estimator.

5.3 3D estimation using a Kalman filter

Now it is possible to reconstruct the three-dimensional representation of the marker positions given the measured marker coordinates in the separate images. A Kalman filter can be used for filtering noise from measurements, providing an estimated state based on weighting of the previous state and on the measurement performed. Furthermore it can predict the next state of the system. The Kalman filter is optimal in the sense that it minimizes the uncertainty of the estimation result in linear, Gaussian systems. Although measurements in this system are not Gaussian distributed, a Kalman filter may still provide good results. First, a start will be made by defining a state vector. This vector includes the 3D marker positions and -velocity, but no acceleration as this makes the system unnecessary slow. The state vector will then be as follows:

$$\mathbf{s}_u = \begin{bmatrix} \mathbf{U} \\ \mathbf{U}_v \end{bmatrix} \quad (5.8)$$

Here, \mathbf{U} is the position vector as defined in the previous section, and \mathbf{U}_v is its corresponding equally sized velocity vector. Equation 5.7 can be used to derive a measurement model, for which it can be rewritten to:

$$\mathbf{a}(\mathbf{u}, P_{ext}) = B(\mathbf{u}, P_{ext}) \mathbf{U} \quad (5.9)$$

Now the step towards a measurement model can be made. For this purpose, a new variable $\mathbf{z}_u(i)$ is defined as the measurement of $\mathbf{u}(i)$. When introducing the new state vector, adding time-dependency and taking into account measurement noise, we arrive at the following expression:

$$\mathbf{a}(\mathbf{z}_u(i), P_{ext}) = [B(\mathbf{z}_u(i), P_{ext}) \mathbf{0}] \mathbf{s}_u(i) + \mathbf{n}_a(i) \quad (5.10)$$

Here, $\mathbf{0}$ is a matrix of zeros of the same size as $B(\mathbf{z}_u(i), P_{ext})$. Although the formula has the general shape of a Kalman form, it is unusual that the measurement matrix is dependent on the measured image locations, while the measurement vector is not the 'real' measurement at all! Still, the mathematics remain valid and can be used very well.

The noise $\mathbf{n}_a(i)$ can be described by a covariance matrix C_a . This is valid under the assumption that the noise is Gaussian distributed. This is not the case for template matching, but the noise description may be a

sufficiently accurate approximation. Assuming that the noise is uncorrelated the following can be constructed from $\mathbf{a}(\mathbf{z}_u(i), P_{ext})$:

$$C_a = \text{diag} \left(\sigma_{z_u} \sum_{\text{row}} \begin{bmatrix} P_{ext_{34}} \\ P_{ext_{34}} \end{bmatrix} \right)^2 \quad (5.11)$$

Here, σ_{z_u} is the standard deviation of the error of localizing the marker in the image expressed in pixels. A system matrix must be constructed to model the dynamics of the marker. For this, a set of autoregressive equations is used:

$$\mathbf{S}_u(i) = F_u \mathbf{S}_u(i-1) + \mathbf{w}_u(i) \quad (5.12)$$

Where F_u is the system matrix, and $\mathbf{w}_u(i)$ is a representation of the time-dependent system noise which can be described by covariance matrix C_u :

$$F_u = \begin{bmatrix} \mathbf{I}_{3N} & T\mathbf{I}_{3N} \\ \mathbf{0}_{3N} & \mathbf{I}_{3N} \end{bmatrix}, C_u = \begin{bmatrix} \mathbf{0}_{3N} & \mathbf{0}_{3N} \\ \mathbf{0}_{3N} & \sigma_v^2 \mathbf{I}_{3N} \end{bmatrix} \quad (5.13)$$

Here, T represents the time between samples and σ_v the standard deviation of the error of the velocity. Furthermore, \mathbf{I}_{3N} represents a $3N \times 3N$ identity matrix. The dynamics of the system make the position dependent on the previous position and velocity, and the velocity dependent on the previous velocity. System noise is added to the velocity. This noise incorporates the uncertainty due to the incomplete modeling of the tongue by not modeling higher order derivatives.

Now a full description of the system has been given. The standard form of the discrete Kalman filter is not suitable for this problem. For an estimation, the standard form requires an estimation of the measurement vector. However, as the measurement matrix $z(i)$ is dependent on the true measurement itself, this is quite problematic. Hence, a different but fully equivalent form of the Kalman update is used, denoted by the following equations:

$$C_{est}(i) = (C_{pred}^{-1}(i) + H^T C_n^{-1} H)^{-1} \quad (5.14)$$

$$\mathbf{X}_{est}(i) = C_{est}(i) (C_{pred}^{-1}(i) \mathbf{X}_{pred}(i) + H^T C_n^{-1} \mathbf{z}(i)) \quad (5.15)$$

The Kalman prediction is still in its standard form:

$$\mathbf{x}_{pred}(i+1) = F \mathbf{x}_{est} \quad (5.16)$$

$$C_{pred}(i+1) = F C_{est}(i) F^T + C_w \quad (5.17)$$

The Kalman filter as proposed in this chapter is used as a state estimator for the facial markers only. The tongue markers are also tracked by a Kalman filter, but by one whose dynamics include a deformable 3D model as will be explained in the next chapter.

Chapter 6

Principal Component Model

The marker-tracking algorithm encounters some problematic situations in which the exact location of the marker can not be estimated. These occur in the situation in which a marker is occluded in the image of all cameras. Also when a marker is only visible in one camera, no depth information can be estimated. Additionally, measurement errors are a problem, which can cause the system to reconstruct a false tongue shape. A solution is needed which can detect outliers and occluded markers and correct their measured locations.

When looking at a set of measurements of markers on the tongue, one can observe that although each marker is described by multiple coordinate variables, their position is related to the neighboring markers. In other words, the coordinates of the separate markers are correlated with each other. This would mean that the state of the set of markers can be described with less variables than required to describe the coordinates of each marker separately. Such a representation would not only reduce the amount of variables needed to describe the tongue state, but this reduced set of variables would be able to explain only a subset of the tongue states which could be described with the original representation. If done well, this representation would not allow the occurrence of physically impossible tongue shapes.

There are several techniques known to perform such a transformation to a reduced set of variables. The one selected for this thesis is the *principal component analysis* (PCA). This is a mathematical procedure which transforms a set of observations towards a set of linearly uncorrelated variables. Mathematically, this relation is given by the following equation:

$$\mathbf{U} = \overline{\overline{\mathbf{U}}} + V\mathbf{y} \quad (6.1)$$

Here, \mathbf{U} is the observation vector, in this case the 3D tongue marker positions, $\overline{\overline{\mathbf{U}}}$ the mean shape, V is the matrix containing the vectors spanning the linear subspace (called *principal component coefficients*), and \mathbf{y} is the principal component weight vector. Each column of V represents a linear displacement of the data vector, where the amount of displacement is determined by vector \mathbf{y} . Varying the weight of a single principle component vector can give insight in the behavior of the observation vector.

A principal component analysis transforms the data to a new coordinate system such that the eigenvectors of the data are aligned along the axes. With other words, the weight vector directly determines the magnitude of the different eigenvectors of the data. As the eigenvectors of a system are uncorrelated, it may seem that less variables can be used to describe the data than in a situation in which the different variables are not uncorrelated.

It may be clear that a desired PCA model describes the three-dimensional shape of the tongue, meaning the observation vector \mathbf{U} consists of the nonhomogeneous 3D-representation of the tongue markers. The process of obtaining $\overline{\overline{\mathbf{U}}}$ and V is called the PCA training stage.

6.1 Degrees of freedom

A good PCA model is obtained by training onto a set of 3D tongue shapes. These shapes can be taken from stereo-images of a tongue, after reconstructing the 3D-shape. The PCA analysis can then be performed in order calculate the desired principal components, describing the different tongue shapes. These components, describing change in shape of the tongue such as widening, curling and left-to-right orientation will be referred to as *shape components*.

However, one has to take into account the possibility that in different image sets, the camera system may have been calibrated differently, that the person has a different head orientation or that more than one person has been the donor of such tongue images. This results in 3D shapes which are displaced and rotated with respect to each other, which is not a desired situation. These problems also play an important role for the

situation in which the PCA model is actually used in the tracking process; the person may be different, the camera setup may be different and the head orientation may be different.

Therefore, a normalization of the 3D shapes with respect to rotation and translation has to be performed before the actual training of the PCA model, and also before using the trained model during tracking. Figure 6.1 visually illustrates this.

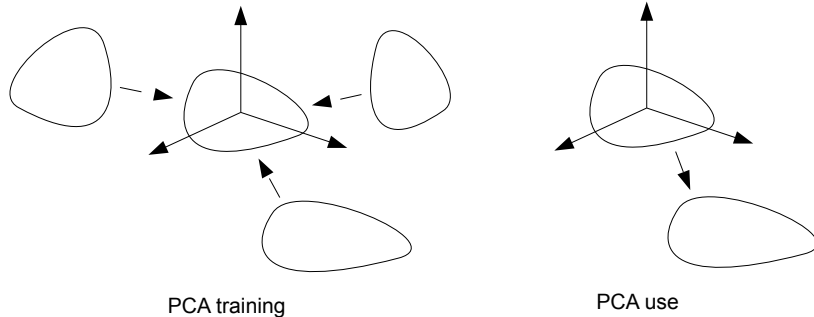


Figure 6.1: Before training the PCA model, 3D shapes of the tongue should be normalized with respect to rotation and translation (left). Before using the PCA onto during tracking, the PCA model should be rotated and translated to the subjects' head position (right).

Facial markers offer a solution for obtaining the current rotation of the head both during training and use. They can be placed onto anatomical clear locations guaranteeing some reproducibility. However, this does not solve the translational problem as each person has different facial proportions, resulting in no reproducible absolute marker placement with respect to the tongue. A solution to this problem is by including free translation vectors (three in total, for each of the directions) in the PCA model, allowing the translation to be estimated during tracking. Also addition of a free scaling factor of the tongue is possible this way. These added components will be referred to as the *general components*.

6.2 Training the PCA model

A general method for training the PCA model makes use of a statistical approach. One has to take several observations of the data vector which well reflects the different states the data vector can adopt. A good set includes as much unique states of the data vector as possible, taking into account statistical relevance. Then, by performing an eigenvalue decomposition onto the variance of the different variables of the system, the linear components are subtracted.

For the application of the tongue, a good set includes as many tongue shapes as possible. To ensure modeling the variation between different persons, the set should include data taken from different persons. Data should be in the form of reconstructed 3D tongue shapes, corrected by orientation. Care must be taken to the statistical relevance of the states; states occurring less frequently in reality should be less occurring in the training set. Illustrating an extreme situation: when many 'extreme' tongue states are recorded and are dominant along the training set, the primary subtracted principal components will be focused on especially describing these shapes, which in reality will not occur that frequently.

The process of subtracting a PCA model is illustrated by the diagram in figure 6.2, and the steps are further explained in figure 6.3.

There are several possibilities of PCA component extraction. Well-known methods make use of eigenvector decomposition and singular value decomposition. Figure 6.4 gives the pseudo-code of this process.

Step 6 in this process not only involves the reduction of dimensionality, but also results in loss of descriptive power. This means that the set of training shapes cannot be fully reconstructed anymore. However, it is still possible to approximate it to a certain extent. The total squared error over all the training data made by this approximation is:

$$\sum_{t=1}^T (\mathbf{U}^t - \tilde{\mathbf{U}}^t)^2 = (T - 1) \sum_{m=M+1}^{3N} \lambda_m \quad (6.2)$$

Where $\tilde{\mathbf{U}}_t$ is the reconstructed approximation of the t 'th training shape, and λ_m is the eigenvalue corresponding to the m 'th eigenvector.

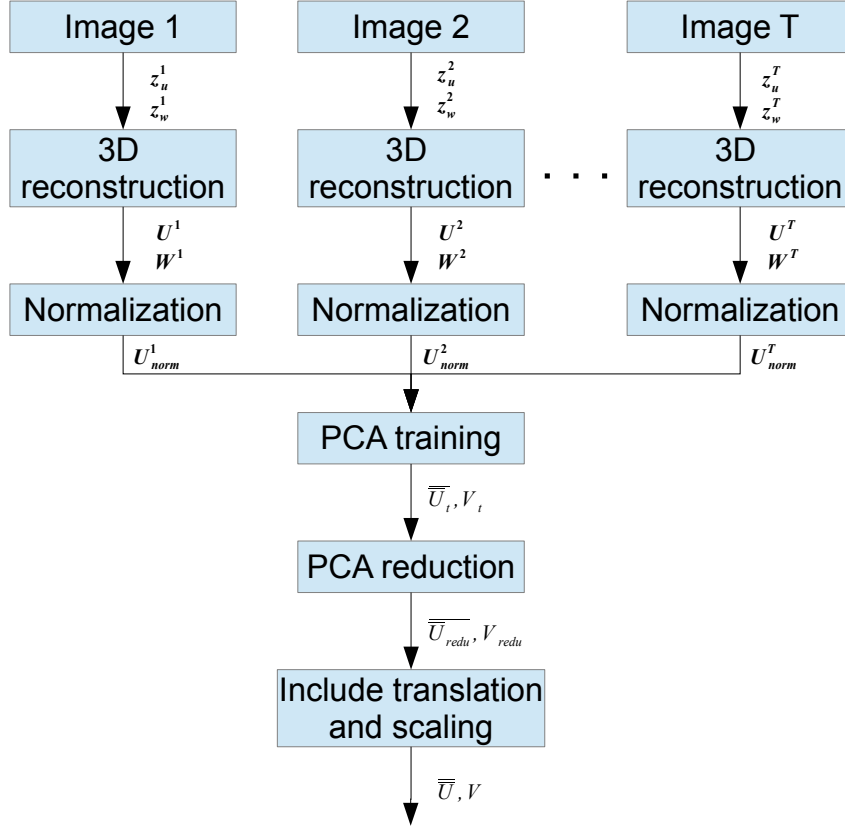


Figure 6.2: Block diagram for training the PCA model.

1. Take a good training collection of T image pairs/triples, of a tongue containing N markers. This should include data of different persons.
2. From each image set $t \in T$ select the set of facial markers \mathbf{z}_w^t and the set of tongue markers \mathbf{z}_u^t .
3. Make a 3D reconstruction of these sets, arriving at the 3D facial markers \mathbf{W}^t and 3D tongue markers \mathbf{U}^t .
4. Normalize all sets
 - From the facial markers, estimate the horizontal face unit vector \mathbf{e}_x .
 - From the facial markers, estimate the vertical face unit vector \mathbf{e}_y .
 - Estimate the face-orthogonal unit vector $\frac{\mathbf{e}_x \times \mathbf{e}_y}{\|\mathbf{e}_x \times \mathbf{e}_y\|}$. (\mathbf{e}_x and \mathbf{e}_y might not be orthogonal)
 - Calculate the rotation R with respect to the xy-plane of the world coordinate system (using for instance Horns' method [8]).
 - Calculate the center of gravity COG of \mathbf{U}^t .
 - Normalize the coordinates: $\mathbf{U}^t = R^{-1}(\mathbf{U}^t - COG)$
5. Perform general PCA analysis to arrive at $\bar{\mathbf{U}}_t$ and V_t .
6. Reduce the dimensionality by taking a select number of columns of V_t
7. Include the general PCA components:
 - Excluding scaling: $\bar{\mathbf{U}} = \bar{\mathbf{U}}_{redu}$, $V = [V_{redu} \mathbf{t}_x \mathbf{t}_y \mathbf{t}_z]$ such that $\mathbf{U} = \bar{\mathbf{U}} + V \begin{bmatrix} \mathbf{y} \\ a_x \\ a_y \\ a_z \end{bmatrix}$
 - Including scaling: $\bar{\mathbf{U}} = \mathbf{0}$, $V = [V_{redu} \mathbf{t}_x \mathbf{t}_y \mathbf{t}_z \bar{\mathbf{U}}_{redu}]$ such that $\mathbf{U} = V \begin{bmatrix} \mathbf{y} \\ a_x \\ a_y \\ a_z \\ s \end{bmatrix}$

Figure 6.3: Pseudo-code for training a PCA model

1. Calculate the mean shape $\bar{\bar{\mathbf{U}}}$ by averaging along the variables of all observations.
2. For all observations, calculate the deviation from this mean by subtracting $\bar{\bar{\mathbf{U}}}$.
3. Calculate the covariance matrix from these residuals.
4. Perform an eigenvector-decomposition on this covariance matrix. This results in a $3N \times T$ matrix, N being the number of tongue markers.
5. Order the eigenvectors in descending order according to their corresponding eigenvalues.
6. Take only the first M columns, where $M < T$, which is the dimension-reduction step. The resulting matrix is V_{redu} .

Figure 6.4: The PCA component subtraction process

6.3 Training with incomplete data

The training method as described in the previous section works well when considering every 3D tongue shape can be obtained with high precision. It however is quite challenging to acquire a good and complete set of 3D tongue shapes to begin with. This is caused by the fact that the 3D tongue shapes are acquired in the same way as the tracking algorithm is, namely based on images of multiple camera's taken through the mouth opening. In order to span most tongue shapes, many measurements will involve occlusion of markers. One way to correct for this, is by letting one or several users estimate the position of the occluded markers, this way completing the incomplete set. However, a method independent of human intervention is desired. Such an approach, further referred to as the *gappy PCA method*, has been used before for marred eigenfaces [15] and is more generally described in *Bayesian reasoning and machine learning* [4].

The method described by the latter source was used, as the author provides a working Matlab script of this method on its website. The training of the PCA model as described in the *gappy PCA* section takes a slightly different approach compared to the method described earlier on in this chapter. Rather than using an eigenvalue-decomposition or singular value decomposition in order to obtain the eigenvectors, the method focuses on an iterative optimization scheme using a limited amount of PCA components in order to minimize the squared error between the measured marker locations and the reconstructed PCA state. It starts off with the following PCA representation:

$$\tilde{U}_i^t = \sum_{j=1}^M y_j^t v_i^j \quad (6.3)$$

Where \tilde{U}_i^t represents the reconstructed i 'th variable of the t 'th tongue shape of the trainingsset. y_j^t represents the weight of the j 'th PCA component of the t 'th tongue shape of the trainingsset. v_i^j represents the i 'th variable of the j 'th PCA component vector. Note that the mean shape is not included in this representation. Now the basis for defining and optimizing the PCA model is defined by the error between the trainingsset and its reconstruction by the limited amount of PCA components:

$$E(\mathbf{V}, \mathbf{Y}) = \sum_{t=1}^T \sum_{i=1}^D \gamma_i^t \left[U_i^t - \sum_{j=1}^M y_j^t v_i^j \right]^2 \quad (6.4)$$

Here, \mathbf{V} and \mathbf{Y} represent respectively the set of PCA component vectors and the weights vectors of the shapes as contained in the trainingsset. γ_i^t represents a mask, equal to 0 if the i 'th variable of the t 'th observation is occluded and 1 otherwise. This means that the error function is only dependent on non-occluded variables. When minimizing this error function, the algorithm will thus only take into account the non-occluded markers. An iterative scheme is used to minimize the error function, which can be seen in figure 6.5.

As mentioned before, this method does not include a mean shape. However, when scaling in tongue size is desired, this poses a problem. This can be solved by calculating the mean shape of the tongue by the average of the non-occluded markers along all observations beforehand. This can be set as the mean shape, and subtracted from the data set before estimating a set of PCA components.

6.4 Training results

The influence of PCA components on the model can be visualized by setting all weights of the model to 0, and by varying a single one. Training has been performed based on a set originating from a movie involving tongue

1. Initialize the coordinate values of the occluded markers: set them to the average of the corresponding non-occluded variables along all observations.
2. Define the initial PCA model by use of a singular value decomposition onto the trainingset (note that also the earlier-described eigenvalue decomposition can be used)
3. Minimize the error function with respect to \mathbf{Y} : Differentiate $E(\mathbf{V}, \mathbf{Y})$ with respect to y_j^t and set this equal to 0. Then solve the expression for y_j^t .
4. Minimize the error function with respect to \mathbf{V} : Differentiate $E(\mathbf{V}, \mathbf{Y})$ with respect to v_i^j and set this equal to 0. Then solve the expression for v_i^j .
5. If the error is above a threshold and the maximum number of iterations are not reached yet, go back to 3.

Figure 6.5: The process of gappy PCA component subtraction, in which occluded markers can be selectively ignored.

motion including in-and outward motion and left-to-right motion. One in five frames have been used, with a total of 40 frames. The PCA model has been trained in two ways: the first one involving training using the general PCA algorithm, where the coordinates of the occluded markers have been estimated by a human. The second method involved training using the gappy PCA algorithm. The results can be seen in figures 6.6 and 6.7.

The training results are quite similar. Small differences can be observed, but the main change in shape does not differ. The source differences can be caused by a difference in mean shape. Furthermore, the fundamental component subtraction is quite different, as the gappy PCA training method tries to minimize an error including only a part of the markers of the training set, while the conventional training tries minimizing all markers across the training set. The observed change in shape in general for both methods of training can be described as follows:

- PCA component 1: Left-to-right motion
- PCA component 2: Upward-downward tilting
- PCA component 3: Tongue body curling
- PCA component 4: Tongue widening

For the case of conventional training, it is possible to study the percentages of the different components, which illustrate their explanatory power. It can be observed that the percentages of the first four components add up to 92.23%, able to explain that amount of percentage of the variance of the complete training set. No such percentile has been given for the gappy training, as this has run an optimization algorithm instead of picking a subset of a larger group of components.

6.5 Role of PCA within the system

The role of the PCA model within the system is twofold. The first role is to provide a way of detecting and correcting measurement errors. The measurements of all cameras at a certain moment can be tested for errors by assuming that only the states spanned by the PCA model are possible. Measurement which are not conform that model are considered false. Chapter 9 gives more details about this correction. The second role of the PCA model is that it can be used as a state vector in the Kalman filter. Far less variables have to be tracked compared to when the state vector consists the 3D coordinates of each marker. As a result, analysis of the tracking result is much clearer.

Both methods are in need of a direct relation between the PCA components and the 2D measurements. This is no difficult step, as in the previous chapters, mathematical functions were already given describing the relation between 2D and 3D coordinates, and between 3D coordinate and the PCA vector. Only a substitution of equation 6.1 into equation 5.9 is needed, resulting in the following:

$$\mathbf{a}(\mathbf{u}, P_{ext}) = B(\mathbf{u}, P_{ext})\mathbf{U} = B(\mathbf{u}, P_{ext})\left(\overline{\overline{\mathbf{U}}} + V\mathbf{y}\right) = B(\mathbf{u}, P_{ext})\overline{\overline{\mathbf{U}}} + B(\mathbf{u}, P_{ext})V\mathbf{y} \quad (6.5)$$

In a similar way as in chapter 5 a state vector can be introduced, in order to create a Kalman filter from this system:

$$\mathbf{a}(\mathbf{z}_u, P_{ext}) - B(\mathbf{z}_u, P_{ext})\overline{\overline{\mathbf{U}}} = [B(\mathbf{z}_u, P_{ext})V \ \mathbf{0}]\mathbf{S}_y(i) \quad (6.6)$$

Although it may be clear that Kalman variables like the system matrix F_y are slightly different than the ones used when tracking a 3D state vector, only the measurement noise matrix C_a changes fundamentally:

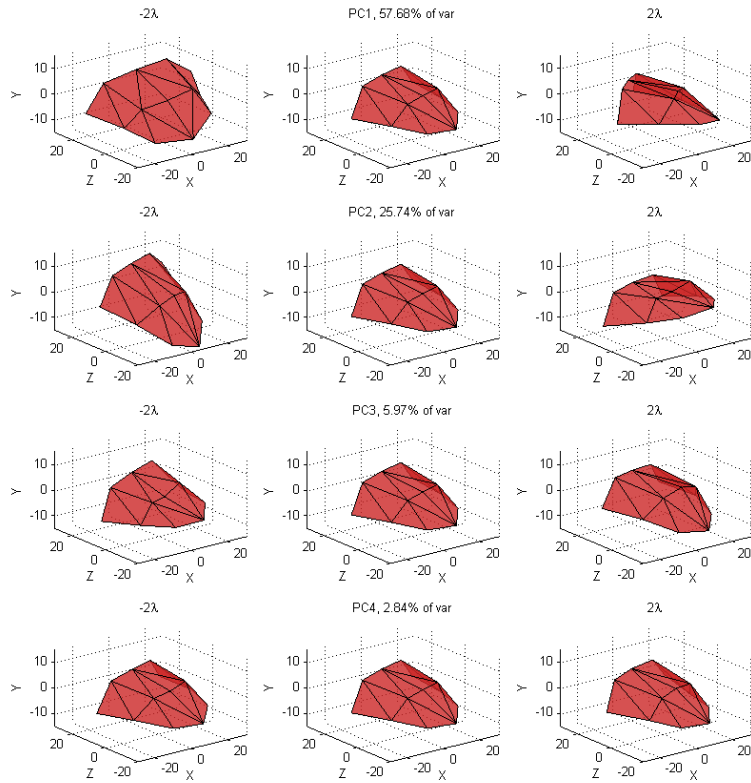


Figure 6.6: Result of varying several PCA weights. Training based on a conventional method.

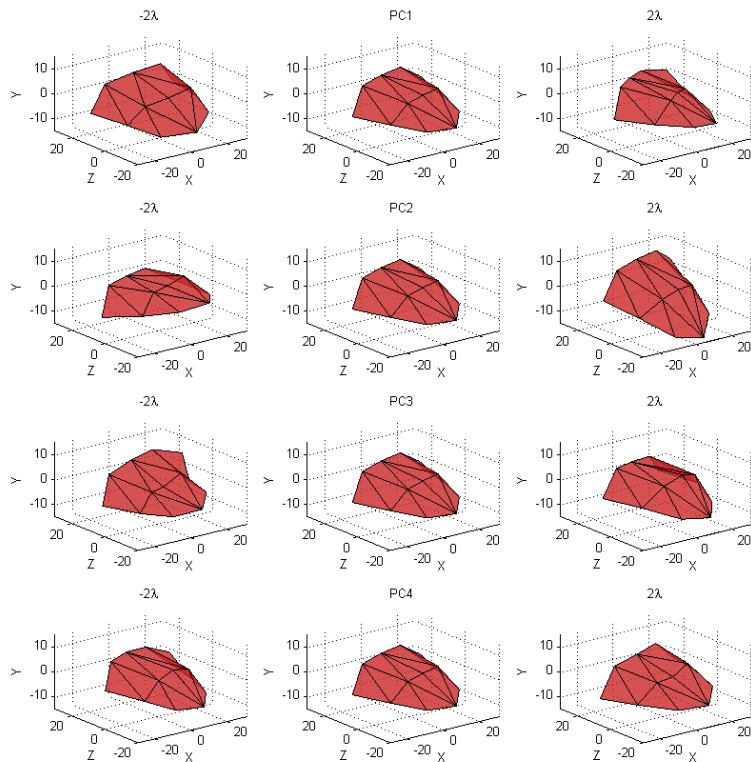


Figure 6.7: Result of varying several PCA weights. Training based on a gappy PCA method.

$$C_a = \text{diag} \left(\sigma_{z_u} \sum_{\text{row}} \begin{bmatrix} P_{ext_{34}} \\ P_{ext_{34}} \end{bmatrix} \right)^2 + \text{diag} \left(\sigma_{z_u} \begin{bmatrix} P_{ext_{34}} \\ P_{ext_{34}} \end{bmatrix} \overline{\overline{\mathbf{U}}} \right)^2 \quad (6.7)$$

In order to retrieve the image coordinates again, a different operation has to be used, as this requires the use of homogeneous coordinates. For this purpose, equation 6.1 will be substituted into equation 5.6:

$$\mathbf{u}_h = P_{ext} \mathbf{U}_h = P_{ext} \begin{bmatrix} \mathbf{U} \\ \mathbf{1} \end{bmatrix} = P_{ext} \begin{bmatrix} \overline{\overline{\mathbf{U}}} + V\mathbf{y} \\ \mathbf{1} \end{bmatrix} = P_{ext} \begin{bmatrix} \overline{\overline{\mathbf{U}}} + [V \ \mathbf{0}] \mathbf{s}_y \\ \mathbf{1} \end{bmatrix} \quad (6.8)$$

Then, the resulting homogeneous set of 2D-coordinates \mathbf{u}_h can easily be rewritten to a nonhomogeneous representation.

The Kalman filter as proposed in this chapter has been used for tracking the tongue markers. The facial markers are also tracked by a Kalman filter, but use a 3D state vector rather than a PCA-based one.

Chapter 7

Marker layout

Earlier research was focused on tracking the shape of the lips. Also here, a marker-based method has been used. This allowed the use of more hazardous substances in order to raise contrast of these with the environment, such as the use of fluorescent markers. The tongue is located in a more critical environment, close to the digestive system and to mucous membranes. This excludes the use of poisonous and other hazardous substances, such as permanent marker ink and fluorescent ink. The moist environment and smooth surface imposes additional constraints on marker fixation. In order to apply them successfully, markers should:

- provide enough contrast with the tongue, either in color difference or in intensity difference;
- be easy to apply and remove within a limited time span;
- remain fixed over several hours;
- not change size, shape or color over that time, and;
- not be poisonous or dangerous in any other kind;

With these points in mind, a suitable marker layout has been developed, consisting of a bandage on which markers are placed beforehand in a fixed structure. This can then be placed inside the mouth using a sticky substance. This provides a relatively easy way to fixate markers while guaranteeing a more reproducible marker layout.

7.1 Materials

Several potential marker materials were studied for use within the mouth. Among these materials were the use of edible ink, pieces of white paper and operation cloth. Most of them suffered from degradation of quality over time. Felt proved to be a relatively good material. First of all, it comes in many colors, such that the user can select them for contrast. Furthermore, it has a more rough structure, so that sticky substances will easily stick to it. It furthermore holds its shape and color over time, even in a moist environment. Finally, it has a slight thickness, which allows the marker to be observable even when seen from the side.

Markers can be fixated to the tongue by using *Fixodent*, a paste used by people to stick dentures to their gums. This paste initially has a peanut butter-like structure, but transforms into a more gum-like structure after contact with saliva.

Early experiments where the markers were directly fixated to the tongue were successful. An image of such a marker structure can be seen in figure 7.1. Most of the markers can hold their position for quite some time at the rough parts of the tongue, including the top surface and the tongue tip. However, at the sides of the tongue, a good fixture is not guaranteed. This is caused by a more smooth tongue surface and the frequent scraping of those areas along the teeth.

During applying markers, it proved to be very hard to place the markers accurately in a fixed layout as determined beforehand. Furthermore, applying them was a very time-consuming job, costing up to half an hour for good fixation. A new idea was to prepare the bandage beforehand, sticking the markers in a regular pattern, which then can be stuck to a tongue in one go. This makes the process of applying markers much faster and reliable. Furthermore, this allows one to prepare the marker layout in a regular grid, guaranteeing reproducible layouts with relative high accuracy. In order to prepare such a bandage, markers are stuck onto a piece of *Fixomull*, a thin and stretchy bandage. Markers can be stuck to it for instance by using super glue, which is harmless after being hardened. Although *Fixomull* has a sticky layer, this cannot be used onto the tongue due to its moist nature. Therefore, also *Fixodent* is used to fixate the bandage. This step requires some training to be accurately and reliably. Applying the bandage restricts the motion of the tongue to a certain



Figure 7.1: The result of fixating pieces of felt directly onto the tongue.

extent. Currently, the advantages it brings are larger than this disadvantage but in future steps new, more flexible materials need to be investigated. Figure 7.2 shows the result of having applied the bandage.



Figure 7.2: The result of fixating a beforehand-prepared bandage onto the tongue

7.2 Marker color

Choice of the marker color is important, as this determines the contrast with the tongue, which has to be high in order for the algorithm to successfully detect the markers. Contrast can be enhanced digitally, but the result will be best if the recorded contrast is high to start with.

Two mechanisms determine the amount of contrast: color difference, the wavelength of the light reflected by the markers, and intensity difference, the amount of light reflected by the marker. In reality, these two are similar: color difference actually is the wavelength-dependent intensity of the reflected light. The light perceived by the camera follows the RGB-scheme: three channels with intensity measured corresponding to the colors red (R), green (G) and blue (B). A combination of these intensities is able to reconstruct any other visible color.

7.2.1 Intensity

When using marker detection based on intensity difference, the measured intensity of the markers should differ as much from the tongue intensity as possible, where intensity I is measured by:

$$I = \sqrt{R^2 + G^2 + B^2} \quad (7.1)$$

In the case of the tongue, markers should either reflect all light, perceived as white, or absorb all light, perceived as black, whatever lies furthest away from the tongue reflected intensity. In the case when using a bandage, mainly white and red (due to its transparency), black would be the best choice.

7.2.2 Color

When using marker detection based on color difference, the marker should absorb the dominant reflected colors by the tongue, and should reflect all other colors. Yamamoto [12] published a paper in which an analysis of the spectrum of the different tongue regions, lips and skins has been performed. Figure 7.3 shows one of the graphs in that work. It can be seen that in all of these regions, most important reflective components lie over 600nm, corresponding to the colors red and orange.

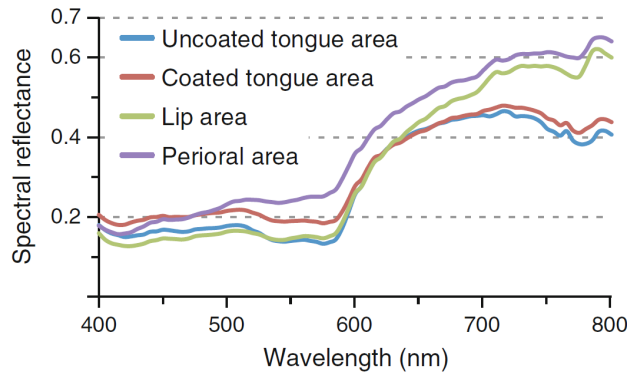


Figure 7.3: Reflectance ratio of the tongue as a function of wavelength.

In figure 7.4, a color wheel can be seen, where the colors red, green and blue are separated an equal angular distance from each other. A set of two colors on any diagonal of the wheel form a complementary set, meaning that when mixed in the proper proportion, a neutral color (white, gray, black) appears. Complementary colors offer very good contrast. For the tongue, this means that colors for markers ranging from blue to cyan offer best contrast. The bandage structure will due to its transparency mainly have a light red color. Blue to cyan will thus yield the best contrast.

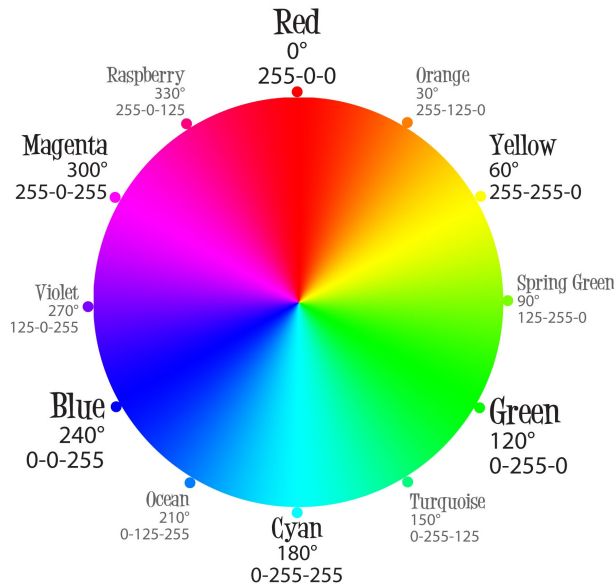


Figure 7.4: Color wheel showing complementary colors.

Color cameras record three colors, stored in three different channels. Exploiting this amount of channels, it is possible to make a distinction in marker type by giving them different colors. This provides the possibility to give the tongue markers different colors in order to make the marker-detection step in the algorithm able to differ between several types of neighboring markers, making the algorithm more robust. Considering that the tongue color will be mainly red, the colors blue and green optimally makes use of the different color channels of the camera. It is possible to use more marker colors, but as more colors are used, the quality of the contrast enhancement step will degrade.

7.3 Final layout

The marker layout is crucial for the usability of the measurements, as its 3D positions define the output of the system. A reproducible marker layout will provide a more constant system output, which is desirable. The marker locations are used to define the deformation of the finite element model of the tongue as a function of the nerve activity, offering the patient-specific model of the tongue. The locations of the markers should be reproducible, and should therefore be preferably fixed anatomical locations on the tongue. However, as stated before, this is hard as the tongue does not offer clear distinguishable locations on the tongue.

Not only reproducibility issues are important, it is also the question which parts of the tongue have to be tracked, discussed before in the chapter about tongue tracking. It is clear that some parts of the tongue, such as the tip and the top surface, are very important to track. The question however is to which extent these parts have to be tracked, and how important it is to track other interesting regions, such as the sides of the tongue. Beforehand, it was not known how much of the tongue has to be tracked in order to achieve a good patient-specific model. In the work of Engwall [6], an overview was given of the tongue regions which are most descriptive for the 3D tongue shape during pronouncing Swedish vowels. From figure 7.5, it can be seen which components contribute most to the shape of the tongue. Although these results give an indication of which tongue locations are important for tracking, the application is entirely different. Where Engwall studied the tongue shape during speech, the system of this thesis will not be used for this purpose. Rather, it is used with opened mouth, assuming random motion of the tongue will be measured. Some parameters used by Engwall, such as *jaw height* will thus lose much of their meaning.

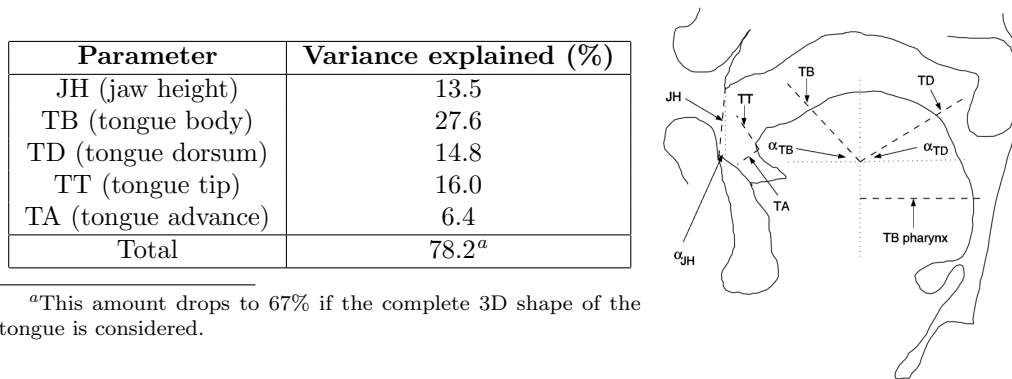


Figure 7.5: Variance explained by the various linear components used by Engwall on the sagittal plane of the tongue during pronouncing Swedish vowels. Left: amount of variance explained by the different linear components. Right: Definition of the linear components projected onto the tongue.

However, from Engwall’s work we can assume that much of the tongue’s variance can be explained by the *tongue tip*, *tongue advance* and *tongue body*, which together explain exactly 50% of the variance in the dataset used by Engwall. The *tongue dorsum* cannot be measured via the mouth opening.

It is expected that, when measuring the part of the tongue that is visible via the mouth opening with greater detail, which means by applying a sufficient amount of markers, the descriptive power will increase. The following reasoning was used to define the to-be-tracked regions: The tip of the tongue is important in Engwall’s work, shows high variability in the case of general tongue movement and can in most cases be observed well. This point and surrounding points will be very important to track. The top surface of the tongue includes the *tongue body*, is in many tongue stances well visible for the cameras and has a big surface. Due to the latter fact, many markers can be applied onto this surface and it is expected that it has high descriptive power. The side surfaces of the tongue are more problematic. Especially when the tongue is retracted far into the mouth, they are prone to occlusion. Furthermore, due to the fact that the texture is very smooth and the surfaces rub along the teeth, markers are hard to apply and to keep fixed. However, in the case of lateral motion, these markers can have much descriptive power, especially when markers on the top of the tongue will become occluded. Therefore, they will be used. The lower surface of the tongue is also more problematic. In most tongue stances, they are not visible. In addition to this, it is hard to create a bandage which is able to cover the lower side of the tongue as it is a 2D shape folded onto a 3D shape, creating overlapping surfaces. The only situation in which these markers are important is when the tongue tip moves to the palate. Due to the hardships they will not be included in the initial marker layout.

Now the surfaces have been defined, it is still the question what the marker density should be. An increased density more accurately tracks the tongue shape. However, when density is too large, it is possible the tracking algorithm can get confused between neighboring markers.

Based on this analysis, a marker layout has been developed. The proposed layout can be seen in 7.6. The layout consists of thirteen markers in either blue or green color. Central in this layout is the line running across the tongue center, with three markers spaced an equal distance from each other. Each of the two markers at the back of that center line has four markers aligned next to it, equidistantly spanning the distance from one side of the tongue to the other. Finally, two other markers are placed next to the tongue center. The color

order is chosen in such a way that most neighboring markers have a different color.

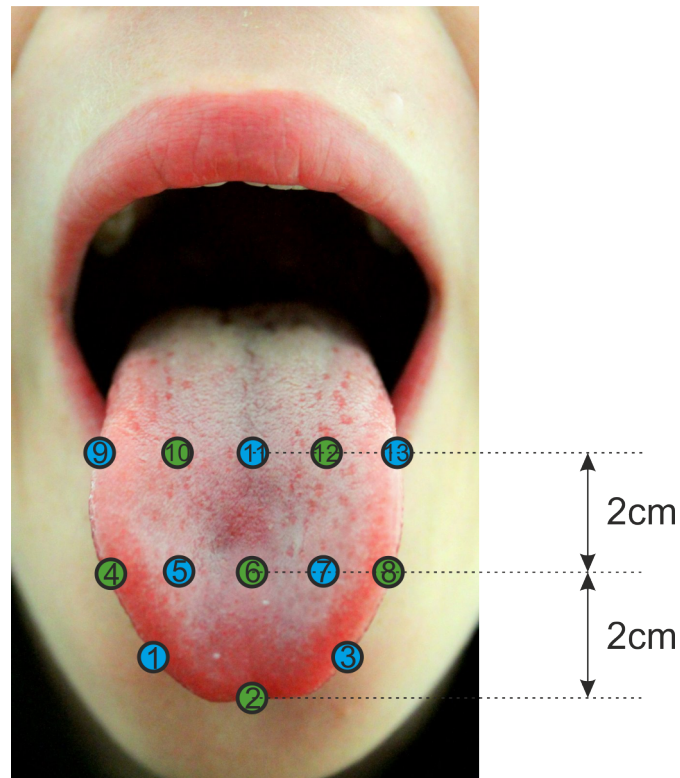


Figure 7.6: Marker placement on the tongue in two colors. Numbers are for marker identification.

One problem arises when creating a patient-specific bandage, as the tongue size and shape is unique for each person. This problem can be overcome by first mapping the shape of the tongue, which has been done by sticking a bandage over it and tracing the central line and the side curves. The resulting shape is then a blueprint for developing the bandage. See figure 7.7 for the result of this method.

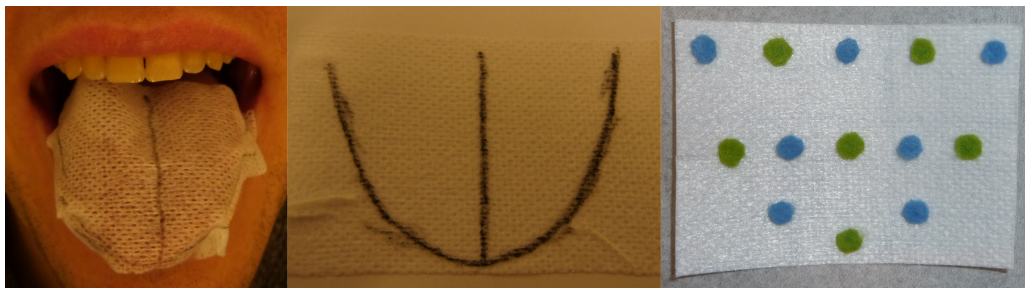


Figure 7.7: Method to create a bandage from the measured tongue curves. Left: measuring tongue curves. Middle: traced tongue curves. Right: Marker layout based on measured tongue curves.

In such custom-made bandages, scaling in the direction of the tongue width is now taken care of. However, the length direction of the tongue should be scaled accordingly too, assuming that the tongue width is proportional to the complete tongue size.

7.4 Facial markers

Facial markers are needed in order to determine the head orientation of the person being imaged. As explained in chapter 6, the markers must be able to provide a vertical head orientation axis and a horizontal head orientation axis. Although three markers are enough to provide this information, a total of four is used in order to make the estimation of these axes independent from each other, allowing some more freedom in placing them. No definite locations have been defined yet, as it has not been investigated which are the most optimal locations. Important is the fact that different persons have different facial proportions, and that the placement of the markers

should result in consistent alignment of the PCA model. Figure 7.8 proposes several anatomically clear locations.

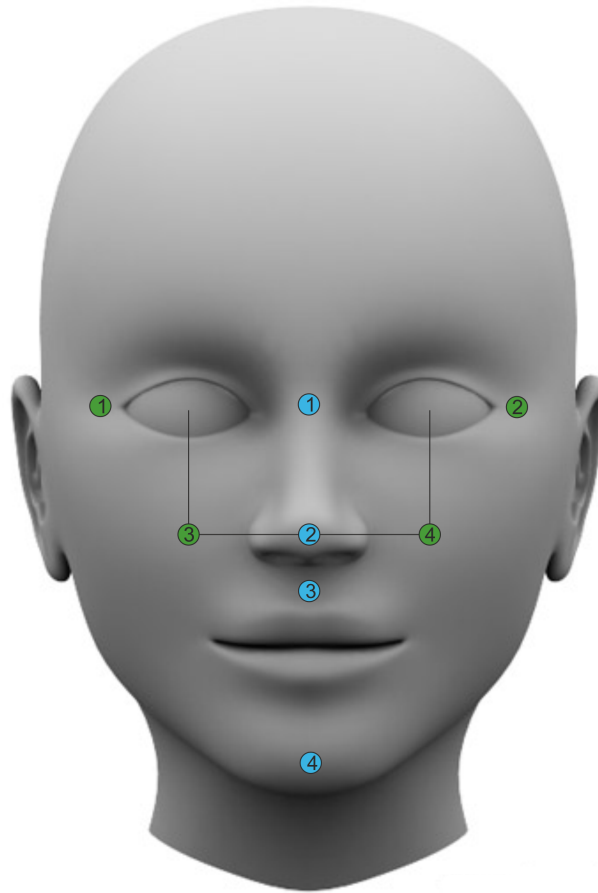


Figure 7.8: Optional locations for placing markers for horizontal head orientation (indicated in green) and vertical head orientation (indicated in blue).

Markers for horizontal alignment do not suffer from inter-personal variation, as the face can be considered symmetrical. As long as they are placed in mirrored positions, horizontal estimation will provide reproducible results. It is advised to place the markers far apart, as this will damp out placement errors. Care must be taken that these markers remain visible to at least two cameras.

Markers for vertical alignment are more problematic, as the face does not have vertical symmetry. Furthermore, inter-personal variation in facial proportions require the markers to be placed on anatomic 'stable' locations, which means that facial locations should be chosen who show as less inter-personal variance as possible. Also here it is advised to choose locations which are spaced sufficiently apart.

Facial markers can be chosen with much more freedom as tongue markers, as the face is a less critical environment. The skin is not located in a moist and slippery environment and allows a higher degree of toxicity in the materials used. Although considering this situation, for the experiments in this research, similar markers were used for facial markers as were used for the tongue.

Chapter 8

Setup

The setup is an important aspect of the system, as ‘good’ data sets provide a basis for successful tracking. A good setup for use within the OR has the following properties:

1. The setup needs to meet the strict safety and hygiene requirements of the OR.
2. The setup needs to make qualitatively good data sets, as this results in a higher probability of successful tracking.
3. The setup should be user-friendly, so that the measurement process is not a time-consuming and error-prone step.

This chapter first describes the older setup, which was used for performing all measurements in this report. Then, some detailed requirements and design of an OR-proof setup will be discussed.

8.1 Initial setup

This setup has been available for some time, and has been used for all experiments in this report. It consists of two Casio Exilim EX-FC100 consumer camera models mounted on a tripod (for some measurements, three cameras have been used). The setup can be seen in figure 8.1. These cameras have the specifications as can be seen in table 8.1. Although the cameras can measure at good resolution and up to high frame rates, they introduce some serious shortcomings.

First of all, the cameras are not synchronized. In the processing step, synchronization must be performed manually based on a visual signal recorded in the measurement step. Furthermore, in the case of a slight misalignment in frame rates, the frames of the different cameras will suffer from temporal drift. Thirdly, the cameras possess a zoom lens, which cannot be set manually. In case of a change in focus after the camera calibration, the calibration result degrades. Additionally, the cameras need to work in a OR-environment, which requires them to be disinfected and put inside a protective sterile bag. This makes them hard and unreliable to operate. Finally, the current setup is rather rigid and does not allow one to flexible place the setup in front of the patient.

Table 8.1: The specification of the Casio Exilim EX-FH100 cameras.

| Specification | Value |
|---------------------|------------------|
| Image resolution | 2292x3456 pixels |
| Resolution @ 30fps | 720x1280 pixels |
| Resolution @ 200fps | 360x480 pixels |
| Sensor size | 1/2.3 |
| Optical zoom | 5x |

8.2 New Setup

The new setup is designed from scratch, and has to meet not only the specifications of the OR-environment, but also technical ones. First, the general and technical requirements will be discussed, after which the design will be presented.



Figure 8.1: The old setup consisting of two consumer model cameras mounted on a tripod. There is a fitting for a light source.

8.2.1 General requirements

The cameras will be placed in a critical environment close to the patient within the OK. The working distance of the cameras will be 30-40cm in front of the patient. This distance needs to be small, as a larger distance will require the cameras to be placed further apart to get the same amount of depth information. Furthermore, small vibrations in the camera system will result in a less shocky video stream when the setup is placed closer to the patient.

The working area of the cameras will be separated from the operation area; a sterile blanket will be placed between the operation area (the neck) and the frontal part of the face (see figure 8.2). This means that the cameras operate in the non-sterile area. Still, the setup needs to be disinfected and placed within a sterile bag (including the cables). The reason for this is that, in order to decrease the chance of infection, an over-pressure is created around the patient by blowing cleaned air from the top. The camera system may disturb and infect the airflow across the operation table, and therefore needs to meet these requirements. It is further given that the platter holding the cameras and the arm holding the platter should influence the air flow as little as possible, again to prevent infected air to reach the patient.

The subsystem controlling the cameras and storing their data, in the form of a computer system, also should meet the hygiene-requirements of the OR.



Figure 8.2: The cameras can be placed in a non-sterile environment when the to-be-imaged face (left) is screened off from the sterile neck operation area (right) with a sterile screen.

Another requirement is that the positioning of the cameras should be easy. This requires not only a user-friendly positioning arm, but also should have a clear interface. During placement and during data acquisition, the camera frames should be observable on a user-friendly interface. Such an interface should also allow some freedom in choosing important camera settings, such as gain, frame rate and resolution, when the situation requires this.

8.2.2 Technical requirements

The camera system should not only be safely usable inside an OR-environment, but should also have a certain set of specifications in order to optimize the tracking algorithm. A good processing algorithm may be able to overcome limitations of the data-recording step, but may require a lot of time and effort. This chapter will describe some of these requirements.

Resolution

This is one of the most typical specification of a camera, and is a measure for how many detail is visible in the recorded data set (assuming a sufficiently high signal-to-noise level). However, it is not known what minimum resolution level should be used for the system. There are ways to evaluate this; for instance by filtering and sub sampling a data set down to several levels, and then evaluating the deterioration of the tracking performance. However, that would be for that specific marker-layout. In future stages of the project, marker shape might become smaller, or lighting conditions may change, such that a different resolution is desired. Therefore, to allow some freedom in future steps, it has been chosen to pick a camera *with at least the same resolution as the current setup*.

Frame Rate

When looking from a purely signal-processing perspective, the Nyquist sampling theorem teaches us that in order to be able to reconstruct a signal, the sampling rate should be at least twice as high as the highest frequency components in the measured signal. Therefore, a test has been performed in which a camera was placed in front of a subject. The subject moved its tongue up- and downward as fast as possible, while a camera was recording this at a frame rate of 200Hz. In a subsequent step, in each frame, the tongue tip was selected manually, something made easier by beforehand having applied a marker onto the tongue tip. Then, a Fourier analysis has been performed onto the vertical coordinates of this signal. Figure 8.3 shows the result of this experiment. As can be seen, the most important frequency components remain below 10Hz. A frame rate of 20Hz then should be sufficient to reconstruct the tongue motion.

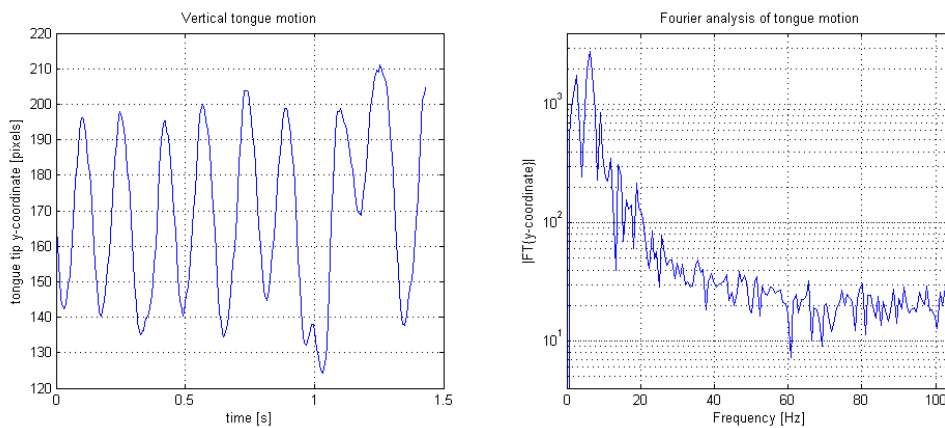


Figure 8.3: Fourier analysis results of a fast-moving tongue. Left: vertical displacement of the tongue tip. Right: Fourier analysis of this signal (absolute value).

However, the tracking algorithm in its current form uses a template matching scheme for marker location detection. It is made more robust by cropping the search region to a limited region around the predicted location. Although a signal may be fully reconstructable if the measurement system meets the Nyquist criterion, in this case the measurement performance is severely infected by sudden changes in marker position and velocity, causing the prediction of the marker locations to be very bad. Therefore, it is chosen to hold on to the rule of thumb from control engineers to oversample the signal by a factor of five, bringing the desired frame rate to 100Hz.

Amount of cameras

At least two cameras are needed in order to be able to reconstruct 3D points. However, due to the curving of the tongue, often there are situations in which a marker is visible by one camera only, resulting in a less qualitative reconstruction. It is unknown how much influence this phenomenon has. To ensure good data acquisition, a set of three cameras is used, improving the chance of capturing a marker with more than one camera.

Lighting

The relative high resolution and frame rate of the cameras require additional lighting for a good signal to noise ratio in the resulting frames. The fact that the tongue is located in an enclosed environment poses some constraints on the lighting source. The source should illuminate the tongue through the opening, and thus should be placed next to or behind the cameras. Preferably this should be a diffuse source, to soften reflections. Finally, it should also provide enough light onto the face, for good contrast between the facial markers and the skin. And of course, the source should be OR-approved.

Electrical safety

Finally, electrical safety is an important aspect, not only protecting the equipment and indirectly the patient against power surges, but also not to infect the power group of the OR. This means that all electrical equipment should meet the IEC 60601 standard. Furthermore, the cameras are in need of a galvanic separation between themselves and the PC.

8.2.3 Design

Due to various sources of delay, the setup has not been built yet. However, a design is already developed. The general overview can be seen in figure 8.4. As can be seen, the cameras will be mounted onto a small form-factor brace, having limited influence on the air stream. A movable arm connects the brace to its mounting point on the IV pole. This arm has many degrees of freedom, and can thus be positioned optimal in front of the subjects' head. A rigid IV pole offers a mounting point for this arm.

The OR-lamp provides an easy and effective illumination solution: it is OR-proof, offers a dense light concentration, is easy to place and due to its wide nature can be placed behind the cameras.

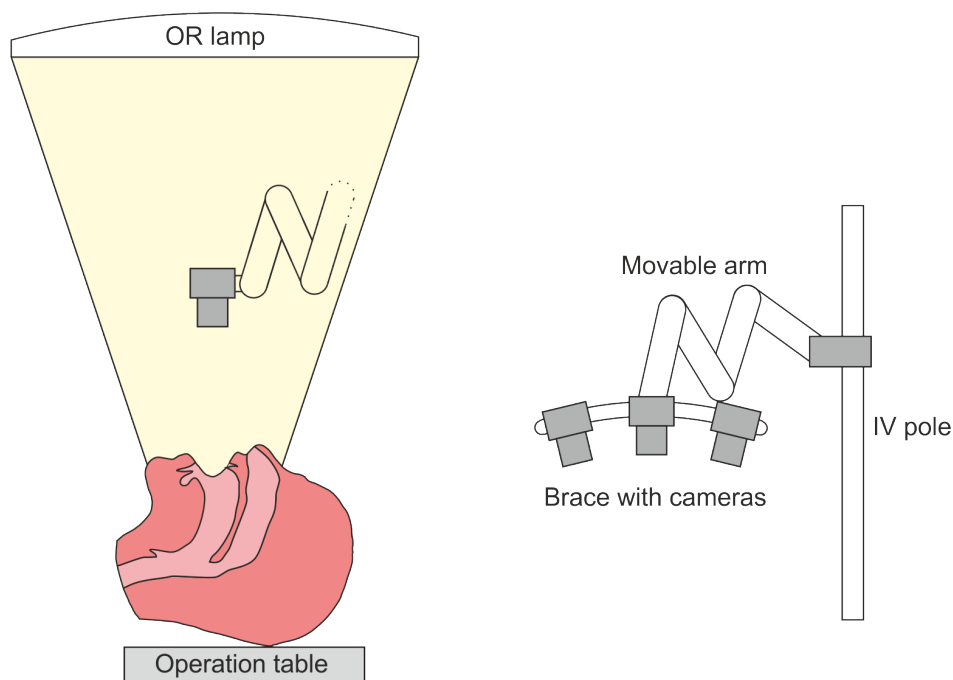


Figure 8.4: Three cameras are mounted to a (thin) brace, and connected to a stable IV pole via a movable arm (right). The OR lamp provides enough light when placed behind the cameras (left).

8.2.4 Hardware

The chosen cameras are the avA1000-100gc (see figure 8.5). Its specifications are given in table 8.2.

A fixed focal 8.5mm lens is mounted on these cameras in order to get a view of 25cm in height (roughly the head size) when placing them 30cm from the subject. The cameras need to be connected to the PC via gigabit ethernet connections. As each camera produces a maximum data rate of around 100 MByte/s, three dedicated ethernet connections must be available on the PC, as well as two solid state drives in order to store the data. A synchronization box will be built, connected via USB to the PC, driving the trigger input of each camera simultaneously in order to guarantee a good online synchronization and preventing temporal drift. The



Figure 8.5: The cameras selected for the camera system.

Table 8.2: The specification of the Basler avA1000-100gc rgb cameras.

| Specification | Value |
|------------------|---|
| Resolution | 1024x1024 pixels |
| Frame rate | 101 fps |
| Sensor | KAI-1050, 1/2 inch |
| Connections | gigE, 12-pin connector (power, synch, etc.) |
| Size (L x B x H) | 40.7 x 62 x 62 mm |
| Weight | 300g |

PC itself is medical model, with a custom power source unit conform the IEC 60601 specification and with anti-bacterial air filters. A diagram of the connections between the subsystems can be seen in figure 8.6.

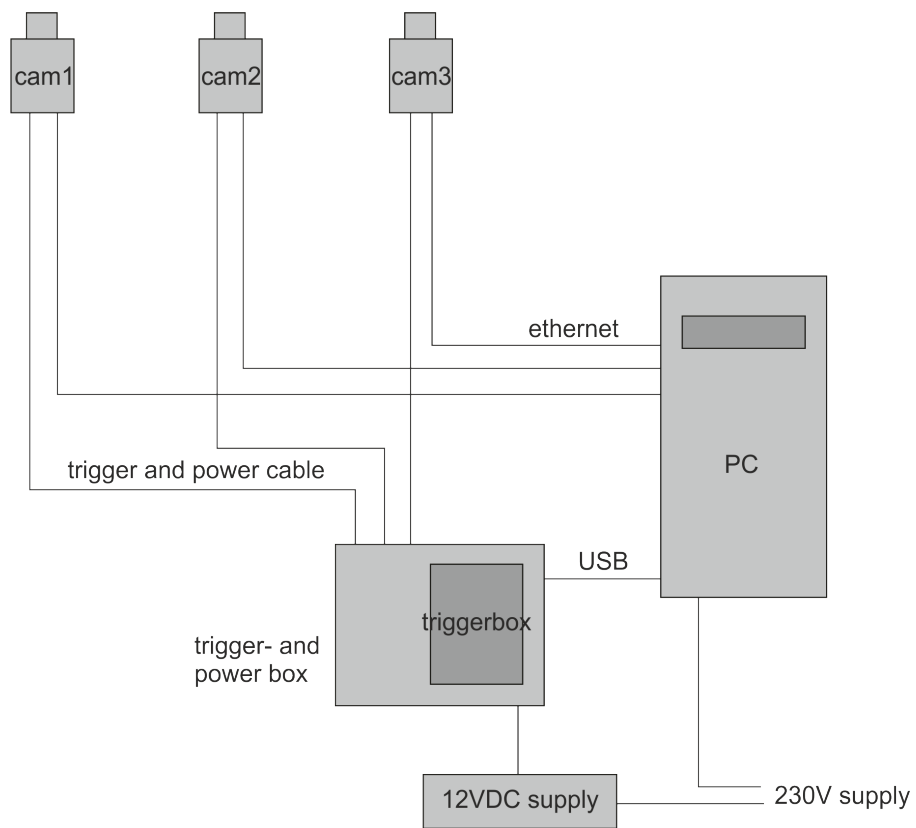


Figure 8.6: An overview of the connections between the subsystems.

Chapter 9

Tracking algorithm

The tracking algorithm is the core program of this thesis. It involves the estimation of the three-dimensional shape of the tongue in subsequent images, based on the detection of markers applied onto the tongue. Figure 9.1 presents a flow chart for a frame-to-frame approach.

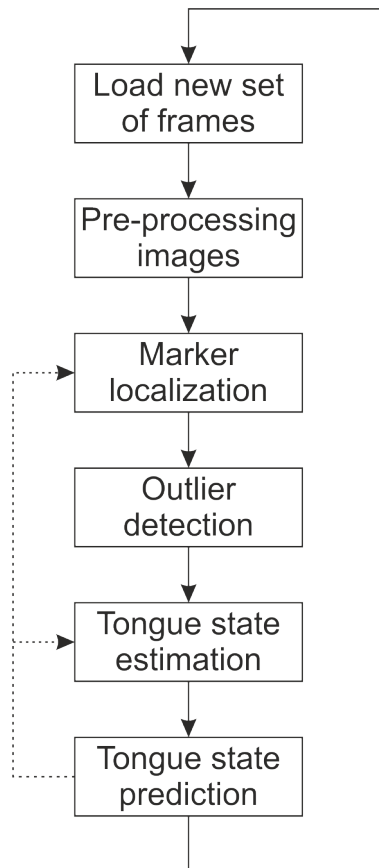


Figure 9.1: Flow chart of the tracking algorithm

First, a set of synchronized frames (one for each camera) is loaded. They are pre-processed for noise reduction and for increasing contrast of the markers with the tongue. After this, a marker detection algorithm is used for measuring marker image coordinates. A tongue state prediction from the previous set of frames provides a prediction of the marker locations in the current set of frames, which increases the search process and makes it more robust. An outlier detection algorithm then has to correct for inaccurately detected markers, due to (partly) occlusion or other measurement errors. This has to be performed before further processing of the measurements, due to the fact that accumulation of errors is not desired.

After outlier detection, the state of the tongue can be estimated. This can be either expressed in the state of the 3D-representation of the markers, or a PCA representation. The latter has been chosen for this project. Information of previous frame sets can be used for this estimation by using a Kalman filter. After state estimation, a prediction is performed, offering a basis for better marker detection and tongue state estimation.

9.1 Overview of the tracking process

So far, three different 'spaces' have been defined, as illustrated in figure 9.2. The first one is the two-dimensional space, including the observations of the cameras. The state of N markers at time instance i is denoted by $\mathbf{u}(i)$. As observations are made by three cameras, and each marker has two variables, it is valid that $\mathbf{u}(i) \in \mathbb{R}^{6N}$. Transition to the three-dimensional domain is possible, where $\mathbf{U}(i)$ represents the coordinates of all markers at time instance i . As each marker has three variables, it is valid that $\mathbf{U}(i) \in \mathbb{R}^{3N}$. The linearized camera calibration result allows a free transform between the 2D and 3D space. Finally, the PCA space allows much freedom in choosing the number of variables of vector $\mathbf{y}(i) \in \mathbb{R}^M$. However, reduction of dimensionality is desired so a general choice is that $M < 3N$. There is a direct transformation possible between 3D and PCA space. There are relations for transition from and to each of these spaces. These relations were explained earlier on in the report, in chapters 5 and 6.

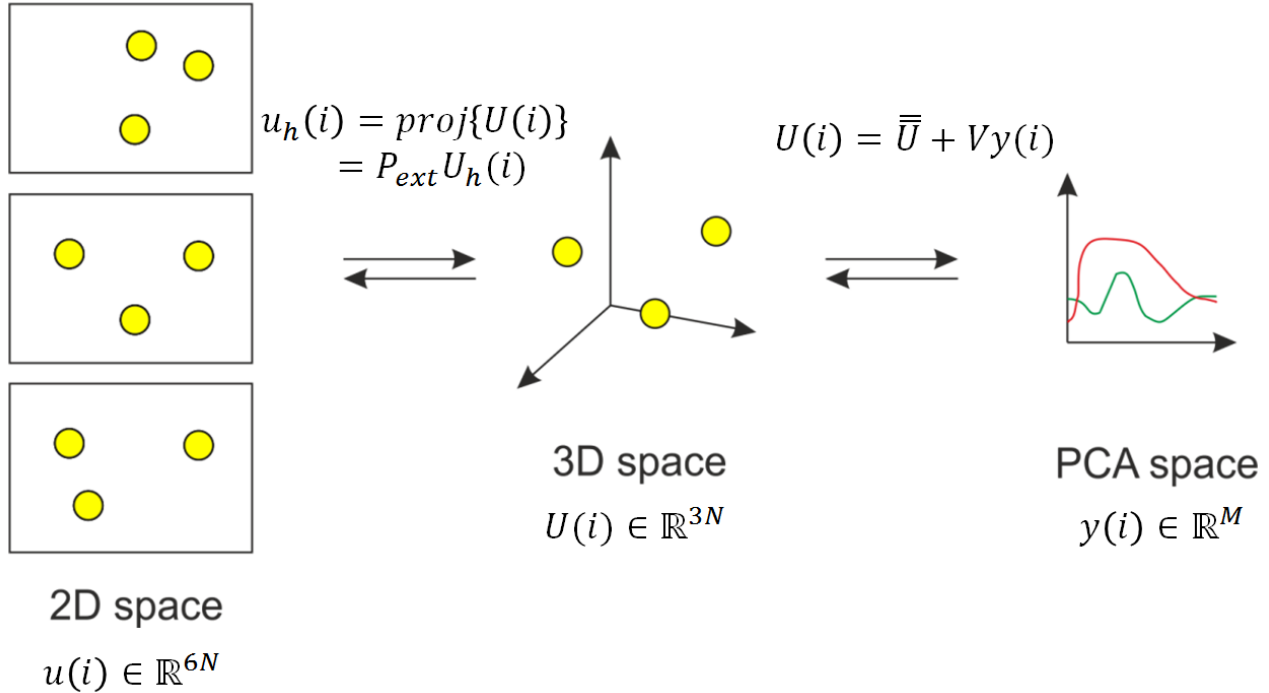


Figure 9.2: The three domains with their transitions.

Figure 9.3 gives an overview of the estimation process in more detail. Most of the mathematical details have been discussed in the previous chapter. The ones that have not been discussed, will be explained in following sections. The block diagram of figure 9.3 is further explained by the pseudo-code in figure 9.4.

The next chapters will describe key processes with more detail.

9.2 Pre-processing images

The goal of pre-processing the images is to reduce noise and to increase the contrast between the markers and the tongue. Noise reduction is especially useful for high-frequency noise, as phenomena like quantization noise are especially disturbing. A simple Gaussian lowpass filter provides enough filtering, if needed at all.

For the contrast transform, we assume the following situation: an image recorded by an RGB-camera is a projection of the 3D environment onto the image plane. The objects in the environment consist of materials, where each material has a color. Knowing this, the image can also be represented by multiple 'material images', where each of those images only shows the material density of one specific material. The pixel intensity of any pixel can then be represented by a linear combination of those material density pixels and their color composition:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = M \begin{bmatrix} mat_1 \\ mat_2 \\ \vdots \\ mat_n \end{bmatrix} \quad (9.1)$$

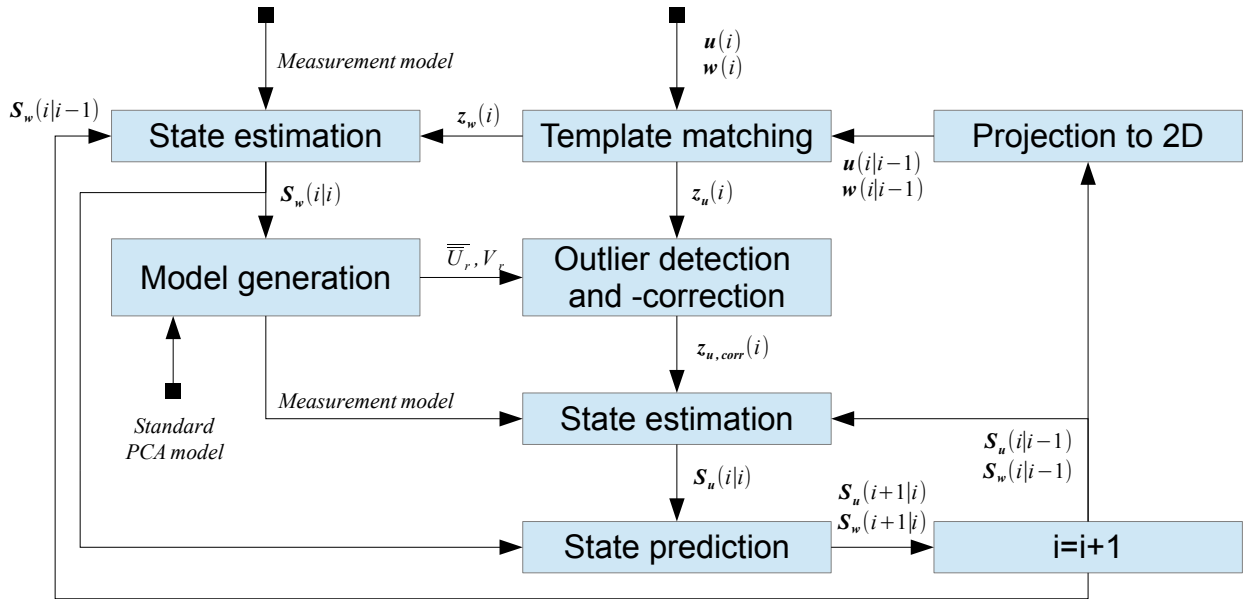


Figure 9.3: A mathematical block diagram of the tongue state estimation.

1. At time i , $\mathbf{w}(i)$ and $\mathbf{u}(i)$ denote the true image coordinates of respectively the facial markers and the tongue markers.
2. Perform template matching to get measurements of these image coordinates: $\mathbf{z}_w(i)$ and $\mathbf{z}_u(i)$ respectively. Perform a search around the predicted marker locations.
3. Using a Kalman filter, estimate the 3D state vector $\mathbf{S}_w(i|i)$ of the facial markers. This requires the input of state prediction $\mathbf{S}_w(i|i-1)$ and a suitable measurement model (see chapter 5).
4. The 3D facial marker state must be used in order to rotate the PCA model accordingly, resulting in a rotated PCA model given by $\overline{\overline{\mathbf{U}}}_r$ and V_r . This rotated PCA model is also needed to generate the measurement model for the tongue state estimation.
5. Now using the rotated PCA model, perform an outlier detection and -correction scheme in order to arrive at a corrected measurement $\mathbf{z}_{u,corr}(i)$.
6. Perform a Kalman estimation for updating the PCA state vector $\mathbf{S}_u(i|i)$. This requires the input of state prediction $\mathbf{S}_u(i|i-1)$ and a suitable measurement model (see chapter 6).
7. Perform a Kalman update to create predictions of the next states $\mathbf{S}_u(i+1|i)$ and $\mathbf{S}_w(i+1|i)$.
8. Go to the next iteration and load a new image set.
9. Create estimated markers locations in the images by projecting $\mathbf{S}_u(i|i-1)$ and $\mathbf{S}_w(i|i-1)$ to 2D. This gives expected image locations $\mathbf{u}(i|i-1)$ and $\mathbf{w}(i|i-1)$ respectively.
10. Return to step 2, and repeat the tracking process until all frames have been processed.

Figure 9.4: The tracking process of figure 9.3 explained in pseudo-code.

Where R , G and B denote the colors of the observed image, mat_x is the material density images of material x , n represents the number of materials observed in the image and M is the $3 \times n$ color transform matrix. When taking the assumption that the image only consists of 3 materials, M becomes square and invertible, such that the material density images can be obtained, given that the color composition of those materials are known (which they are, as they can be selected from the observed image itself):

$$\begin{bmatrix} mat_1 \\ mat_2 \\ mat_2 \end{bmatrix} = M^{-1} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (9.2)$$

When three materials are defined, such as the tongue and two different kinds of markers, a clear distinction can be made between those three groups. Figure 9.5 gives an example of this.

Note that it is possible to solve toward more material images. M will then become underdetermined, but the system can be solved for example by using a pseudo-inverse of M . However, as the system is underdetermined, results will not be as optimal as in the case when three material images are subtracted and thus is not recommended.

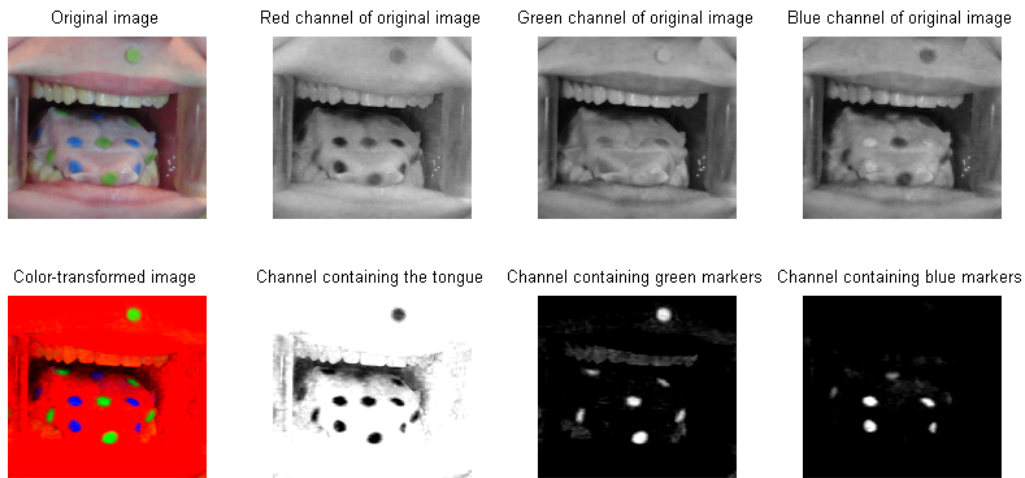


Figure 9.5: Original image and its color channels transformed to material images.

9.3 Marker detection

Template matching is used for detecting markers. In this method, the to-be detected object is represented by a template image: an archetype of the object. A criterion is then defined which defines how well the different parts of the image resembles the template. There are different criteria, but in this case the *sum of squared differences* (SSD) is used. For two vectors \mathbf{x} and \mathbf{y} , this can be notated by:

$$SSD = \|\mathbf{x} - \mathbf{y}\| = \sum_{i=1}^N (x_i - y_i)^2 \quad (9.3)$$

In the case of template matching, \mathbf{x} is composed from N pixels of the template image, and \mathbf{y} is composed of an equal amount of pixels of the image in a window around the image location at which the SSD is calculated. For each marker, a search is performed in a region around its predicted location in order to speed up the system and to make it more robust.

Note that this algorithm *always* finds a match (corresponding to lowest SSD), even when there is no marker within the searching frame. This has been chosen because the environment of the markers change significantly over time (for instance, due to a changing marker orientation), and a suitable threshold level for marker detection proves to be unreliable. Errors in the marker detection process can then be corrected for by the outlier detection step.

9.3.1 Defining the templates

Performance of the measurement (marker detection) is significantly dependent on the choice of the template. This choice allows much freedom, as template shape, size, and color can be arbitrarily chosen.

A straightforward choice could be to select a sub-image from a reference frame, and using this as the template. This offers advantages like a perfect match of color and shape (but only for that specific frame). However, a template based on such a method can also incorporate noise and reflections, which would introduce a systematic error in the template. A second method is to create templates artificially from a model. For circular-shaped markers, for instance, the template can be chosen to be an ideal disk with an uniform colored foreground and uniform colored background.

When choosing templates, it has to be taken into account that due to a change in conditions (such as orientation and visibility of the markers) the performance of the template matching can degrade over time. A solution for this problem can be searched in a dynamic adaptable template. An example of such a method for has been proposed by Nguyen et al. [21] for the case of image-selected templates. This method uses a Kalman filter for each template, each frame adapting the template based on the previous template and information in the new frame. However, this may result in template drift, meaning that the marker itself eventually disappears from the template. If such an approach is taken, a force must be incorporated, pushing the Kalman update towards the center of the marker.

Artificially created templates (generated from a model) do not have this disadvantage, and can be dynamically adapted based on state information of the tongue. For instance, the size of the template can be adapted based on the distance of the cameras with respect to the patient, and the orientation of the marker shape can be altered based on the PCA state of the tongue. This offers a great advantage over image-selected templates.

Due to their flexibility, the system has been designed around artificially created templates in the shape of a disk. Some variables in the generation process include size, amount of background and orientation of the templates. Because state-dependent template adaption requires a fixed, good PCA model, which is not available yet, this is currently not used in the tracking process (but it can be used in the future). An example of templates which can be generated can be seen in figure 9.6. Note that the background is set to black, as the result of successful contrast-enhancement (discussed in the previous section) leaves a dark background in that specific color channel.

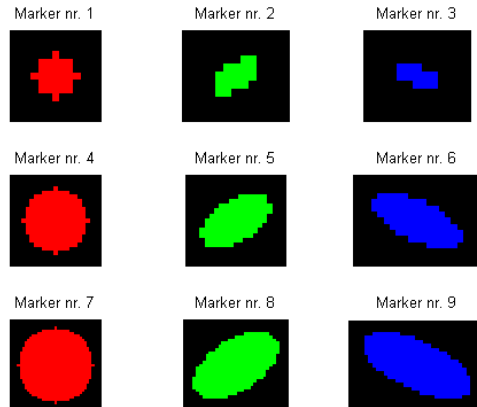


Figure 9.6: Example of custom generated templates, varying in size, color and orientation, all based on a 3D disk model.

9.4 PCA model rotation

In chapter 6 it was explained how the creation process of the PCA model takes place. This involves orientation- and translation normalization. The result is a PCA model with its center of gravity in the world coordinate origin, oriented in such a way that the facial markers are oriented in the xy -plane. Although the translation vectors in the PCA model solve the translation problem by allowing a free 3D translation, the rotation problem needs to be solved deterministic. For this purpose, the facial markers are used in a similar method as described in

chapter 6 for orientation normalization. The rotated PCA model is needed for outlier detection and -correction, as well as for constructing a measurement model for the tongue state Kalman filter. A block diagram of this process is illustrated in figure 9.7. Details of the process are described by the pseudo-code in figure 9.8.

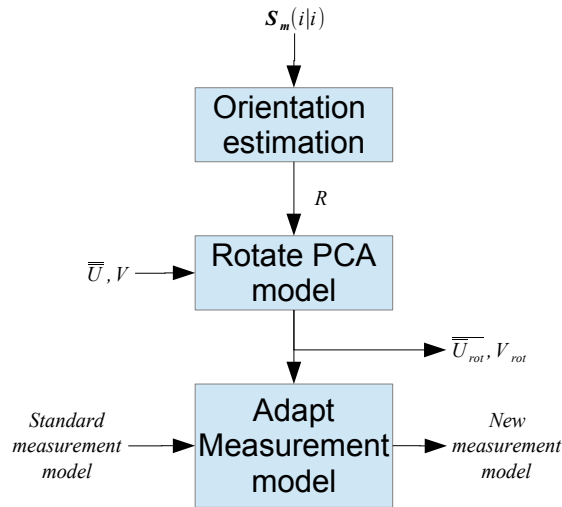


Figure 9.7: Scheme for rotating the PCA model and adapting the measurement model.

9.5 Outlier detection

The algorithm must be robust to missing markers and outliers resulting from occlusion and measurement errors. If only a part of the markers is detected well (the set of inliers), the algorithm should be able to correct the outliers. A PCA model can be used for this purpose, where estimation of the principal components can provide a way of error correction. However, before error correction can be performed, error detection must be applied. Figure 9.9 illustrates this concept for a system fitting a straight line through measured points, by estimating the line parameters. This situation is somehow similar to the situation in which the tongue shape must be fit inside a set of measured markers. When false measurements are performed, a robust estimator is able to recover the shape of the lips using only the well-detected markers, while a regular LSE estimator will not achieve a good result.

In the work of Nguyen [21], a method was introduced for robust recognition of objects in images using eigenimages. Such a method was also used by v.d. Heijden [24]. Although used for images, the technique can very well be used for error detection and -correction for this application.

1. From the state vector $\mathbf{S}_w(i|i)$, subtract the 3D facial marker coordinates $W(i)$.
2. Calculate the rotation of the head with respect to the world coordinate system:
 - From the facial markers, estimate the horizontal face unit vector \mathbf{e}_x .
 - From the facial markers, estimate the vertical face unit vector \mathbf{e}_y .
 - Estimate the face-orthogonal unit vector $\frac{\mathbf{e}_x \times \mathbf{e}_y}{\|\mathbf{e}_x \times \mathbf{e}_y\|}$. (\mathbf{e}_x and \mathbf{e}_y might not be orthogonal)
 - Calculate the rotation R with respect to the xy-plane of the world coordinate system using for instance Horns' method [8].
3. Rotate the PCA model: $\bar{\mathbf{U}}_r = R\bar{\mathbf{U}}$, and $V_r = RV$.
4. Adapt the measurement model (see equation 6.6): $\mathbf{a}(\mathbf{z}_u, P_{ext}) - B(\mathbf{z}_u, P_{ext})\bar{\mathbf{U}}_r = [B(\mathbf{z}_u, P_{ext})V_r \mathbf{0}]\mathbf{S}_y(i)$

Figure 9.8: The process of figure 9.7 explained in pseudo-code

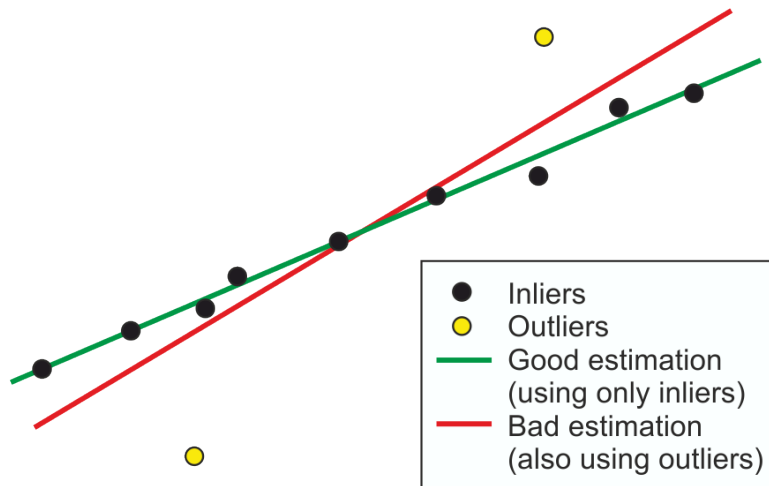


Figure 9.9: Example illustrating that only correctly detected markers should be used for state estimation (in this case, fitting a line between measured points).

The method is based on a PCA model with a small amount of components, as a result of reduction of dimensionality. The PCA vector \mathbf{y} can be estimated from a much larger set of 2D marker image coordinates \mathbf{u} . The system is significant over determined, with generally less than 10 PCA component variables, compared to 78 image coordinate variables in a realistic 3-camera setup. This allows one to use less measured image coordinates in order to estimate the PCA components.

The outlier detection algorithm offers a solution in the form of hypothesis generation and -testing in a similar way as RANSAC. Figure 9.10 gives an overview of the outlier detection process. The pseudo-code given in figure 9.11 further explains this process.

Important is the choice for parameters T_{out} and α . The first is a measure for sensitivity of outlier detection. Choosing this too small results in very few inliers, while choosing too large does not allow outliers to be detected. α Determines the relative importance of the number of inliers with respect to the inlier reconstruction error. Choosing this small will allow a reconstructed shape to have a lot of outliers. Choosing this large will search for a solution with as few outliers as possible.

9.6 Tongue state estimation and -prediction

A Kalman filter is used for tracking the state of the tongue. For this, a PCA state vector is used, not only including the PCA components themselves but also their rate of change. A second Kalman filter is used for tracking the 3D-location of the facial markers. In this case however the the 3D marker positions serve as a basis of the state vector, also incorporating the rate of change. Most of the mathematical details for both these filters have been discussed in chapters 5 and 6 already.

One of the things not discussed yet is how to choose the measurement- and system covariance matrix. That of the measurement noise can be directly determined when looking at the results that template matching gives us. By observing the errors and describing its standard deviation compared to human-selected points, this gives a usable number. One might think of a similar method to subtract a suitable number for the system noise.

However, the choice for a suitable number should not be chosen solely on observed uncertainties. An important aspect of the system is that it relies on marker localization based on a window around a predicted position. This means that the tracking 'speed' has a profound impact on the performance. To illustrate this phenomenon, imagine a system with relative low system noise. In that case the Kalman filter will give a high weighting factor to the predicted state and a low weighting factor to the measurement during the estimation step. This will mean that in the case of abrupt motion of the tongue, the estimation might be too slow to keep up with the true state of the tongue. As a consequence, the predicted marker coordinates may drift from the true coordinates, falling outside the search window of the template matching step. In contrast, when the system noise will be relative high, there will be a high degree of noise in the resulting state. The chance of markers moving out of the search window, however, will be much smaller.

The approach offering a solution to this problem is not by choosing only one method, but by applying them both. For this purpose, the Kalman filter is run twice. During the first run, the system noise is made very large,

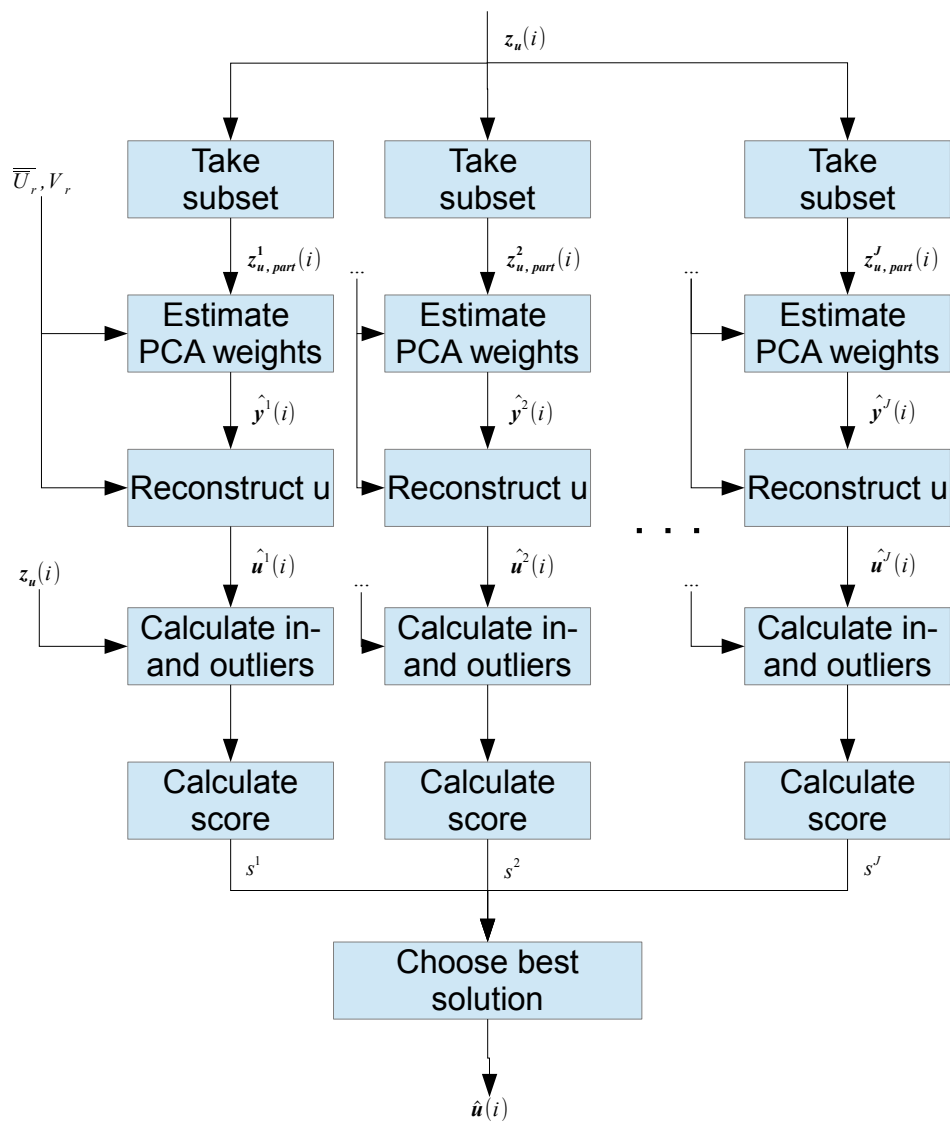


Figure 9.10: Block diagram of the outlier detection process.

1. From measurement $\mathbf{z}_u(i)$, take J subsets. Select an equal amount of markers for each camera. Denote the subset by $\mathbf{z}_{u,part}^j$, where $j \in J$.
2. Estimate the PCA components $\hat{\mathbf{y}}^j(i)$ of these subsets using the *rotated* PCA model.
3. Using the same rotated PCA model, reconstruct the 2D image coordinates of *all* markers, denoted by $\hat{\mathbf{u}}^j$.
4. Determine the in- and outliers:
 - For each marker in all images, determine the Euclidian error between the measured and reconstructed position: $e_{eucl}^n = \sqrt{(z_x(n) - \tilde{z}_x(n))^2 + (z_y(n) - \tilde{z}_y(n))^2}$, with $n \in 3N$, where $z_x(n)$ and $z_y(n)$ are the measured image coordinates and $\tilde{z}_x(n)$ and $\tilde{z}_y(n)$ the reconstructed image coordinates of the n 'th marker.
 - When the Euclidian error is greater than a threshold, denote this marker as an outlier: $N_{out} = \{n | e_{eucl}^n > T_{out}\}$, where $n \in 3N$, and T_{out} the threshold. Denote the set of inliers by N_{in} .
5. Calculate the score of each subset: $s^j = \alpha \left[N_{out}^j \right]_{nrElem} + \sum_{n \in N_{in}} e_{eucl}^n$, N_{in} being the set of inliers, α a weighting factor and $\left[N_{out}^j \right]_{nrElem}$ the amount of elements within the set of outliers.
6. Finally, the solution with the smallest score is chosen as the best solution.

Figure 9.11: The process of figure 9.10 explained in pseudo-code.

offering successful marker detection but relative much noise. When each measurement has been recorded, the algorithm can be run again, using a lower system noise, giving a smoothed response. Note that measurements in this research have been performed on movies with a frame rate of 30fps, while the new setup will have 101fps. This opens up the possibility of filtering during the first run. Still, this does not offer a significant advantage over the double-run Kalman filter, as the estimation process takes up very few time.

In order for the Kalman filter to use *all* measurements instead of only the *past* measurements, a technique known as the *Rauch-Tung-Striebel* smoother can be used. This method is based by first running a general Kalman filter, only taking into account the past and current measurements. Then, this information is combined with a backwards-run Kalman filter. Theory for this method can be found in [25]. A demonstration of filtering the resulting measured signal can be seen when comparing figures 9.12 and 9.13.

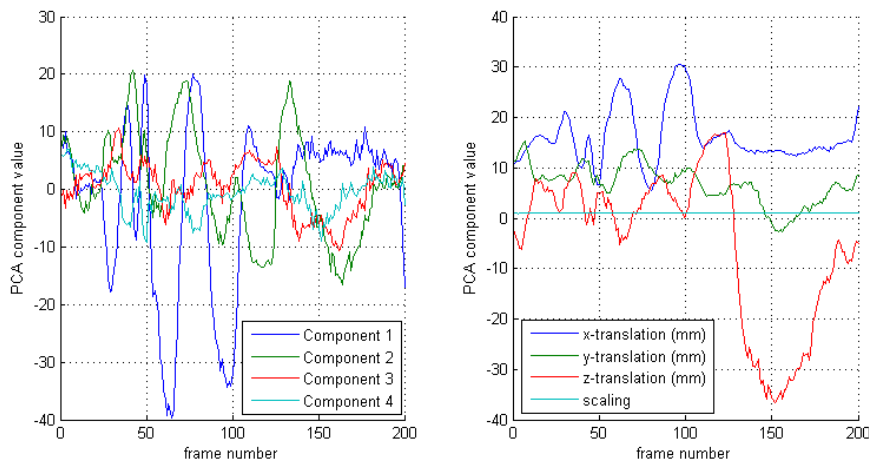


Figure 9.12: Result of tracking with high system noise. Free PCA components are shown left, general components right.

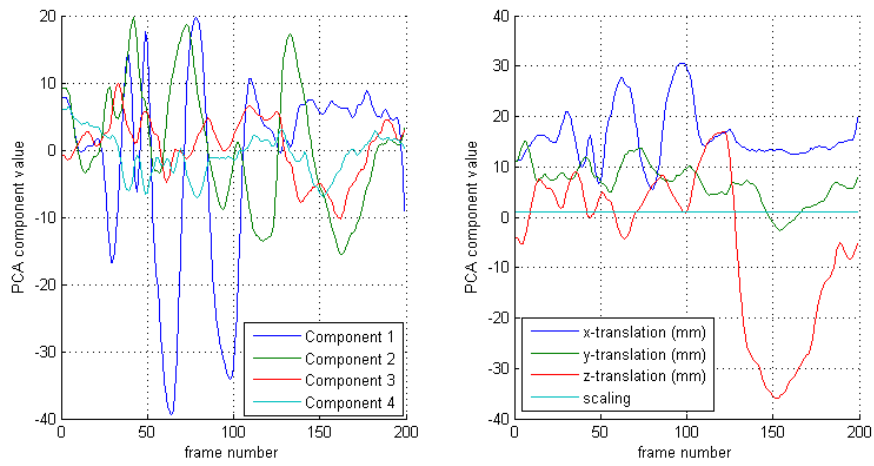


Figure 9.13: Result after reducing system noise and adding the Rauch-Tung-Striebel smoother. Free PCA components are shown left, general components right.

Chapter 10

Experiments

A set of experiments has been performed in order to evaluate the performance of the system. Such experiments can be subdivided in two groups. The first group describes the qualitative performance of the system. These results can prove that the method works well or not, and under which conditions. The second group describes the quantitative performance of the system. From this, the numerical accuracy can be described under various conditions. For instance, a comparison can be made between different training methods of the PCA model.

This chapter describes the type of experiments that were conducted. The results of those experiments can be found in the next chapter. All measurements were performed using the test setup. This is caused by the fact that during the development of the system, we were delayed by problems concerning hygiene and other OR requirements, as well as budget constraints.

10.1 Qualitative experiments

Qualitative experiments can be used to investigate whether the tracking method in a realistic setup is successful or not, and in which situations problems occur. It may be clear that a good PCA model is critical for good operation of the program, and should be usable on any measurement of any person. The following questions will have to be answered:

- Does the tracking algorithm in general work?
- Does the tracking algorithm deliver reproducible results?
- Does the algorithm work for multiple persons?

The first question focuses on the fundamental question if the method actually works, and in this way offers a proof of concept. To provide such a result, the algorithm should be able to find the points in 2D correctly and transforming them to 3D, also in the case of occlusion, without qualitative serious errors. For the second question, it has to be investigated if the algorithm delivers similar tracking results during similar environmental conditions. The system should deliver similar results while running the algorithm on the same data set multiple times. Secondly, reproducibility also implies that multiple data sets of a single person should deliver qualitatively good results, also when measurements are performed in a slightly different environmental conditions. This in fact is also a test whether the PCA model trained on a person is valid for other measurements on that person. The latter question deals with different data sets of different persons, and whether the PCA model trained on one person is usable for other persons. The following experiments have been run in order to be able to answer these questions:

1. **Experiment 1:** Train the PCA model using frames from movie 1 of person A, and perform tracking on movie 1 of person A.
2. **Experiment 2:** Train the PCA model using frames from movie 1 of person A, and perform tracking on movie 2 of person A.
3. **Experiment 3:** Train the PCA model using frames from movie 1 of person A, and perform tracking on movie 2 of person B.

Data sets were taken in a similar way as will be done during measurements in the OR. The process of preparing and executing the experiment can be seen in figure 10.1. Three data sets have been collected in this way, of which two originate from the same person. The videos were recorded using the initial setup, using two cameras operating at a resolution of 720×1280 pixels and a frame rate of 30 Hz. Figure 10.2 shows an example of the state of a person during data acquisition.

- A person-specific marker bandage was prepared and was stuck on the tongue (for details, see section 7.3)
- Additional facial markers were applied, of which the following locations were being used:
 - Tip of the nose (for vertical face orientation)
 - Center of the chin (for vertical face orientation)
 - Left cheek (for horizontal face orientation)
 - Right cheek (for horizontal face orientation)
- The person was placed in front a setup consisting of two cameras and several light sources.
- The person used a set of clips in order to open his mouth wide open.
- A visual synchronization signal (a light source being activated) was placed in front of the already-running cameras.
- The person performed some tongue motions, including protrusion, retraction, and left-to-right motion.

Figure 10.1: Protocol for taking measurements for qualitative analysis.



Figure 10.2: View of the camera of a subject during data acquisition.

10.2 Quantitative experiments

This section deals with the accuracy of the system in realistic settings. Furthermore it is required to find out how the precision of the system is affected when altering the nature of the PCA model. The following questions will have to be answered:

- What is the reconstruction accuracy of the system, both in the cases with- and without (partly) occlusion of markers?
- Which of the two proposed methods of constructing a PCA model, either conventional or gappy training, provides best tracking results?
- How does performance degrade as a result of 'polluting' the PCA model?

The first question deals with the basic accuracy of the system with and without occlusion of markers. The accuracy in this case is defined by the reconstruction error of the 3D position of the tracked markers, which is the Euclidian difference between the tracked 3D position and the true 3D position. It may be clear that the quality of the PCA model has an influence on this. The second question deals with the training method of the PCA model. The third question deals with the quality of the PCA model. Performing tracking on a person with the use of a model trained on that same person will expectantly yield better results than when the training image set is 'polluted' with images of different persons. This 'polluting' step, however, will make the PCA model more general and usable for more persons, which is a desired state.

One of the problems concerning precision analysis is how to secure a good ground truth to compare the tracking results with. A ground truth could be established by using another sensory system of which the (relative high) precision is known. An EM tracker for instance could provide such a ground truth. In this method, small coils can be placed underneath the markers on the tongue, after which the person's head is placed in a generated EM field. The tracker is then able to determine the 3D position of those sensors based on the received EM-signal. Such a system provides an accuracy of around 2mm (95% confidence level), not offering the desired precision. An additional problem lies in the fact that the coils are placed underneath the markers, introducing an error dependent on tongue state. Additionally the difficulty of aligning the EM-trackers' and the cameras' coordinate systems is a source of errors.

A secondary method for establishing a ground truth is by using the 3dMD imaging device, a system able to create 3D images with a geometric accuracy of 0.2mm. This system, however, is very slow in data acquisition and therefore cannot be used for dynamic measurements. The quantitative experiments have therefore been subdivided into two groups: static experiments and dynamic experiments. In the first case, a valid ground truth will be present for accuracy measurements by using the 3dMD device. In the second method, this ground truth will be only partly valid (this will be explained further on).

In both cases, measurements were performed onto a rigid tongue phantom rather than a realistic, deformable tongue. This choice has been taken as this is the only way to provide a good ground truth considering the available hardware. The can be seen in figure 10.3. It consists of an egg-shaped model, which actually resembles the size and shape of a tongue relatively well. Markers have been applied onto the model based on a realistic tongue shape. The phantom was then placed onto a servo motor, allowing it to rotate, mimicking a left-to-right motion. When viewed from the front, several markers will be occluded during various stances of the tongue, simulating occlusion in realistic situations. Facial markers have been fixated to a plane behind the phantom, which can be used for PCA alignment purposes.

The following subsections will describe the static and dynamic experiments. The static experiments will offer the most accurate precision analysis and will therefore provide an answer for the first quantitative question. The dynamic experiments will only be able to present a relative error (this will be explained further on), and can be used to give a dynamic error measure. These experiments will also be used to verify the influence of PCA training on the performance.

10.2.1 Static experiments

During these experiments, the phantom will be set in several, fixed positions. Then, both the 3dMD device and the camera setup will acquire data from that setup. The 3D-reconstruction of both methods will be compared after alignment of the different coordinate systems the setups. This alignment transformation will be based on a alignment object, which consist of a white sheet with several dots marking corresponding 3D-points in both coordinate systems. The following experiments will be executed:

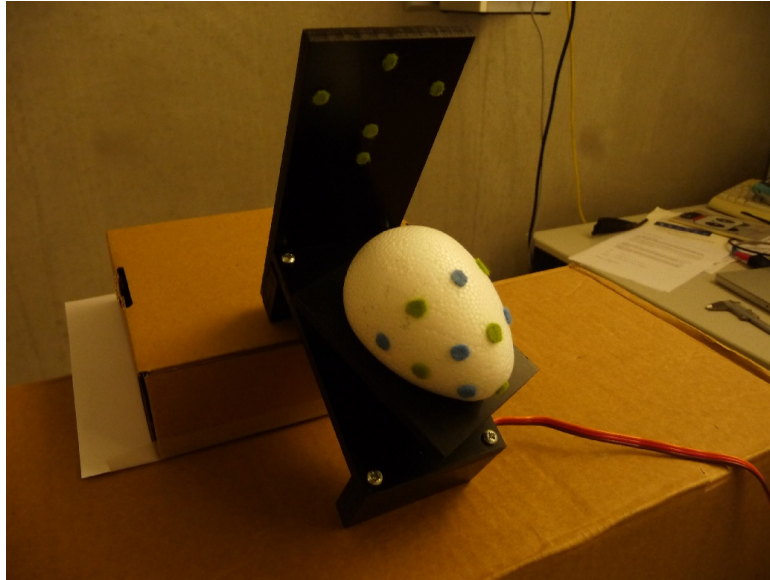


Figure 10.3: An image of the phantom used for analyzing the precision of the vision system.

- **Experiment 4:** Perform static accuracy analysis by comparing the 3D reconstructed points of a 2-camera setup with the ground truth as measured by the 3dMD device.
- **Experiment 5:** Perform static accuracy analysis by comparing the 3D reconstructed points of a 3-camera setup with the ground truth as measured by the 3dMD device.

The process of performing these measurements can be seen in figure 10.4. The 3dMD device takes images of both the tongue model and the alignment object. The 3D coordinates of the markers are then selected manually using specialized software (MeshLab). The camera system also takes measurements of these objects, and additionally takes images of the calibration cube. The 3D alignment points can then be estimated by using equation 5.4. The 3D tongue shape is found by means of the tracking algorithm. In order to compare the 3D marker locations of both systems, an alignment step has to be performed onto the marker coordinates as calculated by the 3dMD device. A rigid transform estimation of the 3D alignment shapes offers such a solution. A suitable error measure is by analyzing the Euclidian error between the tracking result and the ground truth (as provided by the 3dMD device) for each marker. In some situations also the 3dMD device is not able to find all 3D points due to occlusion. In those cases, a complete 3D shape (also originating from the 3dMD device) is taken, and aligned with the non-complete shape by a rigid body transform, based on point correspondences between the two.

The setup used for the experiments can be seen in figure 10.5. As can be seen, the object is placed in front of both the camera system and the 3dMD device. The camera setup is not blocking the view of the latter system, as the cameras of the latter system are placed sufficiently far apart. Although a 3-camera setup can be seen, for experiment 4 only 2 cameras have been used (which are spaced less far apart than the outer cameras of the 3-camera setup). Although static experiments have been executed, the cameras operated in video-mode in order to obtain frames under similar conditions as compared to the dynamic experiments. The cameras were operating at a resolution of 720×1280 pixels and a frame rate of 30 Hz.

10.2.2 Dynamic experiments

A slightly altered method is used for a dynamic analysis of the system. The phantom is not set to fixed angles, but is rotating in time, while no ground truth is available. Instead, only a static ground truth is available. In the processing step, the accuracy will be evaluated while training the PCA model in several ways. The following experiments were performed:

- **Experiment 6:** Run the tracking algorithm by using 7 images of the phantom for PCA training, and using conventional PCA training.
- **Experiment 7:** Run the tracking algorithm by using 7 images of the phantom for PCA training, and using gappy PCA training.
- **Experiment 8:** Running the algorithm by using 7 images of the phantom and a variable amount of images of other shapes for PCA training, using conventional PCA training.

The first experiment simulates an ideal trained PCA situation, by using only training images of the phantom itself. The tracking result can be compared with that of the static case. With the second experiment, a

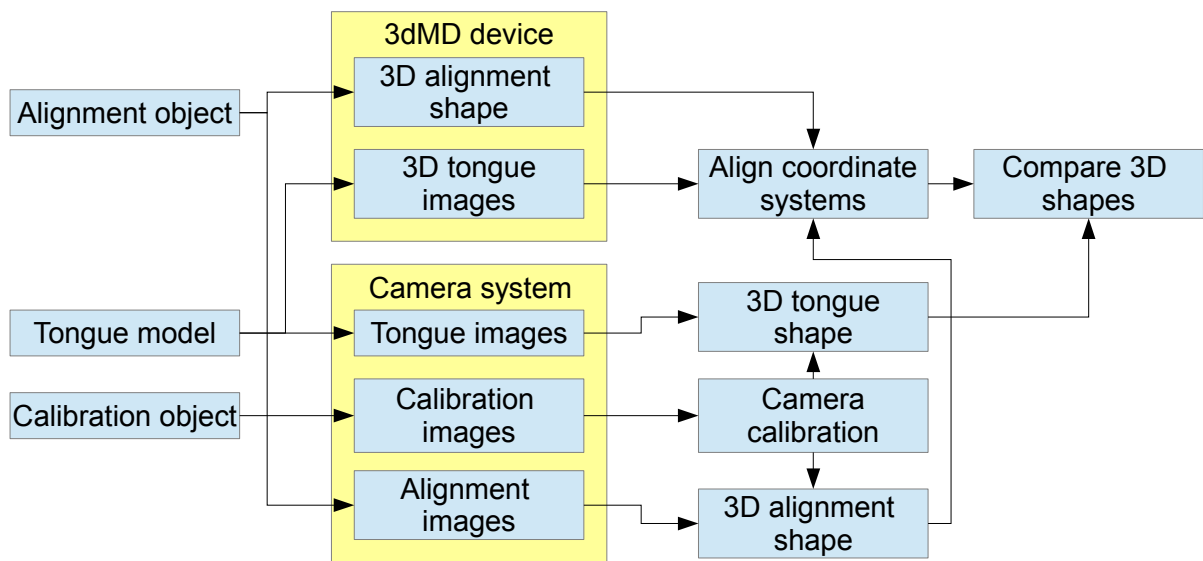


Figure 10.4: Block diagram of the experiment evaluating the static accuracy of the system with respect to the ground truth provided by the 3dMD device.

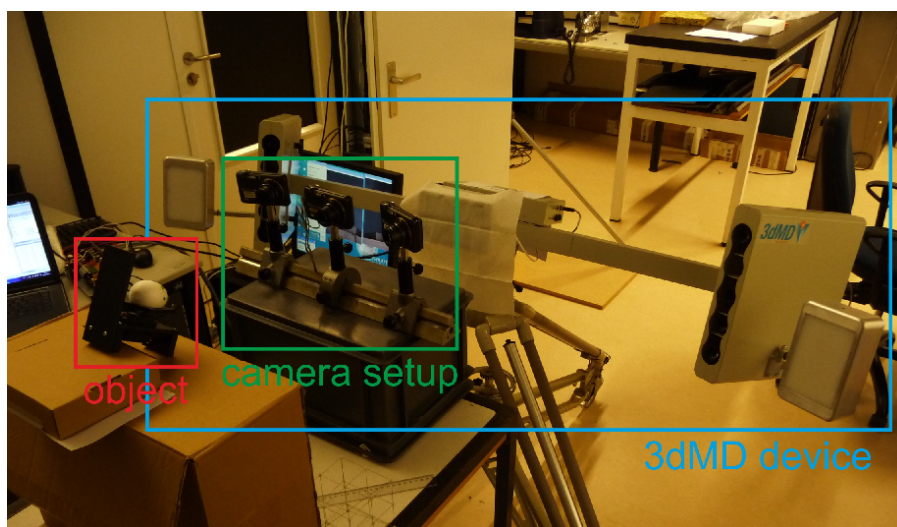


Figure 10.5: Setup used for the static accuracy experiments.

comparison can be made between gappy and non-gappy training. Finally, the last experiment simulates the influence of less-ideal PCA models on the tracking result. The use of more shapes not matching the exact shape of the phantom introduces more variance in the data set. The result is twofold: first of all, there is more shape variance contained within the training set, requiring more PCA components in order to achieve the same percental coverage of all variance contained within the set. More degrees of freedom offer a larger solution space, which also allows the estimation to take on more faulty solutions, making the algorithm more prone to errors. Secondly, when the percentage of images of the phantom within the set of training images becomes smaller, the most important PCA-components will be less focused towards explaining the variance between the different states of the phantom.

Figure 10.6 gives an overview of performing these measurements. As can be seen, the process shows similarities compared to the static situation. The ground truth in the dynamic experiments consists of a rigid 3D-shape. This shape is rotated towards the 3D shape as reconstructed by the tracking algorithm, using only the center markers of the tongue, which are all visible to both cameras in all frames (again, a rigid body transform is used). Also during these experiments, cameras are operating at a resolution of 720×1280 pixels and a frame rate of 30 Hz.

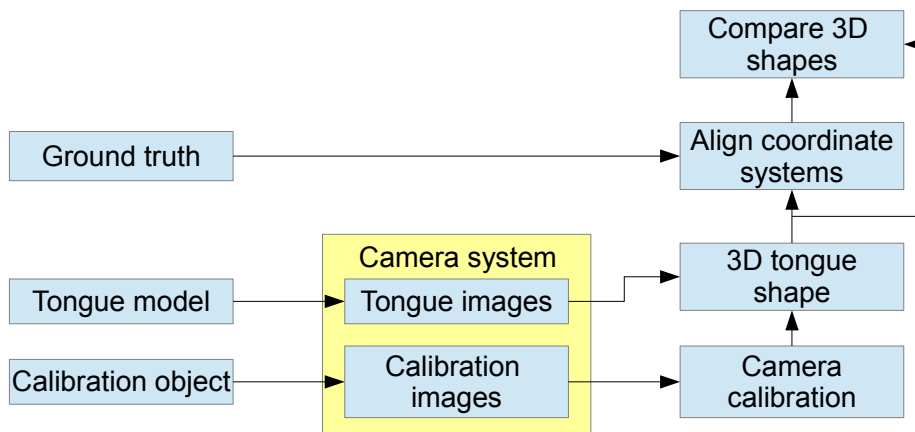


Figure 10.6: Block diagram of the experiment evaluating the dynamic relative accuracy of the system.

The motion as performed by the phantom is depicted in figure 10.7. As can be seen, the phantom starts in a neutral angle, fully rotating to the extreme right position, then rotating to the extreme left position, then returning to the center position again. In fact, the static experiment involve five equally-spaced angle states between the outer most extreme positions.

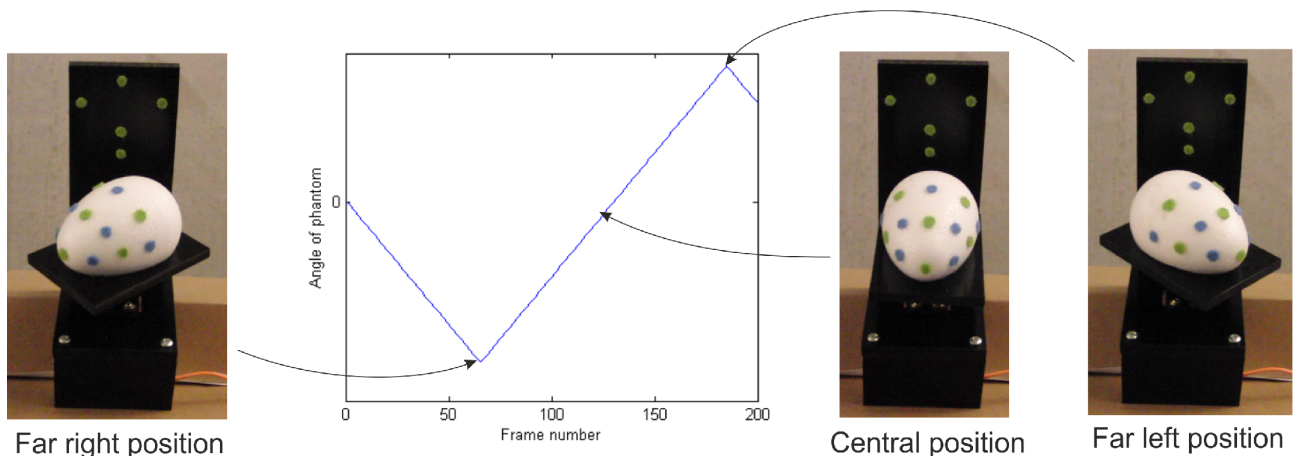


Figure 10.7: Motion of the phantom in the movies used for precision analysis.

Chapter 11

Results

This chapter will discuss the results of the experiments described in the previous chapter. The qualitative results will be discussed first, discussing whether the method is suitable at all and under which conditions. The quantitative results will provide an accuracy analysis, and how this varies under influence of modifications to the PCA model.

11.1 Qualitative results

These experiments were performed on real persons and do not deal with accuracy, but rather with qualitative tracking. In order to verify if the method works, the algorithm must be successful and not lose track of the true shape of the tongue. A good indication for this is by investigating the PCA state of the tongue, and comparing this with the PCA state resulting from marker image coordinates as selected by a human. If tracking was successful, these results should overlap nicely.

11.1.1 Experiment 1

A single movie was used to train the PCA model based on 40 equally-spaced temporal frames. The result of this specific PCA training can be seen in figure 6.6. Four free components describe 92.23% off the variance of the training set.

The tracking algorithm then was run onto the same movie. The process runs successfully, but only if critical parameters (such as marker template size and size of the search area) are chosen well. Figure 11.1 gives an overview of the tongue states during the movie. The (smoothened) PCA components resulting from the tracking process can be seen in figure 11.2. This graph also includes the components subtracted from the human-selected image coordinates.

It can be observed that the results in most cases overlap quite good, and therefore it can be concluded that the tracking algorithm works. Note that in the case of discrepancies the source of errors may not only be caused by faulty tracking, but can also be the result of errors in the manual selection of (occluded) markers.

Closely investigation will show that the results are conform the movie. For instance, in frames 65 and 95, the tongue is moved to the left side of the mouth, while in frame 75 the tongue is located in a relative right-oriented position. When looking at the PCA result, maxima can be observed in the first component, which is actually responsible for describing the left-to-right motion. Valleys can be observed when the tongue has moved to its left position while a peak can be observed in the other case. These states can also be observed in the x-translation component, with peaks during the tongue's left state and valleys during the tongue's right state.

The second component, responsible for describing the tilting of the tongue, can also be clearly seen in the result. In frame 65 the tongue is relative tilted upward, while in frames 115 and 155 the tilting is more downward. This can be observed in component 2 of the tracking result in the form of a maximum for the upward tilted case, and minima for the downward tilting.

When looking more to details of the translational components, there is one very clear valley in the z-translation with a magnitude of around 3cm. This corresponds with the subtraction of the tongue within the mouth. The scaling of the tongue remains closely around factor 1 for the complete movie. This is to be expected, as the PCA model has been trained on this very person.

11.1.2 Experiment 2

In this experiment, the same PCA model was used to track a second data set from the same person. The second data set was taken on a different day than the first, having built up and calibrated the camera system anew,

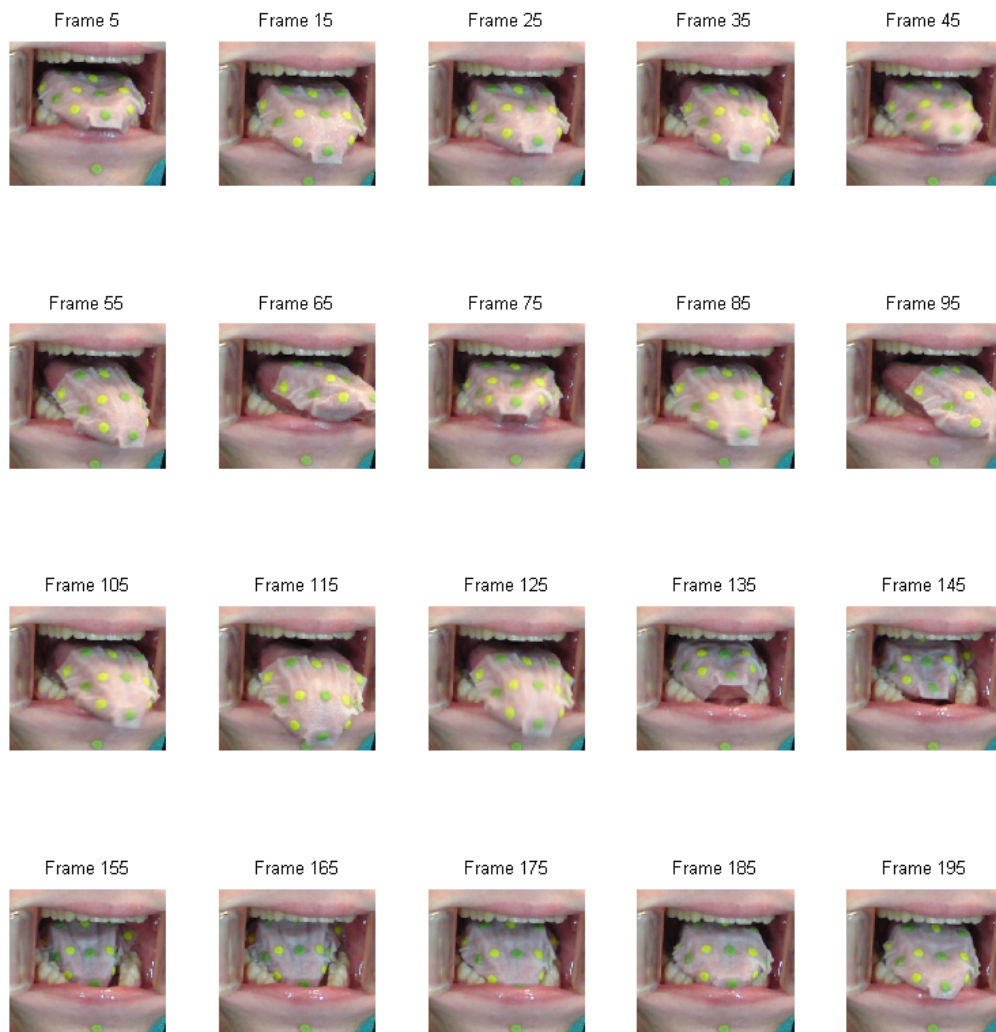


Figure 11.1: Several frames of a movie of the tongue.

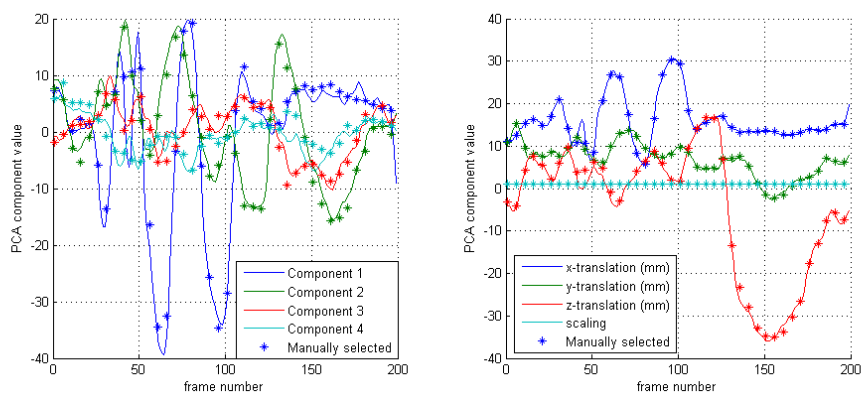


Figure 11.2: Smoothened PCA tracking result. Both automatically subtracted PCA components (solid line) and those subtracted from manual selected markers (stars) are plotted.

and having applied a new but similar marker layout. It was to be expected that because the tongue shape and motions are similar, as well as the marker layout, tracking should be possible. However, when the new data set would contain different motions not included in the first set, the process might go wrong.

The result of the tracking process can be seen in figure 11.3. Again we can see that the tracked components overlap the manually selected results quite nicely, with a few outliers. Analyzing the scaling of the tongue more closely shows that there is some variation in size, but that this is swinging around the value of 1. When evaluating the results more closely, we can identify the situations as indicated in table 11.1. These results match with the visual data.

Based on the high correlation between the tracked PCA components and those originating from manual selection, it can be concluded that the tracking method is reproducible on a single person, under similar environmental conditions.

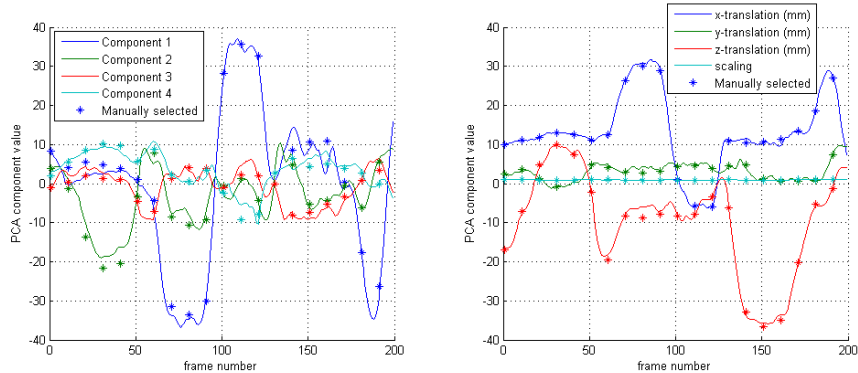


Figure 11.3: Tracking result when having trained on a different data set of the same person.

Table 11.1: Tongue state based on PCA-components.

| Frames | Tongue state | Indicating components |
|---------|--------------|------------------------|
| 25-50 | Protrusion | Comp. 2, z-translation |
| 60-100 | Left | Comp. 1, x-translation |
| 100-125 | Right | Comp. 1, x-translation |
| 130-170 | Retraction | z-translation |
| 180-190 | Left | Comp. 1, x-translation |

11.1.3 Experiment 3

In this experiment, again the same PCA model was used to perform tracking on a different data set taken from a second person. It was to be expected that due to a different tongue shape and different tongue motions the PCA model would not be valid anymore, and would deliver worse results, even if the tracking process would be successful.

The processing step indeed repeatedly went wrong such that no usable result was generated. Generally, after 30 frames, during a protruding motion, the algorithm lost track after which the detected PCA components increased to unrealistic values. Therefore, an additional experiment was performed in which several frames of the data set of the second person was added to the set of training images. This set now consists of 40 frames of the first person and 10 of the second person. Still, four free PCA components were selected. Tracking is now successful, of which the result can be seen in figure 11.4. As can be seen, the estimated PCA component follow those of the manually selected markers in most cases, with a few outliers (for instance, see the first component at frame 160 and the fourth component at frames 40 and 60). Furthermore, note that the scaling factor changes over time to a greater extent compared to the previous experiments. Zooming in proves that in frames 160-170, the scaling factor drops to 0.4.

From this experiment, it can be concluded that it is not enough to train the PCA model on a single person only. When tracking the tongue of a specific person, the PCA model must be able to explain enough inter-personal variance to cover the tongue states of that specific person. A general PCA model suitable for most individuals should thus be trained on a sufficiently large population. As the system will be used for people with deviant tongue shapes due to tumors, the PCA model should also be trained onto such specific tongue shapes.

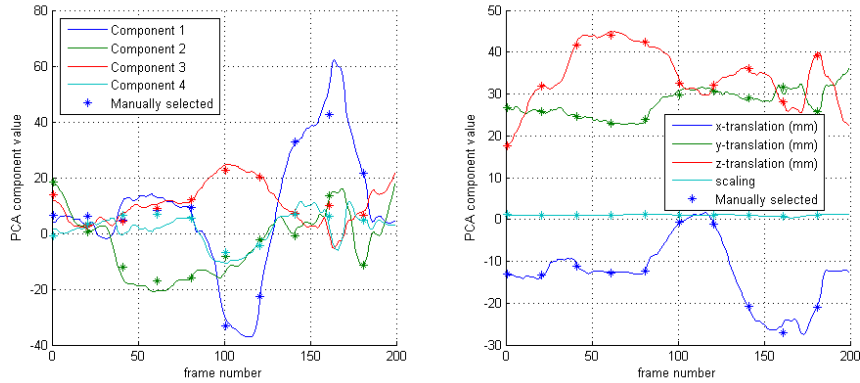


Figure 11.4: Tracking result of person B when having trained a dataset based on 40 frames of person A and 10 frames of person B.

11.2 Quantitative results

These experiments were performed on real persons and do not deal with accuracy, but rather with qualitative tracking. In order to verify if the method works, the algorithm must be successful and not lose track of the true shape of the tongue. A good indication for this is by investigating the PCA state of the tongue, and comparing this with the PCA state resulting from marker image coordinates as selected by a human. If tracking was successful, these results should overlap nicely.

These experiments were performed on the tongue phantom. Static experiments (6 and 7) will describe measurements focused on accuracy, while dynamic experiments (8 to 10) will describe changes in relative accuracy as a function of a training the PCA model differently.

11.2.1 Static results

During these experiments, a PCA model has been constructed from 7 images of the phantom itself. It seems that next to three translation and one scaling components, only two free components are sufficient to describe around 99% of all variance. The model is thus considered to be very good.

Experiment 4

During this experiment, a 2-camera setup was used. Error of the tracked markers have been expressed in the Euclidian distance between the tracked markers and corresponding ground truth originating from the 3dMD device. The results are presented in figure 11.5. Five distinct graphs can be identified, each belonging to a measurement of the phantom viewed in a different angle. The markers are numbered following the scheme as presented in figure 7.6.

As can be seen, the size of the error per marker is quite distinct. Low-error markers are numbers 1, 2, 5, 6, 10, 11 and 12. These are located in the front and center position of the tongue, and are in most phantom states visible. This thus is a logical result. High-error markers are number 4, 8, 9 and 13. These are all located on the side position of the tongue, and often not visible in at least one cameras. This is thus also a logical result. It is to be expected that when markers are not visible, the accuracy will degrade. Therefore, in the far right position it is to be expected that marker numbers 4 and 9 provide high error rates, while the errors of markers 8 and 13 will be minimal. This trend can be indeed observed. The opposite is to be expected for the far left position, which is also valid.

The maximum error of a marker is around 5mm in outer extreme positions. Non-occluded markers do never surpass an error of 2mm. Considering the central position during which no occlusion occurs, marker error remains in most cases below 1mm, a desired state.

The error averaged over all markers per phantom state is shown in table 11.2. As can be observed, the average error is most extreme in the outer positions, which can be explained by the fact that in those cases most occlusion occurs, with a higher error for those markers as a result. The average error in all cases remains well below 2mm, and often goes below 1mm, which is a nice result.

Furthermore, a look can be given towards the direction in which the largest error occurs. It is expected that the error will be largest in the z-direction, as this represents the depth of the system. As the cameras are placed relative close to each other, this means relative few information is obtained by the cameras in that direction.

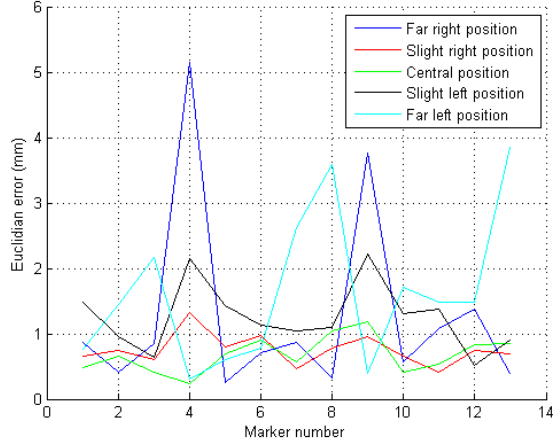


Figure 11.5: Static error per marker type for different tongue states. Camera system consists of 2 cameras. Marker numbering is as depicted in figure 7.6

Table 11.2: Euclidian error per phantom state, averaged over all markers, considering the 2-camera setup.

| Tongue state | Average Euclidian error (mm) |
|-----------------------|------------------------------|
| Far right position | 1.28 |
| Slight right position | 0.76 |
| Central position | 0.68 |
| Slight left position | 1.25 |
| Far left position | 1.63 |

Only the image of the phantom in its central position is considered. The reconstruction error can be seen in table 11.3.

As can be concluded by studying the absolute average error, the expectation does not hold. The error in the Z-direction is slightly smaller than that of the Y-direction. There is no clear explanation for this phenomenon, but it is expected that it is a result of the nature of the PCA model. When looking at the average error, it can be seen that there is a small negative offset for all markers. This indicates that there might be a systematic offset error in the estimation process, but one that is not very large. It is expected that due to the small amount of markers used to track, the systematic offset is rather caused by quantization noise, and will be closer to zero when averaging over more measurements.

Table 11.3: Reconstruction error in the different dimensions.

| Dimension | Average error [mm] | Absolute average error [mm] |
|-----------|--------------------|-----------------------------|
| X | -0.15 | 0.29 |
| Y | -0.21 | 0.39 |
| Z | -0.19 | 0.36 |

Experiment 5

This experiment is identical as the previous experiment, apart from the fact that now three cameras were used to observe the phantom. The addition of an extra camera allowed us to space the outer cameras further apart than compared to the case with only 2 cameras without losing the ability to observe each marker with 2 cameras simultaneously. It was to be expected that due to having a larger angular view of the phantom, the reconstruction error of the outer markers would be lower, as these can be observed with at least one camera in a larger amount of phantom angle states compared to the 2-camera setup. Furthermore, as the outer cameras are spaced further apart, it would be expected that the reconstruction error of the center markers would be lower. This latter statement can be illustrated by figure 5.1, which shows that the uncertainty area should become smaller when increasing the angle between the cameras (with an optimal angle of 90 degrees, of course).

Figure 11.5 presents the results of the measurements in a similar way as done in experiment 4. The first thing that can be observed, is that the highest peaks have significantly decreased in size, as predicted. A second

observation is that the general shape of the graphs match those of experiment 4; markers that are located on the right side of the tongue (4, 9) show a high error when occluded in the far right positions, while the opposite is true in the far left positions. Markers located on the left side of the tongue (8, 13) show a similar behavior, having a relative high error in the far left positions, while having a low error in the far right positions. Markers placed more central, having a more constant visibility across the complete angular range, have a more consistent error. Comparing these values with those of experiment 4, it can be seen that in most cases the error does not remain below 1mm as was the case for experiment 4. A more close look is given to the average error.

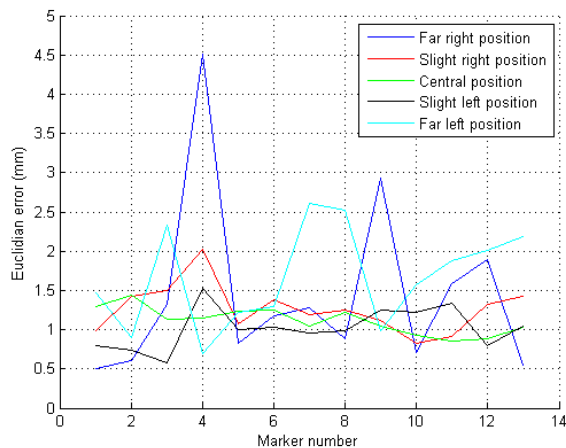


Figure 11.6: Static error per marker type for different tongue states. Camera system consists of 3 cameras.

The error averaged over all markers is indicated in table 11.4. A similar trend can be observed: in the extreme positions, the error generally is larger compared to the more central positions. However, when comparing these values with those of experiment 4, it becomes clear that these results are less well (even in the more extreme positions), which was not expected. An explanation of this phenomenon is that by placing the outer cameras further apart, the outer cameras will suffer from occlusion more often compared to the 2-camera setup. When that error is not large enough to be detected by the outlier-detection algorithm, it distorts the state estimation of the phantom, resulting in a relative higher error for all markers. To overcome this problem, a smart marker-occlusion prediction algorithm can be used which can predict which markers will be occluded during a certain state of the model. This however will be in need of a good general PCA model, which is currently not available, but may be constructed in the future.

Table 11.4: Euclidian error per phantom state, averaged over all markers, considering the 3-camera setup.

| Tongue state | Average Euclidian error [mm] |
|-----------------------|------------------------------|
| Far right position | 1.44 |
| Slight right position | 1.26 |
| Central position | 1.11 |
| Slight left position | 1.02 |
| Far left position | 1.67 |

11.2.2 Dynamic results

During these experiments, a PCA model will be constructed in several ways. During training the PCA model, the approach was to use a model describing a similar percentage of the variance of the training set (around 95%). Therefore, the PCA model in the different experiments can have a different amount of free components.

In this experiments, measurements were again performed on the tongue phantom. Now the error has been calculated as a function of time. The error measure is the Euclidian distance between the tracked 3D position of the marker with respect to the ground truth. This ground truth consists of the marker positions of the tongue phantom in a neutral position. Alignment of this model was performed by a rigid body transform between the center markers, which were visible in all situations. This thus provides only a relative error measure, as a structural error across all markers will be neglected.

Experiment 6

An identical PCA model as in experiments 4 and 5 has been used, only needing two free components to describe around 99% of all variance. Error resulting from the tracking algorithm can be seen in figure 11.7. The resulting measured error as a function of time can be seen in figure 11.7. The left image shows the minimum, average and maximum Euclidian error of any marker as a function of the frame number. The right image shows the Euclidian error per marker, where several marker 'groups' have a different color. A distinction is made between the two markers on the left side of the phantom (4 and 9), the two on the right side of the phantom (8, 13) and all others (for marker numbering, see figure 7.6).

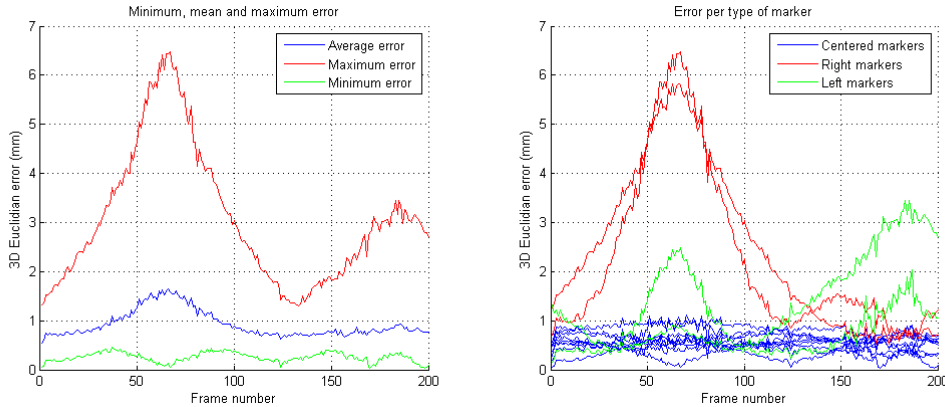


Figure 11.7: Marker tracking error using conventional PCA training with 7 phantom images.

Looking at the left image, two large peaks can be observed both in the maximum and average error. When comparing the shape of the maximum error with figure 10.7, it can be seen that the peaks correspond with the most extreme angular positions of the phantom, something that would be expected as in those situations most occlusion occurs. When comparing these results with those of figure 11.2, it can be seen that there are some differences. The average error in the most extreme right position is roughly 1.7mm compared to 1.28 for the static measurement, and the average error in the most extreme left position is roughly 0.9 compared to 1.63 for the static measurement. In the central position, the errors are roughly similar. The source of discrepancies can be the fact that ground truth during this experiment is determined in a different way, which may have introduced errors. Furthermore, the data sets were taken on different moment, under slightly different conditions, where for instance the camera spacing and phantom orientation could have been slightly different.

When looking at the right part of the image, it can be seen that the large peaks are solely caused by the four markers placed at the sides of the phantom. These peaks reach their maxima at the moment at which the phantom has rotated to its extreme points. The first peak is mainly contributed for by the markers located on the right side of the phantom, while the second peak is caused by the markers located on the left side of the phantom. This sounds very reasonable, as those groups of markers suffer from occlusion at those moments in time, resulting in measurement errors. Not only occlusion during tracking could be the cause of these errors, also the result of occlusion during PCA training and therefore faulty training is a source.

It can be seen that the maximum error does not exceed 6.5mm, while the mean error remains below 1.7mm at all times. In the case of no occlusion, the maximum error remains below 2mm while the mean drops well below 1mm. When looking only at the central markers, the error barely rises up over 1mm along the complete tracking period, which is a result to be content with.

Dynamic results provide the opportunity to study whether the estimation uncertainty corresponds with the observed errors. If this is the case, it will be an indication that the mathematical models used for Kalman filtering are good enough. Several parameters of the model need to be tuned in order to provide well measurement results, especially the values used for the measurement- and system noise. During the first run of the tracking algorithm, system noise needs to be large enough to let the measurements determine the tongue state completely. The measurement noise is proportional to the uncertainty of the template matching process (see equation 5.11). The problem with choosing the value of this, is that the SSD-criterion does not yield a probability density function which is Gaussian distributed, something the Kalman filter assumes. Reasonable and realistic values can lie within a range of 2-10 pixels, considering that the radius of the marker as observed in the camera images is 10 pixels. The choice of the system noise has to be chosen in such a way that has no significant influence on the state estimation during the first run of the algorithm. During the second run,

when all template matching results are known, this variable should be chosen in such a way that the highest frequency components of the estimated time-dependent PCA components should be damped (as they have a high probability of being noise).

The observed 3D-uncertainty averaged over the all markers after 10 frames of tracking can be seen in table 11.5 after choosing various values for the template matching uncertainty. The first thing that can be observed is that the uncertainties in the various directions differ. The uncertainty in the x- and y-direction are very close, while the uncertainty in the z-direction is roughly a factor three larger. This can be explained by the fact that the camera system has been calibrated in such a way that the z-direction is the depth direction of the system, of which the uncertainty grows as the cameras are placed closer together (for a visualization of this effect, see image 5.1). The second thing that can be observed is that the uncertainty seems to grow linear as a function of the chosen template matching precision. Finally, the numerical values can be compared to those in table 11.3. The measured average absolute error lies around 0.3 mm for all dimensions. The estimation uncertainty lies in the same order size when having chosen a suitable value for the template matching uncertainty. Only the uncertainty in the z-direction shows different behavior than measured in reality. For now, it can be concluded that the Kalman filter delivers reasonable measures for the system uncertainty.

Table 11.5: The resulting uncertainty of the 3D estimation for various chosen values for template matching uncertainty.

| Template matching uncertainty (std) [pixels] | 2 | 5 | 10 |
|--|------|------|------|
| Uncertainty X-direction (std) [mm] | 0.16 | 0.39 | 0.79 |
| Uncertainty Y-direction (std) [mm] | 0.14 | 0.36 | 0.71 |
| Uncertainty Z-direction (std) [mm] | 0.42 | 1.04 | 2.09 |

Finally, the estimated 3D uncertainty can be observed over time, when choosing the system noise either large or small. Figure 11.8 shows these results (only the 3 diagonal variables of the 3x3 covariance matrix are plotted, as the others are less significant). As can be seen, when choosing a large value for the system noise, the estimation uncertainty changes little over time, as would be expected because the system memory does not add significant information to decrease the uncertainty. When choosing the system noise lower, however, it can be seen that the uncertainty drops after some frames, eventually converging to a smaller value. The initial uncertainty only drops after a delay as the prediction uncertainty takes several frames to converge.

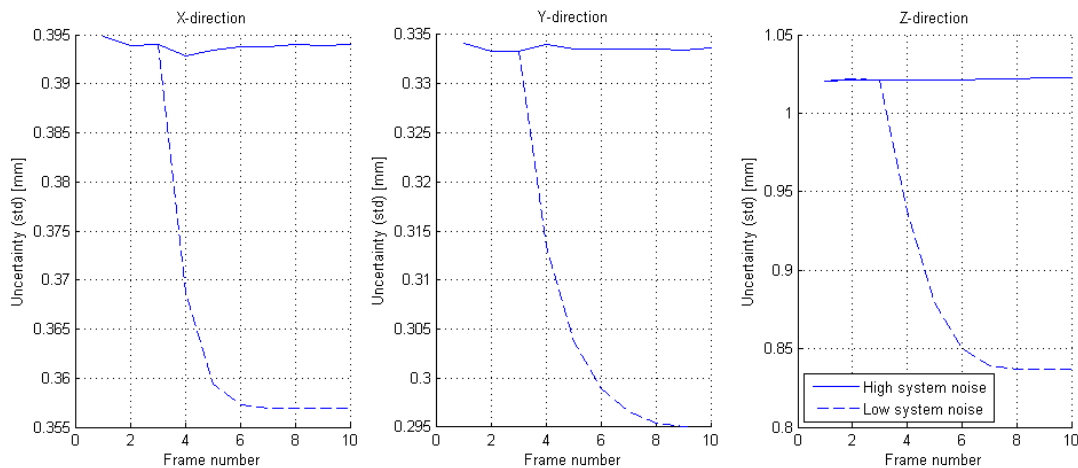


Figure 11.8: Uncertainty of the estimation in all dimensions as a function of the frames, averaged over all markers. Both high and low values for the system noise have been tried.

Experiment 7

Experiment 6 is repeated, but now training the PCA model using the gappy training approach. The same training images are used, and the same number of resulting components are used. The tracking result using this model can be seen in figure 11.9.

Also in this result, two peaks can be clearly identified, again this time being completely determined by the markers placed at the sides of the phantom. The peaks have grown tremendous in size, proving that a significant change in accuracy has taken place in the training step. The maximum error can grow up to around 28mm. The error of the centered markers however has barely grown, just slightly over 1mm across the complete trajectory.

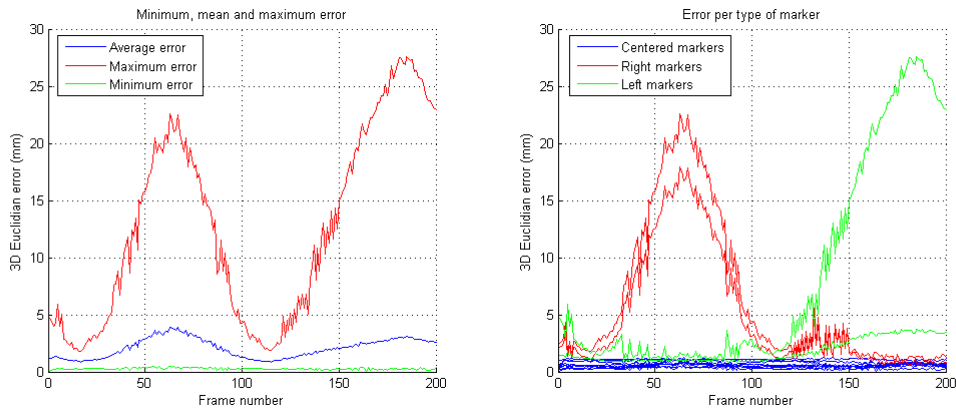


Figure 11.9: Marker tracking error using gappy PCA training with 7 phantom images.

The cause of the high peak problem lies in the fact that occlusion of the outer markers occurs systematically, and therefore there are not records of their position when the phantom has rotated beyond a critical angle. From that point on, the gappy PCA training method can let the coordinates of that specific marker take on any desired position while minimizing the error of known marker coordinates.

With this result, it has been proven that training with the gappy PCA method in this implementation yields worse results compared to conventional training, in which an person manually estimates the location of the occluded markers.

Experiment 8

This experiment was run with conventional PCA training again, but now adding other 3D shapes to the training sequence based on real tongue measurements of up to 2 persons (actually these were based on the data sets used for the qualitative experiments). This 'pollution' step simulates the creation of a general tongue PCA model, which can be used for multiple persons, although expectantly delivering less accuracy compared to a PCA model fully optimized for a single person. Three different runs of the tracking algorithm was performed on the same data set of the phantom. Table 11.6 gives an overview of the details of training the PCA model during each of those runs.

Table 11.6: The different PCA training settings for the different simulations.

| Number of simulation | PCA training images | Number of subtracted free components | Amount of variance of training set explained |
|----------------------|---|--------------------------------------|--|
| 1 | - 7 images of phantom - 7 tongue images of person A | 3 | 95% |
| 2 | - 7 images of phantom - 21 tongue images of person A | 4 | 95% |
| 3 | - 7 images of phantom - 20 tongue images of person A - 10 tongue images of person B | 5 | 96% |

The results can be seen in figures 11.10 to 11.12. Again, for all graphs, two peaks can be clearly observed, corresponding to the situations in which occlusion becomes more apparent. There is some but no fundamental difference in shape and size of these peaks. It would be expected that accuracy would generally drop as more and different shapes would be added to the PCA training phase, but these results do not seem to follow this expectation. Observing the average error across the situations in which the PCA model is being increasingly more polluted, no convincing decrease in system accuracy can be observed. This latter however is not entirely true when comparing the average error to that of figure 11.7, in which no pollution of the PCA model had taken place.

The peaks of the graphs are still mainly caused by the markers located on the sides of the model. The error of the centered markers has grown however, now more significantly contributing to the average error of especially the second peak. In many cases, the error of these centered markers remains below 2mm. Only in the most extreme positions of the tongue model the values surpass the value of 2mm. In the center position of the phantom, the error of the central markers remains below 1.5mm in all cases, while those of the markers

located on the phantoms' side drop below 3mm.

For now, we can conclude that degradation of accuracy will occur when polluting the PCA model with measurements of different tongue shape in order to obtain a general PCA model usable for many persons. However, this degradation of accuracy does not seem to be clearly dependent on the amount of pollution.

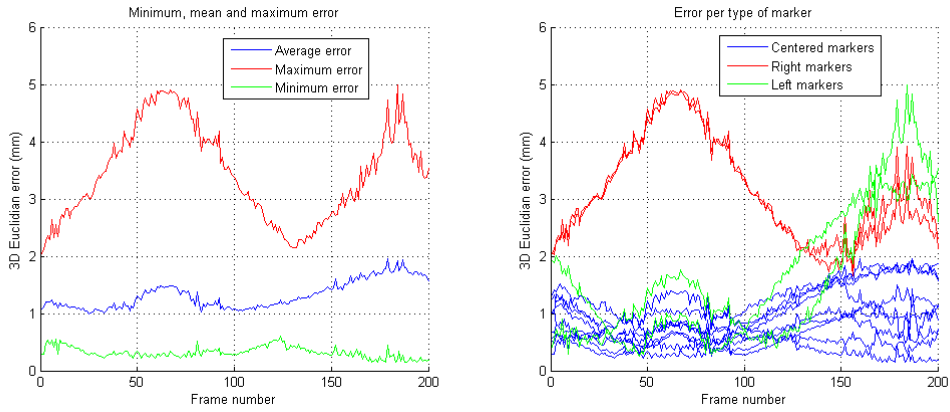


Figure 11.10: Marker tracking error using conventional PCA training with 7 phantom images and 7 real tongue images.

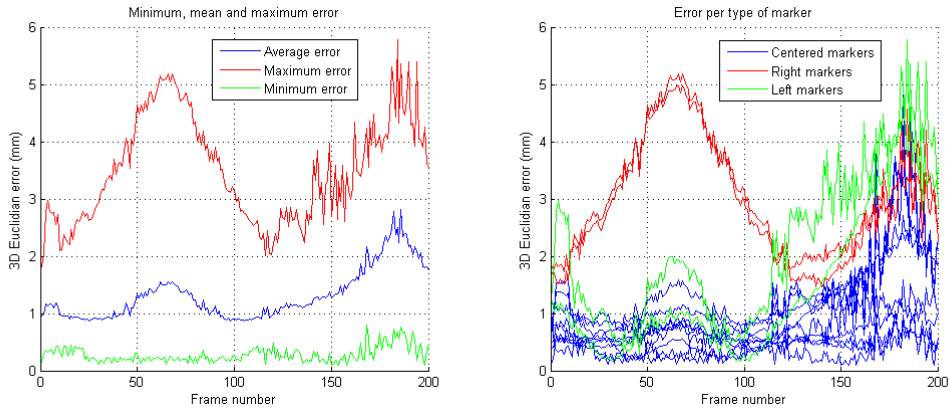


Figure 11.11: Marker tracking error using conventional PCA training with 7 phantom images and 21 real tongue images.

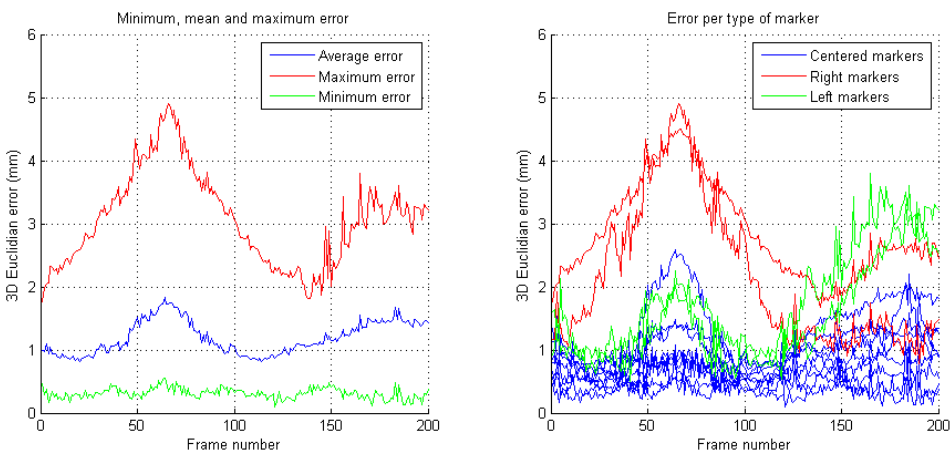


Figure 11.12: Marker tracking error using conventional PCA training with 7 phantom images and 30 real tongue images, originating from two test persons.

Chapter 12

Conclusions

Oral cancer is a disease which can significantly affect one's oral abilities, including speech, food transport, chewing and swallowing. There are several treatment methods, but the choice of treatment is determined by subjective means. The Dynamic Virtual Surgery project aims to develop a system which allows to study post-operative function loss by pre-operative simulations. For tongue cancers, this means that a good patient-specific tongue model must be constructed. The training process of such a model involves tracking the tongue shape in three dimensions during an operation, while simultaneously acquiring EMG data of the tongue.

12.1 Thesis overview

This project was focused on developing a multi-camera system in order to track the 3D shape of the tongue in an OR-environment. Such a system, consisting of 3 cameras, is placed in front of the patient with widely opened mouth. Markers are applied onto the tongue on specific locations, defining high-contrast landmarks. A tracking script then offline calculates the state of the tongue markers in 3D, involving template matching for marker detection and Kalman filtering for 3D reconstruction. To deal with occlusion and measurement errors, a *principal component model* (PCA model) has been proposed which can be used for detecting and correcting outliers.

12.2 Camera calibration

A camera calibration algorithm has been written which can obtain a linear camera model based on a single image of a 3D cube using the *Direct Linear Transform* algorithm. Although the accuracy is not as high as nonlinear methods, its precision is acceptable and its linear nature is very useful for integration in the Kalman filter.

12.3 Setup

Although a design for a 3-camera setup for use within the operating room (OR) has been made on paper, the setup has not been completed as of this moment, due to various sources of delay in the project. All experiments have been run using the initial setup consisting of two or three consumer model cameras.

12.4 Measurement protocol

A measurement protocol has been designed, describing how to perform reproducible measurements. This involves the creation of a patient-specific bandage with markers applied onto it in a specific grid, which can be stuck directly on the tongue, providing the landmarks for the tracking system. By preparing the bandage beforehand, a certain degree of reproducibility can be guaranteed. From experiments performed on the same person it can be concluded that the degree of reproducibility is sufficient.

Although a rather flexible material has been used for bandage material, it is still a problem that the tongue suffers from slight restriction of freedom.

12.5 PCA model

A PCA model of the tongue has been proposed for two reasons. The first involves detecting and correcting errors in marker localization in the frames taken by the cameras. Errors originate from (partly) occlusion of oral regions and non-ideal templates used for template matching. The second reason is using the PCA state vector in a Kalman filter, tracking the state of the tongue. The use of such a model proves to be successful, but currently has only been trained on either one or two persons, and therefore not usable for a wide selection of people.

One of the problems of the model lies within obtaining it, as it is based a statistical analysis of 3D shapes of the tongue. Such shapes must be reconstructed beforehand by converting tongue shapes from 2D to 3D, a process often involving (partly) occlusion of markers. This is a source of errors.

12.6 Qualitative experiments

A set of measurements on two persons has been performed under realistic conditions. First results proved that the tracking method as proposed in this thesis is feasible.

Furthermore, experiments indicate that tracking attempts on multiple data sets of a single person are possible. For this purpose, a PCA model was trained on one set and used for tracking on another set, while those sets were taken on different days. The fact that this experiment succeeded proves that tracking experiments are reproducible on the same person.

A final tracking experiment was performed on a data set of a second person, using a PCA model trained on a data set of the first person. Tracking proved to be impossible using these conditions. Only when including 3D tongue shapes of the second person to the PCA training set, tracking was successful again. It can be concluded that a general PCA model has to be developed, incorporating enough inter-personal variability.

12.7 Quantitative results

Quantitative results indicate that a 2-camera system can achieve a sub-millimeter accuracy, with an average marker error (3D Euclidian distance) down to 0.68mm in the case of no occlusion of markers. In the case of several occluded markers this accuracy degraded to an average error of 1.63mm. Both results were obtained under the condition that the PCA model is perfectly constructed for the considered tongue shape. A second experiment using three cameras, having a wider angular view of the object, obtained a worse accuracy (with an average error down to 1.02mm in the situation of no occlusion), which was not expected. During occlusion this accuracy degraded to an average error of 1.67mm. These unexpected results may be caused by the fact that the outer cameras were spaced further apart, introducing a higher chance of occlusion of markers for those cameras. Although it was expected that the largest error would occur in the Z-dimension, this is not the case in reality. Uncertainty regions as calculated by the Kalman filter provide realistic values in the same order size as the measured errors.

One of the problems of the PCA model is obtaining the 3D training shapes. A first option is by letting a human estimate the 2D position of potentially occluded markers. A second method is by training the PCA model with the use of a gappy training scheme. Quantitative analysis showed the latter method offers much worse result in the case of systematic occlusion.

One may imagine that a PCA model only trained on training shapes of one person works best for that specific person. Constructing a general PCA model involves including shapes of other persons too. Quantitative experiments indicate that this 'pollution' process results in a slight degradation in system accuracy, but makes the model still usable for tracking. In fact, it was observed that more pollution does not directly lead to a decrease in system accuracy. This indicated that it might be very well feasible to develop a good-working general PCA model which can be used for many persons.

Chapter 13

Recommendations

This project described the development of a multi-camera system able to measure the shape of the tongue in 3D. The result is promising, but improvements are necessary to make the method reliable and user-friendly. This chapter recommends points of improvement and possible directions for further research.

13.1 Setup

A first attempt towards construction of an OR-approved camera system has been made. However, due to various sources of delay this process was not finished. The hardware has been selected and ordered, but must be combined to form a complete system. This does not only involve putting together the hardware, but also requires the construction of a user interface via which the camera system can be controlled.

An option for further research is to determine the optimal angle at which the cameras need to be placed in order to make the reconstruction error as small as possible. This can be done by experimenting with different setups and analyzing their precision, but this can also be done by construction of a visual simulation environment, which generates images based on computer tongue models. This allows one to acquire data sets automatically and perform a huge amount of simulations, and maybe even perform an optimization routine. Such an approach, however, should generate realistic images and therefore requires a realistic 3D tongue model to start with.

13.2 Measurement protocol

A measurement protocol has been developed describing the details of performing measurements on a person. Among other things, this involves the construction of a (reproducible) bandage containing markers and the placement of facial markers. However, only limited effort was put into investigating which marker type (size and shape), marker density and marker grid provides the best tracking results. Furthermore, it has not been evaluated how well the entire tongue shape can be reconstructed using the limited amount of markers. Experimenting with these variables and comparing the quality of the tracking results will be valuable information, as well as provide an answer to the question under which conditions (marker type and grid) the tracking process has a high degree of reproducibility.

The fact that the bandage acts as a movement-restrictive object can be a limiting factor in constructing a good patient-specific model. If that would be the case, it is possible to look for more flexible alternatives for the bandage base material. A different alternative can be the use of colored needles. This however means intentionally hurting the patient, which is not a desired situation.

13.3 PCA model

A PCA model has been proposed, as both its use in outlier correction and for integration in the Kalman state vector was successful. However, as of this moment, this model has only been trained on a maximum of two persons, which offers too few inter-personal descriptive power to be usable for a wide range of patients. A data set must be constructed based on tongue images of many individuals. It is recommended to pick persons of different age, gender and ethnicity in order to gather most variance.

When such a general model is present, one can make the state estimation more robust by predicting outliers based on the PCA state. By studying the states in which occlusion of one or several markers occur, a function can be constructed predicting the occurrence of occluded markers based on the predicted PCA state of the tongue. Excluding those measurements from the state estimation step will expectantly provide better results.

The use of the PCA model can be extended to tracking the occluded surfaces of the tongue. Using advanced imaging methods, such as MRI, also tongue surfaces not visible through the mouth opening can be tracked. In the case occluded surfaces (such as the back side of the tongue) show high correlation with the visible parts of the tongue, a well-trained PCA model can estimate the state of these hidden surfaces given only information about the visible parts of the tongue.

13.4 Marker detection

Marker detection is currently being performed using template matching with pre-determined, disk-shaped templates. However, during tongue motion, the orientation, distance and background of the observed markers may change significantly. Detection can be made more robust by using time- and state-dependent templates, allowing the orientation and size of those templates to vary based on the predicted state of the system. This of course is only possible when a general PCA model has been constructed.

13.5 Kalman tracker

The current Kalman trackers for the facial and tongue markers are based on a white noise acceleration model. Measurement errors can result in a wrongfully tracked state. Especially in the case of the tongue state (using a PCA state vector), this can result in extreme states far from realistic values, unable to recover to the true tongue state. Using an auto-regressive model instead of a white noise acceleration model may improve robustness of the system.

List of Abbreviations

| Abbreviation | Full text | Description |
|--------------|---|---|
| CCD | Charge-Coupled Device | Type of image sensor found in digital cameras |
| CT | (X-ray) Computed Tomography | Medical imaging technique which reconstructs 3D-volume of a patient using X-ray radiation |
| EM (tracker) | ElectroMagnetic (tracker) | Medical navigation technology able to reconstruct the 3D position of small coil sensors |
| EMG | ElectroMyoGraphy | Technique for evaluating and recording the electrical activity produced by skeletal muscles |
| FEM | Finite Element Method/Model | Numerical technique for finding approximate solutions to differential equations and their systems by dividing up a complicated problem into small, inter-related solvable elements. Models may be constructed from a number of interconnected mass-spring elements. |
| IEC | International Electrotechnical Commission | Commission which defines technical standards, under which standards for the safety and effectiveness of medical equipment |
| MRI | Magnetic Resonance Imaging | Medical imaging technique which reconstructs 3D-volume of a patient using magnetic fields |
| OR | Operating Room | Hospital room in which surgical procedures are performed |
| PCA | Principal Component Analysis | Mathematical procedure that using an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables |
| RANSAC | RANdom SAmple Consensus | Iterative procedure to estimate parameters of a model when observations suffer from outliers |
| RGB | Red, Green, Blue | Color model used in digital cameras |
| SAS | Signals And Systems | Research chair at which this research has been performed |
| SSD | Sum of Squared Differences | Mathematical criterion for defining an error as $SSD = \ \mathbf{x} - \mathbf{y}\ = \sum_{i=1}^N (x_i - y_i)^2$ |

Bibliography

- [1] Department of head and neck oncology. "<http://www.hoofdhalskanker.info/>". "[Online; accessed 02-08-2012]".
- [2] D. Beautemps; P. Badin; G. Bailly. Linear degrees of freedom in speech production: Analysis of cineradio- and labio-film data and articulatory-acoustic modeling. *Acoustical Society of America*, pages 2165–2180, 2001.
- [3] A.M. Kreeft; I.B. Tan; C.R. Leemans; A.J.M. Balm. The surgical dilemma in advanced oral and oropharyngeal cancer: how we do it. *Clinical Otolaryngology*, 36:252–279, 2011.
- [4] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.
- [5] J. Bouguet. Camera calibration toolbox for matlab. "http://www.vision.caltech.edu/bouguetj/calib_doc/". "[Online; accessed 26-06-2012]".
- [6] O. Engwall. Combining mri, ema and epg measurements in a three-dimensional tongue model. *Speech communication*, pages 303–329, 2002.
- [7] F. Remondino; C. Fraser. Digital camera calibration methods: considerations and comparisons. *IAPRS*, 36:7, September 2006.
- [8] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, 1989.
- [9] J. Cheng; P. Huang. Real-time mouth tracking and 3d reconstruction. *Congress on Image and Signal Processing*, 4:15224–1528, 2010.
- [10] S. Maeda. Compensatory articulation during speech: evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. *Speech production and modelling*, pages 131–149, 1990.
- [11] L. Liu; B. Sun; N. Wei; C. Hu; M.Q. Meng. A novel marker tracking method based on extended kalman filter for multi-camera optical tracking systems. *Bioinformatics and Biomedical Engineering*, 2011.
- [12] S. Yamamoto; N. Tsumura; T. Nakaguchi; T. Namiki; Y. Kasahara; K. Terasawa; Y. Miyake. Regional image analysis of the tongue color spectrum. *International Journal of Computer Assisted Radiology and Surgery*, 6:143–152, 2011.
- [13] I. Steiner; S. Ouni. Progress in animation of an ema-controlled tongue model for acoustic-visual speech synthesis. *Elektronische sprachsignalverarbeitung*, pages 245–252, 2011.
- [14] J. Heikkilä; O. Silvén. A four-step camera calibration procedure with implicit image correction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 1106, 1997.
- [15] R. Everson; L. Sirovich. Karhunen-loève procedure for gappy data. *Journal of the Optical Society of America*, 12(8):1657–1664, August 1995.
- [16] Z. Liu; H. Wang; H. Xu; S. Song. 3d tongue reconstruction based on multi-view images and finite element. *Advances in information sciences and service sciences (AISS)*, 3, 2011.
- [17] K. Mishima; T. Yamada; K. Fujiwara; T. Sugahara. Development and clinical usage of a motion analysis system for the face: Preliminary report. *Craniofacial Journal*, 41:559–564, 2004.
- [18] B. Lindblom; J. Sundberg. Acoustical consequences of lip, tongue, jaw, and larynx movement. *Acoustical Society of America*, 4 (part 2):1166–1179, 1971.

- [19] C. Tzou. Evolution of the 3-dimensional video system for facial motion analysis: ten years experiences and recent developments. *Annals of plastic surgery*, 69:173–185, 2012.
- [20] M.J.A. van Alphen. Development of an oral tongue model. Master’s thesis, University of Twente, 2011.
- [21] H.T. Nguyen; M. Worring; R. van den Boomgaard. Occlusion robust adaptive template tracking. *Proceedings - IEEE International Conference on Computer Vision*, 1:678–683, 2001.
- [22] A. Kreeft; I.B. Tan; M.W.M. van den Brekel; F.J. Hilgers; A.J.M. Balm. The surgical dilemma of ‘functional inoperability’ in oral and oropharyngeal cancer: current consensus on operability with regard to functional results. *Clinical Otolaryngology*, 34:140–146, 2009.
- [23] F. van der Heijden. Camera calibration using cubes. Nonpublished document.
- [24] F. van der Heijden; P.P.L. Regtien. Id number recognition in pork industry - a feasibility study. 2005. Nonpublished.
- [25] F. van der Heijden; R.P.W. Duin; D. de Ridder; D.M.J. Tax. *Classification, Parameter Estimation and State Estimation*. John Wiley & Sons, Ltd, 2004.
- [26] H.M. Karara Y.I. Abdel-Aziz. Direct linear transformation into object space coordinates in close-range photogrammetry. *Proc. symposium on close-range photogrammetry*, pages 1–181, 1971.
- [27] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, pages 666–673, 1999.