Technology
Arts Sciences
TH Köln

Masterthesis Medientechnolgie

mit einem eventuell
ganz langen Untertitel

vorgelegt von

Oliver Kalbfleisch

vorgelegt an der

TECHNISCHEN HOCHSCHULE KÖLN
CAMPUS DEUTZ
FAKULTÄT FÜR INFORMATIONS-,
MEDIEN- UND ELEKTROTECHNIK

im Studiengang
MEDIENTECHNOLOGIE

Erster Prüfer: Prof. Dr. Peter Silie
                 Technische Hochschule Köln

Zweiter Prüfer: Prof. Dr. Maria Musterprof
                 Technische Hochschule Köln

Köln, im August 2017

Adressen: Max Mustermann
Musterstraße 1
12345 Musterstadt
max@mustermann.de

Prof. Dr. Peter Silie
Technische Hochschule Köln
Institut für Informatik
Steinmüllerallee 1
51643 Gummersbach
peter.silie@th-koeln.de

Prof. Dr. Maria Musterprof
Technische Hochschule Köln
Institut für Informatik
Steinmüllerallee 1
51643 Gummersbach
maria.musterprof@th-koeln.de

Contents

Kurzfassung

Hier folgt die Kurzfassung.

Abstract

Hier folgt die Kurzfassung auf Englisch.

1 Einleitung

The standard interface between human and computer has for long years been mouse and keyboard. But with the advance of technology, new interfaceing methods were developed in the last few years.

Touch technology for interfacing with mobile devices and desktop computers has become a reliable technology and has been integrated into our everyday lives. Advances in capabilites of CPU as well as GPU hardware has build a foundation for the use of advanced AR and VR Technology. 3D and Stereoscopic rendering can now be accomplished even by mobile hardware (with some limitations)without the need of specifi Hardware. For An intensly immersive experience VR Googles are used to explore digitally created worlds.

But with this level of immerivenes, a touch device or even a mouse and keyboard setuop is rather hindering the user experience. First attempts of solving this problem came with the introduction of tracked controllers for the interaction with the digital world, but these can also only supply a fration of the capabilities of our natural interaction devices, namely the human hands.

2 Description of hand in digital space

Tracking of the human hand has always been a challenging Problem. In comaprison to other larger bodyparts like the Arm or the head, the human hand itself contains a large variety of smaller parts, namely bones and muscles. These components have to be taken into acount when trying to replicate the natural motion of the hand in digital space.

2.1 Physiological structure of the human hand

Lee and Kuni [19] describe the human hand as "an articulated structure with about 30 degrees of freedom [which] changes shape in various ways by its joint movemnents."
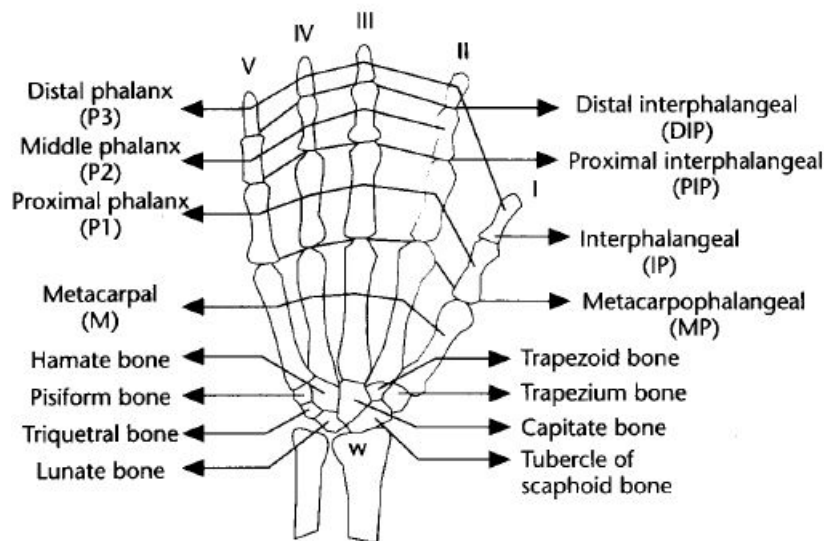

Figure 2.1: Bone structure of the human left hand ([19])

All of the hand components are connected to at least one neighboring component via a joint. Teh joints affect the position of the connected components. To describe the movement of the hand components, we can use the roation angles of the joints to correlate to a specific position.

To do so, we define a local coordinate system for each of the exiting hand joints. By doing so, we achieve a sequence of rotaions in the local coordinate systems of the joints. Such a sequence can then be used to describe a specific movement and/or position of

a component. Not all of the joints in the human hand have equal degrees of freedom. Their functionality can be classified in the amount of DOFs (Degrees of freedom)[16]

- 1 DOF
  - A joint movement that can perfom a **flexion** or **twist** in one direction

- 2 DOF
  - A joint movement that can perform **flexion** in more than one direction (**directive**)

- 3 DOF
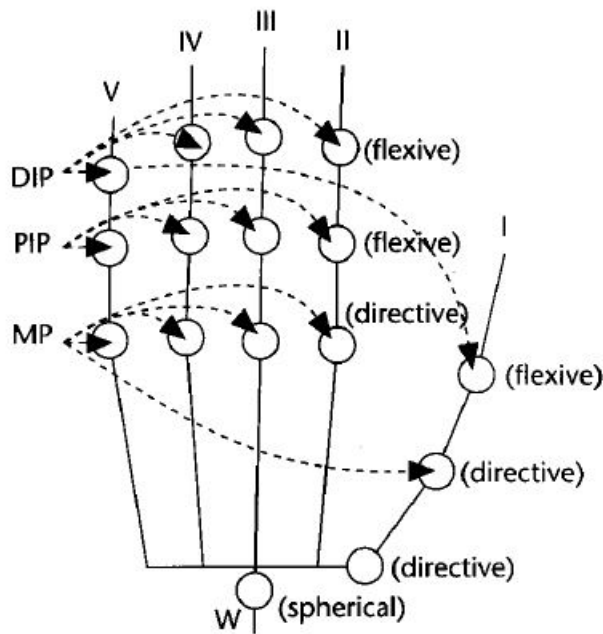 - A joint movement that permits simultaneous **directive** and **twist** movements.(**spherical**)

Figure 2.2: Representation of the DOFs of the human hand

When looking at the DOFs diplayed in Figure 2.2, each finger (II-V) sums up to 4 DOFs and the thumb to 5 DOFs. Also considering 6 DOFs for the rotation and position of the whole and itsself, the result gets us to 27 DOFs for the human hand.

2.1.1 Constraints in Hand Motion

A full usage of all the declared DOFs would lead to al large amount of possible combination. Since the hand is not only made up of bones but also Muscles and the skin, we can impose some constraints([3]) to the movement of the joints. Ling, Wu and Huang([20]) propsed following classification for the constrainsts:

- **Type I constraints**

  -A constraint that limits the range of finger motions based on hand anatomy

- **Type II constraints**

  - A constraint that the position of the joints during finger movement

- **Type III constraints**

  -A constraint that limits position based on natural hand motions

The **Type I** and **Type II** constarints rely on the physiological and mechanical properties of the humand. **Type III** constraints are results of common and natural movements and can be differing form person to person. As these movements are to some degree simular for everyone, a broad grouping can be applied. The curling of the fingers at the sane time when forming a fist is way more natural than curling each finger by itsself. Here the motion of the hand is quite simular between different persons, but the constraints cannot be described in a mathematical form.

A **Type I** constraint example would be that the position of the fingertip is kimited by the length of the other finger segments and therby can only reach as far as the combined length.

An example for **Type II** constraints would be that, for your fingertip to touch your hand palm, all joints in the finger have to be bend to achieve this position. The following inequalities can be used to describe these constraints:

Type I:

$$0° \leq \Theta_{MP\_flex} \leq 90°$$
$$0° \leq \Theta_{PIP\_flex} \leq 110°$$
$$0° \leq \Theta_{DIP\_flex} \leq 90°$$
$$-15° \leq \Theta_{MP\_abduct/adduct} \leq 15°$$

(2.1)

A further constraint that is specific to the middle finger is, that this fingers MP normally does not abduct and adduct much. Therefore we can infer an approximation and thereby remove 1DOF from the model:

$$\Theta_{MP\_abduct/adduct} = 0°$$

(2.2)

The same behavior can be seen in the combination of hand parts labeled W(the connection point between hand and lower arm). This approximation allso eliminates one DOF on the connected thumb:

$$\Theta_{W\_abduct/adduct} = 0°$$

(2.3)

Since the DIP,PIP and MP jonts of our index, middle, ring, and little fingers only have 1DOF for flexion, we can further asume that their motion is limited to movement in one plane.

Type II:

The **Type II** constraints can be split into interfinger and intrafinger constraints. Regarding intrafinger constraints between the joints of the same finger, human hand anatomy implies that to bend the DIP joints on either the index, middle, ring or little fingers,the corresponding PIP joints of thath finger must also be bent. The approximation for this relation[28] can be described as :

$$\Theta_{DIP} = \frac{2}{3}\Theta_{PIP} \tag{2.4}$$

Interfinger constraints can be imposed between joints of adjacent fingers. Interfinger constraints describe that the bending of an MP joint in the index finger forces the MP joint in the middle finger to bend as well.

When combinig the constraints described in the above equations, the starting number 21 DOF's of the human hand can be reduced to 15. Inequalities for these cases, obtained through empiric studies, can be found in [19].

2.2 Kinematics

The preceeding sections gave an overview of how we can describe a model of the human hand and introduced some limiting constraints. With the model and the constraints, we can now start to build a kinematic system for the animation of the model.

Kinematic systems contain so called *kinematic chains*, which consist of a *starting point*or *root*, kinematic elements like *joints*, *links* and an *endpoint*, also called *end effector*. Applied to the human hand, the whole hand model represents the kinematic system. This system contains several *kinematic chains*, namely the fingers of the hand with the fingertips beeing the *end effectors* of each of these chains.

As we begin to move our hands, the states of the kinemtic chains begin to change. Joint angles and end effector positions are modified until the end position is reached. To represent the new position and agnle dataset of our physical hand with our kinematic system, two major paths for achieving a solution can be taken.

2.2.1 Forward Kinematics

Forward Kinmatics (FK) uses the knowledge of the new angles and positions after the application of known transformations to the kinematic chain. The data of the *joints* and *links* between the *root* and the *end effector* is then used to solve the problem of finding the *end effector's* position.

We can denote the exisiting end effectors relative postion to an origin as $s_1, ..., s_k$. The s_i positon is the result of th combination of all the joint angles in the corresponding kinemtaic chain. Respectively, we define the target positon of the end effectors as $t_1, ..., t_i$,

with t_i beeing the target positiojn for the end effector s_i. The required positional change for the end effector can now be described as $e_i = t_i - s_i$. In systems with more than one end effector, like our hand system, the components can be written as vectors.

$$\vec{\mathbf{s}} = (\mathbf{s}_1, ..., \mathbf{s}_n)^T$$
$$\vec{\mathbf{t}} = (\mathbf{t}_1, ..., \mathbf{t}_n)^T \tag{2.5}$$
$$\vec{\mathbf{e}} = \vec{\mathbf{t}} - \vec{\mathbf{s}}$$

As the vector components of $\vec{\mathbf{s}}$ are reults of the chain joint angles $\theta_1, ..., \theta_n$ and therefore are effected by them, we define

$$\vec{\mathbf{s}_i} = f_i(\boldsymbol{\theta})$$
$$\vec{\mathbf{s}} = f(\boldsymbol{\theta}) \tag{2.6}$$

With $\boldsymbol{\theta}$ beeing the column vector $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)^T$. The second vector equation displayed in (2.6) is also callled the *Forward Kinematics*(FK) solution.

The advantage of an FK solution is that there is always an unique solution to the problem. In consequence, this approach is commonly used in the field of robotics, where the information on the chain elements is easily available.

The tracking of the human hand and all if its chain components is rather complicated. Therefore a solution which takes a known position of the *end effektor* and calculates the parameters for the rest of the cain would be more desireable.

2.2.2 Inverse Kinematics

> Inverse Kinematics (IK) is a method for computing the posture via estimating each individual degree of freedom in order to satisfy a given task [1]

The concept of *Inverse Kinematics* (IK) already describes it's principle in it's name. It takes the reversed approach in comparison to the FK principle in chapter 2.2.1. Instead of knowing the states of the chain elements and calculating the resulting position of the *end effector*, we take the position of the *end effector* and try to retrieve the possible states of the other chain elements.

$$\boldsymbol{\theta} = f^{-1}(\vec{\mathbf{s}_d}) \tag{2.7}$$

The result of this equation is the vector $\boldsymbol{\theta}$ for which the values of $\vec{\mathbf{s}}$ coincide to the desired configuration $\vec{\mathbf{s}_d}$. In the case of an optimal result, this configuration would have the same position values as the target positions.

The main problem with this method occurs in the calulaton of the f^{-1} function, due to it beeing a highly non linear operator which is not easily invertible. The approaches that

[1] describes to counter this problem will be displayed later on in this chapter.

In contrary to having a unique solution with the FK approach, the IK approach can end at the point of not finding a suitable solution. Figure 2.3 diplays three possible outcomes for the IK approach.
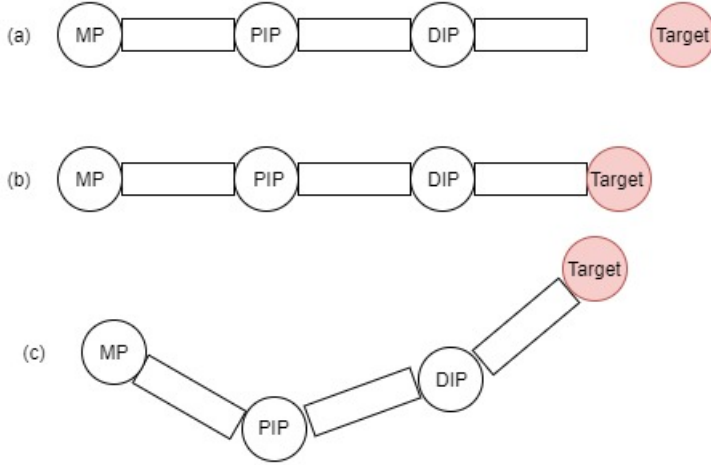

Figure 2.3: Possible solution for an IK problem of a human finger:

(a) The given target position of the end effector can not be reached. (b) The given target can only be reached by one solution.(c) The target position can be reached with multiple different solutions.

2.2.3 Jacobian Inverse

One common approach to solve the IK problem is the utilization of a Jacobian Matrix and an iterative calulaton process. This matrix contains the partial derivates of the chain systems relative to the end effector **s**. When using the Jacobian, a linear approximation of the IK problem will be applied for solving. The approximation's components model the end effector's realtive movement to changes in transitions of the systems link translations and joint angles. Therefore, the resulting function is dependant on the joint angles $\boldsymbol{\theta}$ values and can be defined as

$$J(\boldsymbol{\theta})_{ij} = \left(\frac{\partial \mathbf{s}_i}{\partial \theta_j}\right)_{ij} \tag{2.8}$$

with $i=1,...,k$ and $j=1,...,n$. Further readings on methods for the calculation of Jacobian matrices can be found in [26]. Based on definition (2.8), an entry for the j-th rotational joint would be calculated as follows:

$$\frac{\partial \mathbf{s}_i}{\partial \theta_j} = \mathbf{v}_j \times (\mathbf{s}_i - \mathbf{p}_j) \tag{2.9}$$

where $\mathbf{p}_j$ is the position of the joint, and $\mathbf{v}_j$ is the unit vector pointing along the current axis of rotation for the joint. Taking the derivate of definition (2.6) with respect to time gives the basicequation for forward kinematics that describes the velocities of the end effectors:

$$\dot{\vec{s}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{2.10}$$

Now having all the values for the anlges, the *end effector* position and the target positions, we can compute the resulting Jacobian matrix. Thereafter we seek an update value $\Delta\boldsymbol{\theta}$ for incementing the current joint values:

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{curr} + \Delta\boldsymbol{\theta} \tag{2.11}$$

The idea here is that the chosen value for $\Delta\boldsymbol{\theta}$ should lead to the resulting $\Delta\vec{s}$ beeing approximately equal to $\vec{e}$ from (2.5). The $\vec{e}$ can be approximated by:

$$\Delta\vec{s} \approx \boldsymbol{J}(\boldsymbol{\theta_{curr}})\Delta\boldsymbol{\theta} \tag{2.12}$$

Using this approximation we can reformulate the FK problem as $\vec{e} = J\Delta\boldsymbol{\theta}$ and therefore our inverse kinematics problem form 2.7 can be expressed as $\Delta\boldsymbol{\theta} = J^{-1}\vec{e}$.
The problem we run into with this solution is the construction of the inverse Jacobian matrix. The Jacobian J may not be square or invertible. In the case of it beeing invertible , the result may only work inferiorly because of it beeing nearly singular. Beeing singular means that no change in joint angle values may achieve the desired end effector position as an outcome.

Jacobian Transpose

One approach to calculating the value of $\Delta\boldsymbol{\theta}$ without having to calcualte the inverse of **J** is done by relpacing the inverse with the transpose of **J**.

$$\Delta\boldsymbol{\theta} = \alpha\boldsymbol{J}^T\vec{e} \tag{2.13}$$

Of course the transpose and the inverse of **J** are not the same thing. When using the theorems displayed in [36**?**] we can show that:

$$\langle JJ^T\vec{e}, \vec{e}\rangle = \langle J^T\vec{e}, J^T\vec{e}\rangle = \|J^T\vec{e}\| \geq 0 \tag{2.14}$$

Under a sufficiently small $\alpha > 0$ the updated angles from 2.11 will change the end efector positions by approximately $\alpha JJ^T\vec{e}$. They also state that the value of α can be calculated by minimising the new value of the error vector $\vec{e}$ after each update.

$$\alpha = \frac{\langle\vec{e}, JJ^T\vec{e}\rangle}{\langle JJ^T\vec{e}, JJ^T\vec{e}\rangle} \tag{2.15}$$

Jacobian Pseudo Inverse

Instead of calculating the normal inverse of the Jacoboian, which can lead to the problems described before, we can use the so called *pseudo-inverse*[7] for the calculation. The *pseudoinverse* is defined for all matrices J, even ones which are not square or not of full row rank.

$$\Delta\boldsymbol{\theta} = J^\dagger\vec{e} \tag{2.16}$$

The *pseudoinverse* represents the best possible solution for $J\Delta\boldsymbol{\theta} = \vec{e}$ in respect to least squares, but does suffer from instability near singularities. It has the property that the matrix $(I - J^\dagger J)$ performs a projection onto the nullspace of **J** . Therefore, for all vectors φ, $J(I - J^\dagger J)\varphi = 0$. Therefore $\Delta\boldsymbol{\theta}$ can be set by

$$\Delta\boldsymbol{\theta} = J^\dagger\vec{e} + (I - J^\dagger J)\varphi \tag{2.17}$$

for any vector φ and still obtain a value for $\Delta\theta$ which minimises the value $J\Delta\boldsymbol{\theta} - \vec{e}$. The pseudoinverse of **J** can be derived as follows:

$$\boldsymbol{J}^\dagger = (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T$$
$$\Delta\boldsymbol{\theta} = (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T\vec{e} \tag{2.18}$$
$$(\boldsymbol{J}^T\boldsymbol{J})\Delta\boldsymbol{\theta} = \boldsymbol{J}^T\vec{e}$$

Using $\vec{y} = \boldsymbol{J}^T\vec{e}$, we can now solve the equation $(\boldsymbol{J}^T\boldsymbol{J})\Delta\boldsymbol{\theta} = \vec{y}$

Singular Value Decomposition

The usage of a *singular value decomposition* is another method of calculating the *pseudoinverse* of a jacobian matrix. A singular value decomposition of **J** consists of expressing **J** in the form

$$\boldsymbol{J} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V^T} \tag{2.19}$$

For an $m \times n$ Jacobian matrix, **U** is $m \times m$ and the columns are the eigenvectors of JJ^T. **D** is $m \times n$, and the entries are zero except along the diagonal where $d_{ii} = \sigma_i$ with $\boldsymbol{\sigma_i}$ beeing the singular value and $\sigma1 \geq \sigma2 \geq ... \geq \sigma m \geq 0$. Singular values are the nonnegative square roots of the eigenvalues of JJ^T and $J^T J$.**V** is $n \times n$ and the columns are the eigenvectors of $J^J J$. The construction of the pseudonverse is done as follows:

$$\boldsymbol{J}^\dagger = \boldsymbol{V}\boldsymbol{D^\dagger}\boldsymbol{U^T} \tag{2.20}$$

The pseudo-inverse of the diagonal matrix **D** is obtained by replacing each positive diagonal entry with it's reciprocal [12].

2.2.4 Damped Least Squares

The *Dampened Least Squares* method is is applied for inverse kinematics as it avoids the singularity problems that can occur when using the pseudoinverse.It further provides a numerical stable method for selecting $\Delta\theta$. First occurences for inverse kinematics can be found in [24, 33]. Instead of finding the minimum for $\Delta\theta$ as a best seolution for the equation displayed in 2.5, the goal of this methodd is to find the value that minimizes the following quantity with $\lambda \in \mathbb{R}$ as a dampening factor.

$$\| J\Delta\boldsymbol{\theta} - \vec{e} \|^2 + \lambda^2 \| \Delta\boldsymbol{\theta} \|^2 \tag{2.21}$$

which can be rewritten to the corresponding

$$(\boldsymbol{J}^T\boldsymbol{J} + \lambda^2 I)\Delta\boldsymbol{\theta} = \boldsymbol{J}^T\vec{e} \tag{2.22}$$

and therefore

$$\Delta\boldsymbol{\theta} = (\boldsymbol{J}^T\boldsymbol{J} + \lambda^2 I)^{-1}\boldsymbol{J}^T\vec{e} \tag{2.23}$$

The selection of a correct λ value is essential for this method, small values of λ give accurate solutions but low robustness to the occurrence of singular and near-singular configurations, while large values of λ result in low tracking accuracy even when a feasible and accurate solution would be possible[1, 5].

2.2.5 Newton Methods

The Newton family of methods is based on a second order Taylor series expansion of the object function f(x):

$$f(x + \sigma) = f(x) + [\triangledown f(x)]^T\sigma + \frac{1}{2}\sigma^T H_f(x)\sigma \tag{2.24}$$

where Hf(x) is the Hessian matrix. The downside of this approach is that the calculation of the Hessian matrix bares high computational cost for each iteration as a result of its complexity. Like the approximation of the Jacobian inverse, several attemps to lower the computational cost have benn made by utilizing an approximation of the Hessian matrix based on a function gradient value.

Since the Newton methods are posed as a minimisation problem, they return smooth motion without erratic discontinuities. The Newton methods also have the advantage that they do not suffer from singularity problems, such as that which occurs when finding the Jacobian Inverse.

2.2.6 Sequential Monte Carlo Method

Sequential Monte Carlo Methods (SMCM) Methods are a relative young approch in solving IK problems. Courty and Arnaud [6] proposed such a technique, whose main advantage is the usage of a sampling approach. When doing so,the IK problem can be solved through the usage of direct kinematics, avoiding the numerical inversion of the forward operator.

2.2.7 FABRIK Algorithm

A more recent approach in the kinematics field is the **FABRIK** algorithm proposed by Aristidou and Lasenby[2].As shown in section 2.2.2, these approaches all depend on computational intensive matrix operation like calculating an inverse and may have problems with matrix singularity.

The **FABRIK** algorithm does not depend on these matrix operations as is solves for the position of a point on a line to retrieve the new joint positions. This is done in an forward and also inverse solving approach, iterating these steps until the calculated position converges towards the target position from the tracking data.

Figure 2.4 illustrates the steps that are contained in one iteration step. The chain joints are denoted as $\mathbf{p}_i$ with the distance $\mathbf{d}_i$ being $|\mathbf{p}_{i+1} - \mathbf{p}_i|$. The target point for the end effector is denoted as $\mathbf{t}$. Step **(a)** displays the starting pont of the iteration. The joint positions are taken from either a previous iteration or from an initial calibration.

But before calculations can begin, the algorithm has to check whether the intended target point $\mathbf{t}$ is reachable for the end effector. This is done by measuring the distance between the root of the kinamtic chain and the target point $\mathbf{t}$. This value is then compared with the sum of the distances $\mathbf{d}_i$.
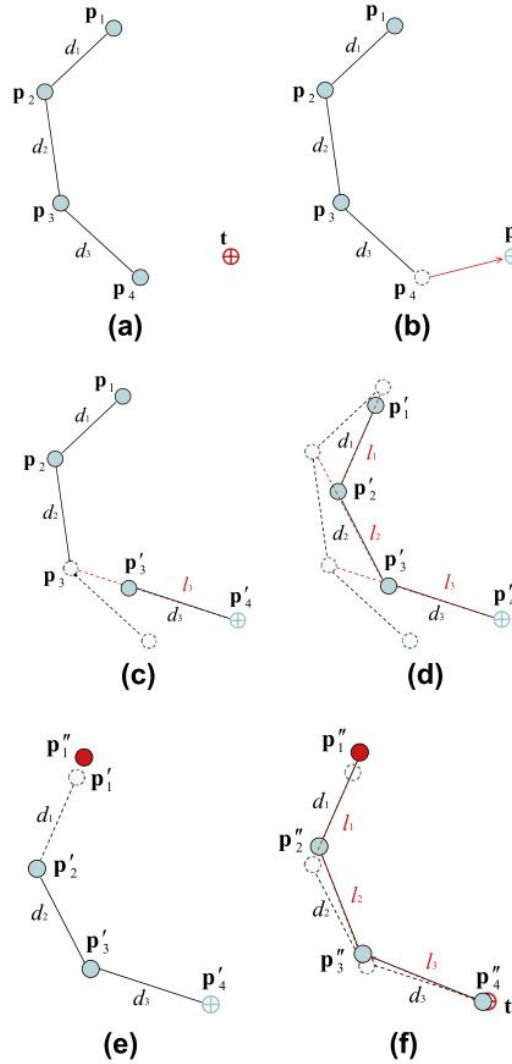

Figure 2.4: Forward and backward calculation steps for one iteration of the **FAB-RIK**algorithm.(a) initial position of the system.(b)End effector is moved to target position.(c) Deterimne position of next joint on constructed line.(d)repeat until root is reached. (e)move root joint to initial position. (f) repeat calculation in reverse direction [2]

$$dist_{t,d_1} < \sum_{k=1}^{i} d_i \qquad (2.25)$$

The inverse calculation step is the first step, which is diplayed in **(b)** and **(c)**.The calculation is started at the end effector, moving to the root of the chain. If the summed distance is greater, then the **t** is within the reach of the system and the calculation can continue, otherwise the calculation has to be aborted and the error has to be manged otherwise. Assuming this requirement to be met, we can now begin with the first calculation. Therefore we assume that the new position $\mathbf{p}'_n$ with n=4,...,1 is equal to **t**.

$$\mathbf{p}'_n = \mathbf{t} \tag{2.26}$$

From this new point, we can construct a line that goes through $\mathbf{p}'_n$ and $\mathbf{p}_{n-1}$.

$$
\begin{aligned}
A &= \mathbf{p}'_n \\
B &= \mathbf{p}_{n-1} \\
\mathbf{l}_{n-1} &= \overline{AB}
\end{aligned}
\tag{2.27}
$$

The resultng position of the new $\mathbf{p}'_{n-1}$ point is located in this line with the distance of $\mathbf{d}_{n-1}$ from $\mathbf{p}'_n$ (see **(c)**).

$$\mathbf{p}'_{n-1} = \mathbf{p}'_n + \left(\frac{\overline{AB}}{|\overline{AB}|} \cdot \mathbf{d}_{n-1} \right) \tag{2.28}$$

Consecutively, this is done with the remaining joints until the root joint is reached(see **(d)**). This finishes the first halv of the iteration step.With the calculated positions, we now perform a forard calculation, starting from the root until we reach the end effector. Since the root of the system normally does not move from it's initial position, we have to reset the root jont to this value before starting to calculate the new positions of the subsequent joints(see**(e)**).

Analogous to the procedure in the inverse step, we construct the lines between the points and determine the new positon values of the joints. The end result of this step is shown in **(f)**. At this point, we can decide if the result position of the end effector is appropriate in comparison to the value of **t**. A simple threshold value for this case could be the position difference between these two points.

2.3 Digital hand models

2.3.1 1

3 Tracking in real space

The previous sections provided information on the structure of the human hand and how to represent it in the digital space. To apply the algorithms presented in Section 2.2 to our digital hand model, we need input data from our real world representative.

3.1 General tracking technologies

The methods for gaining positional data can be roughly categorized into two major groups, glove based methods and vision based methods. Glove based methods have already been in developement since the 1980's [4] and have since then resulted in several solution attempts. Sturman and Zeltzer gave a survey on the exsisting tracking mehtods in their paper [?].

They distinguish between two areas of tracking, first the 3D positional tracking of the hand (and also other bodyparts) without regard to the hands shape and secondly the tracking of the hand shape with glove technologies.

These tracking technologies presented are still applied today in modern tracking solutions [29, 35]. They account for solutions based in optical tracking based on marker detection, magnetic detection via measurements of an artificial megnetic field[27] and accoustic measurements via triangulaton of ultrasonic pings.

3.1.1 Optical tracking

The components for an optical tracking systems are several cameras for object detection and some kind of tracking characteristic of the object to be tracked. These characteristics can be either artificially applied ones like active flashing infrared LED on key tracking positions of the body or infrared reflective markers.

A series of cameras positioned around the tracking subject will then track these markers inside their visual fields. The second method uses a single camera to capture the silhouette image of the subject, which is analyzed to determine positions of the various parts of the body and user gestures.

The image data is supplied to special software which correlates the marker positions in the multiple viewpoints and uses the different lens perspectives to calculate a 3D coordinate for each marker. These image interpretation and correlation tasks require computationally costly operations. The marker tracking is also prone to errors through variation in lighting of the scene, material reflection properties and also marker occlusion as the trackers are moved. Also most of the systems rely on several tracking cameras for

a complete coverage of the tracking space. This leads to a higher system complexity in terms of setup and calibration.
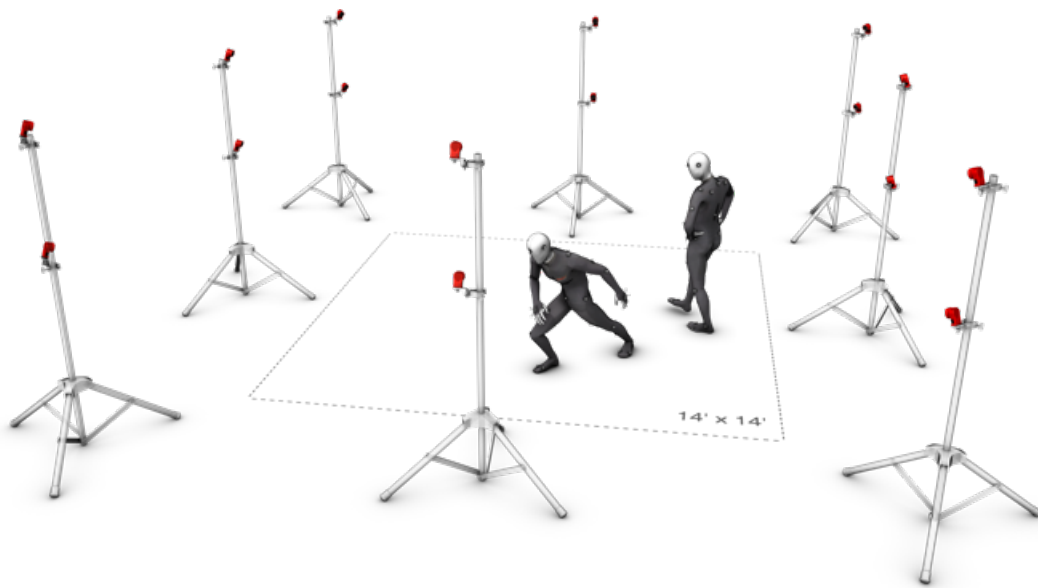

Figure 3.1: Example for an optical tracking system with passive infrared reflector markers [25]

3.1.2 Magnetic tracking

The usage of a magnetic field for position tracking is a relatively uncomlicated technique. The earth already provides us with a magnetic field to orient on. Simular to the optical trackers, magnet field sensors can be placed at key tracking positions. These sensors measure field strength in 3 orthogonally orented axes to produce a 3D vector of the unit's orientation with respect to the excitation. As the earths magnetic field prone to changes based on geographical location, data corrections have to be applied to the measurements.

Another solution is to generate an local magnetic field via a multicoil source unit. The source unit coils are energized in sequence and the corresponding magnetic field vector is measured in the sensor unit. With three such excitations, you can estimate the position and orientation of the sensor unit with respect to the source unit.

the downside of this technology is that ferromagnetic and conductive material in the surrounding environement will have an effect on the generated magnetic field. These distortion effects will form small unwanted secondary source units through the induction of small Foucault currents by the main sources magnetic field. Magnetic fields have the property of having an inverse cubic falloff as a function of distance from the source, which limits the rango of operation for the system. Position resolution in the radial direction from source to sensor depends on the gradient of the magnetic field strength, and thus the positional jitter grows as the fourth power of the separation distance.

In comparison to other tracking technologies, the magnetic field tracking solution has the conveniece of not suffering from line of sight problems from tracker occlusion. The

magnetic fields are capable of passing through the human body. Also the sensor size for measuring magnetic fields is rather small, giving the trackers a small volume. Furthermore, only one source unit is needed for the tracking of multiple sensor units.

3.1.3 Accoustic tracking

The principles of accoustic tracking are very simular to those of the optic tracking technologies. Instead of using lightwaves, the systems utilize acoustic pulses of ultrasonic wavelenghts to time the time of flight between emitter and sensor for range measurement. To get a good measuring result from the systems, the used accoustic transducerst have to be as omnidirectional as possible,so that the signal can be detected no matter how the emitter is positioned or oriented in the tracking volume. For the speakers to acchieve a wide beam width, their size has to be small.

To be able to build the microphones into the tracker, they can only have active surfaces a few millimeters in diameter. This leads to a reduction in range as the efficiency of accoustic transducers is proportial to the active surface area. Also accoustic systems can have problems with ambient noises occluding the signal. This becomes even mor critical when using such a system outdoors.

Soundwaves travel at a much slower speed than lightwaves which brings benefits and downsides with it. Soundwaves can be reflected from objects, producing so echos which arrive at the recieving sensor at a later point in time. Here the slower speed can be beneficial as we can await the first sound occurence to arrive at the sensor and filter out all later reflections from the data. The reflections of the previous pluse also have to be subsided before a new measurement can be made, lowering the uodate rates of the system. The air the soundwaves travel through is also a limiting factor as humidity, airpressure and air currents can influence the travelling soundwaves.In comparison to the optical systems, the accoustic systems are not as prone to occlusion errors since soundwaves have a better ability to bend around obstacles than lightwaves.

Most of these downside can be adressed with a combination of these systems with another form of tracking like in [10].

3.2 Hand tracking Systems

In comparison to the tracking of the whole human body, the tracking of the human hands with their maximum of 27 DoF's each in a small space is rather difficult to handle.

The systems that try to achieve this have to encounter deveral difficulties. Self Occlusion also plays a great role in the Systems design, especiall when working with only one tracking optical system for reduced complexity. Also these systems eed to maintain a certain amount of processing speed for achieving a fluent reproduction of the captured motion. This demands the capability of processing large amaounts of data in very short intervals. Most prototpy testing for the described systems will be done in an controlled

environemtn where the background is known. For a more widespread use , the systems need to becapable of registering the hand on an unrestricted background, which may have a wide range of patterns, xolor and lighting differences. Also human hand motion itsself is quite fast, resulting in a higher framerate demand for the tracking cameras.

The following section will describe different approaches to soleve the aforementioned difficulties.

3.2.1 Sensor based approach

Tracking systems thath make use of sensors mounted to the hand via straps or a glove have alread been in use since the 1970's. First prototype glove systems included the Sayre Glove[32] and the Digital Entry Data Glove[13].

The Sayre glove is based off an idea of Richard Sayre. The system utilizes a light sourec and a photocekk which are connected via a flexible tube. These components are mounted along each finger of the hand. The light that passes through the tube is infulenced by the bending angle of the tube which corresponds to the finger bending. a large bending angle of the finger induces a larger bending angle of the tube, reducing the intensity of the light that reaches the photodiode. The resulting voltage in the photocell can then be mapped to a specific bending angle of the finger.

The Digital Entry Data Glove was patented by Gary Grimes in 1983. Instead of using only one type of sensors for measurement, this glove based systems used several sensor types for different measurements. Touch or proximity sensors were used for determining whether the user's thumb was touching another part of the hand or fingers. To measure the flexion of the joints in the thumb, index, and little finger four "knuckle-bend sensors" were used.

To get measurements for the tilting of the hand in respect to the horizontal plane two tilt sensors were mounted. Finally two inertial sensors for measuring the twisting of the forearm and the flexing of the wrist were utilzed.This glove was intended for creating "alphanumeric characters" from hand positions. Recognition of hand signs was performed via a hard-wired circuitry, which mapped 80 unique combinations of sensor readings to a subset of the 96 printable ASCII characters.

These early systems only provided a limited amount of sensors and were rather unpractical in usse. As they were

Figure 3.2: Acceleglove system with the described sensor positions [14]

developed to serve very specific applications, they were used only briefly, and never commercialized.

Newer developements in this sector include [14, 17, 21]. The *AcceleGlove* presented by [14] consistsis of six dual-axis accelerometers, mounted on the fingers and the back of the palm, reporting position with respect to the gravitational vector. Sensors are placed

on the back of the middle phalanges, on the back of the distal phalange of the thumb, and on the back of the palm.

Kuroda et al [17] introduced the *StrinGlov*, which obtains full degrees of freedom of human hand using 24 Inductcoders and 9 contact sensors, and encodes hand postures into posture codes on its own DSP. The bending angles of the fingers are measured with the inductcoders. These sensors realte the finger movement to a change in magnetic flux induced by the movement of the sensor parts thath are attached to the finger. The sensor functionality is simular to the functionality of the light sensors from [32]. The 9 contact sensors, also based on magnetic fields, are put on the fingertips and on the inside of the hand to be able to measure contact between two fingetips or the fingertips and the hand. The system furthermore benefits from simple structure, which leads to low production cost. Majeau et al [21] propsed a systems that uses optical flexion sensors to determine the bending angles of the fingers. The opticcal flexion sensors consist of a LED, a Photodetector and an optical fibre. The LED emits light which travels through the optical fibres and the intensity that results as the output from the fibre end is measured by the Photosensor. This measuring principle also corresponds to the principle of [32]. Furthermore the system is also capable of measuring the abduction of two finger. Here the LED is mounted to one finger and the detector on the other. The optical fiber is run from the LED down the finger to the knucklebase and then up to the detector. When abducting the two fingers, the angle at the knucklebase changes, resulting in an intensity change at the detector through the loss of intensity at the curvature of the optic fibre. The cange in intensity of both measurting techniques can be mapped to corresponding hnd movements.

Further readings on sensor based hand tracking systems can be found in [8, 31]

3.2.2 Optical marker approach

Optical approaches use properties of the human hand to hand to estimate the current position. The evaluation of the hand pose is usually split up into two steps. The first one is the registration of the "real" hand.

This is mostly done by at least one camera which is aimed at the hand, supplying a "continous" data stream to the evaluation Software. In the secinbd step, the evaluation software then tries to find key spots in the provided images. This data is then used for the search of the mathcing digital pose. The definition of these key spots can be achieved by several techniques. An example is the usage of some kind of marking for finger segments as displayed in

Figure 3.3: Color glove setup used by Wang and Popovic [34]

[9, 11, 34]. In the method described by Wang and Popovic [34],a specially colored glove is used. The glove is segemnted into ten segments, colored randomly from a pool of ten distinct colors (see Figure 3.4).

The color pool was limited by the color distinguishing capabilities of the camera used in the experiment. The setup used was engineered to fit the standard home user capabilities, so when using a higher grade camera more color variation could be possible. They used more large pattern areas on the glove to minder the risk of occlusion problems which could appear with smaller patch areas. The pattern is createdby selecting twenty seed triangles on a 3-D hand mod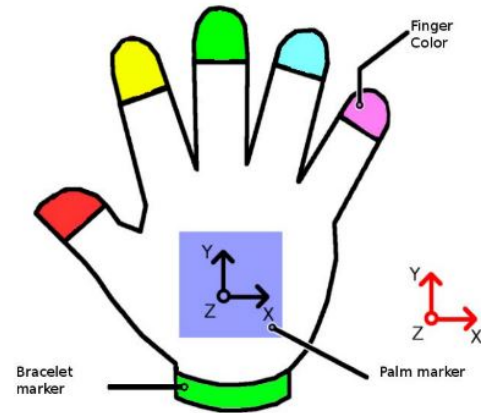el that are maximally distant from each other. The remaining triangles are assigned into patches based on their distance to each of these seeds. Each patch is as-

Figure 3.4: Color glove setup used by Fredriksson and Ryen [11]

signed one of the ten colors at random. The jagged boundaries between the patches are artifacts from the low triangle count of the used hand model. Algortihms that use other characteristics for tracking like texture or shading [18] rely on an acurate pose from a previous frame to constrain the posture search in the current frame. When using bare hand pose estimation two different handposes can map to similar images. This can lead to an inacurate pose estimation and the breakdown of the algorithm if the wrong estimation is made. The benefit of the color glove is that the unique patch pattern, hand poses always map to ditinguishable images, simplifying the search process. Therefore it can effectively recover itsself at each frame without the need of a previous frame. With the colored glove as a tracking feature, the images from the camera are prepared for a database sampling step. The database that was used consisted of 18,000 finger configurations. A distance metric between two configurations was defined as the root mean square (RMS) error between the vertex positions of the corresponding skinned hand models. With this distance metric, a low dispersion low-dispersion sampling was used to draw a uniform set of samples from the overcomplete collection of finger configurations. The selected configurations are rendered at various 3-D orientations using a (synthetic) hand model at a fixed position from the (virtual) camera. The camera image was compared then compared to the databse iamges. The comaprison is done via a nearest neighbor lookup using a Hausdorff-like distance [15] and penalizing the distance to the closest pixel of the same color in the other image.The resulkting pose from the database can then be applied to the digital hand model. The method presented by Fredriksson and Ryen [11] also uses a color coded glove. In constrast to the fully colored glove described before, the used glove only has colored fingerstips, color markers for the palm and a colored band for the wrist.

The palm markers and the wrist band are used to retrieve the 3D position of the hand. The wrist marker is furthermore used to define the bondingbox of the the tracking area in a calibration process. Defining a bounding greatly simplifies further image analysis operations by taking away a large amount of uneeded image data. After determining position and orientation of the hand in 3d by utilizing wrist band an palm marker position, the system tracks the grip angle and the lateral tilt angle for each finger. The grip angle is defined as as the curling of the finger towards the palm.The second angle is defined as the lateral tilt angle which measures the spread of the fingers. The grip angle is calulated by comparing the density of finger pixels around an estimated knuckle line. from the density, an estimations for the finger's Y position around an origin position at the knuckle's base is made. This value is then transformed into an angle value using an inverse tangent operation.

One downside of this tracking method is that it does not yet incoorporate the movement of the thumb. The thumb has other movement possibilities and therefore needs a differnt algorithmic approach. Also the approach only focusses on the the tracking of the hand data and does not yet implement a digital counterpart like the previously described system.

3.2.3 Image analysis approach

Hand tracking with image anlysis and optical markers has the flaw that the markers have to be put onto the hand in some kind of fashion. Using a glove for this purpose imposes just a litte bit of inconvenience, but have the drawback of not fitting every handshape equalliy and also having to deal with hygene when switching between users. The most natural way of tracking the human hand would be by simply using the properties of the human hand like skin color and shap for tracking. This would remove the need for any artificaly added tracking features and have the greatest amount of comfort for the human user. A full overview of the technological advances in this area can be found in [22, 23]. This paper will only highlight some of the main topics displayed.

As already explained before, the human hand motion leads to a large variety of shapes with many self-occlusions, which becomes even more critical when trying to reproduce the human hand pose via Computer vision. Also the seperaton between background image information and the relevant hand information is a major topic, which is usually also the first step in the algorithmix process. An established method for the extraction of Background data was presented by Stauffer and Grimson [30]. It uses the Idea of representing each pixel value as a mixture of Gaussian functions (*MoG*) and updating these pixels during runtime with new guassian functions. The conbeniece of this method is that it is also able to substract background information from uncontrolled environments, making it possible for systems to work outside of lab conditions.

4 Fazit

List of Figures

List of Tables

Bibliography

[1] ANDREAS ARISTIDOU AND JOAN LASENBY: Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver: University of Cambridge, Department of Engineering. (2009)

[2] ARISTIDOU, Andreas ; LASENBY, Joan: FABRIK: A fast, iterative solver for the Inverse Kinematics problem. In: *Graphical Models* 73 (2011), Nr. 5, S. 243–260. `http://dx.doi.org/10.1016/j.gmod.2011.05.003`. – DOI 10.1016/j.gmod.2011.05.003. – ISSN 15240703

[3] BADLER, Norman ; MANOOCHEHRI, Kamran ; WALTERS, Graham: Articulated Figure Positioning by Multiple Constraints. In: *IEEE Computer Graphics and Applications* 7 (1987), Nr. 6, S. 28–38. `http://dx.doi.org/10.1109/MCG.1987.276894`. – DOI 10.1109/MCG.1987.276894. – ISSN 02721716

[4] BOLT, Richard A.: Put-that-there. In: THOMAS, James J. (Hrsg.): *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. New York, NY : ACM, 1980. – ISBN 0897910214, S. 262–270

[5] CHIAVERINI, S. ; SICILIANO, B. ; EGELAND, O.: Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. In: *IEEE Transactions on Control Systems Technology* 2 (1994), Nr. 2, S. 123–134. `http://dx.doi.org/10.1109/87.294335`. – DOI 10.1109/87.294335. – ISSN 10636536

[6] COURTY, Nicolas ; ARNAUD, Elise: Inverse Kinematics Using Sequential Monte Carlo Methods. In: HUTCHISON, David (Hrsg.) ; FISHER, Robert B. (Hrsg.) ; KANADE, Takeo (Hrsg.) ; KITTLER, Josef (Hrsg.) ; KLEINBERG, Jon M. (Hrsg.) ; MATTERN, Friedemann (Hrsg.) ; MITCHELL, John C. (Hrsg.) ; NAOR, Moni (Hrsg.) ; NIERSTRASZ, Oscar (Hrsg.) ; PANDU RANGAN, C. (Hrsg.) ; PERALES, Francisco J. (Hrsg.): *Articulated Motion and Deformable Objects*. Berlin and Heidelberg : Springer, 2008 (Lecture Notes in Computer Science). – ISBN 978–3–540–70517–8, S. 1–10

[7] DAHMEN, Wolfgang ; REUSKEN, Arnold: *Numerik für Ingenieure und Naturwissenschaftler*. Zweite, korrigierte Auflage. Berlin Heidelberg : Springer-Verlag Berlin Heidelberg, 2008 (Springer-Lehrbuch). `http://dx.doi.org/10.1007/978-3-540-76493-9`. – ISBN 9783540764939

[8] DIPIETRO, L. ; SABATINI, A. M. ; DARIO, P.: A Survey of Glove-Based Systems and Their Applications. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38 (2008), Nr. 4, S. 461–482. `http://dx.doi.org/10.1109/TSMCC.2008.923862`. – DOI 10.1109/TSMCC.2008.923862. – ISSN 1094–6977

[9] DUCA, Florin ; FREDRIKSSON, Jonas ; FJELD, Morten: Real-time 3d hand interaction: Single webcam low-cost approach. In: *Proceedings of the Workshop at the IEEE Virtual Reality 2007 Conference; Trends and Issues in Tracking for Virtual Environments*, 2007, S. 1–5

[10] FOXLIN, Eric ; HARRINGTON, Michael ; PFEIFER, George: Constellation. In: CUNNINGHAM, Steve (Hrsg.): *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. New York, NY : ACM, 1998. – ISBN 0897919998, S. 371–378

[11] FREDRIKSSON, Jonas ; RYEN, Sven B. ; FJELD, Morten: Real-time 3D hand-computer interaction. In: TOLLMAR, Konrad (Hrsg.): *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges*. New York, NY : ACM, 2008. – ISBN 9781595937049, S. 133

[12] GOLUB, Gene ; KAHAN, William: Calculating the singular values and pseudo-inverse of a matrix. In: *ISAM* 2 (1965), Nr. 2, 205–224. `http://statistics.uchicago.edu/~lekheng/courses/302/classics/golub-kahan.pdf`

[13] GRIMES, Gary J.: *Digital data entry glove interface device*. 1983

[14] HERNANDEZ-REBOLLAR, Jose L. ; KYRIAKOPOULOS, Nicholas ; LINDEMAN, Robert W.: The AcceleGlove. In: APPOLLONI, Tom (Hrsg.): *ACM SIGGRAPH 2002 conference abstracts and applications*. New York, NY : ACM, 2002. – ISBN 1581135254, S. 259

[15] HUTTENLOCHER, D. P. ; KLANDERMAN, G. A. ; RUCKLIDGE, W. J.: Comparing images using the Hausdorff distance. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (1993), Nr. 9, S. 850–863. `http://dx.doi.org/10.1109/34.232073`. – DOI 10.1109/34.232073. – ISSN 01628828

[16] KOREIN, James U.: *A geometric investigation of reach: Zugl.: Pennsylvania Univ., Diss. : 1984*. Cambridge, Mass. : MIT Press, 1985 (ACM distinguished dissertations). – ISBN 0262111047

[17] KURODA, T. ; TABATA, Y. ; GOTO, A. ; IKUTA, H. ; MURAKAMI, M. u. a.: Consumer price data-glove for sign language recognition. In: *Proc. of 5th Intl Conf. Disability, Virtual Reality Assoc. Tech., Oxford, UK*, 2004, S. 253–258

[18] LA GORCE, Martin de ; PARAGIOS, Nikos ; FLEET, David J.: Model-based hand tracking with texture, shading and self-occlusions. In: *IEEE Conference on Computer Vision and Pattern Recognition, 2008*. Piscataway, NJ : IEEE Service Center, 2008. – ISBN 978–1–4244–2242–5, S. 1–8

[19] LEE, Jintae ; KUNII, T. L.: Model-based analysis of hand posture. In: *IEEE Computer Graphics and Applications* 15 (1995), Nr. 5, S. 77–86. `http://dx.doi.org/10.1109/38.403831`. – DOI 10.1109/38.403831. – ISSN 02721716

[20] LIN, John ; WU, Ying ; HUANG, T. S.: Modeling the constraints of human hand motion. In: *Proceedings, Workshop on Human Motion*. Los Alamitos, Calif : IEEE Computer Society, 2000. – ISBN 0–7695–0939–8, S. 121–126

[21] MAJEAU, L. ; BORDUAS, J. ; LORANGER, S. ; EL-IRAKI, Y. ; LAVOIE, J. ; BANVILLE, D. ; LATENDRESSE, V. ; BELAND, V. ; DANIEL-RIVEST, J. ; THIAW, A. ; BAMBARA, H. ; BEAUSOLEIL, T. P. ; TROTTIER-LAPOINTE, W. ; LAPOINTE, J.: Dataglove for consumer applications: Low cost dataglove using optical fiber sensor. In: *7th International Workshop on Fibre and Optical Passive Components (WFOPC), 2011*. Piscataway, NJ : IEEE, 2012. – ISBN 978–1–4577–0861–9, S. 1–4

[22] MOESLUND, Thomas B. ; GRANUM, Erik: A Survey of Computer Vision-Based Human Motion Capture. In: *Computer vision and image understanding* 81 (2001), Nr. 3, S. 231–268. `http://dx.doi.org/10.1006/cviu.2000.0897`. – DOI 10.1006/cviu.2000.0897

[23] MOESLUND, Thomas B. ; HILTON, Adrian ; KRÜGER, Volker: A survey of advances in vision-based human motion capture and analysis. In: *Computer vision and image understanding* 104 (2006), Nr. 2-3, S. 90–126. `http://dx.doi.org/10.1016/j.cviu.2006.08.002`. – DOI 10.1016/j.cviu.2006.08.002

[24] NAKAMURA, Yoshihiko ; HANAFUSA, Hideo: Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. In: *Journal of Dynamic Systems, Measurement, and Control* 108 (1986), Nr. 3, S. 163. `http://dx.doi.org/10.1115/1.3143764`. – DOI 10.1115/1.3143764. – ISSN 00220434

[25] OPTITRACK: *flex13MocapVolume.png (680×375)*. `http://www.optitrack.com/public/images/flex13MocapVolume.png`. Version: 2017

[26] ORIN, David E. ; SCHRADER, William W.: Efficient Computation of the Jacobian for Robot Manipulators. In: *The International Journal of Robotics Research* Vol.3 (1984), Nr. No.4. – ISSN 0278–3649

[27] Raab, Frederick ; Blood, Ernest ; Steiner, Terry ; Jones, Herbert: Magnetic Position and Orientation Tracking System. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-15 (1979), Nr. 5, S. 709–718. http://dx.doi.org/10.1109/TAES.1979.308860. – DOI 10.1109/TAES.1979.308860. – ISSN 0018–9251

[28] Rijpkema, Hans ; Girard, Michael: Computer animation of knowledge-based human grasping. In: Thomas, James J. (Hrsg.): *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. New York, NY : ACM, 1991. – ISBN 0897914368, S. 339–348

[29] Rolland, Jannick P. ; Baillot, Yohan ; Goon, Alexei A.: A SURVEY OF TRACKING TECHNOLOGY FOR VIRTUAL ENVIRONMENTS: Center for Research and Education in Optics and Lasers (CREOL) University of Central Florida, Orlando FL 32816. In: *(Fundamentals of wearable computers and augumented reality* (2001)

[30] Stauffer, C. ; Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, Ca : IEEE Computer Society, 1999. – ISBN 0–7695–0149–4, S. 246–252

[31] Sturman, D. J. ; Zeltzer, D.: A survey of glove-based input. In: *IEEE Computer Graphics and Applications* 14 (1994), Nr. 1, S. 30–39. http://dx.doi.org/10.1109/38.250916. – DOI 10.1109/38.250916. – ISSN 02721716

[32] Thomas A. DeFanti ; Daniel J. Sandin: *Final Report to the National Endownment of the Arts*

[33] Wampler, Charles: Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. In: *IEEE Transactions on Systems, Man, and Cybernetics* 16 (1986), Nr. 1, S. 93–101. http://dx.doi.org/10.1109/TSMC.1986.289285. – DOI 10.1109/TSMC.1986.289285. – ISSN 0018–9472

[34] Wang, Robert Y. ; Popović, Jovan: Real-time hand-tracking with a color glove. In: *ACM transactions on graphics (TOG)* Bd. 28, 2009, S. 63

[35] Welch, G. ; Foxlin, E.: Motion tracking: No silver bullet, but a respectable arsenal. In: *IEEE Computer Graphics and Applications* 22 (2002), Nr. 6, S. 24–38. http://dx.doi.org/10.1109/MCG.2002.1046626. – DOI 10.1109/MCG.2002.1046626. – ISSN 02721716

[36] Wolovich, W. ; Elliott, H.: A computational technique for inverse kinematics. In: *The 23rd IEEE Conference on Decision and Control*, IEEE, 1984, S. 1359–1363

Eidesstattliche Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Gummersbach, xx. August 2016

Max Mustermann