Computer Graphics International (CGI), June 16–19, 2004, Crete, Greece. IEEE Computer Society Press.

Visual-fidelity dataglove calibration

Ferenc Kahlesz Gabriel Zachmann Reinhard Klein
Dept. of Computer Science II, University of Bonn, Germany
{fecu, zach, rk}@cs.uni-bonn.de

Abstract

This paper presents a novel calibration method for datagloves with many degrees of freedom [1]. *The goal of our method is to establish a mapping from the sensor values of the glove to the joint angles of an articulated hand that is of "high visual" fidelity. This is in contrast to previous methods that aim at determining the absolute values of the real joint angles with high accuracy. The advantage of our method is that it can be simply carried through without the need for auxiliary calibration hardware (such as cameras), while still producing visually correct mappings. To achieve this, we developed a method that explicitly models the cross-couplings of the abduction sensors with the neighboring flex sensors. The results show that our method performs superior to linear calibration in most cases.*

1. Introduction

A quintessential service of immersive VR/AR systems should be to let users grab and manipulate virtual objects in a way which closely resembles natural interaction with 3D objects in the real world. The most widespread input devices for this kind of task are various instrumented datagloves. These gloves measure directly or indirectly joint angles of the human hand and these measurements drive a virtual hand in a virtual environment. Simulating interactions between the virtual hand and a virtual object, such as grasping, manipulation could be imitated in the virtual environment.

Note that the quality of any simulation is inherently dependent on this measurement. Unfortunately, on many gloves that collect enough information, the sensors are cross-coupled,because of their placement, usually in a non-linear way, which should be taken into account during glove calibration. A tempting solution to overcome this issue would be to use external sensors (e.g. vision systems or hand masters) to provide the ground-truth of the should be measurements to handle the cross-coupling. The drawback of this approach is that any extra hardware required for calibration will prevent the method from achieving wide-spread usage. Tedious calibration procedures are also among the reasons why most gesture recognition subsytems in VR can accept only a few distinct gestures. This, in turn, enables only clumsy and unnatural interaction with the virtual environment. Therefore, *there is a demand for calibration methods*, which:

- are simple and easy to carry out,
- possibly work without complex external sensory hardware,
- make for robust tracking of fine hand movements.

The work described in this paper addresses the problem of providing a calibration method for high-degree of freedom datagloves, which handles the cross-coupling of the sensors and does not require any external hardware elements (only an edge of a table and similarly simple devices are needed). Our goal is to create a calibration, which is visually appealing — this means that the virtual hand driven by the dataglove should closely resemble the posture taken by the user (eg. if the user's pinky finger and thumb touch each other, the virtual hand should do the same), however we do not require that fingertip positions should be measured precisely.

We believe that *this kind of calibration has high potential for application in VR interactions*, since the perception-action control loop is closed through a virtual hand, which behaves like the hand of the user. This will make 3D interaction more intuitive and efficient.

The paper is organized as follows. First, we review previous work. Then describe the chosen kinematics of the hand and develop a model of cross-coupling of the index, middle, ring and pinky finger sensors based on their statistical behaviour. Finally we present experimental results and conclude the paper.

1.1. About datagloves

At the time of this writing there are two datagloves on the market that have the capability of measuring high-degree of freedom hand movements, namely Immersion's Cyberglove® and the 5DT Dataglove 16. Both gloves

[1]Gloves that measure at least 2 flexes per finger plus abduction/adduction, eg. Immersion's Cyberglove®.

have similar sensor layout and officially supported independent linear sensor calibration. Therefore, they share the same cross-coupling problems. To evaluate our calibration method, we used a Cyberglove, which is actually the *de facto* industrial standard for complex finger movement measurement.

Of course, the described method can be applied to any dataglove, where similar cross-coulings arise.

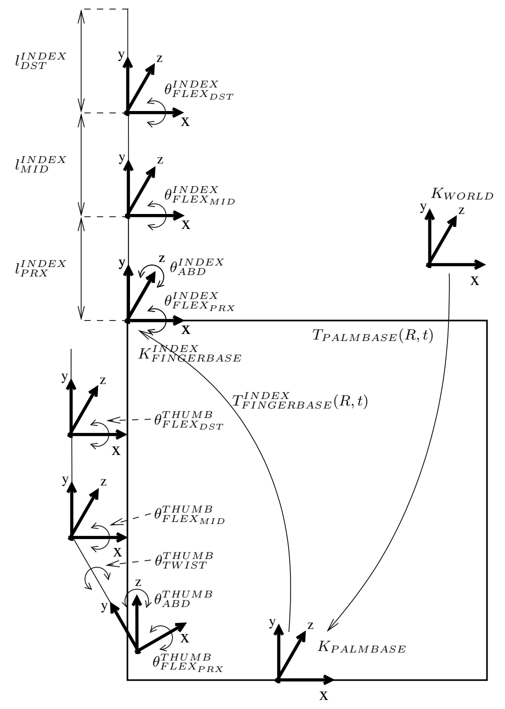
2. Previous work

The official VirtualHand®User's Guide[9] and Immersion's FAQ document[1] describe a tedious non-automatic process if one is interested in "exact" calibration of the sensors of the glove. Moreover it's also noted in the FAQ that "It is usually difficult to get the ring and especially the pinkie finger tip to touch the thumb tip.". According to the FAQ this is caused by not utilizing the palm arch sensor in the calibration.

Chou et al.[2] developed a method for Cyberglove calibration, which used linear regression to establish the mappings between joint angle values and raw sensor readings of the dataglove. While most of the mappings are one-to-one mappings (the measured joint angle depends exactly on one sensor reading), they also allowed for one-to-two mappings (a joint angle depends on two sensor readings). The one-to-two mappings are used to simulate an absolute abduction sensor for the Cyberglove, making it possible to measure four abduction values. The general applicability of this method is limited by the need for a calibrated vision system to measure the ground-truth joint angle values.

Although the attainable level of precision seemed to be satisfactory for simple VR systems driven by a few gestures, the robotics telemanipulation research community needed a much better recovery of the fingertip positions. Fischer et al.[3] used a stereo vision system to measure the real 3D positions of the fingertips, while also storing the joint sensor readings of the dataglove. A neural network was trained with the gathered data to achieve an average tip position error of 0.2mm, with a worst case error of 1.8mm. As the driven robotic hand has only four fingers, only four fingers of the hand (TIMR - Thumb,Index,Middle,Ring) are calibrated. Like method [2], the wide-spread applicability is constrained by the need for a vision system.

Griffin et al.[4] and Turner[7] presented a calibration algorithm, which eliminated the need for a vision system, albeit with a greater RMS error (an average of 12mm in case of their four finger (TIMR) calibration and an average of 5.26mm for their two finger (TI) one). The basic idea of their method is to move the index and thumb (and the MT and RT, respectively) with the fingertips connected, record this trajectory, then carry out a closed kinematic chain calibration using least squares regression iteration. They model cross-coupling between the thumb abduction, flexion and twist (see section 3) linearly. However, no discussion is given about the adequacy of the linear model. A user can be calibrated in a matter of minutes.



l_{DST}^{INDEX}
l_{MID}^{INDEX}
l_{PRX}^{INDEX}
$\theta_{FLEX_{DST}}^{INDEX}$
$\theta_{FLEX_{MID}}^{INDEX}$
θ_{ABD}^{INDEX}
$\theta_{FLEX_{PRX}}^{INDEX}$
K_{WORLD}
$T_{PALMBASE}(R,t)$
$K_{FINGERBASE}^{INDEX}$
$T_{FINGERBASE}^{INDEX}(R,t)$
$\theta_{FLEX_{DST}}^{THUMB}$
$\theta_{FLEX_{MID}}^{THUMB}$
θ_{TWIST}^{THUMB}
θ_{ABD}^{THUMB}
$\theta_{FLEX_{PRX}}^{THUMB}$
$K_{PALMBASE}$
x
y
z

Figure 1. The kinematics of our hand model. For the sake of simplicity only the thumb and the index finger are shown.

A recently reported calibration method[5], which was developed for non-teleoperational purposes, abandons the idea of handling cross-couplings. It collects data from several different hand-postures and uses linear regression to relate the sensor values with the hand-postures. This calibration has the same purpose as ours, as the authors want to calibrate the CyberGlove® for "believable performance". However, [5] *takes no cross-couplings into account*, as the sensors are treated individually. The only distinction between [5] and the simple linear calibration is that in [5] the linear model is fitted to more than two sample points.

3. Hand model

We have chosen to base our hand model (see Figure 1) on the one in [4], as the correctness of the model of the index finger (the middle, ring and pinky fingers have the same kinematic structure) is exhaustively proven in [6] and the degrees of freedom of the thumb closely resemble the sensor layout of the Cyberglove. There is also a non-measured joint (θ_{TWIST}^{THUMB}), which makes for the anatomical correctness [4, 7].

The proximal link (PRX) of every finger has two degrees of fredom: flexion and abduction, while the middle

(MDL) and distal (DST) link have only flexion. The middle, ring and pinky fingers have the same layout as the index, the only difference is that the middle finger *has no abduction*, as the Cyberglove® does not have an absolute abduction sensor, so one abduction had to be removed. We do not have the thumb abduction axis offset reported in [4]. This way, we have a simpler analytic inverse geometry solution for the thumb, which we plan to use in the calibration of the thumb sensors. The fingerbase transformations ($T^{INDEX}_{PALMBASE}(t)$,...) and the fingerlink lengths (l^{INDEX}_{PRX},...) are fixed parameters of the model. The fingerlink lengths could be manually set to correspond to the user's length values, but our calibration method does not make it necessary.

The hand model also involves constraints on the angular values of the joints. These restrictions represent the minimal and maximal abductions and flexions. We call these "static hand constraints", as they are independent of the actual hand posture (see also Section 5).

4. Cross-coupled sensors

In this section we present the results of our experiments regarding the cross-coupling of the glove sensors. Please note that the sensors corresponding to the thumb are not covered by these experiments, as we do not handle the cross-couplings of the thumb at the moment.
The purpose of the tests we carried out were twofold, namely to verify that:

- A sensor is not or neglibly coupled with others. If such a test is successful, then the given sensor can be calibrated independently.
- The change of a sensor value is correlated with other(s). If this is the case, then the sensors in question should be calibrated together.

All the tests consist of moving the fingers according to some fixed pattern. These test patterns are constructed in such a way that the examined sensor value should theoretically remain unchanged and the other sensors are bent to the largest possible extent. The readings of the sensor in question are recorded and then the total-variaton and standard deviation relative to the full workspace of the sensor are computed. Should these be "small enough" (in pratice, they are never exactly zero), it indicates that the value of the given sensor is not dependent on other moving sensors, whereas if this value is "relatively high", it means that the sensor is cross-coupled with some other.

4.1. Independent sensors

In this section, we will investigate the correctness of our first hypothesis that all of the flexion sensors can be handled with an independent linear calibration (see Section 5 for details on the linear calibration).

Table 1. Proximal flex changes vs. abduction. (ws=full workspace, tv=total variation, %=$100\frac{tv}{ws}$, sd=std. dev, %=$100\frac{sd}{ws}$)

flex	ws	tv	%	sd	%
index	135	19	14.0	5.3	3.8
middle	129	12	9.3	2.7	2.2
ring	118	16	13.6	5.0	4.2
pinky	123	22	17.9	6.5	5.3

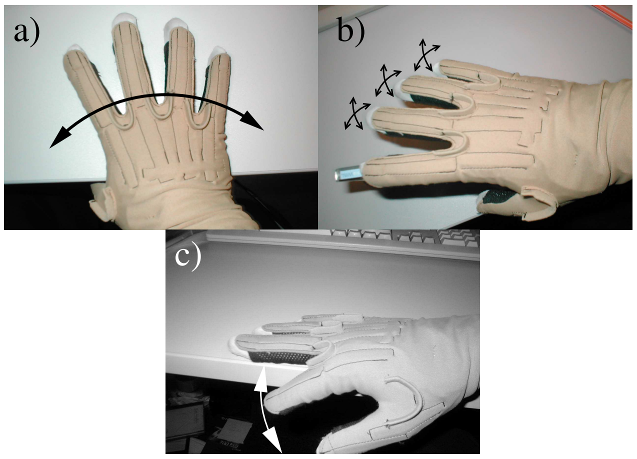


Figure 2. a) The abduction experiment. b) Examinig the proximal flex and abduction changes on the index proximal flex sensor. c) Index-middle abduction test.

For the distal and middle flex sensors we did not carry out any experiments as it seemed reasonable enough that the other sensors do not influence their value. To check the independence of the proximal flexion sensors, we made two tests.

In the first test, we examined whether abduction/adduction forces the proximal flex sensors to significantly change their value. The person who executed the experiment abducted/adducted her fingers on a table, while taking care to keep the flexions at zero (see Figure 2a). The results are presented in Table 1.

In the second test, the effect of the free movements of other fingers on one selected flexion was checked. The person who carried out the experiment kept one of her fingers at zero flex, while freely moving the other fingers. To help keeping the flex at zero, a CD jewel case has been put under the examined finger (see Figure 2b). See Table 2 for the results.

Obviously, there is some cross-coupling between the abduction and proximal flexion sensors (when one abducts one's index finger, the index flex sensor will be a little bit

Table 2. Proximal flex changes vs. free movement of the other fingers. (The notations are the same as in Table 1.)

flex	ws	tv	%	sd	%
index	135	8	5.9	2.0	1.5
middle	129	13	10.0	2.4	1.9
ring	118	12	10.1	1.8	1.5
pinky	123	10	8.1	1.7	1.4

streched). This explains that the worst case absolute deviations almost reach 20% of the workspace range in the first experiment. Nonetheless, the standard deviation values are small ($< 5.5\%$). Moreover, we definitely overestimate the sensor variations, as even if one carries out the experiments with utmost care, one tends to unwillingly change the flexion of the examined finger to a small extent.

Our findings presented in Tables 1 and 2 confirm our choice to neglect the cross-coupling effects of the proximal flexion sensors.

4.2. Cross-coupling of the abduction sensors

In this section, we investigate our second hypothesis that the abduction sensor readings depend on the proximal flexions of the IMRP[2] fingers. Let us consider the following hand movement (see Fig. 2c): an index finger is flexed along the edge of a table. During the movement, the abduction of the index-middle finger is zero. However, as the index finger flexes, the abduction sensor will stretch, thereby changing its value. To measure the extent of this change, we recorded the following three trajectories to examine the behaviour of the index-middle abduction sensor:

1. flexion/extension of the index finger, while the middle finger has zero flexion;
2. common flexion/extension of the index and middle finger by clenching and releasing a fist, keeping the abduction zero;
3. flexion/extension of the middle finger, while the index finger has zero flexion.

We also made such recordings for the other two abduction sensors. From the results in Table 3, it is clear that the value of an abduction sensor is highly dependent on its neighboring flex sensors.

5. Calibration

As we concluded in the previous section, independent linear calibration suffices for all flex sensors. However, the

[2]index,middle,ring,pinky

Table 3. Abduction sensor changes vs. neighboring flexion changes. (The notations are the same as in Table 1.)

abd.	ws	tv	%	sd	%
IM	170	106	62.4	21.7	12.8
MR	156	194	124.4	51.6	33.1
RP	145	215	148.3	51.5	35.5

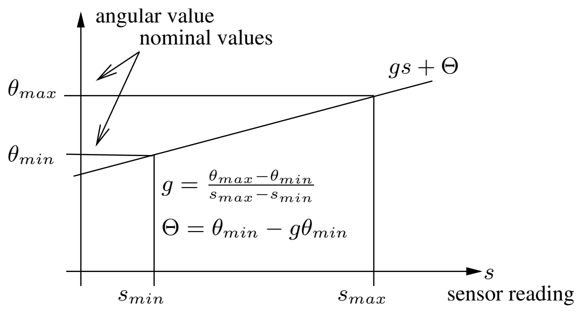


Figure 3. Linear calibration of a sensor.

abduction angles should be computed based on the value of the given abduction sensor and the neighboring flexion sensors:

$$\theta_{ABD} = f(s_{ABD}, s_{FLEX}^{left}, s_{FLEX}^{right}). \tag{1}$$

For example, the function of the middle-ring abduction on a right-handed glove depends on the sensor readings from the middle-ring abduction sensor (s_{ABD} in Eq. 1, called "ring-middle abduction sensor" in [8]), the middle finger flexion sensor (s_{FLEX}^{left}, "middle MPJ") and the ring flexion (s_{FLEX}^{right}, "ring MPJ").

5.1. Flex sensors

These sensors can be calibrated linearly, which means that the input data for the linear calibration of the flex sensors consist only of the minimal/maximal sensor readings (the two endpoints of the sensor workspace) along with the corresponding nominal angle values (the limits of the joint movements in angles, i.e. the static hand constraints for the flex joints). For details about the linear calibration see Figure 3.

5.2. Abduction sensors

To calibrate the abduction sensors, one has to find a sensible definition of the function of Eq. 1, which is a $R^3 \rightarrow R^1$ mapping. Therefore we consider it as 3D density function, so the points in $\{s_{ABD}, s_{FLEX}^{left}, s_{FLEX}^{right}\}$ space that correspond to the same abduction value define an isosurface. Our task now is to estimate the real abductin angle based on the sensor values and one or more isosurfaces.

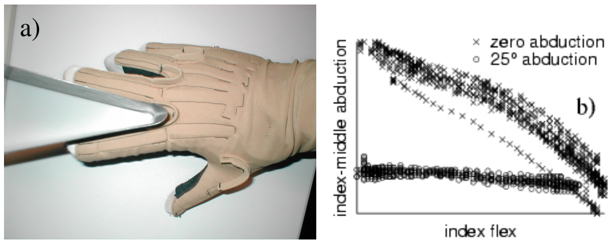


Figure 4. a) Using a wedge to record the points corresponding to the 25° isosurface. b) The trajectories of different isosurfaces may intersect.

5.2.1. Analysis of the isosurfaces.. As mentioned above, the points of an isosurface correspond to an abduction angle. So using a wedge (see Fig 4a), we can easily acquire three pointclouds, analogously to the ones measured in section 4.2. They should lie on the isosurface corresponding to the angle of the wedge. To gather points for the isosurface corresponding to zero abduction, one simply records the trajectories from the experiment in Section 4.2.

What we expect for all trajectories is that the value of the abduction sensor decreases as the neighboring fingers flex, because the sensor stretches (the abduction sensors of the dataglove report smaller values for greater abduction angles). We also expect (based on the experiments described in Section 4.1) that when only one finger flexes (left or right trajectory), the readings of the flex sensor of the non-moving finger should just slightly vary (see Fig. 5a,b). An example of sensor values measured for the zero isosurface can be seen in Fig. 5c.

An obvious approach to define function f in Eq. 1 would be to interpolate between isosurfaces of different abductions. Unfortunately, it turns out that trajectories belonging to the same flex sensor, but with different abductions may intersect, see Fig. 4b. In fact, the abduction is a function not only of $s_{ABD}, s_{FLEX}^{left}, s_{FLEX}^{right}$, but also of the values of some other sensors. However, one cannot easily increase the dimensions of the density space by taking other sensor(s) into account, because a) it is not clear, which sensor(s) should be added to the variables of the abduction function and — this is the more serious problem —, b) by measuring only a few trajectories one cannot fill that higher dimensional space with enough samples to carry out effective approximations.

To overcome this problem, we decided to approximate only the zero abduction isosurface and define the abduction as a function of the distance from this surface. This is clearly just an approximation. However, we found that it works surprisingly well (please remember, that we are not interested in absolute precision, we only want visually correct calibration of the virtual hand). To measure the abduction the proposed way, we define an isosurface function, $S_0(\cdot)$, which will be fitted to the measured data and a distance function, $D_0(\cdot)$.

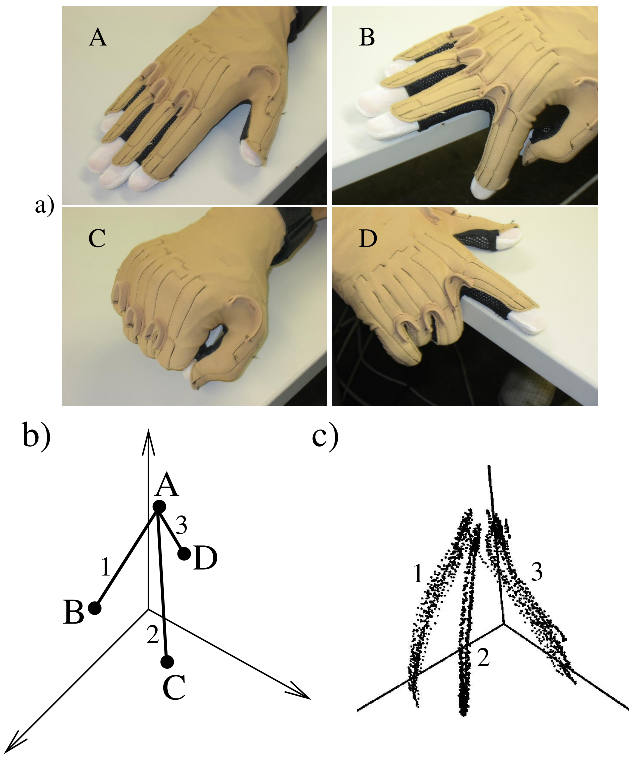


Figure 5. Structure of the grabbed trajectories for the zero isosurface of the right hand index-middle abduction. a) endpositions of the trajectories b) a sketch of the expectations about the behaviour of an abduction sensor (see text) with the endpositions shown c) real grabbed data (1 - only the index finger flexed, 2 - both index and middle fingers flexed, 3 - only the middle finger flexed)

We create $S_0(\cdot)$ through the following steps:

1. Project the points of the three trajectories to three appropriate vertical planes (the s_{ABD} axis points up), see Fig. 6, top left (for more detail about the selection of the planes, see the following subsection).
2. Fit cubic functions to each of the projected trajectories. The coordinate systems of the planes should be aligned so that the origins are common (the origin plays the role of point “A” from Fig. 5a), the y axis has the same direction as s_{ABD} and the x axes point into the planes, away from s_{ABD}. Sample the computed cubic functions equidistantly along the arc length in N points to get three pointsets in the density space: $\{p_{left}^i\}, \{p_{common}^i\}$ and $\{p_{right}^i\}, i = 1..N$. The first point of the pointsets is the common origin, the last point is the intersection of the functions with the bounding box of the dataset, in positive x direction, see Fig. 6,top right.

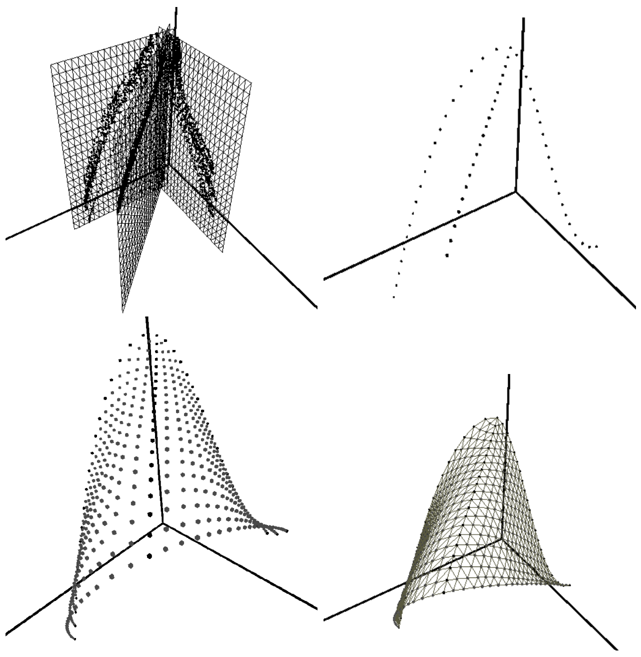


Figure 6. Steps of the zero abduction isosurface definition (see text).

3. In the planes defined by the $(p^i_{left}, p^i_{common}, p^i_{right})$, $i = 2..N$ triplets, fit parabolas to the triplets with the minima of the parabolas being at p^i_{common}. Sample these parabolas to generate additional points of the isosurface, see Fig. 6,bottom left.
4. Triangulate the isosurface using the points from step 2 and 3 to get a local linear interpolation of $S_0(\cdot)$. See Fig. 6, bottom right.

Carrying out the previous steps, we get a surface definition such that $S_0(s^{left}_{FLEX}, s^{right}_{FLEX})$, the domain of S_0 is the bounding rectangle of the dataset in the $(s^{left}_{FLEX}, s^{right}_{FLEX})$ plane. The values of S_0 should be stored in a lookup-table over its domain for fast evaluation of the function. We define $D_0(\cdot)$ as the vertical distance to this surface, and the θ_{ABD} as the linear function of this distance:

$$\theta_{ABD} = gain \cdot (S_0(s^{left}_{FLEX}, s^{right}_{FLEX}) - s_{ABD}) \qquad (2)$$

Eq. 2 does not have an offset, as if $S_0(s^{left}_{FLEX}, s^{right}_{FLEX}) = s_{ABD}$ the abduction should be zero. To determine $gain$ an additional point in density space is needed, where the abduction has known nonzero value. We measure this point at zero flex of the left and right fingers and maximum abduction.

5.2.2. Details of the isosurface creation.. In the previous subsection we defined $S_0(s^{left}_{FLEX}, s^{right}_{FLEX})$ through four steps. We now give details about the important issues of each step:

1. We want to project the recorded trajectories onto three vertical planes. The plane of the left flexion dataset should be parallel with the $(s^{left}_{FLEX}, s_{ABD})$ plane and the plane of the right trajectory with the $(s^{right}_{FLEX}, s_{ABD})$ plane (based on our experiments, we suppose that the flex readings of the sensor, which keeps zero flex, does not change during the left and right trajectories). The plane of the common pointcloud should go through the intersection line of the left and right planes (because the trajectories theoretically intersect in the "A" posture of Fig. 5a). These planes are completely determined by three parameters: X_L, the intersection of the left plane with the s^{right}_{FLEX} axis, X_R, a similar parameter of the right plane and X_M the angle of the common plane. Let $\mathbf{P}_{X_L}$, $\mathbf{P}_{X_R}$ and $\mathbf{P}_{X_M}$ denote the projection matrices into the appropriate planes, then finding the optimal planes can be cast into the following minimization problem:

$$\min_{X_L, X_R, X_M} \Big(\sum_{y \in LEFT} (\mathbf{P}_{X_L} y - y)^T (\mathbf{P}_{X_L} y - y) + \sum_{y \in COMMON} (\mathbf{P}_{X_M} y - y)^T (\mathbf{P}_{X_M} y - y) + \sum_{y \in RIGHT} (\mathbf{P}_{X_R} y - y)^T (\mathbf{P}_{X_R} y - y) \Big) \qquad (3)$$

As the Jacobian of the error function can be computed, Eq. 3 could be solved with the Levenberg-Marquardt method. Initial values for X_L, X_R can be chosen so that the planes go through the centres of gravity of the left and right trajectories, then X_M is determined so that the common plane lies on the c.o.g. of the common dataset and the intersection of the initial left and right planes.
2. The crucial part of this step is that the three cubic functions should be fitted so that they are equal at $x = 0$. To achieve this, first cubic functions without a constant tag ($Ax^3 + Bx^2 + Cx$) are fitted individually to the projected points via SVD. Using this fit as initialization, a nonlinear least squares fit can be carried out, where the cubic functions have separate A, B, C coefficients, but a common constant tag. The Jacobian of the fit error function can be determined analytically, so one can apply the Levenberg-Marquardt algorithm for this problem as well. The equidistant sampling of the curves can be done using a subdivision based on the arc length, which could be computed with numeric integration.
3. When fitting parabolas to the corresponding three points of the three approximated datasets, care should be taken that the coordinate system in the fitting plane is aligned in such a way that the extremum of the fitted parabolas are as close as possible to the point orig-

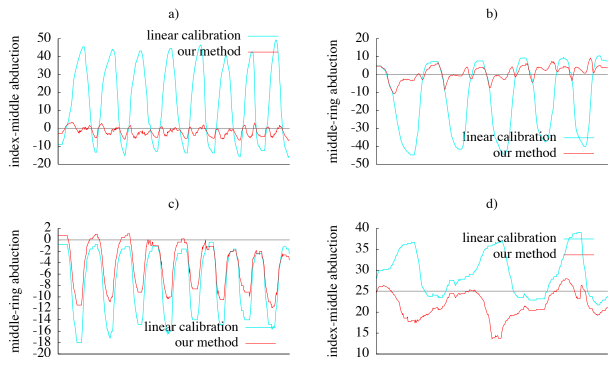


Figure 7. Comparison of our calibration algorithm and the linear method (see text).

inating from the cubic fit of the common dataset, thus ensuring that the common trajectory truly constitutes the “spine” of the zero isosurface. As we did not find an analytical solution to the problem of determining a coordinate system, in which the extremum of the fitted parabola is exactly lies in the desired point, we have chosen our planar coordinate system to have its origin in the point of the common dataset and generated a number of fits by rotating the coordinate system in the plane. Then, the coordinate system aligment, which provided the smallest absolute value of derivative of the fitted parabola at $x = 0$ was selected. The equidistant subdivision of the resulting parabola can be done based on the arc length, similarly to step 2.

4. The realization of this step is straightforward.

6. Results

We tested our calibration and compared it to other methods ([9], [2],[5]). To judge the performance of our approach, we asked two questions:

1. To which extent is our algorithm able to compensate for the learned incorrect abduction sensor readings?
2. How does it perform in other circumstances?

6.1. Comparison with the linear independent method

To examine the compensation capabilities, we recorded the abduction angles generated by our method and by the independent calibration ([9]). The results of such a measurement can be seen in Fig. 7a,b,c. The plots show the abduction measurements of the zero abduction trajectory of the index, middle, and ring finger. The x axes show the sample count, the y axes the measured abduction value in degrees as the finger is flexed and extended. The nominal value of the abduction is zero everywhere. The results are clearly satisfying: our method reports far closer values to zero than does the linear calibration. We also checked the other zero trajectories and got similar results. This means that our method is capable to compensate for the faulty abduction sensor readings.

To answer the second question, we tested our method by measuring trajectories with 25° abduction using a wedge. A result for the index-middle abduction can be seen in Fig. 7d. While not being able to perfectly compensate the incorrect sensor values, our method maintains equal performance.

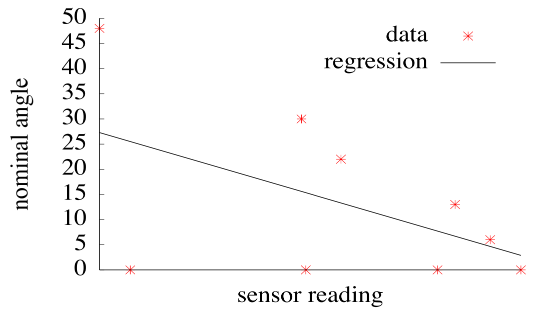


Figure 8. Angle/sensor value pairs for linear regression calibration and a fitted linear model.

6.2. Comparison with the linear regression methods

Next, we compared our methods to the ones which utilize linear regression to carry out the calibration ([2],[5]) [3].

These methods gather some corresponding angle/sensor value pairs and then fit a linear model to the data (see Fig. 8). The four points on the ‘x’ axis correspond to the A,B,C,D points from section 5.2.1. It is clear that taking these points into account as well, fitting of a linear model is not justified, as these points should be rather treated as outliers in this setting.

Similarly to section 6.1, we compared the methods on two different trajectories. One is along the learned zero abduction movement and the other is at the unlearned 25° abduction. Results are presented in Fig. 9. It can be seen that although the linear regression algorithm performs better than the two-point linear calibration, our method produces much more accurate results in the case of the learned trajectory (Fig. 9a), while performs similarly to the regression method at 25° (Fig. 9b).

6.3. Other methods

We did not compare our methods to [3] and [4], as the goals of the these two calibration schemes and ours differ fundamentally: both were developed for high absolute accuracy needed in robotic telemanipulation, while ours tries

[3][5] calibrates only the index and pinky finger finger abduction this way. The ring and middle finger abduction is calibrated with multiple linear regression analysis and the input data for these sensor calibrations are generated with a vision system, so we did not compare these two sensor calibrations to ours.

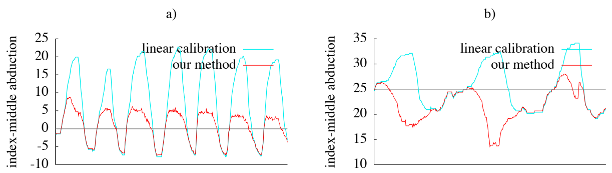


Figure 9. Comparison of our calibration and the linear regression method.

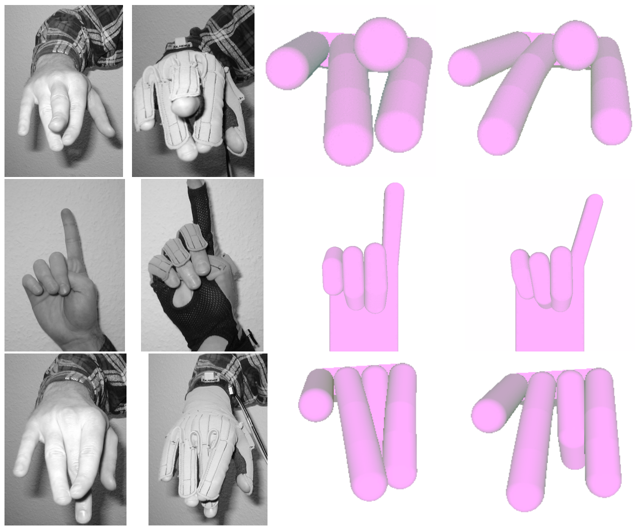


Figure 10. Comparison of our calibration and the linear regression method. The first two columns depict the real hand posture, the third the virtual hand configuration generated by our calibration and the fourth by the linear regression method.

to achieve a visually plausible virtual hand (important is the relative accuracy) that allows for fine manipulation of virtual objects.

While these methods most probably would provide better results than ours for the fingertip positions, [3] requires a calibrated stereo vision system with an adorned dataglove for the calibration, while we do not need any external hardware device.

6.4. Visual fidelity

We illustrate the visual performance of our method through some example real world hand gestures and their corresponding virtual representation (see Fig. 10). For each gesture, two virtual hand configurations are presented. One refers to the joint angle values obtained by our calibration of the abduction sensors and the other by the linear regression method. Obviously our method provides much more visually plausible results.

7. Conclusions and future work

We developed a new method that explicitly models the cross-couplings of the abduction sensors with the neighboring flex sensors of a high-degree of freedom dataglove. This method provides a very simple calibration procedure that can be performed without additional devices. It computes each abduction as a function of three sensors (left/right flex and the actual abduction). Despite its simplicity, our method produces much more visually convincing hand postures than the linear regression method, although the calibration procedure is much simpler. The calibration of a new user can be performed within a couple of minutes.

Our most important avenue for future work is to develop a simialar thumb calibration scheme, which allow for high-visual fidelty performance of the virtual thumb.

References

[1] http://www.immersion.com/3d/support/faqs.php. 2

[2] T.-S. Chou, A. Gadd, and D. Knott. Hand-eye: A vision based approach to data glove calibration. 2000. 2, 7

[3] M. Fischer, P. van de Smagt, and G. Hirzinger. Learning techniques in a dataglove based telemanipulation system for the DLR hand. 1998. 2, 7, 8

[4] W. B. Griffin, R. P. Findley, M. L. Turner, and M. R. Cutkosky. Calibration and mapping of a human hand for dexterous telemanipulation. 2000. 2, 3, 7

[5] A. S. Menon, B. Barnes, R. Mills, C. D. Bruyns, A. Twombly, J. Smith, K. Montgomery, and R. Boyle. Using registration, calibration, and robotics to build a more accurate virtual reality simulation for astronaut training and telemedicine, 2003. 2, 7

[6] R. N. Rohling and J. M. Hollerbach. Modeling and parameter estimation of the human index finger. 2

[7] M. L. Turner. *Programming Dexterous Manipulation by Demonstration*. PhD thesis, 2001. 2

[8] Virtual Technologies, Inc. *CyberGlove®Reference Manual*, 1998. 4

[9] Virtual Technologies, Inc. *VirtualHand v2.5 User's Guide*, 2001. 2, 7