

# Analysis of Hand Synergies for Inverse Kinematics Hand Tracking

Matthias Schröder<sup>1</sup>, Jonathan Maycock<sup>2</sup>, Helge Ritter<sup>2</sup>, Mario Botsch<sup>1</sup>  
<sup>1</sup>Computer Graphics & Geometry Processing Group, <sup>2</sup>Neuroinformatics Group,  
Bielefeld University, Germany

**Abstract**—We present a method for real-time bare hand tracking that utilizes natural hand synergies to reduce the complexity and improve the plausibility of the hand posture estimation. The hand pose and posture are estimated by fitting a virtual hand model to the 3D point cloud obtained from a Kinect camera using an inverse kinematics approach. We use real human hand movements captured with a Vicon motion tracking system as the ground truth for deriving natural hand synergies based on principal component analysis. These synergies are integrated in the tracking scheme by optimizing the posture in a reduced parameter space. We show that this synergistic hand tracking approach improves runtime performance and increases the quality of the posture estimation.

## I. INTRODUCTION

Tracking the complete articulation of a freely moving hand is a problem that is an ongoing research topic. Many existing hand tracking solutions either require the user to wear cumbersome equipment, are expensive or are inadequate for real-time tracking. Some methods that use consumer-level depth sensors can detect the positions of individual fingers and provide a means for rough gesture interaction, but do not accurately reconstruct the user's hand posture with full degrees of freedom (DoFs). We have built a hand tracking system that uses a Kinect camera to estimate the full articulation of a user's bare hand in real-time. Our method is a generative approach that is based on fitting a virtual hand model to the 3D point cloud obtained from the Kinect sensor's depth camera. We estimate the hand articulation by finding the pose and posture parameters that minimize the error between the point cloud and the model surface using inverse kinematics. In doing so, we find the deformation of the 3D hand model that best approximates the observed state of the user's hand. A prevalent issue in tracking a highly articulated object like a hand is the number of DoFs that must be optimized. The analysis of hand synergies aims to identify high-level relationships in hand articulation in order to sensibly reduce the dimensionality of hand posture representations. We obtain such hand synergies through the principal component analysis of motion capture data and use them directly in the tracking process to reduce the parameter space and to naturally constrain the hand posture estimation.

## II. RELATED WORK

There are two main approaches to the hand pose estimation problem, namely appearance based [1], [2] and model based methods [3], [4]. In a recent paper [5], we used a data-driven appearance based approach to control an anthropomorphic robot hand. A color glove was detected and using a nearest

neighbor search in an image database the closest matching image and corresponding posture and coarse rotation were retrieved. However, appearance-based methods suffer when the hand strays from configurations that are not known and therefore can perform poorly in certain free moving hand situations. It is for this reason that we have decided to investigate a model based approach.

Recently, existing model based approaches that heretofore had proved too computationally expensive for real-time applications [6] are now becoming feasible. Oikonomidis *et al.* presented impressive results using a multi-camera setup [3] and using the Kinect camera [7], [8], but these approaches suffer from high computational complexity and had to be optimized to run on a GPU. Ballan *et al.* [4] used features such as edges, optical flow and salient points to estimate the articulated pose within a single differentiable function. They achieved lower posture estimation errors than those of Oikonomidis *et al.* [3], but thus far their approach is not real-time. The so-called *curse of dimensionality* is an issue that has to be addressed by all hand posture estimation approaches and a possible solution we have considered is to use synergies to reduce the associated computational complexity.

Bernstein [9] was the first to come up with the idea of synergies and defined them to be high-level control schemes for kinematic parameters. He suggested that they could provide a mechanism by which the central nervous system controls the high DoFs human hand in an efficient way. It was not, however, until Santello *et al.* [10] published their paper on hand synergies that the rehabilitative protheses and robots research communities realised their potential in terms of controlling articulated hands and arms. Santello revealed that 90% of the variance in the data of grasps directed towards household objects could be described by as little as 3 principal components (PCs). Many other studies have since supported this view [11], [12].

The potential of synergies for the field of robotics has received a lot of interest over the past few years. In 2007, Ciocarlie *et al.* [13] used the PC space of grasps to not only reduce control complexity, but as an interface that allowed control of robot hands with different kinematic structures. Bicchi *et al.* [14] suggested that synergies that contribute most to the pre-grasping phase are likely the ones needed to ensure force closure grasps. Indeed, Gambini *et al.* [15] have suggested that as little as two synergies are sufficient to establish force-closure in a simulated robot hand and that

increasing the numbers of synergies used to control such a hand has limited or no effect thereafter.

In the medical domain, synergies have shown great promise to aid the control of prosthetic devices. An early paper by Popovic and Popovic [16] demonstrated the feasibility of a synergistic control of an elbow neural prosthesis device. More recently, this approach was extended to include some finger movements [17]. Vinjamuri *et al.* [18] used a simplified convolutive mixture model to convert kinematic synergies into temporal postural synergies, which revealed interesting strategies of finger coordination that they suggest could be used to help control prosthetic devices.

Using synergies or other methods to reduce the dimensionality of the problem of hand pose estimation for hand tracking applications has a precedence. In an early paper, Lee and Kunii [19] placed a set of constraints on joint angle limits and movement types to reduce the hand model's DoFs or reject infeasible inverse kinematics solutions. Wu *et al.* [20] used the fact that hand articulations could be represented in a lower dimensional configuration space to perform a Monte Carlo tracking algorithm. Unlike Lee and Kunii, they were able to track the hand in real-time, but their method was view dependent (hand had to palm-wise face the camera) and rotation and scaling were not considered. Another view dependent hand tracking particle filter approach [21] also reduced the dimensionality of the problem by using independent component analysis to compute five basis components, one for each finger. Using a synergistic approach to reduce the DoFs of a virtual hand model, we also reduce the high computational complexity associated with model based approaches, while at the same time realise realistic (view unrestricted) real-time bare hand tracking.

### III. KINEMATIC HAND MODEL

We use a kinematic hand model consisting of 16 joints: three for the proximal, intermediate and distal phalanges of each finger and one wrist joint. The articulation of the hand is represented by 22 degrees of freedom in our model: each joint has a flexion-extension axis, the wrist joint and the fingers' base joints have an additional abduction-adduction axis. In addition to the 22 joint angles controlling the hand's posture, the pose of the hand is represented by 6 degrees of freedom for the global translation and rotation. In total we use 28 parameters to control the pose and posture of the hand in our system.

These parameters and the kinematic chains of the joint hierarchy define the forward kinematics of the hand, which can be expressed in terms of a product of affine transformations. A joint's *local* transformation consists of the rotation defined by its joint angle parameters and the translation relative to its parent joint, if there is one. The *global* transformation  $\mathbf{T}_j$  matrix of joint  $j$  is given by the product of the local transformations along its kinematic chain:

$$\mathbf{T}_j = \prod_{i=1}^n T_i(\theta_i), \quad (1)$$



Fig. 1. Virtual hand model with its control skeleton.

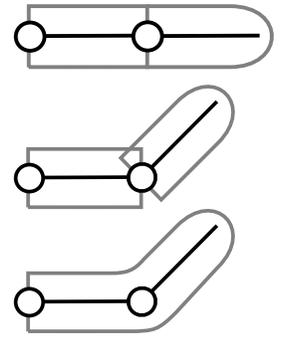


Fig. 2. Rest pose, forward kinematics and skinning.

where  $T_i(\theta_i)$  is the local transformation matrix associated with the element  $\theta_i$  of the kinematic parameter vector  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{28})^T$ .

The virtual hand used to approximate the user's hand posture in our system is represented as a triangle mesh and is deformed according to the articulation of the joints defined in the kinematic hand model. The joint hierarchy serves as the *skeleton* of the virtual hand model. Figure 1 shows the virtual hand model and its skeleton. A point  $\mathbf{v}$  on the surface of the mesh can be transformed relative to a joint  $j$  based on the forward kinematics of the skeleton:

$$\mathbf{v}' = \mathbf{T}_j \hat{\mathbf{T}}_j^{-1} \mathbf{v}, \quad (2)$$

where  $\hat{\mathbf{T}}_j$  is the rest pose transformation of joint  $j$  and its inverse is used to transform  $\mathbf{v}$  to the joint's local coordinate frame. Since the transformation matrices  $\mathbf{T}_j$  depend on the parameter vector  $\boldsymbol{\theta}$ , the transformation of a point  $\mathbf{v}$  based on the skeleton can be expressed as a function of the parameters:  $\mathbf{v} = \mathbf{v}(\boldsymbol{\theta})$ . We use this expression to calculate the forward kinematics during the tracking process. However, calculating the deformed vertex positions of the mesh in this naive manner results in an unrealistic and blocky animation of the model. In order to obtain a smooth deformation of the model for the visualization, we use linear blend skinning [22] (LBS). LBS generates a smooth deformation of a polygon mesh by calculating its deformed vertex positions as a weighted sum of the affine transformations of multiple joints. For each vertex there is a set of weights  $(\omega_1, \dots, \omega_{16})$  which add up to 1 and describe the influence of the 16 joint transformations on the vertex deformation. The position  $\mathbf{v}'$  of vertex  $\mathbf{v}$  according to LBS is given by

$$\mathbf{v}' = \sum_{j=1}^{16} \omega_j \mathbf{T}_j \hat{\mathbf{T}}_j^{-1} \mathbf{v}. \quad (3)$$

The result of applying this position update to all vertices of the hand mesh is a smooth deformation of the virtual hand model in accordance to the control parameters of the kinematic model. Figure 2 illustrates the forward kinematics and skinning for a kinematic chain of two joints.

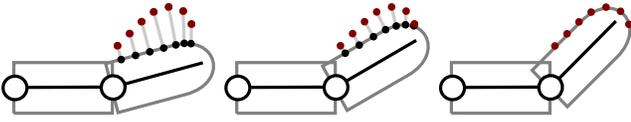


Fig. 3. Inverse kinematics posture estimation. The red points are target positions in the sensor point cloud, the black points are the corresponding effector positions on the surface of the model. The joint angle is updated such that the target-effector error is minimized.

#### IV. INVERSE KINEMATICS HAND TRACKING

In our hand tracking approach, the pose and posture of the user’s hand is estimated by fitting the virtual hand model to the point cloud obtained from a Kinect [23] sensor. By finding the articulation of the hand model that minimizes the distance to the point cloud, the state of the user’s hand that causes the observation is approximated.

The point cloud is calculated from the Kinect’s color and depth images based on a precomputed RGBD-mapping, which maps color values to the pixel coordinates of the depth image and uses the camera parameters of the Kinect’s color and depth cameras to calculate the global 3D positions of the sensor points. The hand is then segmented by detecting skin-colored pixels and omitting points whose coordinates are outside of a predefined working volume. The remaining points in the point cloud define the target constraints for the hand model fitting process.

These target points are matched to their closest points on the surface of the hand model by calculating the minimal point-to-triangle distances. Based on these point correspondences we formulate the problem of estimating the posture of the hand as finding the posture parameters (including joint angles and global pose) that transform the hand’s skeleton such that the error between the model and target positions is minimized. This is an inverse kinematics (IK) problem in which the points on the model surface are regarded as effector positions relative to the skeleton that are constrained to move towards their corresponding target positions in the sensor point cloud. Figure 3 illustrates the principle with a simplified example. We solve the IK problem using the popular damped least squares method described in [24].

Following the notation of [24], the effector positions can be written as a stacked vector  $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_k)^T$  and the target positions as  $\mathbf{t} = (\mathbf{t}_1, \dots, \mathbf{t}_k)^T$ . As stated in Section III, the effector positions can be expressed as functions of the parameters  $\mathbf{s}_i = \mathbf{s}_i(\boldsymbol{\theta})$ ,  $i = 1, \dots, k$  or  $\mathbf{s} = \mathbf{s}(\boldsymbol{\theta})$ . The solution to the IK problem,  $\mathbf{t} = \mathbf{s}(\boldsymbol{\theta})$ , can be found by iteratively seeking updates,  $\Delta\boldsymbol{\theta}$ , that solve the equation

$$\mathbf{J}\Delta\boldsymbol{\theta} = \mathbf{e}, \quad (4)$$

where  $\mathbf{e} = \mathbf{t} - \mathbf{s}$  is the target-effector error and  $\mathbf{J}$  is the  $3k \times 28$  Jacobian matrix of the effector positions:

$$\mathbf{J} = \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}} = \left( \frac{\partial \mathbf{s}_i}{\partial \theta_j} \right)_{i,j}. \quad (5)$$

The straightforward calculation of the Jacobian entries is described in [24]. The damped least squares solution of the IK problem is the value of  $\Delta\boldsymbol{\theta}$  that minimizes

$$\|\mathbf{J}\Delta\boldsymbol{\theta} - \mathbf{e}\|^2 + \lambda^2\|\Delta\boldsymbol{\theta}\|^2. \quad (6)$$

The damping term  $\lambda^2\|\Delta\boldsymbol{\theta}\|^2$  penalizes large changes in the parameter vector and stabilizes the solution. This leads to the parameter vector update

$$\Delta\boldsymbol{\theta} = (\mathbf{J}^T\mathbf{J} + \lambda^2\mathbf{I})^{-1}\mathbf{J}^T\mathbf{e}. \quad (7)$$

We only apply a small amount of damping to the posture parameters and none to the global pose parameters. In addition to the numerical stabilization provided by the damping term, we stabilize the update by performing step length control. If the error did not decrease after a parameter update, the update step is halved. This is iterated until the error decreases or converges. This iterative process stabilizes the update in cases where no unique solution can be found and prevents oscillation.

The overall hand tracking process is as follows. After segmenting the hand in the Kinect point cloud and finding the target-effector correspondences, the pose estimation is initialized by finding the rigid transformation between the target and effector point clouds using a common approach based on eigenvector analysis and quaternions [25] and transforming the hand model accordingly. Next, new correspondences are computed and used as input for the IK-based pose and posture estimation. During this process, the parameter update is computed according to Eq. 7 and the effector points are moved according to the updated skeleton forward kinematics. This is iterated until the target-effector error converges. As a result the virtual hand model is aligned with the observation point cloud, which yields the hand posture estimation. The final pose and posture estimation of the current frame is used as initialization for the next frame.

#### V. PRINCIPAL COMPONENT ANALYSIS

In the approach outlined in the previous section all 28 parameters of the kinematic model are freely optimized during the IK update, which allows for high freedom of movement, but can also result in implausible hand articulations, especially in cases of incomplete or ambiguous sensor data. The hand posture optimization space can be reduced in a meaningful way using hand synergies derived from the principal component analysis of a data set containing real human hand motions. Performing dimension reduction based on the most significant principal components provides a way to decrease the number of parameters that need to be optimized and to implicitly constrain the estimated hand postures to plausible ones resulting from the ground truth data set.

In order to obtain a data set that could be used to this end, we captured various human hand motions with a Vicon motion tracking system [26]. The positions of 16 retro-reflective markers placed on a human hand were tracked by

the Vicon system and afterwards used to calculate joint angles corresponding to our kinematic model [27]. The global pose information was omitted from the data, since they have no influence on the hand synergies we are interested in. The captured hand motions included various manual interaction tasks, such as different grasping and twisting motions, sign language and general hand movements exploring the hand's natural degrees of freedom. This gave us a varied set of hand postures that covered many aspects of natural hand articulation.

The final data matrix,  $\mathbf{X}$ , of  $m$  entries of the 22-dimensional posture data was pre-processed to have zero mean before PCA was performed on it. This resulted in a  $22 \times 22$  matrix of eigenvectors,  $\mathbf{V}$ , and the set of 22 eigenvalues  $\lambda = (\lambda_1, \dots, \lambda_{22})$ . Taking the eigenvectors in  $\mathbf{V}$  corresponding to the  $l$  largest eigenvalues yields a  $22 \times l$  matrix of principal components,  $\mathbf{W}$ . Since the data used for PCA only contains the 22 joint angles and not the additional 6 pose DoFs, we construct the conversion matrix  $\mathbf{M}$  that maps from the reduced  $(6 + l)$ -dimensional principal component-space (PC-space) to the  $(6 + 22)$ -dimensional parameter space:

$$\mathbf{M} = \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{W} \end{pmatrix}, \quad (8)$$

where  $\mathbf{I}$  is a  $6 \times 6$  identity matrix, requiring the global pose parameters to be the first 6 in the parameter vector. The full parameter vector  $\theta \in \mathbb{R}^{6+22}$  is thus converted to the reduced parameter vector in PC-space,  $\alpha \in \mathbb{R}^{6+l}$ , by applying the mapping

$$\alpha = \mathbf{M}^T(\theta - \mu), \quad (9)$$

where  $\mu \in \mathbb{R}^{6+22}$  is the mean vector of the data matrix  $\mathbf{X}$  with additional zero-entries for the global pose DoFs. The inverse mapping is given by

$$\theta = \mathbf{M}\alpha + \mu. \quad (10)$$

This allows the PC-space parameters to be expressed as a function of the kinematic parameters,  $\alpha = \alpha(\theta)$ , and vice versa,  $\theta = \theta(\alpha)$ . In order to perform inverse kinematics hand tracking (described in Section IV) using the reduced PC-space parameters, the parameter update rule must be adapted.

Given the above mapping, the forward kinematics of an effector point  $s_i$  can be written as a function of the PC-space parameters  $\alpha = (\alpha_1, \dots, \alpha_l)^T$ :  $s_i = s_i(\alpha) = s_i(\theta(\alpha))$ . According to the chain rule, the  $3k \times l$  Jacobian matrix  $\mathbf{J}_{PC}$  of the effector positions w.r.t. the PC-parameters  $\alpha$  is:

$$\mathbf{J}_{PC} = \frac{\partial s}{\partial \alpha} = \frac{\partial s}{\partial \theta} \cdot \frac{\partial \theta}{\partial \alpha} = \mathbf{J} \cdot \mathbf{M}, \quad (11)$$

where  $\mathbf{J}$  is the Jacobian matrix defined in Equation 5. The PC-space parameter update  $\Delta\alpha$  is obtained by substituting  $\mathbf{J}_{PC}$  for  $\mathbf{J}$  in Equation 7.

This facilitates hand tracking as described in Section IV in the reduced PC-space, which decreases the size of the matrices in the update calculation and reasonably constrains

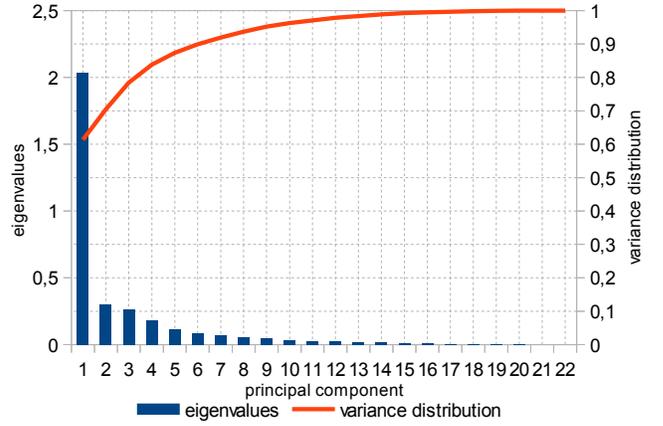


Fig. 4. Eigenvalues and variance distribution among the principal components of the full set of captured hand posture data.

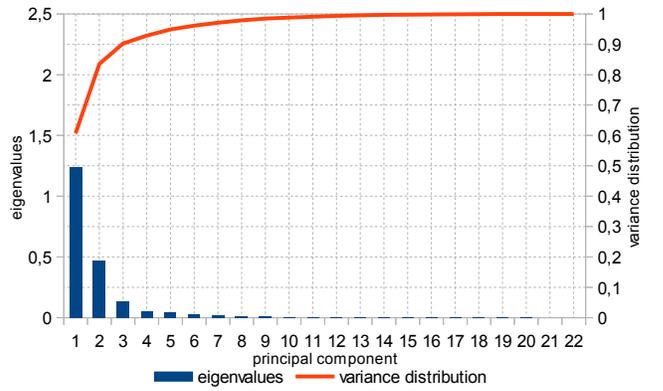


Fig. 5. Eigenvalues and variance distribution among the principal components of a data set containing only grasping movements.

the possible hand postures estimated by the tracking system. The number of PCs used for the dimension reduction,  $l$ , depends on the distribution of the data variance among the principal directions. In the following section we illustrate the PCA of the captured hand posture data and show some results of our tracking approach.

## VI. RESULTS

We present the results of the PCA for two ground-truth posture data sets: the complete data set of all captured movements (including various manual interaction motions, sign language and general hand movements) and a data set containing only grasping movements.

Figure 4 shows the distribution of the data variance among the principal components for the complete data set. Approximately 80% of the variance is covered by the principal components associated with the largest 3 eigenvalues, and approximately 90% of the variance is covered by 6 principal components. The 2 most significant principal components cover about 70% of the data variance and can already be used to represent meaningful hand synergies, which is illustrated in Figure 6. For the grasping data set the 3 most significant principal components already cover 90% of the variance,

## VII. DISCUSSION

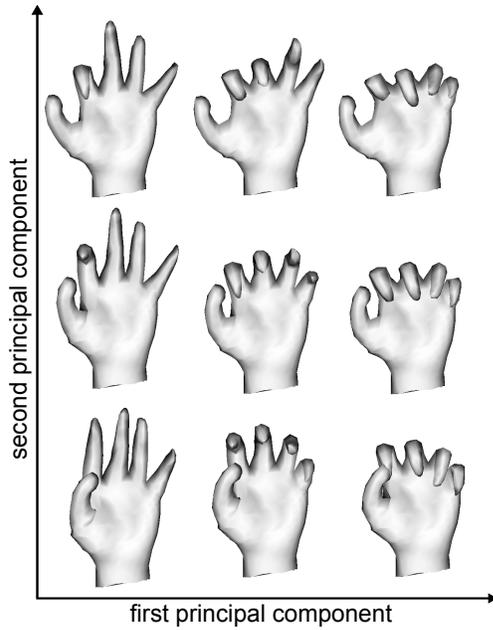


Fig. 6. Visualization of the DoFs represented by the first and second most significant principal components of the captured hand posture data set.

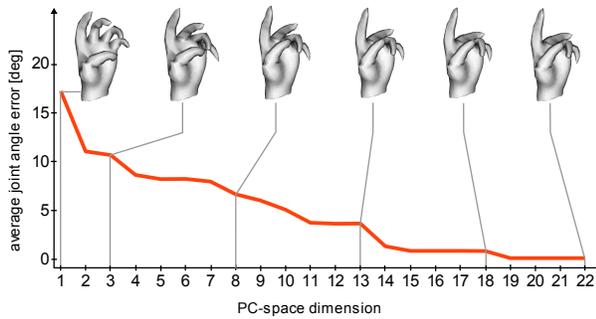


Fig. 7. Posture reconstruction and approximation error for one particular posture w.r.t. the number of principal components used in the dimension reduction.

which can be seen in Figure 5. The less varied a data set is in terms of movements contained, the less principal components are needed to cover most of its variance. To clarify the information loss resulting from dimension reduction, Figure 7 shows the approximation error for a specific hand posture w.r.t. the number of dimensions used to represent it.

Figure 8 shows an example of a tracking sequence in which the full-DoF posture estimation results in an unnatural state due to a rapid hand motion causing incomplete sensor data, whereas the reduced-DoF estimation results in a natural approximation of the user’s hand. This illustrates the benefit of performing the optimization in a reduced parameter space based on natural hand synergies.

Our tracking system currently runs at approximately 10–13 fps with full-DoF tracking and approximately 14–17 fps with (6 + 6)-DoF tracking in PC-space on a Quad Core Intel Xeon E5530 2.4 GHz CPU with 4 GBs of RAM.

In our hand tracking approach, the hand posture is estimated using positional information from a Kinect point cloud as geometric constraints to fit a virtual hand model by means of inverse kinematics. The extension of the method to allow for optimization in a dimensionally reduced PC-space is simple and serves to constrain the parameter space in a meaningful way. Using a varied set of motion capture data as ground truth facilitated the derivation of natural hand synergies that covered a wide range of natural hand movements.

Optimizing in a reduced PC-space as opposed to the high-dimensional hand posture space improves runtime performance and prevents implausible hand postures by constraining the estimation to realistic postures. These constraints can reduce the mobility of the posture estimation to some extent, but the overall tracking greatly benefits from the increase in stability and plausibility of the estimated hand postures.

The number of principal components needed to cover most of the variance in the data depends on the number and type of movements contained in the data set. The less varied the captured movements are, the less parameters are needed to represent the most meaningful hand synergies involved. This simplifies the problem of tracking specific movements, such as various types of grasping, to an optimization of only 2 or 3 posture parameters, but impairs the approximation of more general hand movements. To overcome the fact that the low-DoF estimation is limited by the postures contained in the ground truth data, we plan to implement a hierarchical optimization scheme that allows for incremental local fine-tuning of successively added posture parameters.

The accuracy of the approximation of the user’s hand posture increases when the dimensions of the virtual model closely match those of the user’s hand. We are investigating a calibration procedure that automatically adjusts the virtual model’s structural parameters, such as scale and finger segment lengths, to arbitrary hands.

Recent short-range sensors such as the Creative Interactive Gesture Camera<sup>1</sup> or the Leap Motion Controller<sup>2</sup> present promising alternatives to the Kinect camera, as they were designed with highly accurate gestural interaction in mind. Using these sensors in our hand tracking system could be beneficial to the posture estimation accuracy and may open up many application avenues.

## ACKNOWLEDGMENT

This work was supported by the DFG Center of Excellence “Cognitive Interaction Technology” (CoE 277: CITEC) and the DFG grant “Real-Time Acquisition and Dynamic Modeling of Human Faces, Upper Bodies, and Hands” (BO 3562/1-1).

<sup>1</sup>[click.intel.com/intelsdk](http://click.intel.com/intelsdk)

<sup>2</sup>[leapmotion.com](http://leapmotion.com)

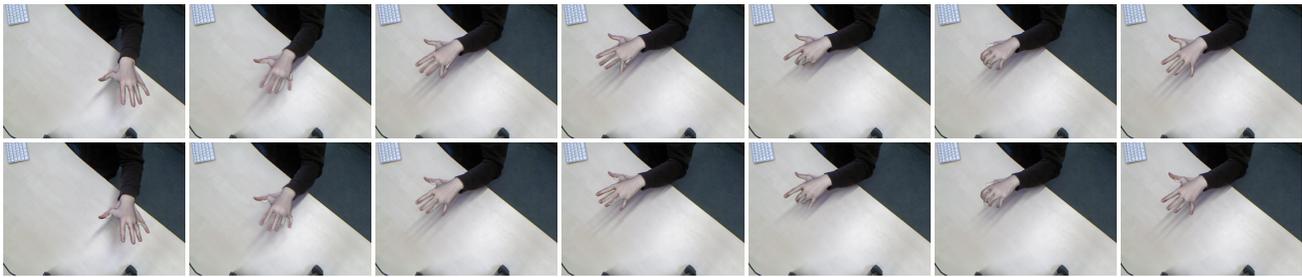


Fig. 8. Example hand tracking sequence. Top row: full-DoF posture estimation. Bottom row: reduced-DoF posture estimation. The full-DoF posture estimation fails to correctly track the posture during a rapid hand movement. The ring and pinky fingers collapse into the same point cloud segment, resulting in an unnatural posture reconstruction. The reduced-DoF posture estimation has less mobility but maintains a plausible hand posture reconstruction across the whole sequence.

## REFERENCES

- [1] J. Romero, H. Kjellström, and D. Kragic, "Hands in action: real-time 3D reconstruction of hands in interaction with objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 458–463.
- [2] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," in *ACM Transactions on Graphics SIGGRAPH 2009*, pp. 63:1–63:8.
- [3] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011, pp. 2088–2095.
- [4] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *European Conference on Computer Vision (ECCV)*, Firenze, Italy, October 2012, pp. 640–653.
- [5] M. Schroeder, C. Elbrechter, J. Maycock, R. Haschke, M. Botsch, and H. J. Ritter, "Real-time hand tracking with a color glove for the actuation of anthropomorphic robot hands," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Osaka, Japan, 2012.
- [6] H. Hamer, K. Schindler, E. Koller-Meier, and L. V. Gool, "Tracking a hand manipulating an object," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 1475–1482.
- [7] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *22nd British Machine Vision Conference (BMVC)*, 2011.
- [8] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1862–1869.
- [9] N. Bernstein, *The Co-ordination and Regulation of Movements*. Pergamon Press Ltd., 1967.
- [10] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use," *Journal of Neuroscience*, vol. 18, pp. 10 105–10 115, 1998.
- [11] A. Daffertshofer, C. J. Lamoth, O. G. Meijer, and P. J. Beek, "Pca in studying coordination and variability: A tutorial," *Clinical Biomechanics*, vol. 19, no. 4, pp. 415–428, 2004.
- [12] M. C. Tresch, V. C. Cheung, and A. d'Avella, "Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets," *Journal of Neurophysiology*, vol. 95, no. 4, pp. 2199–2212, 2006.
- [13] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 3270–3275.
- [14] A. Bicchi, M. G. M., and M. Santello, "Modelling natural and artificial hands with synergies," *Phil. Trans. R. Soc. B*, vol. 366, pp. 3153–3161, 2011.
- [15] M. Gabbicini, A. Bicchi, D. Prattichizzo, and M. Malvezzi, "On the role of hand synergies in the optimal choice of grasping forces," *Autonomous Robots*, vol. 31, pp. 235–252, 2011.
- [16] M. Popovic and D. Popovic, "Cloning biological synergies improves control of elbow neuroprostheses," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 1, pp. 74–81, 2001.
- [17] S. D. Iftime, L. L. Egsgaard, and M. B. Popovic, "Automatic determination of synergies by radial basis function artificial neural networks for the control of a neural prosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 4, pp. 482–489, 2005.
- [18] R. Vinjamuri, M. Sun, C. C. Chang, H. N. Lee, R. J. Scلابassi, and Z. H. Mao, "Temporal postural synergies of the hand in rapid grasping tasks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 4, pp. 986–994, 2010.
- [19] J. Lee and T. Kunii, "Model-based analysis of hand posture," *IEEE Computer Graphics and Applications*, vol. 15, pp. 77–86, 1995.
- [20] Y. Wu, J. Y. Lin, and T. S. Huang, "Capturing natural hand articulation," in *IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 426–432.
- [21] M. Kato, Y.-W. Chen, and G. Xu, "Articulated hand motion tracking using ICA-based motion analysis and particle filtering," *Journal of Multimedia*, pp. 52–60, 2006.
- [22] D. Jacka, A. Reid, and B. Merry, "A comparison of linear skinning techniques for character animation," in *In Afrigraph*. ACM, 2007, pp. 177–186.
- [23] "Microsoft Corp. Redmond WA. Kinect for Xbox 360."
- [24] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, Tech. Rep., 2004.
- [25] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.
- [26] J. Maycock, D. Dornbusch, C. Elbrechter, R. Haschke, T. Schack, and H. Ritter, "Approaching manual intelligence," *Künstliche Intelligenz – Issue Cognition for Technical Systems*, pp. 287–294, 2010.
- [27] J. Maycock, J. Steffen, R. Haschke, and H. Ritter, "Robust tracking of human hand postures for robot teaching," in *IEEE/RSJ Int. Conf. on Intel. Robots and Systems (IROS)*, 2011, pp. 2947–2952.