# Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data

Srinath Sridhar, Antti Oulasvirta, Christian Theobalt
MPI Informatik and Saarland University
{ssridhar,oantti,theobalt}@mpi-inf.mpg.de

## Abstract

*Tracking the articulated 3D motion of the hand has important applications, for example, in human–computer interaction and teleoperation. We present a novel method that can capture a broad range of articulated hand motions at interactive rates. Our hybrid approach combines, in a voting scheme, a discriminative, part-based pose retrieval method with a generative pose estimation method based on local optimization. Color information from a multi-view RGB camera setup along with a person-specific hand model are used by the generative method to find the pose that best explains the observed images. In parallel, our discriminative pose estimation method uses fingertips detected on depth data to estimate a complete or partial pose of the hand by adopting a part-based pose retrieval strategy. This part-based strategy helps reduce the search space drastically in comparison to a global pose retrieval strategy. Quantitative results show that our method achieves state-of-the-art accuracy on challenging sequences and a near-realtime performance of* $10$ *fps on a desktop computer.*

## 1. Introduction

Interactive markerless tracking of *articulated hand motion* has many applications in human–computer interaction, teleoperation, sign language recognition, and virtual character control among others. *Marker* or glove-based solutions exist for tracking the articulations of the hand [25], but they constrain natural hand movement and require extra user effort. Recently, many commercial sensors have been developed that detect 3D fingertip locations without using markers but these sensors do not recover a semantically meaningful skeleton model of the hand.

In this paper we describe a novel *markerless* hand motion tracking method that captures a broad range of articulations in the form of a *kinematic skeleton* at near-realtime frame rates. Hand tracking is inherently hard because of the large number of degrees of freedom (DoF) [9], fast motions, self-occlusions, and the homogeneous color distribution of skin. Most previous realtime markerless approaches (see Section 2) capture slow and simple articulated hand motion since reconstruction of a broader range of complex motions requires offline computation. Our algorithm follows a hybrid approach that combines a generative pose estimator with a discriminative one (Figure 1). The input to our method are RGB images from five calibrated cameras, depth data from a monocular time-of-flight (ToF) sensor and a user-specific hand model (Section 3). The output of our method are the global pose and joint angles of the hand represented using 26 parameters.

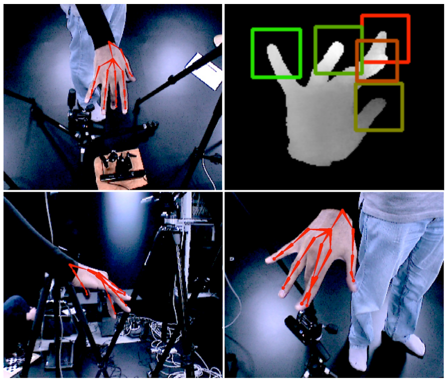

Figure 1. Our approach combines two methods (a) Generative pose estimation on multiple RGB images using local optimization (bottom row and top left) (b) Part-based pose retrieval on five finger databases indexed using detected fingertips (top right).

Our approach is informed by the robustness and accuracy of recent hybrid methods for realtime full-body tracking [2]. However, using the same strategy for hand tracking is challenging because of the absence of sufficiently discriminating image features, self-occlusions caused by fingers, and the large number of possible hand poses.

Figure 2 gives an overview of our algorithm. Similar to previous work in full-body motion tracking [2, 26, 29] we instantiate two pose estimators in parallel. First, the gener-

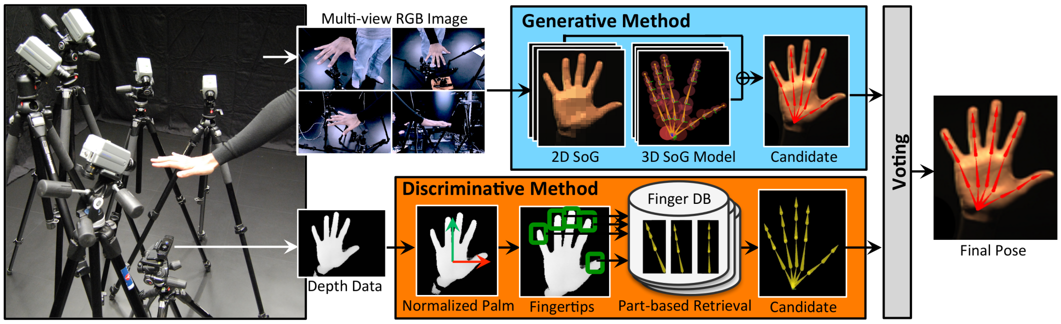

Figure 2. Overview of our approach. SoG stands for Sum of Gaussians

ative pose estimator uses local optimization and a similarity metric based on the Sum of Gaussians (SoG) model [23] to find the pose that best explains the input RGB images (Section 4). Second, the discriminative pose estimator, the key technical contribution of this paper, is a *part-based retrieval technique* that allows us to recover poses spanning a large hand articulation space while dealing with self-occlusions. Our discriminative pose estimation method first detects fingertips on depth data using a linear SVM classifier (Section 5.3). The detected fingertips are then used in a hypothesize-and-test framework along with five finger pose databases to obtain multiple pose hypotheses, each of which is tested using two criteria (Section 5.4). The final (complete or partial) hand pose is the pose that has the least error between the estimated and observed fingertip positions. This is then used as initialization for local optimization in the generative pose estimator. This part-based approach reduces the database size dramatically as only the articulations of each finger need to be indexed. The evidence from both pose estimators are fused using an error metric to obtain a final hand pose (Section 6).

To critically assess our method, we report evaluations using challenging, kinesiologically motivated datasets. While there are numerous benchmark datasets for full-body pose estimation, we know of none for hand motion tracking. We therefore created seven annotated datasets recorded using multiple calibrated sensors. The motions cover the full abduction–adduction and flexion–extension ranges of hand. Quantitative results show that we can cover a broad range of motions with an average error of around 13 mm. Our approach compares favorably in terms of accuracy and computational cost to a previous state-of-the-art approach [14]. To sum up, the primary contributions of this paper are:

- A hybrid approach that combines a generative pose estimator based on local optimization with a novel part-based pose retrieval strategy.
- A near-realtime framework that captures hand motions with a level of precision and speed necessary for interactive applications.
- An extensive, annotated benchmark dataset consisting of general hand motion sequences.

## 2. Related Work

One of the first kinematics-based hand motion tracking methods was presented by Rehg and Kanade [18]. The first study of size of the motion space of hand articulations when using kinematic skeletons was done by Lin *et al*. [11, 28]. They identified three types of constraints: joint angle limits (type I), intra-finger constraints (type II) and naturalness of hand motion (type III). Subsequent surveys of vision-based hand tracking methods [5] have divided methods into two categories—generative methods based on local or global optimization and discriminative methods based on learning from exemplars or exemplar pose retrieval.

**Generative Methods**: Oikonomidis *et al*. [14] presented a method based on particle swarm optimization for full DoF hand tracking using a depth sensor. They reported a frame rate of 15 fps with GPU acceleration. Other generative approaches have been proposed that use objects being manipulated by the hand as constraints [7, 8, 15, 19]. One such approach by Ballan *et al*. [3] used discriminatively learned salient features on fingers along with edges, optical flow, and collisions in an optimization framework. However, this method is unsuitable for interactive applications due to its large computation time. Other model-based global optimization approaches suffer from the same runtime performance problem [12, 22].

**Discriminative Methods**: A method for 3D hand pose estimation framed as a database indexing problem was proposed by Athitsos and Sclaroff [1]. Their method used a database of 26 hand shapes and a chamfer distance metric to find the closest match of a query in the database. The idea of using a global pose retrieval from a database of hand poses was explored by Wang *et al*. [24, 25]. However, in order to cover the whole range of hand motions the size of the database required would be large. Keskin *et al*. [10] proposed a method for hand pose estimation by hand part labeling but not as a kinematic skeleton.

**Full-Body Motion Tracking**: Given the similarity, volume, and success of existing research in full-body tracking, it would be natural to adopt one of those techniques for hand motion tracking. Several methods produce a 3D mesh and/or kinematic skeleton as their output [13, 17]. Some techniques such as Stoll *et al*. [23] rely on multiple RGB cameras while many others use depth information from time-of-flight (ToF) or structured light depth cameras [2, 6, 20]. However, direct application of these methods to hand tracking is not straightforward because of homogeneous skin color, fast motions, and self-occlusions.

Our approach takes inspiration from hybrid approaches to full-body pose estimation, such as Ye *et al*. [29], Baak *et al*. [2], and Wei *et al*. [26]. However, our discriminative pose estimator uses a *part-based* pose retrieval technique as opposed to global pose retrieval.

## 3. Input Data and Hand Modeling

Figure 2 shows our setup consisting of multiple RGB cameras and a monocular ToF depth sensor. The image data from RGB cameras provides high visual accuracy for tracking. The complementary single-view depth data helps us to retrieve poses effectively, as we can resolve depth ambiguities and detect fingertip features in the 2.5D data. Retrieval efficiency is also supported by having to consider monocular image data only.

**RGB Images**: We use multiple, synchronized, and calibrated cameras to obtain RGB image data. We position $n_k$ cameras in an approximate hemisphere such that typical hand motions within this hemispherical space would be visible in multiple cameras. All cameras are calibrated to obtain both the intrinsic and extrinsic parameters. We denote the RGB image produced by each camera as $I_r^k$. In all our experiments we used five Sony DFW-V500 cameras set at a resolution of $320 \times 240$ and a frame rate of 30 fps.

**Depth Data**: The other input to our method comes from a single time-of-flight (ToF) depth camera. The ToF camera is placed such that the hand motion space is within its range and is extrinsically calibrated along with the RGB cameras. We denote the depth image produced by the ToF camera as $I_d$ and the unprojected point cloud representation of the scene as $C_d$. We used the Creative Interactive Gesture Camera as our ToF depth data sensor.

**Hand Modeling**: In order to capture the articulations of the hand we model it as a kinematic chain consisting of 32 joints (see Figure 4). We model the 26 degrees-of-freedom (DoF) of the hand using parameters $\Theta = \{\theta_i\}$, where $0 \leq i \leq 25$ (20 joint angles, 3 global rotations, and 3 global translations). Each joint angle is limited to a fixed range, $\theta_i \in [l^i_{min}, l^i_{max}]$, taken from studies of the hand [21]. Since we use a SoG model based generative tracking approach we also augment the kinematic skeleton with 30 uniform 3D Gaussians with a fixed mean, variance, and color (c.f. [23]). Finally, we attach a 3D mesh, $\mathcal{M}$, consisting of 1774 vertices to the skeleton. The final output of our method are the parameters $\Theta$ of the kinematic skeleton.

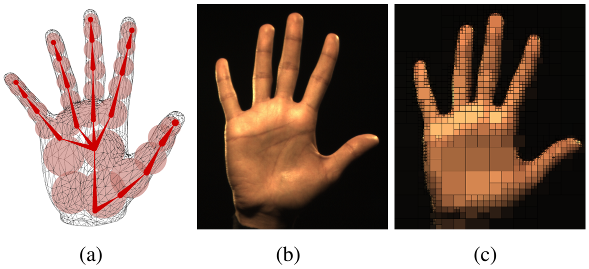

Figure 3. (a) Hand model consisting of a kinematic skeleton, attached 3D Gaussians (with radius set to the Gaussian variance for illustration), and a mesh. (b, c) Quadtree clustering of image into 2D SoG.

## 4. Generative Hand Pose Estimation

Generative tracking estimates the hand pose parameters $\Theta_G$ that best match a given set of $n_k$ input RGB images according to a consistency energy. We adopt a local energy maximization approach similar to that of Stoll *et al*. [23] but modified to account for hand motions which are different from full-body motion. In this approach both the hand and the input measurements are modeled using a *Sum of Gaussians* (SoG) representation. SoGs are mathematically smooth, yield analytical expressions for the energy functional and its derivative thereby enabling fast optimization. Our consistency energy is given as

$$\mathcal{E}(\Theta) = E(\Theta) - w_l E_{lim}(\Theta), \qquad (1)$$

where $E(\Theta)$ is a model-to-image similarity measure (Section 4.1). The second term, $w_l E_{lim}(\Theta)$, is a soft constraint on skeleton joint limits and has the same formulation as Stoll *et al*. The weight parameter $w_l$ was set to be 0.1 in all of our experiments.

### 4.1. Model-to-Image Similarity Measure

Given a 3D SoG based model of the hand and multiple input RGB images, we want to have a measure of similarity between the model and the images. We approximate each image with a 2D SoG model by performing quadtree clustering into regions of similar color, and fitting a 2D Gaussian with an average color to each region (Figure 3). Given two $2D$ SoGs $\mathcal{K}_a$ and $\mathcal{K}_b$ with associated colors $\mathbf{c}$, their similarity is defined as [23],

$$\begin{aligned}
&E(\mathcal{K}_a, \mathcal{K}_b)\\
&\quad = \int_\Omega \sum_{i\in\mathcal{K}_a} \sum_{j\in\mathcal{K}_b} d(\mathbf{c}_i, \mathbf{c}_j)\mathcal{B}_i(\mathbf{x})\mathcal{B}_j(\mathbf{x})\,\mathrm{d}\mathbf{x}\\
&\quad = \sum_{i\in\mathcal{K}_a} \sum_{j\in\mathcal{K}_b} E_{ij},
\end{aligned} \tag{2}$$

where $\mathcal{B}(\mathbf{x})$ is a Gaussian basis function

$$\mathcal{B}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mu\|^2}{2\sigma^2}\right). \tag{3}$$

$E_{ij}$ is the similarity between a pair of Gaussians $\mathcal{B}_i$ and $\mathcal{B}_j$ given their colors $\mathbf{c}_i$ and $\mathbf{c}_j$ and is defined as

$$\begin{aligned}
E_{ij} &= d(\mathbf{c}_i, \mathbf{c}_j) \int_\Omega \mathcal{B}_i(\mathbf{x})\mathcal{B}_j(\mathbf{x})\,\mathrm{d}\mathbf{x}\\
&= d(\mathbf{c}_i, \mathbf{c}_j) 2\pi \frac{{\sigma_i}^2{\sigma_j}^2}{{\sigma_i}^2+{\sigma_j}^2} \exp\left(-\frac{\|\mu_i-\mu_j\|^2}{{\sigma_i}^2+{\sigma_j}^2}\right).
\end{aligned} \tag{4}$$

The color similarity function $d(\mathbf{c}_i, \mathbf{c}_j)$ measures the Euclidean distance between $\mathbf{c}_i$ and $\mathbf{c}_j$ in the HSV color space and feeds the result into a Wendland function [27]. This renders $d$, a smooth function bounded in $[0, 1]$ (0 for dissimilar input and 1 for similar input).

Using the above defined similarity measure, we can find how similar a particular pose of the 3G SoG hand model is to the observed RGB images. To this end, the 3D Gaussians are projected onto the images using a projection operator $\Psi(\mathcal{K}_m)$ [23]. We now define the final similarity measure as

$$\begin{aligned}
&E_{sim}(\mathcal{K}_I, \mathcal{K}_m(\Theta))\\
&\quad = \sum_{i\in\mathcal{K}_I} \min\left(\left(\sum_{j\in\Psi(\mathcal{K}_m)} w_j^m E_{ij}\right), E_{ii}\right),
\end{aligned} \tag{5}$$

where $w_j^m$ is a weighting factor for each projected 3D Gaussian $\Psi(\mathcal{K}_m)$. With this parameter we control the relative influence of each 3D Gaussian on the final similarity.

To prevent overlapping projected 3D Gaussians from contributing multiple times in the above sum and distorting the similarity function, we clamp the similarity to be at most $E_{ii}$, which is the similarity of the image Gaussian with itself. This can be seen as a simple approximation of an occlusion term.

The offline step in this optimization method is to perform person-specific customization of the hand model's shape and dimensions, once for each actor. We adopt the *semi-automatic* process described by Stoll *et al.* [23] to our default hand skeleton template. We captured four static hand poses in which joints were clearly visible, and manually positioned our default hand skeleton to fit the poses. After this step, the position, variance, and color of the 3D Gaussians and bone lengths are optimized. This hand model is used throughout in all stages of our method.

### 4.2. Optimization

The goal of the optimization step is to estimate the pose parameters $\Theta_t$ at every time instant. We adapted the gradient ascent local optimization method proposed by Stoll *et al.* which enables realtime estimation of the pose parameters at every time instant $t$, as analytical gradients can be computed for our energy function. Each iteration of the optimization is initialized by extrapolating the estimated pose from two previous times steps as

$$\Theta_0^t = \Theta^{t-1} + \alpha(\Theta^{t-1} - \Theta^{t-2}), \tag{6}$$

where $\alpha$ is set to $0.5$. In Section 5, we describe how our part-based pose retrieval strategy can be used to initialize the optimization.

Even though the generative pose optimization method is fast and proven to be reliable for full-body tracking, it quickly reaches its limits during hand tracking and fails by converging to local pose optima from which it cannot recover. This is because the hand exhibits a higher articulation complexity than the body (thus allowing for a much wider range of poses in a small space), faster motions, and homogeneous color. The consequences are frequent self-occlusions and large visible displacements of the hand between two frames which challenge a local pose optimizer. Furthermore, the uniform skin color of the bare hand makes model-to-image associations much more ambiguous than in the case of humans wearing colored clothing. We therefore complement our generative tracker with an efficient discriminative hand pose estimation algorithm described in the following sections. It generates hand pose hypotheses in parallel to the generative method and is able to re-initialize it in case of convergence to a wrong pose.

## 5. Part-based Pose Retrieval

The goal of our discriminative pose estimation method is to estimate a complete or *partial* pose, $\widetilde{\Theta}_D$, of the hand from a single depth image $I_d$. We do this by adopting a part-based strategy *i.e.* instead of trying to recover the full hand pose, we separately recover the pose of each finger $\Theta_D^f$. This is achieved by extracting fingertips on the depth image using a linear SVM classifier, and by using the detected positions to find the closest match in multiple exemplar *finger pose databases*. Having separate databases for each finger has several advantages. First, for combinatorial reasons, the articulation space that we are able to represent in a pose database of necessarily limited size is much larger than when using a single pose database with exemplars for the entire hand (Section 5.1). Second, our approach has the advantage of being able to recover a partial hand pose (*i.e.* missing some finger poses) even when some of the fingers are occluded. The recovered finger poses are then assembled using a hypothesize-and-test framework to form a complete or partial pose $\widetilde{\Theta}_D$.

### 5.1. Multiple Finger Pose Database Generation

We briefly motivate the need for using multiple finger databases as opposed to a single global pose database. The global pose retrieval method of Wang and Popović [25] uses $18,000$ poses sampled from real hand motion. Although one of their goals was to avoid oversampling, the size of their database is still insufficient to span the range of articulations that can occur in natural motion. One way to quantitatively assess the relationship between the range of articulations and the size of the database is to consider discretizations of joint angles within allowable joint limits. Ignoring global motion, we model the hand using 21 joint angles (DoFs). If each joint angle were discretized into 3, then for global pose retrieval the size of the database would be of the order of $10^{10}$. On the other hand, part-based pose retrieval would need five databases, each with a size of $81$. Thus, part-based pose retrieval results in much smaller databases for the hand than global pose retrieval. This prevents oversampling while still keeping the articulation space large.

Previous approaches [2, 25] that use global pose retrieval capture real data using motion capture systems for generating a pose database. However, complex hand motions are difficult to capture using mocap systems because of self-occlusions and glove constraints. We therefore obtain our finger pose database by synthetically generating the poses over discretizations of all joint angles for each finger. To this end we use the person-specific model of the hand obtained earlier (Section 4.1)

For each synthetic pose generated per finger, $\Theta_S^f$, we compute the end effector position $\mathbf{x}_s^f$ with respect to a local skeleton coordinate system (see Section 5.2). We use the computed 3D end effector position as our database indexing feature since it uniquely identifies a pose of the finger and can be detected comparatively easily on depth data. We use a $k$-d tree for indexing the features. In all our experiments we used a database size of $4096$ corresponding to a joint discretization of $8$ levels per DoF.

### 5.2. Palm and Hand Orientation Estimation

Since our finger pose databases are indexed based on features relative to the hand model, we need to normalize the detected query features so that they lie in the same frame of reference. To this end, we extract the palm and its orientation from the depth data. We first apply a box filter on the depth image $I_d$ to extract the depth image, $I_b$, and unprojected point cloud, $C_b$, corresponding to the hand only. We use the morphological operations erode and dilate on $I_b$ to remove fingers but retain the palm. The result is a binary mask of the palm which is used to obtain a basic segmented point cloud of the palm, $C_s$. However, $C_s$ might contain fingers that lie on the line of sight between the sensor and the palm. We therefore fit a plane, $P$, to $C_s$ using RANSAC with a consensus threshold of 5 mm to obtain the final segmented point cloud of the palm, $C_p$. We compute the center of the palm as the point that lies on $P$ and is the centroid of the axis aligned bounding box of $C_p$. We then perform principal component analysis (PCA) of $C_b$ projected onto the plane $P$ to find the principal directions of the hand and palm. As a final step, we use a Kalman filter in order to reduce jitter in the estimated orientation. The detected palm center and orientations serve to stabilize the results of the finger pose database look up (see Figure 4).

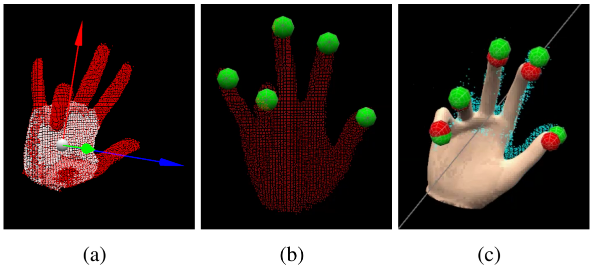

Figure 4. (a) Palm extracted from the point cloud (white) and hand orientation normalization (arrows). (b) Fingertips detected using a linear SVM classifier. (c) Estimated partial or complete hand pose.

### 5.3. Fingertip Detection

For our part-based pose retrieval strategy, we need to reliably detect the end effector positions in the depth data. Previous work in full-body pose estimation has used features such as Geodesic extrema [2, 16] which do not work well for the hand and result in spurious extrema which are difficult to disambiguate from the real extrema. In order to overcome this problem, we use a machine learning approach to detect fingertips using a linear SVM classifier and HOG descriptors as features. We follow the object detection framework of Dalal and Triggs [4] on depth images instead of RGB images. For training our linear SVM we used a combination of manually annotated real sequences, annotated synthetic sequences, and rotated versions of both (4 orientations). We use a fingertip detection window size of $32 \times 32$. Because of the high cost of not detecting a fingertip in the pose retrieval step we adjusted the parameters of the linear SVM for higher recall rates. We found that most false positives could be eliminated using assumptions about the position of the finger *i.e.* a fingertip cannot lie far away or too close to the center of the palm. After elimination, we obtain five or less fingertip candidate points $\mathbf{x}_c^f$. Figure 1 shows one depth frame with detected fingertips overlaid and Figure 4 shows the filtered fingertips on the point cloud.

### 5.4. Finger Pose Estimation

The final step of discriminative pose estimation is to find the complete or partial pose of the hand, $\widetilde{\Theta}_D$. However, in order to query the finger pose databases we would need to label each detected fingertip. This is a hard problem since there is tremendous variation in fingertip appearance in depth or RGB images. We instead adopt a hypothesize-and-test framework to test all elements in the set of permutations of labels, $\Sigma$, using two criteria. First, for each permutation $\sigma_i \in \Sigma$ we reject a hypothesized pose early based on the distance of each detected fingertip to the nearest neighbor in the finger pose database corresponding to the current labeling for that fingertip. We set a distance threshold $\mu = 20$ mm in all our experiments. Only those hypotheses that pass the first stage are tested with the distance measure which is given as

$$\delta(\sigma_i, \widetilde{\Theta}) = \frac{1}{r} \|\mathbf{x}_i - \mathbf{x}_c^f\|_2, \tag{7}$$

where $r$ is the number of detected fingertips, $\mathbf{x}_i$ is the position of a fingertip corresponding to a candidate fingertip $\mathbf{x}_c^f$ and $\widetilde{\Theta}$ is the current hypothesis pose. The pose that has the lowest distance measure is selected as the best pose $\widetilde{\Theta}_D$. In the case of less than five detected fingertip locations, a partial pose with the lowest distance is still recovered since partial poses are also part of the permutations set $\Sigma$.

## 6. Pose Candidate Fusion

At this stage, we have two hand pose candidates, $\Theta_G$ and $\widetilde{\Theta}_D$, from the generative and discriminative methods. In order to combine them together to find the best pose, we first initialize a second instance of the generative tracker with $\widetilde{\Theta}_D$ instead of extrapolation. Those pose parameters that are not part of $\widetilde{\Theta}_D$ are extrapolated using Equation 6. Upon optimization we obtain the pose $\Theta_D$ and an associated optima energy $\mathcal{E}(\Theta_D)$ (see Equation 1). The final pose, $\Theta_F$, is the pose that has the higher energy given by

$$\Theta_F = \underset{\Theta \in \{\Theta_G, \Theta_D\}}{\arg\max} \; \{\mathcal{E}(\Theta_G), \mathcal{E}(\Theta_D)\}. \tag{8}$$

## 7. Results

We implemented and tested our method, algorithmic variants of it, and a related algorithm from literature [14] on a computer with a clock speed of 3.30 GHz, 8 GB of RAM, and an Nvidia NVS 300 GPU. On this machine, our method achieved an interactive frame rate of 10 fps. With our unoptimized C++ code, the most time consuming components were the local optimization for generative pose estimation (53 ms) and multiscale fingertip detection (40 ms).

We will now present results from extensive experimental evaluation that we conducted using our method on a variety of sequences. Unlike previous approaches that used a combination of synthetic and real data for evaluation, we used a large corpus of real data. We collected seven real sequences consisting of synchronized and calibrated multi-view RGB images, as well as monocular ToF and Kinect data (see Figure 2). All sequences were manually annotated to mark fingertip and palm center positions in the depth data. In total, our test sequences consist of 2137 frames of data containing both slow and fast motions with a static background and general illumination conditions. The sequences that we captured span a range of hand movement from flexion–extension (*e.g.* `fingerwave`, `flexex1`, `pinch`, `fingercount`), abduction–adduction (*e.g.* `abdadd`), and included random motions (*e.g.* `random`, `fingerwave`).

Overall quantitative results from our experiments show that our approach of combining a generative pose estimation method with a discriminative *part-base pose retrieval* technique (**SoG + PBPFingertip**) performs better than other alternatives in most cases. Our algorithm is stable and all results were recorded using the same parameters. We compared our approach with several algorithmic alternatives—(a) generative pose estimation with SoG model only (**SoG**), (b) generative pose estimation method combined with a global pose retrieval technique based on normalized depth images (**SoG + GPImage**), and (c) publicly available implementation of the method proposed by Oikonomidis *et al.* [14] (**FORTH**, one sequence only).

**Evaluation Metric**: To enable relative comparison with other methods we adopted an error metric similar to that used by Oikonomidis *et al.* [15]. The Euclidean distance between the estimated and ground truth fingertip positions and palm center positions are computed for each frame for all datasets. We find the average error, $\widetilde{\Delta}$, over all frames within each dataset.

**Quantitative Results**: Figure 5 compares our result (SoG + PBPFingertips) with using only SoG and SoG + GPImage. Our hybrid method produces better results and achieves an accuracy of 13.24 mm on average which is close to the best offline methods [3]. This can be attributed to the fact that each time generative pose estimation fails, the discriminative part-base pose retrieval strategy re-initializes it appropriately. This is clear in Figure 6 which shows the error as a function of the frame number. The error starts accumulating in the generative method at about frame 25 and never goes down. But our method periodically re-initializes so as to maintain a constant error rate even in long sequences. Most notably, towards the end of the sequence our method produces errors that are not too different from the first few frames. One surprising result here is that SoG + GPImage produces a higher error than SoG only which indicates that image based global retrieval is sensitive to noise in the depth data.

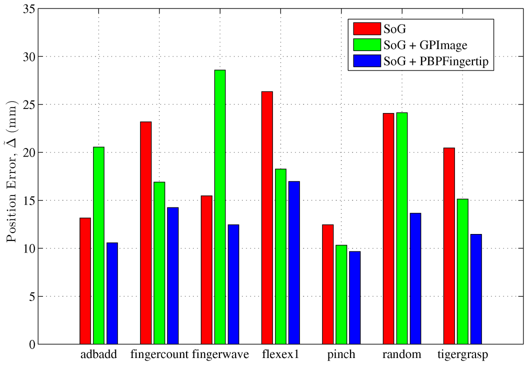

Figure 5. The average position error over the entire dataset for SoG only, SoG + GPImage and SoG + PBPFingertips (ours).

We also tested the FORTH method on a dataset containing motions similar to `fingerwave` but under different illumination conditions as we were unable to get their method to work on any one of our seven sequences. Their method is based on GPU acceleration and runs at only 2.5 fps compared to our 10 fps on the same machine indicating that optimization of our code could lead to faster frame rates. We then computed the error measure, $\widetilde{\Delta}$, for the FORTH method over the entire (similar) sequence and found it to be 10.31 mm. This compares favorable with the mean error of our method which was 13.24 mm. Thus, our method performs well for similar datasets while using less computational budget than FORTH.

## 8. Conclusion and Future Work

In this paper, we presented a novel method for tracking the articulated 3D motion of the human hand using a hybrid method. Our method advances the state-of-the-art by demonstrating high accuracy across a large corpus of motions with a frame rate that is sufficient for many interactive applications. Our main contribution was the use of a new method for part-based pose retrieval in conjunction with image-based pose optimization. Part-based pose retrieval enables recovery and stable tracking of poses with self-occlusions that are characteristic of hand motion, and enables a dramatic reduction of the pose database size.

Although our method achieves good performance on real sequences there is still room for improvement. Many of our datasets exhibit motion blur due to fast motions. Therefore, we would like to explore the use of new temporal priors along with high frame rate cameras. Our calibrated multi-camera setup requires time to setup. We would therefore like to explore reducing the number of cameras and incorporating depth data into generative pose optimization. Finally, we would like to achieve faster frame rates by using the parallel structure of generative pose optimization.

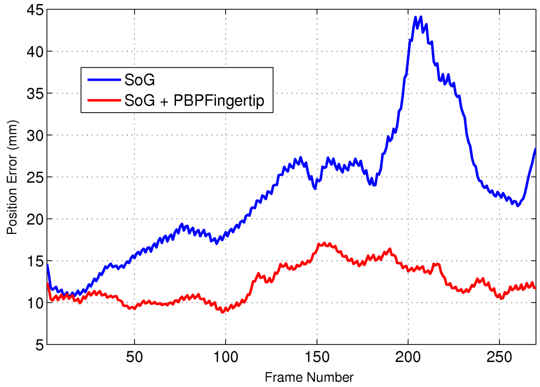

Figure 6. The average position error over the `fingerwave` dataset for SoG only and SoG + PBPFingertip (ours).

**Acknowledgments**: This work was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) and the ERC Starting Grant CapReal. We would like to thank Han Du for helping us with image annotation and James Tompkin for helping with the supplementary video voice-over. Our dataset is available at the project webpage[1].

## References

[1] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. CVPR*, volume 2, pages II – 432–9 vol.2, June 2003. 2

[2] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. ICCV*, pages 1092 –1099, Nov. 2011. 1, 3, 5

[3] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *Proc. ECCV*, volume 7577, pages 640–653. Springer Berlin / Heidelberg, 2012. 2, 6

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, volume 1, pages 886–893. IEEE, 2005. 5

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *CVIU*, 108(12):52–73, Oct. 2007. 2

[6] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-time human pose tracking from range data. In *Proc. ECCV*, volume 7577, pages 738–751. Berlin, Heidelberg, 2012. 3

[7] H. Hamer, J. Gall, T. Weise, and L. Van Gool. An object-dependent hand pose prior from sparse training data. In *Proc. CVPR*, pages 671–678, 2010. 2

[8] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *Proc. ICCV*, pages 1475–1482, 2009. 2

[9] L. A. Jones and S. J. Lederman. *Human Hand Function*. Oxford University Press, USA, 1 edition, Apr. 2006. 1

[1] `http://handtracker.mpi-inf.mpg.de`

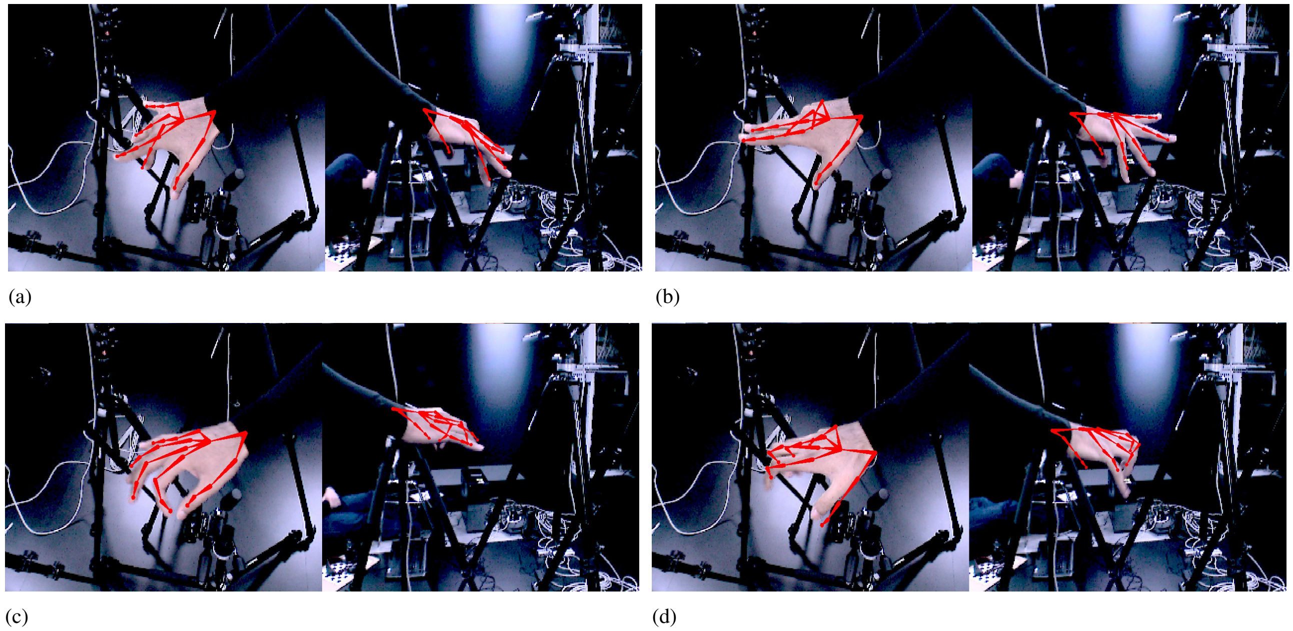

Figure 7. Qualitative results of our method as seen from two camera views. Results in (a), (b) show general slow motion. Results in (c) show successful tracking even in the presence of fast motion. Result (d) shows a failure case due to fast motion.

[10] C. Keskin, F. Kraç, Y. E. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proc. ECCV*, pages 852–863. Springer Berlin Heidelberg, 2012. 2

[11] J. Lin, Y. Wu, and T. Huang. Modeling the constraints of human hand motion. In *Proc. HUMO*, pages 121–126, 2000. 2

[12] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. ECCV*, number 1843, pages 3–19. Jan. 2000. 2

[13] T. B. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(23):90–126, Nov. 2006. 3

[14] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. In *Proc. BMVC*, pages 101.1–101.11, 2011. 2, 6

[15] I. Oikonomidis, N. Kyriazis, and A. Argyros. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Proc. ICCV*, pages 2088–2095, 2011. 2, 6

[16] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Realtime identification and localization of body parts from depth images. In *Proc. ICRA*, pages 3108–3113, May. 5

[17] R. Poppe. Vision-based human motion analysis: An overview. *CVIU*, 108(12):4–18, Oct. 2007. 3

[18] J. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *Proc. ECCV*, volume 801, pages 35–46. Springer Berlin / Heidelberg, 1994. 2

[19] J. Romero, H. Kjellstrom, and D. Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *Proc. ICRA*, pages 458–463, 2010. 2

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*, pages 1297 –1304, June 2011. 3

[21] E. Simo Serra. *Kinematic Model of the Hand using Computer Vision*. PhD thesis, Institut de Robòtica i Informàtica Industrial, 2011. 3

[22] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. 28(9):1372–1384, 2006. 2

[23] C. Stoll, N. Hasler, J. Gall, H. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *Proc. ICCV*, pages 951 –958, Nov. 2011. 2, 3, 4

[24] R. Wang, S. Paris, and J. Popović. 6D hands: markerless hand-tracking for computer aided design. In *Proc. ACM UIST*, pages 549–558. ACM, 2011. 2

[25] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM TOG (Proc. SIGGRAPH)*, (3):63:163:8, July 2009. 1, 2, 5

[26] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM TOG (Proc. SIGGRAPH Asia)*, 31(6), Nov. 2012. 1, 3

[27] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv Comput Math*, 4(1):389–396, Dec. 1995. 4

[28] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *Proc. ICCV*, pages 426 –432 vol.2, 2001. 2

[29] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3D pose estimation from a single depth image. In *Proc. ICCV*, pages 731–738, 2011. 1, 3