Abstract

Hand pose estimation from 3D sensor data matches a point cloud to a hand model, and has broad applications from gestural interfaces to scene understanding. We propose a novel scheme to index into a database of precomputed hand poses to initialize the match. Our index describes 2D hand silhouettes, which can be computed from either depth maps or standard video, in the form of simple yet expressive signatures. We compare signatures to each other through a new variant of the Earth Mover’s Distance that makes small distances in feature space correlate highly with those in pose space. We present a new technique that uses a depth sensor and a sensor glove to create databases of real images and ground-truth poses for both training and testing. We show state-of-the-art accuracy and speed for both gesture classification and joint-pose regression, even when comparing our 2D single-frame method with those that employ RGB-D features or multi-sensor inputs and report quantitative results.

1. Introduction

Tracking the detailed motions of a human hand with good accuracy and minimal intrusion would enable applications ranging from gestural interfaces and finger-spelling recognition to medical diagnosis, musical tutoring systems, remote surgery, or animation. Tracking hands with 3D input amounts to matching a point cloud and a hand model by optimizing some measure of fit between them. Finger motions that are fast when compared with typical sensor frame rates suggest viewing each data frame as a separate problem: What is the best estimate of a hand’s pose—wrist and finger-joint angles—given just the current point cloud? A popular approach to per-frame hand pose estimation is to use the current frame as an index into a pre-built database of (frame descriptor, hand pose) pairs: Use a descriptor of the input frame to find the database entry with the most similar descriptor, and return the associated hand pose as the estimate. This paper addresses several technical challenges within this approach.

First, our indices describe hand silhouettes, which have the advantage of being easy to compute from point clouds, and even from standard video imagery if hand and background look different enough. Thus, although we use 3D input for tracking, our descriptors are 2D, for added flexibility.

Second, we design both our descriptors and a dissimilarity measure between them so as to capture the main features relevant to matching hands, such as which finger matches which, or how well separated two adjacent fingers are. To this end, we segment a silhouette boundary into its main convexities and concavities using a measure of topological persistence to separate important features from irrelevant ones. This segmentation results into a variable-length descriptor we call a signature. We then measure the dissimilarity between two signatures by a variant of the Earth Mover’s Distance (EMD), a measure of the amount of work needed to transform one signature into the other. Our variant makes sure that implausible matches between fingers are discarded, and then modulates a measure of dissimilarity between the remaining matches in such a way that similar signatures tend to correspond to similar poses.

Third, instead of sampling the set of all hand poses finely, we use low-dispersion sampling to build a database that populates the space of all natural poses well, given a limited number of pairs one can afford to record and store. We build our database by recording point clouds with an RGB-D sensor.
while measuring true hand motions with an opto-
mechanical hand tracker. The resulting database
of real images paired with real hand configura-
tions automatically underrepresents unnatural hand
poses and requires no manual annotation—a labor-
intensive and quantitatively imprecise alternative.

Fourth, we describe several experiments on both
regression and classification tasks that show promis-
ing accuracy and speed even when we pit our method
based on 2D descriptors against those that employ
3D features or multiple sensors.

Section 2 reviews related work. Sections 3 and
4 describe a hand model and how we construct
databases. Sections 5 and 6 describe signatures and
the dissimilarity measure between them. Section 7
shows experiments and Section 8 concludes.

2. Related Work

With the introduction of affordable depth sensors,
methods that use strictly 2D data [11, 1] are quickly
being replaced by those that use depth [12, 14, 8, 9,
19]. Full body tracking methods [1-4] typically treat
the hand and wrist as a single, rigid object [14]. Un-
like multi-camera approaches [10, 15, 16] or meth-
doing that use colored gloves [21] or data-gloves [2]
or bands to identify the wrist [12], we use a single depth
camera and no markers. Erol et al. provide a general
literature review [4].

Ren et al. represent the hand using finger seg-
ments from a silhouette [12] while Sridhar et al.
use a sum of Gaussians [15] and later a sum of
anisotropic Gaussians model [16]. Several recent ap-
proaches [8, 19, 18] build on the success of full body
tracking methods using random forests [14]. Sridhar
et al. search five separate finger databases for finger
articulation to reduce the database size [15]. Others
[19, 18] use a training set to learn a map from input to
pose and dispose of the need for a runtime database,
but are limited by the form of the map.

Tracking based on motion models [4, 10, 9, 15,
16] fails in the presence of fast hand motion [4] and
is subject to drift [18]. Gradient descent (Stoll et al.
[17]), Particle Swarm Optimization [10], inter-
polation [20], temporal and kinematic constraints [9]
or specific hand-assumptions [10, 19] are sometimes
used to improve initial guesses. Wu et al. use a
CyberGlove to learn hand-motion constraints [22].

Our work is most closely related to that of Ren et al.
[12] in our use of hand silhouettes, discrete de-
scriptors, and EMD to address the problems above.
However, we use topological persistence as a robust
method to detect and describe segments, and our
variant of the EMD accounts for matches between
fully and partially extended fingers, and for cases
where small differences in hand pose correspond to
different categories in classification tasks.

3. Hand Model and Pose Distance

We describe the pose of a hand with a vector \( \chi \)
that collects 6 degrees of freedom (DoF) for wrist
rotation/translation plus either 21 angles (\( \chi^o \), in
degrees) or 60 position coordinates (\( \chi^{mm} \), in mm) for
the 15 joints of a hand (three joints per finger) and 5
fingertips (Figure 1). The thumb has 5 angular DoF
in our model: two for flexion and abduction of the
trapeziometacarpal (TM) and metacarpophalangeal
(MCP) joint and one flexion of the interphalangeal
(IP) joint. Each of the other fingers has 4 DoF:
one flexion angle for distal interphalangeal (DIP)
and proximal interphalangeal (PIP) joint, and two for
flexion and abduction of the MCP joint. In our exper-
iments we keep the carpometacarpal (CMC) joints
and the fingertips fixed. We label the thumb, index,
middle, ring, and little fingers as \( T, I, M, R, \) and \( L \).
We use forward and inverse kinematics to convert be-
tween angles and positions.

We measure pose distance between frames \( \lambda \) and
\( \lambda' \) as the average Euclidean distance over all joints
of interest, using either angles or positions (not both) as
needed to compare with existing literature [10]:

\[
d_{\chi, \mathcal{J}}(\lambda, \lambda') = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} ||\chi_j - \chi'_j||_2
\]

where \( \mathcal{J} \) is the set of parameters of interest.

4. Database

We assume that the space of natural hand poses
is much smaller than that of all possible poses [1],
and create a database of \( 320 \times 240 \) RGB-D images
recorded with an Intel DepthSense 325 sensor. To
avoid the need for manual annotation, which is labor
intensive and quantitatively imprecise, we simulta-
nuously record poses with a CyberGlove III sensor
glove that records 23 joint angles every 11 ms, cali-
bruired and mapped to our model (Figure 1) by stan-
dard methods [5, 6]. We synchronize the two sensors
counting time from an initialization motion recogniz-
able in both sensors: ‘fist’, ‘open hand’, ‘fist.’

We asked a single subject to assume 75 prede-
defined poses of their right hand (examples in Figure 3)
and then add many examples of random motion. We
keep the elbow fixed and ask the subject to move the wrist through 3 abduction/adduction angles, at each of which the subject undergoes a set of complete flexion/extension motions of the wrist. Our initial set $\Omega$ has 14,230 (image, pose) pairs. In comparison, Wang and Popović use 18,000 samples of finger articulation (no wrist motion) [21].

Figure 1: Kinematic model and CyberGlove III sensors.

Low-dispersion sampling [21] is used on $\Omega$ to obtain a smaller database $\Lambda$ such that the distance $\min_{\lambda' \in \Lambda} d_{x^o}(\lambda, \lambda')$ between any sample $\lambda \in \Lambda$ and its nearest neighbor in $\Lambda$ is bounded from below. $\Lambda$ is initialized with the furthest apart pair of points in $\Omega$ then iteratively adds the sample from $\Omega$ furthest from its nearest point currently in $\Lambda$. We iterate until we run out of samples or $\max_{\lambda' \in \Omega} \min_{\lambda \in \Lambda} d_{x^o}(\lambda, \lambda')$ falls below a predefined $\epsilon$.

5. Silhouette Signatures

The convexities and concavities of outer boundaries of a hand’s silhouette capture information about the anatomical features of the hand—fingers, palm, wrist [12]. They are simple closed curves that can be made to be somewhat invariant to wrist motion and the geometry and appearance of individual hands. Silhouettes can be extracted from either color [7] or depth information, or both. We use depth, and assume that the hand is the closest object to the sensor and is well away from the background [12, 18]. If $D_{\text{min}}$ is the smallest depth in the image and $\tau_{R}$ is a bit larger than the length of a large hand, the hand is defined as the largest connected component of the pixels whose depth $D < D_{\text{min}} + \tau_{R}$.

Let $c$ and $r$ be the center pixel and radius of the maximum circle inscribed in the hand region. We trace the silhouette’s boundary starting from the leftmost silhouette pixel along the horizontal line through $c$ to obtain a closed polygon $B$ with $n$ points. We map pixels $p_s$ on $B$ to their polar coordinates relative to $c$ and divided by $r$ in norm and $2\pi$ in angle:

\[
x_s = (x_s, y_s) = (p_s - c)/r
\]

\[
\phi_s = \frac{\arctan2(y_s, x_s) + \pi}{2\pi}
\]

\[
\rho_s = \sqrt{x_s^2 + y_s^2}.
\]

For the index $s \in [0, \ldots, n - 1]$ we define $s \oplus z = (s+z) \mod n$ and $s \ominus z = s - z$. Centering and normalization account for variations in hand size, distance from the camera, and image position. We found $c$ to provide a more reliable reference than the hand’s centroid, which depends on finger pose. Polar coordinates can be used to yield invariance to 2D rotations in the image by using a standardized starting point—for example, the middle of the wrist.

To reflect changes in hand pose while being insensitive to skin or muscle deformations or noise, we decompose a boundary into a set of segments that separate its main convexities and concavities, as shown next. We then describe the resulting list of segments by a variable-length descriptor, our signature, and define a dissimilarity measure between signatures.

5.1. Boundary Decomposition

To decompose a boundary into segments, its pixels are swept in order of decreasing value of $\rho_s$ to determine one segment per local maximum of $\rho_s$. Boundary segments associated with maxima of low persistence [3] are merged with one of their neighbors. Persistence measures the lifetime of an extremum during the sweep [23]. We first describe the sweep, and then explain the role of persistence.

Boundary index $s$ precedes $s'$, iff either $\rho_s > \rho_{s'}$ or $(\rho_s = \rho_{s'}$ and $s < s')$, and we then write $s \succ s'$. Index $s$ is a local maximum of $\rho$ if $s \succ N_s$ and a local minimum if $N_s \succ s$ where $N_s = \{s \ominus 1, s \oplus 1\}$. Let $s_l$ be the index of the $l^{th}$ pixel encountered in the sweep, so that $s_0$ is the global maximum. Algorithm 1 sweeps the boundary pixels in the order $\succ$ and produces label $m_s$ for index $s$ if this index belongs to the segment associated with a local maximum at $m_s$.

Initially, all labels are unlaabeled (set to $-1$). The label $m_{s_l}$ of $s_l$ is updated depending on the number of elements in $N_{s_l}$ that have a valid label:
0 labels: $s_l$ is a local maximum, set $m_{s_l} \leftarrow s_l$.
1 label m: $s_l$ is a regular point, set $m_{s_l} \leftarrow m$.
2 labels: $s_l$ is a local minimum.

The two neighboring segments are kept distinct if their persistence is sufficiently high, or are merged otherwise. The minimum $s_l$ is given the label of the older segment. The predicate $\text{merge}(m_1, m_2)$ in Algorithm 1 is described next.

**Algorithm 1 Boundary Segmentation**

1. **Inputs:** $r = [\rho_0, \ldots, \rho_{n-1}], \tau_{\text{size}}, \tau_{\pi}, \tau_{\text{value}}$
2. **Output:** $m = [m_0, \ldots, m_{n-1}]$
3. $(r, s) = \text{sort}(r, \succ)$ \triangleright Sort $r$ by $\succ$. Also return sorted indices $s = [s_0, \ldots, s_{n-1}]$
4. for $i = 0$ : $(n - 1)$ do
   5. \hspace{1em} $m_{s_i} = -1$ \triangleright Initialize labels
   6. end for
7. for $l = 0$ : $(n - 1)$ do
   8. switch $|N_{s_l}| > -1$ do
      9. case 0: $m_{s_l} \leftarrow s_l$ \triangleright New local maximum
      10. case 1 (m): $m_{s_l} \leftarrow m$ \triangleright Regular point
      11. case 2: $N_{s_l} = \{m_1, m_2\}$ with $m_1 > m_2$
         \hspace{1em} $m_{s_l} \leftarrow m_1$
         12. if $\text{merge}(m_1, m_2, \tau_{\text{size}}, \tau_{\pi}, \tau_{\text{value}})$
           13. \hspace{1em} relabel all $m_2$ in $m$ to $m_1$
           14. end if
         15. end switch
     16. end for
17. return $m$

$\text{merge}(m_1, m_2, \tau_{\text{size}}, \tau_{\pi}, \tau_{\text{value}}) = \{ s : s = m_2 \} \leq \tau_{\text{size}} \lor (\rho_{m_2} - \rho_s \leq \tau_{\pi} \land \rho_{m_1} - \rho_{m_2} \leq \tau_{\text{value}})$

5.1.1 Relevant Segments

Persistence [3] helps distinguish between ephemeral local maxima in $\rho_s$ from those that are more likely to correspond to anatomical hand features such as fingers or knuckles. The persistence of a local maximum is the vertical distance between its birth and death. More precisely, a local maximum $s \in B$ is $\delta$-stable if there exist integers $a$ and $b$ with $-n < a < 0 < b < n$ such that (i) for all $z \in [a, b]$ other than $s$ we have $s > z \oplus z$ and (ii) $\rho_s \geq \max\{\rho_{z \oplus a}, \rho_{z \oplus b}\} + \delta$. The persistence of $s$ is then the maximal $\delta$ for which $s$ is $\delta$-stable.

We use persistence as follows. The two immediate neighbors of index $s_l$ at a local minimum belong to segments that are associated with two distinct local maxima, call them $m_1, m_2$ during the sweep. Assume that $m_1 > m_2$, so that $m_2$ is “younger” than $m_1$. Segment $m_2$ is merged into $m_1$ if $m_2$ is either insignificant in extent along the boundary, or both of the following conditions are met: the persistence of $m_2$ is too small and the radial coordinates of the two local maxima $m_1, m_2$ are too close to each other. This yields line 18 of Algorithm 1, where $\tau_{\text{size}}, \tau_{\pi}$, and $\tau_{\text{value}}$ are positive thresholds which we set to 5 pixels, .1 radii, and .1 radii in all our experiments.

The threshold $\tau_{\text{size}}$ removes very small segments. Of the remaining segments, highly-persistent ones are meant to represent at least partially extended fingertips. Segments that cover enough of the boundary and have a local maximum that is significantly lower than the maximum of an adjacent segment are retained with the intent to capture knuckles, or shorter fingers that touch longer ones.

A boundary signature is a concatenation of the descriptors for each segment found by Algorithm 1. Each segment is described by (i) the normalized angular coordinates $\phi_a$ and $\phi_b$ of its endpoints; (ii) the persistence $\pi_m$ of its local maximum $(\phi_m, \rho_m)$; and (iii) a weight $w = (\phi_b \ominus \phi_a)(\rho_m - 1)$ that approximates the area between the segment and the largest inscribed circle. The arm is typically the segment with the greatest weight $w$, which we remove from the signature. We also remove all segments whose normalized angle $\phi$ is within 0.2 from the wrist, because they are unlikely to represent fingers.

6. Comparing Signatures

We define a measure of dissimilarity between signatures $S = \{S_1, \ldots, S_m\}$ and $T = \{T_1, \ldots, T_n\}$ by (i) defining soft (that is, fractional) matches between segments in $S$ and segments in $T$ and then (ii) measuring the aggregate discrepancies between matched segments, as described next.

6.1. Soft Matches

We perform soft matching through a variant of the Earth Mover’s Distance (EMD, [13]), which solves a linear program to determine the smallest amount of work needed to transform the masses $w^{S_i}$ in $S$ into the masses $w^{T_j}$ in $T$ or vice versa. Work is the sum

$$\text{WORK}(S, T, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} \quad (5)$$

$\uparrow$ Since the persistence for $m_2$ is greater than or equal to $\rho_{m_2} - \rho_s$. 

104
where the matrix \( F \) of unknown flows \( f_{ij} \) must be nonnegative, have rows and columns that add up to \( w_i^S \) or \( w_j^T \), and have \( L_1 \) norm equal to the smaller of the signature masses \( w_i^S = \sum_{i=1}^{m} w_i^S \) and \( w_j^T = \sum_{j=1}^{n} w_j^T \). We set the ground distances \( d_{ij} \) in (5) to be equal to the angular distance \( \phi_{ij} \) used in EMPD [12]: zero when the two angular supports overlap fully and \( \min(|\phi_{a}^S \ominus \phi_{a}^T|, |\phi_{b}^S \ominus \phi_{b}^T|) \) otherwise.

Since it makes little sense for a segment in one signature to match one in the other with a very different angular support, we introduce a distance threshold \( \Delta \) that is of the order of half the angular extent of a finger width. We set \( \Delta = 0.06 (\approx 22^\circ) \) in all our experiments, finding the value non-critical. We then add extra segments \( S_0 \) and \( T_0 \) to signatures \( S \) and \( T \) with weights \( w_i^S = w_i^T = w_i^S \), and define

\[
d_{ij} = \begin{cases} 
\phi_{ij}, & i \neq 0, j \neq 0 \\
\Delta, & \text{otherwise}
\end{cases}
\]  

(6)

Flows between signatures such that \( \phi_{ij} > \Delta \) are thus shunted into the extra segments, because they incur less work, and matches between excessively distant segments are discarded, i.e., replaced by matches with the extra segments. Setting \( d_{i0} = \Delta \) prevents \( S_0 \) and \( T_0 \) from simply matching each other.

6.2. Signature Dissimilarity

The EMD provides an initial measure for the dissimilarities between signatures if modified as follows to account for the extra segments:

\[
\text{EMD}^*(S, T) = \frac{\sum_{i=1}^{m} \sum_{j=0}^{n} d_{ij} \hat{f}_{ij}}{\min\{\sum_{i=1}^{m} w_i^S, \sum_{j=1}^{n} w_j^T\}}
\]  

(7)

where the flows (soft matches) \( \hat{f}_{ij} \) minimize (5). The denominator does not include the extra segments, so finger segments in one signature are either matched to nearby segments in the other signature or discarded if no such match exists. The missing matches are still penalized by \( \Delta \) units per unit of flow.

Rather than using EMPD directly to measure signature dissimilarity, we introduce an additional cost \( C \) for the reasons that follow. Fractional matches between segments account properly for segmentation errors. In addition, a fractional match may represent a match between fingers that are stretched to different extents in the two hands. For instance, Figure 2 shows the signature of a hand with index and middle finger fully extended in (a), and only partially extended in (b). In (c), only the index is visible, fully extended. The EMD computes the correct flows both for the (a, b) pair and for the (a, c) pair. However, the two corresponding signature distances (7) are approximately equal to each other, and this is often undesirable: The pose of the hand in (a) is not too far from that in (b), while that in (c) is more distant from that in (a), at least in the \( L_2 \) or \( L_\infty \) norm. To address this issue, we introduce optional, additional ground distances \( k_{i0} \) and \( k_{0j} \) between regular and extra segments, and define these as some convex function of the EMD flows, for instance

\[
k_{i0} = \left( \frac{\hat{f}_{i0}}{w_i^S} \right)^2 \quad \text{and} \quad k_{0j} = \left( \frac{\hat{f}_{0j}}{w_j^T} \right)^2.
\]  

(8)

Before explaining how these terms are used, we discuss the possible need for another term. In some classification tasks, the pose of a hand whose extended
index and middle finger touch may represent a different category from one where the two fingers are kept slightly separate—for example, the letters ‘U’ (fingers touching) and ‘V’ (separate fingers) in fingerspelling. When these distinctions are important, it is useful to add an “abduction” term

\[ k_{ij}^{\text{ab}} = \alpha^{\text{ab}} (\pi^i_m - \pi^j_m) \]

which assumes that persistence reflects part separation. For example, in Figure 2 the configurations in (a) and (d) can be differentiated by the persistence of the index finger, which is much greater in (a) than in (d). We set \( \alpha^{\text{ab}} \) to 0 for regression and to 1 for classification in our experiments.

Our *signature dissimilarity* multiples \( k_{i0}, k_{oj}, \) and \( k_{ij}^{\text{ab}} \) by the EMD flows \( f_{ij} \), and leaves the resulting penalties after the EMD computation:

\[ D_{\text{SIG}}(S, T) = (1 - \alpha) \text{EMD}^2 + \alpha C \]

where we set \( \alpha = 0.5 \) in all our experiments,

\[ C(S, T, \hat{F}, K) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} k_{ij} f_{ij}}{\min(\sum_{i=1}^{m} w^i, \sum_{j=1}^{n} w^j)} \]

and the matrix \( K \) has entries

\[ k_{ij} = \begin{cases} k_{ij}^{\text{ab}} & i \neq 0, j \neq 0 \\ k_{i0} & i \neq 0 \\ k_{oj} & j \neq 0 \\ -\Delta & i = 0, j = 0 \end{cases} \]

The term \(-\Delta\) for \( i = 0, j = 0\) subtracts away an irrelevant cost for any flow between extra segments.

**Indexing Speedup.** The EMD between two signatures is no less than the distance between their centroids if the two signatures have equal mass and the ground distance is induced by a norm [13]. While these assumptions do not hold for our signatures, we have found a threshold on this distance, normalized by total flow, to be an effective heuristic for limiting the number of EMD computations. We also cluster our database by the number of segments with persistence greater than a threshold and match signatures in the same cluster. The combination of these heuristics leads to significant speedups (Section 7). As long as the thresholds used in them are generous (in the sense of over-clustering), their values are not critical.

**7. Results**

We compare with state-of-the-art methods for gesture classification and joint-pose regression that report quantitative results. The methods compared (including ours, SIG) are summarized in Table 3. Table 4 describes both the test datasets used for evaluation (Q) and the databases (A) we used for pose estimation. These include both standard benchmarks and our own databases (CG). We build the latter with an Intel DepthSense 325 depth sensor and a 23 DOF CyberGlove III sensor glove to collect input and ground truth data without manual annotation.

**Classification.** Table 1 compares our method (SIG) with Ren et al.’s thresholded and near-convex FingerEarth Mover’s Distance (FEMD) [12] methods by the correct-classification rate on the \( Q^{\text{FEMD}} \) gesture recognition dataset [12]. Like our method, FEMD does not add local optimization or use temporal information. However, it requires the user to wear a black band to identify the wrist [12]. FEMD does not account for partial correspondences nor does it accommodate an abduction term, and would fail to differentiate, for instance, between ‘U’ and ‘V’ in finger-spelling. Our results show significant improvements both in classification rate and speed (when speedup is used). When using the speedup we actually get a slight improvement in classification rate. This is due to eliminating possible false positives and the nature of \( Q^{\text{FEMD}} \’s \) small size, specific gesture classes and variance.

**Regression.** Figure 4 shows a comparison of our single-camera method with several state-of-the-art
regression methods using multiple test datasets. For consistency with the literature, we use the mean Euclidean distance between the estimated and ground truth joint positions of the fingertips (in millimeters) averaged over a test dataset.

We compare against Tang et al.’s latent regression forests (LRF) [18], Keskin et al.’s multi-layer random decision forest classifier (KESKIN) [8] and Melax et al.’s model-based approach (MELAX) [9] using the publicly available test datasets QLRF [18] and previous reported results [18]. QLRF is left-handed, so instead of our right-handed database \( \Lambda^{\text{SIG}} \) we use the provided left-handed training database \( \Lambda^{\text{LRF}} \)[18].

We also compare with Sridhar et al.’s sum of anisotropic Gaussians (SAG, 5 RGB cameras) [16] and earlier sum of Gaussians (SoG, 5 RGBD cameras) [15] approaches using their publicly available Dexter 1 (Q\text{\textsuperscript{DEX}} [16]) with 7 datasets with varying speed, degrees of wrist rotation and occlusions.

We use Q\text{\textsuperscript{DEX}} to compare with SAG, SoG, and LRF, which was evaluated on 3 of the 7 datasets. Performance for our methods on Q\text{\textsuperscript{DEX}} was evaluated using both database \( \Lambda^{\text{SIG}} \) and an approach similar to cross-validation, because \( \Lambda^{\text{SIG}} \) does not contain the range of wrist rotations present in Q\text{\textsuperscript{DEX}}. Specifically, we test each of the 7 subsets of Q\text{\textsuperscript{DEX}} using the other 6 subsets as \( \Lambda^{\text{DEX}} \). Our performance on Q\text{\textsuperscript{DEX}} has a much greater standard deviation when using database \( \Lambda^{\text{DEX}} \) instead of \( \Lambda^{\text{SIG}} \), as some subsets contain poses not found in other subsets. However, the error for our method and baseline is much lower using \( \Lambda^{\text{DEX}} \), likely because of similarity in hand shape, environment and motions to Q\text{\textsuperscript{DEX}}.

Our own baseline method (NN) uses the ground truth to search for the nearest neighbor in pose space to show the relative difficulty of the test datasets, listed increasing to the right: \( Q^{\text{ASL}} \), \( Q^{\text{SIG}} \), \( Q^{\text{LRF}} \), \( Q^{\text{DEX}} \). This is possibly a result of both the motion complexity and occlusions, as well as database coverage.

We also compare the results for our method using the first nearest neighbor (SIG1) and the best result from the 5 nearest neighbors (SIG5).

Our method is the only one that does not take advantage of local optimization or temporal information. We examine the potential impact of adding a local optimization step to our methods by finding the rotation and translation that minimizes the error between ground-truth and estimated pose (Figure 4).

Unlike our methods that only requires single-camera input, SAG uses 5-camera RGB input and SoG requires 5-camera RGB and depth input. Both SAG and SoG also require the user to wear a long black sleeve to help identify the wrist.

8. Conclusion

We estimate per-frame hand pose by encoding image silhouettes with signatures whose elements correlate well with fingers and knuckles, and indexing a database of real (signature, pose) pairs through a novel variant of the EMD that discards implausible matches but treats partial matches appropriately. Our method starts with a single 2D descriptor yet fares well even when compared with multi-camera methods. Our database samples natural hand poses in a balanced way and requires no manual annotation.

The obvious next step is to refine our pose estimates by fitting a hand model to the input point cloud, taking advantage of 3D data to resolve sensitivities to wrist and finger rotation. It will also be interesting to see if our methods can be used to train a pose regressor or a classifier [1], making the database unnecessary at runtime.

References

For consistency with the literature, ground truth pose is either for all DoF of our model (and we had both RMT or for finger root, middle, and tip (\(RMT\)) \cite{ren11}, \(FTIP\) \cite{sridhar13}, \(SoG\) \cite{soong14}). DoF were captured using an Intel DepthSense 325 and CyberGlove III (CG) simultaneously to obtain RGB-D and ground truth.

### Table 3: Methods Compared

<table>
<thead>
<tr>
<th>Method</th>
<th>Author</th>
<th>Data Obs.</th>
<th>Features</th>
<th>Global Est.</th>
<th>Local Opt./Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIG</td>
<td>us</td>
<td>RGB or D</td>
<td>convex components</td>
<td>index (\Lambda)</td>
<td></td>
</tr>
<tr>
<td>NN baseline</td>
<td></td>
<td>RGB or D</td>
<td>(\chi(Q))</td>
<td>index (\Lambda)</td>
<td>error reg. + glob. kin. const.</td>
</tr>
<tr>
<td>LRF</td>
<td>Tang ‘14 \cite{tang14}</td>
<td>RGB-D</td>
<td>vertex info gain</td>
<td>latent reg. forests</td>
<td></td>
</tr>
<tr>
<td>SAG</td>
<td>Sridhar ‘14 \cite{sridhar14}</td>
<td>RGB-D</td>
<td>2D quadtree (color)</td>
<td>2D aniso. gauss. error</td>
<td>joint and surface const.</td>
</tr>
<tr>
<td>SoG</td>
<td>Sridhar ‘13 \cite{sridhar13}</td>
<td>D, 5x RGB</td>
<td>color-depth edges</td>
<td>part based retrieval</td>
<td>voting + sum of gauss. fitting</td>
</tr>
<tr>
<td>MELAX</td>
<td>Melax ‘13 \cite{melax13}</td>
<td>D</td>
<td>3D point cloud</td>
<td>rigid body simulation</td>
<td>temporal, kin., collision const.</td>
</tr>
<tr>
<td>KESKIN</td>
<td>Keskin ‘12 \cite{keskin12}</td>
<td>RGB-D</td>
<td>pixel color+pos.</td>
<td>global expert network</td>
<td>local expert network</td>
</tr>
<tr>
<td>FEMD_{ncvx}</td>
<td>Ren ‘11 \cite{ren11}</td>
<td>RGB-D</td>
<td>near-convex parts</td>
<td>index (\Lambda)</td>
<td>Finger-EMD</td>
</tr>
<tr>
<td>FEMD_{thresh}</td>
<td>Ren ‘11 \cite{ren11}</td>
<td>RGB-D</td>
<td>height thresh. seg.</td>
<td>index (\Lambda)</td>
<td>Finger-EMD</td>
</tr>
</tbody>
</table>

Table 3: Methods Compared: FEMD is a classification only method; all others estimate hand-model parameters. All assume the hand to be the foremost object in the view. Additionally, SAG and SoG require a black long-sleeve shirt and FEMD a black wrist band. SAG and SoG are multi-camera approaches, and all except SIG, NN and FEMD do local optimization.

### Table 4: Test Datasets

| Test \(Q\) | \(|Q|\) | Ground-truth pose (\(\chi\)) | Database \(\Lambda\) | \(|\Lambda|\) | Description | Compared |
|------------|-------|-------------------------------|----------------------|----------|-------------|----------|
| \(Q_{ASL}(CG)\) | 1,900 | class + \(M^{nm}\) mm (CG)  | \(\Delta_{ASL}(CG)\) | 4,900    | ‘A’-‘Z’, ‘0’-‘9’ |          |
| \(Q_{SIG}(CG)\) | 4,140 | \(M^{nm}\) mm (CG)          | \(\Delta_{SIG}(CG)\) | 14,230   | random motion |          |
| \(Q_{FEMD}\) \cite{femd12} | 1,000 | class                         | \(\Delta_{FEMD} \cap q\) | 999      | 10 gestures | FEMD_{thresh}, FEMD_{ncvx} |
| \(Q_{FTIP}^{\text{prime}}\) \cite{tang14} | 1,850 | \(RMT^{nm}\) mm (9) + manual | \(\Delta_{FTIP}^{\text{prime}}\) \cite{tang14} | 301,095  | random motion | LRF, KESKIN, MELAX |
| \(Q_{DEX}\) \cite{femd16} | 3,155 | \(FTIP^{nm}\) mm (manual)   | \(\subset Q_{DEX}^{\text{prime}} / \Delta_{SIG}(CG)\) | \(<3, 155/14, 230\) | 7 subsets | SAG, SoG, LRF |

Table 4: Test Datasets: All include RGB-D data and assume the hand to be the foremost object. Our new datasets (CG) are captured using an Intel DepthSense 325 and CyberGlove III (CG) simultaneously to obtain RGB-D and ground truth. For consistency with the literature, ground truth pose is either for all DoF of our model (\(M\)), or for positions of fingertips (\(FTIP\) \cite{sridhar13, melax13}) or for finger root, middle, and tip (\(RMT\) \cite{tang14}). \(Q_{DEX}^{\text{prime}}\) is evaluated by the nearest neighbor for each query (\(q\)) in the dataset other than itself. We evaluate \(Q_{DEX}\) with \(\Delta_{SIG}(CG)\) (see text).

### Figure 4: Regression Results (best viewed in color)

Grey background is related work; white background is ours. Boxplot median at notch, mean at yellow diamond. When two boxplots overlap, the left one is actual results and right one is error for same method but with local optimization applied (using ground truth to show the potential). NN provides a bound for our performance. SIG shows the results of SIG’s nearest neighbor and SIG\(_{S}\) shows the results for SIG using the best of the 5 nearest neighbors. For KESKIN and MELAX standard deviation is set \(\sigma = 1\) and because only mean error for finger root, mean and tip (\(RMT\) was available \cite{tang14}, fingertip (\(FTIP\)) error is found using an adjustment ratio obtained by results for which we had both \(RMT\) and \(FTIP\) error such that \(d_{RMT}^{nm} / d_{FTIP}^{nm} \approx .8\). SAG and SoG require a black long-sleeve shirt and FEMD a black wrist band. SAG and SoG are multi-camera, and all except SIG, NN and FEMD do local optimization.


