Depth-based hand pose estimation: data, methods, and challenges

James Steven Supančič III[1], Grégory Rogez[2], Yi Yang[3], Jamie Shotton[4], Deva Ramanan[5]
University of California at Irvine[1], Inria[2], Baidu[3], Microsoft[4], Carnegie Mellon University[5]

Abstract

Hand pose estimation has matured rapidly in recent years. The introduction of commodity depth sensors and a multitude of practical applications have spurred new advances. We provide an extensive analysis of the state-of-the-art, focusing on hand pose estimation from a single depth frame. To do so, we have implemented a considerable number of systems, and will release all software and evaluation code. We summarize important conclusions here: (1) Pose estimation appears roughly solved for scenes with isolated hands. However, methods still struggle to analyze cluttered scenes where hands may be interacting with nearby objects and surfaces. To spur further progress we introduce a challenging new dataset with diverse, cluttered scenes. (2) Many methods evaluate themselves with disparate criteria, making comparisons difficult. We define a consistent evaluation criteria, rigorously motivated by human experiments. (3) We introduce a simple nearest-neighbor baseline that outperforms most existing systems. This implies that most systems do not generalize beyond their training sets. This also reinforces the under-appreciated point that training data is as important as the model itself. We conclude with directions for future progress.

1. Introduction

Human hand pose estimation empowers many practical applications, for example sign language recognition [14], visual interfaces [17], and driver analysis [20]. Recently introduced consumer depth cameras have spurred a flurry of new advances [4, 14, 15, 17, 25, 33, 36, 38, 42].

Motivation: Recent methods have demonstrated impressive results. But differing (often in-house) testsets, varying performance criteria, and annotation errors impede reliable comparisons [19]. In the field of object recognition, comprehensive benchmark evaluation has been vital for progress [6, 8]. Our goal is to similarly diagnose the state-of-affairs, and to suggest future strategic directions, for depth-based hand pose estimation.

Contributions: Foremost, we contribute the *most extensive* evaluation of depth-based hand pose estimators to date.

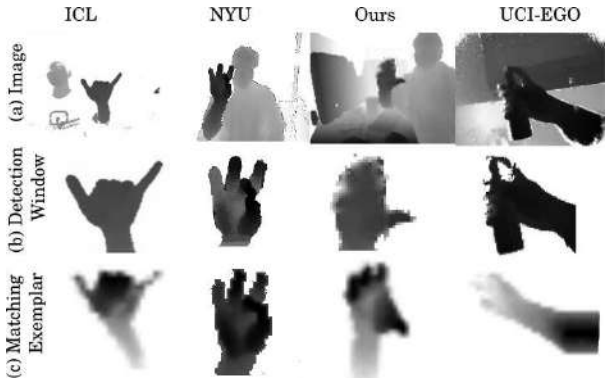

Figure 1. We evaluate a broad collection of hand pose estimation algorithms on different training and testsets under consistent criteria. Test sets which contained limited variety, in pose and range, or which lacked complex backgrounds were notably easier. To aid our analysis, we introduce a simple 3D exemplar (nearest-neighbor) baseline that both detects and estimates pose suprisingly well, outperforming most existing systems. We show the best-matching detection window in (**b**) and the best-matching exemplar in (**c**). We use our baseline to rank dataset difficulty, compare algorithms, and show the importance of training set design.

We evaluate 13 state-of-the-art hand-pose estimation systems across 4 testsets under uniform scoring criteria. Additionally, we provide a broad *survey* of contemporary approaches, introduce a *new testset* that addresses prior limitations, and propose a *new baseline* for pose estimation based on nearest-neighbor (NN) exemplar volumes. Surprisingly, we find that NN exceeds the accuracy of most existing systems (Fig. 1). We organize our discussion along three axes: test data (Sec. 2), training data (Sec. 3), and model architectures (Sec. 4). We survey and taxonomize approaches for each dimension, and also contribute novelty to each dimension (*e.g.* new data and models). After explicitly describing our experimental protocol (Sec. 5), we end with an extensive empirical analysis (Sec. 6).

Preview: We foreshadow our conclusions here. When hands are easily segmented or detected, current systems perform quite well. However, hand "activities" involving interactions with objects/surfaces are still challenging (motivating the introduction of our new dataset). Moreover, in such cases even humans perform imperfectly. For reasonable er-

Dataset	Chal.	Scn.	Annot.	Frms.	Sub.	Cam.	Dist. (mm)
ASTAR [42]	A	1	435	435	15	ToF	270-580
Dexter 1 [33]	A	1	3,157	3,157	1	Both	100-989
MSRA [25]	A	1	2,400	2,400	6	ToF	339-422
ICL [36]	A	1	1,599	1,599	1	Struct	200-380
FORTH [21]	AV	1	0	7,148	5	Struct	200-1110
NYU [38]	AV	1	8,252	8,252	2	Struct	510-1070
UCI-EGO [26]	AVC	4	364	3,640	2	ToF	200-390
Ours	AVC	10+	1182	23,640	10	Both	200-1950

Challenges (Chal.): A-Articulation V-Viewpoint C-Clutter

Table 1. **Testing data sets:** We group existing benchmark testsets into 3 groups based on `challenges` addressed - articulation, viewpoint, and/or background clutter. We also tabulate the number of captured `scenes`, number of `annotated` versus `total frames`, number of `subjects`, `camera` type (structured light vs time-of-flight), and `distance` of the hand to camera. We introduce a new dataset (**Ours**) that contains a significantly larger range of hand depths (up to 2m), more scenes (10+), more annotated frames (24K), and more subjects (10) than prior work.

ror measures, annotators disagree 20% of the time (due to self and inter-object occlusions and low resolution). This has immediate implications for test benchmarks, but also imposes a challenge when collecting and annotating training data. Finally, our NN baseline illustrates some surprising points. Simple memorization of training data performs quite well, outperforming most existing systems. Variations in the training data often dwarf variations in the model architectures themselves (*e.g.*, decision forests versus deep neural nets). Thus, our analysis offers the salient conclusion that "it's all about the (training) data".

Prior work: Our work follows in the rich tradition of benchmarking [8, 28] and taxiomatic analysis [7, 29]. In particular, Erol *et al.* [7] reviewed hand pose analysis in 2007. Contemporary approaches have considerably evolved, prompted by the introduction of commodity depth cameras. We believe the time is right for another look. We do extensive cross-dataset analysis (by training and testing systems on different datasets [39]). Human-level studies in benchmark evaluation [16] inspired our analysis of human-performance. Finally, our NN-baseline is closely inspired by non-parametric approaches to pose estimation [30]. In particular, we use volumetric depth features in a 3D scanning-window (or volume) framework, similar to [32]. But, our baseline does not need SVM training or multi-cue features, making it simpler to implement.

2. Testing Data

Test scenarios for depth-based hand-pose estimation have evolved rapidly. Early work evaluated on synthetic data, while contemporary work almost exclusively evaluates on real data. However, because of difficulties in manual annotation (a point that we will revisit), evaluation was not always quantitative - instead, it has been common to show select frames to give a qualitative sense of performance [5, 21]. We fundamentally assume that quantitative evaluation on real data will be vital for continued progress.

Test set properties: We have tabulated a list of contemporary test benchmarks in Table 1, giving URLs in Sec. A of the supplementary material. We refer the reader to the caption for a detailed summary of specific dataset properties. Per dataset, Fig. 3 visualizes the pose-space covered using multi-dimensional scaling (MDS). We plot both joint positions (in a normalized coordinate frame that is centered and scaled) and joint angles. Importantly, the position plot takes the global orientation (or camera viewpoint) of the hand into account while the angle plot does not. Most datasets are diverse in terms of joint angles but many are limited in terms of positions (implying they are limited in viewpoint). Indeed, we found that previous datasets make various assumptions about articulation, viewpoint, and perhaps most importantly, background clutter. Such assumptions are useful because they allow researchers to focus on particular aspects of the problem. However it is crucial to make such assumptions explicit [39], which much prior work does not. We do so below.

Articulation: Many datasets focus on pose estimation with the assumption that detection and overall hand viewpoint is either given or limited in variation. Example datasets include MSRA [25], A-Star [42], and Dexter [33]. We focus on **ICL** [36] as a representative example for experimental evaluation because it has been used in multiple prior published works [4, 36].

Art. and viewpoint: Other testsets have focused on both viewpoint variation and articulation. FORTH [21] provides five test sequences with varied articulations and viewpoints, but these are unfortunately unannotated. In our experiments, we analyze the **NYU** dataset [38] because of its wider pose variation (see Fig. 3) and accurate annotations (see Sec. 3).

Art. + View. + Clutter: The most difficult datasets contain cluttered backgrounds that are not easy to segment away. These datasets tend to focus on "in-the-wild" hands undergoing activities and interacting with nearby objects and surfaces. The **UCI-EGO** [26] dataset provides challenging sequences from an egocentric perspective, and so is included in our benchmark analysis.

Our testset: Our empirical evaluation will show that *in-the-wild hand activity* is still challenging. To push research in this direction, we have collected and annotated our own testset of real images (labeled as **Ours** in Table 1, examples in Fig. 2). As far as we are aware, our dataset is the first to focus on hand pose estimation *across multiple subjects and multiple cluttered scenes*. This is important, because any practical application must handle diverse subjects, scenes, and clutter.

Figure 2. **Our new test data** challenges methods with clutter, object manipulation, low-res, and various viewpoints. We collected data in diverse environments (8 offices, 4 homes, 4 public spaces, 2 vehicles, and 2 outdoors) using time-of-flight (Intel/Creative Gesture Camera) and structured-light (ASUS Xtion Pro) depth cameras. Ten (3 female and 7 male) subjects were given prompts to perform natural interactions with objects in the environment, as well as display 24 random and 24 canonical poses.

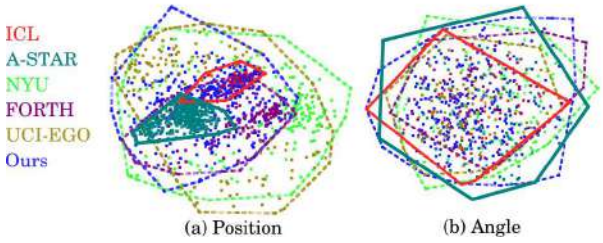

Figure 3. **Pose variation:** We use MDS (multi-dimensional scaling) to plot the pose space covered by various hand datasets. For each testset, we plot the convex hull of its poses. We plot joint positions (left) and joint angles (right). In terms of joint angle coverage (which does not consider the "root" orientation of the hand itself), most datasets are similar. In terms of joint position, some datasets are limited because they consider a smaller range of viewpoints (*e.g.*, ICL and A-STAR). We further analyze various assumptions made by datasets in the text.

3. Training Data

Here we discuss various approaches for generating training data (ref. Table 2). Real annotated training data has long been the gold standard for supervised learning. However, the generally accepted wisdom (for hand pose estimation) is that the space of poses is too large to manually annotate. This motivates approaches to leverage synthetically generated training data, discussed further below.

Real data + manual annotation: Arguably, the space of hand poses exceeds what can be sampled with real data. Our experiments identify a second problem: perhaps surprisingly, human annotators often disagree on pose annotations. For example, in our testset, human annotators disagreed on 20% of pose annotations (considering a 20mm threshold) as plotted in Fig. 9. These disagreements arise from limitations in the raw sensor data, either due to poor resolution or occlusions (as shown in Sec. G of the supplementary material). These ambiguities are often mitigated by placing the hand close to the camera [25, 36, 42]. As an illustrative example, we evaluate the **ICL** training set [36].

Real data + automatic annotation: Data gloves directly obtain automatic pose annotations for real data [42]. However, they require painstaking per-user calibration and distort the hand shape that is observed in the depth map. Alternatively, one could use a "passive" motion capture system. We evaluate the **NYU** training set [38] that annotates real data by fitting (offline) a skinned 3D hand model to high-quality 3D measurements.

Dataset	Generation	Viewpoint	Views	Size	Subj.
ICL [36]	Real + manual annot.	3rd Pers.	1	331,000	10
NYU [38]	Real + auto annot.	3rd Pers.	3	72,757	1
UCI-EGO [26]	Synthetic	Egocentric	1	10,000	1
libhand [41]	Synthetic	Generic	1	25,000,000	1

Table 2. **Training data sets:** We broadly categorize training datasets by the method used to `generate` the data and annotations: real data + manual annotations, real data + automatic annotations, or synthetic data (and automatic annotations). Most existing datasets are `viewpoint`-specific (tuned for 3rd-person or egocentric recognition) and limited in `size` to tens of thousands of examples. NYU is unique in that it is a multi`view` dataset collected with multiple cameras, while ICL contains shape variation due to multiple (10) `subjects`. To explore the effect of training data, we use the public libhand animation package to generate a massive training set of 25 million examples.

Quasi-synthetic data: Augmenting real data with geometric computer graphics models provides an attractive solution. For example, one can apply geometric transformations (*e.g.*, rotations) to both real data and its annotations [36]. If multiple depth cameras are used to collect real data (that is then registered to a model), one can synthesize a larger set of varied viewpoints [38]. Finally, mimicking the noise and artifacts of real data is often important when using synthetic data. Domain transfer methods [4] learn the relationships between a small real dataset and large synthetic one.

Synthetic data: Another hope is to use data rendered by a computer graphics system. Graphical synthesis sidesteps the annotation problem completely: precise annotations can be rendered along with the features. When synthesizing novel exemplars, it is important define a good sampling distribution. The **UCI-EGO** training set [26] synthesizes data with an egocentric prior over viewpoints and grasping poses. A common strategy for generating a sampling distribution is to collect pose samples with motion capture data [3, 10]. To further examine the effect of training data, we created a massive custom training set of 25,000,000 RGB-D training instances with the open-source **libhand** model (some examples are shown in Table 4).

4. Methods

Next we survey existing approaches to hand pose estimation (summarized in Table 3). We conclude by introducing a novel volumetric nearest-neighbor (NN) baseline.

Method	Approach	Model-drv.	Data-drv.	Detection	Implementation	FPS
Simulate [17]	Tracker (simulation)	Yes	No	Initialization	Published	50
NiTE2 [24]	Tracker (pose search)	No	Yes	Initialization	Public	> 60
Particle Swarm Opt. (PSO) [21]	Tracker (PSO)	Yes	No	Initialization	Public	15
Hough Forest [42]	Decision forest	Yes	Yes	Decision forest	Ours	12
Random Decision Forest (RDF) [14]	Decision forest	No	Yes	-	Ours	8
Latent Regression Forest (LRF) [36]	Decision forest	No	Yes	-	Published	62
DeepJoint [38]	Deep network	Yes	Yes	Decision forest	Published	25
DeepPrior [19]	Deep network	No	Yes	Scanning window	Ours	5000
DeepSegment [9]	Deep network	No	Yes	Scanning window	Ours	5
Intel PXC [12]	Morphology (convex detection)	No	No	Heuristic segment	Public	> 60
Cascades [26]	Hierarchical cascades	No	Yes	Scanning window	Provided	30
EPM [43]	Deformable part model	No	Yes	Scanning window	Ours	$1/2$
Volumetric Exemplars	Nearest neighbor (NN)	No	Yes	Scanning volume	Ours	$1/15$

Table 3. **Summary of methods:** We broadly categorize the pose estimation systems that we evaluate by their overall `approach`: decision forests, deep models, trackers, or others. Though we focus on single-frame systems, we also evaluate trackers by providing them manual initialization. `Model-driven` methods make use of articulated geometric models at test time, while `data-driven` models are trained beforehand on a training set. Many systems begin by `detect`ing hands with a Hough-transform or a scanning window/volume search. Finally, we made use of public source code when available, or re-`implement`ed the system ourselves, verifying our implementation's accuracy on published benchmarks. 'Published' indicates that published performance results were used for evaluation, while 'public' indicates that source code was available, allowing us to evaluate the method on additional testsets. We report the fastest speeds (in `FPS`), either reported or our implementation's.

4.1. Taxonomy

Trackers versus detectors: We focus our analysis on single-frame methods. For completeness, we also consider several tracking baselines [12, 21, 24] needing ground-truth initialization. Manual initialization may provide an unfair advantage, but we will show that single-frame methods are still nonetheless competitive, and in most cases, outperform tracking-based approaches. One reason is that single-frame methods essentially "reinitialize" themselves at each frame, while trackers cannot recover from an error.

Data-driven versus model-driven: Historic attempts to estimate hand pose optimized a geometric model to fit observed data [5, 21, 34]. However, such optimizations remain notoriously difficult due to local minima in the objective function. As a result, model driven systems have found their successes mostly to the tracking domain, where initialization constrains the search space [17, 25, 33]. For single image detection, various fast classifiers [12, 14] have obtained real-time speeds. Most of the systems we evaluate fall into this category. When these classifiers are trained with data synthesized from a geometric model, they can be seen as efficiently approximating model fitting.

Multi-stage pipelines: It is common to treat the initial detection (candidate generation) stage as separate from hand-pose estimation. Some systems use special purpose detectors as a "pre-processing" stage [12, 14, 21, 27, 38, 42]. Others use a geometric model for inverse-kinematic (IK) refinement/validation during a "post-processing" stage [17, 33, 38, 42]. A segmentation pre-processing stage has been historically popular. Typically, the depth image is segmented with simple morphological operations [23] or the RGB image is segmented with skin classifiers [40]. While RGB features compliment depth [11, 26], skin segmentation appears difficult to generalize across subjects and scenes with varying lighting [25]. We evaluate a depth-based segmentation system [12] for completeness.

4.2. Architectures

In this section, we describe popular architectures for hand-pose estimation, placing in bold those systems that we empirically evaluate.

Decision forests: Decision forests constitute a dominant paradigm for estimating hand pose from depth. **Hough Forests** [42] take a two-stage approach of hand detection followed by pose estimation. **Random Decision Forests (RDFs)** [14] and **Latent Regression Forests (LRFs)** [36] leave the initial detection stage unspecified, but both make use of coarse-to-fine decision trees that perform rough viewpoint classification followed by detailed pose estimation. We experimented with several detection front-ends for RDFs and LRFs, finally selecting the first-stage detector from Hough Forests for its strong performance.

Part model: Pictorial structure models have been popular in human body pose estimation, but they appear rare in hand pose estimation. For completeness, we evaluate a deformable part model defined on depth image patches. We specifically train an **exemplar part model (EPM)** constrained to model deformations consistent with 3D exemplars [43], which will be described further in a tech report.

Deep models: Recent systems have explored the use of deep neural nets for hand pose estimation. We consider three variants in our experiments. **DeepJoint** [38] uses a three stage pipeline that initially detects hands with a decision forest, regresses joint locations with a deep network,

and finally refines joint predictions with inverse kinematics (IK). **DeepPrior** [19] is based on a similar deep network, but does not require an IK stage and instead relies on the network itself to learn a spatial prior. **DeepSeg** [9] takes a pixel-labeling approach, predicting joint labels for each pixel, followed by a clustering stage to produce joint locations. This procedure is reminiscent of pixel-level part classification of Kinect [31], but substitutes a deep network for a decision forest.

4.3. Volumetric exemplars

We propose a nearest-neighbor (NN) baseline for additional diagnostic analysis. Specifically, we convert depth map measurements into a 3D voxel grid, and simultaneously detect and estimate pose by scanning over this grid with volumetric exemplar templates. We introduce several modifications to ensure an efficient scanning search.

Voxel grid: Depth cameras report depth as a function of pixel (u, v) coordinates: $D(u, v)$. To construct a voxel grid, we first re-project these image measurements into 3D using known camera intrinsics f_u, f_v.

$$(x, y, z) = \left(\frac{u}{f_u} D(u, v), \frac{v}{f_v} D(u, v), D(u, v) \right) \quad (1)$$

Given a test depth image, we construct a binary voxel grid $V[x, y, z]$ that is '1' if a depth value is observed at a quantized (x, y, z) location. To cover the rough viewable region of a camera, we define a coordinate frame of M^3 voxels, where $M = 200$ and each voxel spans 10mm^3. We similarly convert training examples into volumetric exemplars $E[x, y, z]$, but instead use a smaller N^3 grid of voxels (where $N = 30$), consistent with the size of a hand.

Occlusions: When a depth measurement is observed at a position $(x', y', z') = 1$, all voxels behind it are occluded $z > z'$. We define occluded voxels to be '1' for both the test-time volume V and training exemplar E.

Distance measure: Let V_j be the j^{th} subvolume (of size N^3) extracted from V, and let E_i be the i^{th} exemplar. We simultaneously detect and estimate pose by computing the best match in terms of Hamming distance:

$$(i^*, j^*) = \underset{i,j}{\operatorname{argmin}} \operatorname{Dist}(E_i, V_j) \qquad \text{where} \quad (2)$$

$$\operatorname{Dist}(E_i, V_j) = \sum_{x,y,z} \mathcal{I}(E_i[x, y, z] \neq V_j[x, y, z]), \quad (3)$$

such that i^* is the best-matching training exemplar and j^* is its detected position.

Efficient search: A naive search over exemplars and subvolumes is prohibitively slow. But because the underlying features are binary and sparse, there exist considerable opportunities for speedup. We outline two simple strategies. First, one can eliminate subvolumes that are empty, fully occluded, or out of the camera's field-of-view. Song *et al.* [32] refer to such pruning strategies as "jumping window" searches. Second, one can compute volumetric Hamming distances with 2D computations:

$$\operatorname{Dist}(E_i, V_j) = \sum_{x,y} |e_i[x, y] - v_j[x, y]| \qquad \text{where} \quad (4)$$

$$e_i[x, y] = \sum_z E_i[x, y, z], \qquad v_j[x, y] = \sum_z V_j[x, y, z].$$

Intuitively, because our 3D volumes are projections of 2.5D measurements, they can be sparsely encoded with a 2D array (see Fig. 4). Taken together, our two simple strategies imply that a *3D volumetric search can be as practically efficient as a 2D scanning-window search*. For a modest number of exemplars, our implementation still took tens of seconds per frame, which sufficed for our offline analysis. We posit faster NN algorithms could yield real-time speed [18].

Comparison: Our volumetric exemplar baseline uses a scanning volume search and 2D depth encodings. It is useful to contrast this with a "standard" 2D scanning-window template on depth features [13]. First, our exemplars are defined in metric coordinates (Eq. 1). This means that they will *not* fire on the small hands of a toy figurine, unlike a scanning window search over scales. Second, our volumetric search ensures that the depth encoding from a local window contain features only within a fixed N^3 volume. This gives it the ability to segment out background clutter, unlike a 2D window (Fig. 5).

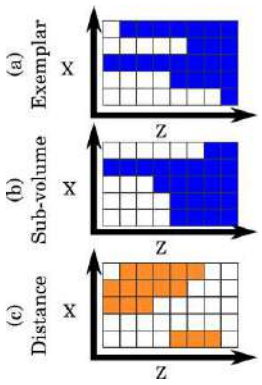

Figure 4. **Volumetric Hamming distance:** We visualize 3D voxels corresponding to an exemplar (a) and subvolume (b). For simplicity, we visualize a 2D slice along a fixed y-value. Because occluded voxels are defined to be '1' (indicating they are occupied, shown in blue) the total Hamming distance is readily computed by the L1 distance between projections along the z-axis (c), mathematically shown in Eq.(4).

Figure 5. **Windows v. volumes:** 2D scanning windows **(a)** versus 3D scanning volumes **(b)**. Volumes can ignore background clutter that lies outside the 3D scanning volume but still falls inside its 2D projection. For example, when scoring the shown hand, a 3D volume will ignore depth measurements from the shoulder and head, unlike a 2D window.

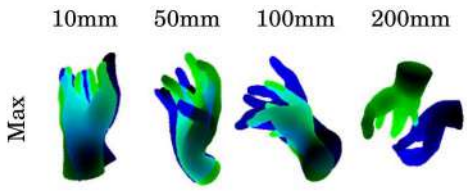

Figure 6. **Our error criteria:** For each predicted hand, we calculate the average and maximum distance (in mm) between its skeletal joints and a ground-truth. In our experimental results, we plot the fraction of predictions that lie within a distance threshold, for various thresholds. This figure visually illustrates the misalignment associated with various thresholds for max error. A 50mm max-error seems visually consistent with a "roughly correct pose estimation", and a 100mm max-error is consistent with a "correct hand detection".

5. Evaluation protocol

Reprojection error: Following past work, we evaluate pose estimation as a regression task that predicts a set of 3D joint locations [14, 21, 25, 36, 37]. Given a predicted and ground-truth pose, we compute both the average and max 3D reprojection error (in mm) across all joints. We use the skeletal joints defined by libhand [41]. We then summarize performance by plotting the proportion of test frames whose average (or max) error falls below a threshold.

Error thresholds: Much past work considers performance at fairly low error thresholds, approaching 10mm [36, 38, 42]. Interestingly, [19] show that established benchmarks such as the **ICL** testset include annotation errors of above 10mm in over a third of their frames. Ambiguities arise from manual labeling of joints versus bones and centroids versus surface points. We rigorously evaluate human-level performance through inter-annotator agreement on our new testset (Fig. 9). Overall, we find that max-errors of **20mm** approach the limit of human accuracy for closeby hands. We present a qualitative visualization of max error at different thresholds in Fig. 6. **50mm** appears consistent with a roughly correct pose, while an error within **100mm** appears consistent with a correct detection. Our qualitative analysis is consistent with empirical studies of human grasp [1] and gesture [35] which also suggest that 50mm is sufficient to capture difference in gesture or grasp. For completeness, we plot results across a large range of thresholds, but highlight 50 and 100mm thresholds for additional analysis.

Detection issues: Reprojection error is hard to define during detection failures: that is, false positive hand detections or missed hand detections. Such failures are likely in cluttered scenes or when considering scenes containing zero or two hands. If a method produced zero detections when a hand was present, or produced one if no hand was present, this was treated as a "maxed-out" reprojection error (of ∞ mm). If two hands were present, we scored each method against both and took the minimum error. Though we plan to release our evaluation software, Section F of the supplementary material provides pseudocode.

Missing data: Another challenge with reprojection error is missing data. First, some methods predict 2D rather than 3D joints [9, 12, 23, 38]. Inferring depth should in theory be straightforward with Eq. 1, but small 2D errors in the estimated joint can cause significant errors in the estimated depth. We report back the centroid depth of a segmented/detected hand if the measured depth lies outside the segmented volume. Past comparisons appear not to do this [19], somewhat unfairly penalizing 2D approaches [38]. Second, some methods may predict a subset of joints [12, 23]. To ensure a consistent comparison, we force such methods to predict the locations of visible joints with a post-processing inverse-kinematics (IK) stage [38]. We fit the libhand kinematic model to the predicted joints, and infer the location of missing ones. Third, ground-truth joints may be occluded. By convention, we only evaluate visible joints in our benchmark analysis.

Implementations: We use public code when available [12, 21, 24]. Some authors responded to our request for their code [26]. When software was not available, we attempted to re-implement methods ourselves. We were able to successfully reimplement [14, 19, 42], matching the accuracy on published results [19, 36]. In other cases, our in-house implementations did not suffice [36, 38]. For these latter cases, we include published performance reports, but unfortunately, they are limited to their own datasets. This partly motivated us to perform a multi-dataset analysis. In particular, previous benchmarks have shown that one can still compare algorithms across datasets using head-to-head matchups (similar to approaches that rank sports teams which do not directly compete [22]). We use our NN baseline to do precisely this. Finally, to spur further progress, *we will make all implementations publicly available, together with our evaluation code.*

6. Results

We now report our experimental results, comparing datasets and methods. For more detailed plots and additional error criteria, please see Supp. Sec. B. We first address the "state of the problem": what aspects of the problem have been solved, and what remain open research questions? We conclude by discussing the specific lessons we learned and suggesting directions for future systems.

Mostly-solved (distinct poses): Fig. 7 shows that hand pose estimation is mostly solved on datasets of uncluttered scenes where hands face the camera (*i.e.* ICL). Deep models, decision forests, and NN all perform quite well, both in terms of articulated pose estimation (85% of frames are within 50mm max-error) and hand detection (100% are within 100mm max-error). Surprisingly, NN outperforms decision forests by a bit. However, when NN is trained

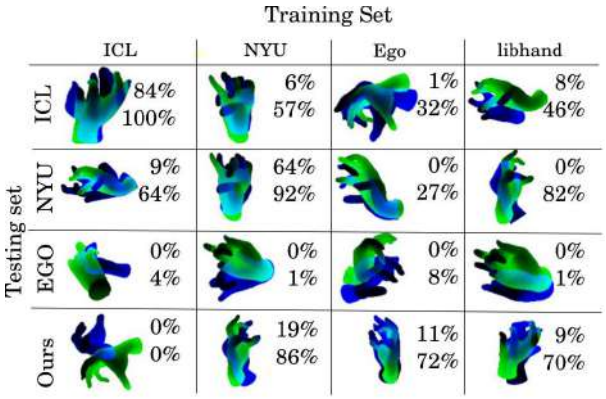

Table 4. **Cross-dataset generalization:** We compare training and test sets using a 1-NN classifier. Diagonal entries represent the performance using corresponding train and test sets. In each grid entry, we denote the percentage of test frames that are correct (50mm max-error, above, and 50mm average-error, below) and visualize the median error using the colored overlays from Fig. 6. We account for sensor specific noise artifacts using established techniques [2]. Please refer to the text for more details.

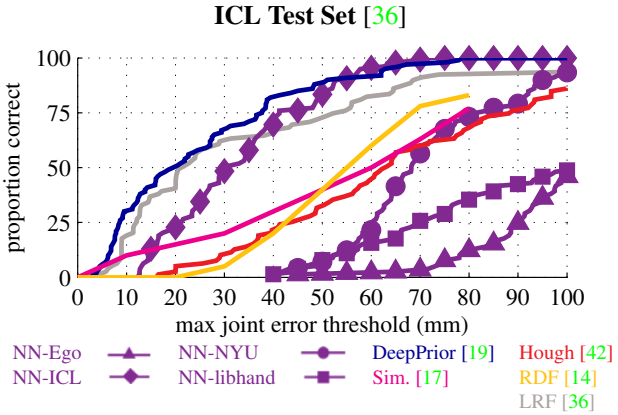

Figure 7. We plot results for several systems on the ICL testset using max-error (we include avg-error in supplemental Section B). Except for 1-NN, all systems are trained on the corresponding train set (in this case ICL-Train). To examine cross-dataset generalization, we also plot the performance of our NN-baseline constructed using alternate sets (NYU, EGO, and libhand). When trained with ICL, NN performs as well or better than prior art. One can find near-perfect pose matches in the training set (see Fig. 1). Please see text for further discussion.

on other datasets with larger pose variation, performance is considerably worse. This suggests that the test poses remarkably resemble the training poses. Novel poses (those not seen in training data) account for most of the remaining failures. More training data (or better model generalization) should correct these. Yet, this may be reasonable for applications targeting sufficiently distinct poses from a finite vocabulary (*e.g.*, a gaming interface). These results suggest that *the state-of-the-art accurately predicts distinct poses* (*i.e.* 50 mm apart) *in uncluttered scenes.*

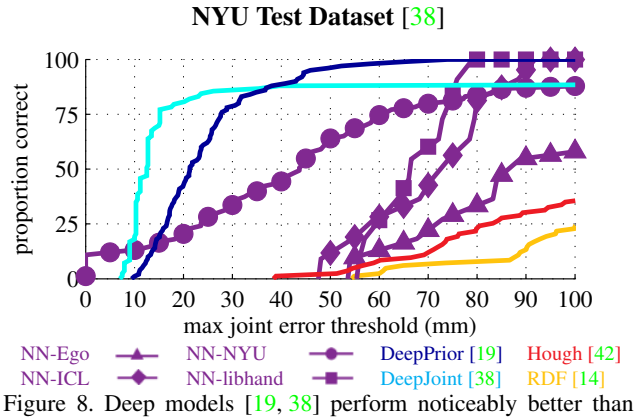

Figure 8. Deep models [19, 38] perform noticeably better than other systems, and appear to solve both articulated pose estimation and hand detection for uncluttered single-user scenes (common in the NYU testset). In Supp. Sec. C.1, we show that 1-NN is much more competitive under average error, due to the fact that test hands often match a training example in all but one finger. Please see text for further discussion.

Major progress (unconstrained poses): The NYU testset still considers isolated hands, but includes a wider range of poses, viewpoints, and subjects compared to ICL (see Fig. 3). Fig. 8 reveals that deep models perform the best for both articulated pose estimation (96% accuracy) and hand detection (100% accuracy). While decision forests struggle with the added variation in pose and viewpoint, NN still does quite well. In fact, when measured with average (rather than max) error, NN nearly matches the performance of [38]. This suggests that exemplars get most, but not all fingers, correct (see Supp. Sec. C.1). *We see noticeable progress on unconstrained pose estimation since 2007* [7].

Unsolved (low-res, objects, occlusions, clutter): When considering datasets with distant (low-res) hands and scenes cluttered with objects or interacting surfaces (Fig. 9 and 10), results are significantly worse. Note that many applications [31] often demand hands to lie at distances greater than 750mm. For such scenes, hand detection is still a challenge. Scanning window approaches (such as our NN baseline) tend to outperform multistage pipelines [9, 14], which may make an unrecoverable error in the first (detection and segmentation) stage. We show some illustrative examples in supplementary Section H. Yet, overall performance is still lacking, particularly when compared to human performance. Notably, human (annotator) accuracy also degrades for low-resolution hands far away from the camera (Fig. 9). Our results suggest that *scenes of in-the-wild hand activity are still beyond the reach of the state-of-the-art.*

Training data: We use our NN-baseline to analyze the effect of training data in Table 4. Our NN model performed better using the NYU training set [38] (consisting of real data automatically labeled with a geometrically-fit 3D CAD model) than with the libhand training set. While enlarg-

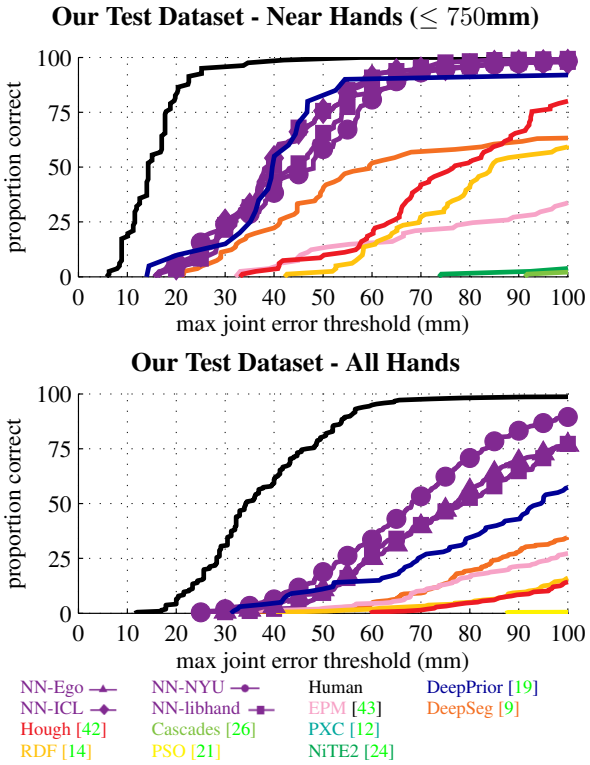

Figure 9. We designed our dataset to address the remaining challenges of in "in-the-wild" hand pose estimation, including scenes with low-res hands, clutter, object/surface interactions, and occlusions. We plot human-level performance (as measured through inter-annotator agreement) in black. On nearby hands (within 750mm, as commonly assumed in prior work) our annotation quality is similar to existing testsets such as ICL [19]. This is impressive given that our testset includes comparatively more ambiguous poses (see Supp. Sec. G). Our dataset includes far away hands, for which even humans struggle to accurately label. Moreoever, several methods (Cascades,PXC,NiTE2,PSO) fail to correctly localize any hand at any distance, though the mean-error plots in Supp. Sec. B are more forgiving than the max-error above. In general, NN-exemplars and DeepPrior perform the best, correctly estimating pose on 75% of frames with nearby hands.

ing the synthetic training set increases performance (Supp. Sec. D), computation fast becomes intractable. This reflects the difficulty in using synthetic data: one must carefully model priors [19], sensor noise, [11] and hand shape variations between users [37]. Moreover, in some cases, the variation in the performance of NN (dependent on the particular training set) exceeded the variation between model architectures (decision forests versus deep models) - Fig. 7. Our results suggest the diversity and realism of the *training set is as important than the model form learned from it*.

NN vs Deep models: Overall, our 1-NN baseline proved to be suprisingly strong, outperforming or matching the performance of most prior systems. This holds true even for moderately-sized training sets with tens of thousands of examples, suggesting that much prior work essentially memorizes training examples. One contribution of our analysis is the notion that *NN-exemplars provides a vital baseline for understanding the behavior of a proposed system in relation to its training set.* In fact, DeepJoint [38] and DeepPrior [19] were the sole approaches to significantly outperform 1-NN (Figs. 7 and 8). This indicates that deep architectures generalize well to novel test poses. This may contrast with existing folk wisdom about deep models: that the need for large training sets suggests that these models essentially memorize. Our results indicate otherwise.

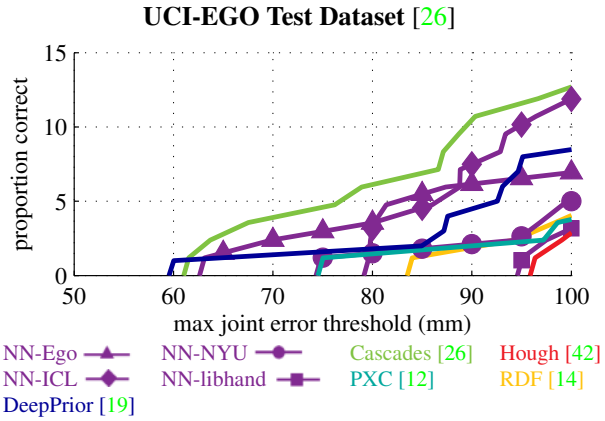

Figure 10. For UCI-EGO, randomized cascades [26] (the only approach specifically tackling egocentric viewpoints) and our NN baseline do about as well, but overall, performance is considerably worse than other datasets. No methods are able to correctly estimate the pose (within 50mm) on *any* frames. Egocentric scenes contain more background clutter and object/surface interfaces, making even hand detection challenging for many methods. Egocentric hand pose estimation remains an open problem.

Conclusion: The past several years have shown tremendous progress regarding hand pose: training sets, testing sets, and models. Some applications, such as gaming interfaces and sign-language recognition, appear to be well-within reach for current systems. Less than a decade ago, this was not true [7,23]. Thus, we have made progress! But, challenges remain nonetheless. Specifically, when segmentation is hard due to active hands or clutter, many existing methods fail. To illustrate these realistic challenges we introduce a novel testset. We demonstrate that realism and diversity in training sets is crucial, and can be as important as the choice of model architecture. In terms of model architecture, we perform a broad benchmark evaluation and find that deep models appear particularly well-suited for pose estimation. Finally, we demonstrate that NN using volumetric exemplars provides a startlingly potent baseline, providing an additional tool for analyzing both methods and datasets.

Acknowledgement: NSF Grant 0954083, ONR-MURI Grant N00014-10-1-0933, and the Intel Science and Technology Center - Visual Computing supported JS&DR. The European Commission FP7 Marie Curie IOF grant "Egovision4Health" (PIOF-GA-2012-328288) supported GR.

References

[1] I. M. Bullock, S. Member, J. Z. Zheng, S. D. L. Rosa, C. Guertler, and A. M. Dollar. Grasp Frequency and Usage in Daily Household and Machine Shop Tasks. *IEEE T. Haptics*, 6(3), 2013.

[2] M. Camplani and L. Salgado. Efficient spatio-temporal hole filling strategy for Kinect depth maps. In *SPIE*, 2012.

[3] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo. Using object affordances to improve object recognition. *IEEE TAMD*, 3(3), 2011.

[4] T. Y. D. Tang and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013.

[5] Q. Delamarre and O. Faugeras. 3D Articulated Models and Multiview Tracking with Physical Forces. *CVIU*, 2001.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. IEEE, 2009.

[7] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *CVIU*, 2007.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 2010.

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013.

[10] T. Feix, J. Romero, C. H. Ek, H. Schmiedmayer, and D. Kragic. A Metric for Comparing the Anthropomorphic Motion Capability of Artificial Hands. *IEEE Robotics*, 2013.

[11] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*. Springer, 2014.

[12] Intel. Perceptual computing SDK, 2013.

[13] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3D object dataset: Putting the kinect to work. In *CDC4CV*. 2013.

[14] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV 2012*. 2012.

[15] C. Li and K. M. Kitani. Pixel-Level Hand Detection in Egocentric Videos. *CVPR*, June 2013.

[16] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5), 2004.

[17] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3D skeletal hand tracking. *ACM SIGGRAPH - I3D*, 2013.

[18] M. Muja and D. G. Lowe. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *TPAMI*, 36(11), Nov. 2014.

[19] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands Deep in Deep Learning for Hand Pose Estimation. *CVWW*, 2015.

[20] E. Ohn-Bar and M. M. Trivedi. Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations. *IEEE TITS*, 2014.

[21] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. In *BMVC*, 2011.

[22] Y. Pang and H. Ling. Finding the Best from the Second Bests - Inhibiting Subjective Bias in Evaluation of Visual Tracking Algorithms. *ICCV*, Dec. 2013.

[23] P. Premaratne, Q. Nguyen, and M. Premaratne. *Human computer interaction using hand gestures*. Springer, 2010.

[24] PrimeSense. Nite2 middleware, 2013. Version 2.2.

[25] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. *CVPR 2014*.

[26] G. Rogez, M. Khademi, J. S. S. III, J. M. M. Montiel, and D. Ramanan. 3D hand pose detection in egocentric rgb-d images. *CDC4CV Workshop*, 2014.

[27] J. Romero, H. Kjellstrom, and D. Kragic. Monocular Real-Time 3D Articulated Hand Pose Estimation. *IEEE-RAS Humanoids*, 2009.

[28] O. Russakovsky, J. Deng, Z. Huang, A. C. Berg, and L. Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *ICCV*. IEEE, 2013.

[29] D. Scharstein. A Taxonomy and Evaluation of Dense Two-Frame Stereo. *IJCV*, 47(1), 2002.

[30] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003.

[31] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1), 2013.

[32] S. Song and J. Xiao. Sliding Shapes for 3D Object Detection in Depth Images. *ECCV*, 2014.

[33] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. *ICCV*, 2013.

[34] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical Bayesian filter. *IEEE TPAMI*, 28(9), Sept. 2006.

[35] W. C. Stokoe. Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of deaf studies and deaf education*, 10(1), 2005.

[36] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3D articulated hand posture. *CVPR*, 2014.

[37] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *CVPR*. IEEE, 2014.

[38] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM T. Graphics*, 33, August 2014.

[39] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528. IEEE, 2011.

[40] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. Graphicon*, volume 3. Moscow, Russia, 2003.

[41] M. Šarić. Libhand: A library for hand articulation, 2011.

[42] C. Xu and L. Cheng. Efficient Hand Pose Estimation from a Single Depth Image. *ICCV*, Dec. 2013.

[43] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, 2012.