International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

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

James Steven Supančič III · Grégory Rogez · Yi Yang · Jamie Shotton · Deva Ramanan

Received: date / Accepted: date

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 have released software and evaluation code. We summarize important conclusions here: (1) Coarse pose estimation appears viable for scenes with isolated hands. However, high precision pose estimation (required for immersive virtual reality) and cluttered scenes (where hands may be interacting with nearby objects and surfaces) remain a challenge. 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.

James Steven Supančič III
University of California, Irvine

Grégory Rogez
Univ. Grenoble Alpes, Inria, CNRS, INP, LJK, France

Yi Yang
Baidu Institute of Deep Learning

Jamie Shotton
Microsoft Research

Deva Ramanan
Carnegie Mellon University

Keywords hand pose; RGB-D sensor; datasets; benchmarking

1 Introduction

Human hand pose estimation empowers many practical applications, for example sign language recognition (Keskin et al. 2012), visual interfaces (Melax et al. 2013), and driver analysis (Ohn-Bar and Trivedi 2014a). Recently introduced consumer depth cameras have spurred a flurry of new advances (Ren et al. 2011, Keskin et al. 2012, D. Tang and Kim 2013, Li and Kitani 2013, Melax et al. 2013, Xu and Cheng 2013, Tang et al. 2014, Tompson et al. 2014, Qian et al. 2014, Sridhar et al. 2015).

Motivation: Recent methods have demonstrated impressive results. But differing (often in-house) testsets, varying performance criteria, and annotation errors impede reliable comparisons (Oberweger et al. 2015a). Indeed, a recent meta-level analysis of object tracking papers reveals that it is difficult to trust the "best" reported method in any one paper (Pang and Ling 2013). In the field of object recognition, comprehensive benchmark evaluation has been vital for progress (Fei-Fei et al. 2007, Deng et al. 2009, Everingham et al. 2010). 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. 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

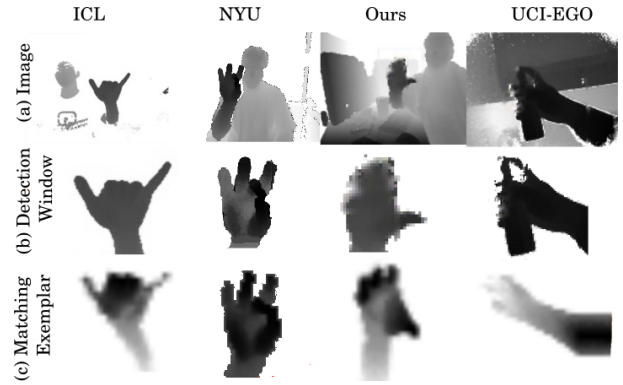


Fig. 1 NN Memorization: We evaluate a broad collection of hand pose estimation algorithms on different training and testsets under consistent criteria. Test sets which contained limited variety, in pose and range, or which lacked complex backgrounds were notably easier. To aid our analysis, we introduce 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 provide a detailed analysis of which problem types are currently solved, what open research challenges remain, and provide suggestions for future model architectures.

Dataset	Chal.	Scn.	Annot.	Frms.	Sub.	Cam.	Dist. (mm)
ASTAR [70]	A	1	435	435	15	ToF	270-580
Dexter 1 [55]	A	1	3,157	3,157	1	Both	100-989
MSRA-2014 [42]	A	1	2,400	2,400	6	ToF	339-422
ICL [59]	A	1	1,599	1,599	1	Struct	200-380
FORTH [37]	AV	1	0	7,148	5	Struct	200-1110
NYU [63]	AV	1	8,252	8,252	2	Struct	510-1070
HandNet [69]	AV	1	202,198	202,198	10	Struct	200-650
MPG-2014 [65]	AV	1	2,800	2,800	1	Struct	500-800
FingerPaint [51]	AV	1	113,800	113,800	5	ToF	400-700
CVAR-EGO [32]	AV	2	2,166	2,166	1	ToF	60-650
MSRA [58]	AV	1	76,375	76,528	9	ToF	244-530
KTH [39]	AVC	1	NA	46,000	9	Struct	NA
LISA [35]	AVC	1	NA	3,100	1	Struct	900-3780
UCI-EGO [44]	AVC	4	364	3,640	2	ToF	200-390
Ours	AVC	10+	23,640	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.

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 error 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 (Everingham et al. 2010, Dollar et al. 2012, Russakovsky et al. 2013) and taxiomatic analysis (Scharstein 2002, Erol et al. 2007). In particular, Erol *et al.* (Erol et al. 2007) 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 (Torralba and Efros 2011). Human-level studies in benchmark evaluation (Martin et al. 2004) inspired our analysis of human-performance. Finally, our NN-baseline is closely inspired by non-parametric approaches to pose estimation (Shakhnarovich et al. 2003). In particular, we use volumetric depth features in a 3D scanning-window (or volume) framework, similar to Song and Xiao (2014). 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 (Delamarre and Faugeras 2001, Bray et al. 2004, Oikonomidis et al. 2011, Pieropan et al. 2014). We fundamentally assume that quantitative evaluation on real data will be vital for continued progress.

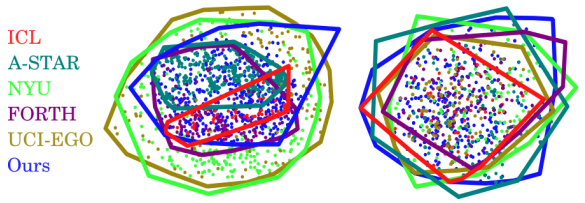
Test set properties: We have tabulated a list of contemporary test benchmarks in Table 1, giving URLs on our website[1]. We refer the reader to the caption for a detailed summary of specific dataset properties. Per dataset, Fig. 2 visualizes the pose-space covered using multi-dimensional scaling (MDS). We embed both the camera viewpoint angles and joint angles (in a normalized coordinate frame that is centered, scaled and rotated to the camera viewpoint). We conclude that previous datasets make different 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 (Torralba and Efros 2011), 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-2014 (Qian et al. 2014), A-Star (Xu and Cheng 2013), and Dexter (Sridhar et al. 2013). While these test sets focus on estimating hand articulation, not all test sets contain the same amount of pose variation. For example, a sign language test set will exhibit a small number of discrete poses. To quantify articulation, we fit a multi-variate Gaussian distribution to a test set's finger joint angles. Then we compute the *differential entropy* for the test set's distribution:

$$h(\Sigma) = .5 \log\left((2\pi e)^N \det(\Sigma)\right) \quad (1)$$

where Σ is the covariance of the test set's joint angles and N is the number of joint angles in each pose vector. This analysis suggests that our proposed test set contains greater pose variation (entropy, $h = 89$) than the ICL ($h = 34$), NYU ($h = 82$), FORTH ($h = 65$) or A-STAR ($h = 79$) test sets. We focus on **ICL** (Tang et al. 2014) as a representative example for experimental evaluation because it has been used in multiple prior published works (Tang et al. 2014, D. Tang and Kim 2013, Oberweger et al. 2015a).

[1] http://www.ics.uci.edu/~jsupanci/#HandData



(a) Viewpoint Angle (b) Joint Articulation Angle

Fig. 2 Pose variation: We use MDS (multi-dimensional scaling) to plot the pose space covered by a set of hand datasets with compatible joint annotations. We split the pose space into two components and plot the camera viewpoint angles (a) and finger joint angles (b). For each testset, we plot the convex hull of its poses. In terms of joint angle coverage, most testsets are similar. In terms of camera viewpoint, some testsets consider a smaller range of views (e.g., ICL and A-STAR). We further analyze various assumptions made by datasets in the text.

Art. and viewpoint: Other testsets have focused on both viewpoint variation and articulation. FORTH (Oikonomidis et al. 2011) provides five test sequences with varied articulations and viewpoints, but these are unfortunately unannotated. The CVAR-EGO (Oberweger et al. 2016) dataset provides highly precise joint annotations but contains fewer frames and only one subject. In our experiments, we analyze the **NYU** dataset (Tompson et al. 2014) because of its wide pose variation (see Fig. 2), larger size, 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 performing activities and interacting with nearby objects and surfaces. The KTH Dataset (Pieropan et al. 2014) provides a rich set of 3rd person videos showing humans interacting with objects. Unfortunately, annotations are not provided for the hands (only the objects). Similarly, the LISA (Ohn-Bar and Trivedi 2014a) dataset provides cluttered scenes captured inside vehicles. However, joint positions are not annotated, only coarse gesture. The **UCI-EGO** (Rogez et al. 2014) dataset provides challenging sequences from an egocentric perspective with joint level annotations, 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. 3). As far as we are aware, our dataset is the first to focus on hand pose estimation *across multiple subjects and multiple cluttered scenes.*

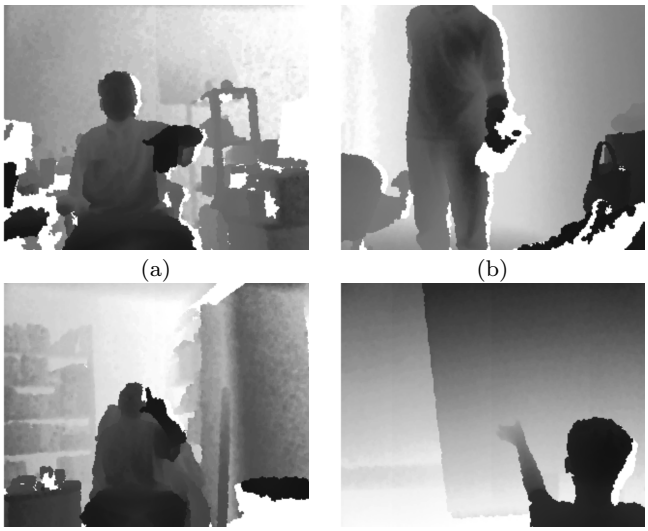


Fig. 3 Our new test data challenges methods with clutter (a), object manipulation (b), low-res (c), and various viewpoints (d). We collected data in diverse environments (8 offices, 4 homes, 4 public spaces, 2 vehicles, and 2 outdoors) using time-of-flight (Intel/Creative Gesture Camera) and structured-light (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.

This is important, because any practical application must handle diverse subjects, scenes, and clutter.

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 disagree on 20% of pose annotations (considering a 20mm threshold) as plotted in Fig. 19. These disagreements arise from limitations in the raw sensor data, either due to poor resolution or occlusions. We found that low resolution consistently corresponds to annotation ambiguities, across test sets. See Sec. 5.2) for further discussion and examples. These ambiguities are often mitigated by placing the hand close to the camera (Xu and Cheng 2013, Tang et al. 2014, Qian et al. 2014, Oberweger et al. 2016). As an illustrative example, we evaluate the **ICL** training set (Tang et al. 2014).

Dataset	Generation	Viewpoint	Views	Size	Subj.
ICL [59]	Real + manual annot.	3rd Pers.	1	331,000	10
NYU [63]	Real + auto annot.	3rd Pers.	3	72,757	1
HandNet [69]	Real + auto annot.	3rd Pers.	1	12,773	10
UCI-EGO [44]	Synthetic	Egocentric	1	10,000	1
libhand [67]	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.

Real data + automatic annotation: Data gloves directly obtain automatic pose annotations for real data (Xu and Cheng 2013). However, they require painstaking per-user calibration. Magnetic markers can partially alleviate calibration difficulties (Wetzler et al. 2015) but still distort the hand shape that is observed in the depth map. When evaluating depth-only systems, colored markers can provide ground-truth through the RGB channel (Sharp et al. 2015). Alternatively, one could use a "passive" motion capture system. We evaluate the larger **NYU** training set (Tompson et al. 2014) that annotates real data by fitting (offline) a skinned 3D hand model to high-quality 3D measurements. Finally, integrating model fitting with tracking lets one leverage a small set of annotated reference frames to annotate an entire video (Oberweger et al. 2016).

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 (Tang et al. 2014). 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 (Sridhar et al. 2015, Tompson et al. 2014). Finally, mimicking the noise and artifacts of real data is often important when using synthetic data. Domain transfer methods (D. Tang and Kim 2013) 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

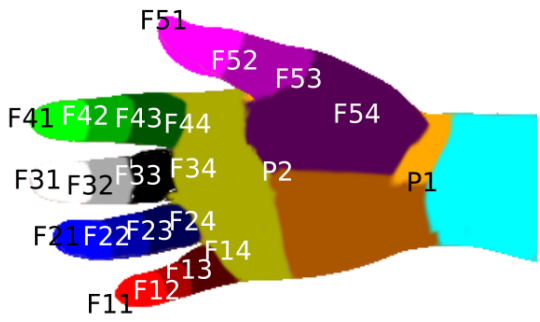


Fig. 4 libhand joints: We use the above joint identifiers to describe how we sample poses (for libhand) in table 3. Please see http://www.libhand.org/ for more details on the joints and their parameters.

sidesteps the annotation problem completely: precise annotations can be rendered along with the features. One can easily vary the size and shape of synthesized training hands, a fact which allows us to explore how user-specific training data impacts accuracy. Our experiments (ref. Sec. 6) verify that results may be optimistic when the training and test datasets contain the same individuals, as non-synthetic datasets commonly do (ref. Table 2). When synthesizing novel exemplars, it is important define a good sampling distribution. A common strategy for generating a sampling distribution is to collect pose samples with motion capture data (Castellini et al. 2011, Feix et al. 2013). The **UCI-EGO** training set (Rogez et al. 2014) synthesizes data with an egocentric prior over viewpoints and grasping poses.

3.1 libhand training set:

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 Fig. 7). We modified the code to include a forearm and output depth data, semantic segmentations, and keypoint annotations. We emphasize that this *synthetic* training set is distinct from our new test dataset of *real* images.

Synthesis parameters: To avoid biasing our synthetic training set away from unlikely, but possible, poses we do not use motion capture data. Instead, we take a brute-force approach based on rejection-sampling. We uniformly and independently sample joint angles (from a bounded range), and throw away invalid samples that yield self-intersecting 3D hand poses. Specifically, using the libhand joint identifiers shown in Fig. 4, we generate poses by uniformly sampling from bounded ranges, as shown in Table. 3.

Quasi-Synthetic backgrounds: Hand synthesis engines commonly under-emphasize the importance of image backgrounds (Šarić 2011, Oikonomidis et al. 2011, Tompson et al. 2014). For methods operating on pre-segmented images (Keskin et al. 2012, Sridhar et al. 2013, Qian et al. 2014), this is likely not an issue. However, for active hands "in-the-wild", the choice of synthetic backgrounds, surfaces, and interacting objects becomes important. Moreover, some systems require an explicit negative set (of images not containing hands) for training. To synthesize a robust background/negative training set, we take a quasi-synthetic approach by applying random affine transformations to 5,000 images of real scenes, yielding a total of 1,000,0000 pseudo-synthetic backgrounds. We found it useful to include human bodies in the negative set because faces are common distractors for hand models.

4 Methods

Next we survey existing approaches to hand pose estimation (summarized in Table 4). We conclude by introducing a novel volumetric nearest-neighbor (NN) baseline.

4.1 Taxonomy

Trackers versus detectors: We focus our analysis on single-frame methods. For completeness, we also consider several tracking baselines (Oikonomidis et al. 2011, PrimeSense 2013, Intel 2013) 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.

Discrete versus continuous pose: We further concentrate our analysis on the continuous pose regression problem. However historically, much prior work has tackled the problem from a discrete gesture classification perspective (Mo and Neumann 2006, PrimeSense 2013, Premaratne et al. 2010, Ohn-Bar and Trivedi 2014b). Yet, these perspectives are closely related because one can tackle continuous pose estimation using a large number of discrete classes. As such, we evaluate several discrete classifiers in our benchmark (Muja and Lowe 2014, Rogez et al. 2015a).

Description	Identifiers	bend	side	elongation
Intermediate and Distal Joints	$F_{1:4,2:3}$	$U(\frac{-\pi}{2}^r, \frac{\pi}{7}^r)$	0	0
Proximal-Carpal Joints	$F_{1:4,4}$	$U(\frac{-\pi}{2}^r, \frac{\pi}{7}^r)$	$U(\frac{-\pi}{8}^r, \frac{\pi}{8}^r)$	0
Thumb Metacarpal	$F_{5,4}$	$U(-1^r, .5^r)$	$U(-.7^r, 1.2^r)$	$U(.8^r, 1.2^r)$
Thumb Proximal	$F_{5,3}$	$U(-1^r, -.6^r)$	$U(-.2^r, .5^r)$	0
Wrist Articulation	P_1	$U(-1^r, 1^r)$	$U(-.5^r, .8^r)$	0

Table 3 Synthetic hand distribution: We render synthetic hands with joint angles sampled from the above uniform distributions. `bend` refers to the natural extension-retraction of the finger joints. The proximal-carpal, wrist and thumb joints are additionally capable of `side`-to-`side` articulation. We do not consider a third type of articulation, `twist`, because it would be extremely painful and result in injury. We model anatomical differences by `elongating` some bones fanning out from a joint. Additionally, we apply an isotropic global metric scale factor sampled from the range $U(\frac{2}{3}, \frac{3}{2})$. Finally, we randomize the camera viewpoint by uniformly sampling tilt, yaw and roll from $U(0, 2\pi)$.

Method	Approach	Model-drv.	Data-drv.	Detection	Implementation	FPS
Simulate [28]	Tracker (simulation)	Yes	No	Initialization	Published	50
NiTE2 [41]	Tracker (pose search)	No	Yes	Initialization	Public	> 60
Particle Swarm Opt. (PSO) [37]	Tracker (PSO)	Yes	No	Initialization	Public	15
Hough Forest [70]	Decision forest	Yes	Yes	Decision forest	Ours	12
Random Decision Forest (RDF) [23]	Decision forest	No	Yes	-	Ours	8
Latent Regression Forest (LRF) [59]	Decision forest	No	Yes	-	Published	62
DeepJoint [63]	Deep network	Yes	Yes	Decision forest	Published	25
DeepPrior [33]	Deep network	No	Yes	Scanning window	Ours	5000
DeepSegment [14]	Deep network	No	Yes	Scanning window	Ours	5
Intel PXC [21]	Morphology (convex detection)	No	No	Heuristic segment	Public	> 60
Cascades [44]	Hierarchical cascades	No	Yes	Scanning window	Provided	30
Ego. WS. [45]	Multi-class SVM	No	Yes	Whole volume classif.	Provided	275
EPM [73]	Deformable part model	No	Yes	Scanning window	Ours	1/2
Volumetric Exemplars	Nearest neighbor (NN)	No	Yes	Scanning volume	Ours	1/15

Table 4 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.

Data-driven versus model-driven: Historic attempts to estimate hand pose optimized a geometric model to fit observed data (Delamarre and Faugeras 2001, Bray et al. 2004, Stenger et al. 2006). Recently, Oikonomidis *et al.* (Oikonomidis et al. 2011) demonstrated hand tracking using GPU accelerated Particle Swarm Optimization (PSO). However, such optimizations remain notoriously difficult due to local minima in the objective function. As a result, model driven systems have historically found their successes mostly limited to the tracking domain, where initialization constrains the search space (Sridhar et al. 2013, Melax et al. 2013, Qian et al. 2014). For single image detection, various fast classifiers and regressors have obtained real-time speeds (Keskin et al. 2012, Intel 2013, Oberweger et al. 2015a, Oberweger et al. 2015b, Tang et al. 2015, Sun et al. 2015, Li et al. 2015, Wan et al. 2016). 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: Systems commonly separate their work into discrete stages: detecting, posing, refining and validating hands. Some systems use special purpose detectors as a "pre-processing" stage (Girard and Maciejewski 1985, Oikonomidis et al. 2011, Keskin et al. 2012, Cooper 2012, Xu and Cheng 2013, Intel 2013, Romero et al. 2009, Tompson et al. 2014). A segmentation pre-processing stage has been historically popular. Typically, RGB skin classification (Vezhnevets et al. 2003) or morphological operations on the depth image (Premaratne et al. 2010) segment the hand from the background. Such segmentation allows computation of Zernike moment (Cooper 2012) or skeletonization (Premaratne et al. 2010) features. While RGB features compliment depth (Rogez et al. 2014, Gupta et al. 2014), skin segmentation appears difficult to generalize across subjects and scenes with varying lighting (Qian et al. 2014). We evaluate a depth-based segmentation system (Intel 2013) for completeness. Other systems use a model for inverse-kinematics/IK (Tompson et al. 2014, Xu and Cheng 2013), geometric refinement/validation (Melax et al. 2013, Tang et al. 2015), or collaborative filtering (Choi et al. 2015) during a

"post-processing" stage. For highly precise hand pose estimation, recent hybrid pipelines compliment data-driven per-frame reinitialization with model-based refinement (Taylor et al. 2016, Ballan et al. 2012, Sridhar et al. 2015, Qian et al. 2014, Ye et al. 2016).

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** (Xu and Cheng 2013) take a two-stage approach of hand detection followed by pose estimation. **Random Decision Forests (RDFs)** (Keskin et al. 2012) and **Latent Regression Forests (LRFs)** (Tang et al. 2014) 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 (Yang and Ramanan 2013), but they appear somewhat rarely in the hand pose estimation literature. For completeness, we evaluate a deformable part model defined on depth image patches (Felzenszwalb et al. 2010). We specifically train an **exemplar part model (EPM)** constrained to model deformations consistent with 3D exemplars (Zhu et al. 2012).

Deep models: Recent systems have explored using deep neural nets for hand pose estimation. We consider three variants in our experiments. **DeepJoint** (Tompson et al. 2014) 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** (Oberweger et al. 2015a) 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** (Farabet et al. 2013) 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 (Shotton et al. 2013), 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.

$$(x, y, z) = \left(\frac{u}{f_u} D(u, v), \frac{v}{f_v} D(u, v), D(u, v) \right) \tag{2}$$

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:

$$(i^*, j^*) = \underset{i,j}{\operatorname{argmin}} \operatorname{Dist}(E_i, V_j) \qquad \text{where} \tag{3}$$

$$\operatorname{Dist}(E_i, V_j) = \sum_{x,y,z} \mathcal{I}(E_i[x, y, z] \neq V_j[x, y, z]), \tag{4}$$

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.* (Song and Xiao 2014) refer to such pruning strategies as "jumping window" searches.

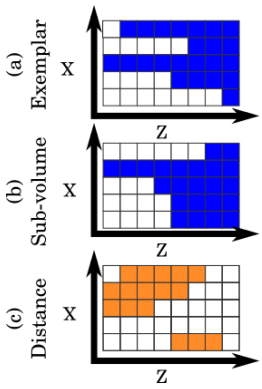


Fig. 5 Volumetric Hamming distance: We visualize 3D voxels corresponding to an exemplar (a) and subvolume (b). For simplicity, we visualize a 2D slice along a fixed y-value. Because occluded voxels are defined to be '1' (indicating they are occupied, shown in blue) the total Hamming distance is readily computed by the L1 distance between projections along the z-axis (c), mathematically shown in Eq.(5).

Second, one can compute volumetric Hamming distances with 2D computations:

$$\text{Dist}(E_i, V_j) = \sum_{x,y} |e_i[x,y] - v_j[x,y]| \qquad \text{where} \qquad (5)$$

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

Intuition for our encoding: Because our 3D volumes are projections of 2.5D measurements, they can be sparsely encoded with a 2D array (see Fig. 5). 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 (Moore et al. 2001, Muja and Lowe 2014).

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 (Janoch et al. 2013). First, our exemplars are defined in metric coordinates (Eq. 2). 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. 6).

5 Protocols

5.1 Evaluation

Reprojection error: Following past work, we evaluate pose estimation as a regression task that predicts a set of 3D joint locations (Oikonomidis et al. 2011, Keskin et al. 2012, Qian et al. 2014, Taylor et al. 2014, Tang et al. 2014). 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 (Šarić 2011). We then summarize performance by plotting the proportion of test frames whose average (or max) error falls below a threshold.

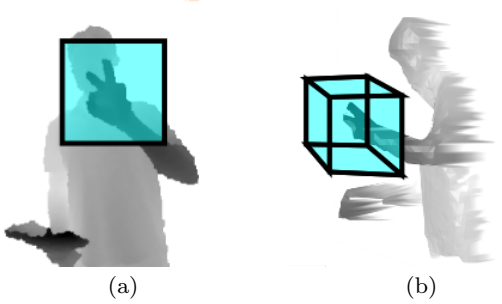


Fig. 6 Windows v. volumes: 2D scanning windows (a) versus 3D scanning volumes (b). Volumes can ignore background clutter that lies outside the 3D scanning volume but still falls inside its 2D projection. For example, when scoring the shown hand, a 3D volume will ignore depth measurements from the shoulder and head, unlike a 2D window.

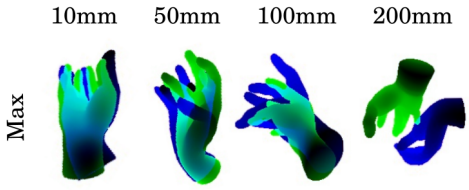


Fig. 7 Our error criteria: For each predicted hand, we calculate the average and maximum distance (in mm) between its skeletal joints and a ground-truth. In our experimental results, we plot the fraction of predictions that lie within a distance threshold, for various thresholds. This figure visually 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".

Error thresholds: Much past work considers performance at fairly low error thresholds, approaching 10mm (Xu and Cheng 2013, Tang et al. 2014, Tompson et al. 2014). Interestingly, (Oberweger et al. 2015a) 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. 19). 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. 7. **50mm** appears consistent with a roughly correct pose, while an

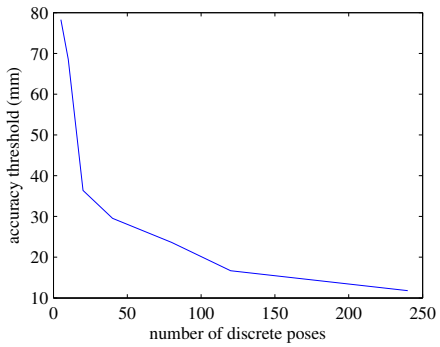


Fig. 8 Required precision per discrete pose: Larger pose vocabularies require more precision. We plot this relationship by considering the sparsest distribution of N poses. A max-joint-error precision of 20mm suffices to perfectly disambiguate a vocabulary of 100 discrete poses, while 10mm roughly disambiguates 240 poses. If perfect classification is not needed, one can enlarge the effective vocabulary size.

error within **100mm** appears consistent with a correct detection. Our qualitative analysis is consistent with empirical studies of human grasp (Bullock et al. 2013) and gesture communication (Stokoe 2005), which also suggest that a max-joint difference of 50mm differentiates common gestures and grasps. But in general, precision requirements depend greatly on the application; So we plot each method's performance across a broad range of thresholds (Fig. 8). We highlight 50 and 100mm thresholds for additional analysis.

Vocabulary size versus threshold: To better interpret max-error-thresholds, we ask "for a discrete vocabulary of N poses, what max-joint-error precision will suffice?". Intuitively, larger pose vocabularies require greater precision. To formalize this notion, we assume the user always perfectly articulates one of N poses from a discrete vocabulary Θ, with $|\Theta| = N$. Given a fixed vocabulary Θ, a recognition system needs to be precise within `prec` mm to avoid confusing any two poses from Θ:

$$\texttt{prec} < \min_{\theta_1 \in \Theta, \theta_2 \in \Theta} \frac{\text{dist}(P(\theta_1) - P(\theta_2))}{2} \tag{6}$$

where θ_1 and θ_2 represent two poses in Θ, $P(\theta)$ projects the pose θ's joints into metric space, and dist gives the maximum metric distance between the corresponding joints from each pose. To find the minimum precision required for each N, we construct a maximally distinguishable vocabulary Θ by maximizing the value of `prec`, subject to the kinematic constraints of libhand. Finding this most distinguishable pose vocabulary is an NP-hard problem. So, we take a greedy approach to optimize a vocabulary Θ for each vocabulary size N.

```
input  : predictions and ground truths for each image
output: a set of errors, one per frame
forall the test_images do
    P ← method's most confident prediction;
    G ← ground truths for the current test_image;
    if G = ∅ then
        /* Test Image contains zero hands      */
        if P = ∅ then
            errors ← errors ∪ {0};
        else
            errors ← errors ∪ {∞};
        end
    else
        /* Test Image contains hand(s)         */
        if P = ∅ then
            errors ← errors ∪ {∞};
        else
            best_error ← ∞;
            /* Find the ground truth best
               matching the method's prediction
               */
            forall the H ∈ G do
                /* For mean error plots, replace
                   max_i with mean_i            */
                /* V denotes the set of visible
                   joints                       */
                current_error ← max_{i∈V} ||H_i − P_i||_2;
                if current_error < best_error then
                    best_error ← current_error;
                end
            end
            errors ← errors ∪ {best_error};
        end
    end
end
```

Algorithm 1: Scoring Procedure: For each frame we compute a max or mean re-projection error for the ground truth(s) G and prediction(s) P. We later plot the proportion of frames with an error below a threshold, for various thresholds.

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 have released our evaluation software, we give pseudocode in Alg. 1.

Missing data: Another challenge with reprojection error is missing data. First, some methods predict 2D screen coordinates for joints, not 3D metric coordinates (Premaratne et al. 2010, Intel 2013, Farabet et al. 2013, Tompson et al. 2014). Approximating $z \approx D(u, v)$, inferring 3d joint positions should be straight-

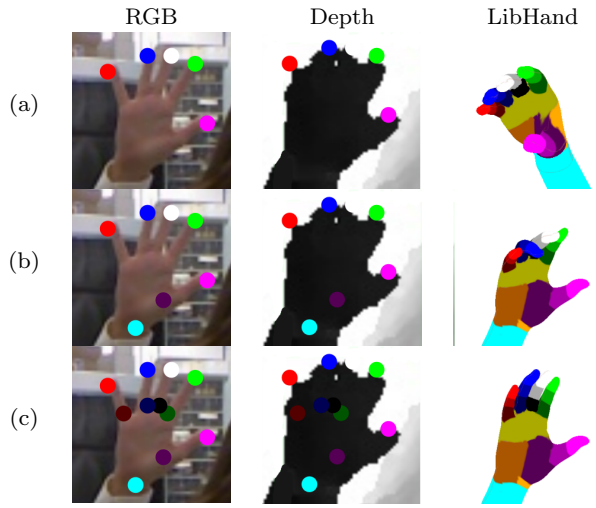


Fig. 9 Annotation procedure: We annotate until we are satisfied that the fitted hand pose matches the RGB and depth data. The first two columns show the image evidence presented and keypoints received. The right most column shows the fitted libhand model. The IK solver is able to easily fit a model to the five given keypoints (a), but it doesn't match the image well. The annotator attempts to correct the model (b), to better match the image, by labeling the wrist. Labeling additional finger joints finally yields and acceptable solution (c).

forward with Eq. 2. But, small 2D position errors can cause significant errors in the approximated depth, especially around the hand silhouette. To mitigate, we instead use the centroid depth of a segmented/detected hand when the measured depth lies outside the segmented volume. Past comparisons appear not to do this (Oberweger et al. 2015a), somewhat unfairly penalizing 2D approaches (Tompson et al. 2014). Second, some methods may predict a subset of joints (Intel 2013, Premaratne et al. 2010). To ensure a consistent comparison, we force such methods to predict the locations of visible joints with a post-processing inverse-kinematics (IK) stage (Tompson et al. 2014). 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 (Oikonomidis et al. 2011, PrimeSense 2013, Intel 2013). Some authors responded to our request for their code (Rogez et al. 2014). When software was not available, we attempted to re-implement methods ourselves. We were able to successfully reimplement (Keskin et al. 2012, Xu and Cheng 2013, Oberweger et al. 2015a), matching the accuracy on published results (Tang et al. 2014, Oberweger et al. 2015a). In other cases, our in-house implementations did not suffice (Tompson et al. 2014, Tang et al. 2014). 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 (Pang and Ling 2013)). We use our NN baseline to do precisely this. Finally, to spur further progress, *we have made our implementations publicly available, together with our evaluation code.*

5.2 Annotation

We now describe how we collect ground truth annotations. We present the annotator with cropped RGB and depth images. They then click semantic key-points, corresponding to specific joints, on either the RGB or depth images. To ease the annotator's task and to get 3D keypoints from 2D clicks we invert the forward rendering (graphics) hand model provided by libhand which projects model parameters θ to 2D keypoints $P(\theta)$. While they label joints, an inverse kinematic solver minimizes the distance between the currently annotated 2D joint labels, $\forall_{j\in J} L_j$, and those projected from the libhand model parameters, $\forall_{j\in J} P_j(\theta)$.

$$\min_{\theta} \sum_{j\in J} \|L_j - P_j(\theta)\|_2 \tag{7}$$

The currently fitted libhand model, shown to the annotator, updates online as more joints are labeled. When the annotator indicates satisfaction with the fitted model, we proceed to the next frame. We give an example of the annotation process in Fig. 9.

Strengths: Our annotation process has several strengths. First, kinematic constraints prevent some possible combination of keypoints: so it is often possible to fit the model by labeling only a subset of keypoints. Second, the fitted model provides annotations for occluded keypoints. Third and most importantly, the fitted model provides 3D (x,y,z) keypoint locations given only 2D (u,v) annotations.

Disagreements: As shown in in Fig. 19, annotators disagree substantially on the hand pose, in a surprising number of cases. In applications, such as sign language (Stokoe 2005) ambiguous poses are typically avoided. We believe it is important to acknowledge that, in general, it may not be possible to achieve full precision. For our proposed test set (with an average hand

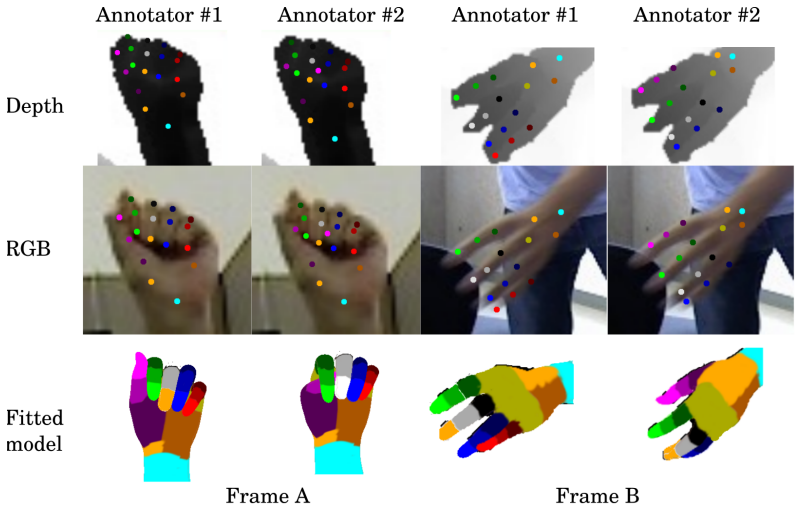


Fig. 10 Annotator disagreements: With whom do you agree? We show two frames where annotators disagree. The top two rows show the RGB and depth images with annotated keypoints. The bottom row shows the `libhand` model fit to those annotations. In *Frame A*, is the thumb upright or tucked underneath the fingers? In *Frame B*, is the thumb or pinky occluded? Long-range (low resolution) makes this important case hard to decide, In one author's opinion, annotator 1 is more consistent with RGB evidence while annotator 2 is more consistent with depth evidence (we always present annotators with both).

distance of 1100mm), we encountered an average annotation disagreement of about 20mm. For only nearby hands ($\leq$ 750mm from the camera, with an average distance of 550mm) we encountered an average annotation disagreement of about 10mm. The ICL dataset (Tang et al. 2014) exhibits similar annotation inconsistencies at similar ranges (Oberweger et al. 2015a). For hands at an average distance 235mm from the camera, (Oberweger et al. 2016) reduced annotation disagreements to approximately 4mm. This suggests that distance (which is inversely proportional to resolution) directly relates to annotation accuracy. Fig. 10 illustrates two examples of annotator disagreement on our test set.

6 Results

We now report our experimental results, comparing datasets and methods. We first address the "state of the problem": what aspects of the problem have been solved, and what remain open research questions? Fig. 11 qualitatively characterizes our results. We conclude by discussing the specific lessons we learned and suggesting directions for future systems.

Mostly-solved (distinct poses): Fig. 12 shows that coarse hand pose estimation is viable 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 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 (perhaps user-specific) or better model generalization should correct these. Yet, this may be reasonable for applications targeting sufficiently distinct poses from a small and finite vocabulary (e.g., a gaming interface). These results suggest that *the state-of-the-art can accurately predict distinct poses* (i.e. 50 mm apart) *in uncluttered scenes.*

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. 2). Fig. 20 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 (Tompson et al. 2014). This suggests that exemplars get most, but not all fingers, correct (see Fig. 13 and *cf.* Fig. 11 (c,*ii*) vs. (d,*ii*)). Overall, *we see noticeable progress on unconstrained pose estimation since 2007* (Erol et al. 2007).

Unsolved (low-res, objects, occlusions, clutter): When considering our testset (Fig. 19) with distant (low-res) hands and background clutter consisting of ob-

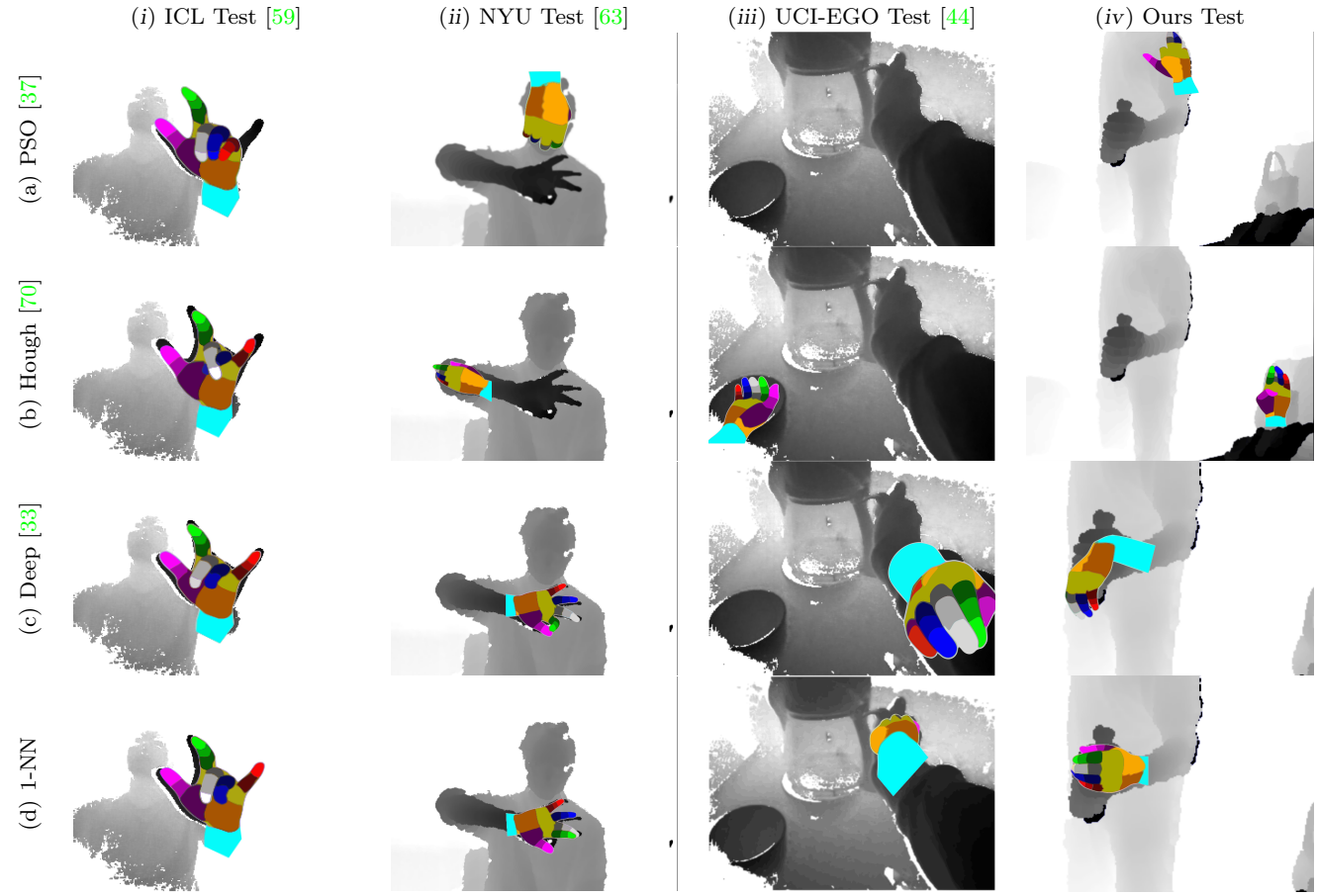


Fig. 11 Characteristic results: The PSO (Oikonomidis et al. 2011) tracker tends to miss individually extended fingers, in this case the pinky (a,*i*), due to local minima. Faces are common distractors for all methods. But, the PSO tracker in particular never recovers once it locks onto a face. The first-stage Hough forest (Xu and Cheng 2013) detector can recover from failures. But, the trees vote independently for global orientation and location using only local patch evidence. This local evidence seems insufficient to differentiate hands from elbows (b,*ii*) and other hand sized clutter (b,*iv*). The second-stage Hough (Xu and Cheng 2013) forests typically provide poorer finger-tip localization deeper inside the hand silhouette; here (b,*i*) they confuse the ring and middle finger because without global context the local votes are noisy and unspecific. NN exemplars most often succeeded in localizing the hand while the deep model (Oberweger et al. 2015a) more accurately estimated precise hand pose. See Sec. 6 for further discussion.

jects or interacting surfaces (Fig. 14), results are significantly worse. Note that many applications (Shotton et al. 2013) 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 (Keskin et al. 2012, Farabet et al. 2013), which may make an unrecoverable error in the first (detection and segmentation) stage. We show some illustrative examples in Fig. 15. 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. 19). This annotation uncertainty ("Human" in Fig. 19) makes it difficult to compare methods for highly precise pose estimation. As hand pose estimation systems become more precise, future work must make test data annotation more precise (Oberweger et al. 2016). Our results suggest that *scenes of in-the-wild hand activity are still beyond the reach of the state-of-the-art.*

Unsolved (Egocentric): The egocentric setting commonly presents (Fig. 17) the same problems discussed before, with the exception of low-res. While egocentric images do not necessarily contain clutter, most data in this area targets applications with significant clutter (see Fig. 16). And, in some sense, egocentric views make hand detection fundamentally harder. We cannot merely assume that the nearest pixel in the depth image corresponds to the hand, as we can with many 3rd person gesture test sets. In fact, the *forearm* often provides the primary salient feature. In Fig. 11 (c-d,*iii*) both the deep and the 1-NN models need the arm to

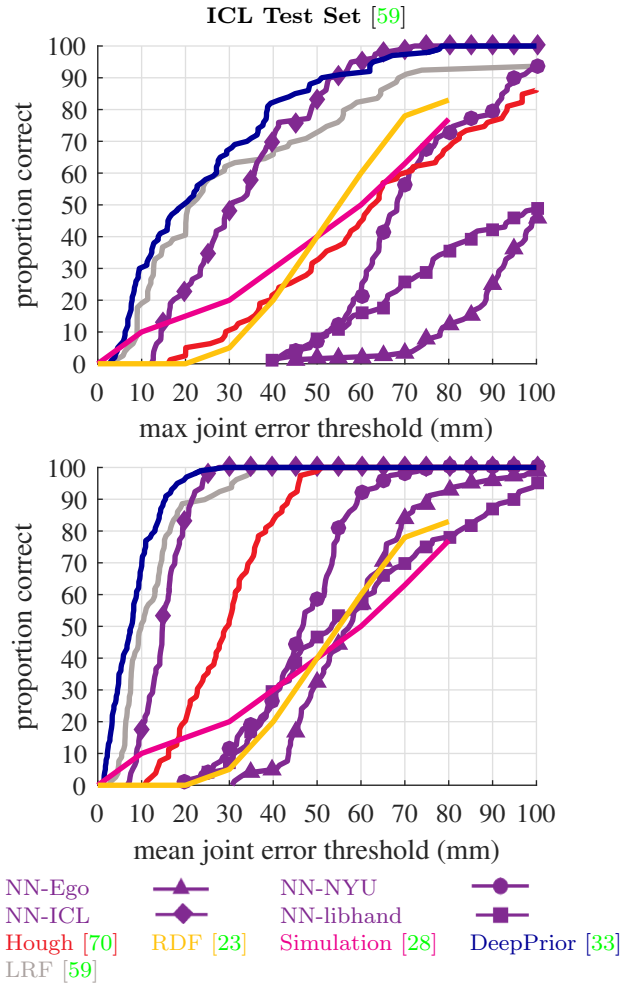


Fig. 12 We plot results for several systems on the ICL test-set using max-error (top) and average-error (bottom). Except for 1-NN, all systems are trained on the corresponding train set (in this case ICL-Train). To examine cross-dataset generalization, we also plot the performance of our NN-baseline 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.

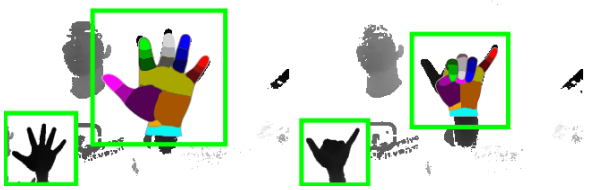


Fig. 13 Min vs max error: Compared to state-of-the-art, our 1-NN baseline often does relatively better under the average-error criterion than under the max-error criterion. When it can find (nearly) an exact match between training and test data (left) it obtains very low error. However, it does not generalize well to unseen poses (right). When presented with a new pose it will often place some fingers perfectly but others totally wrong. The result is a reasonable mean error but a high max error.

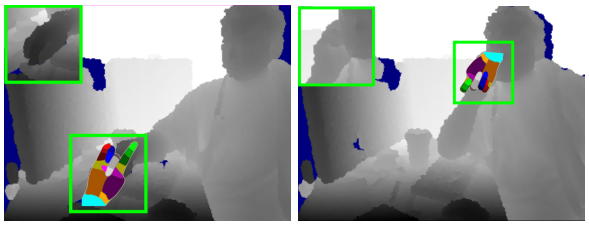


Fig. 14 Complex backgrounds: Most existing systems, including our own 1-NN baseline, fail when challenged with complex backgrounds which cannot be trivially segmented. These backgrounds significantly alter the features extracted and processed and thus prevent even the best models from producing sensible output.

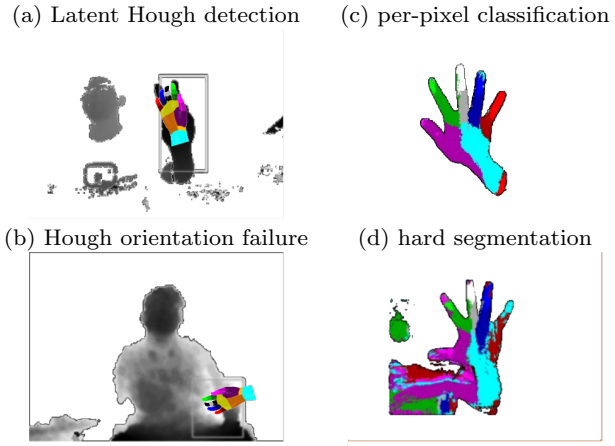


Fig. 15 Risks of multi-phase approaches: Many approaches to hand pose estimation divide into three phases: (1) detect and segment (2) estimate pose (3) validate or refine (Keskin et al. 2012, Intel 2013, Xu and Cheng 2013, Tompson et al. 2014, Tang et al. 2014). However, when an earlier stage fails, the later stages are often unable to recover. When detection and segmentation are non-trivial, this becomes to root cause of many failures. For example, **Hough forests** (Xu and Cheng 2013) (a) first estimate the hand's location and orientation. They then convert to a cardinal translation and rotation before estimating joint locations. (b) When this first stage fails, the second stage cannot recover. (c) Other methods assume that segmentation is solved (Keskin et al. 2012, Farabet et al. 2013), (d) when background clutter is inadvertently included by the hand segmenter, the finger pose estimator is prone to spurious outputs.

estimate the hand position. But, 1-NN wrongly predicts that the palm faces downwards, not towards the coffee maker. With such heavy occlusion and clutter, these errors are not surprising. The deep model's detector (Tompson et al. 2014, Oberweger et al. 2015a) proved less robust in the egocentric setting. Perhaps it developed sensitivity to changes in noise patterns, between the synthetic training and real test datasets. But, the NN and deep detectors wrongly assume translation-invariance for egocentric hands. Hand appearance and position are linked by perspective effects coupled with the kinematic constraints imposed by the arm. As a re-

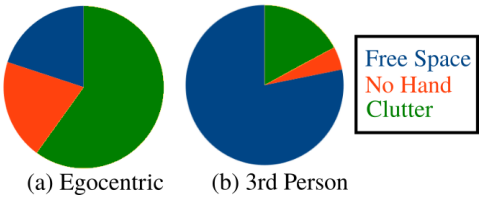


Fig. 16 Egocentric versus 3rd Person Challenges: A robust hand-pose estimator must contend with isolated hands in `free space`, frames with `no hands` visible, and hands grasping objects in `cluttered` scenes. Uniformly sampling frames from the test data in Table 1 we show the distribution of challenges for both Egocentric (UCI-EGO and CVAR-EGO) and 3rd person test sets. Empirically, egocentric data contains more object manipulation and occlusion. In general, egocentric datasets target applications which involve significant clutter(Rogez et al. 2015a, Li and Kitani 2013, Fathi et al. 2011, Rogez et al. 2015b). While, 3rd person test sets historically focus on gesture recognition, involving less clutter.

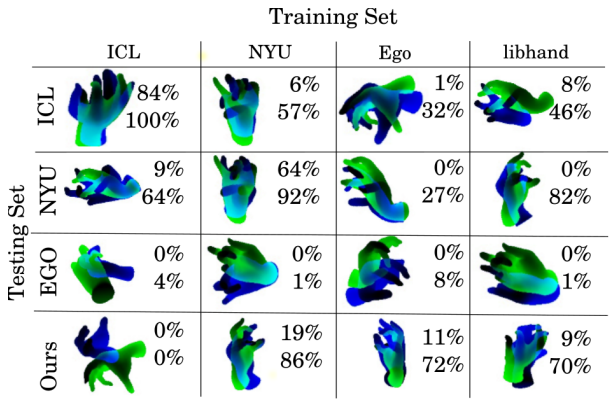


Table 5 Cross-dataset generalization: We compare training and test sets using a 1-NN classifier. Diagonal entries represent the performance using corresponding train and test sets. In each grid entry, we denote the percentage of test frames that are correct (50mm max-error, above, and 50mm average-error, below) and visualize the median error using the colored overlays from Fig. 7. We account for sensor specific noise artifacts using established techniques (Camplani and Salgado 2012). Please refer to the text for more details.

sult, an egocentric-specific whole volume classification model (Rogez et al. 2015a) outperformed both.

Training data: We use our NN-baseline to analyze the effect of training data in Table 5. Our NN model performed better using the NYU training set (Tompson et al. 2014) (consisting of real data automatically labeled with a geometrically-fit 3D CAD model) than with the libhand training set. While enlarging the synthetic training set increases performance (Fig. 18), computation fast becomes intractable. This reflects the difficulty in using synthetic data: one must carefully model priors (Oberweger et al. 2015a), sensor noise, (Gupta et al. 2014) and hand shape variations between

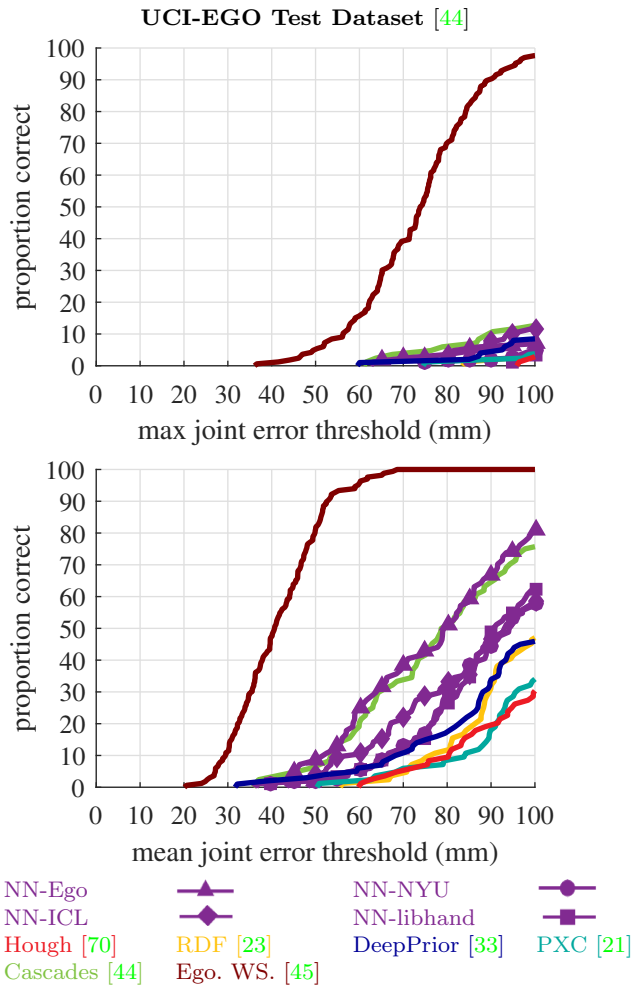


Fig. 17 For egocentric data, methods that classify the global scene (Rogez et al. 2015a) tend to outperform local scanning-window based approaches (including both deep and NN detectors). Rogez at al (Rogez et al. 2015a) make the argument that kinematic constrains from the arm imply that the location of the hand (in an egocentric coordinate frame) effects its local orientation and appearance, which in turn implies that recognition should not be translation-invariant. Still overall, performance is considerably worse than on other datasets. Egocentric scenes contain more background clutter and object/surface interactions, making even hand detection challenging for most methods.

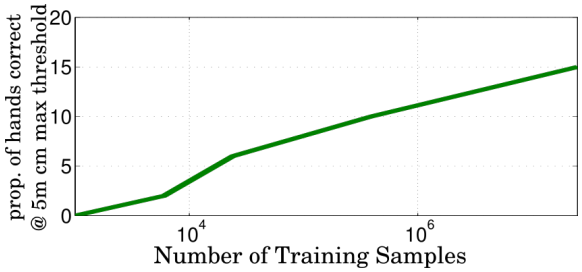


Fig. 18 Synthetic data vs. accuracy: Synthetic training set size impacts performance on our test testset. Performance grows logarithmically with the dataset size. Synthesis is theoretically unlimited, but practically becomes unattractively slow.

users (Taylor et al. 2014, Khamis et al. 2015). In Fig. 21 we explore the impact of each of these factors to uncover two salient conclusions: First, training with the test-time user's hand geometry (user-specific training data) showed modestly better performance, suggesting that results may be optimistic when using the same subjects for training and testing. Second, for synthetic hand data, modeling the pose-prior (i.e., choosing likely poses to synthesize) overshadows other considerations. Finally, 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. 12. Our results suggest the diversity and realism of the *training set is as important as the model learned from it.*

Surprising NN performance: Overall, our 1-NN baseline proved to be surprisingly potent, outperforming or matching the performance of most prior systems. This holds true even for moderately-sized training sets with tens of thousands of examples (Tompson et al. 2014, Tang et al. 2014), suggesting that simple memorization outperforms much prior work. To demonstrate generalization, future work on learning based methods will likely benefit from more and better training data. 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.*

NN vs Deep models: In fact, DeepJoint (Tompson et al. 2014) and DeepPrior (Oberweger et al. 2015a) were the sole approaches to significantly outperform 1-NN (Figs. 12 and 20). This indicates that deep architectures generalize well to novel test poses. Yet, the deep-model (Oberweger et al. 2015a) did show greater sensitivity to objects and clutter than the 1-NN model. We see this qualitatively in Fig. 11 (c-d,*iii-iv*) and quantitatively in Figs. 19 and 17. But, we can understand the deep-model's failures: we did not train it with clutter, so it "generalizes" that the bottle and hand are a single large hand. 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. Finally, the deep-model performed worse on more distant hands; this is understandable because it requires a larger canonical template (128x128) than the 1-NN model (30x30).

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 (Erol et al. 2007, Premaratne et al. 2010, Cooper 2012). 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. Thus, future work should investigate building large, realistic, and diverse training sets. 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: National Science Foundation Grant 0954083, Office of Naval Research-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. Ballan, L., Taneja, A., Gall, J., Gool, L. J. V., and Pollefeys, M. (2012). Motion capture of hands in action using discriminative salient points. In *ECCV (6)*. 7
2. Bray, M., Koller-Meier, E., Müller, P., Van Gool, L., and Schraudolph, N. N. (2004). 3D hand tracking by rapid stochastic gradient descent using a skinning model. In *1st European Conference on Visual Media Production (CVMP)*. 3, 6
3. Bullock, I. M., Member, S., Zheng, J. Z., Rosa, S. D. L., Guertler, C., and Dollar, A. M. (2013). Grasp Frequency and Usage in Daily Household and Machine Shop Tasks. *Haptics, IEEE Transactions on*. 9
4. Camplani, M. and Salgado, L. (2012). Efficient spatio-temporal hole filling strategy for kinect depth maps. In *Proceedings of SPIE*. 14
5. Castellini, C., Tommasi, T., Noceti, N., Odone, F., and Caputo, B. (2011). Using object affordances to improve object recognition. *Autonomous Mental Development, IEEE Transactions on*. 5
6. Choi, C., Sinha, A., Hee Choi, J., Jang, S., and Ramani, K. (2015). A collaborative filtering approach to real-time hand pose estimation. In *Proceedings of the*

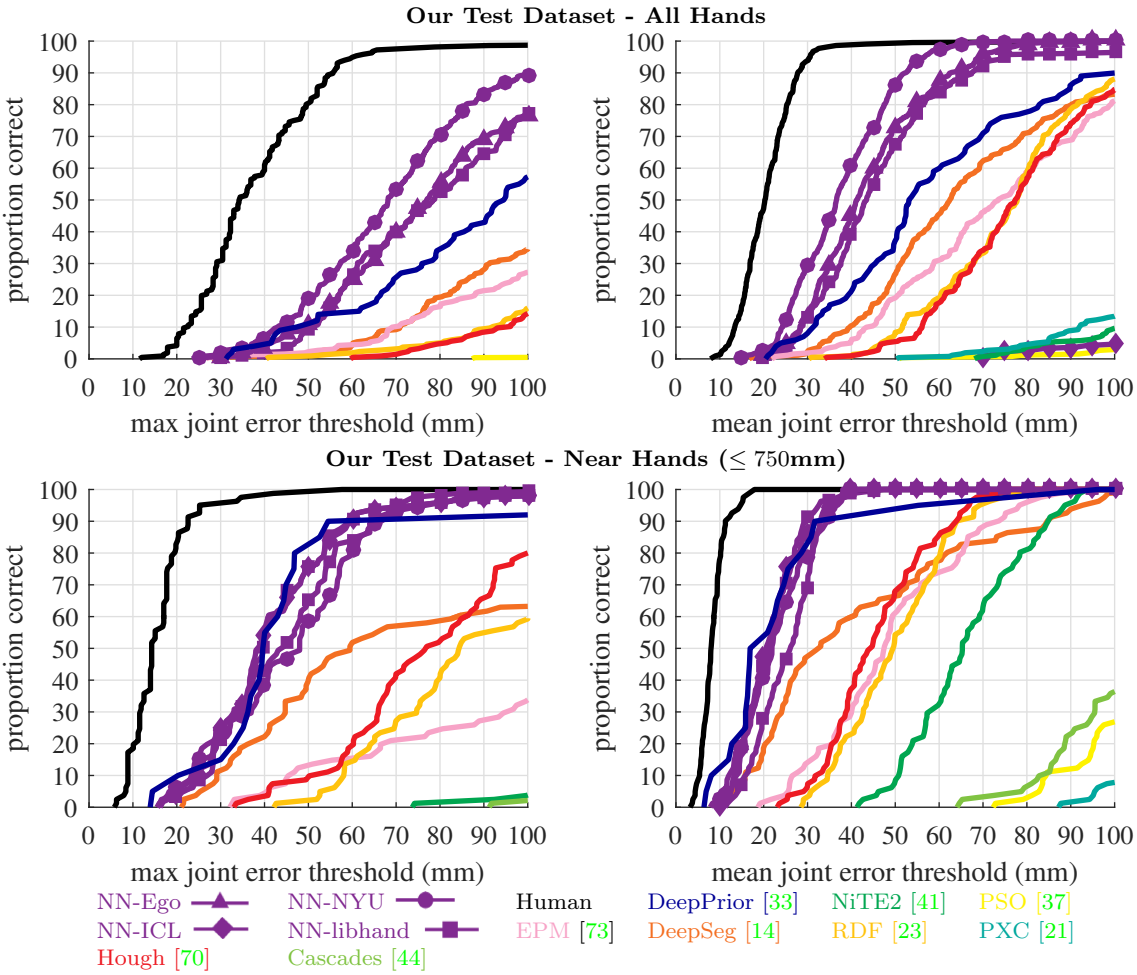


Fig. 19 We designed our dataset to address the remaining challenges of in "in-the-wild" hand pose estimation, including scenes with low-res hands, clutter, object/surface interactions, and occlusions. We plot human-level performance (as measured through inter-annotator agreement) in black. On nearby hands (within 750mm, as commonly assumed in prior work) our annotation quality is similar to existing testsets such as ICL (Oberweger et al. 2015a). This is impressive given that our testset includes comparatively more ambiguous poses (see Sec. 5.2). 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 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.

IEEE International Conference on Computer Vision, pages 2336–2344. 6

7. Cooper, H. (2012). Sign Language Recognition using Sub-Units. *The Journal of Machine Learning Research*. 6, 15
8. D. Tang, T. Y. and Kim, T.-K. (2013). Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *International Conference on Computer Vision (ICCV)*. 1, 3, 4
9. Delamarre, Q. and Faugeras, O. (2001). 3D Articulated Models and Multiview Tracking with Physical Forces. *Computer Vision and Image Understanding*. 3, 6
10. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 1
11. Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 2
12. Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. (2007). Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*. 2, 11, 15
13. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International journal of computer vision*. 1, 2

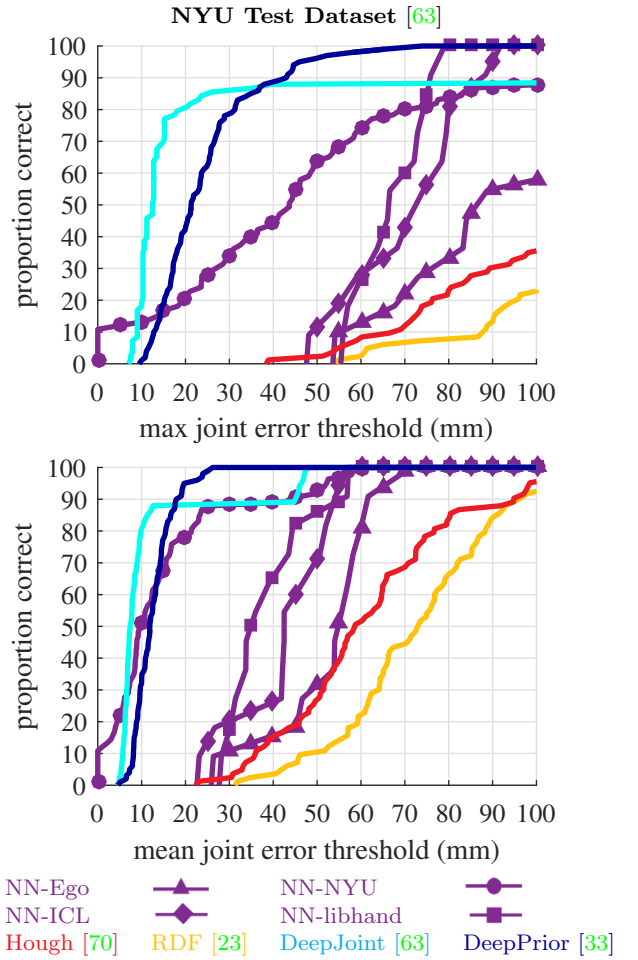


Fig. 20 Deep models (Tompson et al. 2014, Oberweger et al. 2015a) perform noticeably better than other systems, and appear to solve both articulated pose estimation and hand detection for uncluttered single-user scenes (common in the NYU testset). However, the other systems compare more favorably under average error. In Fig. 13, we interpret this disconnect by using 1-NN to show that each test hand commonly matches a training example in all but one finger. Please see text for further discussion.

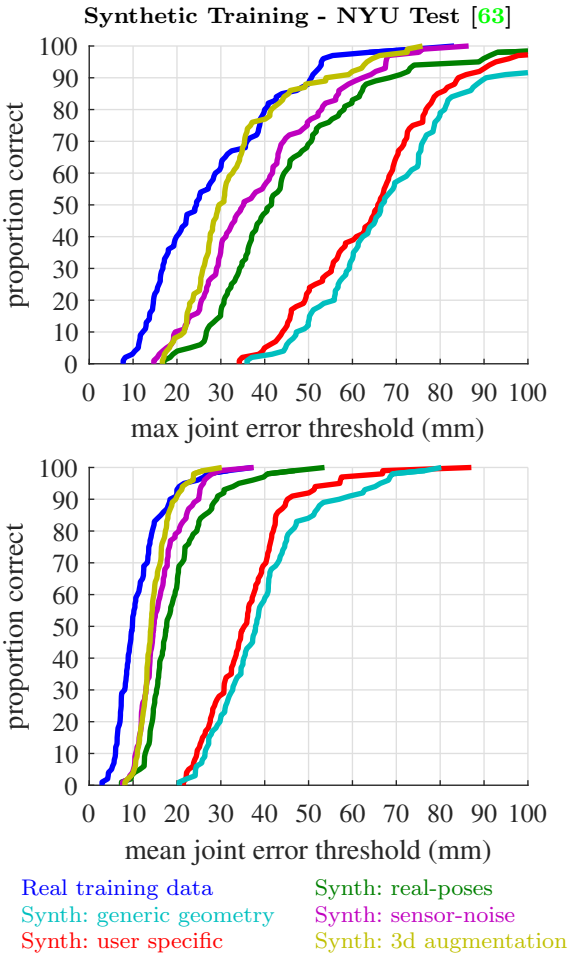


Fig. 21 Challenges of synthetic data: We investigate possible causes for our synthetic training data's lackluster performance. To do so, we synthesize a variety of training sets for a deep model (Oberweger et al. 2015a) and test on the NYU test set. Clearly, real training data (blue) outperforms our generic synthetic training set (cyan), as described in Sec. 3.1). By fitting our synthesis model's geometry to the test-time users we obtain a modest gain (red). However, the largest gain by far comes from synthesizing training data using only "realistic" poses, matching those from the NYU training set. By additionally modeling sensor noise (Gupta et al. 2014) we obtain the magenta curve. Finally, we almost match the real training data (yellow vs. blue) by augmenting our synthetic models of real-poses with out-of-plane rotations and foreshortening.

14. Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.* 6, 7, 9, 12, 13, 16
15. Fathi, A., Ren, X., and Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On*, pages 3281–3288. IEEE. 14
16. Fei-Fei, L., Fergus, R., and Perona, P. (2007). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding.* 1
17. Feix, T., Romero, J., Ek, C. H., Schmiedmayer, H., and Kragic, D. (2013). A Metric for Comparing the Anthropomorphic Motion Capability of Artificial Hands. *Robotics, IEEE Transactions on.* 5
18. Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.* 7

19. Girard, M. and Maciejewski, A. A. (1985). Computational Modeling for the Computer Animation of Legged Figures. *ACM SIGGRAPH Computer Graphics*. 6
20. Gupta, S., Girshick, R., Arbeláez, P., and Malik, J. (2014). Learning rich features from RGB-D images for object detection and segmentation. In *European Conference on Computer Vision (ECCV)*. Springer. 6, 14, 17
21. Intel (2013). Perceptual computing SDK. 5, 6, 9, 10, 13, 14, 16
22. Janoch, A., Karayev, S., Jia, Y., Barron, J. T., Fritz, M., Saenko, K., and Darrell, T. (2013). A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*. Springer London. 8
23. Keskin, C., Kıraç, F., Kara, Y. E., and Akarun, L. (2012). Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision (ECCV)*. 1, 5, 6, 7, 8, 10, 12, 13, 14, 16, 17
24. Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., and Fitzgibbon, A. (2015). Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2540–2548. 15
25. Li, C. and Kitani, K. M. (2013). Pixel-Level Hand Detection in Ego-centric Videos. *Computer Vision and Pattern Recognition (CVPR)*. 1, 14
26. Li, P., Ling, H., Li, X., and Liao, C. (2015). 3d hand pose estimation using randomized decision forest with segmentation index points. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 819–827. 6
27. Martin, D. R., Fowlkes, C. C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 2
28. Melax, S., Keselman, L., and Orsten, S. (2013). Dynamics based 3D skeletal hand tracking. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '13*. 1, 6, 13
29. Mo, Z. and Neumann, U. (2006). Real-time hand pose recognition using low-resolution depth images. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1499–1505. IEEE. 5
30. Moore, A. W., Connolly, A. J., Genovese, C., Gray, A., Grone, L., Kanidoris II, N., Nichol, R. C., Schneider, J., Szalay, A. S., Szapudi, I., et al. (2001). Fast algorithms and efficient statistics: N-point correlation functions. In *Mining the Sky*. Springer. 8
31. Muja, M. and Lowe, D. G. (2014). Scalable Nearest Neighbor Algorithms for High Dimensional Data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 5, 8
32. Oberweger, M., Riegler, G., Wohlhart, P., and Lepetit, V. (2016). Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4957–4965. 2, 3, 4, 11, 12
33. Oberweger, M., Wohlhart, P., and Lepetit, V. (2015a). Hands Deep in Deep Learning for Hand Pose Estimation. *Computer Vision Winter Workshop (CVWW)*. 1, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17
34. Oberweger, M., Wohlhart, P., and Lepetit, V. (2015b). Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324. 6
35. Ohn-Bar, E. and Trivedi, M. M. (2014a). Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations. *Intelligent Transportation Systems, IEEE Transactions on*. 1, 2, 3
36. Ohn-Bar, E. and Trivedi, M. M. (2014b). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377. 5
37. Oikonomidis, I., Kyriazis, N., and Argyros, A. (2011). Efficient model-based 3D tracking of hand articulations using kinect. In *British Machine Vision Conference (BMVC)*. 2, 3, 5, 6, 8, 10, 12, 16
38. Pang, Y. and Ling, H. (2013). Finding the Best from the Second Bests - Inhibiting Subjective Bias in Evaluation of Visual Tracking Algorithms. *International Conference on Computer Vision (ICCV)*. 1, 10
39. Pieropan, A., Salvi, G., Pauwels, K., and Kjellstrom, H. (2014). Audio-visual classification and detection of human manipulation actions. *In International Conference on Intelligent Robots and Systems (IROS)*. 2, 3
40. Premaratne, P., Nguyen, Q., and Premaratne, M. (2010). *Human computer interaction using hand gestures*. Springer. 5, 6, 9, 10, 15
41. PrimeSense (2013). Nite2 middleware. Version 2.2. 5, 6, 10, 16
42. Qian, C., Sun, X., Wei, Y., Tang, X., and Sun, J. (2014). Realtime and robust hand tracking from depth. In *Computer Vision and Pattern Recognition (CVPR)*. 1, 2, 3, 4, 5, 6, 7, 8
43. Ren, Z., Yuan, J., and Zhang, Z. (2011). Robust hand gesture recognition based on finger-earth

mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM international conference on Multimedia*. ACM. 1

44. Rogez, G., Khademi, M., Supancic, III, J., Montiel, J. M. M., and Ramanan, D. (2014). 3D hand pose detection in egocentric RGB-D images. *CDC4CV Workshop, European Conference on Computer Vision (ECCV)*. 2, 3, 4, 5, 6, 10, 12, 14, 16
45. Rogez, G., Supancic, III, J., and Ramanan, D. (2015a). First-person pose recognition using egocentric workspaces. In *Computer Vision and Pattern Recognition (CVPR)*. 5, 6, 14
46. Rogez, G., Supancic, J. S., and Ramanan, D. (2015b). Understanding everyday hands in action from rgb-d images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3897. 14
47. Romero, J., Kjellstr, H., and Kragic, D. (2009). Monocular Real-Time 3D Articulated Hand Pose Estimation. *Humanoid Robots, International Conference on*. 6
48. Russakovsky, O., Deng, J., Huang, Z., Berg, A. C., and Fei-Fei, L. (2013). Detecting avocados to zucchinis: what have we done, and where are we going? In *International Conference on Computer Vision (ICCV)*. IEEE. 2
49. Scharstein, D. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo. *International journal of computer vision*. 2
50. Shakhnarovich, G., Viola, P., and Darrell, T. (2003). Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision (ICCV)*. IEEE. 2
51. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., and Izadi, S. (2015). Accurate, robust, and flexible real-time hand tracking. In *Computer-Human Interaction, ACM Conference on*. 2, 4
52. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*. 7, 12
53. Song, S. and Xiao, J. (2014). Sliding Shapes for 3D Object Detection in Depth Images. *European Conference on Computer Vision (ECCV)*. 2, 7
54. Sridhar, S., Mueller, F., Oulasvirta, A., and Theobalt, C. (2015). Fast and robust hand tracking using detection-guided optimization. In *Computer Vision and Pattern Recognition (CVPR)*. 1, 4, 7
55. Sridhar, S., Oulasvirta, A., and Theobalt, C. (2013). Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. *International Conference on Computer Vision (ICCV)*. 2, 3, 5, 6
56. Stenger, B., Thayananthan, A., Torr, P. H. S., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical Bayesian filter. *Pattern Analysis and Machine Intelligence, IEEE transactions on*. 6
57. Stokoe, W. C. (2005). Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of deaf studies and deaf education*. 9, 10
58. Sun, X., Wei, Y., Liang, S., Tang, X., and Sun, J. (2015). Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832. 2, 6
59. Tang, D., Chang, H. J., Tejani, A., and Kim, T.-K. (2014). Latent regression forest: Structured estimation of 3D articulated hand posture. *Computer Vision and Pattern Recognition (CVPR)*. 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 15
60. Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.-K., and Shotton, J. (2015). Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3325–3333. 6
61. Taylor, J., Bordeaux, L., Cashman, T., Corish, B., Keskin, C., Sharp, T., Soto, E., Sweeney, D., Valentin, J., Luff, B., et al. (2016). Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143. 7
62. Taylor, J., Stebbing, R., Ramakrishna, V., Keskin, C., Shotton, J., Izadi, S., Hertzmann, A., and Fitzgibbon, A. (2014). User-specific hand modeling from monocular depth sequences. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 8, 15
63. Tompson, J., Stein, M., Lecun, Y., and Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *Graphics, ACM Transactions on*. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17
64. Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2, 3
65. Tzionas, D., Srikantha, A., Aponte, P., and Gall, J. (2014). Capturing hand motion with an RGB-D sensor, fusing a generative model with salient points. In *German Conference on Pattern Recognition (GCPR)*, Lecture Notes in Computer Science. Springer. 2

66. Vezhnevets, V., Sazonov, V., and Andreeva, A. (2003). A survey on pixel-based skin color detection techniques. In *Proc. Graphicon*. Moscow, Russia. 6
67. Šarić, M. (2011). Libhand: A library for hand articulation. Version 0.9. 4, 5, 8
68. Wan, C., Yao, A., and Van Gool, L. (2016). Hand pose estimation from local surface normals. In *European Conference on Computer Vision*, pages 554–569. Springer. 6
69. Wetzler, A., Slossberg, R., and Kimmel, R. (2015). Rule of thumb: Deep derotation for improved fingertip detection. In *British Machine Vision Conference (BMVC)*. BMVA Press. 2, 4
70. Xu, C. and Cheng, L. (2013). Efficient Hand Pose Estimation from a Single Depth Image. *International Conference on Computer Vision (ICCV)*. 1, 2, 3, 4, 6, 7, 8, 10, 12, 13, 14, 16, 17
71. Yang, Y. and Ramanan, D. (2013). Articulated pose estimation with flexible mixtures-of-parts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 7
72. Ye, Q., Yuan, S., and Kim, T.-K. (2016). Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *European Conference on Computer Vision*, pages 346–361. Springer. 7
73. Zhu, X., Vondrick, C., Ramanan, D., and Fowlkes, C. (2012). Do we need more training data or better models for object detection?. In *British Machine Vision Conference (BMVC)*. 6, 7, 16