BigHand2.2M Benchmark: Hand Pose Dataset and State of the Art Analysis

Shanxin Yuan[1*] Qi Ye[1*] Björn Stenger[2] Siddhant Jain[3] Tae-Kyun Kim[1]

[1]Imperial College London [2]Rakuten Institute of Technology [3]IIT Jodhpur

Abstract

In this paper we introduce a large-scale hand pose dataset, collected using a novel capture method. Existing datasets are either generated synthetically or captured using depth sensors: synthetic datasets exhibit a certain level of appearance difference from real depth images, and real datasets are limited in quantity and coverage, mainly due to the difficulty to annotate them. We propose a tracking system with six 6D magnetic sensors and inverse kinematics to automatically obtain 21-joints hand pose annotations of depth maps captured with minimal restriction on the range of motion. The capture protocol aims to fully cover the natural hand pose space. As shown in embedding plots, the new dataset exhibits a significantly wider and denser range of hand poses compared to existing benchmarks. Current state-of-the-art methods are evaluated on the dataset, and we demonstrate significant improvements in cross-benchmark performance. We also show significant improvements in egocentric hand pose estimation with a CNN trained on the new dataset.

1. Introduction

There has been significant progress in the area of hand pose estimation in the recent past and a number of systems have been proposed [5, 8, 11, 14, 15, 18, 24, 30, 7, 1, 4, 33, 6]. However, as noted in [12], existing benchmarks [18, 26, 28, 31, 34] are restricted in terms of number of annotated images, annotation accuracy, articulation coverage, and variation in hand shape and viewpoint.

The current state of the art for hand pose estimation employs deep neural networks to estimate hand pose from input data [31, 13, 14, 40, 2, 38]. It has been shown that these methods scale well with the size of the training dataset. The availability of a large-scale, accurately annotated dataset is therefore a key factor for advancing the field. Manual annotation has been the bottleneck for creating large-scale benchmarks [18, 25]. This method is not only labor-intensive, but can also result in inaccurate position labels. Semi-automatic capture methods have been devised where 3D joint locations are inferred from manually annotated 2D joint locations [12]. Alternatives, which are still time-consuming, combine tracking a hand model and manual refinement, if necessary iterating these steps [26, 28, 31]. Additional sensors can aid automatic capture significantly, but care must be taken not to restrict the range of motion, and to minimise the depth appearance difference from bare hands, for example when using a data-glove [36]. More recently, less intrusive magnetic sensors have been employed for finger tip annotation in the *HandNet* dataset [34].

In this paper, we introduce our million-scale *BigHand2.2M* dataset that makes a significant advancement in terms of completeness of hand data variation and annotation quality, see Figure 1 and Table 1. We detail the capture set-up and methodology that enables efficient hand pose annotation with high accuracy. This enables us to capture the range of hand motions that can be adopted without external forces. Our dataset contains 2.2 million depth maps with accurately annotated joint locations. The data is captured by attaching six magnetic sensors on the hand, five on each finger nail and one on the back of the palm, where each sensor provides accurate 6D measurements. Locations of all joints are obtained by applying inverse kinematics on a hand model with 31 degrees of freedom (dof) with kinematic constraints. The *BigHand2.2M* dataset also contains 290K frames of egocentric hand poses, which is 130 times more than previous egocentric hand pose datasets (Table 2). Training a Convolutional Neural Network (CNN) on the data shows significantly improved results. The recent study by Supancic *et al.* [27] on cross-benchmark testing showed that approximately 40% of poses are estimated with an error larger than 50mm. This is due to a different capture set-up, hand shape variation, and annotation schemes. Training a CNN using the *BigHand2.2M* dataset, we demonstrate state-of-the-art performance on existing benchmarks, 15-20mm average errors.

2. Existing benchmarks

Existing benchmarks for evaluation and comparison are significantly limited in scale (from a few hundred to tens

*indicates equal contribution

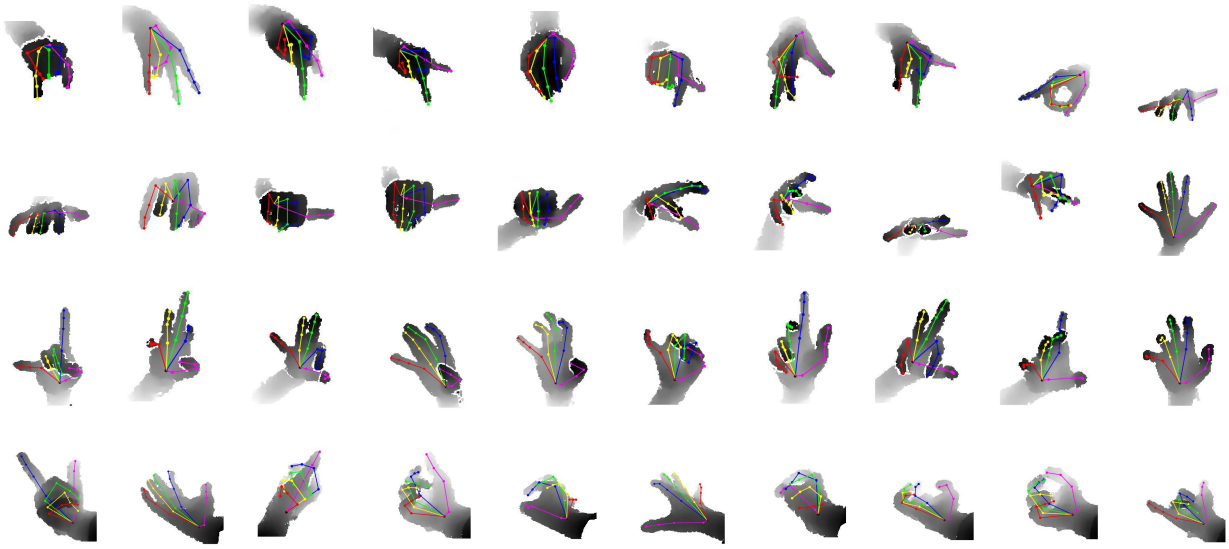

Figure 1: Example images from the *BigHand2.2M* dataset. The dataset covers the range of hand poses that can be assumed without applying external forces to the hand. The accuracy of joint annotations is higher than in previous benchmark datasets.

Dataset	Annotation	No. frames	No. joints	No. subjects	View point	Depth map resolution
Dexter 1 [25]	manual	2,137	5	1	3rd	320×240
MSRA14 [18]	manual	2,400	21	6	3rd	320×240
ICVL [28]	track + refine	17,604	16	10	3rd	320×240
NYU [31]	track + refine	81,009	36	2	3rd	640×480
MSRA15 [26]	track + refine	76,375	21	9	3rd	320×240
UCI-EGO [20]	semi-automatic	400	26	2	ego	320×240
Graz16 [12]	semi-automatic	2,166	21	6	ego	320×240
ASTAR [36]	automatic	870	20	30	3rd	320×240
HandNet [34]	automatic	212,928	6	10	3rd	320×240
MSRC [24]	synthetic	102,000	22	1	3rd	512×424
BigHand2.2M	automatic	2.2M	21	10	full	640×480

Table 1: Benchmark comparison. Existing datasets are limited in the number of frames, due to their annotation methods, either manual or semi-automatic. Our automatic annotation method allows collecting fully annotated depth images at frame rate. Our dataset is collected with the latest Intel RealSense SR300 camera [3], which captures depth images at 640×480-pixel resolution.

of thousands), annotation accuracy, articulation, view point, and hand shape [18, 24, 26, 28, 31, 34, 12]. The bottleneck for building a large-scale benchmark using captured data is the lack of a rapid and accurate annotation method. Creating datasets by manual annotation [18, 25] is labor-intensive and can result in inaccurate labels. These benchmarks are small in size, *e.g.* MSRA14[18] and Dexter 1 [25] have only 2,400 and 2,137 frames, respectively, making them unsuitable for large-scale training. Alternative annotation methods, which are still labor-intensive and time-consuming, track a hand model and manually refine the results, if necessary they iterating these two steps [26, 28, 31]. The ICVL dataset [28] is one of the first benchmarks and it was annotated using 3D skeletal tracking [9] followed by manual refinement. However, its scale is small and limitations of the annotation accuracy have been noted in the literature [26, 12]. The NYU dataset [31] is larger with a wider range of view points. Its annotations were obtained by model-based hand tracking on depth images from three cameras. Particle Swarm Optimization was used to obtain the final annotation. This method often drifts to incorrect poses, where manual correction is needed to re-initialize the tracking process. The MSRA15 dataset [26] is currently the most complex in the area [12]. It is annotated in an iterative way, where an optimization method [18] and manual re-adjustment alternate until convergence. The annotation

also contains errors, such as occasionally missing finger and thumb annotations. This benchmark has a large view point coverage, but it has only small variation in articulation, capturing 17 base articulations and varying each of them within a 500-frame sequence.

Two small datasets were captured using semi-automatic annotation methods [12, 20]. The UCI-EGO dataset [20] was annotated by iteratively searching for the closest example in a synthetic set and subsequent manual refinement. The Graz16 dataset [12] was annotated by iteratively annotating visible joints in a number of key frames and automatically inferring the complete sequence using an optimization method, where the appearance as well as temporal, and distance constraints are exploited. However, it remains challenging to annotate rapidly moving hands. It also requires manual correction when optimization fails. This semi-automatic method resulted in a 2,166-frame annotated egocentric dataset, which is also insufficient for large-scale training.

Additional sensors can aid automatic capture significantly [17, 32, 34, 36], but care must be taken not to restrict the range of motion. The ASTAR dataset [36] used a *ShapeHand* data-glove[23], but wearing the glove influences the captured hand images, and to some extent hinders free hand articulation. In the works of [17, 32], full human body pose esimtation was treated as a state estimation problem given magnetic sensor and depth data. More recently, less intrusive magnetic sensors have been used for finger tip annotation in the HandNet dataset [34], which exploits a similar annotation setting as our benchmark with trakSTAR magnetic sensors [10]. However, this dataset only provides fingertip locations, not the full hand annotations.

Synthetic data has been exploited for generating training data [19, 21, 37], as well as evaluation [24]. Even though one can generate unlimited synthetic data, there currently remains a gap between synthetic and real data. Apart from differences in hand characteristics and the lack of sensor noise, synthetically generated images sometime produce kinematically implausible and unnatural hand poses, see Figure 10. The MSRC benchmark dataset [24] is a synthetic benchmark, where data is uniformly distributed in the 3D view point space. However, the data is limited in the articulation space, where poses are generated by random sampling from six articulations.

3. Full hand pose annotation

In this section we present our method to do accurate full hand pose annotations using the trakSTAR tracking system with 6D magnetic sensors.

3.1. Annotation by inverse kinematics

Our hand model has 21 joints and can move with 31 degrees of freedom (dof), as shown in Figure 2. We capture

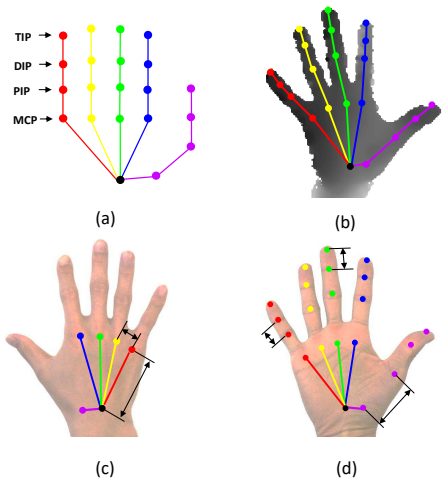

Figure 2: Hand model definition. (a) Our hand model has 21 joints and moves with 31 degrees of freedom (dof). (b) Model fitted to hand shape. (c) and (d) show how hand shape is measured.

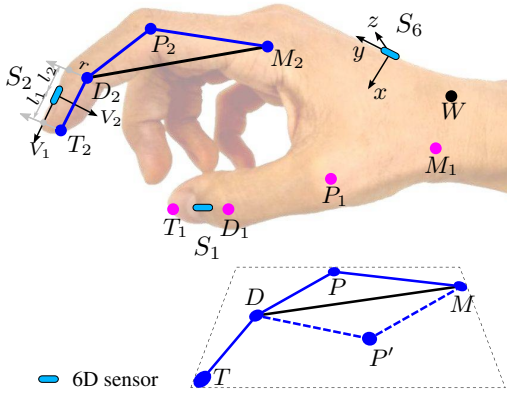

Figure 3: Hand pose inference using six 6D magnetic sensors. Global hand pose can be inferred from the location and orientation of sensor S6 on the back of the palm. Each sensor on the nail is used to infer the TIP and DIP joints of the corresponding finger. Each PIP joint can be calculated using bone lengths and physical constraints.

31 dimensions, including 6 dimensions for global pose and 25 joint angles. Each finger's pose is represented by five angles, including the twist angle, flexion angle, abduction angle for the MCP joint and flexion angles for the DIP and PIP joints. For each subject, we manually measure the bone lengths, see Figure 2 (c) and (d).

Given the six magnetic sensors, each with 6D dof (location and orientation), along with a hand model, we use inverse kinematics to infer the full hand pose defined by the locations of 21 joints, as shown in Figure 2. The physical constraints per subject are (1) the wrist and 5 MCP joints

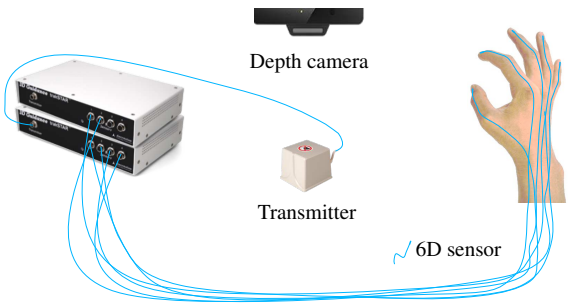

Figure 4: Annotation settings. The equipment used in our annotation system are: Two hardware synchronized electromagnetic tracking units, six 6D magnetic sensors, one mid-range transmitter, and an Intel RealSense SR300 camera.

are fixed relative to each other, (2) bone lengths are constant, and (3) MCP, PIP, DIP, and TIP joints for each finger lie on the same plane.

Similar to [22] five magnetic sensors (from thumb to little finger, the sensors are S1, S2, S3, S4, S5) are attached on the five fingers' tips. The sixth sensor (S6) is attached to the back of the palm, see Figure 3. Given the location and orientation of S6, as well as the hand model shape, the wrist (W) and five MCP joints (M1, M2, M3, M4, M5) are inferred. For each finger, given the sensor's location and orientation, the TIP and DIP are calculated in the following way (as shown in Figure 3, take the index finger as an example): the sensor's orientation is used to find the three orthogonal axes, V_1 is along the finger, V_2 is pointing forward from the finger tip. The TIP (T) and DIP (D) joint locations are calculated as:

$$T = L(S) + l_1 V_1 + r V_2, \quad (1)$$

$$D = L(S) - l_2 V_1 + r V_2, \quad (2)$$

where $L(S)$ denotes the sensor location, r is half the finger thickness, and l_1+l_2 = b, where b is the bone length connecting the DIP and TIP joints. The final joint to infer is the PIP, shown at location P in Figure 3, is calculated using the following conditions: (1) T, M, D are given, (2) $\|P - D\|$ and $\|P - M\|$ are fixed, (3) T, D, P, M are on the same plane, and (4) T and P should be on different sides of the line connecting M and D. These constraints are sufficient to uniquely determine P.

3.2. Synchronization and calibration

To build and annotate our dataset, we use a trakSTAR tracking system [10] combined with the latest generation Intel RealSense SR300 depth sensor [3], see Figure 4. The trakSTAR system consists of two hardware synchronized electromagnetic tracking units, each of which can track up to four 6D magnetic sensors. The 6D sensor ("Model 180") is 2mm wide and is attached to a flexible 1.2mm wide and 3.3m long cable. When the cable is attached to the hand using tight elastic loops the depth images and hand movements are minimally affected. We use the mid-range transmitter with a maximum tracking distance of 660mm, which is suitable for hand tracking. The tracking system captures the locations and orientations of the six sensors at 720fps and is stable and without drift in continuous operation. The depth camera captures images with a resolution of 640×480 and runs at a maximum speed of 60fps. The measurements are synchronized by finding the nearest neighboring time stamps. The time gap between the depth image and the magnetic sensors in this way is 0.7 millisecond at most.

The trakStar system and the depth sensor have their own coordinate systems, and we use a solution to the perspective-N-point problem to calibrate the coordinates as in [34]. Given a set of 3D magnetic sensor locations and the corresponding 2D locations in the depth map as well as intrinsic camera parameters, the ASPnP algorithm [39] estimates the transformation between these two coordinate systems.

Benchmarks	Rogez [20]	Oberweger [12]	BigHand2.2M Egocentric
No. Frames	400	2166	290K

Table 2: Egocentric Benchmark size comparison. The egocentric subset of *BigHand2.2M* dataset is 130 time larger than the next largest available dataset.

4. BigHand2.2M benchmark

We collected the *BigHand2.2M* dataset containing 2.2 million depth images of single hands with annotated joints (see Section 3). Ten subjects (7 male, 3 female) were captured for two hours each.

4.1. Hand view-point space

In order to cover diverse view points, we vary the sensor height, the subject's position and arm orientation. The view point space (a hemisphere for the 3rd person view point) is divided into 16 regions (4 regions uniformly along each of two 3D rotation axes), and subjects are instructed to carry out random view point changes within each region. In addition, our dataset collects random changes in the egocentric view point. As the t-SNE visualization in Figure 5(left) shows, our benchmark data covers a significantly larger region of the global view-point space than the ICVL and NYU dataset.

4.2. Hand articulation space

Similar to [35], we define 32 *extremal poses* as hand poses where each finger assumes a maximally bent or ex-

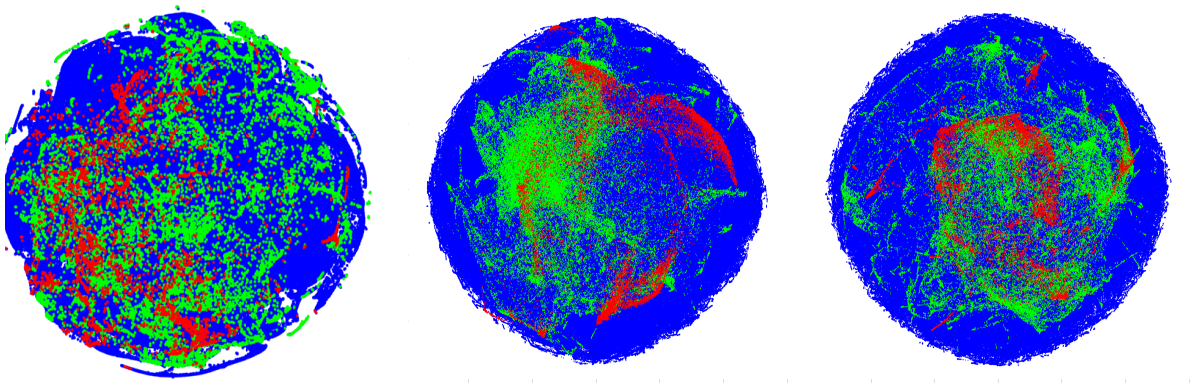

Figure 5: 2D t-SNE embedding of the hand pose space. *BigHand2.2M* is represented by blue, ICVL by red, and NYU by green dots. The figures show (left) global view point space coverage, (middle) articulation space (25D), and (right) combined of global orientation and articulation coverage. Compared with existing datasets, the *BigHand2.2M* contains a more complete range of variation.

tended position. For maximum coverage of the articulation space, we enumerate all $\binom{32}{2} = 496$ possible pairs of these *extremal poses*, and capture the natural motion when transitioning between the two poses of each pair.

In total the *BigHand2.2M* dataset consists of three parts: (1) Schemed poses: to cover all the articulations that a human hand can freely adopt, this contains has 1.534 million frames, captured as described above. (2) Random poses: 375K frames are captured with participants being encouraged to fully explore the pose space. (3) Egocentric poses: 290K frames of egocentric poses are captured with subjects carrying out the 32 *extremal poses* combined with random movements.

As Figure 5 (middle, right) shows, our benchmark spans a wider and denser area in the articulation and the combined of articulation and view-point space, compared to the ICVL and NYU.

4.3. Hand shape space

We select ten participants with different hand shapes (7 male, 3 female, age range: 25-35 years). Existing benchmarks also use different participants, but are limited in annotated hand shapes due to annotation methods. Figure 6 visualizes shapes in different datasets using the first two principal components of the hand shape parameters. The ICVL dataset [28] includes ten participants with similar hand size, and all are annotated with a single hand shape model. The NYU [31] training data uses one hand shape, while its test data uses two hand shapes, one of which is from the training set. The MSRA15 dataset includes nine participants, but in the annotated ground truth data, only three hand shapes are used. The MSRC [24] synthetic benchmark includes a single shape.

In the experiments, we use the dataset of 10 subjects for training, and testify how well the learnt model generalises to different shapes in existing benchmarks (cross-benchmark) and an unseen new shape ("new person" in Figure 6). See section 5 for more explanations.

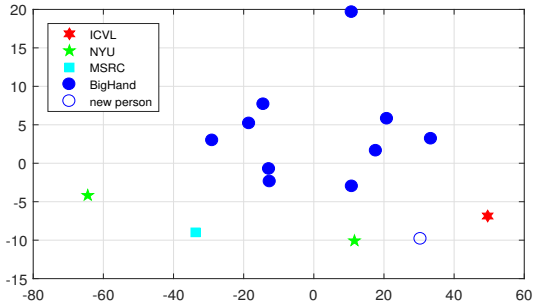

Figure 6: Hand shape variation. Hand variation is visualized by applying PCA to the shape parameters. The *BigHand2.2M* dataset contains 10 hand shapes and an additional shape for testing. The ICVL dataset contains one hand shape due to its annotation method. The NYU dataset includes two hand shapes, the MSRC dataset includes one synthetic hand shape.

5. Analysis of the state of the art

In this section we use the *Holi* CNN architecture [38] as the current state of the art. The detailed structure is shown in the supplementary material. The input for the CNN model is the cropped hand area using the ground truth joint locations. This region is normalized to 96 $\times$ 96 pixels and is fed into the CNN, together with two copies downsampled to 48 $\times$ 48 and 24 $\times$ 24. The cost function is the mean squared distance between the location estimates and the ground truth locations. The CNN is implemented using Theano and is trained on a desktop with an Nvidia GeForce GTX TITAN

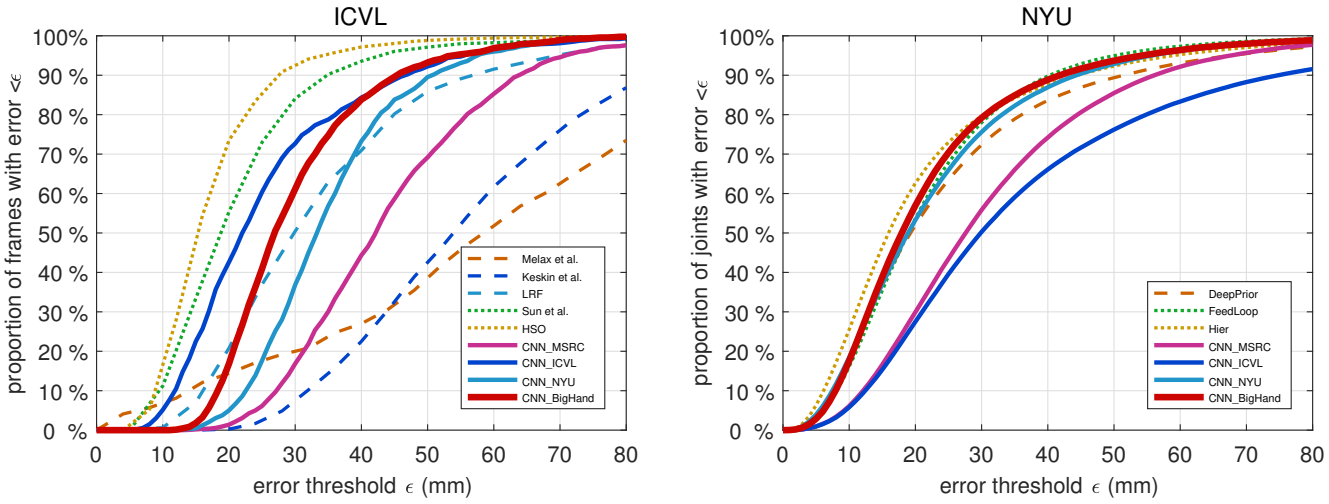

Figure 7: Cross-benchmark performance. CNN models are trained on the ICVL, NYU, MSRC, and the new *BigHand2.2M* dataset, respectively, and evaluated on (left) ICVL and (right) NYU test data. A CNN trained on *BigHand2.2M* achieves state-of-the-art performance on ICVL and NYU, while the CNNs trained on ICVL, NYU, and MSRC do not generalize well to other benchmarks. The networks CNN_MSRC, CNN_ICVL, CNN_NYU, and CNN_BigHand are trained on the training set of MSRC, ICVL, NYU, and *BigHand2.2M*, respectively.

train \ test	ICVL	NYU	MSRC	*BigHand2.2M*
ICVL	**12.3**	35.1	65.8	46.3
NYU	20.1	**21.4**	64.1	49.6
MSRC	25.3	30.8	**21.3**	49.7
BigHand2.2M	**14.9**	**20.6**	**43.7**	**17.1**

Table 3: Cross-benchmark comparison. Mean errors of CNNs trained on ICVL, NYU, MSRC and *BigHand2.2M* when cross-tested. The model trained on *BigHand2.2M* performs well on ICVL and NYU, less so on the synthetic MSRC data. Training on ICVL, NYU, or MSRC does not generalize well to other datasets.

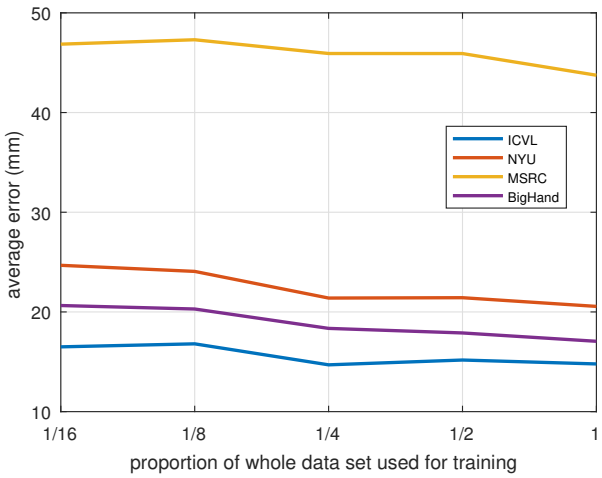

Figure 8: Data size effect on cross benchmark evaluation. When the CNN model is trained on $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and all of the benchmark data, the test results on ICVL, NYU, MSRC, and *BigHand2.2M* keep improving.

Black and a 32-core Intel processor. The model is trained using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\alpha = 0.0003$. We stop the training process when the cost of the validation set reaches the minimum, where each training epoch takes approximately 40 minutes. When training the CNN model on *BigHand2.2M*, ICVL, NYU and MSRC, we keep the CNN structure and β_1, β_2, α of Adam unchanged.

All frames of 10 subjects are uniformly split into a training set and a validation set with a 9-to-1 ratio, which is similar to ICVL, NYU, and HandNet [34]. In addition to the 10 subjects, a challenging test sequence of 37K frames of a new subject is recorded and automatically annotated, as shown "new person" in Figure 6. For a quantitative comparison, we measure the ratio of joints within a certain error bound ϵ [38, 29, 24].

5.1. Cross-benchmark performance

Cross-benchmark evaluation is a challenging and less-studied problem in many fields, like face recognition [16] and hand pose estimation [27]. Due to the small number of training datasets, existing hand pose estimation systems perform poorly when tested on unseen hand poses. As pointed out in [27], in existing datasets, "test poses remarkably resemble the training poses", and they proposed

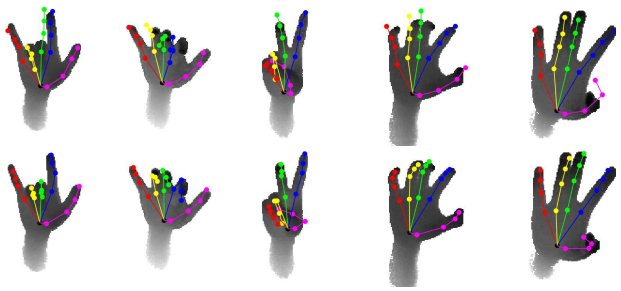

Figure 9: Generalization of the CNN trained on *BigHand2.2M*. The CNN generalizes to the ICVL dataset with a lower error than the original annotated ground truth. (top) ICVL ground truth annotations, (bottom) our estimation results.

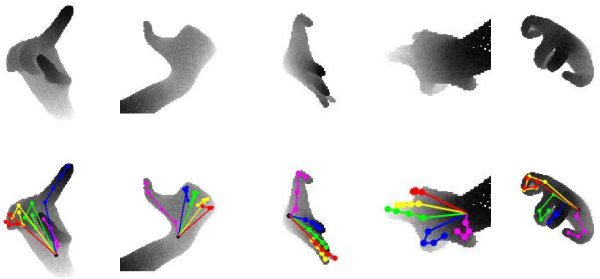

Figure 10: MSRC benchmark examples. Synthetic data lacks real hand shape and sensor noise, and tends to have kinematically implausible hand poses. The top row shows some depth images, the bottom row shows the corresponding ground truth annotation.

"a simple nearest-neighbor base line that outperforms most existing systems".

Table 3 and Figure 7 show the estimation errors of the CNNs trained on ICVL, NYU, MSRC and *BigHand2.2M* when cross-tested. The performance when the CNN is trained on the*BigHand2.2M* training set is still high when evaluated on other datasets. On real test datasets (ICVL and NYU), it achieves comparable or even better performance than models trained on the corresponding training set. This confirms that with high annotation accuracy and with sufficient variation in shape, articulation and viewpoint parameters, a CNN trained on a large-scale dataset is able to generalize to new hand shapes and viewpoints, while the nearest neighbor method showed poor cross-testing performance [27].

The MSRC dataset is a synthetic dataset with accurate annotations and evenly distributed viewpoints. When training the CNN on MSRC and testing on all real testing sets, the performance is worse than the CNN trained on NYU, and significantly worse than when trained on *BigHand2.2M*. Performance is to that of a CNN trained on ICVL which is only one-sixth in size compared to the MSRC training set. On the other hand, the model trained on *BigHand2.2M* shows consistently high performance across all real datasets, but poorer performance on the MSRC test set due to the differences between real and synthetic data. Figure 10 shows examples from the MSRC dataset. Synthetically generated images tend to produce kinematically implausible hand poses, which are difficult to assume without applying external force. There are also differences in hand shape, *e.g.* the thumb appears large compared to the rest of the hand.

Increasing the amount of training data improves the performance on cross benchmark evaluation, see Figure 8. In this experiment, we uniformly subsample fractions of $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and 1 from the training and validation data, respectively. When we train CNNs with the increasing portions of *BigHand2.2M* and test them on ICVL, NYU, MSRC, and *BigHand2.2M*'s test sequences, the performance is fairly improved. These observations support that the larger amount of training data enables CNNs to better generalize to new unseen data. Also, note our dataset is dense such that the random small fractions of the training data still delivers the good accuracies.

5.2. State-of-the-art comparison

In this section we compare our CNN model trained on *BigHand2.2M* with 8 state-of-the-art methods including HSO [29], Sun *et al.* [26], Latent Regression Forest (LRF) [28], Keskin *et al.* [5], Melax *et al.* [9], DeepPrior [13], FeedLoop [14], and Hier [38].

When the CNN model trained on *BigHand2.2M* is used for testing on NYU, it outperforms two recent methods, DeepPrior [13] and FeedLoop [14], and achieves comparable accuracy with Hier [38], even though the model has never seen any data from NYU benchmark, demonstrated in the right of Figure 7. Since the annotation scheme of NYU is different from ours, we choose a common (still deviated to a certain degree) subset of 11 joint locations for this comparison. We expect better results for consistent annotation schemes.

The ICVL test error curve of the CNN model trained on *BigHand2.2M* is shown in Figure 7(left). We choose the maximum allowed error [29] metric. Although it does not appear as good as that trained on ICVL itself, HSO and Sun et al., it outperforms the other methods. Note that the mean estimation error for our CNN model is already as low as 14mm, which means that a small annotation discrepancy between training and test data will have a large influence on the result. As noted in [12], the annotation of ICVL is not as accurate as that of NYU. Many frames of our estimation results look plausible, but result in larger estimation errors because of inaccurate annotations, see Figure 9 for qualitative comparisons. Another reason is that the hand measurement scheme is different from ours. In our dataset, each subject's hand shape is determined by manually measuring

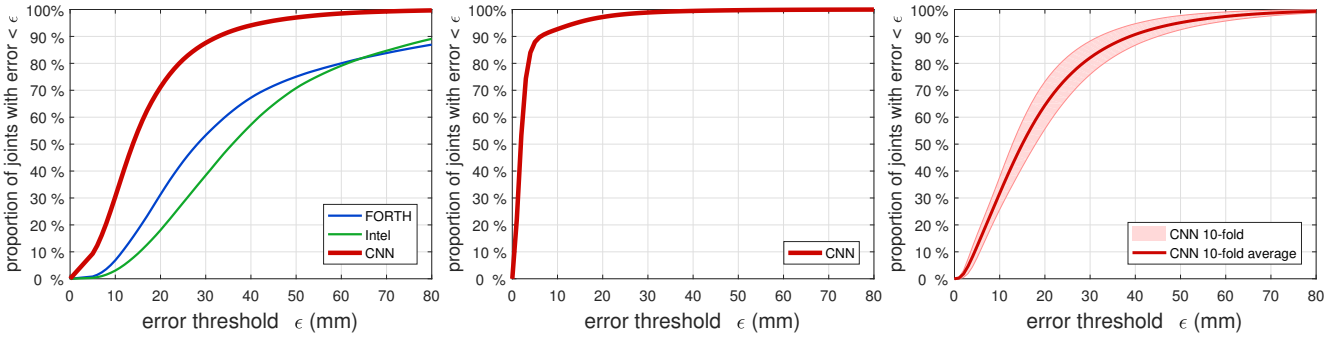

Figure 11: Hand pose estimation performance. (left) baseline performances on the new subject's 37K frames of hand images. The *Holi* CNN significantly outperforms the tracking-based methods FORTH [15] and Intel [3]. (middle) the CNN trained on 90% of the *BigHand2.2M* data achieves high accuracy on the remaining 10% validation images. (right) 10-fold cross-validation result when using the CNN for egocentric hand pose estimation. We achieved a similar-level accuracy to that of third-view hand pose estimation.

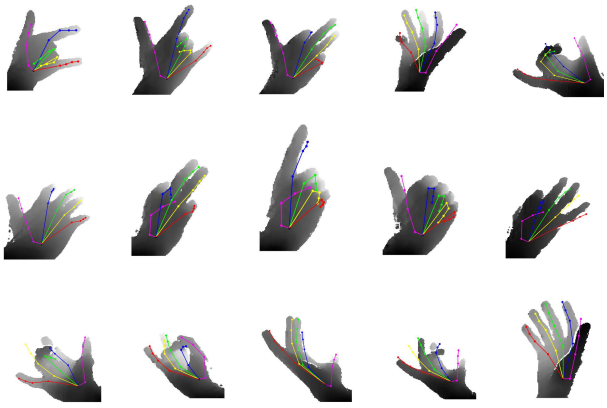

Figure 12: Qualitative results on the egocentric-view dataset. A CNN trained on *BigHand2.2M* achieves state-of-the-art performance in the egocentric-view pose estimation task.

joint distances. In ICVL, the same synthetic model is used for all subjects and the MCP joints tend to slide towards the fingers rather than remaining on the physical joints.

5.3. Baselines on BigHand2.2M

Three baselines are evaluated on our 37K-frame testing sequence, the CNN trained on *BigHand2.2M*, the Particle Swarm Optimization method (FORTH) [15] and the method by Intel [3]. The latter two are generative tracking methods. The CNN model outperforms the two generative methods, see the left plot of Figure 11. As described in the above, we chose a size ratio between training and validation sets of 9:1. Figure 11(middle) shows the result on the validation set, where 90% of the joints can be estimated within a 5mm error bound.

5.4. Egocentric dataset

The lack of a large-scale annotated dataset has been a limiting factor for egocentric hand pose estimation. Existing egocentric benchmarks [20, 12] are small, see Table 2. Rogez *et al*. [20] provide 400 frames and Oberwerger *et al*. [12] provide 2,166 annotated frames. The *BigHand2.2M* egocentric subset contains 290K annotated frames of ten subjects (29K frames each). This dataset enabled us to train a CNN model resulting in performance competitive with that of third view hand pose estimation. In the experiment we train the CNN on nine subjects and test it on the remaining one. This process is done with 10-fold cross validation. We report mean and standard deviation of the ten folds, see Figure 11 (right). Figure 12 shows qualitative results.

6. Discussion and conclusion

Hand pose estimation has attracted a lot of attention and some high-quality systems have been demonstrated, but the development in datasets still lagged behind the algorithm advancement. To close this gap we captured a million-scale benchmark dataset of real hand depth images. For automatic annotation we proposed using a magnetic tracking system with six magnetic 6D sensors and inverse kinematics. To build a thorough yet concise benchmark, we systematically designed a hand movement protocol to capture the natural hand poses. The *BigHand2.2M* dataset includes approximately 290K frames captured from an egocentric view to facilitate the advancement in the area of egocentric hand pose estimation. Current state-of-the-art methods were evaluated using the new benchmark, and we demonstrated significant improvements in cross-benchmark evaluations. It is our aim that the dataset will help to further advance the research field, allowing the exploration of new approaches.

References

[1] C. Choi, A. Sinha, J. Hee Choi, S. Jang, and K. Ramani. A collaborative filtering approach to real-time hand pose estimation. In *ICCV, 2015*. 1

[2] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *CVPR, 2016*. 1

[3] Intel SR300. https://click.intel.com/intelrealsense-developer-kit-featuring-sr300.html. 2, 4, 8

[4] Y. Jang, S.-T. Noh, H. J. Chang, T.-K. Kim, and W. Woo. 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *VR, 2015*. 1

[5] 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*. 1, 7

[6] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *CVPR, 2015*. 1

[7] P. Li, H. Ling, X. Li, and C. Liao. 3d hand pose estimation using randomized decision forest with segmentation index points. In *ICCV, 2015*. 1

[8] H. Liang, J. Yuan, and D. Thalmann. Parsing the hand in depth images. *TMM, 2014*. 1

[9] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3D skeletal hand tracking. In *i3D, 2013*. 2, 7

[10] NDI trakSTAR. https://www.ascension-tech.com/products/trakstar-2-drivebay-2/. 3, 4

[11] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Hand segmentation with structured convolutional learning. In *ACCV, 2014*. 1

[12] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently creating 3D training data for fine hand pose estimation. In *CVPR, 2016*. 1, 2, 3, 4, 7, 8

[13] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. In *CVWW, 2015*. 1, 7

[14] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *ICCV, 2015*. 1, 7

[15] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. In *BMVC*, 2011. 1, 8

[16] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC, 2015*. 6

[17] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixe, M. Mueller, H.-P. Seidel, and B. Rosenhahn. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *ICCV, 2011*. 3

[18] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *CVPR, 2014*. 1, 2

[19] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof. A framework for articulated hand pose estimation and evaluation. In *SCIA, 2015*. 3

[20] G. Rogez, J. S. Supancic, and D. Ramanan. First-person pose recognition using egocentric workspaces. In *CVPR, 2015*. 2, 3, 4, 8

[21] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from RGB-D images. In *ICCV, 2015*. 3

[22] S. Schaffelhofer and H. Scherberger. A new method of accurate hand- and arm-tracking for small primates. *Journal of Neural Engineering*, 9(2), 2012. 4

[23] ShapeHand. 2009. http://www.shapehand.com/shapehand.html. 3

[24] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible real-time hand tracking. In *CHI, 2015*. 1, 2, 3, 5, 6

[25] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV, 2013*. 1, 2

[26] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *CVPR, 2015*. 1, 2, 7

[27] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: methods, data, and challenges. *ICCV, 2015*. 1, 6, 7

[28] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3D articulated hand posture. In *CVPR*, 2014. 1, 2, 5, 7

[29] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *ICCV, 2015*. 6, 7

[30] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013. 1

[31] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. In *TOG, 2014*. 1, 2, 5

[32] T. von Marcard, G. Pons-Moll, and B. Rosenhahn. Human pose estimation from video and imus. *TPAMI, 2016*. 3

[33] C. Wan, A. Yao, and L. Van Gool. Hand pose estimation from local surface normals. In *ECCV, 2016*. 1

[34] A. Wetzler, R. Slossberg, and R. Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *BMVC, 2015*. 1, 2, 3, 4, 6

[35] Y. Wu, J. Y. Lin, and T. S. Huang. Capturing natural hand articulation. In *ICCV, 2001*. 4

[36] C. Xu, N. Ashwin, X. Zhang, and L. Cheng. Estimate hand poses efficiently from single depth images. *IJCV*, 2015. 1, 2, 3

[37] C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *ICCV*, 2013. 3

[38] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. In *ECCV, 2016*. 1, 5, 6, 7

[39] Y. Zheng, S. Sugimoto, and M. Okutomi. ASPnP: An accurate and scalable solution to the perspective-n-point problem. *IEICE TIS, 2013*. 4

[40] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. In *IJCAI, 2016*. 1