2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Precise Hand Segmentation from a Single Depth Image

Minglei Li[1,2,*], Lei sun[2] and Qiang Huo[2]
[1]University of Science and Technology of China, Hefei, China
[2]Microsoft Research, Beijing, China
Email: {v-mingll, lsun, qianghuo}@microsoft.com

***Abstract*—We propose a new approach to segmenting a hand accurately from a single depth image. Given a depth image, we extract first a rough hand region of interest (RoI) including a hand and a part of an arm. Then, the RoI is partitioned into triangles by using a constrained Delaunay triangulation (CDT) approach from which hand segmentation proposals are generated. Each segmentation proposal is evaluated by a shallow convolutional neural network (CNN) which is trained as a regression function to predict a confidence score for each proposal. Finally, the segmentation proposal with the highest confidence score is selected as our hand segmentation result. To evaluate the effectiveness of our approach, we use a set of real data containing more than 370,000 frames of hand depth images collected from 40 subjects with large variations in pose, orientation and sensing distance. Compared with segmentation results achieved by a random decision forest (RDF) based approach, our approach achieves much higher accuracy.**

I. INTRODUCTION

Hand segmentation is an important pre-processing step in applications such as hand pose estimation [1, 2] and gesture recognition [3]. Research on hand segmentation has a long history. In early days, color images were taken as input and skin color models [4–6] were used to locate hands. Although these skin color based methods could get good results in certain environments, they usually suffered from variabilities in skin color, illumination and noisy backgrounds. Recently, given the popularity of consumer depth sensors, a rough hand region of interest (RoI) could be easily extracted from noisy backgrounds by simply setting thresholds [7–11]. However, it is difficult to precisely and consistently separate a hand from the rough RoI for the following reasons: 1) There is no clear boundary between a hand and a forearm on a depth map; 2) Large variations in hand postures and orientations make it hard to separate hand consistently; 3) The low resolution of a depth image captured by a consumer depth sensor imposes additional challenges.

To address the above problem, many methods have been proposed in the literature. Bergh *et al.* [12] and Oikonomidis *et al.* [13] tried to leverage skin color to refine the hand RoI extracted from a depth map. Combining skin color and depth information could narrow the search area for hand and decrease largely the effect of noisy backgrounds. However, they will not be effective for arms without sleeve. Therefore, Ren *et el.* [3] and Qian *et al.* [14] required users to wear black bands on their wrists. Some researchers tried to explore geometric characteristics around the wrist and designed rules to get finer hand segmentation on depth maps [15–18]. Kurakin *et al.* [15] and Doliotis *et al.* [16] desired to cut off an arm at wrist point and tried to identify the wrist area by finding the thinnest part of the arm. Liang *et al.* [17] detected a palm by finding the largest inscribed circle. Qin *et al.* [18] first separated a hand RoI into two parts using thresholds and detected their center points through distance transform, then set the mid-perpendicular line of the segment connecting the two center points as the cut line. These rules set constraints on rotation angles of a hand because their effectiveness was affected greatly by viewpoint variation. Furthermore, clustering methods were also tried. Malassiotis *et al.* [19] used a hierarchical clustering procedure to get a rough hand RoI, which is then modeled by a mixture of two Gaussians. Feng *et al.* [20] used a Gaussian mixture model (GMM) to separate a rough hand RoI from human body, which is then segmented into two classes by K-means clustering method. These methods are not robust because they are sensitive to model initialization. Given the success of random decision forest (RDF) based approach for body parts segmentation [21] and hand pose estimation [22], Tompson *et al.* [1] used RDF to train a binary classifier for hand segmentation. However, according to [23] where a similar RDF binary classifier is also used for hand segmentation, the inferred boundary between hand and arm is not accurate, therefore an error of several centimeters between neighboring frames of depth images in a video is observed.

In this paper, we propose a new approach to segmenting a hand accurately from a single depth image. Fig. 1 shows the overview of our approach. Given an input depth image and a hand joint location, a hand RoI is extracted first by using flood fill method. Then, a constrained Delaunay triangulation (CDT) approach is used to partition the hand RoI into triangles, from which high quality hand segmentation proposals are constructed. Each segmentation proposal is evaluated by a shallow CNN which is trained as a regression function to predict a confidence score for each proposal. Finally, the segmentation proposal with the highest confidence score is selected as our hand segmentation result. To the best of our knowledge, this is the first work to leverage CDT partitioned triangles to construct hand segmentation proposals in a depth

---

[*] Minglei Li contributed to this work when he worked as an intern with the Speech Group, Microsoft Research Asia.

978-1-5090-4846-5/16/$31.00

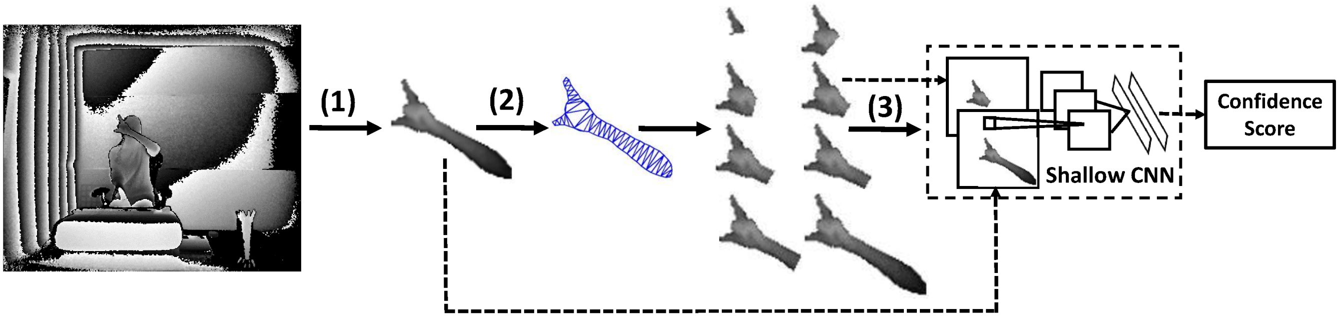

Fig. 1. Overview of our approach: (1) Hand RoI extraction: a hand RoI is extracted from a depth image; (2) Proposal Generation: CDT is used to partition the hand RoI into small triangles from which segmentation proposals are generated; (3) Best proposal selection: Each segmentation proposal is evaluated by a scoring CNN with both the proposal and hand RoI as input, and the segmentation proposal with the highest confidence score is selected as our hand segmentation result.

image. Using both segmentation proposal and hand RoI as the input of the scoring CNN is more effective than using the segmentation proposal only as input. To evaluate the effectiveness of our approach, we use a set of real data containing more than 370,000 frames of hand depth images collected from 40 subjects with large variations in pose, orientation and sensing distance. Compared with segmentation results achieved by an RDF based approach, our approach achieves much higher accuracy.

The rest of this paper is organized as follows. Section II presents the details of our approach. Section III reports experimental results and findings. Finally, the paper is concluded in Section IV.

II. OUR APPROACH

A. Hand RoI Extraction

Given a depth image, there are various methods to get a hand RoI [7–11, 23]. Here, we take a hand joint estimated from Kinect SDK as a seed point and use flood fill method to grow a hand RoI from a square range around the hand joint. In this paper, we focus our work on separating a hand from a rough hand RoI.

B. CDT Based Proposal Generation

Zou *et al.* [24] showed that CDT worked well for estimating skeletons of ribbon-like shapes. Triangles generated by CDT on ribbon-like shapes will be distributed densely and symmetrically alongside their skeletons. Because a human arm has a ribbon-like shape, we use CDT to partition a hand RoI and generate segmentation proposals based on these triangles.

1) Constrained Delaunay Triangulation: Given a rough hand RoI, its external contour is extracted and uniformly sampled, as shown in Fig. 2(b). The sampled contour can be described by a planar straight-line graph (PSLG), where the vertices and edges are the contour pixels and contour segments between adjacent contour pixels, respectively. Given the PSLG, CDT can be constructed efficiently by using the classic divide-and-conquer algorithm [25]. Since partial triangles will be merged into polygons in the following steps, we use a general term *polygon* to denote triangles and polygons.

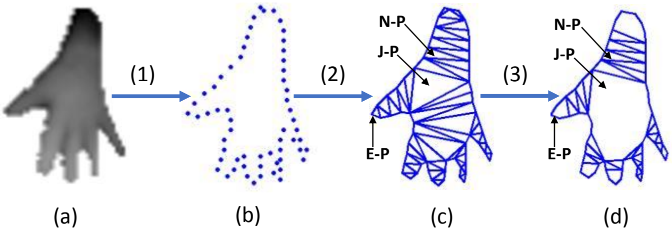

Fig. 2. Merging triangles: (1) The external contour of a hand RoI is extracted and uniformly sampled; (2) CDT of the hand RoI; (3) Partial triangles are merged.

Similar to [24], an edge of a *polygon* is an *external edge* if it is a contour segment; otherwise, it is an *internal edge*. As shown in Fig. 2(c), a *polygon* with one, two, and more than two internal edges is called an *end-polygon (E-P)*, a *normal-polygon (N-P)*, and a *junction-polygon (J-P)*, respectively. Following observations can be made:

- E-P usually appears in the extreme parts of a shape, e.g., fingertips, ends of the arm or small protrusions on the hand RoI;
- J-P usually presents at the center of parts with arched contours, e.g., the palm of a hand;
- N-P exists almost everywhere and mostly shows at ribbon-like/tube-shaped parts of a shape, e.g., an arm or fingers.

Since N-Ps are densely and symmetrically distributed along an arm, we are inspired to use each N-P to bisect the hand RoI and get two segmentation proposals each time, as shown in Fig. 3. Bisections using all the N-Ps (BNP) on a hand RoI would generate dense segmentation proposals, among which there are some high quality proposals that share high overlap with the groundtruth segmentation, as shown in the first three columns of Fig. 3. However, such complete bisections also generate many undesirable proposals, e.g., bisection using N-Ps on a finger, as shown in the last four columns of Fig. 3. Because only bisections around the real cut-line and wrist matter, bisections based on N-Ps in the following three parts, namely extreme parts mentioned above, center of a palm, and

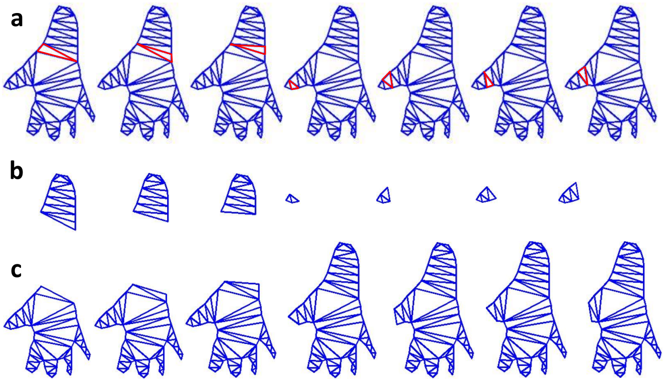

Fig. 3. Bisection using an N-P: Row (a) shows CDTs of hand RoIs with target N-Ps highlighted in red; Row (b) and (c) show the bisection results.

fingers, will be filtered out as described as follows.

2) Merging Triangles: For N-Ps in extreme parts, some of them could be filtered out by adopting the methods of merging end regions in [24]. Besides, by using a simple rule that an N-P shall be merged if its neighbors are both J-P, some of the N-Ps in the palm could be filtered out. After merging step, bisection using all the remaining N-Ps (BNPM) reduces many undesirable proposals, yet high quality proposals are almost unchanged.

3) Bisection Along Mid-line: Although many N-Ps in fingers are still kept after above merging step, most of them can be ignored during proposal generation as follows. As hand RoIs are ribbon-like shapes, a mid-line of a hand RoI will go through its arm and at most one finger. Based on this notion, we firstly find two extreme E-Ps which have maximum or minimum projection on the hand RoI's mid-line, estimated by PCA. Then we traverse from one E-P to the other, as shown in the mid-image of Fig. 4. N-Ps on the traversed path are reserved for further bisection to generate hand proposals. We denote bisections using N-Ps along a mid-line after merging step as BNPMM. By using this step, the number of proposals could be largely reduced for cases that have multiple fingers.

C. Best Proposal Selection

After the above steps, tens of hand segmentation proposals are generated. The segmentation problem becomes to select a proposal that best matches the groundtruth segmentation. By assigning a confidence score to quantify the matching degree, we formulate the segmentation problem as a confidence regression problem. A regression model needs to be trained to predict a confidence score for each proposal. To reduce computations, we use a shallow CNN as the regression model.

1) Regression Model Architecture: Fig. 5 illustrates the network architecture. It takes a proposal and its corresponding hand RoI as inputs. Then, one convolutional layer and a max pooling layer are used to produce a feature map. The feature map is fed into a 3-layer fully connected network to predict a confidence score. The following smooth L1 loss function is used for training CNN:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if \ \ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (1)$$

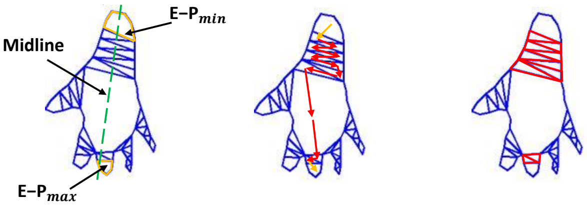

Fig. 4. Get N-Ps along mid-line.

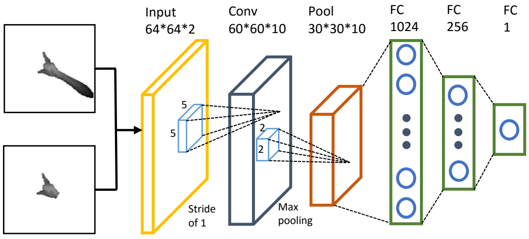

Fig. 5. Regression model for scoring proposals.

2) Training Parameter Setting: The network is trained in a standard way using stochastic gradient descent (SGD). Weights of all layers are initialized from zero-mean Gaussian distribution with standard deviation 0.01. Biases are initialized to 0. All layers use a per-layer learning rate of 1 for weights and 2 for biases and a global learning rate of 0.01. During training, we use inverse decay learning rate policy to gradually lower the learning rate with increasing mini-batch iterations. The inverse decay curve uses gamma of 0.001, power of 0.75. The network's momentum and parameter decay are set 0.9 and 0.0005 (on weights and biases), respectively.

3) Confidence Score Labeling: The groudtruth confidence score for a proposal is set according to its similarity with the groundtruth segmentation. The similarity between a proposal and its corresponding groundtruth can be quantified by pixel intersection over union (IU). Since our final goal is to find a proposal that best matches the groundtruth segmentation, we force the regression model to focus on learning scoring strategies for proposals that have little difference with the groundtruth. Similar to [26], proposals that have IUs lower than 0.6 are treated as negative candidates. These proposals' confidence scores are labeled as 0. The remaining proposals share a large overlap with their corresponding groundtruth and their confidence scores are labeled with a positive value. Denote a proposal image as I_p, its corresponding groundtruth segmentation as I_{gt}, we design a piecewise linear function to label groundtruth confidence score, S_p, for each proposal as follows:

$$S_p = \begin{cases} IU(I_p, I_{gt}) & if \ \ IU(I_p, I_{gt}) > 0.6 \\ 0 & otherwise \end{cases} \quad (2)$$

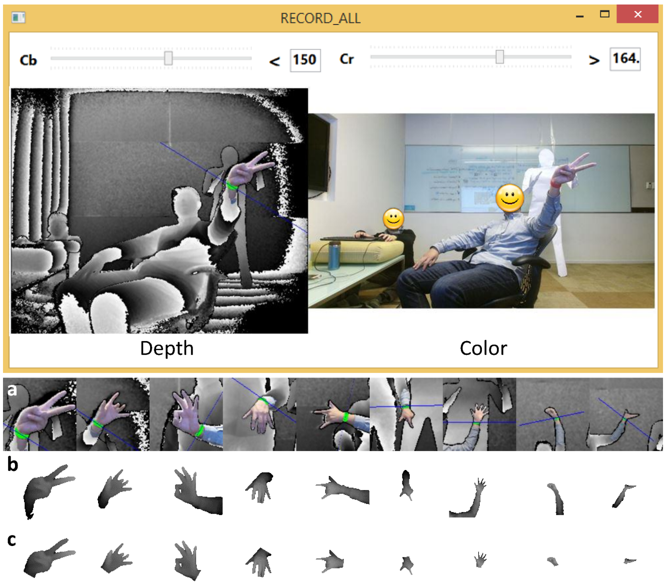

Fig. 6. Data recording and automatic groundtruth segmentation: (1) The upper panel shows the GUI recording tool. The detected red band points are highlighted with green color and a blue cut-line is estimated via PCA. To adapt to various lighting conditions, threshold parameters can be modified online; (2) The bottom panel shows some examples. Row (a) shows the mapping from color to depth. Row (b) shows the extracted hand RoIs in depth images. Row (c) shows the automatic segmentation groundtruth.

III. EXPERIMENT

A. Experimental Setup

1) Dataset: Tompson *et al.* [1] have released a subset of data for hand segmentation in NYU Hand Pose Dataset. It contains 6,735 depth frames with per-pixel labeled groundtruth. As it is collected from only one subject with limited number of frames, it can hardly cover the grand variation of hand configurations. Thus, we decide to collect a larger data set that covers variations in pose, orientation and sensing distance from different subjects.

a) Data Collection: The data set is collected from 40 subjects with 20 males and 20 females using Kinect V2 sensor. To enrich variations in pose and orientation, subjects are encouraged to change their postures and arm orientations as much as they can in both seated and standing scenarios. Besides, they can move freely at a distance from 1 to 3 meters to the Kinect V2 camera. In total, 371,564 depth images are collected. To estimate groundtruth segmentation, a subject needs to wear a red band on his/her wrist during data collection.

b) Groundtruth Labelling: The red band is robustly detected by setting adaptive thresholds in YC_bC_r color space during data recording. Then, a cut-line is modeled as a straight line passing through the center of the detected red band points with its direction estimated by the first component of PCA of the red band. By mapping the detected red points from color image to the corresponding depth image, a cut line on the depth map can be calculated in the same way.

According to joint locations from Kinect SDK, a hand RoI is generated using flood fill method and a rough arm orientation can be estimated. Then, RoI is bisected using the above cut-line into two parts. Finally, a groundtruth hand segmentation can be chosen from the bisected two parts by using the rough arm orientation. The dataset is built by the rough hand RoIs and their corresponding groundtruth hand segmentations. Fig. 6 illustrates how data recording and automatic groundtruth segmentation work.

2) Data Normalization: Translation and scale normalization of hand RoIs in depth maps benefit the segmentation performance and can be easily implemented. By setting the gravity of the hand RoI or the hand joint location as the center point of a captured hand RoI image, translation invariance is achieved. As a camera can be modeled by a usual pinhole, an object's size s in the image is inversely proportional to its sensing distance d to the camera. Thus, the scales of all hand RoIs can be mapped to the same scale level by multiplying a scale factor $\frac{d}{d_{ref}}$, where d is the above center point's depth and d_{ref} is a predefined reference distance. Therefore, hand RoIs' scales are also depth invariant. In this paper, d_{ref} is set as $1500mm$. After translation and scale normalization, a hand RoI image is cropped to a square size of 120 × 120 while keeping the above center point as the center of the square. Besides, gray-scales of hand RoI pixels are linearly mapped to a range from 0 to 150 according to depth.

3) Comparison Method: As RDF is a popular algorithm that has achieved promising segmentation results in various scenarios [1, 21, 22, 27], we compare our approach with an RDF-based hand segmentation approach [1]. As described in [1], the RDF consists of 4 trees with the maximum height of 25. At each node in a tree, 10,000 weak learners is sampled. Since hand RoIs' scales are normalized and being depth invariant. At a given pixel (u, v) on normalized hand RoI image I, each node in the decision tree evaluates

$$I\left(u+\delta u, v+\delta v\right) - I\left(u, v\right) \geqslant d_t \ . \tag{3}$$

To train offset parameters $(\delta u, \delta v)$ and threshold d_t, we generate 100 vectors for offset candidates, and 100 scale values for threshold candidates on each node. Through comparison experiments of using different values for offset range from a discrete set, this classifier performs best with offset range of 40 pixels. Threshold range is set from 0 to 150.

To refine segmentation results from RDF, a post-processing step including median filtering and largest blob detection is implemented in this study.

B. Experimental Results

We adopt $F_{\beta=0.3}$ [28], IU [29] and F_1 to evaluate the performance of hand segmentation at pixel level.

1) Quality of Proposals: To evaluate the quality of generated proposals, two aspects are checked: 1) how similar a hand RoI's proposals are with its groundtruth; 2) how many proposals are generated for a depth image. For similarity, we could use the upper bound of IUs between a hand RoI's proposals and its groundtruth. Let IU_{up} represent IU upper

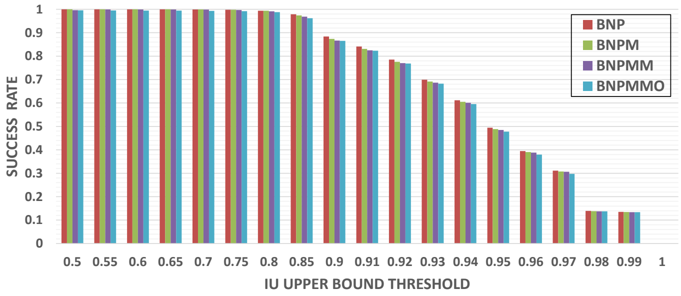

Fig. 7. Histograms of proposal generation success rate.

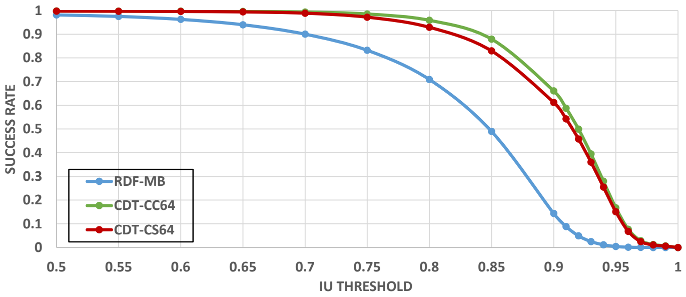

Fig. 8. Success rate curves of different hand segmentation methods.

TABLE I
COMPARISON OF MEAN IU UPPER BOUND AND NUMBER OF PROPOSALS OF PROPOSAL GENERATION METHODS.

	BNP	BNPM	BNPMM	BNPMMO
mean n_p	93.4	71.9	60.9	30.5
mean IU_{up}	0.9459	0.9448	0.9437	0.9423

TABLE II
COMPARISON OF IU AND F MEASURE OF SEGMENTATION METHODS

	Ours(CDT-CC64)	Ours(CDT-CS64)	RDF-MB
IU	**0.9085**	0.8998	0.8175
$Precision$	**0.9628**	0.9588	0.9584
$Recall$	**0.9416**	0.9360	0.8476
$F_{\beta=0.3}$	**0.961**	0.9569	0.9482
F_1	**0.952**	0.9473	0.8996

bound and n_p represent the number of generated proposals of a hand RoI, we have

$$IU_{up} = \operatorname{argmax}\{IU_i\}, i = 0, 1, 2, \cdots, n_p \,. \qquad (4)$$

If we know the arm orientation of a hand RoI, we can further filter out half proposals by reserving one proposal that nears the hand part at each bisection. This strategy is referred to as BNPMMO. Here, we use a vector from the wrist joint to hand joint to represent the arm orientation. Results are shown in Fig. 7 and Table I.

Fig. 7 shows that our proposal generation methods can generate high quality proposals that has high-overlap with groundtruth segmentations. Success rate for IU upper bound threshold at 0.85, 0.9, and 0.95 are around 96%, 86% and 47%, respectively. On average, IU upper bounds of our proposal generation method are around 0.94.

Table I verifies the effectiveness of our proposed methods in filtering out redundant proposals. For our dataset, adding a merging step (BNPM) can reduce 20 redundant proposals on average with a slight decrease of IU upper bound. By searching N-Ps on a hand RoI's mid-line and only using these N-Ps for bisection (BNPMM), 10 more redundant proposals can be reduced on average. If we can know the arm orientation of a hand RoI, we can filter out half of the proposals (BNPMMO). Here, by roughly estimating the arm orientation using wrist and hand joint locations given by Kinect SDK, we lower the number of proposals to 30 on average. However, if the arm orientation is wrongly estimated, using BNPMMO will lose all high quality proposals. The other proposal generation methods, namely BNT, BNPM, BNPMM are not affected by this issue.

2) Segmentation Performance: Recorded data of 40 subjects are randomly partitioned into 30, 2 and 8 for training, validation and testing, respectively. Before training our regression model, inputs need to be resized to 64 × 64. To verify the effectiveness of adding a hand RoI as input (referred to as CDT-CC64 hereinafter), we train a comparison regression model with the same architecture except that only a proposal is used as the input (referred to as CDT-CS64 hereinafter). For RDF training, we randomly sample 10% pixels of each hand RoI. Let RDF-MB denote an RDF based segmentation approach with a post processing consisting of median filtering and blob detection. By calculating IU on each hand RoI, the success rate curve is drawn in Fig. 8 according to different IU thresholds. Table II shows IU and F score measure on whole pixels of test dataset. Furthermore, example results of CDT-CC64 and RDF-MB are shown in Fig. 9. It is observed that our approach achieves better performance in both F score measure and IU evaluation metric. One weakness of the RDF-based method is that its classification results for pixels near the classification boundary are not reliable, which has also been confirmed in [23]. Since our proposals are generated through bisections of some thin polygons, pixels in these proposals are connected and compact, therefore there is no "boundary pixel" issue suffered by RDF.

Adding a hand RoI as input helps. Compared with taking a single proposal as input, regression model gives better segmentation prediction by adding its hand RoI as input. We think this empowers the regression model to see the difference between a proposal and its hand RoI, which is meaningful auxiliary information.

IV. CONCLUSION

In this paper, we have proposed a novel approach for hand segmentation on depth images. Experimental results demonstrate that our approach could get quite accurate hand segmentation in various poses, orientations and sensing distances. Yet, our approach has a limitation in efficiency as it needs to predict a confidence score for each proposal independently. Since proposals have overlaps, we plan to improve our approach's efficiency by sharing computations in overlaps in the future.

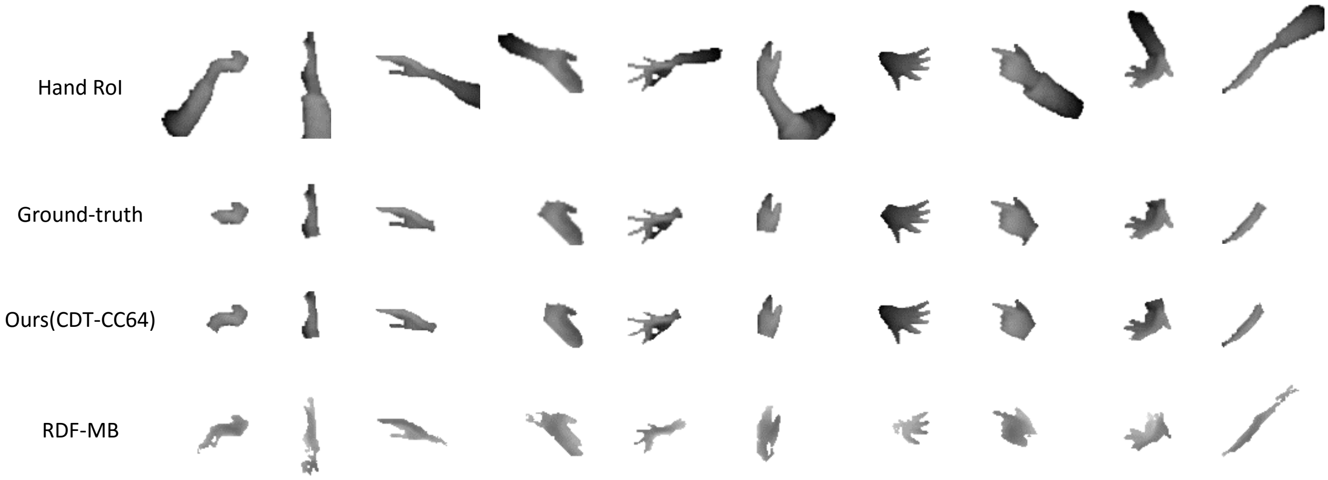

Fig. 9. Example segmentation results

REFERENCES

[1] J. Tompson, M. Stein, Y. LeCun, and K. Perlin, “Real-time continuous pose recovery of human hands using convolutional networks,” *ACM Trans. Graph.*, vol. 33, no. 5, pp. 169:1–169:10, 2014.

[2] A. Sinha, C. Choi, and K. Ramani, “Deephand: Robust hand pose estimation by completing a matrix imputed with deep features,” in *CVPR*, 2016.

[3] Z. Ren, J. Yuan, and Z. Zhang, “Robust hand gesture recognition based on finger-earth mover’s distance with a commodity depth camera,” in *MM*, 2011, pp. 1093–1096.

[4] Y. Wu and T. S. Huang, “View-independent recognition of hand postures,” in *CVPR*, 2000, pp. 2088–2094.

[5] X. Zhu, J. Yang, and A. Waibel, “Segmenting hands of arbitrary color,” in *FG*, 2000, pp. 446–455.

[6] E. Stergiopoulou and N. Papamarkos, “A new technique for hand gesture recognition,” in *ICIP*, 2006, pp. 2657–2660.

[7] P. Breuer, C. Eckes, and S. Müller, “Hand gesture recognition with a novel IR time-of-flight range camera-a pilot study,” in *MIRAGE*, 2007, pp. 247–260.

[8] Z. Mo and U. Neumann, “Real-time hand pose recognition using low-resolution depth images,” in *CVPR*, 2006, pp. 1499–1505.

[9] X. Liu and K. Fujimura, “Hand gesture recognition using depth data,” in *FG*, 2004, pp. 529–534.

[10] T. J. Cerlinca and S. G. Pentiuc, “Robust 3d hand detection for gestures recognition,” in *IDC*, 2011, pp. 259–264.

[11] D. Droeschel, J. Stückler, and S. Behnke, “Learning to interpret pointing gestures with a time-of-flight camera,” in *HRI*, 2011, pp. 481–488.

[12] M. V. den Bergh and L. J. V. Gool, “Combining RGB and tof cameras for real-time 3d hand gesture interaction,” in *WACV*, 2011, pp. 66–72.

[13] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, “Efficient model-based 3d tracking of hand articulations using kinect,” in *BMVC*, 2011, pp. 1–11.

[14] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, “Realtime and robust hand tracking from depth,” in *CVPR*, 2014, pp. 1106–1113.

[15] A. Kurakin, Z. Zhang, and Z. Liu, “A real time system for dynamic hand gesture recognition with a depth sensor,” in *EUSIPCO*, 2012, pp. 1975–1979.

[16] P. Doliotis, V. Athitsos, D. I. Kosmopoulos, and S. J. Perantonis, “Hand shape and 3d pose estimation using depth data from a single cluttered frame,” in *ISVC*, 2012, pp. 148–158.

[17] H. Liang, J. Yuan, and D. Thalmann, “3d fingertip and palm tracking in depth image sequences,” in *MM*, 2012, pp. 785–788.

[18] S. Qin, X. Zhu, Y. Yang, and Y. Jiang, “Real-time hand gesture recognition from depth images using convex shape decomposition method,” *IJSPS*, vol. 74, no. 1, pp. 47–58, 2014.

[19] S. Malassiotis and M. G. Strintzis, “Real-time hand posture recognition using range data,” *Image Vision Comput.*, vol. 26, no. 7, pp. 1027–1037, 2008.

[20] Z. Feng, S. Xu, X. Zhang, L. Jin, Z. Ye, and W. Yang, “Real-time fingertip tracking and detection using kinect depth sensor for a new writing-in-the air system,” in *ICIMCS*, 2012, pp. 70–74.

[21] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *CVPR*, 2011, pp. 1297–1304.

[22] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, “Hand pose estimation and hand shape classification using multi-layered randomized decision forests,” in *ECCV*, 2012, pp. 852–863.

[23] T. Sharp, C. Keskin, D. P. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. W. Fitzgibbon, and S. Izadi, “Accurate, robust, and flexible real-time hand tracking,” in *CHI*, 2015, pp. 3633–3642.

[24] J. J. Zou and H. Yan, “Skeletonization of ribbon-like shapes based on regularity and singularity analyses,” *IEEE Trans. Systems, Man, and Cybernetics*, vol. 31, no. 3, pp. 401–407, 2001.

[25] J. R. Shewchuk, “Triangle: Engineering a 2d quality mesh generator and delaunay triangulator,” in *WACG at FCRC*, 1996, pp. 203–222.

[26] R. B. Girshick, “Fast R-CNN,” in *ICCV*, 2015, pp. 1440–1448.

[27] H. Liang, J. Yuan, and D. Thalmann, “Egocentric hand pose estimation and distance recovery in a single RGB image,” in *ICME*, 2015, pp. 1–6.

[28] J. Dai and R. Chung, “Combining contrast saliency and region discontinuity for precise hand segmentation in projector-camera system,” in *ICPR*, 2012, pp. 2161–2164.

[29] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, “Lending A hand: Detecting hands and recognizing activities in complex egocentric interactions,” in *ICCV*, 2015, pp. 1949–1957.