Egocentric hand pose estimation and distance recovery in a single RGB image

Liang, Hui; Yuan, Junsong; Thalman, Daniel

2015

Liang, H., Yuan, J., & Thalman, D. (2015). Egocentric hand pose estimation and distance recovery in a single RGB image. 2015 IEEE International Conference on Multimedia and Expo (ICME), 1-6.

<https://hdl.handle.net/10356/80278>

<https://doi.org/10.1109/ICME.2015.7177448>

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/ICME.2015.7177448>].

Downloaded on 26 Apr 2023 19:57:28 SGT

EGOCENTRIC HAND POSE ESTIMATION AND DISTANCE RECOVERY IN A SINGLE RGB IMAGE

Hui Liang^{1,2}, Junsong Yuan¹, and Daniel Thalman²

¹School of Electrical and Electronics Engineering, ²Institute for Media Innovation,
Nanyang Technological University, 639798 Singapore.
shrekei@hotmail.com jsyuan@ntu.edu.sg danielthalmann@ntu.edu.sg

ABSTRACT

Articulated hand pose recovery in egocentric vision is useful for in-air interaction with the wearable devices, such as the Google glasses. Despite the progress obtained with the depth camera, this task is still challenging with ordinary RGB cameras. In this paper we demonstrate the possibility to recover both the articulated hand pose and its distance from the camera with a single RGB camera in egocentric view. We address this problem by modeling the distance as a hidden variable and use the Conditional Regression Forest to infer the pose and distance jointly. Especially, we find that the pose estimation accuracy can be further enhanced by incorporating the hand part semantics. The experimental results show that the proposed method achieves good performance on both a synthesized dataset and several real-world color image sequences that are captured in different environments. In addition, our system runs in real-time at more than 10fps.

Index Terms— egocentric vision, hand pose estimation, conditional regression forest

1. INTRODUCTION

Recently, the use of egocentric vision becomes a popular topic for human computer interaction (HCI) [1]. Compared to the third-person viewpoint, the egocentric vision is especially suitable for wearable devices, *e.g.* the Google glasses [2], where the traditional input tools like the mouse and keyboard are inconvenient to carry and use. By contrast, it is much more natural for the users to directly use their hands and fingers in the air for the various control and manipulative tasks on such devices, *e.g.* menu selection, web surfing and text input. To meet this need, we focus on the problem of hand pose recovery from RGB images in egocentric view.

The field of vision-based hand pose estimation has advanced a lot with the advent of the depth cameras, and there have been many practical solutions for hand pose recovery

in depth images [3, 4]. In [3], the authors propose to use the random forest to directly regress for the hand pose from depth images. With a pre-trained forest, each pixel casts its votes for the joint positions individually, and the votes from all the pixels are fused to obtain the final predictions. A similar regression forest base method is proposed in [4], with the new characteristic that transfer learning is utilized to handle the discrepancy between synthesized and real-world data.

Despite their good performance using the depth camera, hand pose estimation on RGB inputs is much more challenging. The RGB image is more sensitive to illumination variations and background clutter, and is less discriminative than the depth image, which introduces high ambiguity in inference. As a result, previous works in hand pose estimation with RGB inputs either require highly controlled environments or can only estimate the pose for low degree-of-freedom (DOF) hand motion. In [5] a variational formulation is proposed to estimate the full DOF hand pose, in which the illumination condition is well controlled so that the texture and shading on the hand can be also modeled to reduce the pose ambiguity. However, such method is difficult to use in real HCI scenarios. In [6] a set of feature points, *i.e.* the fingertips, palm center and wrist, are extracted from the edge map of the hand region to track the hand joints. As the heuristic rules used for feature extraction is quite ad hoc, *e.g.* the fingertip has a circular shape on the edge, and the wrist lies midway on the shortest segmentation line of the arm, this method cannot work well for the general hand postures, such as the bending or occluding fingers.

With the input of an ordinary RGB camera, we cannot expect the fine details of the hand to be captured, *e.g.* the nails, the texture and shading of the skin, due to the usually uncontrolled environment and limited image quality. Also, with the features that can be extracted relatively robustly, *e.g.* the hand edge or silhouette, the traditional methods only work for very restricted hand postures [7, 8]. In addition, to ensure fast recovery from tracking failure and rapid response in interaction, it would be more favorable to rely less on the temporal references that are heavily exploited in literature [9, 10].

To this end, we present a data-driven approach to obtain

This research, which is carried out at BeingThere Centre, is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Fig. 1. The pipeline of the proposed method. Left: RGB inputs. Middle: hand extraction. Right: CRF inference of both pose and distance.

accurate articulated hand poses using only the hand silhouette extracted from a single RGB image, which does not require highly controlled environments and can handle unconstrained hand motion more robustly. We assume that a red tape is stuck to the hand wrist, which is simple to prepare. Particularly, we show that the distance of the hand from the camera can also be recovered with reasonable accuracy. This can be quite useful in many HCI applications, *e.g.* to check whether the hand is in position for interaction. Technically, we utilize the Conditional Regression Forest (CRF) [11] to predict the hand pose and the hand distance jointly with a binary context descriptor, and the hand distance is modeled as a hidden variable for inference. Moreover, motivated by the previous work [12, 13] on two-stage pose inference with classified body parts, we propose to first extract the semantic hand parts from the hand silhouette with a Random Decision Forest (RDF) classifier [14], and use them as the features for pose estimation. This proves to improve the accuracy considerably.

2. THE PROPOSED APPROACH

We aim to recover both the articulated pose and the distance of the hand from single color images, and only require the user to wear a red tape at the wrist to assist hand segmentation. The distance d of the hand is defined as the distance between the wrist center to the image plane of the camera. The articulated hand pose Φ is the 2D positions of a set of hand joints. The processing pipeline is illustrated in Fig. 1. The binary hand silhouette is first extracted using skin color modeling and the color marker cue, and it is then used as the input to recover d and Φ . Note that the hand size and Φ change drastically with different d , we thus model d as a hidden variable and take advantage of the Conditional Regression Forest (CRF) [11] to infer both d and Φ jointly. The details are presented in the following sections.

2.1. Hand extraction

To recover the high DOF hand pose from the color inputs, the hand needs to be extracted from the background accurately, which is quite challenging. Due to the large shape variations of the hand, there is still no reliable shape-based detector to fulfill this task for unconstrained hand motion, and the exist-

ing ones only work well for a limited number of hand postures [15, 16]. Some other work relies on the body part context to improve the detection performance, *e.g.* face or limbs, [17], while such context information is not available in egocentric vision. Also, given that the skin-colored region is extracted robustly, it is still not easy to segment the hand from the arm.

Therefore, we adopt the skin-color modeling methods to generate hand region candidates [18], which works relatively well without body part contexts, and then refine them with a color marker on the wrist. Fig. 2 illustrates the pipeline. The YCbCr space is selected for skin detection as it proves more robust to illumination variation compared to RGB space. The Gaussian Mixture Model (GMM) is used to describe the color distributions for both the skin and non-skin regions:

$$P(v|s) = \sum_{i=1}^N \rho_{i,s} \mathcal{N}(\mu_{i,s}, C_{i,s}), \quad (1)$$

where v is a color value in YCbCr space, $s \in \{1, 0\}$ is the label of skin/non-skin regions, $\mathcal{N}(\mu_{i,s}, C_{i,s})$ is a single Gaussian component with mean $\mu_{i,s}$ and covariance $C_{i,s}$ and $\rho_{i,s}$ is the weight of each Gaussian component. The parameters in (1) are estimated with the Expectation-Maximization algorithm using annotated training data. By assuming equal priors for the skin/non-skin regions, a pixel is classified as skin if $P(s = 1|v) = P(v|s = 1) / \sum_{s \in \{0,1\}} P(v|s) > 0.5$.

As in Fig. 2, the red tape is stuck to the wrist to assist hand segmentation. With the red pixels extracted, we fit a 2D line ℓ_w to them using the RANSAC algorithm, and only retain the ones that fit ℓ_w as the wrist points U_w . Note that in egocentric vision, the position of the arm is mostly under the hand in the image, and thus the pixels below ℓ_w are taken as non-hand regions, *e.g.* the slashed part in the third of Fig. 2. Finally, the contours of the connected components in the refined hand mask are retrieved using the border following algorithm [19]. Let $\{B_c\}$ be the set of detected contours, each of which is a closed polygon. The one satisfying (2) is taken as the hand:

$$c^* = \arg \min_c \left\{ \min_{(p_1, p_2) \in B_c} \left[\sum_{p \in \{p_1, p_2\}} \min_{p_w \in U_w} \|p - p_w\|^2 \right] \right\}, \quad (2)$$

where $(p_1, p_2) \in B_c$ is the two end points of one line segment in the boundary of B_c . Formula (2) basically seeks for the contour whose boundary is closest to the wrists point set U_w , and this rule performs quite well in practice. The final refined hand region is shown in the fourth of Fig. 2.

2.2. CRF inference

With the extracted hand mask I_B , we utilize the CRF formulation in [11] to infer both the distance d and the hand joint positions Φ , which consists of the wrist center, the five fingertips, the inter-phalangeal (IP) and metacarpophalangeal (MCP) joints of the thumb, and the proximal inter-phalangeal (PIP) and MCP joints of other four fingers, *i.e.* $\Phi = \{\phi_k\}_{k=1}^K$,

Fig. 2. First: the original input. Second: the skin mask, the wrist points and the fitted line ℓ_w . Third: the skin parts under ℓ_w are removed. The contour closest to U_w is taken as the hand. Fourth: the final hand mask.

$K = 16$. The left of Fig. 3 illustrates an example of Φ . As the hand size and Φ vary largely for different d due to the projective transform of the camera, both the feature and 2D pose spaces are thus more complex compared to pose estimation with depth images [3, 4]. Therefore, the distance d is modeled as a hidden variable, so that a set of pose estimators can be trained separately conditioned on different d . During testing, the inference problem is formulated as:

$$\begin{aligned} \Phi^*, d^* &= \arg \max_{\Phi, d} P(\Phi, d | I_B) \\ &= \arg \max_{\Phi, d} P(\Phi | d, I_B) P(d | I_B), \end{aligned} \quad (3)$$

where $P(\Phi | d, I_B)$ is the pose distribution obtained by the pose estimator trained for distance d . $P(d | I_B)$ is the overall distance distribution which can be obtained with a separately trained distance regressor. Similar to [11], the regression forest is utilized to estimate the pose distribution $P(\Phi | d, I_B)$ by density estimation with the independent votes from a set of densely sampled pixels. These pixels first cast their votes for Φ independently and their votes are then fused to get more robust prediction. Specifically, with a set of voting pixels $\{p_i\}$, for each joint ϕ_k , the forest trained on distance d retrieves maximum J relative votes $\{\Delta_{ijk}, w_{ijk}\}_{j=1}^J$ for the pixel p_i . Δ_{ijk} is the relative vote for the joint position ϕ_k , *i.e.* the offset between the pixel and the joint ϕ_k , and w_{ijk} is the associated voting weight. By setting $v_{ijk} = \Delta_{ijk} + p_i$ the relative votes are converted to absolute votes, and $P(\Phi | d, I_B)$ can thus be obtained by fusing all the per-pixel votes:

$$P(\Phi | d, I_B) = \prod_k P(\phi_k | d, I_B), \quad (4)$$

$$\begin{aligned} P(\phi_k | d, I_B) &= \sum_i P(\phi_k | d, p_i) \\ &= \sum_{i,j} w_{ijk} \exp\left(-\frac{\|\phi_k - v_{ijk}\|^2}{\delta_\Phi^2}\right), \end{aligned} \quad (5)$$

where we assume a Gaussian kernel with bandwidth δ_Φ for density estimation.

Since I_B is a binary mask, we proposed a binary context descriptor $\mathcal{L}$ as the pixel feature for per-pixel regression, which is defined as the labels of a set of the neighboring context points of a pixel p . Each context point is represented as a relative pixel offset $u_b = [a_b, b_b]^T$. Given the pixel coordinate p of the current pixel, the positions of the context points can be determined by setting $p + u_b$. $\mathcal{L}$ is defined as:

$$\mathcal{L} = \{I_B(p + u_b) | b = 1, \dots, M\}, \quad (6)$$

Fig. 3. Left: the hand joints to be recovered (red circles). Right: the binary context descriptor. The labels of the context points are concatenated from top-left to bottom-right.

where M is the number of context points and $I_B(p)$ is the label of the pixel, which is 0 for the background and 1 for the hand, as shown at the right of Fig. 3. $\mathcal{L}$ is the concatenation of the labels of the context points, and $\mathcal{L}^b = I_B(p + u_b)$ is one dimension of $\mathcal{L}$. Assuming the forest contains T_r trees. During testing, an input pixel p will recursively branch down each tree and reach one of its leaf nodes based on the descriptors $\mathcal{L}$. The pixel reaches T_r leaf nodes in the regression forest and thus retrieves $J = T_r$ votes for each joint ϕ_k .

In addition, we propose to use the regression forest for distance estimation, *i.e.* to evaluate $P(d | I_B)$. To this end, a separate forest is trained with the binary context descriptor $\mathcal{L}$ to predict d . Similar to $P(\phi_k | d, I_B)$, this forest retrieves a set of distance votes $\{d_{ij}, w_{ij}\}_{j=1}^J$ for each voting pixel p_i , where d_{ij} is the vote for the hand distance and w_{ij} is the voting weight. $P(d | I_B)$ is obtained by:

$$P(d | I_B) = \sum_{i,j} w_{ij} \exp\left(-\frac{\|d - d_{ij}\|^2}{\delta_d^2}\right). \quad (7)$$

With the above formulation, the optimal pose and distance can be found with formula (3). However, optimization in a fully joint manner is too time-consuming for real-time applications, which involves evaluation of $P(\Phi, d | I_B)$ for all the values of d . We thus follow the "MaxA" strategy in [11] to find Φ^* and d^* . Specifically, d^* is first obtained by $\arg \max_d P(d | I_B)$ via the Mean-shift algorithm [20]. The corresponding regression forest for joint prediction is then chosen to retrieve the pose votes for the voting pixels. With these votes, Φ^* can be obtained by maximizing formula 4 via the Mean-shift algorithm for each of the individual ϕ_k .

2.3. CRF training

To train the CRF, we synthesize a dataset in which the samples are captured at a set of discrete distances. The sample images are binary silhouettes, and the training sample pixels are randomly sampled from them. Each pixel p_i is annotated with $(\mathcal{L}_i, d_i, \Delta_i)$, where $\mathcal{L}_i$ is the binary context descriptor, Δ_i is the offsets between p_i and the ground truth joint positions Φ . For each discrete d , we train a separate regression forest with the samples captured at that distance, *i.e.* $\{p_i | d_i = d\}$. To learn the tree structures, we start from the root node of the

tree. At each intermediate node, a set of split functions $\{\psi\}$ are generated, each of which is associated with the different dimensions of $\mathcal{L}$ and generated simply by randomly sampling $b \in [1, M]$, *i.e.* $\mathcal{L}^b$. The optimal split function is the one that maximizes the split gain G , which is defined by:

$$G = H(A) - \sum_{l \in \{0,1\}} \frac{|A_l(\psi)|}{|A|} H(A_l(\psi)), \quad (8)$$

where A is the set of samples reaching the node, $H(A)$ is defined as the pose covariance of the samples in A , and A_l , $l \in \{0,1\}$ are the two subsets of A split by ψ , each of which contains the samples satisfying $\mathcal{L}_i^b = l$. With this criterion, the tree structure of the forest is learned with the procedure similar to that in [11]. When reaching the leaf node, we save a pose vote (Δ_k, w_k) for each joint ϕ_k , where Δ_k is the mode of the poses from the training samples reaching the leaf node and w_k is the number of samples that fit Δ_k .

The depth regression forest is learned with the whole training dataset following a similar procedure as above, except that the split gain is estimated with the depth annotations, *i.e.* $H(A)$ estimates the variance of the depths of the samples reaching the intermediate node.

2.4. Incorporating semantic contexts

The binary silhouette lacks the discriminative power for pose recovery. As the semantics of the body or hand parts prove helpful for pose estimation [12, 13], we propose to derive the semantic hand parts from the silhouette as intermediate features to predict Φ , as shown in Fig. 4. Let it be $\mathcal{S}$. Given that the hand parts are obtained, the possible joint positions are largely confined. For instance, the position of the middle fingertip is highly correlated with the hand part 8. Thus the regression forest trained with the label images of the parsed hand parts can produce more consistent joint predictions.

The RDF [14] is used for per-pixel classification of I_B into the semantic parts with the binary context descriptor $\mathcal{L}$. It is trained on the entire training dataset containing both articulation and distance variations. Each foreground pixel p in I_B is classified into one of the twelve categories in Fig. 4. With the parsed parts, the semantic context descriptor $\mathcal{S}$ is defined with the labels of a set of context points similar to $\mathcal{L}$. The labels of the context points lying on the background are still assigned with label 0, while the other context points are assigned with the labels of the corresponding hand parts.

The training and inference procedures with the semantic context descriptor $\mathcal{S}$ are similar to that with $\mathcal{L}$ in Section 2.2 and 2.3, with an extra stage of hand parsing preceding pose regression. Besides, the split gain G needs to be redefined as there are thirteen possible values for each dimension of $\mathcal{S}$ compared to only two in $\mathcal{L}$. Thus G is given by:

$$G = H(A) - \sum_{l \in \{0, \dots, 12\}} \frac{|A_l(\psi)|}{|A|} H(A_l(\psi)), \quad (9)$$

Fig. 4. Left: the hand parts partition for parsing. Right: the semantic context derived from the parsed hand parts.

and the forest trained with $\mathcal{S}$ therefore has maximally thirteen branches at each intermediate node. In the experiment we show that the pose estimation accuracy is largely improved with $\mathcal{S}$.

3. EXPERIMENT

This section presents the results on a synthesized dataset and real-world RGB images for both pose and distance estimation. The synthesized dataset is used for both forest training and quantitative evaluation of the prediction accuracy. The real-world images tests the performance of our methods in real interaction environments with the regression forests trained on the synthesized datasets. The tested methods were coded in C++/OpenCV, and tested on a server with two Intel Xeon X5675 CPUs and 16G RAM. The resolution of the images in all the datasets is 320×240 .

3.1. Quantitative evaluations on synthesized dataset

We use a 3D hand model to synthesize a hand silhouette dataset for various hand configurations. The finger joint angles are clustered into 60 postures. The global hand rotation is confined within $(-40^\circ, 40^\circ)$ for all three axes, and discretized into 175 viewpoints. This produces totally 10.5k candidate hand poses, and Fig. 5 shows several examples. The distance between the hand wrist and the camera is confined within $[0.2m, 0.5m]$. During training, the distance is discretized into 11 values with an interval of $3cm$, and 80% of the 10.5k candidate poses are used to synthesize the training images for each discrete distance, which produces 92.4k training images. During testing, each of the remaining 20% candidate poses is combined with a random distance sampled from $[0.2m, 0.5m]$ to synthesize the query image. The testing distance is made continuous to better evaluate the methods.

The prediction accuracy of d is defined as the percentage of predictions within D_d from the ground truth. Table 1 shows the results with different $D_d \in [3cm, 15cm]$, and we can see 89.7% of the predictions are within an error of $6cm$. Note that the discretization interval in the training data is $3cm$ and the testing distance is continuously sampled, this result is quite reasonable with only the binary silhouette inputs.

Fig. 5. Synthesis hand silhouettes with annotated joints.

Table 1. Prediction accuracy of d on the synthesized data.

D_d	3cm	6cm	9cm	12cm	15cm
Accuracy	62.7%	89.7%	97.7%	99.8%	100.0%

The hand pose prediction accuracy for each joint is defined as the percentage of the predictions that are within a normalized distance of D_Φ pixels from the ground truth. That is, the distance between the prediction and the ground truth is scaled based on the ground truth d with respect to a standard distance, which is $0.25m$ in our experiment. This ensures that the query images at different distances are treated equally when calculating the average accuracy. As a reference, the size of a full stretched hand at $0.25m$ is about 120×150 pixels in the image. The overall accuracy is obtained by the average of the prediction accuracies of the sixteen joint locations. In this experiment we use the CRF to predict the joint positions with both the binary context $\mathcal{L}$ and semantic context $\mathcal{S}$, which are denoted as "CRF+L" and "CRF+S". Fig. 6 provides the overall accuracies of both methods for $D_\Phi \in [6, 42]$. Note that the high accuracy with small D_Φ is more favorable as large D_Φ means imprecise predictions. Both methods achieve quite good results using only the binary silhouette inputs, *i.e.* CRF+L achieves 82.1% and CRF+S achieves 87.7% prediction accuracy for $D_\Phi = 15$, which is approximately the width of the middle finger at the standard distance of $0.25m$. Moreover, the results also show that the parsed hand parts can considerably improve the accuracy when used as the semantic context for regression, but at the extra time cost of RDF classification. CRF+L needs $52.2ms$ to process one frame, while CRF+S needs $91.4ms$. The per-pixel hand parsing accuracy is provided in Fig. 7, which is 79.0% on average.

3.2. Qualitative evaluations on real-world images

We further test the performance of the proposed method with both the binary context and semantic context descriptors on three real-world egocentric image sequences. Each sequence is captured under a different environment to test the robustness of our method, *e.g.* the meeting room, the office cubicle and the sitting room. The lengths of the sequences are between 700 and 900 frames. In these sequences the user

Fig. 6. Comparison of the hand pose prediction accuracies on the synthesized dataset with respect to different D_Φ .

Fig. 7. Hand part classification accuracy with RDF.

performs various hand postures with rapidly changing background. Here the regression forests are trained with the synthesized data in Section 3.1. Fig. 8 illustrates the results on some sample frames on one of the sequences, which contains the parsed hand parts, the recovered hand pose and distance. Due to the lack of ground truth annotations, we do not have the quantitative results. However, as shown in Fig. 8, both the parsed hand parts and the recovered articulated pose are visually very consistent to the RGB inputs. Indeed, we observe the similar results on all three sequences¹.

4. CONCLUSION

In this paper we present a novel method to recover the hand pose and distance from single egocentric RGB images in real-time. The hand silhouette is extracted with pre-trained skin color models and a wrist marker. The CRF models the distance as a hidden variable and infers the pose and distance jointly. The experimental results on both a synthesized dataset and several challenging real-world sequences show the good performance of the proposed method, and the recovered hand joints are visually very consistent with the inputs. Since no temporal info is utilized yet, we may consider applying temporal tracking to further improve the robustness of our system.

¹<https://sites.google.com/site/seraphlh/home>

Fig. 8. Hand pose and distance recovery with real-world inputs. First row: the input images. Second row: the segmented silhouette. Third row: the parsed hand parts. Fourth row: pose recovered with CRF+L. Fifth row: pose recovered with CRF+S. The texts show the recovered hand distances.

5. REFERENCES

- [1] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," 2013, CVPR.
- [2] "Google glass," <http://www.google.com/glass>.
- [3] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," 2013, ICCV.
- [4] D. Tang, T. H. Yu, and T-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," 2013, ICCV.
- [5] M. L. Gorce, N. Paragios, and D. J. Fleet, "Model-based hand tracking with texture, shading and self-occlusions," 2008, CVPR.
- [6] N. Stefanov, A. Galata, and R. Hubbard, "A real-time hand tracker using variable-length markov models of behavior," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 98–115, 2007.
- [7] J. Xu, Y. Wu, and A. Katsaggelos, "Part-based initialization for hand tracking," 2010, ICIP.
- [8] V. Athitsos and S. Sclaroff, "Estimating 3d hand pose from a cluttered image," 2003, CVPR.
- [9] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic markov network," 2003, ICCV.
- [10] B. Stenger, A. Thayananthan, P. H.S. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," 2006, IEEE Trans. PAMI.
- [11] M. Sun, P. Kohli, and J. Shotton, "Conditional regression forests for human pose estimation," 2012, CVPR.
- [12] M. Dantone, J. Gall, C. Leistner, and L. V. Gool, "Human pose estimation using body parts dependent joint regressors," 2013, CVPR.
- [13] H. Liang, J. Yuan, and D. Thalmann, "Resolving ambiguous hand pose predictions by exploiting part correlations," 2014, IEEE Trans. CSVT.
- [14] L. Breiman, "Random forests," 2001, Mach. Learning, 45(1):5-32.
- [15] M. Kolsch and M. Turk, "Robust hand detection," 2004, IEEE Int'l Conference on Automatic Face and Gesture Recognition.
- [16] Y. Zhao, Z. Song, and X. Wu, "Hand detection using multi-resolution hog features," 2012, IEEE Int'l Conference on Robotics and Biomimetics.
- [17] A. Mittal, A. Zisserman, and P. H. S. Torr, "Hand detection using multiple proposals," 2011, British Machine Vision Conference.
- [18] M. Gonzalez, C. Collet, and R. Dubot, "Head tracking and hand segmentation during hand over face occlusion," 2010, ECCV Workshop on Sign, Gesture and Activity.
- [19] S. Suzuki and K. Abe, "Topological structural analysis of digital binary images by border following," 1985, Computer Vision, Graphics and Image Processing.
- [20] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," 2002, IEEE Trans. PAMI.