Appears in: Digital Image Computing: Techniques and Applications (DICTA), Canberra, Dec 2018.

Hand Detection using Deformable Part Models on an Egocentric Perspective

Sergio R. Cruz
Department of Computer Science
City University of Hong Kong
Hong Kong
scruzgome2-c@my.cityu.edu.hk

Antoni B. Chan
Department of Computer Science
City University of Hong Kong
Hong Kong
abchan@cityu.edu.hk

***Abstract*—The egocentric perspective is a recent perspective brought by new devices like the GoPro and Google Glass, which are becoming more available to the public. The hands are the most consistent objects in the egocentric perspective and they can represent more information about people and their activities, but the nature of the perspective and the ever changing shape of the hands makes them difficult to detect. Previous work has focused on indoor environments or controlled data since it brings simpler ways to approach it, but in this work we use data with changing background and variable illumination, which makes it more challenging. We use a Deformable Part Model based approach to generate hand proposals since it can handle the many gestures the hand can adopt and rivals other techniques on locating the hands while reducing the number of proposals. We also use the location where the hands appear and size in the image to reduce the number of detections. Finally, a CNN classifier is applied to remove the final false positives to generate the hand detections.**

I. Introduction

The egocentric perspective has the characteristic of seeing the world as we humans see it – the videos move quite swiftly, which makes the images and objects taken from this perspective to be blurry and hard to detect. When having this perspective we find the hands being the most consistent objects and they appear in a reasonable size that allows enough feature extraction for detection when doing daily activities. Hand detection is a special case on the object detection problem – due to the shape variation it presents, the many joints and fingers of the hand drastically change their appearances creating a hard challenge to tackle. This has led to focusing instead on generating hand proposals as a first step, and using a more computationally expensive classifier to score. When interacting with the world the hands are the means of interaction with other people and objects, which provides information on the environment and the activities the user is doing. This makes the hands appear in the image consistently, since a person concentrates on other people or objects they interact with and makes activity recognition and other objects detection use hand detection in their processes [1], [2].

Some approaches present limited environments by never changing the location during the activities [2], resulting in low variation of illumination and background. Other works choose to approach the problem using segmentation [3], [4], which makes them able to detect hands and other body parts that resemble the skin.

In this work we address hand detection as an object detection problem, instead of using segmentation to detect hands as in previous works, with the person performing daily activities while changing locations, meaning that the egocentric video will contain a variety of objects and backgrounds. We use object detection based on a pictorial approach, where we learn the whole object as well as the parts of the object. This way we are able to detect the hands using the shape, even if the fingers change the appearance of the hands, as we detect the hands as a combination of the whole hand and the fingers. The object parts are an important characteristic and it allows for more flexibility since fingers can move quite drastically. We also use the knowledge that the hands appear in specific areas in daily activities, which can be used as prior knowledge about the position and size of the hand in the image.

II. Related Work

The egocentric perspective is attracting more research due to the new technologies that are being developed and made available to the public, which generates more data to work with. One of the areas of interest using this new perspective is activity recognition [1], [2], [5]–[9], which uses the objects in the image or the background to recognize the activity. From the egocentric perspective, an object the person is interacting with can represent the activity. This however causes the objects move swiftly and have different appearances, which makes the images to be blurry and objects hard to detect. Using the egocentric perspective, video retrieval and summarization methods were also developed [10]–[13]. These papers have as the main task to obtain and visualize the most important parts of video sequences by focusing on the objects the person interacts with over time, and creating the story or sequence of stories derived from the videos. The egocentric perspective also gives the opportunity to do research on object recognition and tracking [14]–[18], where they focus on the different appearances the hand can take but require a hand or object detector. Another work that has been done is in virtual and augmented reality [19], while other works also get information out of this perspective [20]. For the specific case of hand detection in egocentric videos, there is not much work being done since it is a recent topic, and most of the approaches focus on pixel-level segmentation [14], [21]–[23] to detect the hands.

One of the first to handle something close to hand detection was Ren and Gu [17], which is not hand detection specific, as

it concentrates on any object that behaves like a hand. They segment the image using optical flow patterns, and detecting if the pattern corresponds to that of a hand, since there is a noticeable difference between the flow of the hands and the background. However, this makes any object that moves like a hand to be detected. In contrast to [17], we concentrate on the shape of the hands to disambiguate the hands from the objects the hands interact with.

The model proposed by Bambach *et al.* [24] uses a CNN-based technique for detecting, identifying, and segmenting hands in egocentric videos of multiple people interacting with each other. They use the information of a hand to be in a specific location, with a specific size and with color-like features to generate hand proposals, and then use the CNN to score them. Their dataset contains specific activities that include playing board games while sitting down, without changing environments. They concentrate on detecting the hands of multiple people instead of just the user. This approach however needs 2,500 object proposals to find the hands in the image. In our work, we generate a much smaller number of proposals, by using both shape features and location to generate proposals.

Finally, Li and Kitani [14] propose a more diverse database focusing on the variation of illumination. They select the best color feature model for each environment using scene-level feature probes. The database they propose has drastic changes in the background simulating daily activities, such as walking and grabbing objects in the kitchen. They segment the image and find which features are able to distinguish the hand from the background the best. However, these color features also detect other body parts as well. In our work, we focus on the hands alone using the shape to avoid detecting skin like features.

III. Deformable Part Model (DPM)

In order to model the hand, we need a model that can adapt to the many appearance variations of the hand. Since we want to address a daily activity database we also need to handle changes in the background and illumination. In order to address the hand detection problem we chose the Deformable Part Model (DPM) proposed by Felzenszwalb *et al.* [25]. The features used in this approach are the histograms of oriented gradients (HOG). DPM learns an object by considering the whole object as well as the parts that compose the object, and the spatial relationship among the whole object and the parts, as shown in Figure 1. In this example, when representing a hand, DPM uses the fingers as some of the parts, which allows this model to detect a hand even if the fingers change location. In order to use the parts of the object effectively DPM represents the parts at double the resolution for the HOG features, which allows more detail for the local representation of the object. This model provides a score by combining the whole object and the parts with the spatial information – if a part is farther away the score decreases.

Learning a DPM only requires the bounding box annotation of the object, with the part annotations learned automatically. To train models using partially labeled data, DPM uses a latent variable formulation of MI-SVM [26], called latent SVM (LSVM). DPM also can represent different perspectives of the object called components, as seen in Figure 1. This makes DPM able to address the different appearances the hand can take, e.g., a palm and a fist.

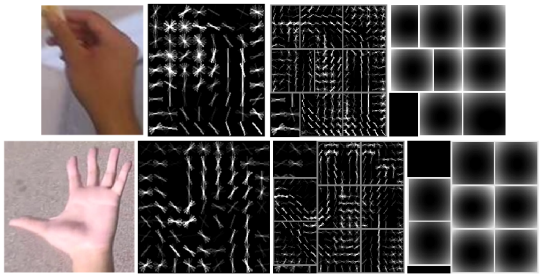

Fig. 1. Multiple hand representations found by DPM, fist (upper row) and palm (lower row), from the training sample (far left). The DPM consists of the whole object representation (middle left), the parts representation (middle right) and the spatial deformation (far right).

A. Formulation

DPM first extracts a dense feature map using linear filters. A feature map is defined as an array whose entries are d-dimensional feature vectors computed from a dense grid of locations in an image. A filter is defined as an array of d-dimensional weight vectors, and the score of a specific filter F at a position (x, y) in a feature map G is the dot product of the filter F and a sub-window of the feature map with it's top-left corner being at position (x, y),

$$F \cdot G(x, y) = \sum_{x', y'} F[x', y']G[x + x', y + y']. \quad (1)$$

Let F be a $w \times h$ filter. Let H be a feature pyramid and $p = (x, y, l)$ specify a position (x, y) in the l-th level of the pyramid. Let $\phi(H, p, w, h)$ denote the vector obtained by concatenating the feature vectors in the $w \times h$ subwindow of H with top-left corner at p in row-major order. The score of F at p is $F' \cdot \phi(H, p, w, h)$, where F' is the vector obtained by concatenating the weight vectors in F in row-major order. The score is denoted as $F' \cdot \phi(H, p)$ since the subwindow dimensions are implicitly defined by the dimensions of the filter F.

The DPM represents an object with n parts as a $(n + 2)$-tuple $(F_0, P_1, ..., P_n, b)$, where F_0 is the whole object representation or root filter, P_i is a model for the i-th part, and b is a real valued bias term. Each part model is composed by a 3-tuple (F_i, v_i, d_i) where F_i is the i-th part representation or part filter, v_i is a two-dimensional vector which represents the parts location relative to the root position which is called the anchor, and d_i is a four-dimensional vector which are the coefficients of a quadratic function, representing the area where the part can move, relative to the anchor position.

The DPM is able to detect an object at a particular position and scale within an image using a feature pyramid H, with each level having the image at a different resolution. Let the $p = (x, y, l)$ specify a position (x, y) in the l-th level of the pyramid. Given a candidate location p_0, DPM calculates the score of this location by combining the score of whole object model, and the highest possible scores of the parts,

$$\text{score}(p_0) = \max_{p_1,...,p_n} \text{score}(p_0, \ldots, p_n), \quad (2)$$

where the point set $(p_0, ..., p_n)$ corresponds to the root and hypothesized part locations. The score of the hypothesized parts' locations considers both the part's appearance and spatial location,

$$\text{score}(p_0, ..., p_n) = \sum_{i=0}^{n} F_i' \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b, \tag{3}$$

where the displacement of the part relative to the anchor position is represented as

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i), \tag{4}$$

and the deformations features as

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2). \tag{5}$$

IV. Learning DPM for Hand Proposals

One characteristics of DPM is its ability to find different appearances or components using the training data, but it requires the number of components be provided. In the original work [25], DPM finds the components using the bounding boxes size ratio as the criteria for splitting the database. They assume that different views of the object will correspond to different bounding boxes ratios. However in this work we propose to do split the components using the shape of the appearances. We also use the location of the hands in the image to restrict the detections. We do this by learning one DPM for the object itself and then use the spatial information to help trim the false positives. Our training pipeline is illustrated on Figure 2.

A. DPM Initialization for Hands

For this work we learn the hands by changing the initialization of the DPM to be unsupervised and to make it more suitable for hands. In the original formulation [25], given the number of components, DPM splits the data for the components using the bounding boxes ratio. This would create components that combine appearances that share the same bounding box ratio. However, for egocentric hands, we found that the hands can take different appearances with similar bounding box ratio, as seen in Figure 1, making the original splitting method not suitable for extracting meaningful clusters of hand appearance.

We propose to use the appearance of the hands to do the splits instead. First we mirror flip all the left hands, so that all hands are normalized to the right hand. We then extract HOG features from each data sample, and cluster them. We select k-means clustering to split the data over a hierarchical clustering approach, since a hierarchical approach can create small clusters that negatively affects the DPM performance due to its sensitivity to database size. This allows us to separate the data samples into the different perspectives or components even though they might have similar bounding box ratios. To estimate the number of components in an unsupervised way, we use the Silhouettes clustering criterion [27] with k-means. Finally, we split each cluster into two using the bounding box ratio, since the DPM is still sensitive to the bounding box ratio for detection. We do not split the data even more to avoid creating too many slices and affecting the performance negatively. Our proposed initialization method is unsupervised and adapts to the nature of the data. For other DPM settings, we use the default parameters. In particular, we use eight parts per component and HOG features, using 8 pixel bins as windows. Finally we lower the SVM threshold in the DPM to -1.1 and set the interval to 10 to get high recall as implemented in the author's code [25].

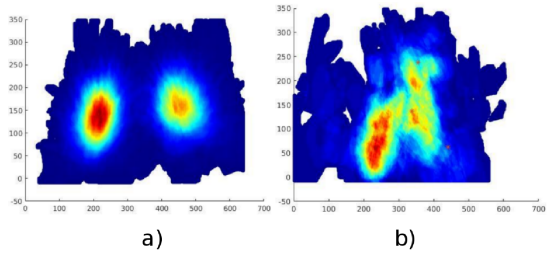

Fig. 3. Hand pixel distribution of the combined EDSH1 and EDHS2 datasets (a), and the EDSHK dataset (b), from the Li and Kitani [14] dataset.

B. IOU Suppression

In the original formulation, DPM has a post-processing step where it uses non-maximum suppression (NMS) to reduce the raw detections it produces. NMS however can suppress too many of the detections by suppressing true positives. Here, we propose a variation of the NMS based on the intersection-over-union (IOU), which takes both bounding boxes areas rather than only one bounding box area as the NMS does, We will call this IOU suppression.

C. Proposal Pruning

When learning an object detector we often concentrate on the object itself, more information can be used for hand detection under an egocentric perspective. Specifically, the hands tend to concentrate in a particular area of the image while doing daily activities. This suggests we can use the location to help prune the hand proposals from the false positives. Figure 3 shows a heat map of the hands' locations in the Li and Kitani's database [14]. The Li and Kitani database [14] consists on 3 main portions EDSH1, EDSH2 and EDSHK, where EDSH1 and EDSH2 changes location similarly going in and out of rooms, and EDSHK where the person is in the kitchen making tea and interacting with the kitchen. The hand locations in the EDSH1 and EDSH2 databases are concentrated in the middle of the image, while in EDSHK the area with hands changes slightly as the person performs a different activity. This suggests that hands are not used in the same way depending on the activity, and we can use this spatial information to trim the detections we know are false positives, since the hands never go to some areas. Figure 4 presents the multiple detections after the IOU post-processing with the bounding boxes from the training data. Some detections are too far away and the shape is too different from what would we expect the hands to have, we can use this knowledge to prune the detections.

In order to use the spatial information to prune the detections we use a classifier to differentiate the possible hands and the false positives that are clearly not hands. We train a Support Vector Machine (SVM) to learn the size and location

Fig. 2. Training pipeline of our proposal generation method. We use the bounding box ground truth to train the DPM and find the different appearances. To train the Spatial SVM we use the DPM to detect the hands on the training dataset and suppress the detections using IOU suppression. As positives we use the detections whose intersection over union with the ground truth is over 0.5, every other detection is taken as negative.

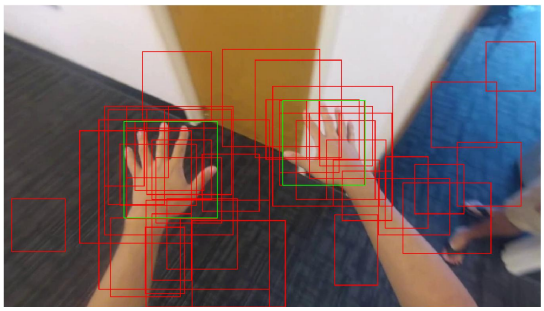

Fig. 4. Hand DPM detections after the IOU suppression (red) and training bounding boxes (green) on a Li and Kitani [14] database sample.

information from the DPM detections. Let D be set of DPM detections obtained by applying the DPM on the training data, and T the training dataset. Denote P_T and N_T to be the positive and negative training sets respectively,

$$P_T = \{d \in D, t \in T : \text{IOU}(d,t) > 0.5\} \tag{6}$$

where the intersection-over-union between d and t is

$$\text{IOU}(d,t) = \frac{|d \cap t|}{|d \cup t|}. \tag{7}$$

The negative set $N_T = D \setminus P_T$ corresponds to detections that do not intersect with any ground truth bounding box. We downsample the negatives set using a random subset to match the positives set size. Finally, we train an SVM using input features $(x, y, width, height)$ to classify the positive and negative sets.

V. Experiments

A. Datasets

To compare our approach we use the databases provided by Li and Kitani [14], which we denote as EDSH, and by Bambach *et al.* [24], which we denote as EgoHands, as they are realistic datasets with high variability. The EDHS dataset consists on 3 portions (EDSH1, EDSH2 and EDSHK) containing different scenarios and actions. We consider these as they are the most variable portions – EDSH1 and EDSH2 change rooms and illuminations, and EDSHK contains hands with the most variable appearances as they handle multiple objects (EDSHK). When comparing on EDSH, we use k-fold cross validation with $k = 5$ to make the results more robust since this database is not built for object detection. We made sure to have each database portion proportionally equally represented.

When comparing on EgoHands we use all the hands without differentiating left from right or different people, and we use their main split with all the activities and locations. This database contains multiple people interacting with each other by playing board games, which makes the appearances of the hands more complex since they interact with the world and each other. Even though this database does not concentrate on changes of illumination, it does present illumination variations when changing activities and backgrounds.

B. Methods

We denote DPM as our trained deformable part model using the intersection over union non maximum suppression, and SVM as the svm with the model spatial features. To compare our hand proposals we use various hand detection and generic object detection methods:

- **Bambach** [24] uses three probability distributions to generate bounding boxes proposals: the hand occurrence in the image, the probability of the bounding box to have a specific location and size, and the probability of the bounding box center pixel to be of that of a hand. We use the code provided by the author to generate object proposals.
- **Selective Search** [28] uses a hierarchical grouping algorithm to split the image into superpixels and generate bounding boxes from them, which is a base for CNN object detections [29]. We use the code provided by the authors and set the threshold $k = 50$.
- **Random Prim** [30] transforms the image to superpixels and creates a connectivity graph with the nodes being the superpixels and the edges being the probability of the superpixels being inside the object. This generates random partial spanning trees with large expected sum of edge weights, which then are transformed into bounding boxes. We use the code provided by the author with the fast and slow version which differ from having low and high recall respectively.
- **Objectness** [31] is an object proposal method which combines in a Bayesian framework several image cues measuring characteristics of objects, the color contrast, edge density and superpixel's straddling to generate bounding boxes. We use the code provided by the author and trained it using random database images.
- **RPN** [32] generates bounding boxes using a convolutional neural network (CNN) which outputs the bounding boxes predictions from features maps. We train RPN on the full training dataset using the code provided by the authors. We use the VGG-16 net version with default settings. As RPN needs a large amount of samples we only use it on the EgoHands database.

For the Random Prim [21] and Selective Search [20] methods we consider only the first 2,500 proposals. For the other

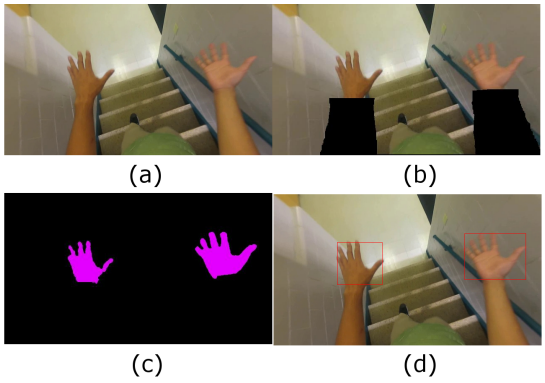

Fig. 5. Li and Kitani [14] database transformation, from original sample (a), to a hands only image (b), to hand segmentation (c) and the resulting bounding boxes (d).

methods we input 2,500 as the number of proposals when using the authors' code.

After generating hand proposals we use the CNN provided by Bambach *et al.* [24] on each method's bounding box proposals to perform the final hand detection. Since the DPM bounding box does not contain context around the hand, the CNN is run on bounding boxes that are enlarged by 1.25 times to cover the entire hand. We also consider an improved DPM detection scheme, denoted a DPM+, where the DPM and CNN scores are combined to obtain the detection score,

$$score = sDPM + \lambda * sCNN, \tag{8}$$

where $sDPM$ is the DPM score and $sCNN$ is the CNN score.

Since the approach proposed by Li and Kitani [14] uses segmentation there cannot be a direct comparison, so we use their approach and then transform the results. We provide this comparison to as a baseline with the dataset while also acknowledging its unfairness. To transform from segmentation to bounding boxes we do as follows. First we change the testing database so that the only skin like features are that of the hands (either left or right) and we set all the other skin-like pixels to black, thus creating images with only hands with skin-like features, as seen in Figure 5. When creating the new images we expand the black pixel mask since the mask is not perfect and we don't want any pixels influencing the results from other skin areas. In order to help their approach, once we have a mask for each image we find all the connected components and calculate the centroid of each one, and for the testing we take a mask as positive if the centroid falls inside a ground truth bounding box.

C. Evaluation metrics

To score the bounding boxes we use the PASCAL VOC criteria where a detected bounding box is considered positive if the intersection over union with a ground truth bounding box is over 0.5. We evaluate the hand proposal approaches using the recall to see how many of the ground truth objects we are finding. The recall represents all the ground truth objects correctly detected, without taking into consideration the false positives, and it can be taken as an upper limit as to what an approach can achieve. Since we are dealing with object

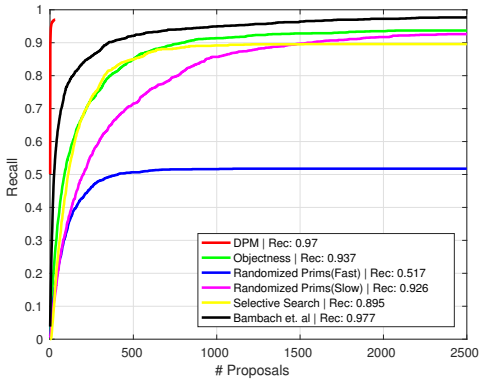

Fig. 6. Recall for hand proposal methods on EDSH.

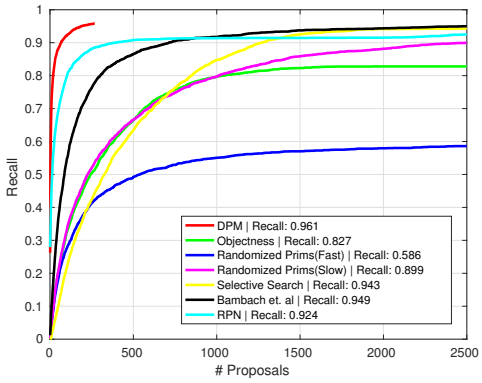

Fig. 7. Recall for hand proposal methods on EgoHands.

proposals pruning we expect to have a decrease of in the recall. We use the average precision to compare the overall detection performance.

D. Results

Figure 6 presents the recall behavior of the methods for proposal generation on the EDSH database. Bambach *et al.* [24] has the highest recall but it requires generating more than 2,000 proposals. In contrast, our method reaches its recall peak with less proposals and the recall is similar. Selective Search, Random Prim and Objectness have similar high recall, showing that color and edge features can be used to detect hands with high illumination variability.

Figure 7 presents the recall behavior of the proposal methods on the EgoHands database. Overall we achieve the highest recall and reach the recall peak with less proposals. DPM is able to find the hands on a database with a highly variable illumination and quickly moving cameras, which is needed on the egocentric perspective.

Selective Search [28] and Bambach *et al.* [24] achieve similar high recall, showing that color, size and location provide information for hand detection even in occluded instances, as the database contains hands interacting with objects and other people. RPN [32] struggles finding some hands but is able to reach high recall with only 500 proposals, as every other

TABLE I. COMPARISON OF THE NUMBER OF PROPOSALS VERSUS RECALL ON EDSH

Steps	Mean	Std. Dev.	Recall
Li and Kitani [14]	1.76	1.01	0.6943
DPM	30.58	9.61	0.9707
DPM + SVM	21.76	6.74	0.9268
Bambach *et al.* [24]	2500	0	0.9774
Selective Search [28]	1152	292.23	0.8958
Random Prim (Fast) [30]	376.6	227.45	0.5176
Random Prim (Slow) [30]	1895	460.38	0.9261
Objectness [31]	1878	155.7	0.937

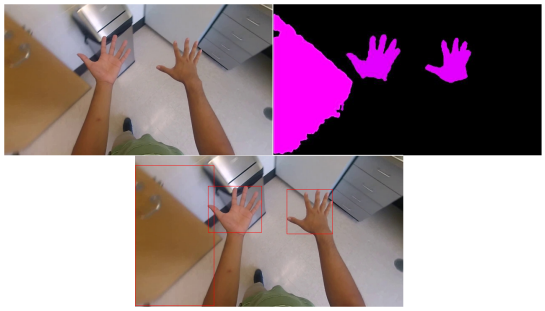

Fig. 8. Testing sample (upper left) where Li and Kitani [14] encountered hand-like color features during segmentation (upper right), which led to false positives (bottom).

method except the DPM needs more than 1,000 proposals to achieve high recall.

Table I presents the average number of proposals the methods generate on EDSH. Our approach generates the second least number of proposals after Li and Kitani [14], but achieves higher recall. Compared to other methods our approach generates considerable less proposals. For the overall performance, we find that the SVM deletes hands if they are too far away in the outer region, which makes it reduce the performance even if we are able to delete on average 10 to 30 percent of the false positives. This is due to the database consisting of only a small portion containing the person interacting with the world, and most of time staying in the same area as the person walks in and out of rooms. Li and Kitani [14] performance is affected by similar color features in EDSH, as seen in Figure 8. The skin-like features found, like wood (tables and doors), negatively affect the performance by either creating false positives or merging with the hands.

Table II presents the number of proposals generated with each method on EgoHands. Our method shows a considerable decrease in the hand proposals from the other methods, as the SVM is able to decrease the number or proposals by about 40%. The SVM has a small impact on the recall, but provides a good tradeoff, especially when comparing with the number of proposals of the other approaches.

Table III presents the times it takes for each method to generate the proposals per image. Overall, our method using CPU takes longer than other methods, as some methods like Bambach *et al.* [24] and Random Prim (Fast) [30] focus on speed. This is a tradeoff for the decrease of number of proposals previously shown. Bambach *et al.* [24] is the fastest

TABLE II. COMPARISON OF THE NUMBER OF PROPOSALS VS. RECALL ON EGOHANDS.

Steps	Mean	Std. Dev.	Recall
DPM Raw	4712.3	2282.0	0.9616
DPM	268.71	92.93	0.9616
DPM + SVM	162.40	53.65	0.9164
Bambach *et al.* [24]	2500	0	0.9499
Selective Search [28]	2500	0	0.9431
Random Prim (Fast) [30]	1844	707.6	0.5860
Random Prim (Slow) [30]	2500	0	0.8994
Objectness [31]	1943	180.5	0.8279
RPN [32]	2500	0	0.9249

TABLE III. PROPOSAL GENERATION AND HAND DETECTION TIMES ON THE EGOHANDS DATASET.

Methods	Proposal Generation Mean (s)	Proposal Generation Std.Dev.	Hand Detection w/ CNN (s)
DPM (CPU)	6.050	0.422	6.09
DPM (GPU)	0.724	0.067	0.77
Bambach *et al.* [24]	0.038	0.006	0.48
Selective Search [28]	2.752	0.534	3.20
Random Prim (Fast) [30]	2.038	0.916	3.08
Random Prim (Slow) [30]	9.241	0.555	9.68
Objectness [31]	7.297	1.107	7.64
RPN (CPU) [32]	7.112	0.644	7.56
RPN (GPU) [32]	0.318	0.050	0.76

method since it can be taken as barely using the image for the proposal generation, followed by RPN [32], as it uses GPU to speed up the CNN. However, when also implemented on GPU, DPM has similar running time to RPN [32].

Finally, Figure 9 presents the precision-recall curve for hand detection on EgoHands. DPM+ shows overall the best performance with the highest AP. DPM+ also obtains higher recall than other methods, showing that our approach can find more hands in heavily occluded environments more accurately. The combination of the CNN and the DPM score is able to improve the performance showing that usefulness of the HOG features and the DPM hand representations of parts and whole. For comparison, RPN [32] performs worse in terms of AP, which demonstrates that there is still room for improvement using deep learning for the hand detection problem. Bambach *et al.* [24] performs worse than shape-focused methods, showing that shape-focused based methods can generate hand proposals more accurately. Moreover location and size can help improve other approaches, like the DPM, by pruning false positives. Selective Search [28] and Random Prim [30] show that color features are limited for hand detection due to illumination changes and occlusion. For hand detection, DPM+ and RPN have similar running time on GPU.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a combination of Deformable Part Models and spatial features SVM for hand detection on an egocentric perspective that achieves higher recall and overall performance than state-of-the-art methods. We showed that DPM can find the hands despite the egocentric perspective having drastic illumination changes and the hand presenting complex appearances. Moreover our approach provides a considerable decrease in the hand proposals without

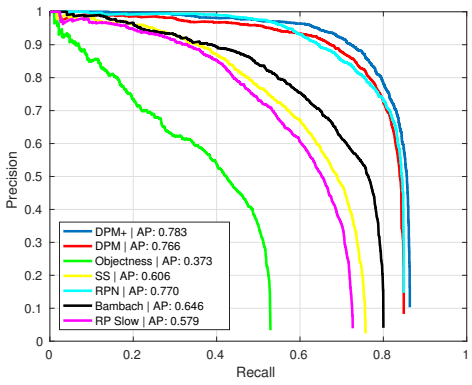

Fig. 9. Precision-Recall curves on the EgoHands database.

dramatically reducing the recall. Using the DPM proposals, hand detection performance becomes better than RPN. For future work, we plan to use end-to-end methods for object detection based on deep neural networks.

REFERENCES

[1] Y. Poleg, C. Arora, and S. Peleg, "Temporal segmentation of egocentric videos," in *CVPR*, 2014.

[2] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 407–414.

[3] A. Betancourt, "A sequential classifier for hand detection in the framework of egocentric vision," ser. CVPRW '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 600–605.

[4] C. Li and K. M. Kitani, "Model recommendation with virtual probes for egocentric hand detection," ser. ICCV '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 2624–2631.

[5] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *CVPR 2012*, June 2012, pp. 1226–1233.

[6] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *CVPR 2012*. IEEE, 2012.

[7] M. S. Ryoo and L. Matthies, "First-person activity recognition: What are they doing to me?" in *CVPR 2013*, June 2013, pp. 2730–2737.

[8] ——, "First-person activity recognition: Feature, temporal structure, and prediction," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 307–328, Sep 2016.

[9] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. Sateesh Babu, P. Phyo San, and N.-M. Cheung, "Multimodal multi-stream deep learning for egocentric activity recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[10] W. Min, X. Li, C. Tan, B. Mandal, L. Li, and J. H. Lim, "Efficient retrieval from large-scale egocentric visual data using a sparse graph representation," ser. CVPRW '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 541–548.

[11] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *CVPR 2012*, June 2012, pp. 1346–1353.

[12] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," ser. CVPR '13, Washington, DC, USA, 2013, pp. 2714–2721.

[13] E. H. Spriggs, F. De la Torre Frade, and M. Hebert , "Temporal segmentation and activity classification from first-person sensing," in *IEEE Workshop on Egocentric Vision, CVPR 2009*, June 2009.

[14] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *CVPR 2013 IEEE Conference on*, June 2013, pp. 3570–3577.

[15] S. Lee, S. Bambach, D. J. Crandall, J. M. Franchak, and C. Yu, "This hand is my hand: A probabilistic approach to hand disambiguation in egocentric video," in *2014 CVPRW*, June 2014, pp. 557–564.

[16] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 3281–3288.

[17] X. Ren and C. Gu, "Figure-ground segmentation improves handled object recognition in egocentric video," in *CVPR 2010*, June 2010, pp. 3137–3144.

[18] M. Kolsch and M. Turk, "Robust hand detection," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, May 2004, pp. 614–619.

[19] T. Gnther, I. S. Franke, and R. Groh, "Aughanded virtuality - the hands in the virtual environment," in *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, March 2015, pp. 157–158.

[20] J. Finocchiaro, A. U. Khan, and A. Borji, "Egocentric height estimation," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2017, pp. 1142–1150.

[21] A. Betancourt, P. Morerio, E. I. Barakova, L. Marcenaro, M. Rauterberg, and C. S. Regazzoni, *A Dynamic Approach and a New Dataset for Hand-detection in First Person Vision*. Cham: Springer International Publishing, 2015, pp. 274–287.

[22] A. Betancourt, P. Morerio, E. Barakova, L. Marcenaro, M. Rauterberg, and C. Regazzoni, "Left/right hand segmentation in egocentric videos," *Computer Vision and Image Understanding*, vol. 154, pp. 73 – 81, 2017.

[23] S. Huang, W. Wang, and K. Lu, "Egocentric hand detection via region growth," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 639–644.

[24] S. Bambach, S. Lee, D. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *ICCV*, 2015.

[25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[26] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 561–568.

[27] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[28] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[29] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[30] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized prim's algorithm," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 2536–2543.

[31] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, Nov 2012.

[32] R. G. Shaoqing Ren, Kaiming He, "Faster R-CNN: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.