Hierarchical Context-Aware Hand Detection Algorithm for Naturalistic Driving

Tian Zhou, Preeti J Pillai, Veera Ganesh Yalla and Kentaro Oguchi
Intelligent Computing Division
Toyota InfoTechnology Center USA
Mountain View, California, USA
{tzhou, gyalla, ppillai, oguchi }@us.toyota-itc.com

***Abstract*—Hand is a critical component in understanding driver's behaviours. Current vision-based hand detection algorithms perform poorly in naturalistic settings, due to various challenges such as global illumination changes and constant hand deformation and occlusion. To achieve a more accurate and robust hand detection system, this paper presents a hierarchical context-aware hand detection algorithm, which explicitly explores context cues in the vehicle such as prevalent hand shapes and locations, preferred driving habit and coupling effect between multiple hands. The proposed context-aware hand detection algorithm significantly outperforms the state-of-the-art on the VIVA hand dataset.**

I. Introduction

This paper proposes a hierarchical context-aware hand detection algorithm for in-vehicle applications. Hand detection is a critical module for human activity understanding, as hands are common mediums for expressing and conveying information [1]. Successful human activity understanding has a great potential for many important in-vehicle applications. For example, potential distraction activities can be recognized through analysis of the interactions between hands and common objects (cellphones, coffee cups and beverages etc) [2]. Accurate hand monitoring can also enable the study of preparatory maneuvers, issuing alert for unsafe maneuvers [3].

There are mainly two approaches to detect hands in the vehicle: touch-based and vision-based. The *touch-based* methods rely on the physical interaction between hands and critical devices in the vehicle (steering wheel, shifting gear and hand brake etc) for hand detection [4]. Though relatively more reliable, these methods cannot capture hands moving in the air. The *vision-based* methods rely on cameras to capture the scene and computer vision algorithm to detect hands. They are capable of recognizing hands every where in the camera view, but suffers from degraded accuracy and robustness.

Hand detection in the naturalistic settings features many challenges, such as volatile global illumination changes, shadow artifacts, hand occlusion and deformation, non-hand color similarity and changing viewpoints [5]. Such challenges have caused the general-purpose hand detectors to perform poorly. However, together with the extra challenges, this specific scenario has also brought in abundant contextual information, which could be used to improve hand detections. For example, the driver's specific driving habit can give a strong priori to filter out uncommon hand poses. Some people prefer to drive with two hands placed on the steering wheel while others with one hand. Therefore, this paper proposes one such algorithm which explicitly explore the contextual information in the vehicle and integrate them with conventional hand detectors, resulting in a more accurate and robust context-aware hand detector. More specifically, this paper makes the following contributions:

- Propose a hierarchical hand detection framework, which consists of context prior estimation, the context-aware model and post-processing steps to aggregate confidences from multiple channels.
- Propose the design and implementation of a context-aware hand detection algorithm, which leverages context cues such as prevalent hand shapes and locations, preferred driving habits and coupling effect between multiple hands.

II. Related Work

The study of human hand detection is an active field in computer vision, human-machine interaction and advanced driver assistance communities. The relevant literature within the scope of this work is summarized below.

General-purpose hand detector: existing general-purpose hand detectors are mainly built based on skin-color similarity [6], edge-based boosting techniques [7] and motions [8]. Hands are also detected as part of a human pictorial structure, when multiple human parts are visible [9]. Mittal [10] fused confidences from various channels, including hand shape, forearm shape and skin-similarity and achieved state-of-the-art performance on several benchmarks.

Hand detector for vehicle: the naturalistic driving condition features more challenges and demands tailed algorithms. A region-based hand detection algorithm was proposed for in-vehicle hand activity recognition [11]. A comprehensive comparison of color and depth features for hand gesture recognition in the vehicle was presented by [12]. In-vehicle hand tracking and motion trajectory analysis was proposed by [13]. Some public datasets have also been released for studying hand detection (VIVA [5]) and 3D hand gesture analysis (CVRR-Hands [14]).

Context detector: context has been used for general object detection task, to calculate saliency [15] and gist of the scene [16]. [17] presents an empirical evaluation of the role of

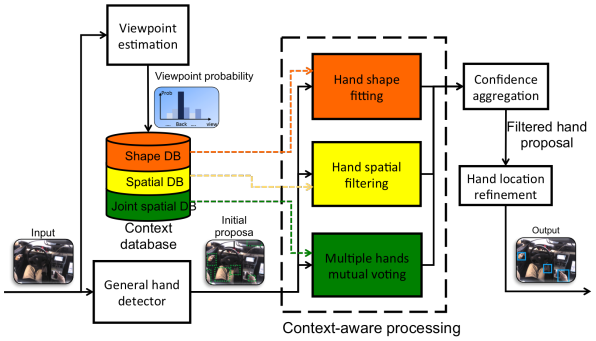


Fig. 1. System architecture for the hierarchical context-aware hand detector

context in the general object detection task. However, the aforementioned context information is mainly based on color, texture and spatial support of the image, ignoring context cues for the specific application scenario.

III. Methodology

This section presents the hierarchical context-aware hand detection algorithm. The system architecture will be firstly introduced, and then the detailed description for each module. The system architecture is shown in Figure 1. For a given input image, the viewpoint of the image was estimated and the associated context models were recalled. A general-purpose hand detector firstly proposed the initial hand proposals, which were then processed by the context detectors, including shape fitting to get rid of uncommon proposal shapes, spatial filtering to enhance common hand locations and mutual voting to enhance confidence for prevalent hand group configurations. The confidences from the three context detectors as well as the initial general-purpose detector were aggregated together and then went through the hand location refinement process to correct any placement error. Each module will be described in the following with detail.

A. Viewpoint estimation

The viewpoint of the camera is a critical priori for context models. After the viewpoint changes, the context model will also change correspondingly. Therefore, given an input image I and a set of pre-defined viewpoints V, the viewpoint estimation modules determines the most-likely viewpoint v^* following the equation:

$$v^* = \arg\max_{v \in V} P(v|G(I)) \tag{1}$$

where $G(I)$ denotes a set of global color and texture features of the input I. The color features encode information of brightness and prevalent color components, represented by color histograms in grayscale and LUV color spaces. The color histogram is calculated on a 8×8 grid on the entire image, with 64 bins for each histogram. The texture features encode information of the appearance of present objects and the overall scene, represented by histogram of oriented gradient (HOG) [18]. The HOG features were extracted on a 8×8 grid size, with 9 orientations and no overlapping cells. The color and texture features were concatenated together and then fed into a SVM model to estimation probability $P(v|G(I))$.

After the optimal viewpoint v^* is determined, the corresponding context detectors will be recalled. Notice that in an actual in-vehicle application the viewpoint information is usually given, thus eliminating the need of viewpoint estimation. But here for generalization purposes this module was still included.

B. General hand detector

The initial hand proposals were generated using a common general-purpose hand detector. In this paper, the state-of-the-art Aggregate Channel Features (ACF) object detector [19] was adopted. The ACF detector utilizes 10 input channels, including 1 normalized gradient magnitude channel, 6 gradient orientation channels and 3 LUV color channels. Features are generated by aggregating and smoothing the input channels, and a decision-tree AdaBoost classifier is trained for classification. Object detection is performed using a sliding-window approach. The output of the ACF detector is a set of axis-aligned bounding boxes (denoted as bb_{ACF}) along with a score (S_{ACF}) proportional to the detection confidence.

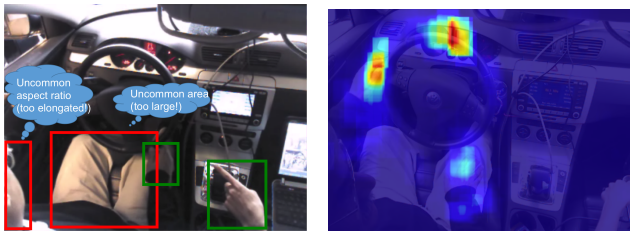
C. Context-aware hand detector

There are three context detectors, characterizing the shape (section III-C1), spatial distribution (section III-C2) and joint spatial distribution (section III-C3) of hands, which will be described in detail in the following.

1) Shape fitting: The shape fitting module builds a model to characterize common hand shapes. Four features were utilized to characterize the proposal shape, named width (w), height (h), aspect ratio ($r = w/h$) and area ($a = w \times h$). The model estimates the distribution of these four features from ground-truth hand proposals, and then calculates a fitting score for each proposal. The uncommon shapes will score lower, such as false positives of large lap area if wearing light-coloured trousers, which have a similar color as hand but has an uncommonly large area. Also the same with thin and long shapes, which is an uncommon shape for hand. An illustration of the shape-fitting model is given in Figure 2a.

The Gaussian Mixture Model (GMM) was used to model the distribution of each bb shape property. One GMM model was built for each viewpoint, using corresponding data from that viewpoint. A GMM model is fully characterized by parameter $\theta := \{\theta_1, \theta_2, ..., \theta_K\}$ where $\theta_i := \{\omega_i, \mu_i, \Sigma_i\}$, with ω_i as the priori probability, μ_i as mean and Σ_i as covariance of one of the K Gaussian components. Under the trained GMM model θ, the fitting score of an input x is $p(x|\theta) = \sum_{i=1}^{K} \omega_i \mathcal{N}(x|\mu_i, \Sigma_i)$, characterizing the likelihood that x was generated by the GMM model.

Training of the GMM model was achieved using EM algorithm. The two hyper-parameters to tune in a GMM model are the mixture number K to control over-fitting, and the covariance type Σ to control computational cost. Both hyper-parameters were selected automatically by use of Bayesian



(a) Shape-fitting model (b) Spatial filtering model

Fig. 2. *(left)* Rectangles are hand proposals, red indicating uncommon shapes whose scores will be lowered. *(right)* The spatial distribution for hand likelihood was displayed in JET color space (red for high probability and blue for low). As revealed, there are three common locations (two on the steering wheel, one on the lap area) where the driver prefers to place hands.

Information Criterion [20]. The average fitting score during training with the optimal GMM model was recorded as the normalization factor γ. The trained model was parametrized as the following :

$$bbs_x(x; d, K^*, cov^*, \gamma) \tag{2}$$

where x stands for input feature, d for input dimensionality, K^* for optimal mixture number, cov^* for optimal covariance type (spherical, diagonal, tied or full) and γ for normalization factor. One GMM model was trained for each bb property individually, and one jointly for all four of them.

In usage, the relevant bb shape information was extracted and then fed into the trained GMM model. The normalized fitting score (dividing by γ) was used to assess the confidence of the proposed bb, from the perspective of each shape property. The average of the 5 normalized fitting scores will be the output of this module, denoted as S_{bbs}.

2) Spatial filtering: The spatial filtering module builds a spatial model for the likelihood of hand presence over the entire image. Figure 2b shows a sample spatial distribution as the output of this module. Using such a model, the hand proposals from rare locations (which are usually false positives) are easier to be filtered out. One spatial model was trained for each viewpoint, since the spatial distribution for different viewpoints differ significantly.

To build the spatial distribution model, each ground truth hand casts vote to its surrounding regions, which is jointed determinedby by hand location, hand size and a spread factor γ_w. The full spatial model generation process is shown in Algorithm 1.

In usage, given a hand proposal in the testing image, its location was normalized following line 8 of Algorithm 1. Then the spatial model $\overline{M}_s$ was queried at the normalized hand location (x_n, y_n), and the resultant score will be denoted as $S_{spatial}$. The only hyper-parameter in this algorithm is the window spread factor γ_w. The smaller the value is, the broader effect each ground truth hand will impose to the spatial model. The optimal value of γ_w was selected through a grid search process ($\gamma_w = 2, 4, 8, 16$).

3) Mutual voting: The mutual voting module aims to build a joint-spatial distribution model to increase the confidences of those hands, who together form a more likely configuration. For example, some drivers prefer to put both hands on the steering wheel, resulting in a constant distance between two hands. If two hands are detected to have a distance similar to the diameter of the steering wheel, their confidences should be increased. Figure 3 shows an illustration of the joint spatial distribution.

In training stage, the pairwise relationship between ground-truth hand pairs was modelled. Given two ground truth hands located at (x_1, y_1) and (x_2, y_2) from the same image whose size is (w_i, h_i) , firstly the two hands are swapped (only if necessary) to make sure that after swapping, hand2 is to the right of hand1, i.e. $x_2 > x_1$. By doing this only one direction will be modelled to prevent repeated calculation

Algorithm 1 Spatial Model Generation

1: **Input:** training images D, window shrink factor γ_w, spatial model shape $(w_0 = 640, h_0 = 480)$, .
2: $M_s \leftarrow zeros(h_0, w_0)$ ▷ init spatial model with zeros
3: **for all** img **in** D **do** ▷ for each training image
4: $w_i, h_i = getShape(img)$ ▷ get width and height
5: $hands \leftarrow getHands(img)$ ▷ get ground truth hands
6: **for all** $hand$ **in** $hands$ **do** ▷ for each hand
7: $[x, y, w, h] \leftarrow hand$ ▷ extract bounding box info
8: $x_n = x/w_i \cdot w_0,\ y_n = y/h_i \cdot h_0$ ▷ normalize x,y
9: $w_n = w/w_i \cdot w_0,\ h_n = h/h_i \cdot h_0$ ▷ normalize w,h
10: $l = max(0, x_n - w_n/\gamma_w)$ ▷ left index
11: $r = min(w_0, x_n + w_n/\gamma_w)$ ▷ right index
12: $t = max(0, y_n - h_n/\gamma_w)$ ▷ top index
13: $b = min(h_0, y_n + h_n/\gamma_w)$ ▷ bottom index
14: $M_s[t : b, l : r] += 1$ ▷ update spatial model
15: **end for**
16: **end for**
17: $\overline{M}_s = \frac{M_s - mean(M_s)}{std(M_s)} + 1$ ▷ normalize spatial model
18: **return** $\overline{M}_s$

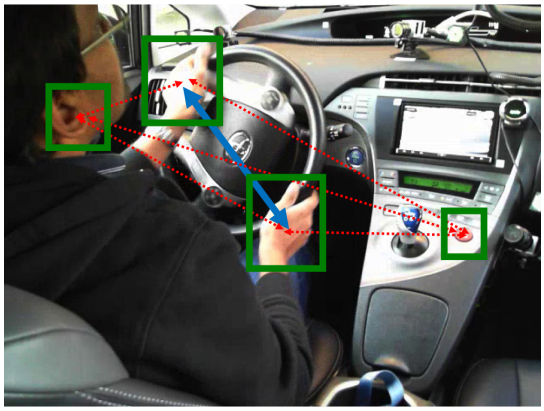


Fig. 3. Illustrations of how joint spatial distribution works. The green rectangles are the initial hand proposals and all the arrows indicate the mutual voting procedure. Red arrows indicate a pair-wise relationship with low probability, while blue arrows for high probability. In this example, the two hands are placed at a distance observed commonly before (the diameter of the steering wheel), thus both confidences will be enhanced.

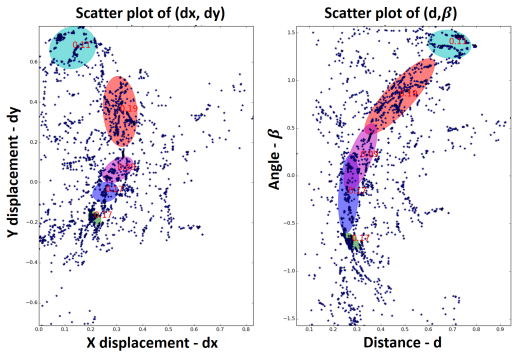


Fig. 4. Scatter plot of the features extracted for the joint spatial distribution, together with the fitted GMM model. The fitted GMM models are represented using coloured ellipse, showing 5 components with highest weights (red text). (left) shows the (dx,dy) scatter plot while (right) shows the (d,β) scatter plot.

from reciprocal symmetry. After the swapping, the pair-wise relationship between the two hands was characterized using a set of 4 values (dx, dy, d, β) where

$$\begin{aligned} dx &:= (x_2 - x_1)/w_i \qquad dy := (y_2 - y_1)/h_i \\ d &:= \sqrt{dx^2 + dy^2} \qquad \beta := arctan(dy/dx) \end{aligned} \tag{3}$$

describing distance, angle, normalized x-displacement and normalized y-displacement. For each image from the training dataset, the pair-wise features among all hand pairs (if more than 1 hand present) were extracted. For the image consisting of n hands, $n(n-1)/2$ pair-wise features will be generated. Such features from all the training images were firstly extracted and then modelled using a GMM model. The optimal hyper-parameters for GMM were selected automatically using BIC. The average GMM fitting scores β was recorded for normalization purposes later. A sample result of the fitted GMM model was shown in Figure 4, together with scatter plot of the extracted relationship features. As revealed, the fitted GMM model captures the common layout of hands, where two hands are mostly placed in a horizontal line with a distance similar to the steering wheel diameter.

In usage, the reciprocal relationship between each pair of hands was captured in the same way as training, and then fitted using the trained GMM model. For each proposed hand x, the fitting scores from all the other hands y were summarized following (4)

$$\begin{aligned} p(x) = \sum_{y \in Y, y \neq x} p(x,y) &= \sum_{y \in Y, y \neq x} p(x|y)p(y) \\ &= \frac{1}{|Y|} \sum_{y \in Y, y \neq x} p(x|y) \end{aligned} \tag{4}$$

where Y indicates all the hand proposals in the current frame. An equal prior was assumed for all hand proposals and the GMM fitting score was used to approximate $p(x|y)$. The likelihood score $p(x)$ was then normalized using β and the resultant score was denoted as S_{joint}.

4) Confidence aggregation: The confidence aggregation module fuses the confidences from the various channels optimally. For each proposed bounding box bb, there is the original ACF score S_{ACF}, the bb shape fitting score S_{bbs}, the spatial filtering score $S_{spatial}$ and the joint spatial distribution score S_{joint}. The aim is to calculate the optimal weight to combine these four scores, i.e $S = \alpha_1 S_{ACF} + \alpha_2 S_{bbs} + \alpha_3 S_{spatial} + \alpha_4 S_{joint}$. The random search technique [21] was adopted. The four weights were randomly initialized following a uniform distribution, i.e. $\alpha_i \sim \mathcal{U}(0, 0.5)$. For each iteration, the step size also follows a uniform distribution, i.e. $\delta\alpha_i \sim \mathcal{U}(-0.1, 0.1)$. The weight will be updated whenever an improvement of performance was observed after the step jump. The random search process stops either after reaching the maximum allowed iterations or has reached a performance plateau. Due to its local optimization nature, the random initialization was repeated ten times and the weight achieving the best performance was chosen as the final optimal weight.

D. Hand location refinement

The hand location refinement module corrects placement errors of hand proposals using bounding box regression technique. Intuitively speaking, the regression model adjusts hand proposal locations after observing certain cues. For example, if a portion of hand is observed in the lower right corner of the bounding box, it should be shifted towards the lower right direction aiming to better fit the entire hand. Given a hand ground-truth G_i, an associated hand proposal P_i and its local features $\phi(P_i)$, a ridge regression model ω^* was trained to shift proposals following $\omega^T \phi(P_i)$, so that the error between projection and the ground truth was minimized,

$$w^* = \arg\min_w \sum_i ||G_i - \omega^T \phi(P_i)||^2 + \lambda ||w||^2 \tag{5}$$

The local features $\phi(P_i)$ include color and texture features, extracted in the same manner as section III-A, but now only on the local patch P_i. The regularization parameter λ and the association threshold (a required minimum Intersection-Over-Union overlap between P_i and G_i) were optimized through grid search on validation set ($\lambda = 1, 10, 100, 100$ and $IoU = 0.1, 0.3, 0.5$).

IV. Experiments

A. Experiment setting

The VIVA hand dataset [5] was used to evaluate the proposed hand detection algorithm. The dataset consists of color images captured under naturalistic driving conditions, from 4 different vehicle types, 8 drivers and 7 different viewpoints. The annotation of hands is given in a form of axis-aligned bounding boxes, specified by top-left point, width and height $bb = [x, y, w, h]$. Since only the annotations of the training split were released publicly, the experiment was conducted based on cross-validation evaluation on the training split.

The VIVA hand challenge established two difficulty levels in evaluation, with L1-level only evaluating backview images, and L2-level evaluating images from all 7 views (more

difficult). Due to the multi-view detection capability of the proposed algorithm, the L2-level was selected as the sole focus of the experiment. As suggested by the challenge, the average recall (**AR**) and area under the PR curve (**AP**) were used as the metric to evaluate detection performance. AR was calculated from the ROC curve over 9 evenly sampled points in log space between 10^{-2} and 10^{0} false positives per image. A hand proposal is considered correct when it satisfies the PASCAL criterion, i.e. the proportion of the overlap between the proposed bb and the ground truth bb is greater than 0.5 [22]. The organizers reported the performance of the benchmark algorithm (ACF) on the testing split, L1-AP: 70.09%, L1-AR: 53.84% L2-AP: 60.06%, L2-AR: 40.42%. Notice that these numbers are not directly comparable to the performance of the proposed algorithm, since their evaluation was based on the testing split (whose annotations were not released). For comparison purposes, the benchmark algorithm was re-trained under the same cross-validation setup as the proposed algorithm, with recommended parameters suggested by the paper [5]. The proposed hand detection algorithm can reach 15 FPS on a PC with Intel Core i7-4790K CPU @ 4.00GHz x 8.

B. Experiment results

There are two experiments conducted, the viewpoint estimation task (described in section IV-B1 and the context-aware hand detection task (described in section IV-B2).

1) Viewpoint estimation: For the viewpoint estimation task, training images from all the viewpoints were used. Notice that the amount of images per viewpoint is quite unbalanced. There are 283 images from unmounted view (view0), 0 images from front left (view1), 48 images from front right (view2), 3380 images from back (view3), 118 images from side (view4), 1478 images from top down(view5) and 193 images from first-person view (view6). Therefore, the discriminative model balances the sample weight of different viewpoints. The color and texture features were extracted, concatenated together and then fed into a SVM model (with RBF kernel) for training. In testing the model output the probability of viewpoints of the input image (e.g. Backview - 0.62, Frontview - 0.13, Sideview - 0.02 etc) and the viewpoint with the maximum probability was selected as the decision. The stratified 10-fold cross validation (with shuffling) was used to preserve the class ratio in each split. The entire dataset was split into training (90%) and testing (10%) set for each fold. The metric used to evaluate the algorithm is the weighted accuracy, where the weight for each class is inversely proportional to class frequencies. The proposed viewpoint estimation algorithm can achieve a weighted accuracy of 99.56%, which is nearly perfect. Discussion about this performance will be given later.

2) Context hand detector: The context hand detector was viewpoint dependent, thus one set of context hand detectors was trained for each viewpoint. The following procedure will be described based on one viewpoint, and the same process will be repeated on all the other viewpoints. All the images from one viewpoint will be split using a 10-fold cross validation to get the training split (90%) and the testing split (10%). Then within the training split, 20% of randomly chosen images were held out as the validation set to tune hyper-parameters.

TABLE I
PERFORMANCES OF HAND DETECTION ALGORITHMS IN AVERAGE RECALL (**AR**) AND AREA UNDER THE PR CURVE (**AP**) FOR EACH VIEWPOINT. AVERAGE AND STANDARD DEVIATION OVER THE TEN FOLDS ARE SHOWN.

Metric	**AR**		**AP**	
Method	ACF	Ours	ACF	Ours
View 0	40.9 ± 15.2	62.9 ± 17.0	74.3 ± 5.8	82.1 ± 5.1
View 2	30.6 ± 25.1	48.7 ± 25.0	31.3 ± 16.4	45.1 ± 16.9
View 3	75.8 ± 5.9	93.9 ± 1.4	90.9 ± 1.3	95.5 ± 0.5
View 4	67.3 ± 18.0	72.5 ± 15.7	77.4 ± 8.2	77.8 ± 7.8
View 5	24.1 ± 5.1	90.7 ± 4.1	84.1 ± 2.0	96.6 ± 0.7
View 6	56.8 ± 26.3	60.8 ± 17.1	72.2 ± 11.4	73.3 ± 9.7

The benchmark algorithm trains an ACF classifier with images from all the viewpoints. To make a fairer comparison, an ACF classifier was re-trained for each viewpoint. Then the context models, as well the as the bb regression model were trained based on the output of the ACF classifier. Finally, after all the context detectors are trained, the random search procedure will be recalled to calculate the optimal weight to combine the ACF and three context scores. All the hyper-parameters were optimized based on the best AR score on the validation split. After the training was finished, the entire algorithm was evaluated on the testing split and the resultant AR and AP performances were calculated. The average performance as well as standard deviation of AR and AP on the 10 testing splits for each viewpoint is presented in Table I.

V. DISCUSSION

The viewpoint estimation algorithm achieves a nearly-perfect accuracy, mainly because training and testing split include images from the same road test, resulting similar scene properties. A cross-drive validation would evaluate the generalization capability better, but there is not enough data in the VIVA dataset for such experiment (only few drives for each viewpoint).

For the context detectors, there is a clear performance improvement using the proposed algorithm compared to the benchmark ACF algorithm, for all viewpoints. The improvement margin ranges from 5% to 65% in AR, for different viewpoints. It is interesting to notice the relationship between average performance and number of training examples for each viewpoint. The correlation coefficient between average AR and the number of the training examples for each viewpoints was calculated, and a strong positive correlation relationship ($\rho = 0.84$) was observed. Therefore, having a larger dataset would improve the performance of the proposed algorithm, showing its scalability.

VI. CONCLUSION

This paper presents a hierarchical context-aware hand detection algorithm for the naturalistic driving setting. The

algorithm consists of context prior estimation (viewpoint detection), the context-aware hand detectors, and a post-processing step to fine-tune hand locations (bounding box regression). Explored context cues include prevalent hand shapes and locations, driver habit and joint spatial distribution between hands. The confidences from various context channels were aggregated together to address the limitations of current general-purpose hand detectors. By evaluating on the public VIVA hand dataset, the proposed context-aware hand detection algorithm was found to significantly outperform the state-of-the-art hand detector.

Future work includes bringing additional context resources, such as CAN bus data which can introduce strong dynamic priori for hand presences near the triggered device. Also hand gesture recognition and human machine interaction through hand gestures are worthy of further investigation. Note that the context model proposed in this paper can be easily adopted by other domains which admit similar contextual information, such as air-plane controlling in the cockpit [23], medical image navigation in the Operating Room through gestures [24] and gesture interactions in robot-assisted tele-surgery [25], all featuring a relatively constrained and structured environment, with many task-specific context cues to leverage.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] S. Goldin-Meadow, "The role of gesture in communication and thinking," *Trends in cognitive sciences*, vol. 3, no. 11, pp. 419–429, 1999.

[2] T. Horberry, J. Anderson, M. A. Regan, T. J. Triggs, and J. Brown, "Driver distraction: The effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance," *Accident Analysis & Prevention*, vol. 38, no. 1, pp. 185–191, 2006.

[3] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, "Predicting driver maneuvers by learning holistic features," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 719–724.

[4] D. Van'tZelfde, V. Gardner, and J. Lisseman, "Steering wheel hand detection systems," Aug. 14 2014, uS Patent App. 14/178,578. [Online]. Available: https://www.google.com/patents/US20140224040

[5] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 2953–2958.

[6] X. Zhu, J. Yang, and A. Waibel, "Segmenting hands of arbitrary color," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE, 2000, pp. 446–453.

[7] M. Kölsch and M. Turk, "Robust hand detection." in *FGR*, 2004, pp. 614–619.

[8] M. Van den Bergh and L. Van Gool, "Combining rgb and tof cameras for real-time 3d hand gesture interaction," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 66–72.

[9] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman, "Long term arm and hand tracking for continuous sign language tv broadcasts," in *Proceedings of the 19th British Machine Vision Conference*. BMVA Press, 2008, pp. 1105–1114.

[10] A. Mittal, A. Zisserman, and P. H. Torr, "Hand detection using multiple proposals." in *BMVC*. Citeseer, 2011, pp. 1–11.

[11] E. Ohn-Bar and M. Trivedi, "In-vehicle hand activity recognition using integration of regions," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1034–1039.

[12] E. Ohn-Bar and M. M. Trivedi, "A comparative study of color and depth features for hand gesture recognition in naturalistic driving settings," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 845–850.

[13] A. Rangesh, E. Ohn-Bar, and M. M. Trivedi, "Long-term, multi-cue tracking of hands in vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[14] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 6, pp. 2368–2377, 2014.

[15] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[16] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 273–280.

[17] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1271–1278.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[19] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west." in *BMVC*, vol. 2, no. 3. Citeseer, 2010, p. 7.

[20] R. J. Steele and A. E. Raftery, "Performance of bayesian model selection criteria for gaussian mixture models," *Dept. Stat., Univ. Washington, Washington, DC, Tech. Rep*, vol. 559, 2009.

[21] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of operations research*, vol. 6, no. 1, pp. 19–30, 1981.

[22] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[23] T. A. Furness, "The super cockpit and its human factors challenges," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 30, no. 1. SAGE Publications, 1986, pp. 48–52.

[24] M. G. Jacob, J. P. Wachs, and R. A. Packer, "Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images," *Journal of the American Medical Informatics Association*, vol. 20, no. e1, pp. e183–e186, 2013.

[25] T. Zhou, M. E. Cabrera, and J. P. Wachs, "Touchless telerobotic surgery-is it possible at all?" in *AAAI*, 2015, pp. 4228–4230.