Sign Language Recognition using Dynamic Time Warping and Hand Shape Distance Based on Histogram of Oriented Gradient Features

Pat Jangyodsuk
Department of Computer Science and Engineering
The University of Texas at Arlington
pat.jangyodsuk@mavs.uta.edu

Christopher Conly
Department of Computer Science and Engineering
The University of Texas at Arlington
cconly@uta.edu

Vassilis Athitsos
Department of Computer Science and Engineering
The University of Texas at Arlington
athitsos@uta.edu

ABSTRACT

Recognizing sign language is a very challenging task in computer vision. One of the more popular approaches, Dynamic Time Warping (DTW), utilizes hand trajectory information to compare a query sign with those in a database of examples. In this work, we conducted an American Sign Language (ASL) recognition experiment on Kinect sign data using DTW for sign trajectory similarity and Histogram of Oriented Gradient (HoG) [5] for hand shape representation. Our results show an improvement over the original work of [14], achieving an 82% accuracy in ranking signs in the 10 matches. In addition to our method that improves sign recognition accuracy, we propose a simple RGB-D alignment tool that can help roughly approximate alignment parameters between the color (RGB) and depth frames.

Categories and Subject Descriptors

H.4 [**Gesture and motion Tracking**]: Miscellaneous

Keywords

Sign Language Recognition, Gesture Recognition, Kinect

1. INTRODUCTION

In the field of computer vision, sign language recognition remains one of the most challenging tasks. With the recent release of the Microsoft Kinect camera, depth-sensing technology has become widely available at an affordable price. The depth camera provides an additional dimension beyond that of RGB images so that a pixel becomes a point in 3D space instead of color intensity values. This greatly increases information found in features, leading to better recognition accuracy. One problem associated with RGB-D input, however, is pixel alignment between the depth and color images. Since the two sensors are separated by a physical distance, their perception of the scene is from a slightly different perspective, and there is not a one-to-one correspondence between pixels in the two images. In most cases, people use only the depth information, as properly calibrating the RGB and depth cameras is a non-trivial task. By discarding the RGB image, valuable information is lost that cannot necessarily be acquired from the depth image, such as hand shape, since the Kinect depth image is captured in low resolution. Therefore, object shape, like that of the hands, cannot be captured in detail. In this paper, we propose a simple calibration tool that approximates 4 alignment parameters, including x-translation, y-translation, x-scale and y-scale, so that the researcher can align the two images and utilize both depth and color information.

The second contribution of this paper is a proposed method that improves sign language recognition rate over that of the work from [14]. In [14], the query sign was recognized using Dynamic Time Warping as a distance measure between hand trajectories. They also compared hand shape in both the first and last frame of signs using the Euclidean distance between shape features described in the paper. While the shape features used in [14] are able to improve accuracy, they do not utilize gradient (edge) information that is the core feature in popular descriptors such as SIFT or HoG. We applied the same method, DTW, for hands trajectory comparison but use Histogram of Oriented Gradient (HoG) [5] to represent hand shape instead and achieve better results. Accuracy increases, on average, about 10%.

2. RELATED WORKS

There are numerous related works in gesture and sign recognition. The most common problems regarding sign language and gesture recognition, in general, are:

1. Image transformation between training and query data. Scale, rotation, affine and illumination, for example.

2. Noise in training and query data. While this problem is common in any machine learning application, it is exaggerated in computer vision applications, where the majority of information founded in the video may be unrelated to the task at hand.

3. Temporal segmentation of the gesture. When does the gesture start and stop?

The solutions to these problems lie in either features or recognition method.

While detecting features, a difficulty arises from the fact that, in a video, there is a lot of irrelevant information–for example, background, face, and clothes. Therefore, when extracting features, only body parts performing the gesture should be considered. This, however, is not a trivial task. Methods abound for hand detection, ranging from simple skin and motion models [19, 20] to shape-based template search [10]. Contextual search using graphical models has been popular in recent years, for example chain model [8] and skeleton tracking on Kinect depth images [7]. Using a depth camera such as the Kinect eases some difficulty in computer vision applications [18], as additional information is available. However, finding hands is still an ongoing research problem.

An alternative approach to searching for individual body parts is to extract features such as HoG [5] features from the whole frame. While this approach does not suffer from the difficulties found in body part or interest point extraction, it does capture noise and, thus, is not tolerant of image transformation.

In gesture recognition methods, the problem is viewed as one application of time series classification. Inspired by speech recognition, the most popular model is Hidden Markov Model (HMM) and its variations [6, 16, 11, 2, 17]. Dynamic Time Warping (DTW), a time series matching algorithm, is also a popular choice [4, 12, 1, 14] due to the fact that it is a distance measure and therefore, no training is required, making it a perfect choice for applications where the number of training examples is small, as is the case with ours. In more recent work, Conditional Random Field (CRF) [9] and Hidden Conditional Random Field (H-CRF) [15] improves upon HMM by removing the 1-to-1 dependency between a hidden state and observation, thus increasing overall accuracy. However, both CRF and H-CRF require a large number of training examples in order to learn a good model.

3. METHODS

Our method of sign recognition continues the work of [14]. To recognize a sign, we use two kinds of features. The first one is hand trajectory. As in [14], the features we use for single-handed signs are:

1. Hand location with respect to face position
2. Unit hand motion velocity vector. Mathematically, given hand location $h(t)$, at frame t, this feature is

$$v(t) = \frac{h(t+1) - h(t-1)}{||h(t+1) - h(t-1)||}$$

For 2-handed signs, the features are:

1. Right hand location with respect to face position
2. Unit right hand motion velocity vector.
3. Left hand location with respect to face position

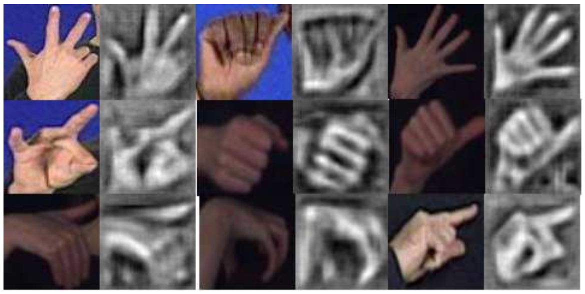

Figure 1: Hand shape sample images from the ASL data set. The color images are the annotated hand regions in the color frame. The grayscale images are the corresponding visualizations of extracted HoG features using inverse HoG [13].

4. Unit left hand motion velocity vector.
5. Right hand position with respect to left hand, designated by $f(t)$
6. Unit motion vector of $f(t)$, given as

$$d(t) = \frac{f(t+1) - f(t-1)}{||f(t+1) - f(t-1)||}$$

We extract the trajectory feature described above from any given query video sign. This feature will be compared to those in the database training signs using DTW to match hand trajectories.

The second part of our algorithm is hand shape matching. Given a hand region on the first frame and the last frame of the sign, we extract features describing hand shape. The features used in the experiment are Histogram of Oriented Gradient (HoG) features. We chose HoG features for robustness against illumination changes and for recent success and popularity in many computer vision applications. Again, the HoG features will be matched with signs in the database using Euclidean distance as a distance metric. Note that the sign type of the query (whether it is one-handed or two-handed) must also be given so that each type is only matched against others of the same type.

To recognize a given sign, Q, we retrieve the top k sign candidates, $\mathbb{S} = \{S_1, S_2, ..., S_k\}$, using the matching method described above. It is considered correctly classified if

$$\exists S_i \in \mathbb{S}, C(S_i) = C(Q)$$

where $C(X)$ is a function returning the class of given video X.

3.1 RGB-D Calibration Tool

As mentioned, we cannot apply the annotated bounding box from the depth image to the RGB image directly, due to misalignment of two images. Calibrating the Kinect camera is a non-trivial task. We propose a simple alignment annotation tool that approximates calibration parameters. These parameters are x-translation, y-translation, x-scale and y-scale. Note that the purpose of the tool is NOT to replace proper camera calibration. It is just for fast, rough and simple approximation of alignment parameters.

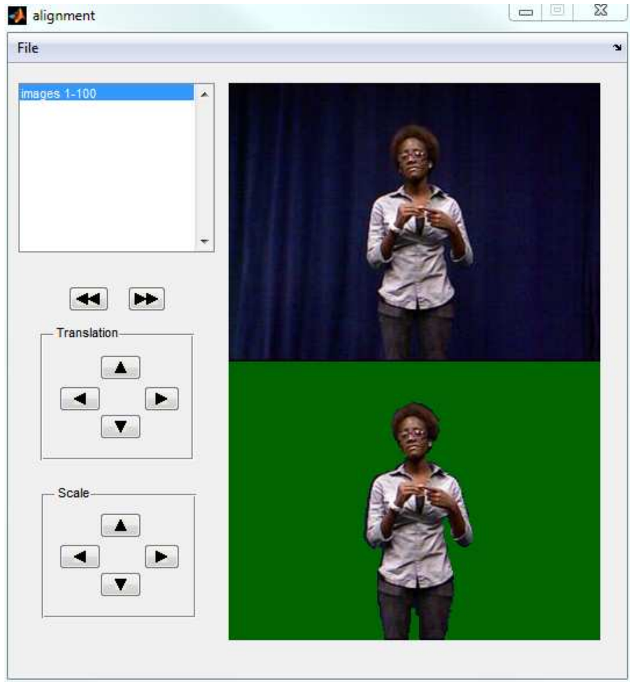

Figure 2: Sample screen shot from the alignment annotation tool. The bottom image is the working space where a human annotator adjusts the segment extracted from the corresponding depth frame to match the RGB frame. In the figure, this segment is the body segment. The top image serves as a reference image.

Figure 2 shows a screen shot of the alignment tool. To use the tool, the RGB frame and a segment extracted from the corresponding depth frame must be provided. In our case, we use body segmentation extracted from depth image. The annotator will align the segment to match that of RGB by adjusting translation and scale parameters such that it matches as they see fit. As can be seen in figure 3, the body segment extracted from depth image has been manually aligned to the RGB image. To learn the alignment parameters of a camera on a specific view, the annotator will perform the manual alignment for a certain number of frames (50 in our experiment). The final parameters are the average values.

4. EXPERIMENTS

4.1 Dataset

We conducted experiments on an American Sign Language (ASL) data set consisting of 1,113 signs. There are 2 types of data.

1. Videos from a data set captured by a standard RGB camera. No depth information is available. We have 3 signers for this type of data. Each signer performs 1,113 signs, making a total of 3,339 signs. This data set is called ASL data set in this paper.

2. Videos from a data set captured by a Kinect camera [3]. There are 2 signers for this data set for a total of 2,226 signs. It is called jku449 in this paper.

Along with the sign videos, we also have bounding box annotations for the hands and face regions. The ASL data set videos have annotations for all 1,113 signs by all 3 signers, while the jku449 data set currently has annotations for 449 signs by one signer.

4.2 Implementation

As mentioned in section 3, sign recognition is done using DTW and shape matching based on HoG features. The feature used for DTW trajectory matching is the same that were used in [14]. In addition, we have made some minor improvements by standardizing features so that the mean value is 0 and the standard deviation is 1.

HoG features parameters were extracted using inverse HoG code [13] from MIT. With ASL data set, the hand regions were extracted using manually annotated bounding boxes from the RGB images. The same cannot be done with jku449 data set since body part labeling was done on the depth images. If we extracted the shape from the depth images, we would not have accurate shape information due to the fact that depth images lack visual appearance information. As mentioned previously, we used our alignment annotation tool to learn estimated alignment parameters and applied the parameters to the depth image annotated bounding boxes. Ideally, the result is bounding boxes properly aligned with the hand regions in the RGB images.

The experiment was performed in a user-independent fashion. For ASL data set, we used signs from one signer as queries and compared them to signs from 2 other signers. The result for ASL data set was the average from all 3 signers. Since we only have annotation from one signer for jku449 data set, the query signs are compared to videos from ASL data set. To extract hand shape features for jku449 data set, we first applied the alignment parameters on the depth image annotated hand bounding boxes to get the hand region in the RGB image. Then, we extracted HoG features on the hand region from the RGB image. The quantitative measurement used was accuracy-top candidates retrieval plot. For each data set, we implement 3 methods to compare.

1. Hand trajectory matching with hand shape distance using HoG features as shape representation

2. Hand trajectory matching with hand shape distance using features in [14] as shape representation

3. Hand trajectory matching without hand shape distance.

5. RESULTS

5.1 RGB-D Alignment

Figure 4 shows examples of RGB-D alignment. It can be seen that the bounding box, while not perfect, captures the majority pixels belonging to the hands.

5.2 Sign Recognition

Figure 5 displays sign recognition accuracy. The x-axis represents the sign rank, the number of signs a user need to look up before finding correct matches at y accuracy. The legend

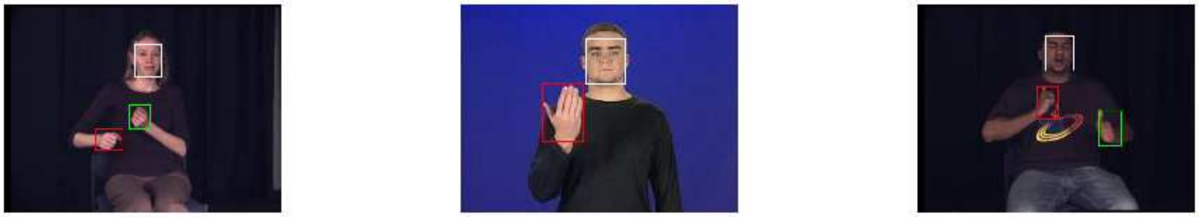

Figure 3: Sample images with annotated regions from the ASL data set. Each image is from a different signer. The rectangles bound various regions, including face and hands.

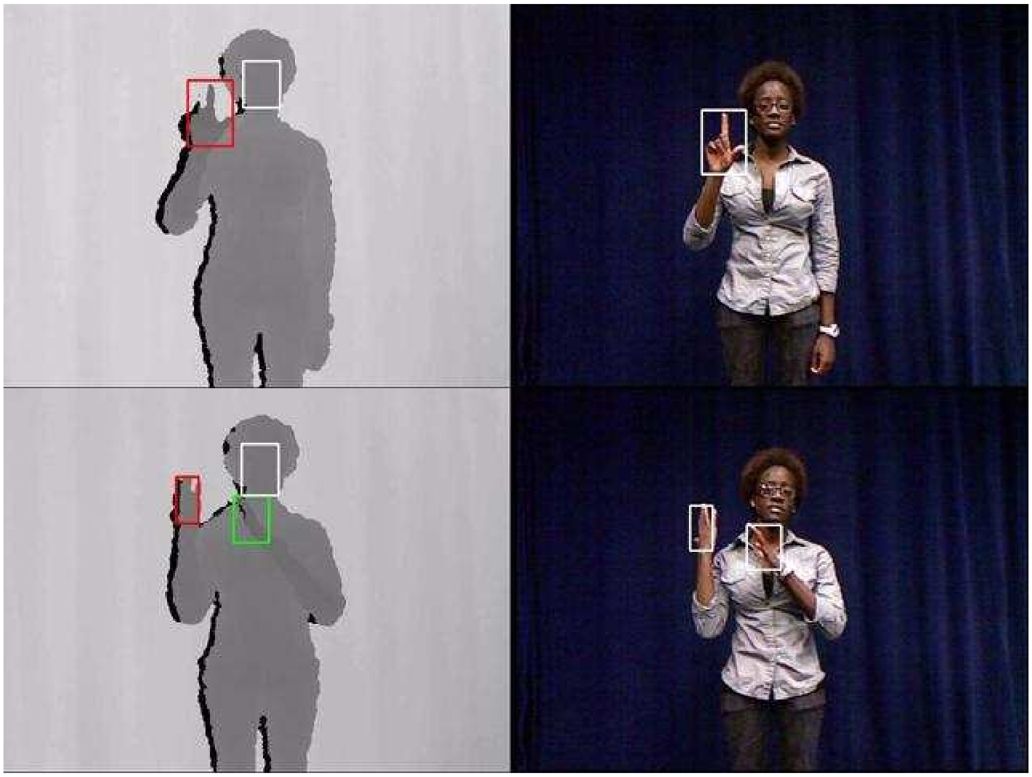

Figure 4: Visualization of RGB-D alignment. Left image is the manually annotated bounding box made on top of the depth frame. The right image is the bounding box after applying alignment parameters on top of corresponding RGB frame. It can be seen that the bounding box does encompass the majority of hand pixels

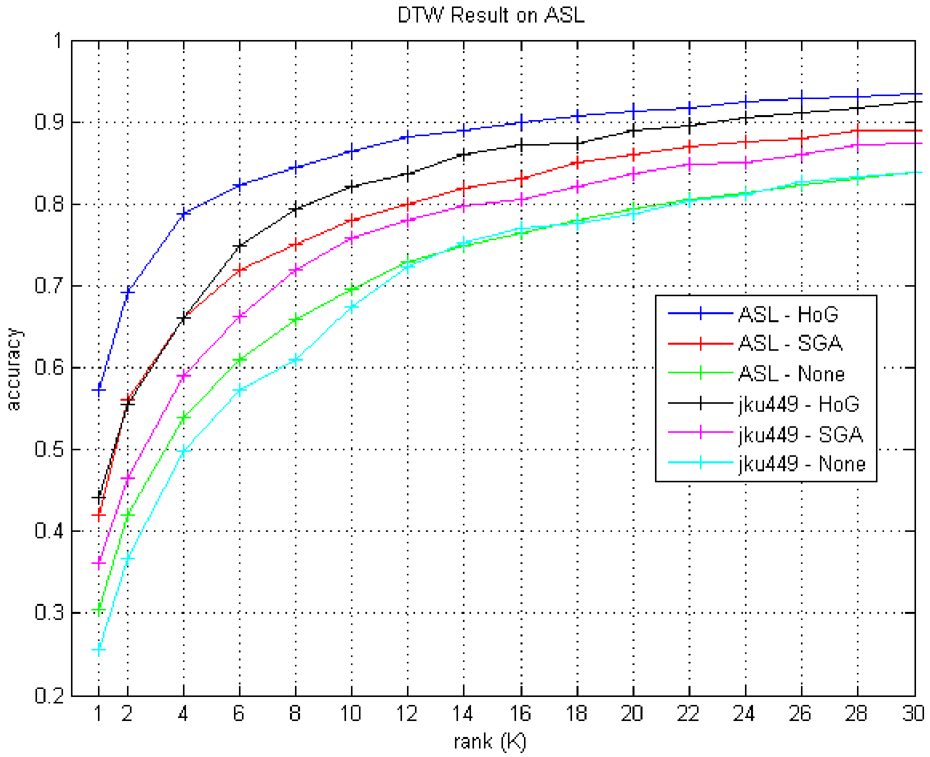

Figure 5: Sign recognition results for ASL and jku449 data set. The x-axis represents top retrieval rank and y-axis represents recognition accuracy. The legend is in the format 'Data set - Hand shape features'.

Top k	1	3	5	10	20	30
ASL - HoG	57.29%	73.94%	80.56%	86.43%	91.25%	93.38%
ASL - SGA	42%	61%	69%	78%	86%	89%
ASL - No shape	30.37%	48.05%	57.5%	69.63%	79.37%	83.95%
jku449 - HoG	44.18%	60.75%	70.45%	82.09%	88.96%	92.54%
jku449 - SGA	36.12%	52.84%	62.69%	75.82%	83.58%	87.46%
jku449 - None	25.67%	43.28%	53.58%	67.46%	78.81%	83.88%

Table 1: Sign retrieval accuracy in number. Top k refers to number of best matches

is in format 'Data set - hand shape features'. It can be seen that hand shape comparison does increase the accuracy by more than 10%. Using HoG for shape representation, the accuracy improves over using the shape presented in [14] by about 8%. At top 10 candidates retrieval, we achieved 86% accuracy compared to 78% in [14].

Accuracy on the jku449 data set is on average about 2-3% lower than that of ASL when using same method. At top 10 rank, the accuracy is 82% for jku449 data set. This is because the estimated calibration parameters, while proving to work well, are not perfect. Therefore, the extracted hand regions obtained from the color images are not always accurate. It can be seen that, without hand shape comparison, the results of ASL (green line) and jku449 (cyan line) are similar but begin to differ when hand shape is considered.

6. DISCUSSION AND FUTURE WORKS

We have demonstrated in this paper that, using HoG features, we can improve the accuracy of sign recognition on an ASL data set up to 8%. Furthermore, we introduced a simple alignment annotation tool capable of approximating the alignment parameters of RGB-D cameras.

There are a number of things left for future work. The simplest one is, using Kinect data for queries, extend the trajectory feature into 3D space. This, in theory, should give better accuracy due to the fact that more information is provided. Another idea would be to conduct a comprehensive experiment using other kinds of features or recognition methods. For instance, HMM and CRF for the recognition method or SIFT for hand shape features. Finally, we will work towards recognizing signs without user-provided information or annotations, such as hand bounding boxes, temporal segmentation and sign type.

7. ACKNOWLEDGMENTS

This work was partially supported by National Science Foundation grants IIS-1055062, CNS-1059235, CNS-1035913, and CNS-1338118.

8. REFERENCES

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1685–1699, 2009.

[2] F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745–758, 2003.

[3] C. Conly, P. Doliotis, P. Jangyodsuk, R. Alonzo, and V. Athitsos. Toward a 3d body part detection video dataset and hand tracking benchmark. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '13, pages 2:1–2:6, New York, NY, USA, 2013. ACM.

[4] A. Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 82–89. IEEE, 2001.

[5] 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*, volume 1, pages 886–893. IEEE, 2005.

[6] S. R. Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[7] A. Kar. Skeletal tracking using microsoft kinect. *Methodology*, 1:1–11, 2010.

[8] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman. The chains model for detecting parts by their context. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 25–32, 2010.

[9] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[10] A. Mittal, A. Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *British Machine Vision Conference*, 2011.

[11] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*, pages 227–243. Springer, 1997.

[12] G. Ten Holt, M. Reinders, and E. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, volume 300, 2007.

[13] e. a. Vondrick Carl. Hoggles: Visualizing object detection features, Jan. 2013.

[14] H. Wang, R. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar. A system for large vocabulary sign search.

[15] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527. IEEE, 2006.

[16] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):884–900, 1999.

[17] R. Yang and S. Sarkar. Gesture recognition using hidden markov models from fragmented observations. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 766–773. IEEE, 2006.

[18] V. M. Z. Zhang, W.H. Liu and V. Athitsos. A viewpoint-independent statistical method for fall detection. In *International Conference on Pattern Recognition*, pages 3626–3630, Nov 2012.

[19] Z. Zhang, R. Alonzo, and V. Athitsos. Experiments with computer vision methods for hand detection. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, pages 21:1–21:6, New York, NY, USA, 2011. ACM.

[20] Z. Zhang, R. Alonzo, and V. Athitsos. Experiments with computer vision methods for hand detection. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 21:1–21:6, 2011.