A Combined Method Based on SVM and Online Learning with HOG for Hand Shape Recognition

Kazutaka Shimada, Ryosuke Muto, Tsutomu Endo

Department of Artificial Intelligence, Kyushu Institute of Technology
680-4 Kawazu Iizuka Fukuoka 820-8502 JAPAN
{shimada, endo}pluto.ai.kyutech.ac.jp

Abstract. In this paper, we propose a combined method for hand shape recognition. It consists of support vector machines (SVMs) and an online learning algorithm based on the perceptron. We apply HOG features to each method. First, our method estimates a hand shape of an input image by using SVMs. Here the online learning method with the perceptron uses the input image as new training data if the data is effective for relearning in the recognition process. Next, we select the final hand shape from the outputs of the SVMs and perceptron by using the score of SVMs. The combined method deals with a problem about decrease of the accuracy in the case that users change. Applying the online perceptron jointly leads to improvement of the accuracy. We compare the combined method with a method using only SVMs. The experimental result shows the effectiveness of the proposed method.

Keywords: Hand shape recognition, SVMs, Online learning, HOG, Combination.

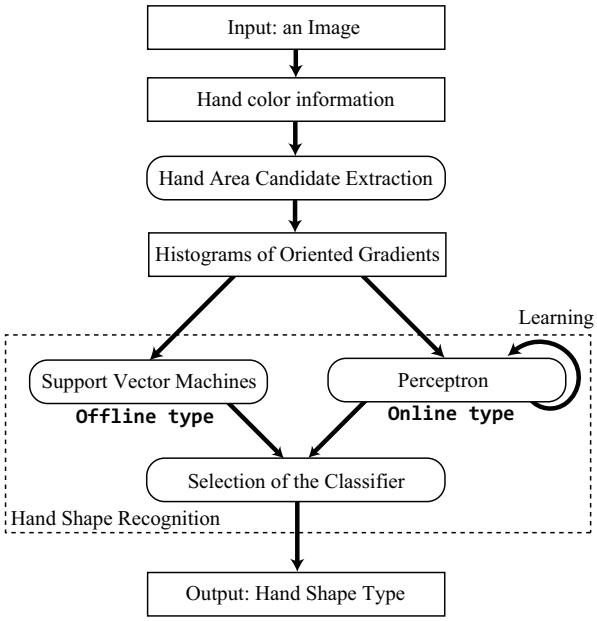

Fig. 1. The outline of our method.

1. Introduction

Human-machine interaction is one of the most important tasks in artificial intelligence. The use of hand gestures is a natural way of interacting with computers and important for an intuitive interface [1]. Many researchers have studied many hand gesture methods [2–4]. Hiranuma et al. [3] have proposed a wide-view working space system based on gesture actions. They used ring devices for robust gesture detection. In general, using particular devices is, however, costly. In this paper, we handle a task based on hand shape recognition using a USB camera.

Another problem is decrease of accuracy in the case that users change. To obtain high accuracy, recognizers based on machine learning usually need a large amount of training data. However, annotation of training data is costly. To solve this problem, we apply an online learning algorithm to the hand shape recognition. By using the online learning, our system can update the model of a classifier during the recognition process.

In this paper, we propose a method for the hand shape recognition. It consists of support vector machines (SVMs) and an online learning algorithm based on the perceptron. We use them in a complementary style. The role of the SVMs is a basic classifier for the recognition process. On the other hand, the online algorithm is used for personalization. The model of the online algorithm is updated during the recognition process. By this updating, our method is automatically customized for a current user. It leads to improvement of hand shape accuracy.

Figure 1 shows the outline of our system. First our system calibrates hand color information of a user. Next, it detects some hand area candidates from an input image by using the calibrated color information. Then, it extracts features for classifiers from the hand area candidates. In this paper, we employ Histograms of Oriented Gradients (HOG) descriptors as features. In the hand shape recognition process, our method estimates a hand shape of an input image by using SVMs. The online learning method with the perceptron also estimates a hand shape and uses the input image as new training data if the data is effective for relearning. Finally, we select the hand shape from the outputs of the SVMs and perceptron by using the score of SVMs. As an application, we develop a multi modal interface based on the hand shape recognition and a speech understanding method for a music player.

Fig. 2. The calibration process.

2. Proposed method

In this section, we explain our method. It consists of three parts: (1) preprocessing, (2) feature extraction and (3) hand shape recognition.

2.1. Preprocessing

In this method, we need to extract hand area candidates from images for the hand shape recognition. For the process, our system requires hand color information. First, we extract candidate colors for the tracking of the hand area, namely a calibration process. Figure 2 shows the color calibration process. In the calibration process, a user sets the palm to the circle in the display. The calibration process is as follows:

1 Convert an image to HSV color model,

2 Select 20 points randomly from the captured area,

3 Detect the maximum and minimum values of H, S, and V in 20 points,

4 Set ranges of hand color as follows:

$$minH - 5 \leq handH \leq maxH - 5$$
$$minS - 5 \leq handS \leq maxS - 5$$
$$minV - 5 \leq handV \leq maxV - 5$$

where $handH$, $handS$ and $handV$ are hand color of H, S, and V respectively.

2.2. HOG feature

In this paper, we use the Histograms of Oriented Gradients (HOG) descriptors reported by Dalal and Triggs [5] for classifiers in the recognition process. The HOG is one of the most effective features for human detection and vehicle detection tasks in computer vision [6, 7]. The HOG descriptors are based on counting occurrences of gradient orientation in localized portions of an image. First, the method computes gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ in each pixel (x,y).

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2}$$

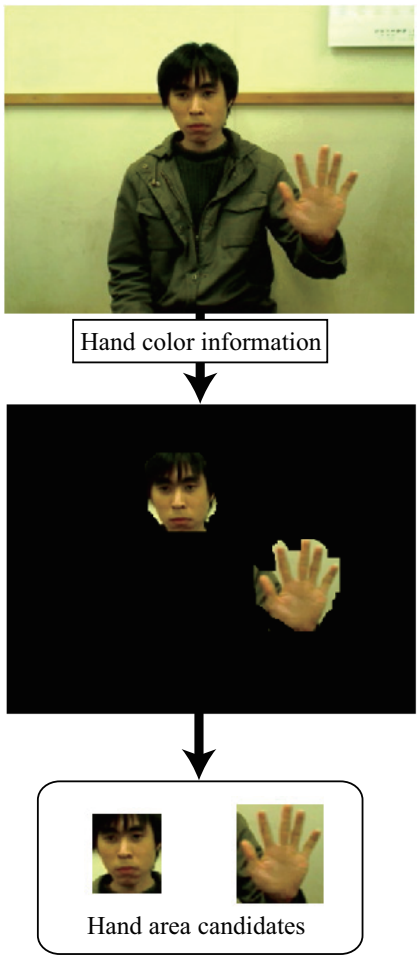

Fig. 3. The hand area candidates.

$$\theta(x,y) = \tan^{-1}\frac{f_y(x,y)}{f_x(x,y)}$$

where

$$f_x(x,y) = I(x+1,y) - I(x-1,y)$$
$$f_y(x,y) = I(x,y+1) - I(x,y-1)$$

Then, it generates the cell histogram consisting of 5 × 5 pixels. Finally, it normalizes the block consisting of 3 × 3 cells.

$$L2 - norm = \frac{v}{\sqrt{||\mathbf{V}||^2 + \varepsilon^2}}$$

where $\mathbf{V}$ is the feature vector of the block. ε is the factor for the block normalization and $\varepsilon = 1$.

We apply the HOG descriptors to hand area candidates. First, we group pixels into components based on the values, that is a labeling process. Then,we extract the rectangles as hand area candidates if the size is lower than a threshold for the size. Figure 3 shows an example of the extracted areas. The HOGs of the extracted candidates are features for classifiers in the recognition process. On the basis of the features, the classifiers judge whether an image contains a hand or not.

Here, there is an overlapping problem in the extracted candidates. The hand area often overlaps with the face

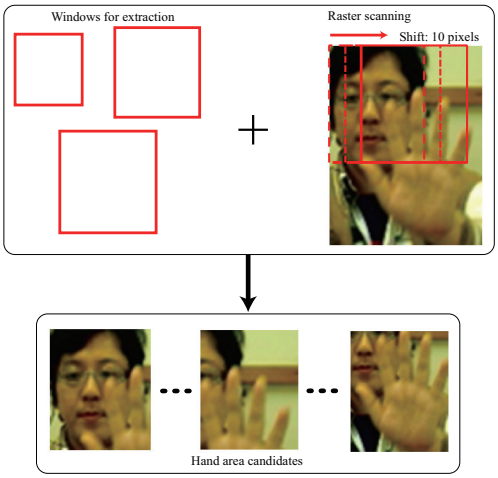

Fig. 4. The overlapping problem.

area. In this situation, this process can not extract the correct hand area. Figure 4 shows an example of the problem. The initial candidate in the figure is not suitable for the recognition process because it contains not only a hand but also a face. To solve this problem, we scan the initial candidate by using three windows if the size is not lower than the threshold. The sizes of the windows are 90×100, 60×80 and 50×60. They are determined heuristically. Finally we extract the rectangles which are more than a threshold for the hand color, as candidates for the recognition process.

2.3. Hand Shape Recognition

2.3.1. Basic idea

In this paper, we use two types of classifiers for the hand shape recognition. They are support vector machines and perceptron. We use them in a complementary style.

The SVM is the basic classifier in the process. Although it is a powerful classifier for many pattern recognition problems, it requires a lot of computational time for learning. Here we discuss a problem of the decrease of accuracy in the case that users change. In the situation, relearning of the SVM is not a realistic approach. On the other hand, perceptron based on online learning is very efficient in terms of speed, and space. Hence, the online perceptron based approach is one of the most suitable methods for relearning in the situation.

Our method learns the models of the SVM and online perceptron by using training data first. Given an input, they output the results on the basis of the current classifiers. Here, the online perceptron is updated by using the current input if some conditions are satisfied. Finally, our method select the best output from two classifiers.

2.3.2. SVMs

We use Support Vector Machines (SVMs) as the classifiers. SVMs are a machine learning algorithm that was

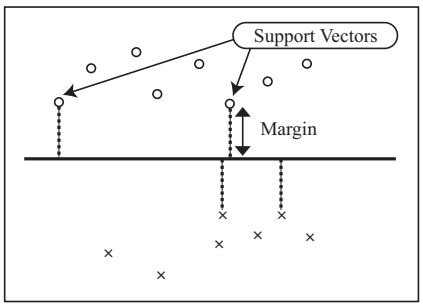

Fig. 5. Support vector machines.

introduced by [8]. They have been applied to tasks such as face recognition and text classification. An SVM is a binary classifier that finds a maximal margin separating hyperplane between two classes. The hyperplane can be written as:

$$y_i = \vec{w} \cdot \vec{x} + b$$

where $\vec{x}$ is an arbitrary data point, i.e., feature vectors, $\vec{w}$ and b are decided by optimization, and $y_i \in \{+1, -1\}$. The instances that lie closest to the hyperplane are called support vectors. Figure 5 shows an example of the hyperplane. In the figure, the solid line shows hyperplane $\vec{w} \cdot \vec{x} + b = 0$.

The SVM is a binary classifier. However, the target of this paper is a multi-class problem. Therefore, we need to extend the binary SVM into multi-label classification. In this paper, we apply the one-versus-one method, which is based on a majority voting strategy between every pair of classes, to the system.

2.3.3. Online Learning

The perceptron is a type of artificial neural network proposed by Rosenblatt [9]. The online perceptron algorithm [10] starts with an initial zero prediction vector. It predicts the label of a new instance $\vec{x}$ by using $\hat{y} = sign(\vec{v} \cdot \vec{x})$. If the prediction differs from the label y, it updates the prediction vector to $\vec{v} = \vec{v} + y\vec{x}$. In other words, it adds the input data into the weight vector. The updated weight vector is suitable to classify the mistake data correctly as compared with the previous weight vector. It becomes a better classifier for the current data set. Then, the process repeats with the next example. In the prediction process, it computes a predicted label on the basis of the updated weight vector.

The online perceptron is also a binary classifier: $y \in \{+1, -1\}$. Therefore, we need to extend it into multi-label classification. The method is as follows:

1 Generate classifiers which classify an image into each hand shape type and an image without a hand, such as a face in Figure 4,

2 Classify an input by using each classifier,

3 Select the hand shape type with the maximum score.

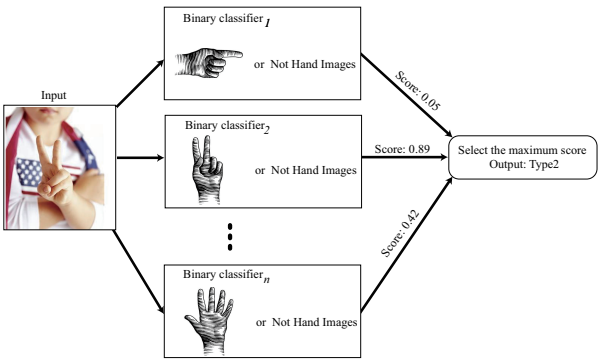

Fig. 6. Multi-label classification with binary online perceptrons.

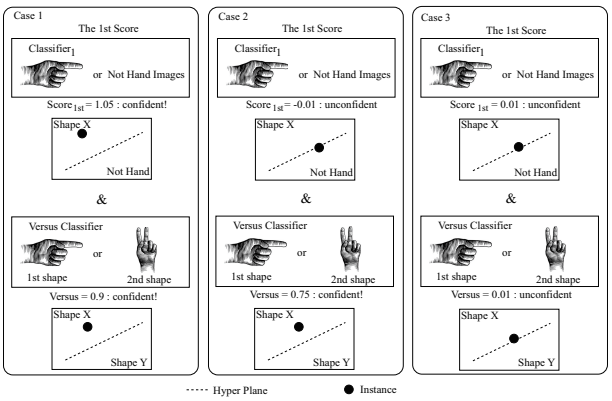

Fig. 7. Three constraints.

Figure 6 shows an example of the process.

Next, we explain a relearning process with the online perceptron. We introduce three constraints for the relearning process. The online perceptron is updated by using the current input if the constraints are satisfied. For this process, we generate all combinations of classifiers of each hand shape type (*Versus* classifiers). Figure 7 shows examples for the constraints.

- Case 1: relearn `1stShape` as new training data if $Score_{1st} > 1.0$ *and* $Versus > 0.7$
- Case 2: relearn `1stShape` as new training data if $Score_{1st} = -0.015 \sim 0$ *and* $Versus > 0.7$
- Case 3: relearn the image as new non-hand data if $Score_{1st} = 0 \sim 0.015$ *and* $Versus = 0 \sim 0.015$

where `1stShape` and $Score_{1st}$ are the hand shape with the maximum score and the score. *Versus* is the score of the *Versus* classifier for hand shapes with $Score_{1st}$ and $Score_{2nd}$. In Figure 7, the "pointing finger" shape is the `1stShape` and the "peace sign" shape is the hand shape with $Score_{2nd}$. The score of the *Versus* classifier is the output of the classifier that classifies an input image into "pointing finger" and "peace sign". The Case 1 denotes an obvious positive instance for relearning. The `1stShape` contains the high score as the hand shape and there is a large difference between the `1stShape` and the 2nd candidate. The Case 2 is also an important constraint for relearning. The $Score_{1st}$ in the Case 2 is not high. It denotes an unconfident output from the original online perceptron as the `1stShape`. On the other hand, there is a large difference between the `1stShape` and the 2nd candidate. This result shows that the `1stShape` is worth the relearning. The Case 3 denotes a latent negative instance for relearning because the score is low and there is no difference between the `1stShape` and the 2nd candidate.

2.3.4. Combination

Our method selects the final output from the SVM and the online perceptron. The selection is based on the output value from the SVM, that is the distance from the hyperplane. The value is 0.3. It is determined experimentally.

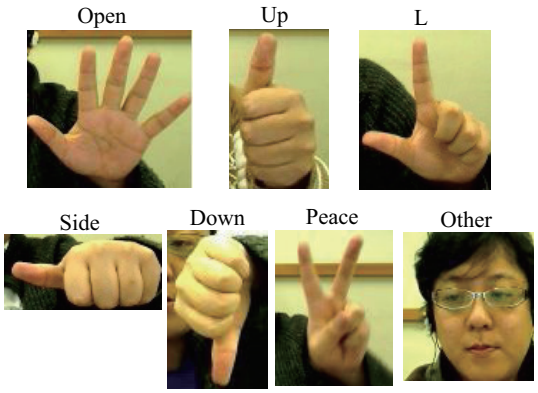

Fig. 8. The hand shapes in this experiment.

In other words, our method selects the output from the SVM if the value is more than 0.3. It selects the output from the online perceptron otherwise.

3. Experiment

We evaluated our combined method with SVMs and online perceptron.

3.1. Settings

The settings in this experiment were as follows:

- PC: OS - Windows XP, CPU - Intel Xeon X5450 2.00 GHz, Memory - 4.00GB
- Camera: Logitech QuickCam Fusion, 320 × 240, 30fps
- HOG: size: 80 × 90, 5 × 5 cells, 3 × 3 blocks, 9 directions (18144 dimensions)

In this experiment, we handled six hand shape types and non-hand. Figure 8 shows examples of them. The hand shapes occasionally included overlapped images, such as "down" in Figure 8. Here "Other" denotes images not

Table 1. The experimental result about 6 shapes (The recall rates).

Type	Method	Subject A inTrain	Subject B UnKwn	Subject C inTrain	Subject D Unkwn
Open	Baseline	1.00	0.85	**1.00**	0.75
	Proposed	1.00	**0.95**	0.95	**0.95**
Up	Baseline	0.90	0.95	1.00	0.85
	Proposed	**0.95**	**1.00**	1.00	**0.95**
L	Baseline	**0.95**	0.80	**0.95**	0.75
	Proposed	0.90	**0.90**	0.90	**0.90**
Side	Baseline	0.55	0.40	0.55	0.40
	Proposed	**0.90**	**0.75**	**0.70**	**0.70**
Down	Baseline	1.00	0.90	1.00	0.90
	Proposed	1.00	**0.95**	1.00	**0.95**
Peace	Baseline	0.60	0.50	0.70	0.50
	Proposed	**0.95**	**0.80**	**0.85**	**0.75**
Other	Baseline	1.00	**1.00**	0.95	0.95
	Proposed	1.00	0.95	**1.00**	0.95
Average	Baseline	0.85	0.77	0.88	0.73
	Proposed	**0.95**	**0.90**	**0.91**	**0.87**
AllAve	Baseline	0.81			
	Proposed	**0.91**			

containing a hand area suitable for the hand shape recognition, such as a face and an overlapped image (left and center ones in Figure 4).

For training data, we collected 560 images from 2 males and 2 females. They consisted of 20 images for each types; 20 images $\times$ 7 types (including "Other") $\times$ 4 persons. Test data also consisted of 560 images of 4 persons (2 males and 2 females). Here, 1 male and 1 female of them were not contained in the training data, namely unknown persons for our system. The online perceptron needs additional data for the relearning process. We prepared 480 images for each person as the additional data.

For implementation of the SVMs and online perceptron, we used LIBSVM[1] and OLL[2].

3.2. Result and discussion

Table 1 show the experimental result. The recall rate was computed by

$$Recall = \frac{\text{\# of images detected correctly}}{\text{\# of images of each hand shape}}$$

"Baseline" in the table denotes a method with only SVMs, that is a method without relearning by the online perceptron. "inTrain" and "UnKwn" denote test subjects included in the training data and not included in the training data, respectively. In other words, there were no customized models for the Subject B and D.

The proposed method, which was based on relearning, outperformed a simple method "Baseline" (0.91 vs. 0.81 on average). It was effective, especially for unknown persons (0.90 vs. 0.77 for B and 0.87 vs. 0.73 for D). The relearning with the online perceptron contributed to the improvement of the accuracy for even persons in the training data (0.95 vs. 0.85 for A and 0.91 vs. 0.88 for C). The accuracy rates of two hand shape types, "Side" and "Peace", by "Baseline" were insufficient. The proposed method improved those of the hand shapes dramatically. It adjusted the prediction model by using additional inputs. Most of the classification errors were produced in the type "Other". In other words, the precision rates of "Other" were not always high although the recall rates of that were high. On the other hand, classification errors between six hand shapes, which did not include "Other", were rare. These results show the effectiveness of the proposed method.

Correct Correct Misrecognized

Fig. 9. Misrecognition caused by rotation.

HOG descriptors are not invariant to rotation. We evaluated our method with several rotated images. Figure 9 shows an example of a misrecognized image caused by rotation. Approximately a 30-degree rotation was the border line to recognize the hand shapes correctly. Therefore, it can be concluded that the problem of the rotation is not critical in our system.

Another problem was the computational time of relearning and prediction in the recognition process. Our method required approximately 0.05 sec for each candidate image. In a non-overlap situation, our method usually detects 2 candidate areas (See Figure 3). Hence, it use 0.1 sec (0.05 $\times$ 2 images) for 1 frame if there is no overlap. This processing speed in non-overlap situations

1. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
2. http://code.google.com/p/oll/wiki/OllMainEn

Fig. 10. The hand shapes and the application.

might be workable as a real time system (10 frames/sec). On the other hand, our method detected many candidates in an overlap situation. The average number of detected candidates in the situation was approximately 30 images. This is a critical problem for utilizing our method as a user interface. Limitation which is to not contain any overlap situations might be unnatural. To solve this problem, we need to discuss an efficient method for the hand candidate detection. One approach for the solution is to utilize depth information [11, 12]. Using Mircosoft Kinect[3] is an easy and efficient way to detect the hand area faster and accurately. Another approach is to decrease the computational time of the relearning process itself. We used 18144 dimensions as HOG features. Reducing the dimensions leads to the decrease of the processing time. We evaluated downsized HOG descriptors which consisted of 1620 dimensions; 30 $\times$ 35 pixels. As a result, it was approximately 20 times faster than original HOG (80 $\times$ 90). However, the method with the downsized HOGs decreased the accuracy of approximately 10%. There is room to discuss the approach about the downsizing.

3.3. Application

We constructed a multi-modal interface with the hand gesture recognition and a speech understanding method as an application. The task was control of iTunes, a music player. We combined a multiple recognizer which can distinguish command utterances for the system from chats between users [13]. As inputs from hand gestures, we use 5 types as shown in Figure 10. Our system obtains outputs from each input method, it directly performs the commands. In subjective evaluation, we obtained positive comments from test subjects.

3. http://www.xbox.com/en-US/Kinect

4. Conclusions

In this paper, we proposed a combined method based on support vector machines (SVMs) and an online learning algorithm for the hand shape recognition. We used them in a complementary style. The SVMs in the method was a basic classifier and the online algorithm relearned the prediction model by using current inputs. We compared the proposed method with a baseline method which was based on only SVMs. The proposed method outperformed the baseline method. It improved the accuracy by 10% (81% to 91%). It was effective for not only unknown users but also users in the training data. The result shows the effectiveness of the proposed method, which included the relearning process. One problem of our method was the computational time of relearning and prediction in the recognition process. Although the processing speed in non-overlap situations might be workable for a real time system, that in overlap situations was insufficient. This problem was caused by the hand area candidate detection. One approach to solve the problem is to utilize depth images, such as Mircosoft Kinect, for the detection process.

Future work includes (1) improvement of computational time using a camera with depth information, (2) a large-scale experiment and evaluation with other hand shapes, and (3) discussion of features for classifiers and applying other machine learning techniques to our method.

References:

[1] Andries van Dam. Post-wimp user interfaces. *Communication of The ACM*, 40(2):63–67, 1997.

[2] W. T. Freeman and C. D. Weissman. Television control by hand gestures. In *Proceedings of IWAFGR 95*, pages 179–183, 1995.

[3] S. Hiranuma, A. Kimura, F. Shibata, and H. Tamura. Interface design of wide-view electronic working space using gesture operations for collaborative work. In *Proceedings of HCI 2007*, pages 1332–1336, 2007.

[4] K. Oka, Y Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *Computer Graphics and Applications*, 22(6):64–71, 2002.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[6] F. Han, Y. Shan, and R. Cekander. A two-stage approach to people and vehicle detection with hog-based svm. In *PerMIS*, pages 133–140, 2006.

[7] Y. Yamauchi and H. Fujiyoshi. People detection based on co-occurrence of appearance and spatio-temporal features. In *International Conference on Pattern Recognition(ICPR)*, 2008.

[8] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1999.

[9] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):389–408, 1958.

[10] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[11] Yuri Pekelny and Craig Gotsman. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum*, 27(2):399–408, 2008.

[12] Andrew D. Wilson and Hrvoje Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 273–282, New York, NY, USA, 2010. ACM.

[13] K. Shimada, S. Horiguchi, and T. Endo. An effective speech understanding method with a multiple speech recognizer based on output selection using edit distance. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation (PACLIC22)*, pages 350–357, 2008.