Hand Gesture Recognition for Robot Control Using the Leap Motion Controller

Kyuhei Honda^{a,*}, Masahiko Yano^a, Yumi Hasegawa^b, Shosaku Tanaka^c

^aNational Institute of Technology, Oita College, 1666 Maki, Oita 870-0152, Japan

^bKindai University, 930 Nishi-mitani, Kinokawa, Wakayama 649-6493, Japan

^cRitsumeikan University, 56-1 Toji-in Kitamachi, Kita-ku, Kyoto 603-8577, Japan

*Corresponding Author: honda@oita-ct.ac.jp

Abstract

In recent years, robots are indispensable in our daily lives. Building a user-friendly system to control robots is necessary for people, especially non-expert users. A gesture recognition system can be useful to build a user interface which is intuitive and natural for people. An input method using hand gestures is effective because it can offer many kinds of instructions with the change of the number and posture of fingers. In conventional systems, the user needs to wear a special sensor such as a data glove in the user's hand. It is a burden for the user to put on such sensors at each time for controlling a robot. In this research, in order to save time and effort to attach sensors to the hand, gesture recognition is performed by using the Leap Motion sensor that is capable of non-contact measurement. Japanese finger characters are used for gestures to be recognized in this research. A finger character is a type of visual language that associates letters with hand shapes and motions. Using the Leap Motion, we have built a recognition system using 41 static characters without movement among 46 finger characters corresponding to Japanese alphabet letters. The recognition is based on SVM (support vector machine) classification. In the experiments, we have examined the detection time and the recognition ability of finger characters in order to construct a recognition system that is optimized for robot control.

Keywords: Human-machine interface, Gesture recognition, Leap Motion, Support Vector Machine, Robot control.

1. Introduction

In recent years, robots are indispensable in our daily lives. Building a user-friendly system to control robots is necessary, especially for non-expert users. A gesture recognition system can be used as a user interface which is intuitive and natural for people. Speech recognition can be also a solution as another input means, it is however, susceptible to noise in various life sounds. Gesture recognition can be applied in places such as factories or sea where speech recognition can be hardly used. An input method using hand gestures is effective because it can offer many kinds of instructions with the change of the number

and posture of fingers.

In several systems, the hand information is obtained using special sensors. Several master-slave methods⁽¹⁻⁴⁾ utilize the angle values of user's finger joints obtained by a data glove. Ueyama et al.⁽¹⁾ utilize a data glove with multiple linear regression models to measure joint angles of the human hand. Ozawa et al.⁽²⁾ and Tsujiuchi et al.⁽³⁾ utilize a data glove for controlling a robotic hand. These researches have the disadvantages that a data glove itself is costly and the user needs to attach the sensor to the body for each time of robot control, which is a burden for the user.

Several systems⁽⁴⁻⁵⁾ use the visual information obtained by a stereo camera. However, a camera used as a passive sensor cannot perform stable measurement under ill lighting condition. The Leap motion can perform the measurement of the user's hand, since it is robust to ill lighting condition. The Leap Motion consists of two cameras and three infrared LEDs. It can perform active measurement by projecting the light of LEDs onto the hand and detecting the hand through its dark plastic cover used as the infrared wavelength pass filter.

As for the conventional systems using the Leap Motion or other active sensors, Bassily et al.⁽⁶⁾ have developed a human-machine interface between the Leap Motion and a 6-DOF robotic arm. M. Mohandes et al.⁽⁷⁾ have proposed a method to recognize the 28 letters of the Arabic alphabet by using the Leap Motion. Marin et al.⁽⁸⁾ have realized hand gesture recognition using the Leap Motion as well as the Kinect devices. Kuznetsova et al.⁽⁹⁾ have proposed a method to recognize 24 static alphabet signs from American sign language based on the data obtained by a single depth camera. Dong et al.⁽¹⁰⁾ have used the Kinect devices for recognizing 24 static alphabet signs.

In this research, we use only the Leap Motion as a non-contact sensor to reduce the physical burden on the user. The sensor is specialized for precisely detecting hands or fingers of humans and is more cost-effective. It has the advantages that it is portable, and easy for a sensor set-up because of its sensor size. We use Japanese finger characters or Japanese alphabet signs as gestures to be recognized. The number of the letters of Japanese alphabet is 46, which is more than that of the English alphabet or the

Arabic alphabet, so that the number of gestures with Japanese alphabet signs becomes larger. A finger character is a type of visual language that associates letters with hand shapes and motions, which provides a basis for sign language.

Sign language is very effective as a way to communicate with the hearing disabilities. Several decades ago, many Japanese deaf schools recommended lip-reading instead of using sign language. In 2006, the United Nations adopted the Convention on the Rights of Persons with Disabilities, in which sign languages are mentioned as a linguistic identity of the disabilities. In Japan, the Basic Law for Persons with Disabilities was amended in 2011 where sign language is described as one of the languages. Since then, sign language ordinances aiming at spreading sign language have been enacted in various parts of Japan one after another. Recently more Japanese deaf schools have recommended the students to use sign language for communication. At many universities, a number of students with normal hearing learn sign language as a second language as similar to a foreign language. Non-Japanese people also need to learn Japanese sign language in order to communicate with Japanese persons with hearing disabilities. For those people, it might be natural to use sign language for robot control. On the other hand, even for people who are not familiar with sign language, it is not formidable to memorize finger characters since each shape is easily recalled from its derivation. This is the reason why we use Japanese finger characters to be recognized for robot control.

In this research, we have built a recognition system using 41 static characters among 46 finger characters corresponding to Japanese alphabet letters. In the experiments, we have examined the detection time and the recognition ability of finger characters in order to construct a recognition system that is optimized for robot control.

2. System Overview

The procedure of our system is shown in Fig.1. There are two phases using the Leap Motion data for finger character recognition, which are a training phase and a recognition phase. In advance, we train classifiers for finger character recognition using an SVM framework with learning data obtained from the Leap Motion.

Fig. 1. Overall procedure for recognition

Fig. 2. Japanese finger characters (Shapes without movement)

Fig. 3. Leap Motion coordinate system and hand features

The illustration of Japanese finger characters to be recognized in our system is shown in Fig. 2. These characters are static or without movement. The illustration shows finger shapes seen from the listener's side. The first five from the top left of the figure are finger characters of Japanese vowels {"a", "i", "u", "e", "o"}. The others are Japanese alphabets that can be formed by combining vowels and 9 basic consonants {'k', 's', 't', 'n', 'h', 'm', 'y', 'r', 'w'}.

2.1 Hand detection of the Leap Motion

Our recognition system first detects the hand by using the Leap Motion. Then, the feature values are calculated based on shape data of the detected hand. Finally, the hand is identified as its corresponding finger character by using SVM classifiers together with calculated feature values.

The Leap Motion sensor is relatively inexpensive and easily purchasable. It is a non-contact type input device specialized for detection of hands or fingers. The Leap Motion consists of two cameras and three infrared LEDs. It can capture images which contain the hand of the user. The hand detection function is realized by sensor hardware and software. The detection range is within about 60 cm cubic area above the Leap Motion sensor. The sensor coordinate system is shown in Fig. 3.

The detection time with the Leap Motion is sometimes longer than what we expect. This is because an improper measurement occurs due to the self-occlusion of the hand or arm of the user depending on the position relationship between the Leap Motion sensor and the hand.

2.2 Calculation of features

As the hand is detected, the system can obtain the following data: hand orientation $\mathbf{h}$, hand normal $\mathbf{n}$, hand center $\mathbf{C}$, and each fingertip coordinate value $\mathbf{F}_i$ where a variable i ($=[1, \dots, 5]$) corresponds to each finger type. The obtained values from the Leap Motion are indicated in Fig.3. Hand orientation $\mathbf{h}$ is a vector from the wrist or the radiocarpal joint to the metacarpophalangeal joint of middle finger. Hand normal $\mathbf{n}$ is a normal vector of the palm.

The features for SVM classification are calculated based on the obtained data. In order to deal with the hand size of various users, the scaling of the features is done by using a scale value S . The scale value is the distance between the hand center position and the fingertip of the middle finger, which is calculated by the following equation.

$$S = |\mathbf{F}_3 - \mathbf{C}| \quad (1)$$

The features calculated in our systems are angle values, distance values, and elevation values regarding each fingertip. These values are basically obtained by transforming fingertip position values from the Cartesian coordinate system to the cylindrical coordinate system where each origin of both coordinate systems is the hand center. These are expressed in intuitive and easy-to-understand expression from the viewpoint of humans. The calculation equations of these three kinds of values are based on the research of Marin et al.⁽⁸⁾. The calculation of these features is shown in the following equations.

(Ft 1) Fingertip angle A_i is an angle of each fingertip with respect to the hand orientation vector $\mathbf{h}$. This feature is calculated in the following equation.

$$A_i = \angle(\mathbf{F}_i^\pi - \mathbf{C}, \mathbf{h}) \quad (2)$$

where $\mathbf{F}_i^\pi$ is the value obtained by projecting $\mathbf{F}_i$ onto a palm plane, which is defined by the normal vector $\mathbf{n}$.

(Ft 2) Fingertip distance D_i is a distance of each fingertip with respect to the hand center $\mathbf{C}$.

$$D_i = \frac{|\mathbf{F}_i - \mathbf{C}|}{S} \quad (3)$$

where S is a scale value defined above.

(Ft 3) Fingertip elevation E_i is a distance of each fingertip with respect to a palm plane, which is defined by the normal vector $\mathbf{n}$.

$$E_i = \frac{\text{sgn}((\mathbf{F}_i - \mathbf{F}_i^\pi) \cdot \mathbf{n})|\mathbf{F}_i - \mathbf{F}_i^\pi|}{S} \quad (4)$$

Fig. 4. Features A (Angle), D (Distance), and E (Elevation)

where $\text{sgn}((\mathbf{F}_i - \mathbf{F}_i^\pi) \cdot \mathbf{n})$ indicates which side of a palm plane each fingertip belongs to.

The features calculated above are expressed in a 15-dimensional vector as follows:

$$\mathbf{Ft}_{ADE} = (A_1, \dots, A_5, D_1, \dots, D_5, E_1, \dots, E_5) \quad (5)$$

We also use two kinds of features of the hand for finger character recognition.

(Ft 4) Hand orientation $\mathbf{h}$ is obtained directly from the Leap Motion sensor.

(Ft 5) Hand normal $\mathbf{n}$ is obtained directly from the Leap Motion sensor.

The features obtained directly from the Leap Motion are expressed in a 6-dimensional vector as follows:

$$\mathbf{Ft}_{hn} = (h_x, h_y, h_z, n_x, n_y, n_z) \quad (6)$$

The features for SVM classification are expressed in a 21-dimensional vector in total using equations (5) and (6).

The illustration in Fig. 4 shows an example of the features of the little finger, which are denoted by using equations (2)-(4) except for scaling.

2.3 SVM classification

The support vector machine (SVM) is a learning algorithm to solve a binary classification problem. It has several advantages that the number of its parameters to be optimized is small and its accuracy rate is good even when the dimension of the features is large. Given training data $\mathbf{x}$, the SVM linear discriminant function or discriminant hyperplane is given by

$$\mathbf{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad (7)$$

where $\mathbf{w}$ is a weight vector, and b is a bias value for the SVM classifier. We use RBF kernel with SVM. There are parameters C and γ to determine the SVM recognition ability. The parameter C decides how much the classifier should allow a few errors of recognition to avoid the overfitting problem when the data are not linearly separable. The parameter γ decides the complexity of the discriminant hyperplane.

2.4 Similarity of finger characters

The ability of SVM classifiers depends on how each data is expressed in the feature vector for training and recognition. The scattering condition of target classes is visualized by using multidimensional scaling(MDS). In MDS, the data are expressed in the reduced dimension while it keeps the distance values between the target classes. In the MDS chart, the classes similar to each other are located closely while the classes dissimilar to each other are located far away from each other.

The similarity between class i and class j in the p -dimensional feature vector is also expressed with the sample correlation coefficient r_{ij} .

$$r_{ij} = \frac{\sum_{k=1}^p (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^p (x_{ki} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^p (x_{kj} - \bar{x}_j)^2}} \quad (8)$$

2.5 Robot control based on finger character recognition

After the finger character recognition is performed, a robot is controlled based on the recognition result. Each finger character corresponds to each motion of the robot. If the robot has only a few kinds of motions, we can reduce the number of finger characters to be recognized by choosing them from 41 static Japanese finger characters. There are several choices for the same motions. In our system, the choosing criteria contain the detection time td_i of each finger character as described in section 2.1, and the sample correlation coefficient r_{ij} as described in the previous section. The evaluation value for a command set for N kinds of robot motions is expressed in the following equation.

$$J = W_1 J_1 + W_2 J_2 \quad (9)$$

$$J_1 = \sum_{i=1}^N td_i \quad (10)$$

$$J_2 = \sum_{i=1}^N \sum_{j=i+1}^N r_{ij} \quad (11)$$

where W_1 and W_2 are weight values for the criteria.

3. Experiments

3.1 Data acquisition

In the experiments, the user first registers a scale S which is denoted as in equation (1) after stretching all fingers above the Leap Motion sensor and making a shape which looks similar to Japanese character “te” in Fig. 2. At this time, the software shows the detected hand shape on the PC screen as in Fig. 5a. This shape is a shape seen from a view of the user. The images captured by the Leap Motion are also shown on the PC screen as in Fig. 5b. With these displays, the user confirms whether the Leap Motion can detect the hand correctly or not.

The user then records the feature data of a target finger character after forming its shape above the Leap Motion sensor and confirming that the Leap Motion detects the hand correctly. Thus, the system obtains each finger character data through the two steps:

Step 1. Register a scale S

Step 2. Record target finger character data

The detection time td_i is measured as an elapsed time from the end of Step 1 to the end of Step 2. The time measurement result for one subject is shown in Fig.6. This graph shows the average time of 15 data acquisition for each finger character detection. This detection time shows the ease of the detection for each finger character and it depends on how familiar the user is to the system. For example, the graph shows the user can easily input data of finger character “ne” to the system. But it takes the longest for the user to input data of finger character “nu” to the system.

(a) Detected shape

(b) Captured image

Fig. 5. Displayed images

Fig. 6. Detection time of each finger character with the Leap Motion

3.2 Result of feature calculation

To verify that the features are calculated as expected, the calculated features regarding three vowels {"a", "i", "u"} with their shapes are shown as examples in Fig.7-9, respectively. The 15-measurement data for each character are shown at a time at each graph. The horizontal axis of each graph shows the type of finger. The angle values are shown in degrees. The distance values and elevation values do not depend on a specific unit system since they are calculated as ratio values to the scale value S .

Fig. 7. Calculated features for "a"

Fig. 8. Calculated features for "i"

Fig. 9. Calculated features for "u"

In Fig. 7, we can observe that the distance from thumb tip from the hand center is the longest and the elevation values except thumb are stably higher. In Fig. 8, the distance of little finger tip from the hand center is the longest and the elevation values except little finger are higher. In Fig. 9, the distance value of middle finger tip or index finger tip from the hand center is longer and the elevation values except middle finger and index finger are higher.

3.3 Result of SVM classification

The SVM classifiers are built by using 13 data out of 15 data per each finger character for classifier training. The other 2 data are used for the evaluation of the classifiers. The recognition rate is 100% using the proper SVM parameters, for example, $\gamma=1$ and $C=20$. This accuracy is achieved because the data are obtained from the only one subject and the variance of the feature values for the same character is not large.

The recognition accuracy varies depending on the SVM parameters. For example, with the different parameters $\gamma=1$ and $C=10$, the recognition error occurs and the data of the finger character "u" is erroneously recognized as the character "ra".

The calculated features regarding finger character "ra" are shown in Fig. 10. In Fig.9 and Fig.10, they look similar. In Fig. 11, the detailed angle values of index finger and middle finger for finger characters "u" and "ra" are shown together with the segment lines which shows the relation between index finger and middle finger for each data. We can observe the difference in the slope of line segments among two finger characters "u" and "ra". With this difference, the system can recognize "u" and "ra".

Fig. 10. Calculated features for "ra"

Fig. 11. Comparison of angle features for "u" and "ra"

Fig. 12. MDS chart of finger characters

3.4 Similarity calculation

We calculate the similarity among Japanese finger characters by using equation (8). The similarity value between “u” and “ra” is 0.99 which is close to 1. This means that their shapes measured by using the calculated features are almost the same and they are strongly related. In order to visualize the similarity, the MDS chart is shown in Fig. 12. With this chart, we can observe the similarity among the finger characters and infer which finger characters we should use for a command set for robot control.

3.5 Choosing a command set for robot control

Based on the above experimental results, we choose a command set for robot control by using equations (8)-(11).

Assuming that we use a robot that moves in a straight line, its motions are provided as follows: {Go forward, Go back, Stop}. When we allocate a finger character to each movement, there are several candidates. One command set Cmd_1 contains {“hu”, “ha”, “su”}. Each finger character of this command set nearly corresponds to the initial sound of each Japanese pronunciation of {Forward, Back, Stop}. Another command set Cmd_2 contains {“ma”, “u”, “to”}. Each finger character of this command set corresponds to the initial letter of Japanese word set {“ma-e”, “u-si-ro”, “to-ma-re”} which is translated from English word set {Forward, Back, Stop}.

The time cost J_1 and similarity cost J_2 of Cmd_1 are 15.5 and 1.64, respectively. The time cost J_1 and similarity cost J_2 of Cmd_2 are 15.0 and 0.75, respectively. The time costs of both command sets are almost the same. The result of these similarity costs indicates that it is better to choose the command set Cmd_2 to reduce the possibility of false recognition occurrence.

We can also choose finger characters to be recognized in the system for command sets used for other robots with different motions, based on the cost criteria in the way mentioned above.

4. Conclusions

We present a recognition system aiming at controlling a robot with the Leap Motion sensor. The recognition is done with SVM classifiers based on the data obtained from the Leap Motion. We have conducted the experiments to measure the detection time and the recognition ability of finger characters in order to choose a command set for robot motions.

In the future, we are planning to survey the SVM recognition rate by using data of more subjects. We are also planning to realize an actual control of a robot by using chosen finger characters. We also need to incorporate the ease to remember as the criteria for choosing finger characters, which will be surveyed with the questionnaire of the subjects.

References

- (1) Y. Ueyama and K.Hoshino, “Application of Multiple Linear Regression Models to Robotic Hand Control with Data Glove,” IEICE PRMU, Vol.104, pp.19-24, 2004.
- (2) R. Ozawa and N. Ueda, “Supervisory control of a multi-fingered robotic hand system with data glove,” in Proc. 2007 IEEE Int. Conf. IROS '07, pp.1606-1611, 2007.
- (3) N. Tsujiuchi et al., “Development of Pneumatic Master-Slave Hand and Joint Control,” Transactions of the Japan Society of Mechanical Engineers, Series C, Vol.74, No.741, pp.1267-1272, 2008.
- (4) K. Ikeuchi and S.B. Kang, “Toward automatic robot instruction from perception-temporal segmentation of tasks from human hand motion,” IEEE Trans. on Robotics and Automation, pp670-681, 1995.
- (5) K. Ogawara, S. Iba, T. Tanuki, H. Kimura, and K. Ikeuchi, “Recognition and Performance of Human Task with 9-eye Stereo Vision and Data Gloves,” Computer Vision and Image Media, pp. 57-64, 2000.
- (6) D. Bassily et al., “Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller,” in Proc. of 41st Int. Symp. on Robotics, pp.78-84, 2014.
- (7) M. Mohandes, S. Aliyu and M. Deriche, “Arabic sign language recognition using the leap motion controller,” Industrial Electronics (ISIE), 2014 IEEE 23rd Int.Sympo. on, pp.960-965, 2014.
- (8) G. Marin, F. Dominio, P. Zanuttigh, “Hand gesture recognition with Leap Motion and Kinect devices,” IEEE Int. Conf. on Image Processing, pp.1565-1569, 2014.
- (9) A. Kuznetsova, L. Leal-Taixe, B. Rosenhahn, “Real-time sign language recognition using a consumer depth camera,” in Proc. IEEE Int. Conf. on Computer Vision Workshops, pp.83-90, 2013.
- (10) C. Dong, M. C. Leu, Z. Yin, “American Sign Language Alphabet Recognition Using Microsoft Kinect,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, pp.44-52, 2015.