Real-Time Hand Gesture Telerobotic System Using the Fuzzy C-Means Clustering Algorithm

JUAN WACHS, URI KARTOUN, YAEL EDAN, HELMAN STERN
Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
P.O.Box 653, Be'er-Sheeva 84105, ISRAEL
Fax: +972-8-6472958; Tel: +972-8-6461434
E-mail: (juan, kartoun, yael, helman)@bgumail.bgu.ac.il

Abstract

This paper describes a teleoperation system in which an articulated robot performs a block pushing task based on hand gesture commands sent thorugh the Internet. A Fuzzy C-Means clustering method is used to classify hand postures as "gestures". The fuzzy recognition system was tested using 20 trials each of a 12 gesture vocabulary. Results revealed a recognition rate (the ratio of unclassified gestures to classified gestures) of 0.96, and a recognition accuracy (the percent of the classified gestures recognized correctly) of 100%. No gestures were recognized incorrectly. Performance times to carry out the pushing task showed rapid learning, reaching standard times within 4 to 6 trials by an inexperienced operator.

1. Objectives

Develop a hand gesture control system for controlling a telerobot.

2. Methods

2.1 System Architecture

To control a robot movement, the user evokes a gesture from a gesture vocabulary. The user lays his/her hand over a video imager, and a raw image is acquired. An interface screen allows the user to view the captured gesture. The gesture is classified using a recognition module based on the FCM algorithm [5] and is sent to the robot for execution. The components of the system consist of a five degree of freedom articulated robot, a PC system with a frame grabber, two USB cameras, and a video imager.

A set of recognized gestures is sent through the TCP/IP communication protocol to a distant robot PC server (Fig. 1). The server is connected to the robot controller and two USB cameras continually capture the robot scene. Both side and front views of the robot scene are sent to the client using the FTP protocol, and then presented in a user interface.

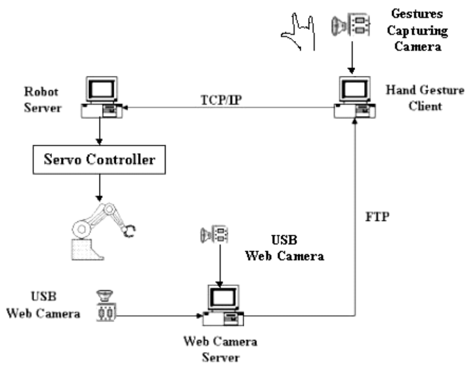


Fig 1. User-Robot Communication Architecture

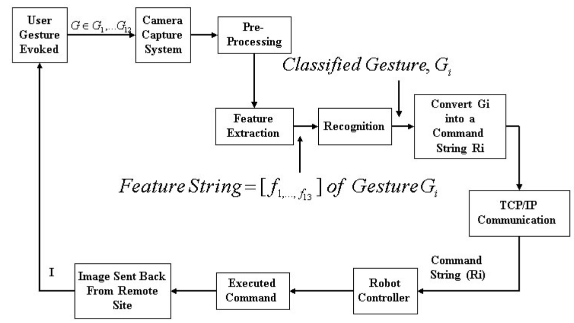


Fig 2. System Flow Diagram

The hand gesture recognition system flow diagram is shown in Fig. 2. Upon presentation of the robotic scene in the user's interface, a gesture G is evoked and selected from the gesture vocabulary { G_1 ,G2, ….., G_{12}}. A vision system converts the captured image of the gesture into a feature string which is subsequently recognized and sent to the robot PC server. After the robot executes the command, camera views of the robot environment are transmitted back to the interface.

2.2 Gesture Classification

2.2.1 Hand Gesture Language

A vocabulary of 12 static gesture poses was designed for robot control tasks (Fig. 3). The *forward* and *back* hand gestures control the X-axis, the *right* and *left* hand gestures control the Y-axis, and the *up* and *down* hand gestures control the Z-axis of the articulated robot arm. The *Roll Right* and *Roll Left* hand gestures rotate the wrist joint, and the *Open Grip and Close Grip* gestures control the robot gripper. The *Stop* hand gesture stops any action the robot performs. The *Home* hand gesture resets and calibrates all robot joints in the home position.

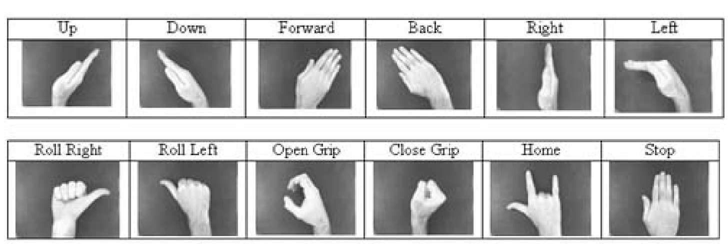


Fig 3. Visual Gesture Recognition Language

2.2.2 Fuzzy C-Means Clustering

We use a Fuzzy C-Means Clustering algorithm (FCM) [5]. Given a training set of n candidate gestures, $X = x_1, \ldots, x_k, \ldots, x_n$,and an objective function:

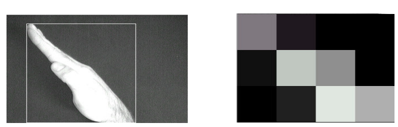
$$J(U,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^{m} d^2(x_k, v_i) \quad (1)$$

where x_k is the feature vector of the kth exemplar gesture in the training set , v_i is a prototype feature vector for cluster i, u_{ik} is the membership value of the kth feature vector to cluster i , m is a weighting exponent, c is the number of gestures as well as the number of clusters. The function $J(U,V)$ is minimized via an iterative process in which the degrees of membership u_{ik} and the cluster center v_i are updated: $v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}}$ (2) and $u_{ik} = \frac{1}{1 + \sum_{j=1}^{c} (\frac{d_{ik}}{d_{ij}})^{\frac{2}{m-1}}}$ (3) where, the u_{ik} satisfies:

$$u_{ik} \in [0,1], \sum_{i=1}^{c} u_{ik} = 1 \; \forall \; k, \quad 0 < \sum_{k=1}^{n} u_{ik} < n \; \forall \; i \quad (4).$$

2.2.3 Preprocessing and Feature Extraction

Preprocessing the captured gesture includes a number of image processing operations such as thresholding and morphological operations. This is followed by constructing a bounding box around the segmented hand. A feature vector of the image with 13 parameters is created. The first feature is the aspect ratio of the bounding box. The last 12 are block mean grayscale values calculated from a 3 by 4 partition of the image. Fig. 4 illustrates a typical user gesture (a), its block mean grayscale values (b), and the resultant feature vector, respectively.



(a) (b)

Feature Vector = [57 176 52 2 2 68 249 171 16 3 13 253 188]

Fig 4. Illustration of a Feature Vector

2.2.4 Training Stage

The training stage involves running the Fuzzy C-Means algorithm for a set of exemplar hand gestures. At least 25 samples for each of the 12 hand gestures in the language are taken to construct the training set. Every image gets a feature vector as described in section 2.3.3.

2.2.5 Classification

User gestures are classified using the highest membership value. In our case, if $X_{k'}$ is the feature vector of the current hand gesture image , its distance to each of the cluster centers v_i is determined and used in (2) to calculate the membership values $\{u_{ik'} \ \forall \ i=1,...,c\}$. The gesture is classified by finding: $u_{i'k'} = \text{Max} \ \{u_{ik'} \ ; \ \forall \ i=1,...,c\}$. A further test is made before recognition is established. This test depends on a recognition threshold , τ. If $u_{i'k'} >= \tau$ is true, then the gesture is recognized as belonging to classification i', otherwise it is said to be unclassified as all the membership values are too low. A recognition threshold of $\tau = 0.75$ was empirically determined to provide the best performance.

3. Results

3.1 Testing the Recognition System

An experiment was designed to test the recognition system. Twenty trials of each gesture were presented to the gesture recognition system. Two measures of performance, based on a distinction between classification and recognition, were used in the evaluation. A gesture k' whose entire set of membership values, $\{u_{ik'} \ ; \ \forall \ i=1,\ldots,c\}$, is below the given threshold value of τ is said to be **unclassified**. A gesture k', with least one membership value above the threshold value τ, is said to be **recognized** as belonging to classification i' , if $u_{i'k'} = \text{Max} \ \{u_{ik'} \ ; \ \forall \ i=1,\ldots,c\}$. We can now define the two performance measures as:

(a) **Recognition Rate -** The ratio of unclassified gestures to classified gestures.

(b) **Recognition Accuracy -** The percent of the classified gestures recognized correctly.

Results indicate a recognition rate of 0.96, and a recognition accuracy of 100%. Unclassified gestures could be attributed to the recognition system or lack of training of the user to sufficiently provide the correct hand configuration for the intended gesture. All other classified gestures were recognized correctly. Gestures 5 (Right) and 6 (Left) were the most common unclassified gestures, while gestures 4 (Back), 9 (Open Grip) and 11 (Home) were rarely unclassified.

3.2 Case Study

An experiment was performed in which an operator using hand gestures controls the remote robot to push a yellow wooden cube, located on a top of a pile into a container. (Fig. 5)

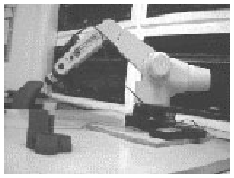


Fig 5. A255 Robot, Plastic Cup Structure, and Yellow Wooden Box

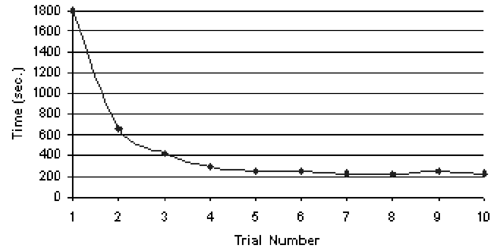


Figure 6. Learning Curve of the Hand Gesture System

An inexperienced operator performed ten identical experiments and the performance times for each were recorded. The system is fast, accurate, and due to a simple hand gesture language developed, the learning curve of task completion time was reduced quickly. As can be seen in Fig. 6 standard times were reached after four to six trials.

4. Conclusions

This project described the design, implementation and testing of a telerobotic gesture-based user interface system using visual recognition. Experimental results showed that the system satisfies the requirements for a robust and user friendly input device. The case study demonstrated the importance of latency for telerobotic systems. Although gestures were recognized quickly and sent in packet forms, successful execution of the commands could not be verified until the image of the robot environment was received at the user interface. This resulted in an *overlapping effect* - sending of new gestures before complete information of the present robot position was received. Future research will be directed to the solve this problem.

References

[1] Katkere A., Hunter E., Kuramura D., Schlenzig J., Moezzi S. and Jain R. "ROBOGEST: Telepresence Using Hand Gestures". Technical report VCL-94-104, Visual Computing Laboratory, University of California, San Diego. 1994.

[2] Kortenkamp, D., Huber, E., and Bonasso, R. P. "Recognizing and Interpreting Gestures on a Mobile Robot". In AAAI96, 1996.

[3] Pavlovic V, Sharma R., and Huang T. "Visual Interpretation of Hand Gestures for Human Computer Interaction: A Review". IEEE PAMI, Vol. 19. pp. 677-695. 1997.

[4] Triesch J. and Malsburg C. V. D. "A Gesture Interface for Human-Robot Interaction". Proc. of 3th IEEE Intl. Conf. on Automatic Face and Gesture Recognition, pp. 546-551. 1998.

[5] Bezdek J.C. "Cluster Validity with Fuzzy Sets". Cybernetics. Vol. 3, No. 3, 58-73. 1973.