PARAMETER SEARCH FOR AN IMAGE PROCESSING FUZZY C-MEANS HAND GESTURE RECOGNITION SYSTEM

Juan Wachs, Helman Stern, Yael Edan

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er-Sheva, Israel, 84105, {juan, helman, yael}@bgumail.bgu.ac.il

ABSTRACT

This work describes a hand gesture recognition system using an optimized Image Processing-Fuzzy C-Means (FCM) algorithm. The parameters of the image processing and clustering algorithm were simultaneously found using a neighborhood parameter search routine, resulting in solutions within 1-2% of optimal. Comparison of user dependent and user independent systems, when tested with their own trainers, resulted in recognition accuracies of 98.9% and 98.2%, respectively. For experienced users, the opposite was true, testing recognition accuracies where better for user independent than user dependent systems (98.2% over 96.0%). These results are statistically significant at the .007 level.

1. INTRODUCTION

Hand gestures are a common method for telerobotic control [1]. This type of communication provides an expressive, natural and intuitive way for humans to control robotic systems. One benefit [2] of such a system is that it is a natural way to send geometrical information to the robot such as: left, right, etc. Gestures may represent a single command, a sequence of commands, a single word, or a phrase, and may be static or dynamic. Correct classification in reasonable times must be obtained for practical use [3].

Human-robot interaction using hand gestures provides a formidable challenge. This is because the environment contains a complex background, dynamic lighting conditions, a deformable hand shape, and a real-time execution requirement. There has recently been a growing interest in gesture recognition systems with a number of researchers providing some novel approaches, many of which are quite elaborate and require intensive computer resources. For example, Quek [4] develops a flow field computational algorithm. Koons, et. al. [5] describes an approach in which hand data is classified into features of posture, orientation and motion. In [6] color, motion, and tracking is used in a hand posture system for robot control. Classification is based on elastic graph matching. An excellent review of gesture modeling approaches is that of Huang, et.al. [7]. Our system is closest to that of the edge based technique to extract image parameters from simple silhouettes [8].

In this paper we define 13 static gesture postures (Fig. 1) for telerobotic control using a Supervised Fuzzy C-Means (FCM) recognition system.

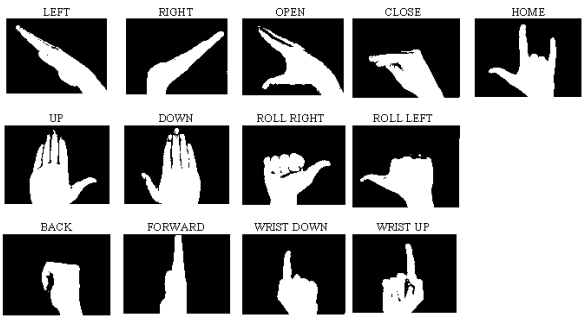

Fig. 1. Hand Gesture Language

The FCM clustering algorithm [9] is a popular algorithm for image recognition tasks [10-13]. In this work image features are classified using a supervised FCM algorithm. A locally optimal set of system parameters is determined by a neighborhood search algorithm. This increases the systems recognition rate over traditional empirical methods using trial and error.

Although the speed of artificial neural network (ANN) classifiers allows real-time operation and comparable accuracy, a FCM is used because it requires smaller training sets and shorter training times. Moreover, it is compatible with future research needs to compare other systems based on cluster variances. ANN's preclude such comparisons, as cluster boundaries are transparent, except for very small networks. The paper describes the gesture recognition algorithm with a focus on the parameter search procedure and its evaluation for user dependent and independent systems.

2. PREPROCESSING AND FEATURE EXTRACTION

Preprocessing of the image starts with segmentation of the hand from the background using a threshold value to obtain a black and white image. after which a bounding box is constructed around the segmented hand and divided by a block partition. A feature vector of the image is comprised of the aspect ratio of the bounding box and the average intensity of each block (ratio of white to black pixels). Fig. 2 illustrates a typical user gesture, its 3 X 4 block intensities, and the resultant feature vector.

A weight coefficient is defined for each feature i as $w_i \quad i = 1,...,N$ (where N is the length of the feature vector). The weights are combined into a weight vector w. For all $i>1$, $w_i=1$. w_1 is the weight parameter for the first feature, which is the aspect ratio of the gesture. The first feature is considered more important than the others for its discriminatory power.

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

Fig. 2. Illustration of a Feature Vector

3.SUPERVISING THE FUZZY C-MEANS CLUSTERING ALGORITHM

The FCM Clustering algorithm is provided with a training set of gestures, each represented by a feature vector. The feature vectors are clustered for subsequent use in a recognition system. Once the clusters have been created, they are labeled using a gesture name. This in effect allows the FCM clustering algorithm to be "supervised". A valid cluster labeling must satisfy two conditions; (i) each cluster receives a unique label (a gesture name), and (ii) each label must be assigned to at least one cluster. Using the following notation A clustering labeling algorithm (CLA) is described for constructing a valid labeling. Let m, n number of clusters and labels ($m \geq n$); n_{ij} = the number of type i gestures in cluster j; $[j] = i$ represents the assignment of label i to cluster j; M, N sets of all clusters and labels; S, S' sets of labeled and unlabeled clusters; L, L' sets of assigned and unassigned labels; S^* = set of permanently single labeled clusters. Assume clusters are non-empty.

Step 1.Initialization. Set $S' = M$, $L = N$, $S^* = \phi$.

Step 2.Maximum Populated Cluster. For each unassigned gesture label $i \in L'$, find a cluster j' such that;

$max\{n_{ij} \mid j \in S'\} = n_{ij'}$. If $n_{ij'} = 0 \ \forall \ j \in S'$, break such ties by selecting $j \in S'$ randomly. Let $[j']=i$. After all gesture labels have been assigned, transfer all newly labeled clusters from S' to S and labels from L' to L. Place all single labeled clusters to S^*. If there don't exist multi-labeled clusters, go to 4.

Step 3.Resolve Multi-labeled Clusters. For all j in S, retain only label i' such that; $max\{n_{ij} \mid all\ i\} = n_{i'j}$. Let $[j]=i'$. Remove the rest of the labels and place them in L'. Remove all new single labeled clusters from S and make them permanent labeled by placing them in S^*. Return to 2.

Step 4.Label Remaining Unlabeled Clusters. For all $j \in S'$, let $max\{n_{ij} \mid i \in N\} = n_{i'j}$. Set $[j] = i'$. (Here, several clusters can share identical labels). Move all j from S' to S^*.

4. RECOGNITION PERFORMANCE AND PARAMETER SEARCH

Gestures performed by a user are recognized 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 to calculate the membership values $\{u_{ik'} \ \forall \ i=1,...,c\}$. Where u_{ik} = membership value of feature k for cluster c_i, and c = number of clusters. The gesture is recognized by finding: $u_{i'k'} = Max\{u_{ik'} ; \forall \ i=1,...,c\}$. System performance is evaluated using a confusion matrix that contains information about actual and classified gestures.

The recognition accuracy in percent is calculated as:

$$A = \frac{(total\ number\ of\ gestures - number\ of\ gestures\ misclassified)}{total\ number\ of\ gestures} 100 \qquad (1)$$

The process of searching optimal parameters for the supervised FCM is described with a flow chart shown in Fig. 3, The output of the process is a near optimal set of parameters p^* achieved by maximizing the recognition accuracy A.

The vector p is the set of input parameters (Table 1). Two types of input parameters are used: image processing features (size of block partition and the aspect ratio weight) and FCM parameters (number of clusters).

Table 1. Parameter Definition

j	Meaning	Values
1	Number of Colums for image partition	*p1=2,3,..,8*
2	Number of Rows for image partition	*p2=2,3,..,8*
3	Weight of the aspect ratio	*p3=1,1.5,2,..,4*
4	Clusters	*p4=13,14,..,22*

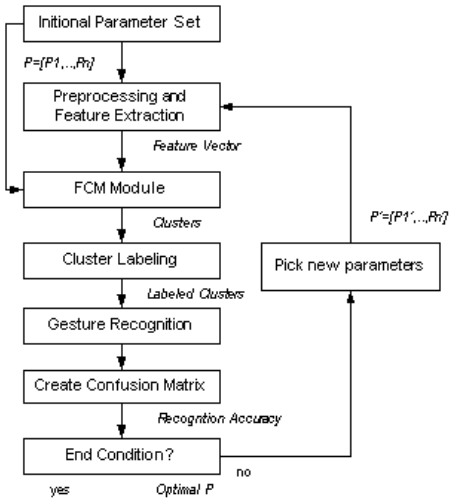

Fig. 3. Supervised FCM algorithm with parameter search

Let p be the vector of parameters, and A the recognition accuracy. For any feasible solution $p=[p_1, ..., p_n]$ for the recognition system, define a set $N(p)$ of neighboring solutions of vector p. The number of neighbors of p is $2n$ as each parameter is incremented up and down. This neighborhood search method starts with an arbitrary initial solution. A pseudo code of the algorithm is shown below:

Algorithm neighborhood search;
Begin
Create an initial feasible solution $p=[p_1, ..., p_n]$
While there is a neighbor $p' \in N(p)$ with $A(p') > A(p)$ **do**

Begin
Replace *p* by *p'*
End
Output *p*, which is the locally optimal solution
End

Define an **iteration** as one cycle starting from an initial solution *p* until the next neighbor solution *p'* is selected. An example sequence of the parameter vectors *p*, appears in Table 2. Recognition accuracy in each iteration is shown in Fig. 4.

Table 2. Optimal Parameter Search

	Parameters			
Iterations	*p1*	*p2*	*p3*	*p4*
1	2	2	3.5	17
2	2	3	3.5	17
3	2	4	3.5	17
4	2	4	3.5	18
5	2	4	3.5	18

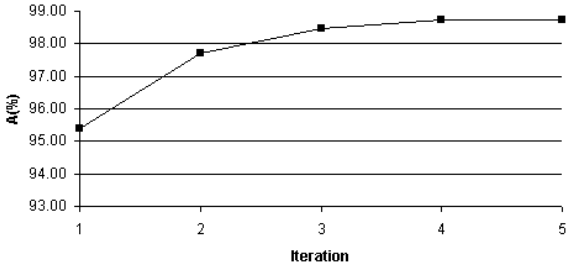

Fig. 4. Recognition Accuracy vs. Iterations

Average complexity of the neighborhood search algorithm is *O(n) (w*here *n* is the size of the parameter vector) times the number of iterations. In the previous example, for the given *p*, the number of neighborhood solutions examined is 2 x 4 x *Ave. no. of iterations* = 8 x 5 = 40 (convergence was fast in the order of 3 to 8 iterations. Complete evaluation requires an evaluation of 2940 (the size of the search space = $7\times7\times6\times10$). It should be noted that the evaluation of each solution requires the determination of a new set of image features, executing the FCM algorithm, cluster label assignments, gesture recognition, and analysis of the confusion matrix.

5. EXPERIMENTS AND RESULTS

Two different types of systems were used to train and test recognition accuracy: user dependent *(D)* and independent *(I)* systems. The *D* and *I* systems are defined as the systems, which are trained by one user and multiple users, respectively.

5.1. K-Fold Cross Validation

A K-Fold Cross Validation was used [14] to partition the original sample set into *k* subsets of equal size. Each subset in turn is used for testing and the remainder for training. The advantage is that all the data can be used for training and none has to be held back in a separate test set. Each partition of the data into a training set and a testing set will be defined as "session". The partition used was *k*=4. The misclassification average is defined as the k-fold cross-validation estimate of the true error rate. The Prediction Error for the cross-validation is defined as:

$$PE_{cv} = \frac{1}{N}\sum_{1}^{N}(y_i \neq \hat{y}^{-(w_i)}(x_i)) \quad (2)$$

Where, *N* is the total sample size for training and testing.

y_i is the expected response of the system for the sample x_i.

$\hat{y}^{-(w_i)}(x_i)$ is the estimation of the response of $\hat{y}(x_i)$ based on the system, trained with the sample set to which x_i doesn't belong.

w_i is the index group in which the index *i* falls.

if $y_i \neq \hat{y}^{-(w_i)}(x_i)$ is true, then the expression equals 1, otherwise 0. For the hand gesture system case, *N*=520. Four sample sets were used *k*=4 and $1 \leq w_i \leq 4$.

5.2. Training the dependent and independent systems

Seven subjects each trained individual systems to obtain seven different *D* systems. Each *D* system used a set of 30 samples of 13 gestures, for a total of 390 samples per session. Thus, training and testing used 390 and 130 samples, respectively. These subjects also were used again, to train one *I* system with 5 samples for each of the 13 gestures, for a total of 455 samples per session.

5.3. Testing the dependent and independent systems

The gesture recognition system was tested using three types of subjects in the experiments: Owners (*O*), Experienced Users (*E*) and Novice Users (*N*). Owners are original trainers who test their own system. Experienced Users are users that tests systems, which were trained by others. These users were reused owners who play the role of experienced users at this stage. Novice Users are those users that have never used the system. All testing sets are composed of 40 instances of each gesture, a total of 520 (40x13) samples. The number of runs of each system is shown in Table 3.

Table 3. Number of Runs

User Type / System Type	Owners (Per System)	Experienced (Per system)	Novice (Per system)	No. Systems
Dependent	1	6	5	7
Independent	7	7	5	1

Fig. 5 shows the average recognition accuracy of each type of user, for the *D* and the *I* systems.

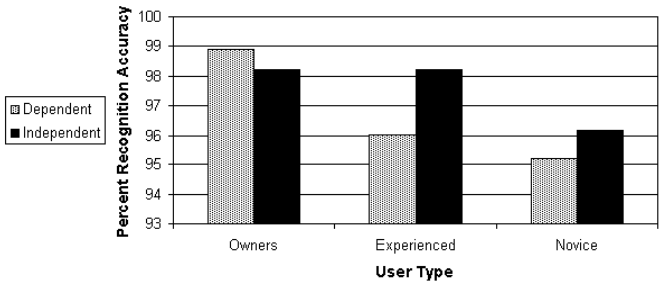

Fig. 5. Average Recognition Accuracy

5.4. Hypotheses

The recognition accuracy of the different systems was compared using the two-tailed t-test. Table 4 shows the hypothesis formulated, the population used to compare each side of the hypothesis, their recognition accuracy, and their variance respectively. The last two columns are: the result of the hypothesis, and the significance level of the hypothesis. The recognition accuracy of system *x* tested with user *y* is represented by *A*(*x, y*). The important results are in the Table 5.

Table 4. Results of performance comparison between systems

No.	Hypothesis	n1	n2	x1 (%)	x2 (%)	S_1^2	S_2^2	Answer	Signif. (%)
1	A(D,E)>A(D,N)	21840	18200	96.01	95.19	0.0383	0.0458	TRUE	0.0032
2	A(I,O)>A(I,N)	3640	2600	98.21	96.19	0.0175	0.0366	TRUE	0
3	A(D,O)>A(D,E)	3640	21840	98.90	96.01	0.0109	0.0383	TRUE	0
4	A(D,O)>A(I,O)	3640	3640	98.90	98.21	0.0109	0.0175	TRUE	0.69
5	A(I,E)>A(D,E)	3640	21840	98.21	96.01	0.0175	0.0383	TRUE	0
6	A(I,N)>A(D,N)	2600	18200	96.19	95.19	0.0366	0.4580	TRUE	1.2

Table 5. System Recognition Accuracy

Type of User	Type of System	
	Dependent (*D*)	Independent (*I*)
Owners (*O*)	**98.9%**	98.2%
Experienced (*E*)	96.0%	**98.2%**

6. CONCLUSIONS

When the systems were tested using their own trainers the *D* system was better than that of the *I* system, with mean recognition accuracies of 98.9% over 98.2%. This is as expected since any learning system should have better performance when tested with its trainer. For experienced users, the opposite was true, testing recognition accuracies where better for *I* than *D* (98.2% over 96.0%). This also is expected as experienced uses were testing systems trained by others. Here, the *I* system was trained with a wide variation of hand gestures samples, and as a result it had better generalization properties. These results are statistically significant at the .007 level. As for experienced users, novice testing accuracy was also better for *I* than *D* systems obtaining 96.1% and 95.1%, respectively. This result is only statistically significant at the .01 level, but is not important as from previous learning rates [10] it is expected that the novice users after several trials should reach the 98-99 % level.

Analysis of the confusion matrices can help improve the feature selection for gesture recognition. Most misclassifications were due to inaccurate “back” and “roll right” gestures. In these gestures either the block partition intensity values lacked discriminatory information or the users provided inaccurate gestures. The near optimal parameter search procedure allows an easy extension of the system concerning both the number of parameters and the complexity of the algorithm, while keeping the robustness and the reliability of the system at the same high level. Therefore, the approach suggested in this paper appears very promising. The system was integrated into a telerobotic control environment and a demonstration application was developed. The demonstration system operates in a simple pick and place scenario [10].

7. ACKNOWLEDGEMENTS

This project was supported by the Ministry of Defense MAFAT Grant No. 2647, and partially supported by the Paul Ivanier Center for Robotics Research & Production Management, Ben-Gurion University of the Negev.

8. REFERENCES

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

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

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

[4] F. K. H. Quek. “Unencumbered Gestural Interaction”, IEEE Multimedia, pp. 36-47, 1996.

[5] D. B. Koons, “Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures”, Intelligent Multimedia Interfaces, M. T. Maybury (Ed), MIT Press, pp. 257-276, 1993.

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

[7] T. S. Huang, and V. I. Pavlovic. “Hand Gesture Modeling, Analysis,and Synthesis”. Proc, of Intl. Conf. on Automatic Face and Gesture Recognition, 1995.

[8] J. Segen, “Controlling Computers with Gloveless Gestures”, Proc.of Virtual Reality Systems, 1993.

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

[10] J. Wachs, H. Stern, Y. Edan, U. Kartoun, “Real-Time Hand Gesture Using the Fuzzy-C Means Algorithm”, In Proc. of WAC 2002, Florida, June 2002.

[11] J. Eisenstein, S. Ghandeharizadeh, L. Huang, C. Shahabi, G. Shanbhag, and R. Zimmerman. “Analysis of Clustering Techniques to Detect Hand Signs”. Proc. of the International Symposium on Intelligent Multimedia, Video and Speech Processing, 2001.

[12] D. Cosic and S. Loncaric. “New Methods for Cluster Selection in Unsupervised Fuzzy Clustering," Proc. of the 41th Anniversary Conference KoREMA,Vol. 4, pp. 1-3, 1996.

[13] N. A. Mohamed, M. N. Ahmed, and A. Farag, “Modified Fuzzy C-Mean in Medical Image Segmentation”. Acoustics, Speech, and Signal Proc., Proceedings 1999 IEEE International Conference, pp. 3429-3432, Vol.6, 1999.

[14] E. Micheli Tzanakou, *Supervised and Unsupervised Pattern Recognition: Feature Extraction and Computational Intelligence*, Rutgers University, Piscataway, New Jersey, USA, pp. 50-52,1999.