International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:6, No:9, 2012

Implementing a Visual Servoing System for Robot Controlling

Maryam Vafadar , Alireza Behrad and Saeed Akbari

Abstract**—**Nowadays, with the emerging of the new applications like robot control in image processing, artificial vision for visual servoing is a rapidly growing discipline and Human-machine interaction plays a significant role for controlling the robot. This paper presents a new algorithm based on spatio-temporal volumes for visual servoing aims to control robots. In this algorithm, after applying necessary pre-processing on video frames, a spatio-temporal volume is constructed for each gesture and feature vector is extracted. These volumes are then analyzed for matching in two consecutive stages. For hand gesture recognition and classification we tested different classifiers including k-Nearest neighbor, learning vector quantization and back propagation neural networks. We tested the proposed algorithm with the collected data set and results showed the correct gesture recognition rate of 99.58 percent. We also tested the algorithm with noisy images and algorithm showed the correct recognition rate of 97.92 percent in noisy images.

Keywords**—**Back propagation neural network, Feature vector, Hand gesture recognition, k-Nearest Neighbor, Learning vector quantization neural network, Robot control, Spatio-temporal volume, Visual servoing

I. INTRODUCTION

FOR years, scientists have been trying to teach machines how to see like humans, Nowadays Artificial vision and visual servoing is developed and Recent research has seemed to show advanced applications in different areas[1-3]. One of these emerging applications is robot control. Artificial vision for visual servoing is a rapidly growing discipline and human-machine interaction plays a significant role for interaction of human with a robot. Human hand gesture recognition is one of the mostly used methods for human-machine interaction and can be used as a perfect method for controlling a robot. Different vision-based hand gesture recognition methods have been proposed including applying Fourier transform [4] or Principal Component Analysis (PCA) [5-7] on images. Edge orientation histogram is also used for hand gesture recognition [8]. Using temporal templates is another method for hand gesture recognition [9-11]. The basic idea of temporal template is the projection of the temporal pattern of motion into a single image-based representation and the extraction of the appropriate features from this image.

In the most of these methods, video frames are primarily processed in the groups of two consecutive frames. For example in temporal template methods, by differentiating one frame from the other, the dynamics occurring in an image sequence are extracted. Edge orientation histogram is another pair-wise approach that recognizes gestures by extracting edge points as features in each frame and matching them between frames. Although the two-frame-based approaches have been very successful in some applications, they face considerable difficulties; if they used for example to reason about non-constant velocity motion [12]. Also, the presence of noise and occlusion affects feature extraction and correspondence stages and degrades the performance of the algorithms. Spatio-temporal volumes have been actively in research as a means to alleviate the shortcoming of the traditional pair-wise approaches. In fact, video sequence can be defined as a spatial intensity pattern that changes with time. Spatio-temporal volume unifies the analysis of spatial and temporal information by constructing a volume of data in which consecutive images are stacked to form a third, temporal dimension. Then processing of an image sequence is done as a 3D volume and required features are extracted from this volume.

The major advantages of this representation are as follows [12]:

1. By conjointly providing spatial and temporal continuity, the complexity of feature correspondence is significantly reduced.
2. Noise in each frame has low effect in the volume and rarely can degrade the performance of algorithm.
3. By analyzing feature structures in this volume, we may reason about non-constant velocities, efficiently.
4. Occlusion events are made much easier to detect, as they are represented explicitly in this volume as truncated paths [13-14].

Several spatio-temporal and spatio-temporal frequency representations have been proposed such as extracting geometric features from spatio-temporal volume [15], applying the derivative of Gaussian transform [16] and Fourier transform [17] on spatio-temporal volume.

This paper presents a new spatio-temporal algorithm for hand gesture recognition aims to control robots. In this algorithm, after applying necessary pre-processing on video frames, spatio-temporal volume is constructed for each gesture.

These volumes are then analyzed and feature vectors are extracted. In comparison with other method, we have used contour of the hand for volume construction. This reduces the computation overhead of the algorithm and make the hand gesture recognition algorithm less or insensitive to illumination change, hand size and hand color.

Maryam Vafadar Department of Electronic, Sama technical and vocational training college, Islamic Azad University, Islamshahr Branch, Islamshahr, Iran, E-mail: vafadar_ma@yahoo.com

Alireza Behrad Department of Electronic, Shahed University, Tehran, Iran, E-mail: behrad@shahed.ac.ir

Saeed Akbari Department of Mathematics, Islamic Azad University, Tehran, Iran, E-mail: sa_akbari1@yahoo.com

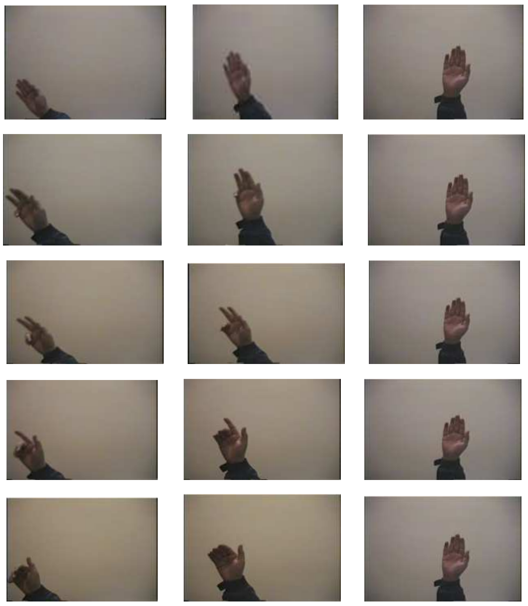
To increase the accuracy of the proposed algorithm and decrease the computation overhead in large databases, we have used a two-stage algorithm for hand gesture recognition.

In the first stage the motion information of contour center is used for classification and in the second stage the information of hand contours is employed. For hand gesture recognition and feature vector classification we tested three different classifiers including k-Nearest neighbor, learning vector quantization and back propagation neural networks. We tested the proposed algorithm with the collected data set and results showed the correct gesture recognition rate of 99.98 percent. We also tested the algorithm with noisy images and algorithm showed the correct recognition rate of 92.08 percent in noisy images.

The paper continues as follows: In section II, we describe the collected data set of hand gestures. Section III explains the pre-processing steps for constructing spatio-temporal volumes. Section IV analyzes spatio-temporal volumes for feature vector extraction. In section V, feature vector classification using three different classifiers are presented. Section VI shows experimental results for the proposed method and conclusion appears in section VII.

II. Collected data set

Data set collected from different persons with different hand size and skin color. No obvious feature or sign exists on the hands. The collected data set contains different videos of 24 hand gestures. These gestures include 4 different motions of the hand including right, left, up and downward motions with different hand fingers positions and motions. Collected data include 4 motion directions and different fingers motions in each direction provide 6 different gestures for each direction. So our collected dataset includes 24 different hand gestures. Sample frames of 6 rightward hand gestures are illustrated in figure 1. Sample frames of 6 rightward hand gestures are illustrated in figure 1.



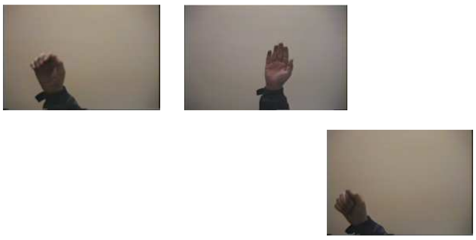


Fig. 1 Sample frames of 6 rightward hand gestures. Each row shows one gesture and frames are arranged from right to left

Gestures are repeated 10 times in different illumination condition and with different user's hand sizes. Hand motion velocities may differ and frames number of each gesture may vary from 18 to 44. Resolution of the video frames is 320*240 pixels.

III. Pre-Processing

A. Hand Segmentation

Prior to hand segmentation, it is necessary to apply noise removing filters to captured video frames. We applied median filter with the window size of 3 to each frame for handling salt&pepper noise. Gaussian filter with standard deviation of 0.8 is also used for removing Gaussian noise.

Skin-color based segmentation has proven to be an effective method for the segmentation of hand in fairly unrestricted environment [18]. Generally video capture systems generate images in RGB format which is known to be sensitive to the conditions of lighting and takes much processing cost, because RGB includes the mixed information of color and intensity [19-21].

Therefore, YIQ and HSV color systems that are known to be less sensitive to lighting than RGB format are adopted. We use both color systems to obtain high accuracy in our algorithm. In YIQ color system, Y means intensity, while I and Q represent color information. Also in HSV system, V denotes intensity, while H and S components specify color information. To reduce the effect of lighting and intensity, only I&Q values and H&S moments are used to build up a skin-color model for hand segmentation.

We separately apply k-means algorithm to I&Q and H&S color information on each frame and cluster color regions in two images. Similar color region are segmented in the same cluster with this method. Common regions of two resultant images denote the hand region. Figure 2 shows the result of hand segmentation algorithm on sample frames.

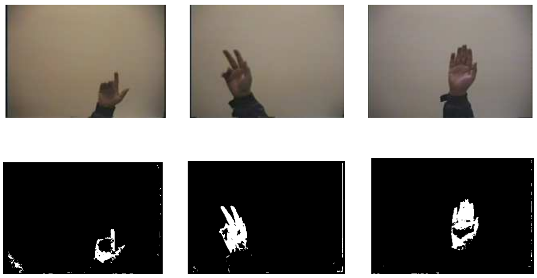


Fig. 2 Result of hand segmentation algorithm on sample frames; top row includes original frames and bottom row shows segmented hands

B. Spatio-temporal Volume Construction

The first stage for spatio-temporal volume construction is the extraction of hand contour in different frames of a video. As it is shown in figure 3, hand region may not be segmented efficiently because of different illumination condition and different hand color. To handle segmentation noise and extract true hand contour, we applied proper morphology operators to the resultant binary image of the previous stage and used edge information of the original image for contour correction as follows:

1. Remove connected component with the size of smaller than P pixels. This removes small regions that are wrongly segmented as hand region.
2. Fill gaps and small dints to have hand area with smoothed contour. We used disk-shaped structuring element with the radius of 10. This value was obtained experimentally.
3. Fill hand region and extract hand contour.
4. Apply canny edge detector to original image and find edge contours.
5. Use edge contours to correct the hand contour obtained in step 3.

Figure 3 shows the result of hand contour extraction algorithm.

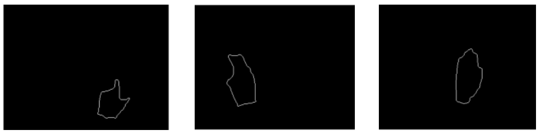


Fig. 3 Result of hand contour extraction algorithm

When hand contours are extracted in all the frames a gesture, they are stacked over each other to constitute a spatio-temporal volume. Figure 4 shows an example of spatio-temporal volume constructed from hand contours of a typical gesture.

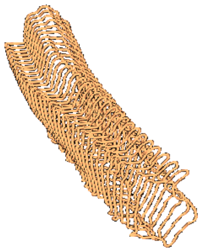


Fig. 4 Example of a spatio-temporal volume constructed from hand contours of a typical gesture

IV. Volume Mathematical Analysis

To increase the accuracy of the proposed algorithm and decrease the computation overhead in large databases, we have used a two-stage algorithm and spatio-temporal volumes are analyzed in two consecutive stages. In the first stage, we extract motion trajectory of hand from spatio-temporal volume. Motion trajectory of hand forms a spatio-temporal curve which we call it “motion curve”.

Motion curve is simply extracted by calculation of the center of mass for all the contours in spatio-temporal volume. The first stage of hand gesture recognition is performed by motion curve matching, however this stage may not be able to differentiate between 24 hand gestures. Because gestures with similar hand motions and different finger motions may generate similar motion curves. Therefore we apply spatio-temporal curve matching to select only candidate gestures and the final gesture is selected by the process of second stage of the algorithm. In the second stage of the algorithm, we consider the shape of spatio-temporal volumes for gesture matching and selection of the final gesture. This two-stage scheme use both motion and shape information for gesture matching; therefore this method not only increases the performance of the algorithm but also decreases the computation overhead of it.

A. Trajectory Matching

As mentioned before hand gestures create a trajectory in 3D spatio-temporal coordinates with time information associated to it. Motion curves are associated with motion trajectory of hand and are extracted by calculation of the center of mass for all the contours in spatio-temporal volumes. Figure 5
shows hand motion curves of our two sample gestures.

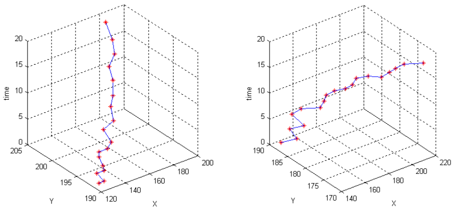


Fig. 5 Hand motion curves of sample gestures

To compensate for different gesture velocities, we apply a similar normalization method on time dimension. For this purpose we divide spatio-temporal volumes into M distinct slice of frames as follow:

$$M = \frac{N}{f} \tag{1}$$

where N is the total number of frames in the volume, M is the number of slices and f is the number of frames in each slice. We then calculate the average of contour centers in each slice and constitute M-tuple vector representing motion curve. Therefore, all gestures with different number of frames are denoted by M-tuple feature vectors, where M is constant. With the increase of hand speed the total number of frames, *N*, decreases, therefore *f* decreases too. Inversely low speed hand produces higher *f* number. For our data set, *N* varies from 18 to 44. We experimentally set *M*=6, therefore f varies from 3 to 7.

To solve the problem of initial hand position, we use the difference of consecutive center as follows:

$$D_i = C_{i+1} - C_i$$
$$D_i = (C_{i+1} + T) - (C_i + T) = C_{i+1} - C_i \tag{2}$$

Where C_i, C_{i+1} are the centers coordinates in two consecutive slices, *i* is the slice index, *T* is initial hand position and D_i is the difference value for slice i . It is obvious that D_i is insensitive to initial hand position T .

B. Volume Shape Matching

Our gesture database contains 24 gestures with different hand and finger motions. The motion of a finger doesn't significantly changes the center of hand contours and the related motion curve. Therefore the vector obtained in the first stage of the algorithm is only used to select some candidate gestures from database. Then the final gesture is selected using the shape information of spatio-temporal volume, which is the second stage of our gesture recognition algorithm. Previously we divided the spatio-temporal volume to M distinct slices. To obtain features from the shape of the volume, we further divide the volume into S equal-angle sectors. Dividing typical frame of a spatio-temporal volume to $S = 5$ and $S = 8$ sectors are illustrated in Figure 6.

Fig. 6 Dividing typical frame of a spatio-temporal volume to $S = 5$ and $S = 8$ sections

Dividing the spatio-temporal into M slices and S sectors, partitions the surface of spatio-temporal volume into a mesh with M*S elements which is called volume elements. The volume elements show the local shape of the volume. We use the shape information of volume elements to extract feature vector for gesture matching.

Similar to motion curve features, shape features should also be invariant to hand size, hand speed and hand initial position. We used two types of shape information as shape features which are unit normal vector and curvature information of volume elements. These two features are invariant to hand size, hand speed and hand initial position.

1. Curvature Based Features

To extract curvature information of volume elements, it is necessary to smooth volume shape. This removes volume noise and makes the shape feature more robust. To smooth volume shape, we simply smooth the entire hand contours in the volume. To smooth hand contour we represent hand contour as 2D circular curve as follow:

$$r(u) = (x(u), y(u)) \qquad u = 0 : T-1 \tag{3}$$

Where *T* is the total number of contour point. We then smooth the contour using one dimensional Gaussian function as follow:

$$X(u,\sigma) = x(u) * g(u,\sigma)$$

$$Y(u,\sigma) = y(u) * g(u,\sigma) \tag{4}$$

$$R(u,\sigma) = (X(u,\sigma), Y(u,\sigma))$$

Where $g(u,\sigma)$ is Gaussian function of width σ and $R(u,\sigma)$ is the smoothed contour.

We then calculate the contours curvature for all the points in the volume using the following equations:

$$\begin{gathered} k(u,\sigma) = \frac{X'Y'' - X''Y'}{((X')^2 + (Y')^2)^{3/2}} \\ X' = \frac{dX(u,\sigma)}{du}, X'' = \frac{d^2X(u,\sigma)}{du^2} \\ Y' = \frac{dY(u,\sigma)}{du}, Y'' = \frac{d^2Y(u,\sigma)}{du^2} \end{gathered} \tag{5}$$

Where $k(u,\sigma)$ denotes contour curvature at point $(x(u), y(u))$. Mean and variance of curvatures in the volume element are used as curvature features for the element. We calculate mean and variance respectively in equation 6.

$$\begin{gathered} \bar{k}_i = \frac{\sum_{p \in E_i} k_p}{P} \\ {\sigma_{k_i}}^2 = \frac{\sum_{p \in E_i} (k_p - \bar{k}_i)^2}{P} \end{gathered} \tag{6}$$

where E_i is the set containing all the contour points in volume element *i* and P is the total number of contour points in the element i.

2. Normal Vector of Volume Elements

Unit normal vector of volume elements are the second feature which we have used as shape feature of the volume. For calculation of normal vector we first fit a plane to contour points in volume elements. Then the normal vector of the plane is considered as normal vector of volume element. To fit the plane, we use least mean squared method. The plane equation has three parameters as follow:

$$ax + by + cz = 1 \tag{7}$$

where x, y and z denote points in spatio-temporal coordinates and a, b and c are normal vector moments of the plane. We use frame number as z. To calculate normal vector of the plane, we form the following equation:

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ x_P & y_P & z_P \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \dots \\ \dots \\ 1 \end{bmatrix} \Rightarrow AX = B \tag{8}$$

where($x_1, y_1, z_1, \dots, x_P, y_P, z_P$) are the coordinates of contour points in the volume element and a, b and c are normal vector moments. By solving the above equation using Least

Mean squared (LMS) method we can obtain the unit normal vector as follows:

$$X = \left(A^T A\right)^{-1} A^T B$$
$$Y = \frac{X}{\|X\|} \qquad (9)$$

Where Y denotes unit normal vector and $\|X\|$ represents the magnitude of vector *X*. Since the vector Y is unit, two elements of the vector is only required to be saved as shape feature.

C. Feature Vector Compression

In some of the gesture recognition algorithms, different compressing methods may be used to decrease the feature vector dimension. Compression rate is determined by a compromise between feature vector size and the algorithm accuracy.

PCA(Principle Component Analysis) and FLD(Fisher Linear Discriminant) are two samples of compressing methods that decrease dimension by mapping feature vectors to another space [22-23]. The number of elements for the feature vector of the proposed algorithm is 2*M for the first stage of the algorithm and 4*S*M for the second parts of the algorithm.

We don’t need to use additional operator to compress the feature vector. The total size of feature vector can be simply changed by M and S values. Larger S and M values increase the size of feature vector as well as the algorithm accuracy. Smaller M and S provides higher compressed rate but less accuracy. We discuss about optimum M and S values in experimental results section.

V.GESTURE RECOGNITION

Gesture recognition process has two parts. In the first part, the feature vectors are extracted for the dataset of known gestures. In the second part, the unknown gesture is matched with the dataset of known gestures to recognize the gesture. As mentioned before, our algorithm has two stages.

In the first stage a feature vector of size M*2 is extracted, where *M* is the number of slices. To recognize an unknown gesture using the extracted information of first stage, we compare it with all gestures in dataset using sum squared difference (SSD) method as follow:

$$SSD_k = \sum_{i=}^{M} \sum_{j=1}^{2} \left[Y(i,j) - S_k(i,j)\right]^2 \qquad (10)$$

Where Y is the feature vector of unknown gesture and S_k is the feature vector for *k’th* gesture in dataset.

As we mentioned before, the feature vector of first stage of the algorithm is not sufficient for selection of final gesture, therefore we use SSD values to select most similar gestures as candidate gestures. Then the candidate gestures are fed into second stage of the algorithm to select the final gesture.

For second stage of the algorithm we used three different classifiers to classify hand candidate gestures:

- k-Nearest Neighbor (kNN)
- Learning Vector Quantization Neural Network (LVQ N.N.)
- Back Propagation Neural Network (BP N.N.)

k-Nearest Neighbor (kNN) algorithm. k nearest neighbor is a supervised learning algorithm based on minimum distance of input hand gesture from the training candidate gestures and determining k-Nearest neighbors to classify gestures. We use Euclidean distance to determine nearest neighbors. Each of these k neighbors is related to a class of gesture, we select a class of gesture with maximum number of occurrence. Learning vector quantization (LVQ) is a method for training competitive layers in a supervised manner. An LVQ network has two layers. First layer is competitive layer and second layer is linear. Back propagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions.

VI. EXPERIMENTAL RESULTS

We implemented the proposed with our collected data set. We tested our algorithm with different *M*, *S* and *K* values where *M* and *S* are slice and sector number and K is the parameter of kNN algorithm. We also tested the second stage of the proposed algorithm with two set of extracted information as follows:

- Only unit normal vector of the volume elements is used (First method)
- Unit normal vectors and curvature based features are used (Second method)

We analysis these two methods with different classifiers and S (Section number) values-criterion of compression. Resulted hand gesture recognition rates for first and second methods are shown in Fig. 8 and 9 respectively.

In LVQ neural network, if two input vectors are very similar, the competitive layer probably will put them in the same class, but there is no mechanism in a strictly competitive layer design to say whether or not any two input vectors are in the same class or different classes. However, we select optimized value of 20 for number of hidden neurons in competitive layer and 300 epoch number for training network. Approximated function of BP neural network may be not enough accurate, too. To cope with this problem we try to select optimum parameters experimentally. We select different training epochs and neuron number for different parameters of the algorithm. As for $S = 9$, we use optimized training epoch number of 600 and hidden neuron number of 30, whereas for $S = 18$, optimized training epoch number of 450 and hidden neuron number of 15 are selected.

Bar diagram of hand gesture recognition rates for different classifiers for first and second methods are shown in Figures 7 and 8, respectively. As it is, k-Nearest neighbor has the best recognition rate in the classifiers. In First method with kNN(k=2) classifier, recognition rates are resulted between 96.66 to 99.58 percent. Similar recognition rates for Second method are resulted between 98.75 to 99.58 percent.

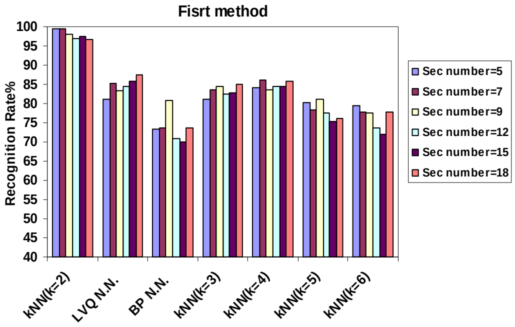


Fig. 7 Bar diagram of hand gesture Recognition Rates for different classifiers of First method

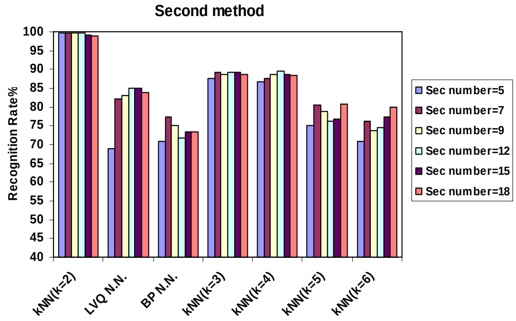


Fig. 8 Bar diagram of hand gesture Recognition Rates for different classifiers of Second method

A. *Experiments On Noisy Data*

We test the robustness of our proposed method in the presence of noise. We add salt & pepper noise and Gaussian noise ($N(0,\sigma)$) on each frame of captured hand gesture, separately. Noise density of salt & pepper noise varies from 0.04 to 0.14. σ value of Gaussian noise increases from 2 to 5. Example frames with salt & pepper and Gaussian noise under two different noise densities and σ values are shown in Figures 9 and 10, respectively.

Figure 11 shows recognition results for various noise density of salt & pepper noise. Notice that we use kNN(k=2) as best classifier and S (Section number)=9 as an optimized value for compression. Recognition result for various σ values of Gaussian noise is shown in Figure 12.

As it is shown, the lowest recognition rate of first method in presence of salt & pepper noise is 94.58%. Similar recognition rate of second method is 97.92%. Also, in presence of Gaussian noise, we approach the lowest recognition rates 92.08% and 95.83% for first and second methods, respectively.

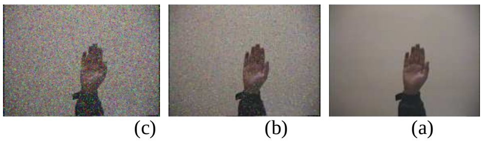

(c) (b) (a)

Fig. 9 (a) Example frame, (b) salt & pepper noisy frame with noise density=0.04, (c) salt & pepper noisy frame with noise density=0.14

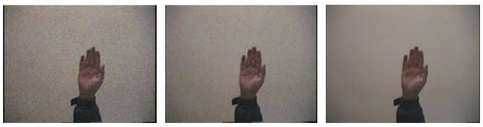

(c) (b) (a)

Fig. 10 (a) Example frame, (b) Gaussian noisy frame with σ = 2, (c) Gaussian noisy frame with σ = 5

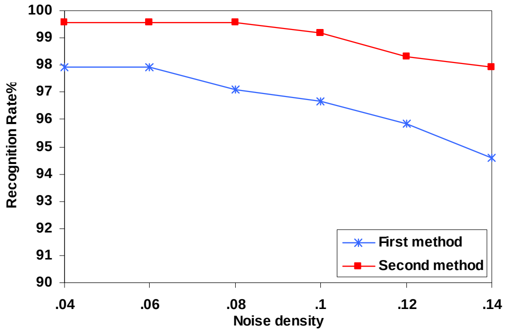


Fig. 11 Hand gesture recognition rates in presence of salt & pepper noise with various noise density

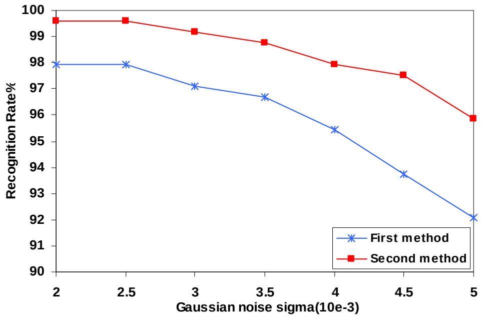


Fig. 12 Hand gesture recognition rates in presence of Gaussian noise with various σ values

VII. Conclusion

A new algorithm based on spatio-temporal volume for visual servoing aims to robot control is presented. In this algorithm, after applying pre-processing steps on video frames, spatio-temporal volumes are constructed for each gesture. These volumes are then analyzed and feature vectors are extracted. At last, three different classifiers including k-Nearest Neighbor, Learning Vector Quantization and Back Propagation Neural Networks are presented for feature vector classification and hand gesture recognition. We tested the proposed algorithm with the collected data set and experimental results shows the reliability of our algorithm even in presence of noise. In addition, the proposed algorithm is robust to traditional problems of gesture recognition like illumination variations, hand position in each frame, different hand gesture velocities and hand sizes. So the algorithm can be used as an efficient visual servoing system for robot controlling.

References

[1] Pavel. A. and Buiu. C.: Development of an embedded artificial vision system for an autonomous robot, International Journal of Innovative

Computing, Information and Control(ICIC 2011), Vol. 7, Num. 2, pp. 745-762, 2011.

[2] Ospina. E., Valencia. J. and Madrigal. C.: Traffic flow control using artificial vision techniques, 2011 6'th Colombian Computing Congress (CCC) available on IEEE explore, 2011.

[3] Cindy. X. You and Mark. A. Tarbell.: Microcomputer-based artificial vision support system for real-time image processing for camera-driven visual prostheses, Journal of Biomedical Optics, Vol. 15, Issue 1, Research Papers: Imaging, 2010.

[4] Harding. P.R.G. and Ellis. T.: Recognition Hand Gesture Using Fourier Descriptors, Proc. of IEEE Conf. on Pattern Recognition(ICPR2004), vol. 3, pp. 286-289, 2004.

[5] Dionisio. C.R.P., Cesar. R.M. and Jr.: A Project for Hand Gesture Recognition", Proc. of IEEE Symposium on Computer Graphics and Image Processing, pp. 345, 2000.

[6] Lamar. M.V., Bhuiyan M.S. and Iwata. A.: Hand Gesture Recognition Analysis and an Iimproved CombNET-II, Proc. Of IEEE Conf. on Man and Cybernetics, vol. 4, 1999.

[7] Moghaddam. B. and pentland. A.: Probabilistic Visual Learning for Object Detection, Conf. on Computer Vision, Cambridge, MA, 1995.

[8] Moghaddam. B.: Principal Manifolds and Bayesian Subspaces for Visual Recognition, Proc. Of IEEE Conf. on Computer Vision, ICCV99, 1999.

[9] Freeman. W.T. and Roth. M.: Orientation Histogram for Hand Gesture Recognition, IEEE Int. Workshop on Automatic face and gesture recognition, 1995.

[10] Shan. C., Yucheng. W., Xianchao. Q. and Tieniu. T.: Gesture Recognition Using Temporal Template Based Trajectories", Int. Conf. on Pattern Recognition, vol. 3, pp. 954 – 957, 2004.

[11] Kumar. S., Kumar. D.K., Sharma. A. and McLachlan. N.: Classification of Hand Movements Using Motion Templates and Geometrical based Moments, vol. 3, pp. 299 – 304, 2004.

[12] Vafadar. M. and Behrad. A.: Human Hand Gesture Recognition Using Motion Orientation Histogram for Interaction of Handicapped Persons with Computer, Proc. Of Int. Conf. on Image and Signal Processing, ICISP2008, 2008.

[13] Collins. T.: Analysing Video Sequences using the Spatio-temporal Volume, Informatic Research Review, 2004.

[14] Konrad. J. and Ristivojevic. M,: Joint Space-time image sequence segmentation:Object Tunnels and Occlusion Volumes, Proc. Of Int. Conf. on Acoustic, Speech and Signal Processing, 2004.

[15] Swaminathan. R., Kang. S.B. and Criminisi. A. and Szeliski. R.: On the Motion and Appearance of Specularities in Image Sequences, Proc. Of European Conf. in Computer Vision, 2002.

[16] Bolduc. M. M. and Deschenes. F.: Collision and Event Detection using Geometric Features in Spatio-temporal Volumes, Proc. Of IEEE Canadian Conf. on Computer and Robot Vision, CRV2005, 2005.

[17] Bloom. J. A. and Reed. T. R.: On the Compression of Video using the Derivative of Gaussian Transform, Proc. Conf. on Signals, Systems and Computers, 1998.

[18] Ohara. Y., Sagawa. R., Echigo. T. and Yagi. Y.: Gait Volume: Spatio-temporal Analysis of Walking, European Conference on Computer Vision, ECCV2004, 2004.

[19] Nianjun Liu Lovell. B. C. and Kootsookos. P. J.: Evaluation of HMM Training Algorithms for Letter Hand Gesture Recognition, IEEE Int. Sym. On Signal Processing and Information Technology, ISSPIT2003, 2003.

[20] Chang. M.C., Matshoba. L. and Preston. S., "A Gesture Driven 3D interface", Technical Report CS05-15-00, University of Cape Town, 2005.

[21] Yoon. H.S., Min. B.W., Soh. J., Bae. Y. and Yang. H.S., "Human Computer Interface for Gesture-based Editing System", IEEE Int. Conf. on Image Analysis and Processing, 1999.

[22] Liu. N., Lovell. B. C., Kootsookos. P. J., "Evaluation of HMM Training Algorithms for Letter Hand Gesture Recognition", IEEE Int. Sym. on Signal Processing and Information Technology, vol. 14-17, pp. 648 – 651, 2003.

[23] Shalbaf. R., vafadoost. M. and Shalbaf. A.: Lipreading Using Image Processing for Helping Handicap", Iranian conf. on Biomedical Engineering, 2007.