Pattern Recognition Letters 31 (2010) 1359–1368

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A person independent system for recognition of hand postures used in sign language

Daniel Kelly *, John McDonald, Charles Markham

Computer Science Department, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland

ARTICLE INFO

Article history:
Received 14 May 2009
Received in revised form 11 January 2010
Available online 16 February 2010
Communicated by G. Borgefors

Keywords:
Feature representation
Moments
Size and shape
Object recognition
Tracking
Classifier design and evaluation

ABSTRACT

We present a novel user independent framework for representing and recognizing hand postures used in sign language. We propose a novel hand posture feature, an eigenspace Size Function, which is robust to classifying hand postures independent of the person performing them. An analysis of the discriminatory properties of our proposed eigenspace Size Function shows a significant improvement in performance when compared to the original unmodified Size Function.

We describe our support vector machine based recognition framework which uses a combination of our eigenspace Size Function and Hu moments features to classify different hand postures. Experiments, based on two different hand posture data sets, show that our method is robust at recognizing hand postures independent of the person performing them. Our method also performs well compared to other user independent hand posture recognition systems.

1. Introduction

Gestures are a form of body language or non-verbal communication. Stokoe and William (2005) defined three gesture aspects that are combined simultaneously in the formation of a particular sign; what acts, where it acts, and the act. These aspects translate into building blocks that linguists describe; the hand shape, the position, the orientation and the movement. Using the four components Stokoe uses to represent a gesture (Stokoe and William, 2005), hand gestures can be classified as either hand postures (hand shape and orientation) or temporal gestures (movement and position) (Wu et al., 1999). Since hand postures not only can express some concepts, but also can act as special transition states in temporal gestures, recognizing hand postures or human postures is one of the main requirements in gesture recognition. In this work, we propose a novel hand posture feature, an eigenspace Size Function, which is robust to classifying hand postures independent of the person performing them.

1.1. Related work

To describe the shape of the hand, a number of methods for 2D shape representation and recognition have been used. These include segmented hand images, binary hand silhouettes or hand blobs, and hand contours. Cui and Weng (2000) used normalized segmented hand images as features and reported a 93.2% recognition rate on 28 different signs. Similarly Kadir et al. (2004) use normalized segmented hand images and greedy clustering techniques to recognize hand shapes with 75% accuracy. Principal component analysis (PCA) has been shown to be successfully applied to gesture data to reduce dimensionality of the extracted features. Deng and Tsui (2002) apply a two-layer principal component analysis/multiple discriminant analysis scheme. A non-user independent experiment showed a recognition rate up to 70% on 110 signs. Imagawa et al. (2000) calculate an eigenspace on segmented hand images and signs are then represented by symbols which correspond to clusters. Results show a recognition rate of 92% on 33 signs. Patwardhan and Roy (2007) uses a predictive Eigen–Tracker to track the changing appearance of a moving hand. The algorithm obtains the affine transforms of the image frames and projects the image to the eigenspace. An accuracy measurement of 100% is reported from tests using 80 gestures, although 64 of the test gestures where used in training and gestures used were very simple and distinct. Holden and Owens (2003) present a topological formation shape representation that measures the fingers only. A recognition rate of 96% was achieved when classifying four distinct hand shapes.

Contour based features have also been shown to perform well in hand posture recognition. Huang and Huang (1998) used Fourier descriptors of the hand contour, a Hausdorff distance measure and graph matching algorithms within a 3D Hopfield neural network to recognize signs with 91% accuracy. Al-Jarrah and Halawani (2001) extracted features by computing vectors between the

* Corresponding author. Tel.: +353 87786149; fax: +353 17083848.
E-mail addresses: dankelly@cs.nuim.ie, dnl.kelly1@gmail.com (D. Kelly).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.02.004

contour's center of mass and localized contour sequences. Recognition of 30 gestures is reported with an accuracy of 92.55%. Handouyahia et al. (1999) presented a sign language alphabet recognition system using a variation of Size Functions (Uras and Verri, 1995) called moment based Size Functions, which recognized 25 different signs with 90% accuracy.

Starner et al. (1998) show that geometric moments perform well in hand gesture recognition. A head mounted camera tracks the hands using skin color. Hand blobs are extracted from video sequences and a 16 element geometric moment feature vector is used to describe hand shape. A recognition rate of 98% for sign language sentences is reported. Tanibata et al. (2002) use a set of six geometric moments to recognize Japanese Sign Language, although it was reported that recognition performed well, no recognition accuracy was specified. Bauer and Hienz (2000) describe a German sign language recognition system where hand shape feature experiments showed that the area of the hands performed well as a feature. It was reported that the system achieved an accuracy of 75% when only taking hand area into account. All training and test data was recorded from the same subject performing signs and a recognition rate of 94% and 91.7% was reported for systems based on a 52 and 97 sign lexicon respectively.

Although the methods described above report high recognition accuracy, most performance measures where results of signer dependent experiments carried out by testing the system on subjects that were also used to train the system. Analogous to speaker independence in speech recognition, an ideal sign recognition system should give good recognition accuracy for signers not represented in the training data set. Farhadi et al. (2007) propose a signer independent ASL transfer learning model to build sign models that transfer between signers. Results show their method achieved classification accuracy of 67% when classifying signs from a 90 word vocabulary, but their method does not explicitly deal with hand posture recognition. User independent hand posture recognition is particularly challenging as a user independent system must cope with geometric distortions due to different hand anatomy or different performance of gestures by different persons. Licsr and Szirnyi (2005) develop a hand gesture recognition system with interactive training. Their proposed solution to user independent hand posture recognition system is based around the idea of an on-line training method embedded into the recognition process. The on-line training is interactively controlled by the user and adapts to his/her gestures based on user supervised feedback where the user specifies if detected gesture were incorrectly classified. This method is shown to work very well in the scenario where the hand posture recognition system is being used as a HCI interface for a camera-projector system allowing users to directly manipulate projected objects with the performed hand gestures. While it is feasible to implement on-line retraining of gestures based supervised user feedback in this HCI scenario, implementing this model in an automatic sign language recognition system would make the performance of sign language un-natural and thus is not a feasible option for this work. Triesch and von der Malsburg (2002) proposed a user independent hand posture recognition system using elastic graph matching which reported a recognition rate of 92.9% when classifying 10 hand postures. The elastic graph matching method showed very promising results but was reported to have a high computational complexity with the method requiring several seconds to analyze a single image. Just et al. (2006) recognize the same of hand postures used by Triesh et al. using the Modified Census Transform with a recognition rate of 89.9%.

It is the goal of this work to develop an accurate user independent hand posture recognition system which can classify hand postures in real time allowing the classification of hand image from continuous video shots.

2. Hand features

In this work we propose a pattern recognition framework to classify segmented hand images. A number of works have proposed techniques for the segmentation of hands from an image sequence. Some hand segmentation techniques include the work of Yang et al. (2009), Holden et al. (2005), Cooper and Bowden (2007) where hand segmentation is carried out using motion and skin color cues (see Fig. 1). These methods produce a binary image, or hand contour, of the hand and experiments show the methods perform robustly in the domain of sign language feature extraction.

Based on the fact that there exists a number of robust hand segmentation algorithms which can be used to produce a binary image of the hand, we propose a technique which can robustly recognizes hand postures using the binary image of the hand.

In this work, we will show how our proposed hand shape features are computed from a segmented hand contour. A thorough evaluation of the discriminatory properties of our proposed features will be carried out as well as an evaluation of our proposed recognition framework. Experiments will be carried out using different hand segmentation methods in order to evaluate our proposed hand shape features.

3. Shape representations

Appearance-based gesture recognition requires an effective feature set that can separate the hand shapes (Pavlovic et al., 1997). This work presents a method of hand shape representation computed from the raw binary image and external contour extracted from the image. We propose a novel eigenspace Size Function shape representation which is calculated from the external contour. A Hu moment feature set is also generated from the raw binary image. Accurate shape representations must be able to identify similar shapes and distinguish between different shapes, therefore performance tests on different variations of the proposed shape representation will be carried out with the goal of achieving the optimal hand shape representation.

3.1. Hu moments

Hu moments (Hu, 1962), which are a reformulation of the non-orthogonal centralized moments, are a set of transition, scale and rotation invariant moments. The set of Hu moments, $I = \{I_1, I_2, I_3, I_4, I_5, I_6, I_7\}$, are calculated from the hand contour.

3.2. Size Functions

Size Functions are integer valued functions which represent both qualitative and quantitative properties of a visual shape (Uras and Verri, 1995).

For a given contour, extracted from the binary image of a hand, let G be a graph whose vertices are the points of the contour. Let φ, the measuring function, be a real-valued function defined on the

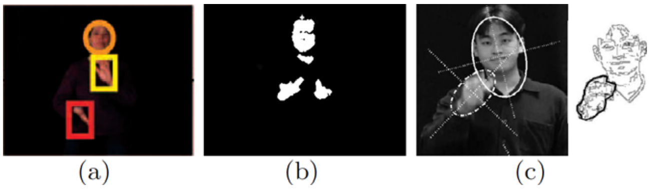


Fig. 1. Examples of different hand segmentation results from (a) Yang et al. (2009) (b) Cooper and Bowden (2007) and (c) Holden et al. (2005).

vertices of G (see Fig. 2a). The Size Function ℓ_φ induced by the measuring function φ, is an integer valued function defined on a real pair (x, y) according to the following algorithm:

1. Find the subgraph $G_{\varphi \leqslant y}$ of G determined by the points p with $\varphi(p) \leqslant y$ (see Fig. 2b).
2. Identify the connected components of $G_{\varphi \leqslant y}$ (see Fig. 2b).
3. The Size Function ℓ_φ at the point (x, y) equals the number of connected components of $G_{\varphi \leqslant y}$ which contain at least a vertex with $G_{\varphi \leqslant x}$ (see Fig. 2c–e);

When identifying the number of connected components of the graphs $G_{\varphi \leqslant y}$ and $G_{\varphi \leqslant y}$, it should be noted that the graphs are circular. Therefore, in Fig. 2d, there exists three connected components of $G_{\varphi \leqslant y}$ which contain at least a vertex with $G_{\varphi \leqslant x}$, and not four which would be the case if the graphs where not circular. This ensures that the number of connected components will remain the same independent of the start and end point for which the measuring function was computed.

The theory of Size Functions does not identify a formal tool to resolve a suitable measuring function. Therefore, a suitable measuring function must be found heuristically. As defined by Stokoes' model (Stokoe and William, 2005), a hand posture is made up of the shape and orientation of the hand. Thus, for the application of classifying hand postures performed in sign language, the measuring function chosen must be sensitive to orientation changes of the hand (although a suitable classifier should not be sensitive to minor changes in hand orientation). With Stokoes model in mind, the measuring function model proposed in this work utilizes a family of measuring functions indexed by the angle $\theta \epsilon \Big\{ 0, 1\frac{2\pi}{N_\Theta}, 2\frac{\pi}{2N_\Theta}, \cdots, (N_\Theta - 1)\frac{2\pi}{N_\Theta} \}$, where N_Θ is the total number of rotation angles used. Each measuring function $\varphi_\theta(p)$ is a function which rotates p about the center of gravity of G and measures the distance between the horizontal axis and a point p on the graph G_θ. The horizontal axis is a line which passes through the minimum point of G_θ. For every θ, a Size Function $\ell_{\varphi\theta}$ is generated, resulting in a set of Size Functions $\Gamma_\varphi = \{\ell_{\varphi 1}, \ell_{\varphi 2}, \ldots, \ell_{\varphi N_\Theta}\}$. The sensitivity of the system to orientation can then be controlled by means of adjusting N_Θ. As N_Θ increases, the number of rotations and Size Functions grows and the margin between each θ decreases. As the margin between each θ decreases, the effect small changes in orientation has on the final classification increases.

To illustrate the concept of Size Functions and their application in analyzing hand postures used in sign language, a specific example will be used. For this example, let $N_\Theta = 4$. The hand contour is rotated to each of the four θ values (see Fig. 3(a)), the measuring function is applied to each of the four rotated contours (see Fig. 3b) and the Size Functions are then generated from each of the measuring functions (see Fig. 3c).

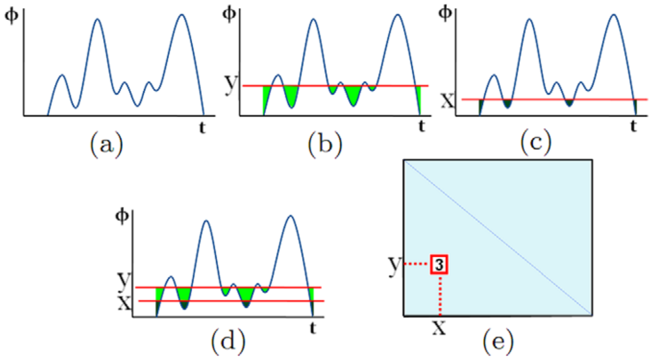


Fig. 2. (a) Graph of some measuring function φ (b) Shaded region $\equiv \varphi \leqslant y$. (c) Shaded region $\equiv \varphi \leqslant x$. (d) Graph depicting $\varphi \leqslant y$ and $\varphi \leqslant x$. (e) Graph of Size Function l_φ with current $l_\varphi(x, y) = 3$.

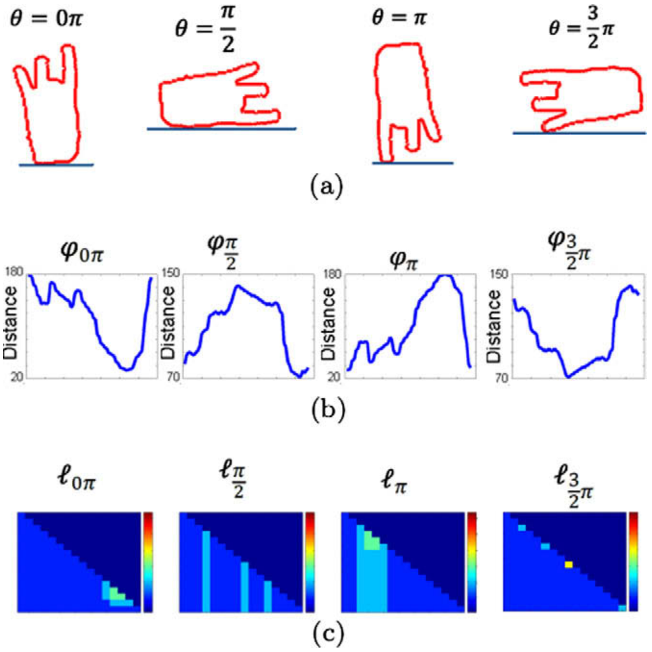


Fig. 3. (a) θ rotation applied to hand contour (b) measuring function φ_θ applied to hand contour (c) Size Function $\ell_{\varphi\theta}$ generated.

3.3. PCA and Size Functions

In order to quantify the shape information held in a Size Function, we propose a more robust method of shape representation, as compared to the unmodified normalized Size Function representation used in (Uras and Verri, 1995; Handouyahia et al., 1999). We make an important improvement to the Size Function technique by developing a Size Function feature which is more robust to noise and small changes in shape which occurs from different people. Our technique is a method of incorporating eigenspace information into the hand posture feature using principal component analysis (PCA). PCA computes a linear projection from a high dimension input space to a low dimensional feature space. It is a statistical technique used for finding patterns in data of high dimensions. Since we are looking for similarities and differences between two Size Functions, we can utilize PCA in order to reduce the influence of noise, and small variations in shape by different persons, and highlight portions of the Size Function useful for user independent hand posture recognition.

To calculate the principal components of a Size Function, the Size Function is described as an $N \times N$ matrix $X_\theta = \ell_{\varphi\theta}$. The vector $\mathbf{u}$ is the empirical mean of X_θ (see Eq. (1)), B_θ is the mean subtracted $N \times N$ matrix (see Eq. (2)) and C_θ is the covariance matrix of B_θ (see Eq. (3)).

$$\boldsymbol{u}_\theta[m] = \frac{1}{N}\sum_{n=1}^{N} X_\theta[m, n] \tag{1}$$

where m and n refers to the row index and column index of the $N \times N$ matrix respectively and N refers to the width and height of the Size Function.

$$B_\theta = X_\theta - [\boldsymbol{u}_\theta, \boldsymbol{u}_\theta, \ldots, \boldsymbol{u}_\theta] \tag{2}$$

$$C_\theta = \frac{1}{N} B_\theta \cdot B_\theta^T \tag{3}$$

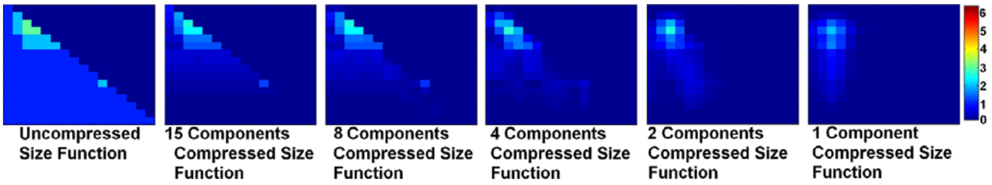


Fig. 4. Size Functions reconstructed with varying numbers of components.

The eigenvectors and eigenvalues of C_θ are calculated according to Eq. (4) were $\mathbf{v}$ is the eigenvector and w is the eigenvalue associated to the eigenvector.

$$C_\theta \boldsymbol{v} = w\boldsymbol{v} \tag{4}$$

The columns of the eigenvector matrix V_θ and the eigenvalue matrix W_θ are sorted in order of decreasing eigenvalue. This records the components in order of significance, the eigenvector with the highest eigenvalue being the *principal component*. Therefore the first column of V_θ, a $1 \times N$ vector, corresponds to the principal component vector.

Fig. 4 shows a Size Function which was reconstructed with varying numbers of components. It should be noted that the reconstructed Size Functions are not used as features, we show the reconstructed Size Functions in order to illustrate the effect PCA dimensionality reduction has on the Size Function. The eigenvectors used to reconstruct the Size Functions are the features we use to recognize hand shapes.

4. Data collection

In the experiments we describe in this work, we evaluate our techniques using hand shape videos and images from two separate data sets.

4.1. Jochen–Triesch static hand posture database

The first data set is a benchmark database called the Jochen Triesch static hand posture database Triesch and von der Malsburg, 2002. We utilize this data set in order to evaluate our hand posture recognition framework and directly compare our system to other hand postures recognition research. The database consists of 10 hand signs (see Fig. 5) performed by 24 different subjects against different backgrounds. All images are greyscale images and the backgrounds are of three types: uniform light, uniform dark and complex. In our system, posture recognition is carried out independent of hand segmentation. Neither motion or color are available in this data set, but,in general, color and motion are two important cues needed to segment the hands from complex backgrounds and this is acknowledged by Triesch and von der Malsburg (2002). Since there is no motion or color cues available, we do not consider the hand images with complex backgrounds. It is still possible to make a like with like comparison with other research in this area as most results in the literature report recognition rates achieved on the uniform background images independent of complex background images. In this data set, we extract contours from each image by segmenting the image using Canny edge detection and extracting the contour from the edge detected image using a border following algorithm (Suzuki and Be, 1985) (see Fig. 6).

4.2. ISL data set

The second data set is an Irish sign language (ISL) data set consisting of 23 hand signs (see Fig. 7), from the Irish sign language alphabet, performed by 16 different subjects wearing colored gloves. A total of 11040 images were collected. Each subject performed the 23 letters an average of three times. During the performance of each letter, a video sequence of 10 image frames was recorded and labeled in order to test the performance of our system when classifying hand images from continuous video shots. When performing a particular sign, subjects followed visual instructions from official Irish Deaf Society materials with the aim of ensuring natural performance of postures. A random selection of the images were validated by a certified Irish sign language teacher to ensure subjects had performed signs correctly.

All hand posture images were recorded with subjects asked to perform the postures as naturally as possible. Due to the natural performance of the hand postures, there was a large variance in the type of hand postures performed for each sign. Variations in performance were only limited by that of sign language limitations (i.e. a large variation in orientation may give a posture a different meaning and thus was not allowed, as instructed by a certified Irish sign language translator). Fig. 8 shows a visual example of a number of different ways the '*D*' sign was performed by different subjects.

In this data set, tracking of the hands is performed by tracking colored gloves (see Fig. 9a) using the mean shift algorithm (Comaniciu et al., 2000). To extract the external contour of the hand (see Fig. 9c) we segment the glove region in the image, using a back projection image computed during the mean shift algorithm (see Fig. 9b), and extract the external contour of the hand blob using a border following algorithm (Suzuki and Be, 1985).

The back projection image, which is used to find the hand contour in an image, typically can hold varying levels of noise. The noise in a back projection image refers to segmentation noise

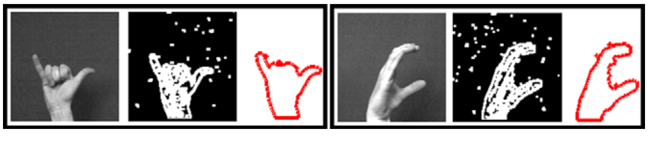


Fig. 6. Example of contour extraction from Y and C hand postures from Triesch data set.

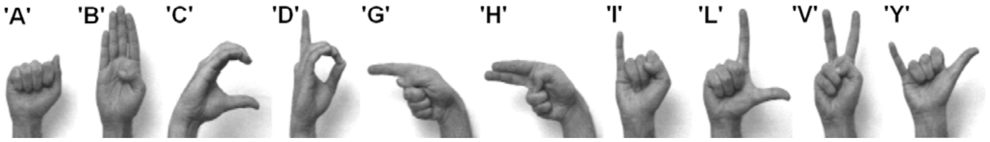


Fig. 5. The ten postures of the Triesch data set, performed by one subject against uniform light background.

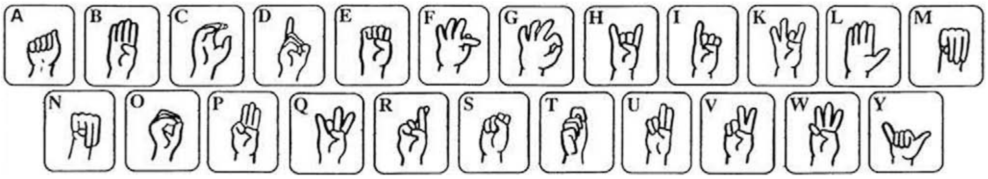


Fig. 7. 23 Static letters of the ISL alphabet (the signs for “j”,“x” and “z” cannot be performed statically and were not further considered).

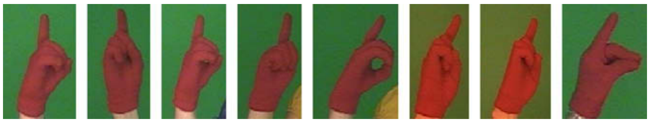


Fig. 8. Example of variation in performance of the ‘*D*’ sign.

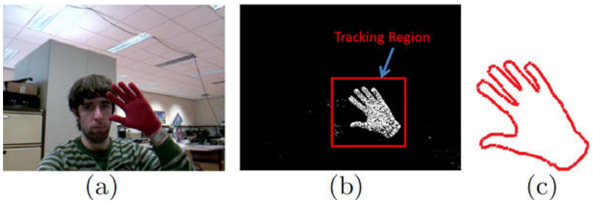


Fig. 9. Feature extraction from image: (a) original image, (b) back projection image, (c) extracted contour.

where white pixels are not part of the hand region, or where black pixels are part of the hand region. Noise in a back projection image can produce hand contours which hold noise. Fig. 10 shows an example of some noisy back projection images, and the corresponding contours extracted from the images, which were used during experiments discussed in later Sections.

In our experiments, the systems ability to deal with noise is tested due to the presence of typical segmentation noise in the back projection images.

5. Evaluation of discriminatory properties

In this section we perform experiments on different variations of Size Functions to find the optimal performing features for discriminating between hand postures. We also perform experiments to evaluate the discriminatory properties of combining Hu moments with our Size Function features and examine whether or not these features offer complementary information about hand posture patterns.

5.1. Size Functions and PCA performance

To examine how the eigenspace representation of a Size Function performs at discriminating between correct and incorrect signs performed by different people, an experiment was carried out to compare the eigenspace Size Function and the unmodified Size Function representations.

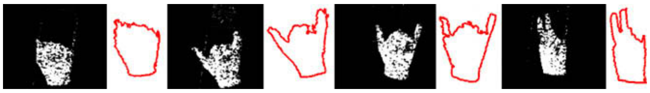


Fig. 10. Example of noisy back projection images and corresponding noisy contours.

We evaluate our proposed features on the Jochen Triesch static hand posture data set and on the ISL data set. For each hand sign in the data set we store a single hand image as a control image. The remaining set of hand images are stored as test images. We evaluate our proposed features by computing the distance between each control Size Function representation and all test contour Size Function representations.

We generated ROC graphs for each of the Size Function representations by calculating a confusion matrix from the control contour and test shape distance comparisons.

We define the function $D()$ as the distance measure computed between the control contour k and the test contour l. This procedure is carried out for both the eigenspace Size Function and the unmodified Size Function representations. To generate multiple points on the ROC graph, a confusion matrix is calculated from different threshold values $T(0 \leqslant T < +\infty)$.

Firstly, we computed a ROC graph for the unmodified Size Function representation where the distance between two unmodified Size Functions is calculated by Euclidean distance measure according to Eq. (5). Results of this test produced a AUC measurement of 0.735 and 0.756 for the ISL data set and the Triesch data set respectively.

$$D'(\Gamma^k, \Gamma^l) = \sqrt{\sum_{\theta=0}^{2\pi} \sum_{i=0}^{N} \sum_{j=0}^{N} (\Gamma_\theta^k[i,j] - \Gamma_\theta^l[i,j])^2} \tag{5}$$

A ROC graph was then computed for the eigenspace Size Function representation. To measure the distance between two eigenspace Size Functions, we generate the eigenvectors and eigenvalues for that Size Function, and choose only the first P eigenvectors, resulting in a matrix M_θ with dimensions $N \times P$. A Euclidian distance measure is then used to compare two eigenspace Size Functions as shown in Eq. (6). Results of this test produced an AUC measurement of 0.789 and 0.801 for the ISL data set and the Triesch data set respectively.

$$D^{\neg}(M^k, M^l) = \sqrt{\sum_{\theta=0}^{2\pi} \sum_{p=0}^{P} \sum_{i=0}^{N} (M_\theta^k[p,i] - M_\theta^l[p,i])^2} \tag{6}$$

A further modification to the eigenspace Size Function representation is proposed. We propose a scaling of the eigenvectors of each eigenspace Size Function based on a variance measure of their associated eigenvalues. We first calculate a weighting factor for each eigenvector x associated with the Size Function indexed by θ according to Eq. (7).

$$\varpi_\theta(x) = \frac{W_\theta[x]}{\sum_{p=0}^{P} W_\theta[p]} \tag{7}$$

where W_θ is the eigenvalues corresponding to the eigenvectors M_θ calculated from the Size Function $\ell_{\varphi\theta}$.

A second weighting factor is then calculated for each set of eigenvectors associated with the Size Function indexed by θ, such that the Size Function with the greatest total variance gets a greater weighting according to Eq. (8).

$$\varrho_\theta = \frac{\sum_{p=0}^{P} W_\theta[p]}{\sum_{\theta=0}^{N_\Theta} \sum_{p=0}^{P} W_\theta[p]} \tag{8}$$

The weighted eigenspace Size Function ζ is then computed according to Eq. (9).

$$\zeta_\theta[p,i] = M_\theta[p,i] \times \varpi_\theta(p) \times \varrho_\theta \tag{9}$$

A Euclidian distance measure between the weighted eigenspace Size Function is then calculated using Eq. (10).

$$D(M^k, M^l) = \sqrt{\sum_{\theta=0}^{2\pi}\sum_{p=0}^{P}\sum_{i=0}^{N}(\zeta_\theta^k[p,i] - \zeta_\theta^l[p,i])^2} \tag{10}$$

A ROC analysis of the proposed weighted eigenspace Size Function produced an AUC measurement of 0.809 and 0.823 for the ISL data set and the Triesch data set respectively. The results of the experiment on the ISL data set show the weighted eigenspace Size Function has a total improvement of 7.4% when compared to the unmodified Size Function while results of the experiment on the Triesch data set show a total improvement of 6.7%. Fig. 11 shows the ROC graphs associated with the AUC measurements reported above.

5.2. Hu moments performance

Along with the eigenspace Size Function representation of a hand, Hu moments of a segmented binary hand image are used as a feature to describe the posture of a hand. To test the suitability of Hu moments as a hand posture feature, a similar experiment to the one described in Section 5.1 was carried out. Hu moments were extracted from the control and test images, described in Section 5.1. The distance between each of the control Hu moments and the test Hu moments were calculated from the total absolute difference between each augmented moment described in (11). Where the augmented moment is a metric implemented in the OpenCV library (Intel-Corporation, 2000) and is described in (12).

A ROC graph was then generated for the Hu moments using a method similar to the method described in Section 5.1. The AUC for the ROC graph was 0.796 and 0.852 for the ISL data set and the Triesch data set respectively.

$$D^{Hu}(H_k, H_l) = \sum_{i=1}^{7} |\Lambda_{H_k}(i) - \Lambda_{H_l}(i)| \tag{11}$$

$$\Lambda_{H_x}(i) = \frac{1}{sign(H_x(i)) \times \log(H_x(i))} \tag{12}$$

5.3. Combining Size Function and Hu moments

The ultimate goal of the system is to find the best possible classification scheme for recognizing hand postures. We have shown that both eigenspace Size Functions and Hu moments features can sufficiently discriminate between positive and negative examples with an AUC of 0.809 and 0.796 respectively for the ISL data set and an AUC of 0.823 and 0.852 respectively for the Triesch data set.

The classification system designed in this work uses a combination of different features, therefore a measure of the performance of the combination of eigenspace Size Functions and Hu moments was carried out.

The boolean expression defined in Eq. (13) is used to determine the combined classifier's output. A true result corresponds to the classifier predicting that hands H_k and H_l are of the same hand posture category.

$$\psi(H_k, H_l) = D^{Hu}(H_k, H_l) \leqslant T_{hu} \cap D(\zeta_k, \zeta_l) \leqslant T_{sf} \tag{13}$$

To examine the combined performance of the two measurements, an exhaustive grid search on the threshold values T_{hu} and T_{sf}, the threshold for the Hu moments measurement and the threshold for the Size Function measurements respectively, was carried out. For each $\{T_{hu}, T_{sf}\}$ a confusion matrix was computed on the output of Eq. (13) when applied to the set of hand images used in Sections 5.1 and 5.2. We then choose the confusion matrix with the best True:False ratio (TF_{Ratio}) described in Eqs. (14) and (15).

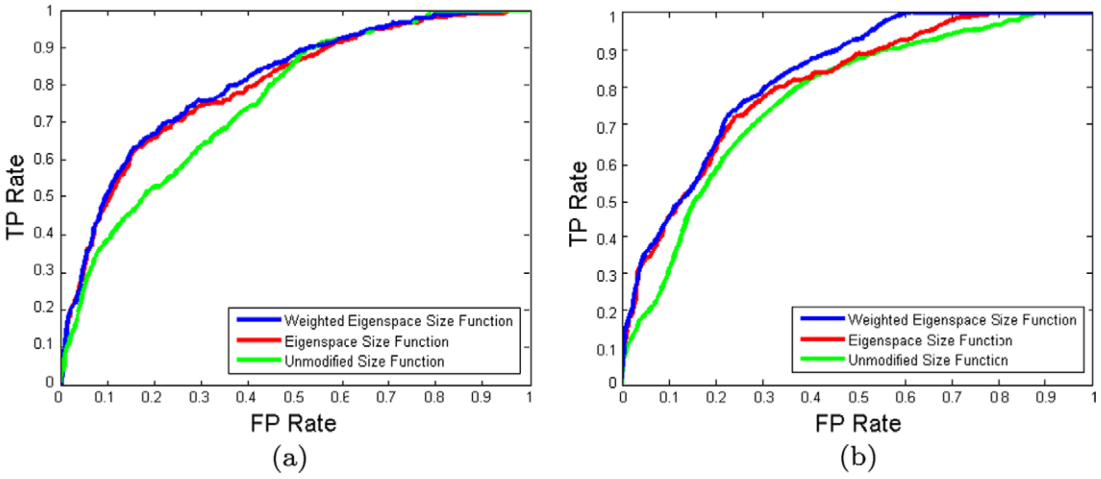
$$TP_{Rate} = \frac{TP}{TotalPositives}, \quad FP_{Rate} = \frac{FP}{TotalNegatives} \tag{14}$$

$$TF_{Ratio} = \frac{TP_{Rate} + (1 - FP_{Rate})}{2} \tag{15}$$

For both data sets, results of the grid search showed that the best True:False ratio computed was better than that of any points on the ROC graphs computed from the individual Size Function features and Hu moment features respectively. Fig. 12 shows the ROC graphs from the experiment and Table 1 details the best True:False ratios for the different features. It can be concluded from the results of this experiment that, based on the data from both data sets, the combination of the eigenspace Size Function and Hu moment representations provide complementary information about the shape of a hand.

5.4. Size Function parameters

To evaluate the best possible combination of the parameters (N, N_Θ, P), the size of the Size Function, the number of graph rotations and the number of principal components respectively, a ROC



(a)

(b)

Fig. 11. ROC graphs of weighted eigenspace Size Function, eigenspace Size Function and unmodified Size Function representations for (a) ISL data set and (b) Triesch data set.

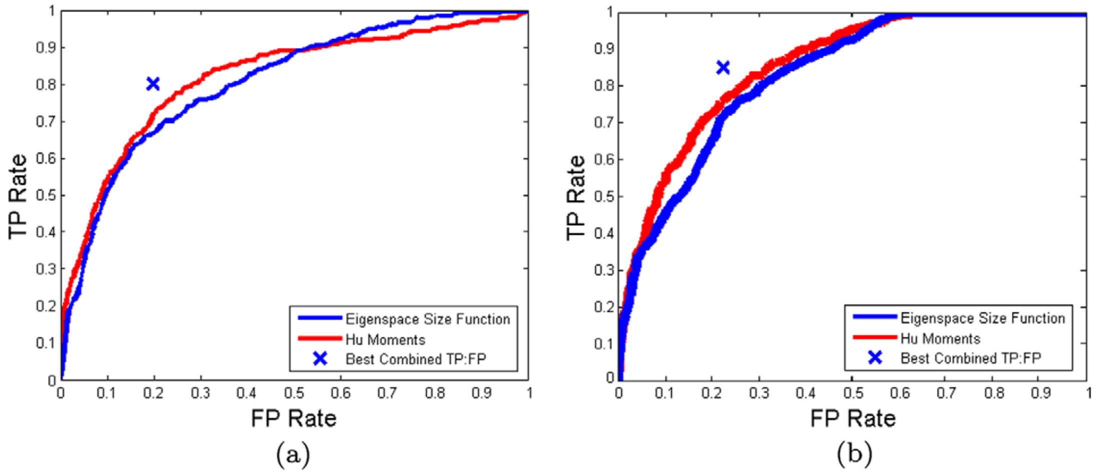


Fig. 12. ROC graph of combined features for (a) ISL data set and (b) Triesch data set.

Table 1
Best True:False ratios.

	ISL data set	Triesch data set
Best combined TF_{Ratio}	0.797	0.794
Best Size Function TF_{Ratio}	0.733	0.75
Best Hu moment TF_{Ratio}	0.764	0.77

Table 2
Parameter combination AUC.

(N, N_{Θ}, P)	ISL Data AUC	Triesch Data AUC	(N, N_{Θ}, P)	ISL Data AUC	Triesch Data AUC
(16,6,1)	**0.809**	**0.823**	(16,6,2)	0.794	0.811
(16,4,1)	0.791	0.822	(16,4,2)	0.783	0.804
(16,8,1)	0.799	0.814	(16,8,2)	0.786	0.792
(8,6,1)	0.782	0.801	(8,6,2)	0.779	0.787
(8,4,1)	0.780	0.796	(8,4,2)	0.769	0.783
(8,8,1)	0.791	0.807	(8,8,2)	0.773	0.792

analysis of the performance of the different parameters was carried out.

The same process described in Section 5.1 was carried out to evaluate the performance of the Size Function using different values of (N, N_{Θ}, P) where $(2 \leqslant N \leqslant 32)$, $(2 \leqslant N_{\Theta} \leqslant 16)$ and $(1 \leqslant P \leqslant N)$. It should be noted that as N increases, the margin between $G_{\varphi \leqslant y}$ and $G_{\varphi \leqslant x}$, the graphs used to calculate the values of the Size Function, decreases. As the margin decreases, smaller variations in the measuring function are identified as separate connected components. Therefore, as N increases, the Size Function becomes more sensitive to small changes in shape and noise. As N decreases, the Size Function become less sensitive to large shape variations and therefore performs poorly at discriminating between signs. Since the data collected in this work uses real sign data, typical segmentation noise can be present in the extracted contours. The existence of noise makes finding an optimal value of N an important goal as we must find an N which is not sensitive to noise but can discriminate between signs. During the performance evaluation of the different parameters (N, N_{Θ}, P), we also found that adjusting P varied the systems' sensitivity to noise. The more principal components used, the more sensitive the system became to noise. We concluded from this that the principal component held the main information about the hand posture, while the lower components held information about small variations in the contour shape.

Table 2 details different ROC AUCs for different parameter combinations computed from the ISL data set and the Triesch data set. Although experiments we carried out exhaustively on different parameter combinations, we present only the 12 best parameter combinations. Parameters within the bounds; $(2 \leqslant N \leqslant 32)$, $(2 \leqslant N_{\Theta} \leqslant 16)$ and $(1 \leqslant P \leqslant N)$, did not have a significant effect on the AUC with the lowest AUCs being 0.681 and 0.693 for our the ISL data set and the Triesch data set respectively. The parameter combination which produced the best AUC was $N = 16$, $N_{\Theta} = 6$ and $P = 1$. This was the parameter combination used in all tests described in Sections 5.1 and 5.3 above, and will be used for the posture recognition techniques which we will discuss in Section 6.

6. Recognition framework

In Section 5 we have shown that our eigenspace Size Functions and Hu moments possess strong hand shape discriminatory properties. We now describe our user independent framework for recognizing hand postures using these shape representations. A set of support vector machines (SVM) (Chang and Lin, 2001) are trained on data, using the discussed shape representations, extracted from labeled images. Given an unknown hand image, the relevant features are extracted and the SVMs use the data to estimate the most probable hand posture classification.

To classify an image z, containing an unknown hand posture, it must to be assigned to one of the C possible posture classes $(\alpha_1, \alpha_2, \ldots, \alpha_C)$. The proposed recognition framework uses two distinct measurement vectors to represent a hand posture. For each posture class α_c a set of two support vector machines $\{\mathrm{SVM}_c^{sf}, \mathrm{SVM}_c^{hu}\}$ is used to calculate $P(\alpha_c | I_z, \zeta_z)$, the probability that image posture z belongs to class α_c given measurement vectors I_z and ζ_z. Where I_z is the set of Hu moments and ζ_z is the weighted eigenspace Size Function extracted from image z.

6.1. Support vector machines

Support vector machines are a set of supervised learning methods used in classification and regression. A one against all SVM model is used in this work, and training of the SVM consists of providing the SVM with data for two classes. Data for each class consists of a set of n dimensional vectors. An RBF kernel is applied to the data and the SVM then attempts to construct a hyper plane in the n-dimensional space, attempting to maximize the margin between the two input classes.

The SVM type used in this work is C-SVM using a non-linear classifier by means of the kernel trick as proposed by Aizerman et al. (1964). The kernel used is a radial basis function (RBF) as defined by $k(x,x') = exp(-\gamma\|x - x'\|^2)$.

SVM is extended to obtain class probability estimates by computing pairwise class probabilities $r_{ij} \approx p(y = i|y = i \text{ } or \text{ } j, X)$ using Lin et al.'s (2007) improved implementation of Platt's method (Platt and Platt, 1999) which defines $r_{ij} \approx \frac{1}{1+e^{A\hat{f}+B}}$.

Where A and B are estimated by minimizing the negative log-likelihood function using known training data and their decision values $\hat{f}$. We then use the second approach proposed by fan Wu et al. (2004) to compute p_i from all r_{ij}'s by solving the optimization problem $\min_p \frac{1}{2}\sum_{i=1}^{C}\sum_{j:j\neq i}(r_{ij}p_i - r_{ij}p_j)^2$ subject to $\sum_{i=1}^{C}p_i = 1$, $p_i \geqslant 0, \forall_i$.

6.2. Training

Given a training set of hand images consisting of multiple labeled images of each hand posture class we train a set of SVM classifiers as follows:

Weighted eigenspace Size Function data and Hu moment data are extracted from the training set images to create the matrices $H_c = (I_{c1}, I_{c2}, \ldots, I_{cN})$ and $\Psi_c = (\zeta_{c1}, \zeta_{c2}, \ldots, \zeta_{cL})$ where L is the total number of training images recorded for each posture class α_c.

To train each SVM_c^{sf}, the matrix Ψ_c is used as the positive labeled training data and $\overline{\Psi_c} := (\Psi_j)_{j\neq c \cap j\epsilon\{1..C\}}$ is used as the negative labeled training data. Similarly, each SVM_c^{hu} is trained using H_c as the positive labeled data and $\overline{H_c} := (H_j)_{j\neq c \cap j\epsilon\{1..C\}}$ as the negative labeled data. The support vector machines SVM_c^{sf} and SVM_c^{hu} are then trained to maximize the hyperplane margin between their respective classes $(\Psi_c, \overline{\Psi_c})$ and $(H_c, \overline{H_c})$.

There are two parameters while using RBF kernels: C and γ. V-fold cross-validation was carried out to compute optimal values for C and γ.

6.3. Posture classification

To classify an unknown image z, each SVM_c^{sf} and SVM_c^{hu} will calculate $P(\alpha_c|\zeta_z)$ and $P(\alpha_c|I_z)$, the probability ζ_z and I_z belong to class α_c respectively using the method outlined in Section 6.1. Classifier weights, used to determine the overall probability, are calculated by Eq. (16), where cv_c^{sf} and cv_c^{hu} are the cross-validation accuracies achieved for each SVM_c^{sf} and SVM_c^{hu} respectively. A weighted combination of the probabilities is then calculated to generate the overall probability $P(\alpha_c|I_z, \zeta_z)$ according to Eq. (17).

$$\mu_c^{sf} = \frac{cv_c^{sf}}{cv_c^{sf} + cv_c^{hu}}, \quad \mu_c^{hu} = \frac{cv_c^{hu}}{cv_c^{sf} + cv_c^{hu}} \tag{16}$$

$$P(\alpha_c|I_z, \zeta_z) = (P(\alpha_c|\zeta_z) \times \mu_c^{sf}) + (P(\alpha_c|I_z) \times \mu_c^{hu}). \tag{17}$$

6.4. Experiments

We evaluate our recognition system using both data sets discussed in Section 4. For the ISL data set, we train the SVMs on 5520 hand posture images. The 5520 training images were comprised of data from 8 of the 16 subjects used. We then test our recognition framework on the remaining 5520 images.

For the Triesch data set we carry out two evaluation protocols (P1 and P2). We first perform an evaluation based on the same protocol as Triesch and von der Malsburg (2002). We train the SVMs on each of the 10 hand signs using data extracted from 3 of the 24 signers. The system is then tested on all hand signs from the remaining 21 subjects. The second evaluation protocol we perform is based on the work of Just et al. (2006), where eight signers are used for training a validation and the remaining 16 are used for testing.

Table 3
Classification AUC performance.

Letter	Training set Recognition	Test set Recognition	Hu moment Weighting μ_c^{hu}	Size function Weighting μ_c^{sf}
A	0.996	0.995	0.57	0.43
B	0.997	0.989	0.58	0.42
C	0.930	0.907	0.49	0.51
D	0.993	0.971	0.54	0.46
E	0.991	0.996	0.57	0.43
F	0.976	0.991	0.46	0.54
G	0.993	0.974	0.48	0.52
H	0.995	0.967	0.48	0.52
I	0.980	0.971	0.50	0.50
K	0.991	0.934	0.47	0.53
L	0.984	0.932	0.48	0.52
M	0.985	0.935	0.55	0.45
N	0.999	0.954	0.52	0.48
O	0.999	0.930	0.54	0.46
P	0.999	1.000	0.58	0.42
Q	0.962	0.989	0.45	0.55
R	0.992	0.960	0.51	0.49
S	1.000	1.000	0.56	0.44
T	1.000	1.000	0.56	0.44
U	0.976	0.993	0.52	0.48
V	1.000	1.000	0.49	0.51
W	1.000	1.000	0.45	0.55
Y	1.000	1.000	0.50	0.50
Mean	**0.989**	**0.973**	**0.52**	**0.48**

Table 4
Recognition performance.

	# Training	# Test	Number	Correct	Percentage	AUC
Our method P1	3	21	418	356	85.1	0.827
Triesch et al.	3	21	418	392	95.2	–
Our method P2	8	16	320	294	91.8	0.935
Just et al.	8	16	–	–	89.9	–

We carry out the following tests for the ISL data set and both of the Triesch evaluation protocols: For each image $z_i\epsilon\{1\ldots L\}$, where L is the total number of images in the test set, the classification probabilities $\Delta_z := [p(I_z, \zeta_z|\alpha_1), \ldots, p(I_z, \zeta_z|\alpha_C)]$ is calculated. To test the performance of the system a ROC analysis was carried out on the classification of the test images. For each posture class $n\epsilon\{1\ldots C\}$ a confusion matrix was calculated. To generate multiple points on the ROC graph, a confusion matrix is calculated from different threshold values T $(0 \leqslant T < 1)$. Table 3 details the AUC of the ROC graph generated from the classification of both the training data and the test data of the ISL data set. It can be seen from the AUC measures, generated from both the training data and test data, that each classifier performs well at classification of each hand shape.

The results also show that the overall recognition rate of the test set is within only 0.016 less accurate than the overall classification of the training set. Since the subjects used to record the test

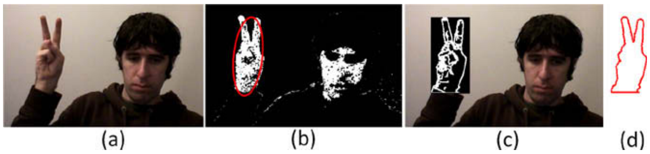


Fig. 13. Feature extraction for continuous recognition experiment: (a) original image, (b) skin color segmentation using mean shift algorithm, (c) edge detected hand region and (d) extracted contour.

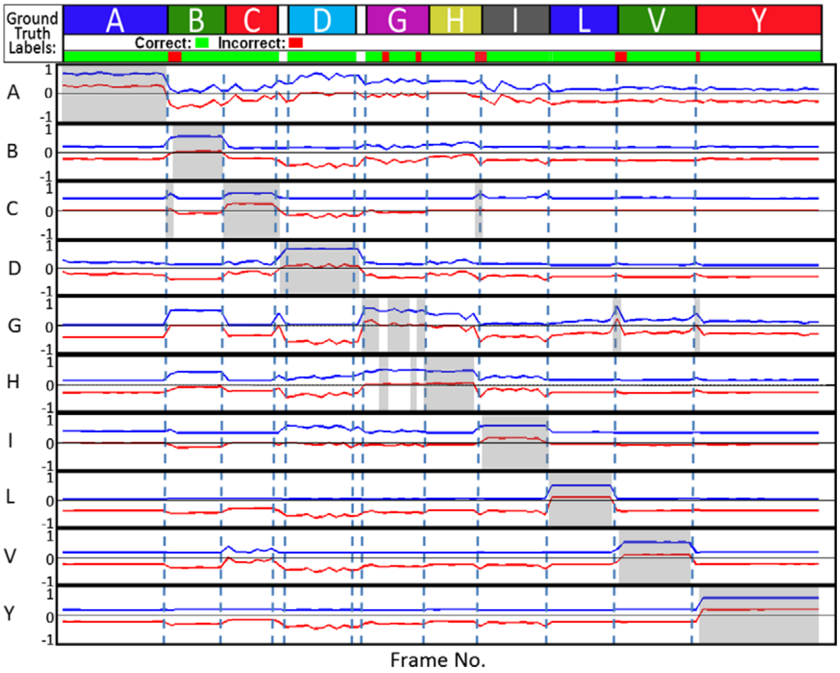


Fig. 14. Recognition probabilities for continuous video stream. For each frame, the classifier which outputs the maximum likelihood is denoted as the grey area. For each classifier output we denote the blue plot as the likelihood output of that classifier, while the red plot denotes the difference between the classifier output and the likelihood of the classifier with the second highest likelihood (thus, for the red plot, values above 0 denotes the maximum likelihood, values equal to 0 denotes the second highest likelihood and values below 0 denotes all other likelihoods).

data were different to the subjects used to record the training data, it can be concluded from these results that the proposed hand posture recognition framework performs well at recognizing hand postures independent of the subjects performing the postures. Table 3 also shows the classifier weights, calculated during the cross-validation stage of the training and used in the recognition experiments. These weights give an indication of which classifier performs best at classifying the different hand shapes. As can be seen from Fig. 7, the hand shape for "A" and "E" are quite similar, as is the hand shapes for "F" and "G". From the classifier weights, we see that the Hu moments have a greater discrimination when classifying shapes with little contour variation, such as the signs for "A" and "E", whereas the eigenspace Size Function has a greater discrimination when classifying signs with a larger variation in the contour shape, such as the signs for "F" and "G".

Results of the evaluation on the Triesch data set, detailed in Table 4, show that the elastic graph matching algorithm of Triesch and von der Malsburg (2002) achieves a higher recognition rate than our method. The small amount of training data would seem to contribute to the lower recognition rate of our method. Results show that as the training set grows, from three subjects to eight subjects, the recognition rate increases by 6.4%. Our method also shows a better recognition rate than the work of Just et al. (2006), when trained on data from eight subjects.

Although the elastic graph matching algorithm achieves a higher recognition rate, it has a high computational complexity requiring several seconds to analyze a single image. In comparison, evaluations carried out on our method shows that the average computation time, including feature extractions, feature analysis and feature classification, was 60 ms. Performance measures were performed on a computer with a 2.16 GHz Intel Core 2 CPU.

6.5. Continuous recognition

To illustrate the robustness of our system when classifying signs from a video stream, we show results of a video based recognition experiment. We utilize a live gesture feedback application, proposed by Kelly et al. (2008), along with the SVMs, trained on the Triesch data set using protocol 2, to classify signs from a continuous video sequence. In the video sequence, an unseen user performs each of the 10 signs one after each other. In the first frame of the video, the hand position is manually selected and a skin color histogram of the hand is recorded. For each successive frame, the mean shift algorithm is used to locate the hand region (Comaniciu et al., 2000). We then perform Canny edge detection on the hand region, followed by a dilation operation, and the contour is then extracted from the edge detected image using a border following algorithm (Suzuki and Be, 1985) (see Fig. 13). Fig. 14 illustrates the results of the video based posture classification, where the graph depicts the probability for each hand posture class for each image frame. It can be seen from the ground truth labels on the graph that our system performs well at classifying postures in each image frame.

7. Conclusion

The main contribution of this work is that we propose a user independent hand shape feature, a weighted eigenspace Size Function, which we show to be a strong improvement over the original Size Function feature. We also show that our method performs well compared to other user independent hand posture recognition systems.

Our eigenspace Size Function performed significantly better at discriminating between different hand postures than the unmodified Size Function when tested on two different user independent hand posture data sets. An increase in performance of 7.4% and 6.7% was shown for our weighted eigenspace Size Function when compared to the unmodified Size Function using a simple Euclidian distance classifier. We proposed a user independent, SVM based, recognition framework using a combination of our weight eigenspace Size Function and Hu moments. Results of a user independent evaluation of the recognition framework showed our system had a ROC AUC of 0.973 and 0.935 when tested on the ISL data set and the Treisch data set respectively. Future work will involve integrating our proposed hand posture recognition framework into a system which can recognize full sign language sentences by incorporating spatiotemporal and non-manual information into the recognition process.

References

Aizerman, A., Braverman, E.M., Rozoner, L.I., 1964. Theoretical foundations of the potential function method in pattern recognition learning. Automation Remote Control 25, 821–837.

Al-Jarrah, O., Halawani, A., 2001. Recognition of gestures in Arabic sign language using neuro-fuzzy systems. Artif. Intell. 133 (1-2), 117–138. ISSN 0004-3702.

Bauer, B., Hienz, H., 2000. Relevant features for video-based continuous sign language recognition. In: FG '00. IEEE Computer Society, Washington, DC, USA, p. 440. ISBN: 0-7695-0580-5.

Chang, C.-C., Lin, C.-J., 2001. LIBSVM: A library for support vector machines. Available from: %3chttp://www.csie.ntu.edu.tw/cjlin/libsvm%3e.

Comaniciu, D., Ramesh, V., Meer, P., 2000. Real-time tracking of non-rigid objects using mean shift. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition 2, 142–149. doi:10.1109/CVPR.2000.854761.

Cooper, H., Bowden, R., 2007. Large Lexicon detection of sign language. In: CVHCI07, pp. 88–97.

Cui, Y., Weng, J., 2000. Appearance-based hand sign recognition from intensity image sequences. CVIU 78 (2), 157–176.

Deng, J.-W., Tsui, H., 2002. A novel two-layer PCA/MDA scheme for hand posture recognition. In: Proc. Pattern Recognition, Vol. 1, pp. 283–286. doi:10.1109/ICPR.2002.1044688, ISSN: 1051-4651.

fan Wu, T., Jen Lin, C., Weng, R.C., 2004. Probability estimates for multi-class classification by pairwise coupling. J. Machine Learn. Res. 5, 975–1005.

Farhadi, A., Forsyth, D., White, R., 2007. Transfer learning in sign language. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR '07, pp. 1–8.

Handouyahia, M., Ziou, D., Wang, S., 1999. Sign language recognition using moment-based size functions. In: Proc. Intl. Conf. on Vision Interface, pp. 210–216.

Holden, E.-J., Owens, R., 2003. Recognising moving hand shapes. Proc. Image Analysis and Processing, pp. 14–19. doi: 10.1109/ICIAP.2003.1234018.

Holden, E.-J., Lee, G., Owens, R., 2005. Automatic recognition of colloquial australian sign language. In: IEEE Workshop on Motion and Video Computing, WACV/MOTIONS '05, Vol. 2, pp. 183–188. doi: 10.1109/ACVMOT.2005.30.

Hu, M.-K., 1962. Visual pattern recognition by moment invariants, information theory. IEEE Trans. 8 (2), 179–187. ISSN 0018-9448.

Huang, C.-L., Huang, W.-Y., 1998. Sign language recognition using model-based tracking and a 3D hopfield neural network. Mach. Vision Appl. 10 (5-6), 292–307. http://dx.doi.org/10.1007/s00138005008, ISSN 0932-8092.

Imagawa, I., Matsuo, H., Taniguchi, R., Arita, D., Lu, S., Igi, S., 2000. Recognition of local features for camera-based sign language recognition system. In: Proc. Pattern Recognition, Vol. 4 , pp. 849–853. doi: 10.1109/ICPR.2000.903050.

Intel-Corporation, 2000. Open Source Computer Vision Library: Reference Manual.

Just, A., Rodriguez, Y., Marcel, S., 2006. Hand posture classification and recognition using the modified census transform. In: 7th Internat. Conf. on Automatic Face and Gesture Recognition, FGR, pp. 351–356. doi: 10.1109/FGR.2006.62.

Kadir, T., Bowden, R., Ong, E.J., Zisserman, A., 2004. Minimal training, large lexicon unconstrained sign language recognition. In: BMVC.

Kelly, D., McDonald, J., Markham, C., 2008. A system for teaching sign language using live gesture feedback. In: 8th IEEE Internat. Conf. on FG '08, pp. 1–2.

Licsr, A., Szirnyi, T., 2005. User-adaptive hand gesture recognition system with interactive training. Image Vision Comput. 23 (12), 1102–1114. doi:10.1016/j.imavis.2005.07.01. ISSN 0262-8856.

Lin, H.-T., Lin, C.-J., Weng, R.C., 2007. A note on Platt's probabilistic outputs for support vector machines. Mach. Learn. 68 (3), 267–276. ISSN 0885-6125.

Patwardhan, K.S., Roy, S.D., 2007. Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive EigenTracker. Pattern Recognition Lett. 28 (3), 329–334. ISSN 0167-8655.

Pavlovic, V., Sharma, R., Huang, T., 1997. Visual interpretation of hand gestures for human–computer interaction: A review. IEEE PAMI 19 (7), 677–695. doi:10.1109/34.59822. ISSN 0162-8828.

Platt, J.C., Platt, J.C., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. MIT Press.

Starner, T., Pentland, A., Weaver, J., 1998. Real-time american sign language recognition using desk and wearable computer based video. IEEE PAMI 20 (12), 1371–1375. http://dx.doi.org/10.1109/34.73581, ISSN: 0162-8828.

Stokoe, J., William, C., 2005. Sign language structure: An outline of the visual communication systems of the american deaf. J. Deaf Stud. Deaf Educ. 10 (1), 3–37. ISSN: 1081-4159.

Suzuki, S., Be, K., 1985. Topological structural analysis of digitized binary images by border following. Comput. Vision Graphics Image Process. 30 (1), 32–46.

Tanibata, N., Shimada, N., Shirai, Y., 2002. Extraction of hand features for recognition of sign language words. In: Internat. Conf. on Vision Interface, pp. 391–398.

Triesch, J., von der Malsburg, C., 2002. Classification of hand postures against complex backgrounds using elastic graph matching. Image Vision Comput. 20 (13–14), 937–943.

Uras, C., Verri, A., 1995. Sign language recognition: An application of the theory of size functions. In: 6th British Machine Vision Conference, pp. 711–720.

Wu, Y., Huang, T.S., Mathews, N., 1999. Vision-based gesture recognition: A review. In: Lecture Notes in Computer Science. Springer, pp. 103–115.

Yang, H.D., Sclaroff, S., Lee, S.W., 2009. Sign language spotting with a threshold model based on conditional random fields. In: IEEE PAMI 99 (1).