WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia FUZZ IEEE

Thai Sign Language Translation System Using Upright Speed-Up Robust Feature and C-Means Clustering

Phonkrit Chanda
Computer Engineering Department,
Faculty of Engineering,
Chiang Mai University,
Chiang Mai, Thailand
mcpe131ta@gmail.com

Sansanee Auephanwiriyakul, *Senior Member, IEEE*
Computer Engineering Department,
Faculty of Engineering,
Biomedical Engineering Center
Chiang Mai University,
Chiang Mai, Thailand
sansanee@ieee.org

Nipon Theera-Umpon, *Senior Member, IEEE*
Electrical Engineering Department,
Faculty of Engineering,
Biomedical Engineering Center
Chiang Mai University,
Chiang Mai, Thailand
nipon@ieee.org

***Abstract*—Sign language is an important communication tool for the deaf. In this paper, we build a dynamic hand gesture translation system with video caption without prior hand region segmentation. In particular, we utilize the upright speed-up robust feature (U-SURF) and the fuzzy C-means (FCM) to find a matched word. We compared the result with that from string grammar hard C-means (sgHCM). This Thai sign language translation system was tested on 42 words. The total number of video sequences used in the experiment is 1470. The best correct classification rate on the signer-dependent blind test set using the FCM is approximately 47 to 73%, whereas that of signer-semi-independent set is around 30 to 40%. The best blind test result for the signer-independent experiment is around 24 to 30%. However, the correct classification rate from the sgHCM is higher than that from the FCM. The best result for the signer-dependent experiment is around 97 to 99%, whereas that of the signer-semi-independent is around 64 to 65%. The correct classification rate of the signer-independent experiment is around 53 to 54%.**

Keywords- Upright speed-up robust featute (U-SURF);Fuzzy C-Means (FCM); String Grammar Hard C-Means (sgHCM)

I. INTRODUCTION

Sign language is a solution to get over the communication difficulty between normal and deaf people. Several research works on sign language translation system have been done with the help of cyber-glove or position tracking [1]–[9].Although, these approaches provide good recognition rates, it is more natural to recognize hand gesture without using a special hardware or device. There are some other research works that propose techniques in recognizing static hand gestures with image caption or dynamic hand gestures with video caption [10]–[20]. However, these methods need to either automatically or manually segment hand regions before performing any recognition. Although, the work done by Phitakwinai *et. al.* does not need any segmentation, the correct classification rate is not very good, i.e., 60 to 80% for signer-dependent experiment and 20 to 40% for signer-independent experiment [20].

In this paper, we build a dynamic hand gesture translation system with video caption without prior hand region segmentation. In particular, we apply the upright speed-up robust features (U-SURF) [21] to match an image frame with that in the signature library. We then utilize the fuzzy C-means [22] to find a matched sign word. We also compare the result with the string grammar hard C-means [23,24]. We implement this system with 42 Thai sign language words (words signed with one hand only), i.e., “หนึ่ง(one)”, “สอง(two)”, “สาม(three)”, “สี่(four)”, “ห้า(five)”, “หก(six)”, “เจ็ด(seven)”, “แปด(eight)”, “เก้า(nine)”, “สิบ(ten)”, “จำนวน(amount)”, “หอม(aromatic)”, “ไก่(chicken)”, “ข้าวเกรียบ(rice crispy crackers)”,“น้ำปลา(fish sauce)”, “อาหาร(food)”, “ทอด(fry)”, “ให้(give)”, “ไป-กลับ(go-back)”, “ดี(good)”, “ปิ้ง(grill)”, “มี(have)”, “ถือ(hold)”, “หิว(hungry)”, “น้ำแข็ง(ice)”, “น้อย(little)”, “ไม่(no)”, “ข้าวเปลือก(paddy)”, “เน่า(putrid)”, “ข้าว(rice)”, “ข้าวสาร(wholegrain rice)”, “เค็ม(salty)”, “ข้าวเหนียว(sticky rice)”, “เหม็น(stink)”, “หวาน(sweet)”, “ชิม(taste)”, “อร่อย(tasty)”, “ใช้(use)”, “น้ำ(water)”, “อะไร(what)”, “ที่ไหน(where)” and “ใช่(yes)”.

II. SYSTEM OVERVIEW

Although, this system recognizes 42 words of the Thai sign language, there are actually 72 hand gestures (examples of hand gesture are shown in figure 1). We assign a number ranging from 1 to72 to represent each hand gesture randomly. We collected these hand gestures from subject no. 1 who wore a black shirt with long sleeves and stood in front of a background. Each video was recorded at different times of day for several days. However, people tend to make slightly difference in each time they perform the signs. We collected each Thai sign language several times to be our signature library in the form of video files. Representative frames (Rframes) of each video file in each Thai sign language were then selected manually. The region of interest in each frame (only hand part) was selected with the size of 190×190. We call each image in the signature library a keyframe. There are approximately 30 to 50 keyframes for each hand gesture, hence there are 2543 keyframes in total.

Figure 1. Examples of hand gesture numbers 15, 18, and 26 in the signature library.

After we collected all keyframes in the signature library, we computed keypoint descriptors of each keyframe using the upright speed-up robust feature (U-SURF) [21]. We kept all keypoint descriptors in the signature library database. In order to translate hand gesture, we need a system that grabs an image through a camera and analyzes it. The features are obtained from the luminance (Y) [25] extracted from each video sequence. The image size from a video sequence is 800×600 pixels. Each image sequence is decimated so that only 14 image frames are achieved for each video file. Then a matched keyframe can be found from the U-SURF. These matched keyframes represent numbers used as a vector in fuzzy C-means (FCM) [22]. These numbers are primitives in string grammar hard C-means (sgHCM) [23, 24].

Now let us briefly describe the U-SURF. The U-SURF is the speed-up robust feature (SURF) [21] without orientation assignment step. In our application, we do not need an orientation invariant feature because some hand gestures with different angles will represent different words. The examples of the hand gestures for the words “จำนวน (amount)” and “หอม (aromatic)” are shown in figure 2. These two words share similar hand gestures but they are in different angles.

(a)

(b)

Figure 2. Hand gestures for (a) “จำนวน (amount)” and (b) “หอม (aromatic)”.

Therefore, there are only 2 steps in the U-SURF, i.e., finding interesting points (keypoints) and computing keypoint descriptors. The U-SURF finds keypoints by first convolving the second-order derivatives of a Gaussian of a given scale σ with the input image to produce the Hessian matrix. The box filters shown in figure 3 are approximation of the second-order Gaussian (with σ= 1.2) partial derivatives in x-direction (D_{xx}), y-direction (D_{yy}), and xy-direction (D_{xy}), respectively.

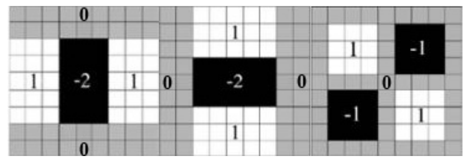
The integral image (I_{Σ}) defined on an input image (I),

$$I_{\Sigma} = \sum_{i=0}^{i\le x} \sum_{j=0}^{j\le y} I(x,y) \tag{1}$$

is used to speed up the calculation. To create a scale-space, a number of different filter sizes are applied to the input image as shown in figure 4. The Hessian determinant defined by

$$\det\left(H_{approx}\right) = D_{xx}D_{yy} - \left(0.9D_{xy}\right)^2 \tag{2}$$

is computed on each pixel on every plane in the scale-space.



(a) (b) (c)

Figure 3. An approximation of the second-order Gaussian partial derivatives in (a) x-direction, (b) y-direction, (c) xy-direction.

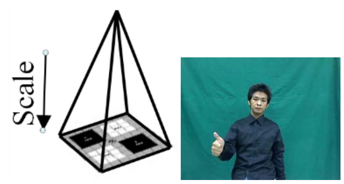


Figure 4. SURF approach scale-space with unchanged input image but varying the filter size.

A non-maximum suppression [26] in a 3×3×3 is applied to find a set of keypoints. The location $\mathbf{x}_0$ of each keypoint is then interpolated [26] by

$$\hat{\mathbf{x}} = \mathbf{x}_0 - \left(\frac{\partial^2 H}{\partial \mathbf{x}^2}\right)\frac{\partial H}{\partial \mathbf{x}}, \tag{3}$$

where $\mathbf{x}_0=[x_0,y_0,\sigma_0]$. The example of found keypoints is shown in figure 5. To generate a keypoint descriptor, a rectangular region with $20s$ ($s=\sigma$) length is set with the keypoint as a center of the region. This region is divided into 4×4 regular subregions. The Haar wavelet transformation with size $2s$ is calculated in each subregion. Then the vector (V_{sub}) is computed by

$$V_{sub} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right], \tag{4}$$

where dx and dy are the wavelet responses in x-direction and y-direction, respectively. Finally, these vectors are normalized to provide contrast invariance. Hence, a keypoint descriptor is a 64-dimensional feature vector.

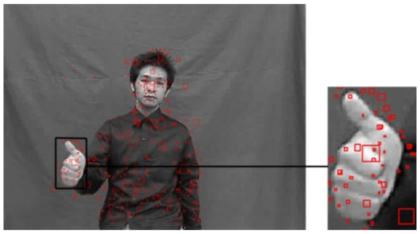
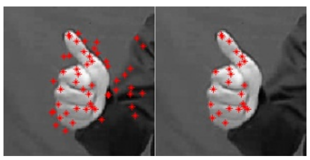


Figure 5. Example of found keypoints.

In our application, we need to find a match around the hand. Therefore, we screen out any keypoints with the original intensity more than 120 as shown in figure 6. This process is done for the keyframes in the signature library as well.



(a) (b)

Figure 6. Keypoints (a) from U-SURF and (b) after screeing out the points where their intensities are greater than 120.

Matching is obtained by selecting the keypoint descriptors in the signature library database that have Euclidean distance to the current keypoint descriptor less than a given threshold. However, if there is more than one keypoint descriptor in a test frame that is matched with the same signature library keypoint descriptor, we will keep only the pair that has smallest distance. The threshold in matching process is varied from 0.05, 0.1, 0.15, and 0.2. An example of matching keypoint process is shown in figure 7. Since the number of keyframes in each hand gesture may be different, the number of keypoints in each keyframe may also be different. We determine that the test image belongs to which hand gesture (symbol) using [20,27]

$$avg\_sym_i = \frac{\text{No. of matched keypoints from symbol } i}{\text{Total no. of keyframes from symbol } i} \tag{5}$$

$$\text{if } avg\_sym_j = \max_{i=1}^{c}\left(avg\_sym_i\right) \tag{6}$$

then test image will be assigned to symbol j.

After a symbol is assigned to each test image frame, we will get a vector of number representing that video sequence. Now we are ready to translate this video sequence into word using the FCM. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of vectors, where each vector is a p-dimensional. In our case, $p = 14$.

The update equation for FCM is as follows [22]

$$u_{ij} = 1 \Big/ \sum_{k=1}^{C}\left(d^2\left(\mathbf{x}_j, \mathbf{c}_i\right) \Big/ d^2\left(\mathbf{x}_j, \mathbf{c}_k\right)\right)^{1/(m-1)} \tag{7}$$

$$\mathbf{c}_i = \sum_{j=1}^{N}\left(u_{ij}\right)^m \mathbf{x}_j \Big/ \sum_{j=1}^{N}\left(u_{ij}\right)^m \tag{8}$$

In the above equation, u_{ij} is the membership value of vector $\mathbf{x}_j$ belonging to cluster i, $\mathbf{c}_i$, is the center of cluster i, and $m \in [1, \infty)$ is the fuzzifier. The algorithm for FCM is:

Fix number of clusters C;
Fix the fuzzifier m;
Initialize centers
Do {
  Update memberships using equation (7)
  Update centers using equation (8)
} **Until** (centers stabilize)

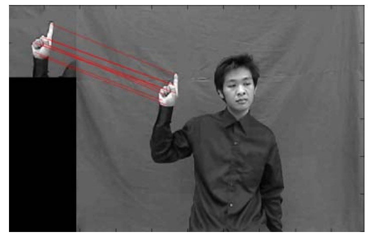


Figure 7. Example of matching keypoint process.

To make more sense, after each prototype is computed, we select the closest vector in the cluster to be our prototype for that cluster.

We also compare our result with that from sgHCM. Let us briefly describe the sgHCM algorithm. The algorithm of the sgHCM [23, 24] is

Store n unlabeled finite strings $X = \{\alpha_k; k = 1,\ldots,n\}$

Initial string prototypes for all C classes

Do {

Compute Levenshtein distance (D_{ik}) [1] (a smallest number of transformations needed to derive one string from the other) between input string i and cluster prototype k

Update membership using

$$u_{ik} = \begin{cases} 1; & D_{ik} < D_{ij} \ \ for\ j \neq i \\ 0; & otherwise \end{cases}$$

Update center string of each cluster k (α_q^k) where

$$q = \arg\min_{1 \le j \le n_k}\left\{ c_j^k = \sum_{s=1}^{n_k}\left(\frac{D_{js}}{n_k}\right)\right\}$$

and n_k is number of strings classified in cluster k

}**Until** (center strings stabilize)

We can use the calculated prototypes directly, since the sgHCM gives prototype in terms of a sequence of primitives already.

Again, the length of a string representing a test image sequence is 14. To match a test image sequence in FCM, the test vector generated from U-SURF will be assigned to a word that the closest prototype belongs to according to the Euclidean distance. In sgHCM, the test string is assigned to a word the closest prototype belongs to according to the Levenshtein distance.

III. Experimental Results

The testing video data sets were recorded at different time of day for several days from 4 subjects. The subject no. 1 is the same person who was recorded for the keyframes in the signature library. In testing video, subjects were asked to wear a black shirt with long sleeves and stood in front of a background while they were performing Thai sign language. We only utilize the data set from subject 1 to 3 to be our training data set. Each subject was asked to perform each word 10 times, 5 for training and the other 5 for testing. We call the training sets from subject no. 1, 2, and 3, the data sets 1a, 2a, and 3a, respectively. We also call the test sets from subject no. 1, 2, and 3, the data sets 1b, 2b, and 3b, respectively. For subject no. 4, we asked him to perform each word 5 times and we utilized this data set as our blind signer-independent. We call this data set as the data set 4. Each data set contains 210 video sequences. In both FCM and sgHCM experiments, we cluster each word separately and set $C = 2$. Hence, there are 84 prototypes in total. In the FCM experiment, we set $m = 2$.

The correct classification rate using the data set 1a as the training set in FCM is shown in table 1. We found that the best blind test result for the signer-dependent (same subjects in train set) experiment (1b) is 72.9% at 0.1 U-SURF threshold. The blind test results for the signer-independent (different subjects in train set) experiment (2b and 3b) are 23.65%, 34.10%, 17.90% and 31% at 0.05, 0.1, 0.15 and 0.2 U-SURF thresholds, respectively. For the blind signer-independent experiment (4), the best result is at 29.6% at 0.1 U-SURF threshold. Table 2 shows the correct classification rate when using 1a and 2a as a training set. We found that the best blind test result for signer-dependent experiment (1b) is 51.5% at 0.1 U-SURF threshold, whereas, that for signer-semi-independent (same subjects in classifier train set but different subjects from the one in the signature library) experiment (2b) is 39.6% at the same threshold. The best blind test result for the signer-independent experiment (3b) is 27.2% at 0.1 U-SURF threshold. At the same U-SURF threshold, the result from the blind signer-independent experiment (4) is 25.3% that is also the best result. Finally, we utilize 1a, 2a, and 3a as our training set and the correct classification rate is shown in table 3. We found that the best blind test result for signer-dependent experiment is 49.1% at 0.1 U-SURF threshold, the blind-test results for signer-semi-independent experiment (2b, 3b) are 17.65%, 30.55%, 20.3% and 30.05% at 0.05, 0.1, 0.15 and 0.2 U-SURF thresholds, respectively. The best result for the blind signer-independent experiment (4) is 24.4% at 0.1 U-SURF threshold. We can see that the classification rate for signer-dependent experiment is always the highest, although, it decreases when we include more subjects into the training data set. The blind signer-independent experiment gives the correct classification rate approximately 24 to 30 %. Again, the classification rate is also decreasing when there are more subjects in the training data set. This might be because different persons might have different ways of making the same sign. One might suspect that this type of data set is not a good data set for FCM clustering method. Because this data set might have some characteristics similar to the time-series data set, we compare the results with that from the sgHCM.

TABLE I. Classification rate from FCM using data set 1a as training set.

Data set	U-SURF threshold			
	0.05	0.1	0.15	0.2
1a	79.1	85.8	54.8	77.7
1b	62.0	72.9	29.6	61.0
2a	25.3	36.7	20.5	29.6
2b	25.8	34.3	20.0	30.5
3a	21.0	32.9	17.2	30.0
3b	21.5	33.9	15.8	31.5
4	16.7	29.6	19.2	27.2

Tables 4, 5 and 6 show the results from the sgHCM with data sets 1a; 1a and 2a; and 1a, 2a, and 3a. We found out that training with 1 subject, the best blind test classification rate for signer-dependent experiment (1b) is 99.1% at 0.05 U-SURF threshold. The blind test classification rate for signer-independent experiment (2b and 3b) are 41.5%, 65.35%, 39.8%, and 53.4% for 0.05, 0.1, 0.15, and 0.2 U-SURF thresholds, respectively. The best result of the blind signer-independent experiment (4) is 52.9% at 0.1 U-SURF threshold. When we train the system with 2 subjects, the best blind test result for signer-dependent experiment (1b) is 98.6% at 0.05 U-SURF threshold, whereas that for signer-semi-independent experiment (2b) is 63.4% at 0.1 U-SURF threshold. The best blind test classification rate for the signer-independent experiment (3b) is 67.7% at 0.1 U-SURF threshold, while that of the blind signer-independent experiment (4) is 53.9% at 0.1 U-SURF threshold.

TABLE II. Classification rate from FCM using data sets 1a and 2a as training set.

Data set	U-SURF threshold			
	0.05	0.1	0.15	0.2
1a	31.5	51.5	32.9	39.1
1b	28.6	46.7	24.3	36.2
2a	32.9	50.0	35.3	43.4
2b	21.0	39.6	22.0	33.4
3a	16.7	27.2	16.7	23.9
3b	16.7	27.2	18.6	24.8
4	18.6	25.3	22.0	25.3

The best blind test correct classification rate for signer-dependent experiment (1b) when we train the system with 3 subjects is 97.2% at 0.05 U-SURF threshold. For the blind test for signer-semi-independent experiment (2b and 3b), the correct classification rates are 45.05%, 65.05%, 40.3%, and 50.55% for 0.05, 0.1, 0.15, and 0.2 U-SURF thresholds, respectively. The best result of the blind signer-independent experiment (4) is 54.3% at 0.1 U-SURF threshold. We can see that the result from sgHCM is better than that from FCM. The correct classification rate of the signer-dependent experiment from sgHCM is around 97 to 99%, whereas that of the signer-semi-independent experiment is around 64 to 65%. The correct classification rate of the signer-independent experiment is around 53 to 54%. When we include more subjects into the training data set, the result of the signer-dependent experiment becomes worse a little bit. However, the performance of the signer-independent experiment increases when there are more subjects in the training data set.

TABLE III. CLASSIFICATION RATE FROM FCM USING DATA SETS 1A, 2A AND 3A AS TRAINING SET.

Data set	U-SURF threshold			
	0.05	0.1	0.15	0.2
1a	36.2	49.1	31.0	41.0
1b	29.6	46.7	21.0	34.8
2a	22.9	33.9	27.7	34.8
2b	19.1	31.5	22.9	31.0
3a	30.0	37.7	25.8	33.4
3b	16.2	29.6	17.7	29.1
4	15.3	24.4	14.5	26.4

TABLE IV. CLASSIFICATION RATE FROM SGHCM USING DATA SET 1A AS TRAINING SET.

Data set	U-SURF threshold			
	0.05	0.1	0.15	0.2
1a	99.1	97.2	45.8	86.2
1b	97.7	97.7	47.7	84.8
2a	36.2	62.4	47.7	48.6
2b	41.0	63.0	49.6	54.8
3a	36.7	65.8	28.1	51.5
3b	42.0	67.7	30.0	52.0
4	30.5	52.9	50.0	45.3

TABLE V. CLASSIFICATION RATE FROM SGHCM USING DATA SETS 1A AND 2A AS TRAINING SET.

Data set	U-SURF threshold			
	0.05	0.1	0.15	0.2
1a	98.6	92.0	44.8	83.4
1b	96.2	91.5	44.3	80.5
2a	50.0	66.7	51.0	50.0
2b	49.1	63.4	55.3	54.3
3a	38.1	63.9	25.3	50.0
3b	44.8	67.2	28.6	50.0
4	32.4	53.9	52.0	43.4

TABLE VI. CLASSIFICATION RATE FROM SGHCM USING DATA SETS 1A, 2A AND 3A AS TRAINING SET.

Data set	U-SURF threshold			
	0.05	0.1	0.15	0.2
1a	97.2	95.8	43.9	81.5
1b	96.2	95.3	44.8	82.4
2a	47.7	69.1	52.9	50.5
2b	47.7	64.8	55.3	52.0
3a	48.1	69.1	28.1	49.1
3b	42.4	65.3	25.3	49.1
4	31.0	54.3	52.4	47.2

IV. CONCLUSION

In this work, we build an automatic Thai sign language translation system for 42 words using the upright speed-up robust feature (U-SURF) and the fuzzy C-means (FCM) algorithm. We compare the result with the string grammar hard C-means (sgHCM) as well. In particular, we find a matched hand gesture from keypoint descriptors in the signature library database collected from 1 subject at a different time of day for several days using the U-SURF. Then we utilize the FCM or sgHCM to find the matched word from the training sequence data set from subject(s). From the results, we found that the best correct classification rate for signer-dependent experiment is approximately 47 to 73% whereas that for signer-semi-independent experiment is around 30 to 40%. The best correct classification rate for the signer-independent experiment is around 24 to 30%. However, the result from the sgHCM is better than that from FCM because of the property of the data set. The best result for the signer-dependent experiment is around 97 to 99%, whereas that for the signer-semi-independent experiment is around 64 to 65%. The correct classification rate for the signer-independent experiment is around 53 to 54%. The result of the signer-independent experiment is increasing when there are more subjects in the training data set.

ACKNOWLEDGEMENTS

The authors would like to thank the Graduate School, Chiang Mai University for partial support of this research.

REFERENCE

[1] Kramer, J. and Leifer, L. (1989), "The talking glove: a speaking aid for nonvocal deaf and deaf-bilid individuals", Proc. of RESNA 12th Annual Conference, pp. 471-472.

[2] Fels, S. S. and Hinton, G. E. (1993), "Glove-talk: a neural network interface between a data-glove and a speech synthesizer", IEEE Trans. On Neural Networks, Vol. 4, No. 1, pp. 2-8.

[3] Su, M. C, Jean, W. F. and Chang, H. T. (1996), "A static hand gesture recognition system using a composite neural network", Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, pp. 786-792.

[4] Chen, L. Y., Mizuno, N., Fujimoto, H., and Fujimoto, M. (2002), "Hand shape recognition using the bending element of fingers in sign language", Transactions of the Japan Society of Mechanical Engineers, Part C, Vol. 68, Issue 12, pp. 3689-3696.

[5] Wu, J. Q., Gao, W., Chen, X. L., and Ma, J. Y. (2000) "Hierarchical DGMM recognizer for Chinese sign language recognition", Journal of Software, Vol. 11, Issue 11, pp. 1430-1439.

[6] Gao, W., Chen, X. L., Ma, J. Y. and Wang, Z. Q. (2000), "Building language communication between deaf people and hearing society through multimodal human-computer interface", Chinese Journal of Computers, Vol. 23, Issue 12, pp. 1253-1260.

[7] Wu, J. Q., Gao, W., Song, Y., Liu, W. and Pang, B. (1998), "Simple sign language recognition system based on data glove", International Conference on Signal Processing Proceedings (ICSP), Vol. 2, pp. 1257-1260.

[8] Gao, W., Fang, G., Zhao, D. and Chen, Y. (2004), "A Chinese sign language recognition system based on SOFM/SRN/HMM", Pattern Recognition, Vol. 37, Issue 12, pp. 2389 – 2402.

[9] Akmeliawati R., OoiM.P.-L., Kuang Y.C. (2007), "Real-time Malaysian sign language translation using colour segmentation and neural network", Instrumentation and Measurement Technology Conference, pp1-6.

[10] Min, B. W., Yoon, H. S., Soh, J., Yang, Y. M. and Ejima, T. (1997), "Hand gesture recognition using Hidden Markov Models", IEEE International Conference on Systems, Man, and Cybernatics (Computational Cybernatics and Simulation), pp. 4232-4235.

[11] Huang, C. L., and Huang, W. Y. (1998), "Sign language recognition using model-based tracking and a 3D Hopfield neural network", Machine Vision and Applications, Vol. 10, Issue 5-6, pp. 292-307

[12] Kobayashi, T. and Haruyama, S. (1997), "Partly-hidden Markov model and its application to gesture recognition", IEEE International

Conference on Acoustics, Speech and Signal Processing – Proceedings (ICASSP), Vol. 4, pp. 3081-3084

[13] Chen, F. S., Fu, C. M. and Huang, C. L. (2003), "Hand gesture recognition using a real-time tracking method and hidden Markov models", Image and Vision Computing, Vol. 21, Issue 8, pp. 745-758

[14] Alon, J., Athisos, V., Yuan, Q., and Sclaroff, S. (2005), "Simultaneous localization and recognition of dynamic hand gestures", Proc. IEEE Workshop on Motion and Video Computing, pp. 254-260.

[15] Lee, Y.-H. and Tsai, C.-Y. (2009), " Taiwan sign language (TSL) recognition based on 3D data and neural networks", Expert Systems with Applications, Vol.36, Issue 2, Part 1, pp.1123 – 1128.

[16] Al-Rouson, M., Assaleh, K., and Tala'a, A. (2009), "Video-based signer-independent Arabic sign language recognition using hidden Markov models", Applied Soft Computing, Vol 9, pp. 990 -999.

[17] Sriboonruang, Y., Kumhom, P., and Chamnongthai, K. (2004), "Hand posture classification using wavelet moment invariant", IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, pp. 78 – 82.

[18] Quan Y. (2010), "Chinese sign language recognition based on video sequence appearance modeling" Industrial Electronics and Applications, pp.1537-1542.

[19] Auephanwiriyakul S. and Chaisatian P. (2004), "Static hand gesture translation using string grammar hard c-means", The Fifth International Conference on Intelligent Technologies, Houston, Texas, USA.

[20] Phitakwinai S., Auephanwiriyakul S., Theera-Umpon N. (2008), "Thai sign language translation using fuzzy c-means and scale invariant feature transform" Lecture Notes in Computer Science, Vol 5073, pp. 1107 – 1119.

[21] Bay H., Ess A., Tuytelaars T., Van Gool L. (2008), "Speeded-up robust features (SURF)" Computer Vision and Image Understanding. Vol 110, Issue 3, pp. 346-359.

[22] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", New York, Plenum Press, 1981.

[23] J. C. Bezdek, J. Keller, R. Krishnapuram and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic Publishers, USA., 1999

[24] . K.S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice Hall, Englewood Cliffs, NJ.

[25] Jiang H., Helal A., Elmagarmid A.K., Joshi A. (1996), "Scene change detection techniques for video database systems", ACM Multimedia Information Systems, 1996.

[26] Brown M., Lowe D. (2002), "Invariant features from interest point groups" British Machine Vision Conference, pp. 656-665.

[27] Suttapak W., Auephanwiriyakul S., and Theera-Umpon N. (2010), "Incorporating SIFT with hard c-means algorithm", International Conference on Computer and Automation Engineering, pp. 437-441.