TRANSLATION OF SIGN LANGUAGE USING GENERIC FOURIER DESCRIPTOR AND NEAREST NEIGHBOUR

Abidatul Izzah [1] and Nanik Suciati [2]

[1,2] Informatics Department, Institute Teknology Sepuluh Nopember, Indonesia

ABSTRACT

Sign languages are used all over the world as a primary means of communication by deaf people. Sign language translation is a promising application for vision-based gesture recognition methods. Therefore, it is need such a tool that can translate sign language directly. This paper aims to create a system that can translate static sign language into textual form automatically based on computer vision. The method contains three phases, i.e. segmentation, feature extraction, and recognition. We used Generic Fourier Descriptor (GFD) as feature extraction method and K-Nearest Neighbour (KNN) as classification approach to recognize the signs. The system was applied to recognize each 120 stored images in database and 120 images which is captured real time by webcam. We also translated 5 words in video sequences. The experiment revealed that the system can recognized the signs with about 86 % accuracy for stored images in database and 69 % for testing data which is captured real time by webcam.

KEYWORDS

Sign Language, Generic Fourier Descriptor, K-Nearest Neighbour, Image Processing

1. INTRODUCTION

The sign language is the fundamental communication between deaf people. Hand gestures language and dynamic movement plays an important role in deaf learning and their communication. The dynamic movement from the deaf people in sign language can means such as greeting hello or good-bye. Moreover, we can divide gestures into static gestures and dynamic gestures. If static gesture only use hand configuration, dynamic gesture use moving gesture by sequence and the hand configuration [1]. There are also two types of gesture interaction: communicative gestures, which work as a symbolic language and manipulative gestures, which provide multi-dimensional control. Sign language is not universal. Different countries have different sign languages in alphabets and word sets, for example American has American Sign Language (ASL) and German has German Sign Language (GSL). The similarity is that sign languages use body movements, right hand, left hand, or both. Signers may also utilize their heads, eyes, and facial expressions [2].

The studies in hand gesture began in the 90s, so that many researches have been developed in sign language recognition. Ding in [3] discovers an innovative algorithm for obtaining the motion, hand shape and Place of Articulation (POA) of the manual sign. Hand shape, motion, and POA are manual component from ASL, whereas facial expressions are non-manual component. POA can identify the location of the hand with respect to the face and torso of the signer. The motion by the dominant hand is detected from video starts to ends. The dominant hand is single handed that usually signer used. Most of signers usually left hand as dominant hand, but in

DOI: 10.5121/ijci.2014.3104

several cases there are right-handed people. Oz in [2] has developed a reliable adaptive filtering system with a Recurrent Neural Network (RNN). It used to determine the duration of ASL signing. By using histogram method to extract features from signs and neural network to classify, the system can convert ASL signs into English words. Other study, Munib in [1] presented a developing of a system for automatic translation of static gestures of alphabets and signs in ASL. By using Hough transform and neural networks, the system trained images to recognize signs. The interesting is that system does not rely on using any gloves or visual markings to recognize so it makes the system more flexible. Besides that research in real time sign language recognition is highly essential. According to [4], there are several issues affecting Real-time sign language recognition, i.e. (i) video image acquisition process depends on environmental conditions like lighting sensitivity, and background condition, (ii) Detecting bounding box as hand shape boundary should be determined automatically from the captured video streaming data, (iii) A sign is affected by the preceding and the subsequent sign. Moreover, in this study we will translate the sign language based on ASL into text form in real time. Although many studies used various method have been done, we still must explore other technique to find a better one which can translates sign language correctly.

In image processing, we firstly need to extract features from image to go to next process. One of the techniques to feature extraction is Generic Fourier Description (GFD). GFD which is derived by applying 2D Fourier Transform is robust in shape descriptor because it captures spectral feature in both radial and circular direction [5]. In other hand, K-Nearest Neighbour (KNN) is a simple classifier that easy to compute. By using a similarity measure, KNN can determine a label of testing data. Because of their good ability, we proposed GFD and KNN algorithm to translate sign language. By using this hybrid method, we hope that it provides better experiment result than the existing method. Finally, this paper aims to create a system that can translate static sign language into textual form automatically based on computer vision using GFD as feature extraction method and KNN as classification approach. Sign language that will be recognized is based on ASL. Translation process has three phase, they are segmentation, feature extraction, and recognition. In this system, signers are not required to wear any gloves or to use any devices. To know the performance, the system tested in 120 stored images in database and 120 images which is captured real time by webcam. We also translated 5 words in video sequences.

2. RESEARCH METHOD

2.1. Data

Sign language used based on static ASL which involves the discrete set of 24 signs corresponding to the letters of the alphabet (without two letters, J and Z). These are shown in Fig 1. The dataset is divided into training and testing data afterward. The dataset used for training in recognition process consists of 360 images, i.e. 15 images for each static signs. The images for training process were taken from 5 different volunteers. The data set used for testing in recognition process consists of 240 images, i.e. 120 stored images in database and 120 images which is captured by webcam. Over all, for each sign, 15 out of 25 captured images were used for training purpose, while the remaining 10 signs were used for testing.

2.2. Skin Detection

In images and videos, skin colour is an indication of the existence of hand shape or human in video. Skin detection aims to segment skin area or hand shape as foreground from background. Skin colour segmented from RGB colour space. RGB colour space is the most commonly used colour space in digital images. It encodes colours as a combination red (R), green (G) and blue

(B) as of three primary colours. In fact, distances in the RGB space is not linearly correspond to human perception. In this colour space, luminance and chrominance are not separated. The luminance of a given RGB pixel is a linear combination of the R, G, and B values [6]. In skin detection, we should transform RGB colour space into YCbCr to retain the luminance. The YCbCr is a family of colour spaces used as a part of the colour image popeline in video. In this colour space Y is the luma component and Cb and Cr are the blue-difference and red-difference chroma components. Transforming from RGB to YCbCr compute based on Eq 1

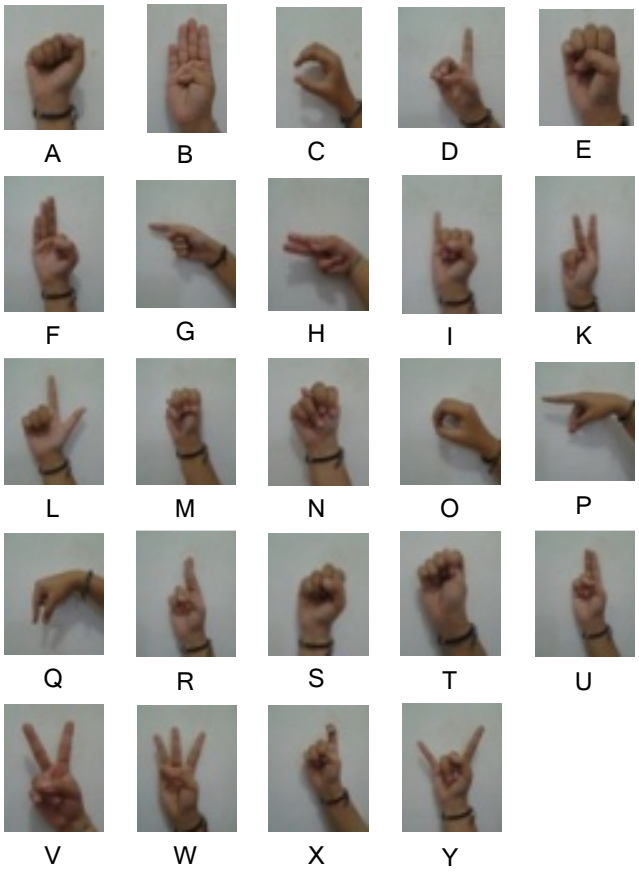


Fig 1. Static American Sign Language

$$
\begin{aligned}
Y &= 0{,}299 * R + 0{,}587 * G + 0{,}114 * B \\
Cb &= -0{,}1687 * R - 0{,}3312 * G + 0{,}5 * B \\
Cr &= 0{,}5 * R - 0{,}4183 * G - 0{,}0816 * B
\end{aligned} \tag{1}
$$

After transforming YCbCr, pixels with certain criteria will be detected as foreground. We used simple criteria to detect skin colour, that are Cr on interval [132,173] or Cb on interval [76,126] [7]. Skin detection processing shown in Fig 2. Fig 2(b) shows transformation result into YCbCr colour space. Fig 2(c) is result of skin area detection. Fig 2(d) is transformation result from YCbCr into RGB. Transforming back from YCbCr into RGB we use Eq 2

Figure 2. Skin Detection Process

$$R = Y + 1{,}4022 * Cr$$
$$G = Y - 0{,}3456 * Cb - 0{,}7145 * Cr$$
$$B = Y + 1{,}7710 * Cb \quad (2)$$

A custom user interface is created using MATLAB 2009a software to facilitate an efficient acquisition of training samples. Sign from hand shape is recognized from significant fingers. Significant fingers are captured from video sequence of the signer which is paused [3]. In this work, we position the camera facing the signer in order to capture the front view of the signer's hand shape. Actually, moving hand detection has the same approach as simple skin detection. It starts to detect skin area and determine the bounding box area. If the image moves, the bounding box will follows until the acquisition process is finished. Fig 3 shows a motion object being detected as hand shape.

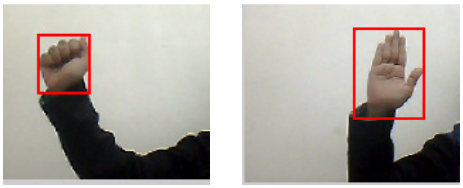


Figure 3. Motion Detection Process

2.3. Generic Fourier Descriptor

Zhang in [5] explained that shape description is one of the key parts of image content description for image retrieval. Most of the existing shape descriptors are usually either application dependent or non-robust, making them undesirable for generic shape description. GFD is proposed to overcome the drawbacks of the existing shape representation techniques that independent and robust. Generally, 1-D Fourier Descriptor (FD) is obtained through Fourier transform (FT) on a shape signature function derived from shape boundary coordinates $\{x(t), y(t), t = 0,1,2, \dots, N-1\}$. GFD is derived by applying 2-D Fourier transform on a polar shape image. In region based techniques, shape descriptors are derived using all the pixel information within a shape region.

Given a shape image $I = \{f(x,y);\ 0 < x < M, 0 < y < N\}$. Then, image *I* is converted from cartesian space to polar space $I_P = \{f(r,\theta); 0 < r < R, 0 < \theta < 2\pi\}$, where R is the maximum radius of the shape. The origin of the polar space is set to be the central of the shape, so that the shape is translation invariant. The centroid $(x_c, y_c)$is given by Eq. 3.

$$x_c = \frac{1}{N}\sum_{x=0}^{N-1} x, \qquad y_c = \frac{1}{M}\sum_{y=0}^{M-1} y \tag{3}$$

where distance *r* and angle θ can be calculated based on Eq. 4.

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \qquad \theta = arctan\frac{y - y_c}{x - x_c} \tag{4}$$

Moreover, if PF is polar Fourier transform, *R* is radial frequency resolution and *T* is angular frequency resolution, *m* is number of radial frequency maximum and *n* is number of angular frequency, we obtained PF based on Eq. 5 and shape descriptor based on Eq. 6.

$$PF(\rho, \emptyset) = \sum_r \sum_i f(r, \theta_i)\exp\left[j2\pi\left(\frac{r}{R}\rho + \frac{2\pi}{T}\emptyset\right)\right] \tag{5}$$

$$GFD = \left\{\frac{PF(0,0)}{2\pi r^2}, \frac{PF(0,1)}{PF(0,0)}, \ldots, \frac{PF(m,n)}{PF(0,0)}\right\} \tag{6}$$

2.4. Translation of Sign Language Using GFD and KNN

The first step of translation of sign language using GFD and KNN is pre processing. In pre processing step, we should detecting and cropping the skin area, resizing, and also filtering the image. After skin detection, cropping is applied based on a bounding box as the boundary that lies from upper left corner to bottom right corner of the skin area. The pre processing result is shown in Fig 4. Fig 4(b) shows the result of skin area detection or hand shape detection based on Section 2.2 and Fig 4(c) shows the generated bounding box as the boundary of the skin area. After that, the images must be resized into a smaller size for an easier computation. In this experiment we resize the images into 20% of the original size. Fig 4(e) shows the resized image. In pre processing, we also need to filter the sign. A moving average or median filter mask 3x3 is used to remove the unwanted noise from the image scenes. Fig 4(f) shows the image after filtering.

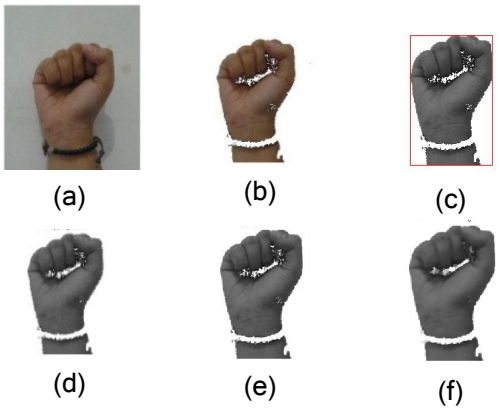


Fig 4. Pre processing

After pre processing, we extract image features using GFD method. Feature extraction is applied on binary image so canny method is used in order to detect the edges. The edge detection result is shown in Fig 5(b). In section 2.3, we also consider *m* radial frequency and *n* angular frequency to

obtained ($nT + m + 1$) features representing an image. In this paper, we choose 4 as radial frequency and 6 as angular frequency.

Fig 5. Edge Detection Process

Then, ($nT + m + 1$) features of each image are processed and converted to a features vector for the training set. For recognition task, we use K Nearest Neighbour (KNN) to classify the testing data. In KNN classification, similarity measures used are Euclidean Distance and Cosine Similarity. Given two pixel $a(x_1, y_1)$ and $b(x_2, y_2)$ similarity measures can be calculated by Eq (6) and (7).

$$d_E = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6}$$

$$\text{cosine}\,(a, b) = \frac{\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}}{\left|\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right| \left|\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}\right|} \tag{7}$$

Finally, we evaluate the performance of the system based on its ability to classify samples to their corresponding classes correctly. The recognition rate is defined as the ratio of the number of correctly classified samples to the total number of samples. Over all, the procedure of this method is shown in Fig 6.

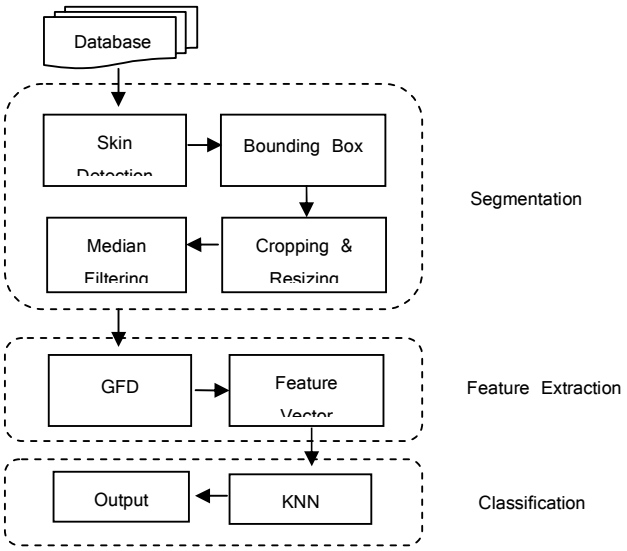


Fig 6. Flowchart Translation of Sign Language Using GFD and KNN

3. Result and Analysis

3.1 Experiment result

The system was implemented in MATLAB version 2009a. We implemented it in two scenarios. In the first scenario, we used 360 images for the training process and 240 images for testing that each sign alphabet we have trained 15 images for training and 10 images for testing. Based on this scenario, we will know how system performance in recognizing for each sign. In the second scenario, we used real time images by capturing several sign sequences. Based on this scenario, we will know how system performance in translating word.

In first scenario, we recognized signs twice. Firstly, we load 5 stored images for each sign in database and secondly we capture 5 images for each sign by real time using webcam. When capturing signs, the system would detect motion object as hand shape. After that, the captured image would be pre processed and compared with the feature vector of the training data. Fig 7 shows how to capture and crop the hand shape area. In this experiment, we used KNN method to classify the data testing by using Euclidean Distance and Cosine Similarity. For KNN's parameter, we choose k = 1, 3, 5 to obtained k nearest Neighbours. The testing results for both stored image recognition and real time image recognition are shown in Table 1.

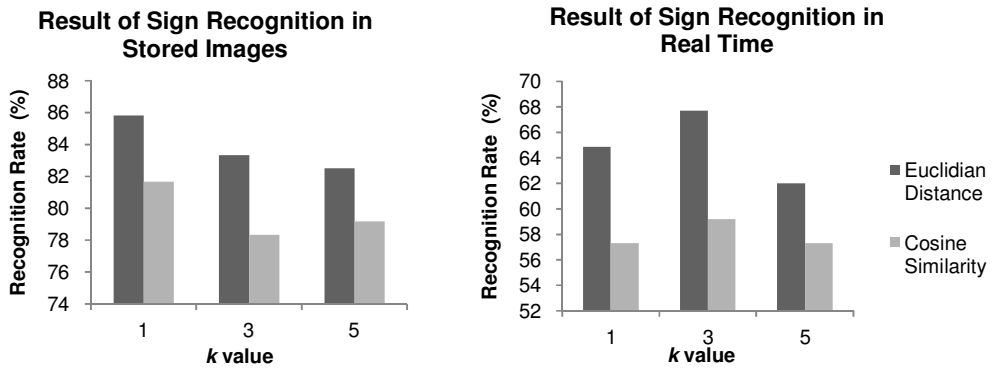
(a) (b)

Fig 7. Capturing Real Time

Table 1. Result of Sign Recognition

Sign	*Recognized rate of Stored Image*						*Recognized rate Real Time*					
	Euclidian			Cosine			Euclidian			Cosine		
	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5
A	0.6	0.6	0.4	0.6	0.4	0.4	0.6	0.8	0.8	0.6	0.6	0.6
B	0.6	0.6	0.6	0.6	0.6	0.8	0.8	0.6	0.6	1.0	1.0	1.0
C	1.0	0.8	0.8	1.0	0.6	0.4	1.0	1.0	1.0	1.0	1.0	1.0
D	0.6	0.8	0.6	0.8	0.8	0.6	0.8	0.8	0.6	0.6	0.6	0.4
E	1.0	1.0	0.8	1.0	1.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0
F	1.0	1.0	1.0	0.4	0.6	0.6	0.6	0.6	0.6	0.8	0.6	0.8
G	0.8	0.6	0.2	0.6	0.6	0.6	0.8	0.8	0.4	0.6	0.6	0.4
H	1.0	0.8	1.0	1.0	0.8	1.0	1.0	1.0	1.0	0.8	1.0	1.0
J	1.0	0.8	0.8	1.0	0.8	0.8	0.6	0.8	0.6	0.6	0.6	0.6
K	1.0	0.8	0.8	1.0	0.6	0.6	0.4	0.6	0.4	0.6	0.6	0.6
L	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
M	1.0	1.0	1.0	1.0	1.0	0.6	0.2	0.0	0.2	0.0	0.0	0.0
N	0.8	0.8	0.8	0.8	0.8	0.8	0.2	0.4	0.2	0.0	0.0	0.0
O	0.4	0.4	0.6	0.4	0.6	0.6	0.8	0.8	0.6	0.8	0.6	0.6
P	0.8	0.8	0.8	0.8	0.8	0.6	0.8	0.8	0.8	0.8	0.8	0.8
Q	0.8	0.8	0.8	0.8	0.6	0.4	1.0	1.0	1.0	1.0	0.8	0.8
R	0.8	0.8	0.8	0.6	0.6	0.8	0.8	0.8	0.8	0.8	0.8	0.8
S	1.0	1.0	0.6	1.0	1.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0
T	1.0	1.0	0.8	1.0	1.0	1.0	0.6	0.4	0.4	0.4	0.4	0.2
U	0.8	0.8	0.8	0.8	0.8	0.8	0.8	1.0	1.0	0.8	1.0	1.0
V	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
W	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.8	0.8	0.2	0.2	0.4
X	0.6	0.4	0.4	0.8	0.8	0.6	0.8	0.8	0.8	0.8	1.0	0.8
Y	1.0	1.0	1.0	1.0	1.0	1.0	0.6	0.8	0.8	0.4	0.8	0.8

Finally, we obtained 86 % accuracy as the highest result when using Euclidean Distance similarity and k = 1 for testing in stored images and 69 % accuracy when using Euclidean Distance similarity and k = 3 for testing in real time images. Result of recognition rate between first recognition and second one shown in Fig 8. Although the first recognition has a good performance, it cannot be guaranteed that the system is good enough. In fact, the second recognition has low accuracy. It may caused by the number of training data is too small that does not have different reasonable orientations.



(a) (b)

Fig 8. Recognized Rate of Sign Recognition

In the second scenario, we used real time images by capturing several signs sequences. We used several words consisting of 2 to 4 letters to test the system, i.e. 'HI', 'AYO', 'AKU', 'KAMU', and 'HALO'. The experiment result show that the system still could not recognized each letter in word. The word 'HI' could only be translated correctly. The result of this scenario is shown in Table 2.

Table 2. Result of Word Translation

Sign	METHOD					
	Euclidian			Cosine		
	k=1	k=3	k=5	k=1	k=3	k=5
HI	HI	HI	HI	HI	HI	HI
AYO	TYA	TYA	AYA	TYA	TYA	AYA
AKU	TKB	TKB	IKU	TKF	TKF	YKU
KAMU	KAEU	KAEU	KAEU	KAMU	KATU	KATK
HALO	HTLA	HTLA	HTLA	BTLA	HTLA	HTLA

We also used PCA for comparing the performance of our method. The performance of the PCA model was tested in 24 static signs in the ASL alphabet using Euclidean Distance and k =1,3,5. The results are given in Figure 9.

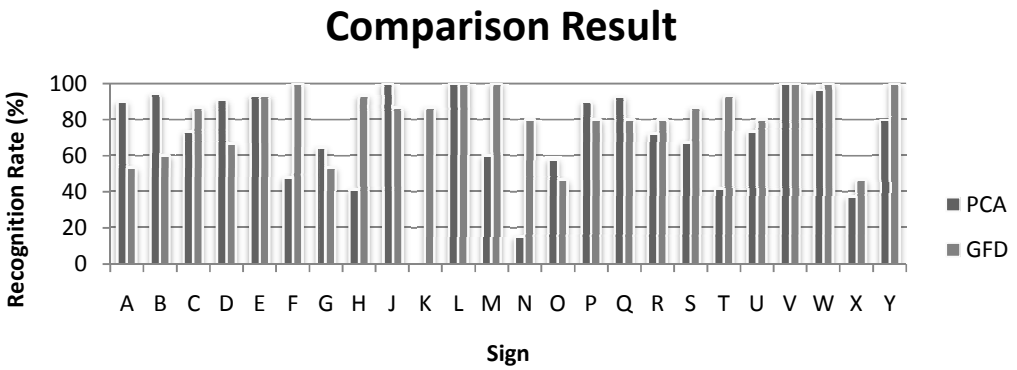


Fig 9. Comparison Result

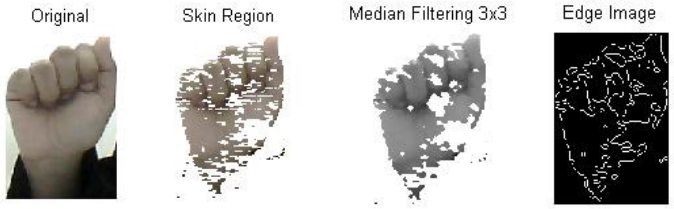
3.2 Discussion

In scenario 1, we translated 24 signs into alphabet. In the first recognition by using stored images as testing data, we obtained good recognized rate. But, in the second one, recognition by using real time images as testing data, we obtained a less accuracy. As shown in the result table, we came across some misclassifications. In 'C' and 'L' case, the system recognized the sign with 100% accuracy. However, in 'A', 'M', 'N', and 'T' case, it could only achieve under 50 % accuracy. It may be caused by their similarity, so when we captured 'A' sign, the system might recognized it as 'M' and vice versa. Moreover, in 'E' and 'S' case, system could not recognized at all. From different directions, some signs looked like different signs e.g. 'A' looked like 'E' if the thumb was blurred and the edge was significantly undetected. In other hand, 'S' looked like 'A' if the index and middle fingers were not raised highly enough. Actually, these problems may cloud be solved by using a better skin detection and filtering method, which the researcher might try to focus on in future research. An example of poor skin detection and filtering is shown in Fig 10. Fig 10(b) shows the result of skin detection in poor lighting condition. In such condition, the skin area might have the same luminosity as the background (in this case a wall) to an extent that some skin area failed to be segmented. Fig 10(c) is the result of image filtering using mask. Mask could only blur little area to reduce noise in bounding hand, but not in central one. If we have used bigger mask, some edges of the fingers would fade. It would be ideal if we could fill the missing skin area in the center of the image.

Fig 10. Poor skin detection and filtering image

In scenario 2, we obtained low recognition rate. Overall, the previous problems were also found in this scenario. Additionally, various backgrounds did not only have the same properties as the skin, but also contains noises. It was seriously difficult to detect hand shape of the hand shape of the captured images that contain noises in the background. Besides that, lighting also affected the

image capture significantly. Fig 11 shows the result of the pre processing of real time image capture. The edge image, as shown in Fig 11(d), does not resemble hand shapes well due to the skin detection and filtering method used in this experiment.



(a) (b) (c) (d)

Fig 11. Poor preprocess real time image

Comparing with PCA, the proposed method had a better performance in 81.39% (GFD-KNN) compared with 69.86% (PCA-KNN). Although PCA had better result in some signs, such as A, B, D, and G, but over all GFD' performance is better than PCA's.

4. Conclusion

In this project, we developed a system with the purpose of translating ASL alphabet. The system has three phases, i.e. segmentation, feature extraction, and recognition. Without the need of any gloves, an image of a sign can be taken with a webcam. The proposed system was able to reach a recognition accuracy of about 86% for testing data in stored images and 69% for testing data in real time images. We also translated 5 words in video sequences and obtained low recognition rate in word translation. In fact, we have some problem in skin detection process that caused some skin area failed to be segmented and affected in recognition rate. Actually, in the future, the researcher might try to focus in these problems by using a better skin detection and filtering method.

REFERENCES

[1] Munib Q, Habeeb M, Takruri B, Al-Malik H. American sign language (ASL) recognition based on Hough transform and neural networks, Expert Systems with Applications. 2007; 32; 24–37
[2] Oz C, Leu M. American Sign Language word recognition with a sensory glove using artificial neural networks. Engineering Applications of Artificial Intelligence. 2011; 24; 1204–1213
[3] Ding L, Martinez A. Modelling and recognition of the linguistic components in American Sign Language. Image and Vision Computing. 2009; 27; 1826–1844
[4] Mekala P, Gao Y, Fan J, Davari A. Real-time Sign Language Recognition based on Neural Network Architecture. IEEE. 2011, 978-1-4244-9592-4/
[5] Zhang D, Lu G. Generic Fourier Descriptor for Shape-based Image Retrieval. 2002
[6] Elgammal A, Muang C, Hu D. Skin Detection - a Short Tutorial, Encyclopedia of Biometrics by Springer-Verlag Berlin Heidelberg. 2009; 1-10
[7] http://www.mathworks.com/matlabcentral/fileexchange/28565-skin-detection

Authors

Abidatul Izzah, received her B.S. degree, in Mathematics, from Airlangga University, Indonesia in 2012. Currently she has been studying in Informatics Engineering, at the third semesster of Master Degree in Institut Teknologi Sepuluh Nopember (ITS), Indonesia. Her research interests include data mining, artificial intelligence, and swarm intelligence.

Nanik Suciati received the B.S. in Computer Engineering from Institut Teknologi Sepuluh Nopember (ITS), Indonesia in 1994 and M.S. degrees in Computer Science from University of Indonesia in 1998, and PhD from Hiroshima University, Japan in 2010. Since 1994-present, she is an academic and researcher staff in Informatics Department, ITS, Indonesia. Her research interests include Computer Graphics, Computer Vision and Image Processing.