PAPER • **OPEN ACCESS**

Sign Language Translator Application Using OpenCV

To cite this article: L Triyono *et al* 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **333** 012109

View the article online for updates and enhancements.

Related content

- Detection of finger gesture using singular spectrum transformation
  Y Tamura, T Umetani and H Nakamura
- Dynamic gesture classification using skeleton model on RGB-D data
  Y Tamura, T Umetani, N Kashima et al.
- Gesture Recognition for Educational Games: Magic Touch Math
  Neo Wen Kye, Aida Mustapha and Noor Azah Samsudin

IOP ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

This content was downloaded from IP address 120.188.66.196 on 01/07/2018 at 23:43

IOP Conf. Series: Materials Science and Engineering **333** (2018) 012109 doi:10.1088/1757-899X/333/1/012109

Sign Language Translator Application Using OpenCV

L Triyono[1], E H Pratisto[2], S A T Bawono[2], F A Purnomo[2], Y Yudhanto[2] and B Raharjo[2]

[1]Informatics Department, Politeknik Negeri Semarang
Jl. Prof. Sudharto, Tembalang, Semarang, Jawa Tengah 50275, INDONESIA
[2]Informatics Engineering Department, Sebelas Maret University
Jl. Ir. Sutami 36A Kentingan Jebres Surakarta 57126, INDONESIA

E-mail: sahirul@mipa.uns.ac.id

Abstract. This research focuses on the development of sign language translator application using OpenCV Android based, this application is based on the difference in color. The author also utilizes Support Machine Learning to predict the label. Results of the research showed that the coordinates of the fingertip search methods can be used to recognize a hand gesture to the conditions contained open arms while to figure gesture with the hand clenched using search methods Hu Moments value. Fingertip methods more resilient in gesture recognition with a higher success rate is 95% on the distance variation is 35 cm and 55 cm and variations of light intensity of approximately 90 lux and 100 lux and light green background plain condition compared with the Hu Moments method with the same parameters and the percentage of success of 40%. While the background of outdoor environment applications still can not be used with a success rate of only 6 managed and the rest failed.

1. Introduction

Sign language is mostly only dislocated by the disable and few people who understand the sign language that usually lives associated with disables such as families, activists, and teachers of *Sekolah Luar Biasa* (SLB). Sign language is divided into two namely natural gestures and formal cues[1]. The natural cue is a cue used by a deaf person (as opposed to body language), a manual (hand-handed) expression agreed between the user (conventionally), known to be limited in a particular group (esoteric), and a substitute for words. A formal gesture is a cue that is intentionally developed and has the same language structure as the spoken language of the community. Various forms of formal sign language developed, among other things, sign language called Sign English or also called Pidgin Sign English (PSE) which is a combination or mixture of original / natural sign language with English, standard sign language American Sign Language (ASL) to explaining the words and concepts[2].

1.1. American Sign Language

American Sign Language (ASL) is one of the most widely used sign language in the world with the method of fingerspelling as a representation of the alphabet on cues[3].

The development of sign language application has been developed by several researchers, including Nuriyanti[4] has made basic sign language introduction application based on Android. The developed application is limited to text recognition and can not translate gesture sign language. Ismawati[5] developed the application of communication aids tool with Deaf through webcam on computer. The

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

application will translate the movements of a finger that is demonstrated to be translated into an alphabet letter. This research will develop a sign language interpreter application on a mobile device to make it easier for everyone to communicate with the Deaf. However, some hand gestures (alphabets) such as K, M, Q, U, X, H, P, S, T have not been translatable because they have a more dominant range of other approaches to the HuMoments alphabet.

1.2. Fingerspelling

Fingerspelling describes the alphabet manually. Hand positions show each letter of the Latin alphabet. Fingerspelling is usually used as a complementary sign language. If there is no sign language for a single word, then fingerspelling is used. Fingerspelling is usually also used to mention names appropriately or when people are unsure of sign language for a particular word[6] .

2. Experimental

An overview of the application's design and implementation as seen at Figure 1.

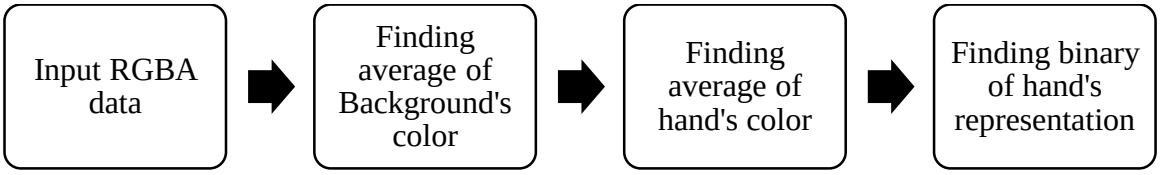

Figure 1. Background Subtraction Flow Chart

2.1. Finding background color

At the beginning, the application will detect the background color on 7 sections coordinates that are marked with 7 boxes as shown on Figure 2. The average color will be stored in a 7 x 3 matrix array. The application starts with two pre-sampling steps, which collects the background color and the user's hand using the 7 small squares displayed on the screen. This color data is used to calculate the threshold for obtaining binary imagery from RGBA data input. For simplicity, adaptive methods are not used to find the best threshold. Next, the user is asked to put his hand close to the screen to cover 7 boxes so that the app can get hand color data.

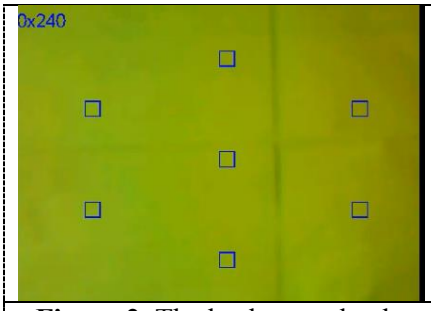

Figure 2. The background color sampling is stored in the array matrix

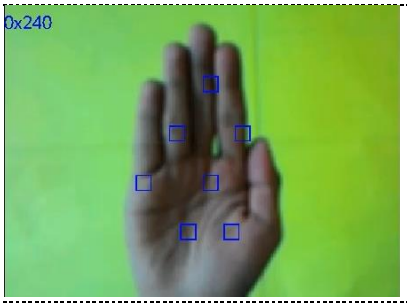

Figure 3. Hand color sampling that will be stored in array matrices

2.2. Finding hand color

The average color will be stored in a 7 x 3 matrix array like the image. After 7 color data for hand is obtained, 7 upper and lower borders for hand area are calculated, which can be represented as 2 dimensional array. The array's color boundaries are authorized by the author and can not be changed. Therefore, the actual segmentation performance depends on the choice of the array color limit. The original color of the frame ie RGBA will be converted to obtain the best color samples to detect the same hand in different light conditions.

2.3. Binary of the hand representation

After the limit of each 7 colors is obtained, the next 7 binary images of hand representation can be computed by performing on each binary image using the 'OR' logical operation. The same process is done in the background color to produce a binary image. The result of the binary image of the hand is then performed an operation logic 'AND' with binary image of the background. Furthermore, the blurring is done and the last process is by performing morphological operations (dilation and erosion) to remove noise (pixels that are not in want) to gain optimal result. After all process were done, the obtained representation of the hand as shown on Figure 4.

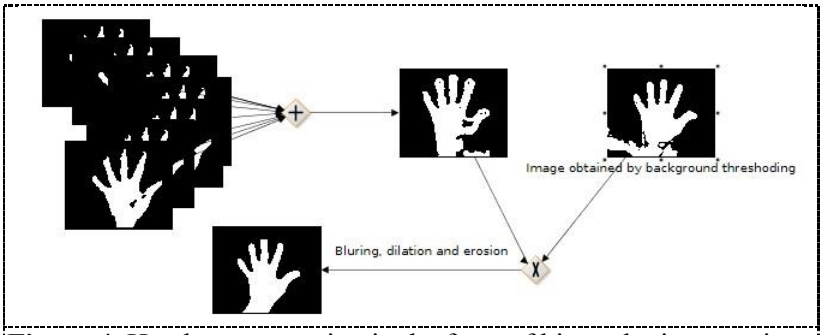

Figure 4. Hand representation in the form of binary logic operation Source: http://eaglesky.github.io/images/posts/hand-gesture/thresholding-back.jpg

2.4. Finding contour and feature extraction

The process of Feature Extraction as seen at Figure 5. The result of hand representation on binary image will be used to search contours. Furthermore, the contour will be described next operation such as drawing bounding box, convex hull, and convexity defects. It will also be drawn a circle on the palm of the hand. This circle is useful for finding the midpoints that will be used as features along with the location of the finger to train and then in the prediction. Figure 6 shows an illustration of the results of the functionality found in OpenCV.

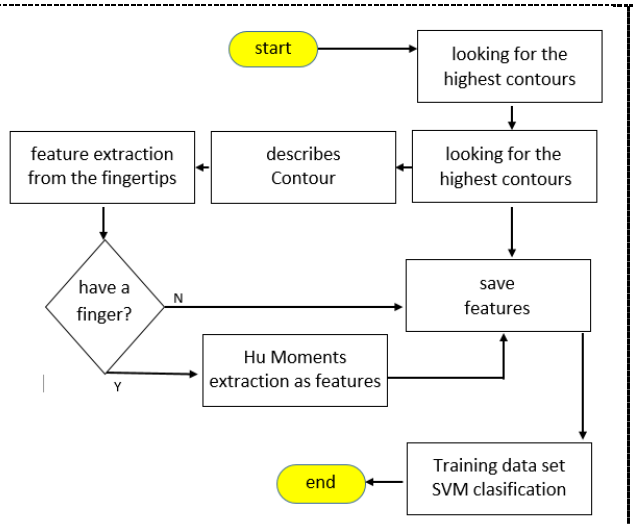

Figure 5. Feature Extraction Flowchart

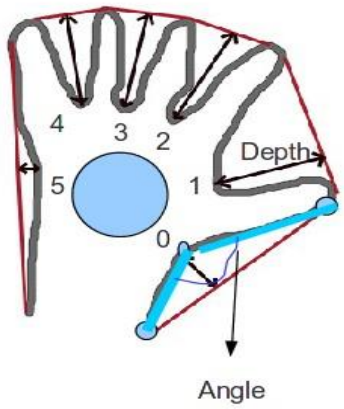

Figure 6. Features extraction on a hand Source: http://eaglesky.github.io/ images/posts/ hand-gesture/defects-elimination.jpg

Gray line represents contours, red line is convex hull. The location between the red lines of contours with convex hull is called convexity defects. The location of the fingertips is calculated using a defect point. Redundancy defect points will be eliminated by checking the depth and angle of the defect point. As illustrated in Figure 6, there will be 4 checks of conditions, ie Depth <= circleRadius * 0.7, Cosine (Angel) <= -0.7, close to the limit, the number of defects> 4. If there are unfulfilled consissions it will not be assume a hand or not processed.

The condition applies if a finger opens. However, when the hands are clenched or none of the fingers open then feature extraction is performed by another method by searching 7 Hu Moments values. The value of Hu Moments will be included in the SVM data model and subsequently used to predict the outcome.

2.4.1. *Data training.* After. The result of feature extraction ie fingertip coordinates or Hu Moments values will be entered on SVM data model with sequential label. The labels are sequential from 1 for A, 2 for B and so on. In doing the training then the sequence should not be reversed because it can be ascertained the prediction results will also be wrong.

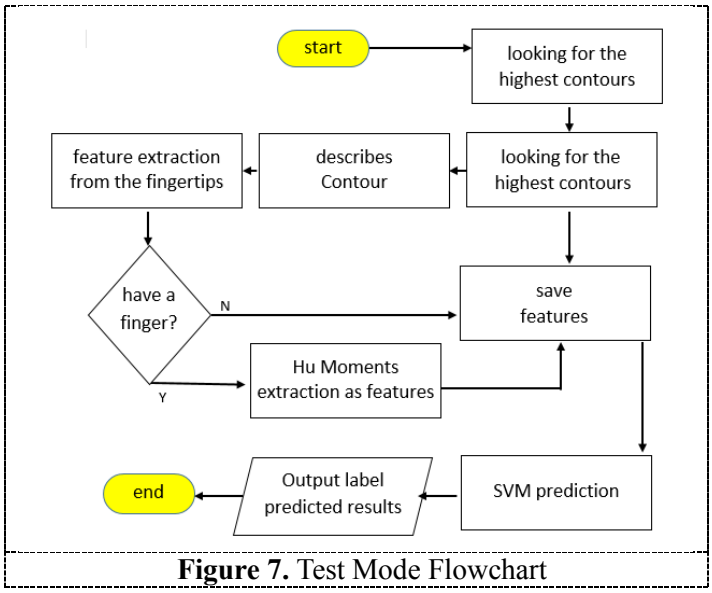

Figure 7. Test Mode Flowchart

The application test process as seen at Figure 7. The initial process in this section is the same as the previous explanation of contours search and feature extraction. Predicted results are only done in test mode. The result of feature extraction will be predicted by SVM based on the data set that has been stored during the training to produce sequence label then compared with alphabetical order and displayed on screen.

3. Result and Discussion

Data training is data generated from the training process and stored in the ReadMySign folder in the root of internal storage. This data will be used to predict the results of labels to display the results of hand gesture readings. Training data includes:

1. *Gesture image.* This data is an image of a saved gesture. Noteworthy of this data is that this image data is not used for application or programming purposes. Gesture image data is only a palm / fist with the naming of an Arabic number that is adjusted to the order of adding data. An image with the name 1 represents label 1 and in this study is considered the letter 'A' then label 2 as 'B' and so on.

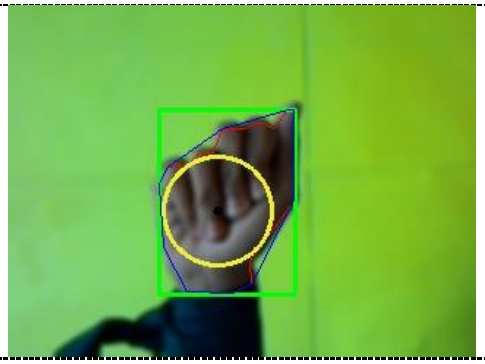

Figure 8. Alphabet "A" with Hu Moments Features

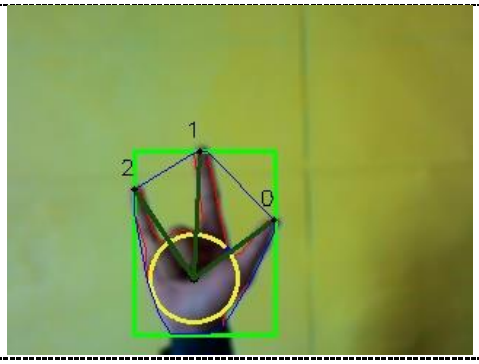

Figure 9. Alphabet " **I LOVE U** " with Fingers Features

2. *Features File.* This data is text that is named as train_data.txt which contains features of each gesture. Features for Hu Moments method there are 7 values while for the fingertip search method varies depending on the number of open fingers. Each gesture is labeled 1 to n and each gesture maintains the same 10 units of labels but contains different values but with the same amount for each gesture. Ten units of the label are taken from 10 frames during the training, each frame contains 1 unit of feature. The value of features with the fingertip search method numbered vary, according to the number of open fingers when there are 2 fingers open then there are 4 value features, if there are 3 fingers open then aka tone 6 value features. The value is obtained from the coordinates of each finger (x and y), each calculated first through the division with the radius of the circle in order to obtain the desired result.
3. *SVM Model File*. This file in the application is named 'model'. The model file is a generated file from the train_data.txt file performed by SVM. This file contains a training data model that SVM will use to predict the label.

Table 1. Test Condition

	Object distance from camera (cm)	Light intensity (lux)	Background color	Marker color
1	35	950 - 1050	light green	light green
2	35	105 - 115	light green	light green
3	55	105 - 115	light green	light green
4	35	2435 - 2521	outdoor	light green
5 (using gloves)	35	334 - 354	outdoor	light green

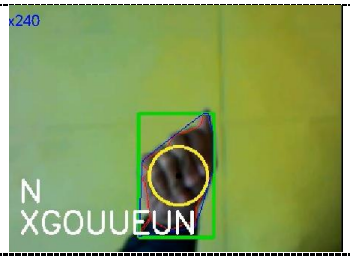

Figure 10. Gesture "A" on Test 1

Figure 11. Gesture "A" on Test 4

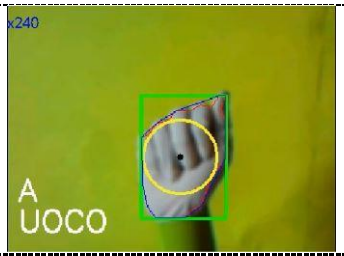

Figure 12. Gesture "A" on Test 5

Figure 13 shows that the search method of fingertip coordinates is better than the search for Hu Moments values. Almost all gestures that use the value of fingertip coordinates as features show 100% success except gesture letter H. This can happen because even in the training of the letter H using the coordinates of the fingertips but when the test is detected with Hu Moments and gesture the letter H is also very

similar to gesture U with a little slope. Gestures of the letters E, M, N, and T have not been recognized because the three gestures are very similar to each other. Overall Hu Moments method is used for 11 gestures with 4 of them failing.

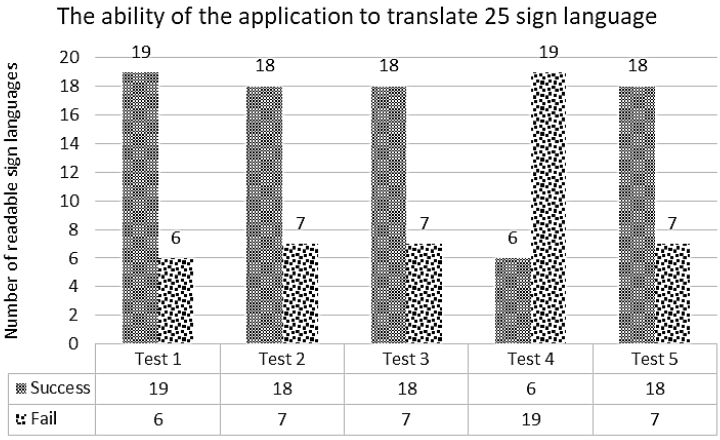

Figure 13. Test result

In the 4th experiment with a light intensity above 2000 lux and the background is a plantation environment, the application made poor hand gesture recognition and achieved little success. It happens because the background is too crowded with a color that could just resemble skin color, other than that the intensity of light that is too high coming from the front of the camera also gives a silhouette effect on the hand. In the experiments using hand shirts it appears that there is a difference that the application can read with both the gesture E and H. Compared to research by [5] this study can recognize the letters K, Q, U, X, H, P or 80% more efficient than the previously developed method of only 64%.

4. Conclusion

The method of color difference can be used to detect the color of the skin as a recognized object as a sign language. Some gestures still can not be translated because they have a high similarity to each other like M, N, T, and S; contour comparison motion with 7 Hu Moments values for hand sign language recognition has a small success rate of 50%. Applications can not translate gestures properly without a certain background. The solution for dynamic sign language on every word, further research can take advantage of Motion Templates on OpenCV. This method is very useful in detection motion in a video. The important feature of this technique is that motion can be detected even in small regions of a frame.

References

[1] McInnes J M and Treffry J A 1993 *Deaf-blind Infants and Children: A Developmental Guide* (Toronto : University of Toronto Press)
[2] Martin D S 2003 *Cognition, Education, and Deafness: Directions for Research and Instruction* (Washington : Gallaudet University Press)
[3] Williams P and Evans M 2013 *Social Work with People with Learning Difficulties* (California: SAGE Publications)
[4] Nuriyanti Y and Tresnawati D 2015 *Algoritma* **12** 1-9bvcgghfd
[5] Ismawati R D 2011 *Pembuatan Software Alat Bantu Komunikasi Penyandang Cacat Tuna Rungu-Tuna Wicara (Berbahasa Isyarat Tangan) Berbasis Webcam*. (EEPIS Repository: Politeknik Elektronika Negeri Surabaya)
[6] Smith M and Levack N 1996 *Teaching Students with Visual and Multiple Impairments: A Resource Guide* (Texas : Texas School for the Blind and Visually Impaired)