Songklanakarin J. Sci. Technol.
41 (2), 436-444, Mar. - Apr. 2019

http://www.sjst.psu.ac.th

Original Article

A real-time hand segmentation method using background subtraction and color information

Pawin Prasertsakul*, Jakkrit Dulayatrakul, Toshiaki Kondo, and Itthisek Nilkhamhang

School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani, 12121 Thailand

Received: 23 December 2016; Revised: 2 October 2017; Accepted: 1 December 2017

Abstract

This paper presents a real-time hand segmentation method that is based on background subtraction and color information. A hand, as foreground, is extracted from an image by background subtraction where unit gradient vectors (UGVs) are used instead of image intensities. The UGV-based method is more stable under dynamic lighting conditions because the UGVs are invariant to changes in illumination. Meanwhile, the hand is also detected using color information. These two method results lead into the final hand segmentation. Experimental results show that the proposed method can segment a hand in an image robustly under various lighting conditions. We have implemented the proposed method using a low-cost embedded board Raspberry Pi.

Keywords: unit gradient vector, real-time, hand segmentation, embedded board

1. Introduction

Hand segmentation is one of the steps in digital image processing and computer vision applications, such as hand gesture recognition (Pavlovic *et al*., 1997). Various hand segmentation methods have been proposed in the literature (Kakumanu *et al*., 2007). Adaptation approaches such as the Gaussian mixture model (Zhu *et al*., 2000) and skin locus (Storring *et al*., 2003) are examples of more advanced methods. However, if these approaches lose the target, the adaptation may wrongly adapt to a non-target. Traditional background subtraction fails to segment a hand when lighting conditions of two images, reference and current images, are not the same (Ogihara *et al*., 2006). Using color information is a direct method for retrieving a skin color region (i.e. hand) from images directly (Avinash *et al*., 2013; de Dios & Garcia, 2003). However, it is ineffective when skin color objects appear in the background. Two approaches, skin color region retrieving by Wang *et al*. (2011) and background subtraction by Musa *et al*. (2011), use RGB color space to segment the target skin-color. However, RGB color space is not suitable because it is sensitive to dynamic lighting conditions. A background subtraction based on two visual features, color information and edge features, is presented by Jabri *et al*. (2000). Since this method uses RGB color space, it is hard to subtract the background when it is under irregular lighting conditions. A combination method with edge detection and background subtraction is proposed by Javed *et al*. (2002). However, it does not work because of the limitations of background subtraction when two images have different lighting conditions. From literature reviews, we learned that the intensity-based approaches are inefficient when illumination is changed. These problems are mainly caused by the change of a light source or image acquisition from the visual sensors or camera’s auto-gain function. The object’s color in the background is also an important factor because the hand and background can be segmented by detecting the same color.

The main contribution of this paper is to propose a hand segmentation algorithm that is robust for varying lighting conditions. The proposed method uses unit gradient vectors (UGVs) and robust color information as effective

*Corresponding author
Email address: prasertsakul.p@gmail.com

features to perform the background subtraction. The benefit of using UGVs is that the UGVs are rather independent of lighting conditions, whereas image intensities are directly affected by the lighting conditions. Both experimental tests and real-world implementation show that the proposed method works robustly under varying lighting conditions.

2. Materials and Methods

From the comparison among various color spaces for skin color regions (Chaves-González *et al.,* 2010), the HSV color space, especially hue, is selected for our approach because it is robust to changing lighting conditions. Figure 1 shows a flowchart of the proposed method. In Figure 1a, the proposed method starts with retrieving the first frame from a video camera to register it as a background image. In Figure 1b, the subsequent frame is retrieved from the video camera. Then it is registered as the current frame (Figure 1c). Figure 1d is a background subtraction technique for removing background regions, which are referred to the background image. Figure 1e is a skin color segmentation to extract the skin color regions from full color images. Figure 1f is the integration between the two results from the background subtraction and the skin color segmentation, in order to solve for disadvantages of each technique. Figure 1g is a post-processing step for improving correctness in the integration result.

2.1 Background subtraction

Our background subtraction is not a traditional approach based on image intensities. We use the unit gradient vectors (UGVs) or normalized gradient vectors, which are robust in various lighting conditions (Kondo, 2011). Figure 2 shows the UGV-based background subtraction of two masks.

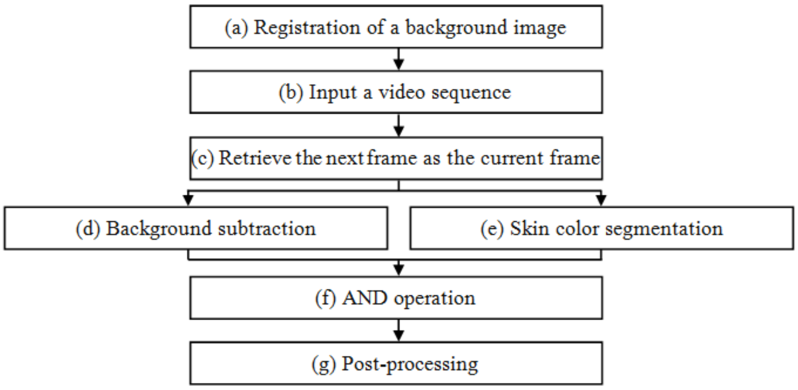
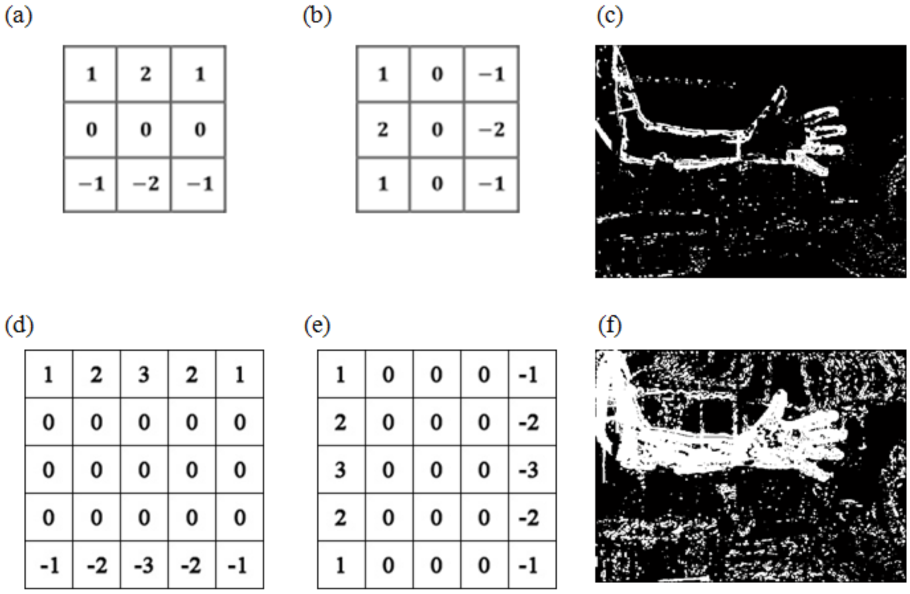

(a) Registration of a background image
(b) Input a video sequence
(c) Retrieve the next frame as the current frame
(d) Background subtraction
(e) Skin color segmentation
(f) AND operation
(g) Post-processing

Figure 1. Flowchart of the proposed method.



1	2	1
0	0	0
−1	−2	−1

1	0	−1
2	0	−2
1	0	−1

1	2	3	2	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
-1	-2	-3	-2	-1

1	0	0	0	-1
2	0	0	0	-2
3	0	0	0	-3
2	0	0	0	-2
1	0	0	0	-1

Figure 2. UGV-based background subtraction (a) vertical Sobel operator, (b) horizontal Sobel operator, (c) background subtraction using Sobel operators, (d) vertical extended Sobel operators, (e) horizontal extended Sobel operators, and (f) background subtraction using the extended Sobel operators.

First, the partial derivatives of the image intensities are computed in the horizontal direction I_x and vertical direction I_y. The traditional masks, Sobel operators (Figures 2a and 2b), are replaced by extended Sobel operators of size 5 by 5 pixels (Figures 2d and 2e) because they are more effective for suppressing high-frequency components in an image due to their more powerful low-pass filtering, as shown in Figures 2c and 2f. Then, the UGVs are computed as in Equation 1 and 2.

$$UGV_x(x,y) = I_x \Big/ \sqrt{I_x^2(x,y) + I_y^2(x,y)} \qquad (1)$$

$$UGV_y(x,y) = I_y \Big/ \sqrt{I_x^2(x,y) + I_y^2(x,y)} \qquad (2)$$

where UGV_x and UGV_y are UGVs in horizontal and vertical directions, while I_x and I_y are the partial derivatives of image intensities I in horizontal and vertical directions, respectively.

We assign zeros to UGV_x and UGV_y when the denominator is less than 0.08 in order to avoid division by zero. These two parameters correspond to regions where there are less or no gradients. This threshold value is determined experimentally by testing from 0.07 to 0.12. The two UGV directions in both the background image and the current frame are compared by calculating the Euclidean distance, as shown in Equation 3.

$$d(x,y) = \sqrt{d_x^2(x,y) + d_y^2(x,y)}$$

where $d_x(x,y) = UGV_{x,BG}(x,y) - UGV_{x,Cur}(x,y)$, and (3)

$$d_y(x,y) = UGV_{y,BG}(x,y) - UGV_{y,Cur}(x,y).$$

Let $UGV_{x,BG}$ and $UGV_{y,BG}$ denote the UGVs of the background image in horizontal and vertical directions and let $UGV_{x,Cur}$ and $UGV_{y,Cur}$ be from the current image in horizontal and vertical directions.

To set the optimal threshold value d, we have tested from 0.20 to 0.30. Finally, we found that the foreground pixels can be extracted when the threshold value d is greater than 0.24 (i.e. 13 degrees of angle). A comparison, using the two masks, is shown in Figures 2c and 2f. From the comparison, the extended Sobel operator segments a hand better than the traditional Sobel operator.

2.2 Skin color segmentation

Color information is used to extract the hand region from the input images. The proposed method utilizes hue and saturation of the HSV color space to segment the hand from full-color images. We have conducted experiments with Asian participants to find adequate ranges of the hue and saturation, of the skin color region. The hue threshold values were determined experimentally. From such experiments, we have selected hue ranges from 0.0 to 0.12 and from 0.88 to 1.0, which perform well for Asian skin color. Note that the hue values 0.0 and 1.0 are identical to the same pure red color. Therefore, the two ranges above are actually continuous, and thus, a single range. On the other hand, the valid range for the saturation is set from 10 to 255, which performs well for all hands, to ignore colorless regions.

2.3 Integration of the two methods

From the previous sections, we have obtained two results: UGV-based background subtraction and skin color segmentation. These contain the same target foreground (i.e. hand). Since the two results are presented as binary images, they can be integrated by an AND operation. The purpose of applying the AND operation to the two results is to select the same foreground from the two results, and to discard the false foreground which is proved by each result.

2.4 Post-processing

From the integration results, a segmented hand often contains holes within the hand region and also bright pixels in the background. We provide a circular structure element with 5-pixel diameter for morphological operations. First, the integration results are applied by morphological opening, to remove small isolated components in the background. Second, all small holes in the segmented hand are filled up by morphological closing. The largest connected component of bright pixels is select as the final target (i.e. hand). To fill up the large holes in the segmented hand, the binary image of the segmented hand is complemented. Then, the largest bright area is removed since it corresponds to the background. Finally, the remaining bright pixels are used to fill up the holes by an OR operation.

3. Results and Discussion

We conducted experiments on MATLAB with 2.0 GHz CPU and a 2.0M pixels built-in camera of a laptop. The frame resolution is 640 by 480 pixels. The camera is set to stationary. The experiments are performed on both plain and complex backgrounds under constant and irregular lighting conditions. To perform experiments under constant lighting condition, the brightness in the subsequent frames are maintained at the same level. To perform experiments under an irregular lighting condition, the intensity is decreased about 50% by turning off the room light.

3.1 Discussion of the experiment

Figures 3 to 6 show hand segmentation results in several situations. The results of Figure 3 belong to the hand with the plain background. The background has a white color with a pink object (Figure 3a). A hand exists in the frame under the same lighting condition (Figure 3b). Figure 3c is the traditional method result, the sum of absolute difference (SAD), which is retrieved by computing different intensity values of Figures 3a and 3b. The hand is unsuccessfully segmented because both background and current frames have similar intensity values. In the background subtraction result (Figure 3d), the hand is successfully extracted. However, the segmented hand contains numerous holes. The result of Figure 3e is cleaner than the result of Figure 3d, but the hand and the object are segmented together because they have hue values within the threshold range. The AND result in Figure 3f is

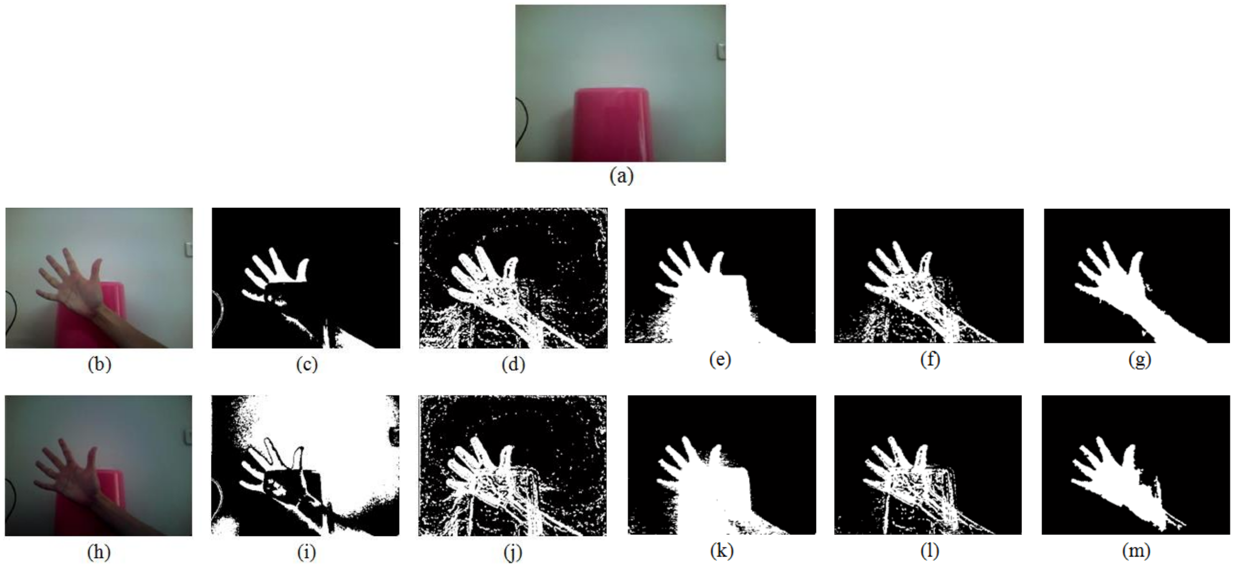


Figure 3. Hand segmentations on a plain background (a) background image, (b) current frame under constant lighting, (c) traditional method result, (d) UGV-based background subtraction, (e) skin color segmentation, (f) AND result, (g) after post-processing, (h) current frame under irregular lighting, (i) traditional method result, (j) UGV-based background subtraction, (k) skin color segmentation, (l) AND result, and (m) after post-processing.

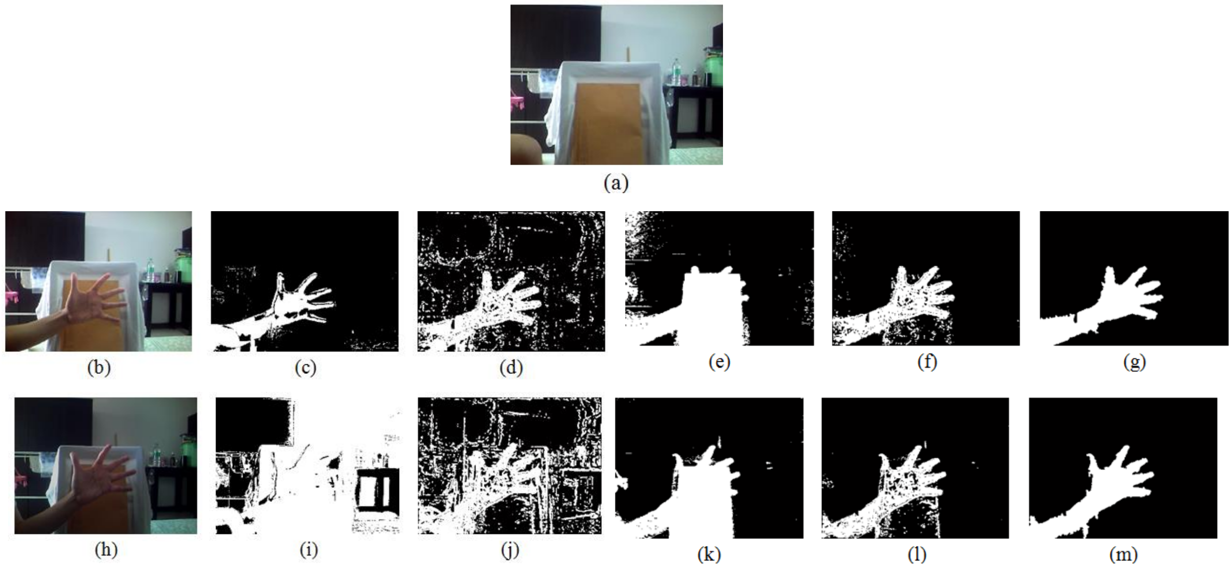


Figure 4. Hand segmentations on a complex background with skin color object (a) background image, (b) current frame under constant lighting, (c) traditional method result, (d) UGV-based background subtraction, (e) skin color segmentation, (f) AND result, (g) after post-processing, (h) current frame under irregular lighting, (i) traditional method result, (j) UGV-based background subtraction, (k) skin color segmentation, (l) AND result, and (m) after post-processing.

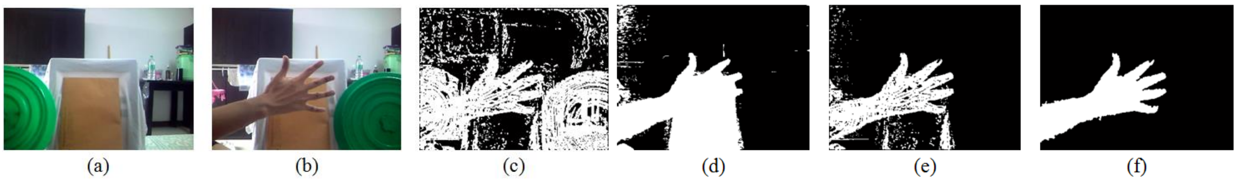


Figure 5. Hand segmentations on a complex background with a moving non-skin color green object (a) background image, (b) current frame, (c) UGV-based background subtraction, (d) skin color segmentation, (e) AND result, and (f) after post-processing.

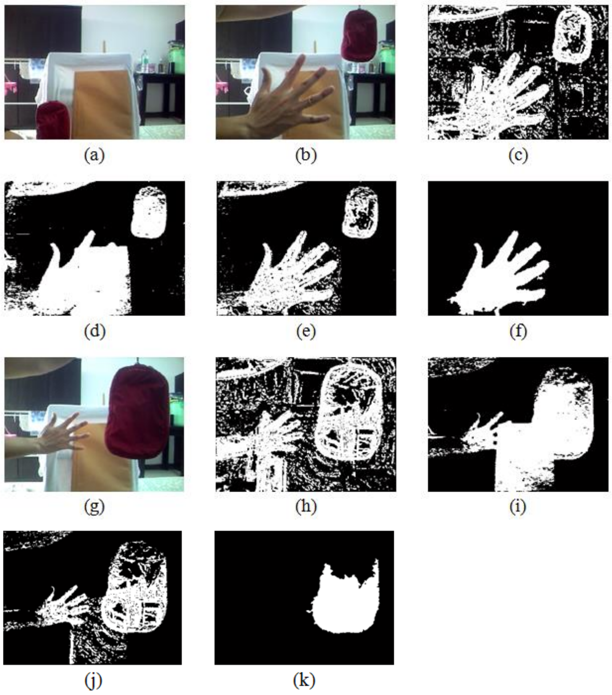


Figure 6. Hand segmentations on a complex background with a fixed skin color object and a moving red color object (a) background image, (b) current frame with the object moving away from the camera, (c) UGV-based background subtraction, (d) skin color segmentation, (e) AND result, (f) after post-processing (e), (g) current frame with the object moving toward the camera, (h) UGV-based background subtraction, (i) skin color segmentation, (j) AND result, and (k) after post-processing (j).

given by applying the AND operation to the binary images in Figures 3e and 3f. It shows that the scattered white pixels in the background are cleaned up, while the bright pixels which correspond to the pink object are mostly removed. Finally, we apply post-processing to the integration result as mentioned in the previous section, to remove small isolated components in the background and to fill up the holes inside the segmented hand region (Figure 3g). The room lighting conditions are changed from Figures 3b to 3h. The SAD result of Figure 3i is even worse than the SAD result of Figure 3c. The background subtraction results of Figure 3j are slightly different from Figure 3d, while the skin color segmentation result in Figure 3k is similar to Figure 3e. After applying the AND operation, the hand is segmented successfully, but there are a scatter of white dots in Figure 3l. Finally, we apply the same post-processing to eliminate the scattered white dots, as shown in Figure 3m. From the comparison between the two methods, the proposed method can segment the hand with a plain background under various lighting conditions.

We changed the background from a plain background to a complex background (Figure 4a). The background contains various stationary objects with different colors. We repeat the same sequence of patterns that shows a hand under constant lighting conditions (Figure 4b). The SAD can segment the hand, but there are holes inside the segmented hand since the hand has intensity values close to the skin color object in the background (Figure 4c). There are more isolated pixels in the background subtraction result since there is a larger gradient due to the objects (Figure 4d). The hand is unsuccessfully segmented because the skin color object in the background is also segmented (Figure 4e). After integrating by the AND operation, the segmented object is eliminated, and the isolated pixels are also cleaned up (Figure 4f). Finally, the segmented result is improved by applying post-processing the same as the processing with the plain background (Figure 4g).

We darkened the room to change the lighting conditions from Figure 4b to Figure 4h. In this situation, the SAD cannot perform hand segmentation due to changing lighting conditions (Figure 4i). The hand is segmented by the background subtraction successfully, but there are still the isolated pixels in the background (Figure 4j). The hand cannot be segmented by the color segmentation because of the skin color object (Figure 4k). Thus, the AND operation is used to select the true foreground (i.e. hand), as shown in Figure 4l. Figure 4m shows the final result of the segmented hand that is improved by applying the post-processing. It shows that the proposed method still can segment the hand with the complex background under various lighting conditions.

We then tested the proposed method in the complex background by including moving objects in the background, as shown in Figures 5 and 6. Figure 5 shows the result with a green object in the background. Figure 5a shows the background with a green object, which is moving. The green object moves from the left to right positions in Figure 5a to Figure 5b, while the hand exists in the image frame. The background subtraction fails to segment the hand because the object and the hand are considered as foreground (Figure 5c). By color segmentation, the object cannot be segmented since the object does not contain skin color (Figure 5d). Because of the color segmentation result, there is only the hand in the segmentation (Figure 5e). Finally, we applied post-processing to the AND result (Figure 5f).

The experimental result of the complex background with a fixed skin color and a moving red object is shown in Figure 6. Figure 6a shows the background image of this situation. The hand appears in the image frame while the red object moves from the lower left to upper right positions, as shown in Figure 6a and Figure 6b. The UGV-based background subtraction extracts both the hand and the object as the foreground (Figure 6c). In the color segmentation, the result is the same as the background subtraction result since the red object has the color values in range of the threshold value, as mentioned in Section 2.2 (Figure 6d). In the AND result, both the hand and the object are detected (Figure 6e). After post-processing, the proposed method considers the largest connected component as the target foreground (i.e. hand). Thus, the hand is successfully segmented (Figure 6f). Then, we change the hand's position and the red object's position in Figure 6g. Both background subtraction and color segmentation still extract the hand and the object (Figures 6h and 6i). After applying the AND operation, both the hand and the red object remain in this step (Figure 6j). Finally, the hand is lost after the post-processing is applied (Figure 6k), since the largest connected component is the red object. Therefore, the proposed method segments the red object wrongly.

3.2 Evaluation

We evaluate our proposed method using statistical performance and computational time. The proposed hand segmentation algorithm is executed with MATLAB, which has commands “tic” and “toc” to measure the execution time. The computational time is measured by selecting the minimum and the maximum times from 100 frames of a hand. The execution time results are presented in Table 1. The SAD has only one process, and segments the hand in 0.5 milliseconds per frame (2,500 frames per second). The proposed method has four processes, as mentioned in Section 2. It uses time to segment the hand within 46.2–182.1 milliseconds per frame (5–21 frames per second). From all processes, the proposed method mostly spends time on UGV-based background subtraction and post-processing. For UGV-based background subtraction, the proposed method spends time to compute UGV at every pixel. For post-processing, it requires different times to process, depending on the number of holes inside the segmented hand.

Table 1. Execution time of each process.

Process name	Computational times			
	Traditional method		Proposed method	
	Min (msec)	Max (msec)	Min (msec)	Max (msec)
Sum of Absolute Difference (SAD)	0.5	0.5	–	–
UGV-based background subtraction	–	–	24.2	62.1
Color segmentation	–	–	3.8	9.4
AND operation	–	–	0.5	1.2
Post-processing	–	–	17.7	109.4
Total process time	0.5	0.5	46.2	182.1

To evaluate the performance of the proposed method, the hand images are manually segmented and used as ground truths. The segmentation results are measured by a confusion matrix. The statistical performances, accuracy, sensitivity, specificity, and precision, are summarized in using the following statistics, Equation 4 to 7:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100\% \tag{4}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \tag{5}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \times 100\% \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \tag{7}$$

where TP (true positive) indicates the number of pixels within a successfully segmented hand, TN (true negative) indicates pixels in successfully segmented background, FP (false positive) denotes background pixels wrongly segmented as a hand, and FN (false negative) denotes hand pixels wrongly segmented as background.

Table 2 shows a numerical comparison between the traditional method (i.e. SAD) and the proposed method for both plain background and complex background, under varying lighting condition. The proposed method’s performances slightly drop when the lighting conditions are changed, while the traditional method’s performances, except sensitivity, rapidly drop when the lighting conditions are iregular. Note that mostly TN and FP in the traditional method are dramatically changed when the lighting conditions are irregular.

3.3 Real-time application

The proposed method is used as part of a one-box solution for remote control of an audio player by using hand gesture recognition (Dulayatrakul *et al*., 2015). Since this paper is about hand segmentation, the details about hand segmentation are explained in this section while the fingertip detection is described in Prasertsakul and Kondo (2015). The system is implemented by using an embedded board Raspberry Pi 2 Model B (Raspberry Pi, 2016) which has a 900 MHz quad-core ARM Cortex-A7 CPU, 1 GB of Memory, and Pi camera module, which has an OmniVision OV5647 sensor (Pi camera, 2016). Figure 7a shows the process flow of the proposed method on the Raspberry Pi. Since this application supports at most two hands, each hand is processed by each CPU core separately. Figure 7b shows an example of the system,

Table 2. Segmentation performances of the traditional method and the proposed method with plain background and complex background under constant and irregular lighting conditions.

Methods	Background + Lighting condition	Accuracy	Sensitivity	Specificity	Precision
Traditional method (SAD)	Plain + Constant	90.8%	41.2%	98.9%	86.4%
	Plain + Irregular	39.5%	43.8%	38.9%	10.2%
	Complex + Constant	96.1%	81.4%	98.1%	85.3%
	Complex + Irregular	39.4%	90.4%	31.8%	16.4%
Proposed method	Plain + Constant	98.3%	99.6%	98.1%	89.2%
	Plain + Irregular	96.9%	92.7%	97.5%	84.9%
	Complex + Constant	98.0%	97.3%	98.1%	87.7%
	Complex + Irregular	97.2%	90.5%	98.1%	87.3%

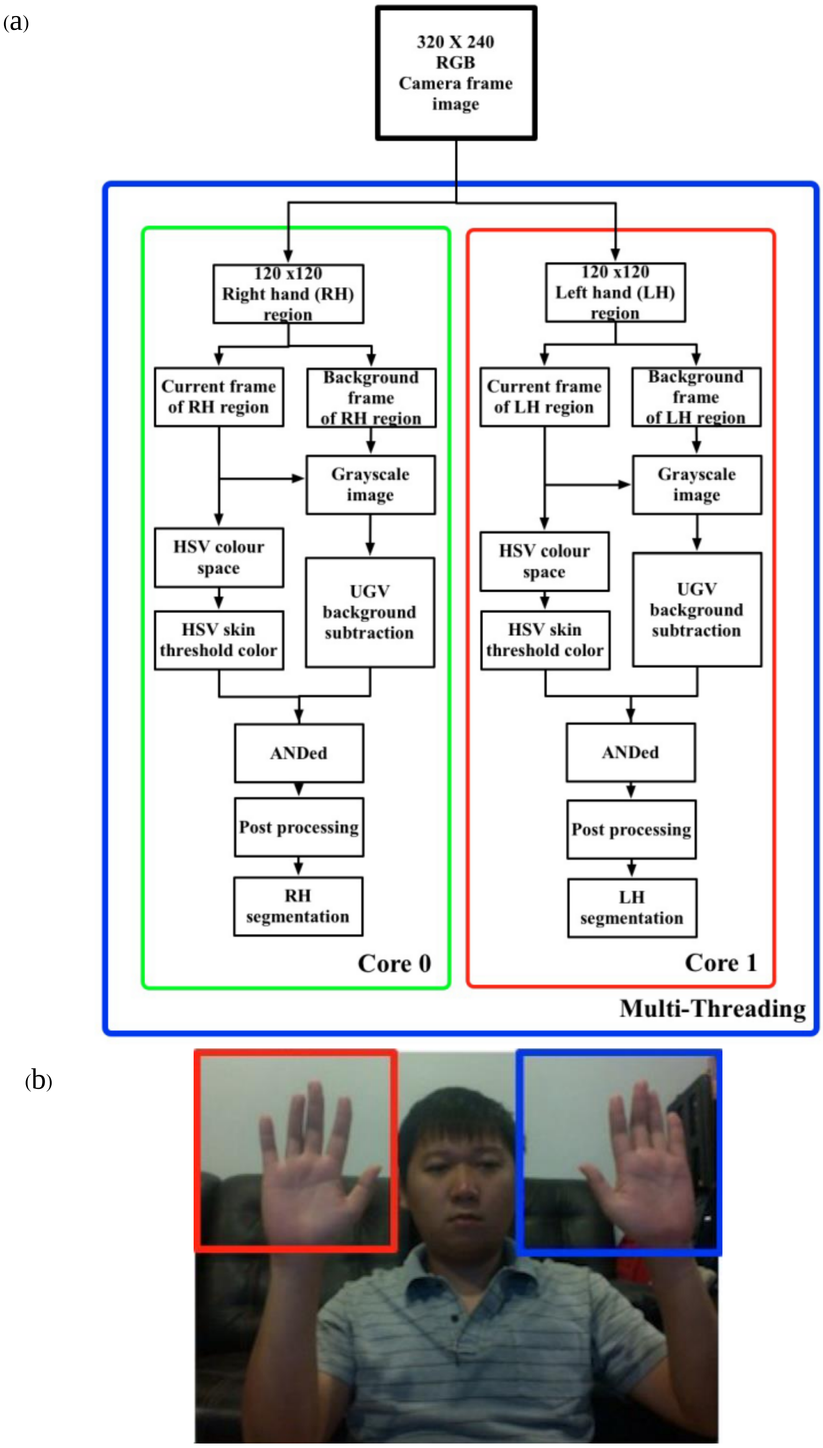

(a)
320 X 240 RGB Camera frame image
120 x120 Right hand (RH) region
Current frame of RH region
Background frame of RH region
Grayscale image
HSV colour space
HSV skin threshold color
UGV background subtraction
ANDed
Post processing
RH segmentation
Core 0
120 x120 Left hand (LH) region
Current frame of LH region
Background frame of LH region
Grayscale image
HSV colour space
HSV skin threshold color
UGV background subtraction
ANDed
Post processing
LH segmentation
Core 1
Multi-Threading
(b)

Figure 7. Hand segmentations on the Raspberry Pi (a) process flow and (b) two regions of interest for hand segmentation.

using the proposed method to segment and detect the hands. Each region of interest corresponds to each process in a CPU core, as mention previously. Due to the hardware specification, the input frame resolution is reduced to 320 by 240 pixels.

Figure 8 shows real-time hand segmentation, using the Raspberry Pi. Figure 8a is a background image, which is retrieved from the first frame of the image sequence. Subsequent frames under the same brightness level (Figure 8b) and under darker conditions (Figure 8f), with a hand in the foreground, are also shown for the UGV-based background subtraction. The hand is detected as the foreground correctly in both lighting levels (Figures 8c and 8g). As a result of the weak point in UGV-based background subtraction, the generated image contains noise in the background region. There is less error in the results from the skin color segmentation since hue is robust to dynamic lighting conditions (Figures 8d and 8h). Comparing the two results in both lighting levels, they are similar to each other. This signifies that the application can segment a hand under dynamic lighting conditions using the proposed method (Figures 8e and 8i). In addition, a Raspberry Pi can process hand segmentation at 42.7–46.4 milliseconds per frame (21–23 frames per second).

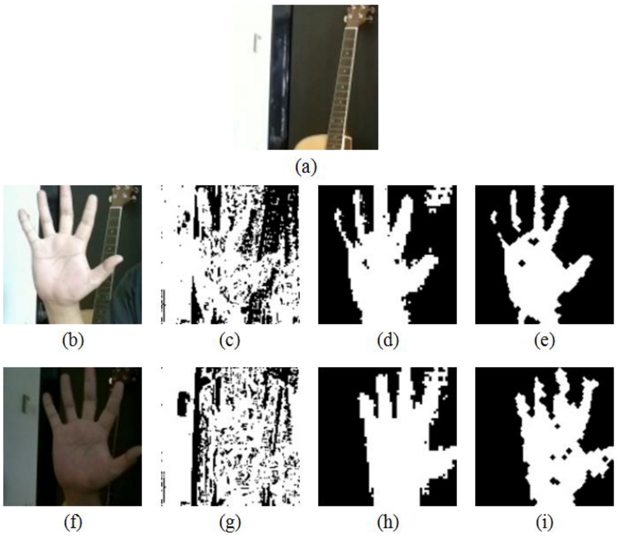


Figure 8. Real-time hand segmentations on Raspberry Pi (a) background image, (b) current frame under constant lighting, (c) UGV-based background subtraction, (d) skin color segmentation, (e) final result, (f) current frame under irregular lighting, (g) UGV-based background subtraction, (h) skin color segmentation, and (i) after post-processing.

4. Conclusions

In conclusion, this paper presents a hand segmentation method which is composed of background subtraction using unit gradient vectors (UGVs) and hue based skin color segmentation. The UGV-based background subtraction can effectively extract foreground, including a hand, even if there are skin-colored objects in the background. Meanwhile, hue based skin color segmentation extracts both hand and skin-colored objects in the background. A combination of the two methods can then segment only skin-color foreground, that is, a hand in this paper. A significant advantage of the proposed method is that it performs hand segmentation robustly under dynamic lighting conditions, compared with the traditional method. This is achieved because both UGVs and hue are invariant to varying image intensities, often caused by dynamic lighting conditions, and also internal functions of a camera, such as auto-gain control (AGC). Comparing the computational time, the proposed method requires more time, 46.2–182.1 milliseconds per frame (5–21 frame per second), in order to segment the hand robustly. We also employ the proposed method in a Raspberry Pi. This shows that this algorithm works well on a low-cost computer.

Acknowledgements

This research is financially supported by The Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NS TDA), Tokyo Institute of Technology, and Center of Excellence in Biomedical Engineering, Sirindhorn International Institute of Technology (SIIT), Thammasat University (TU).

References

Avinash, B. D., Ghosh, D. K., & Ari, S. (2013). Color hand gesture segmentation for images with complex background. *Proceedings of the International Conference on Circuits, Power and Computing Technologies*, 1127-1131.

Chaves-González, J. M., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2010). Detecting skin in face recognition systems: A colour study. *Digital Signal Processing, 20*(3), 806-823.

de Dios, J. J., & Garcia, N. (2003). Face detection based on a new color space YCgCr. *Proceedings of the International Conference on Image Processing*, 909-912.

Dulayatrakul, J., Prasertsakul, P., Kondo, T., & Nilkhamhang, I. (2015). Robust Implementation of hand gesture recognition for remote human-machine interaction. *7th International Conference on Information Technology and Electrical Engineering*, 247-252.

Jabri, S., Duric, Z., & Wechsler, H. (2000). Detection and location of people in video images using adaptive fusion of color and edge information. *Proceedings of the 15th International Conference on Pattern Recognition*, 627-630.

Javed, O., Shafique, K., & Shah, M. (2002). A hierarchical approach to robust background subtraction using color and gradient information. *Proceedings of the Workshop on Motion and Video Computing*, 22-27.

Kakumanu, P., Makrogiannis, S., & Bourbakis, N. (2007). A survey of skin color modeling and detection methods. *Pattern Recognition, 40*(3), 1106-1122.

Kondo, T. (2011). An image sequence segmentation method using gradient orientation information. *Proceedings of the SICE Annual Conference*, 34-36.

Musa, Z., Jumari, K., & Zainal, N. (2011). A method of human skin detection base on background subtraction and color enhancement. *IEEE Symposium on Business, Engineering and Industrial Applications*, 498-502.

Ogihara, A., Matsumoto, H., & Shiozaki, A. (2006). Hand region extraction by background subtraction with renewable background for hand gesture recognition. *International Symposium on Intelligent Signal Processing and Communications*, 227-230.

Pavlovic, V., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interacttion: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 677-695.

Pi camera. (2016, October 20). Camera module 1 specification. Retrieved from https://www.raspberrypi.org/documentation/hardware/camera/README.md

Prasertsakul, P., & Kondo, T. (2015). A new fingertip detection method using the top-hat transform. *Thammasat International Journal of Science and Technology*, *20*(3), 19-27.

Raspberry Pi. (2016, October 20). Hardware specification. Retrieved from https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

Storring, M., Kocka, T., Andersen, H. J., & Granum, E. (2003). Tracking regions of human skin through illumination changes. *Pattern Recognition Letters, 24*(11), 1715-1723.

Wang, X., Zhang, X., & Yao, J. (2011). Skin color detection under complex background. *Proceedings of the International Conference on Mechatronic Science, Electric Engineering and Computer*, 1985-1988.

Zhu, X., Yang, J., & Waibel, A. (2000). Segmenting hands of arbitrary color. *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 446-453.