Journal of the ACS, Vol. 10, May 2019

Real-time hand area segmentation for hand gesture recognition in desktop environment

Dr. Osama Shafik Elshehry
Department of Computer Science,
High Institute of Computers & Information Technology,
Shorouk Academy, Shorouk City, Cairo, EGYPT.
dr.osama.elshehry@sha.edu.eg

Abstract— the process of separating the hand area from a complex background, known as hand segmentation, is a prerequisite for any vision-based hand gesture recognition system. In some applications, only a rough estimate of the hand area is needed, but in other applications an exact segmentation of the hand, if possible, is needed. In this paper, three methods for extracting the hand from the background in real-time were tested. These methods use off-line learning of skin color; these methods are "Histogram Intersection", "Color Histogram", and "Skin Color Modeling and Adaptation". The three methods were tested using a video sequence of 100 frames using four different lighting conditions, with 400 frames in total. The different lighting conditions are fluorescent light only, a mixture of fluorescent and daylight, daylight without the sun light, and daylight with the sun light. These video streams were taken from the same person under the above different lighting conditions. A comparative study between the three methods was performed. The results showed that hand segmentation using "Skin Color Modeling and Adaptation" with Fluorescent light produced the best results among the three methods. The objective of our research is to accurately recognize the hand gestures, and then use it in many different applications. The work in this paper is the first stage in our research. The contribution in this paper is the comparative study between three different techniques that use skin-color modeling under different lighting conditions.

Keywords— Hand segmentation, Hand gesture recognition, Histogram intersection, Color Histogram, Skin Color Modeling and Adaptation, Human-computer interaction.

I. INTRODUCTION

Separating the hand from a complex background, which is known as hand segmentation, is a prerequisite for any vision-based hand gesture recognition system.

Image segmentation is the process of partitioning an image into constituent parts of objects. Many techniques have been used to achieve the image segmentation task. Most techniques might be classified into one of three major categories.

The first category uses motion analysis to achieve segmentation. Motion is an effective characteristic of image sequences that reveals the dynamics of scenes. The task of motion analysis remains a challenging and fundamental problem of computer vision [1].

The second category and the most used is skin-color modeling. The main idea of the techniques of this category is to build a model of the skin-color either off-line, on-line, or a combination of both on-line and off-line. The off-line skin model is built using patches of skin [2, 3] or using a large database of skin-color images [4]. The on-line skin model is built using a patch of skin during processing [5]. The off-line method is mostly used in off-line applications such as content-based retrieval applications [6] [7]. The disadvantage of the off-line method is that it is not reliable under different lighting conditions and with people with different skin colors. This is because there is no chance to get a patch of skin in run-time and there is no control over the background or the lighting conditions. The on-line method is preferred in real-time applications because of the ability to get a patch of skin during processing. The combination of both methods is also possible by building an off-line model, then improving that model during processing to be invariant under different working environments like different lighting conditions, different skin-colored people,

different camera parameters, and different backgrounds [8]. Using skin-color is a popular approach for hand and face segmentation because it is the simplest attribute of human skin.

Using skin-color as a feature for tracking the human hand has several advantages. Processing color is much faster than processing other features. Under certain lighting conditions, color is orientation invariant.

Tracking human hands using color as a feature has several problems. First, the color representation of a hand obtained by a camera is influenced by many factors such as ambient light, object movement, etc. Second, different cameras produce significantly different color values even for the same person under the same lighting condition. Finally, the colors human skin differ from individual to another. To use color as a feature for hand tracking, these problems have to be solved [3].

The third category is based on Background Modeling [9] [10] [11] [12]. A model of the background is built during processing. Then a subtraction of the incoming frame from the model reveals changes in the scene, these changes represent the new object(s) which appeared in the scene or/and the old objects which disappeared from the scene. This new set of objects might be further analyzed and classified. The advantage of the techniques of this category is that they are not only restricted to human limbs but also can be used to segment any objects like animals, cars, etc.

Although most of the existing techniques belong to one of these categories, any combination of these techniques is also possible. In some applications, only a rough estimate of the hand area is needed, but in this work an exact segmentation of the hand, if possible, is needed. For this reason, three methods for extracting the hand from the background in real-time were tested. These methods belong to the second category and use off-line learning of skin color; these methods are Histogram Intersection, described in section 3.1, Color Histogram in section 3.2, and Skin Color Modeling and Adaptation in section 3.3.

The three methods were tested using a video sequence of 100 frames for four different lighting conditions, with a total of 400 frames. These video streams were taken for the same person. The results showed that hand segmentation using the last method produced the best results among the three methods. These results are presented and discussed in section vi.

We did not try different people with different skin color because our experimental results showed that trying to build a generic color model that works with different skin-colored people does not produce good results. This might be useful in some applications if a rough estimate of the hand is sufficient, which is not sufficient for our work.

II. IMAGE ACQUISITION AND HARDWARE ARRANGEMENT

The real-time video stream was captured using a Sony SSC-DC18P camera connected to a Silicon Graphics 320 machine with a single 550MHz PIII processor and 512MB of memory. A library that provides the facility to grab images from different sources was used. The supported sources are Video for Windows (VFW) video files, AVI files, and Silicon Graphics media. The camera is connected to the machine through a frame grabber which produces 25 frames/second, with a frame size of 188x144 pixels. The frame grabber can produce a stream of images using either the YUV or the RGB color models. The normalized (r, g) color model was used at the beginning of this research, because this model contains chromatic information only and has some invariance with respect to illumination as in [13]. However, for the sake of speed, the YUV color model is used instead of the normalized (r, g) color model. The YUV model separates the chromatic information (U, V) from the lightness (Y).

The hardware arrangement shown in figure 1 is designed so that the camera is situated over the desk looking vertically downward. This arrangement has two advantages over the web-camera position, in which the camera is mounted over or besides the monitor and facing the user. The first is that it avoids the distractions of background movements, and the second is that it ensures user comfort by allowing the hand to rest over the desk as in the usual mouse position.

All the algorithms in this work were implemented using raw C++ programming language without using and additional libraries. Microsoft Foundation Classes (MFC) was used for Graphical User Interface (GUI) on Microsoft Windows operating system.

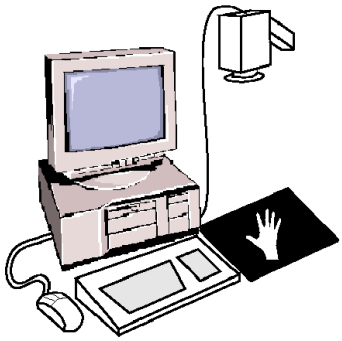

Figure 1: The hardware arrangement

III. HISTOGRAMS INTERSECTION

In this method, a color histogram is obtained by discretizing the image colors and counting the number of times each discrete color occurs in the image. Histograms are invariant on the viewing axis to translation and rotation. They change only under change of viewing angle, change in scale, and occlusion. To build a model of the human skin, a histogram representing a small patch (8X8) of an image of a human hand is used. This process may be considered as the training phase. During the processing phase, the incoming image is examined, patch by patch using the histogram intersection-matching algorithm described in [13]. The match score M between histograms p and q is defined as:

$$M = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\min(H_{i,j}^{P},H_{i,j}^{q})}{\sum_{i=1}^{n}\sum_{j=1}^{n}H_{i,j}^{P}} \qquad (1)$$

Where $H_{i,j}^{P}$ is the model histogram for a patch of size n x n and $H_{i,j}^{q}$ is a histogram representing a patch of size n x n of the current frame. The score M takes values from 0 to 1. Those patches with a score above a certain threshold are supposed to be skin-colored, and are assumed to belong to the hand; otherwise, the patch is considered as a background. The technique can be summarized as follows:

- Choose a patch of skin.
- Partition the (U,V) components into a number of bins, for example (8x8).
- Build the histogram.
- Go through the image and match using equation (1).

The results, which have been achieved using different skin patches were taken from different location of the hand, are shown in Figure 2.

To get the best of this technique it is required to answer three questions.

First, what is the best bin size? Second, what is the best patch size? Third, what is the best threshold? There are no analytical methods to answer these questions. It can be answered only on the light of the experimental results.

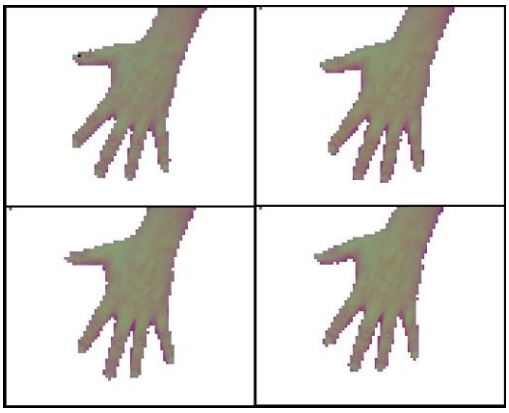

Figure 2: Histogram intersection results using different skin patches taken from different areas of the same person's hand

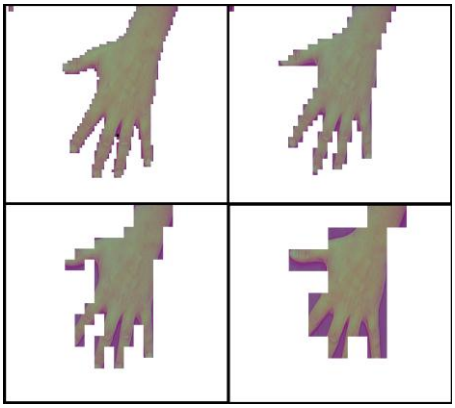

Figure 3: Histogram intersection results using patches of different size. From top to bottom and from left to right, these sizes are 4x4, 8x8, 16x16, and 32x32

Figure 3 shows the results of using different skin patch sizes, Figure 4 shows the results of using different bin sizes, and finally, Figure 5 shows the result of using different thresholds. From figures 2, 3, 4 we can conclude that choosing the patch size, the bin size, and the threshold has a great effect on the result. It was found that 8-bin histogram with 4x4 patch size gave the best results. The best threshold was found by trying different thresholds and then choosing the one giving the best results.

IV. COLOR HISTOGRAMS

Segmentation with the color histogram relies on the assumption that homogeneous objects, such as skin, exist as clusters in the color space. A two-dimensional histogram is used to represent the skin tones. By using the two parameters of a color system, which do not correspond to intensity or illumination, the histogram should be more robust to changes in illumination.

To build the color histogram, some patches representing the skin are used as a training set. For each pixel in the skin image, the appropriate cell in the histogram is incremented. Each cell in the histogram should have a value equal to the number of skin pixels having that combination of color components. After all the patches have been processed, the histogram is normalized.

The normalization is performed by dividing the value in each cell by the largest cell value, or by dividing by the total number of the processed pixels. A histogram representing a part of a hand skin is shown in figure 6.

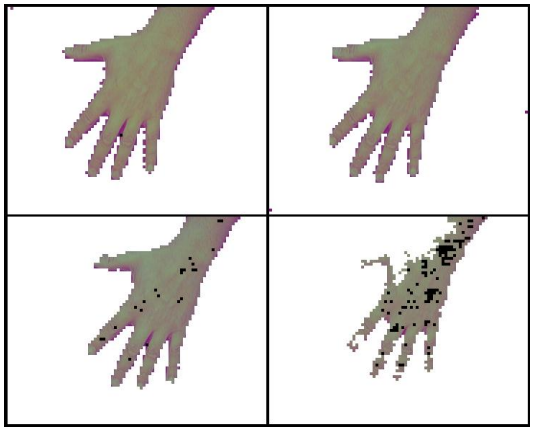

Figure 4: Histogram intersection results using channels of different bin size. From top to bottom and from left to right, the bin sizes are 4,8, 6, and 32

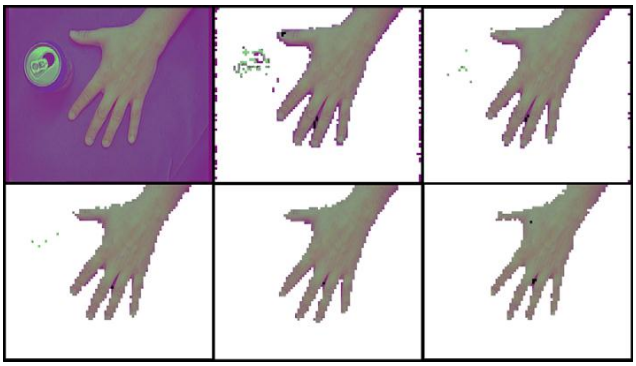

Figure 5: Histogram intersection results using different threshold values. The original image is on the top left. The rest of images from top to bottom and from left to right were produced using threshold values of 0.1, 0.3, 0.5, 0.7, and 0.9 respectively

To perform skin detection, for each pixel in the image, the color values are used to find the appropriate cell in the histogram. If this value is greater than a threshold, the pixel is marked as skin. Otherwise, the pixel is considered non-skin.

Figure 7 shows some results of applying a skin histogram to extract skin pixels using some skin patches taken from the same person. Although the hand shown in figure 6 belongs to the same person, the performance is dramatically affected by varying lighting conditions. This is not a surprise, because although the skin color is clustered in the color space, it is clustered in different locations in the color space, even for the same person, under different lighting levels. Another known problem of this technique is that it is not possible to use the same histogram for different people, especially with different skin color.

Trying to build a generic histogram using skin patches taken from different people does not solve the problem. Because the resulted histogram will be flattened as shown in figure 8.

Experimental results showed that the generality archived by this histogram affects the performance of this technique as shown in figure 9. There are two solutions to this problem. The first solution is to tolerate the environment. This can be achieved by training the system to only one user at a time, and in certain lighting conditions. The second one is to adapt the system to varying conditions. Although the first solution is easier, it puts some constrains. These constraints represent a limitation on the system capabilities. On the other hand, the adaptation of the system is more realistic, but it is difficult to achieve.

In the next section, a technique that uses adaptation is presented. It models the foreground (the hand), and adapt it to the varying environment.

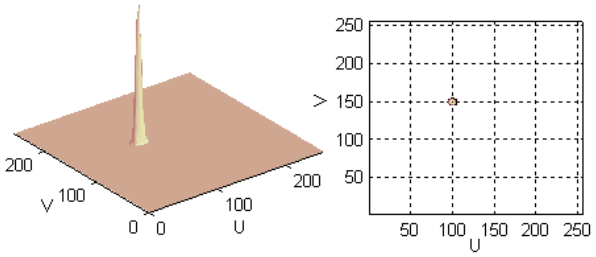

Figure 6: Histogram representing skin patch

Figure 7: Results of applying color histogram

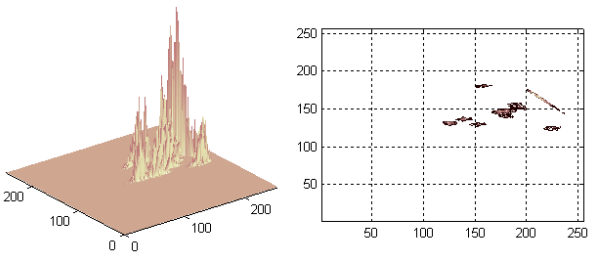

Figure 8: Histogram representing skin patches from the same person under different lighting conditions

Figure 9: Result of applying generic histogram

V. SKIN-COLOUR MODELLING AND ADAPTATION

Although the skin-colors of different people are clustered in different locations in the chromatic color space, the shape of the histogram remains similar. By closely investigating the skin color cluster, it was found that the skin colors of different people under different lighting conditions in the chromatic space have Gaussian distributions [3]. Therefore, the skin color distribution can be represented by a Gaussian model with$N = (m, \Sigma^2)$, where $m = (\overline{u}, \overline{v})$ with

$$\overline{u} = \frac{1}{N}\sum_{i=1}^{N} u_i \qquad (2)$$

$$\overline{v} = \frac{1}{N}\sum_{i=1}^{N} v_i \qquad (3)$$

And

$$\Sigma = \begin{bmatrix} \sigma_{uu} & \sigma_{uv} \\ \sigma_{uv} & \sigma_{vv} \end{bmatrix} \qquad (4)$$

Where

$$\sigma_{ab} = \frac{1}{N}\sum_{i=1}^{N}(a - \overline{a})\,(b - \overline{b}) \qquad (5)$$

The algorithm for creating the skin-color model is as follows:

- Take an image containing the hand skin.
- Select the skin-colored regions.
- Estimate the mean and the covariance of the color distribution in chromatic color space based on equations 2, 3, and 4.
- Substitute the estimated parameters into the Gaussian distribution model as shown in equation 6

$$f(x) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)`\Sigma^{-1}(x-\mu)} \qquad (6)$$

Where $x = \binom{u}{v}$, $\mu = \binom{\mu_u}{\mu_v}$, and Σ as in equation 4.

Since the model has only six parameters, it is easy to estimate and adapt them to different people and different lighting conditions. The adaptive method can be applied to two situations where either the tracking object or lighting condition can be changed. The mean vector is assumed as a linear combination of the last N mean values.

$$\hat{\mu} = \sum_{k=1}^{N} a_k \, \mu_k \qquad (7)$$

Where $\hat{\mu}$ is the estimated mean vector; $a_k \leq 1$, k = 1. . . N are weighting factors; μ, k = 1. . . N, are the previous mean vectors. The covariance matrix is assumed to be a linear combination of the last N covariance matrices

$$\hat{\Sigma} = \sum_{k=1}^{N} \beta_k \, \Sigma_k \qquad (8)$$

Where $\hat{\Sigma}$ is the estimated covariance matrix; $\beta_{k \leq 1, \ k=1,\ldots,N}$ are weighting factors;$\hat{\Sigma}$, $k = 1, \ldots, N$, are the previous covariance matrices. The weighting factors α and β in equations 7, and 8 respectively determine how much the past parameters will influence current parameters. For example, setting α, and β to $\frac{1}{N}$ lets the parameters within window N make the same contribution to the current parameter estimations. The windows size N determines how long the past parameters will influence the current parameters. The adaptive speed will decrease as N increases. The result of filling a histogram using the modeling of the skin color as a bi-Gaussian distribution is shown in figure 11. The sample skin used and the histogram built from the bi-Gaussian distribution is shown in figure 12.

Although the achieved result is promising, the question that arises here is how can the system get a part of the skin that can be used to model? We can conclude that this method is very promising, reliable, and fast.

It can be used in the system if we can find a method to get skin patch at run time. This can be achieved by allowing the user to interactively choose a patch of skin while displaying the video stream. The only problem with modeling the skin color using Gaussian distribution is that the Gaussian distribution smooths the histogram, which affects the performance slightly.

Figure 12 shows two histograms from the same skin patch. The top histogram was produced directly, and the bottom one was produced using bi-Gaussian.

Figure 10: The result of applying histogram built from the bi-Gaussian distribution: the original image is on the left, and the result on the right.

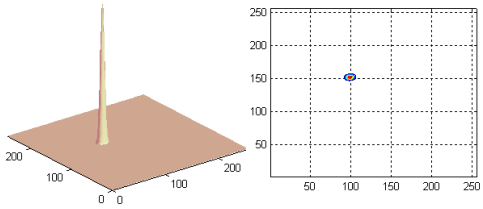

Figure 11: A histogram built from bi-Gaussian distribution.

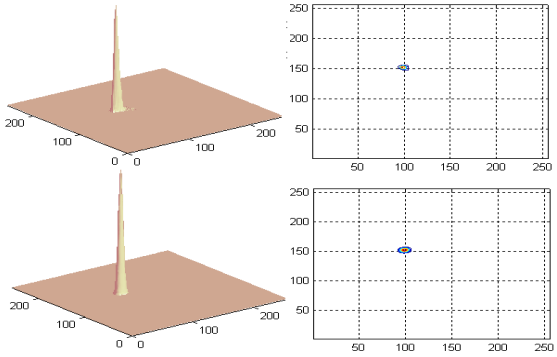

Figure 12: The histogram produced from Gaussian on the top and the histogram produced directly on the bottom

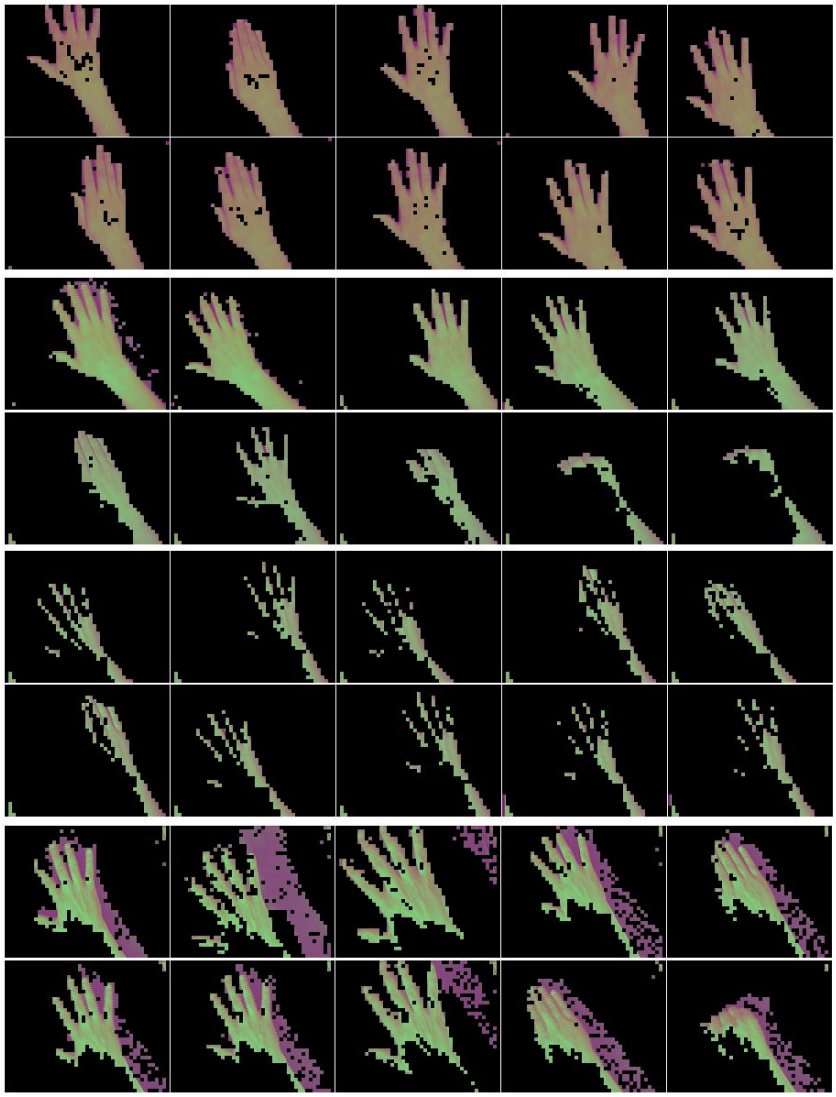

Figure 13: Results of applying histogram intersection under different lighting conditions. The four sets represent different lighting conditions. These lighting conditions from top to bottom are Fluorescent only, the mix of Fluorescent and daylight, daylight only without the sun, and the sun only.

From the results, we concluded that "Skin Color Modeling and Adaptation" technique achieved the best results among the three techniques. Another conclusion is that the best results for all methods were achieved when using the Fluorescent light only. This is because Fluorescent light provides more homogenous lighting than the other lighting conditions.

For future work, we will try the techniques that depend on background subtraction and compare it to the techniques provided in this work,

VI. CONCLUSION AND FUTURE WORK

The presented techniques were tested under four different lighting conditions. These lighting conditions are:

- Fluorescent light only
- A mix of Fluorescent and daylight
- Daylight without the sun light
- Daylight with the sun

Four videoed streams were captured and saved for testing purpose. Each stream consists of 100 frames. These video streams were used to compare the performance of the presented

techniques under the four different lighting conditions. A patch of skin was taken from the testing streams offline. Figures 13, 14, and 15 show sample frames for each lighting conditions for each technique. From each 100 testing frames for each lighting conditions, 10 frames are shown in the figures. These frames represent the 10th, 20th,..,100th frame in each stream. A total of 40 frames are shown in each figure. The first 10 frames represent the Fluorescent only, the second 10 frames represent the mix of Fluorescent and daylight, the third 10 frames represent daylight only without the sun, and the last 10 frames represent the sun only lighting condition.

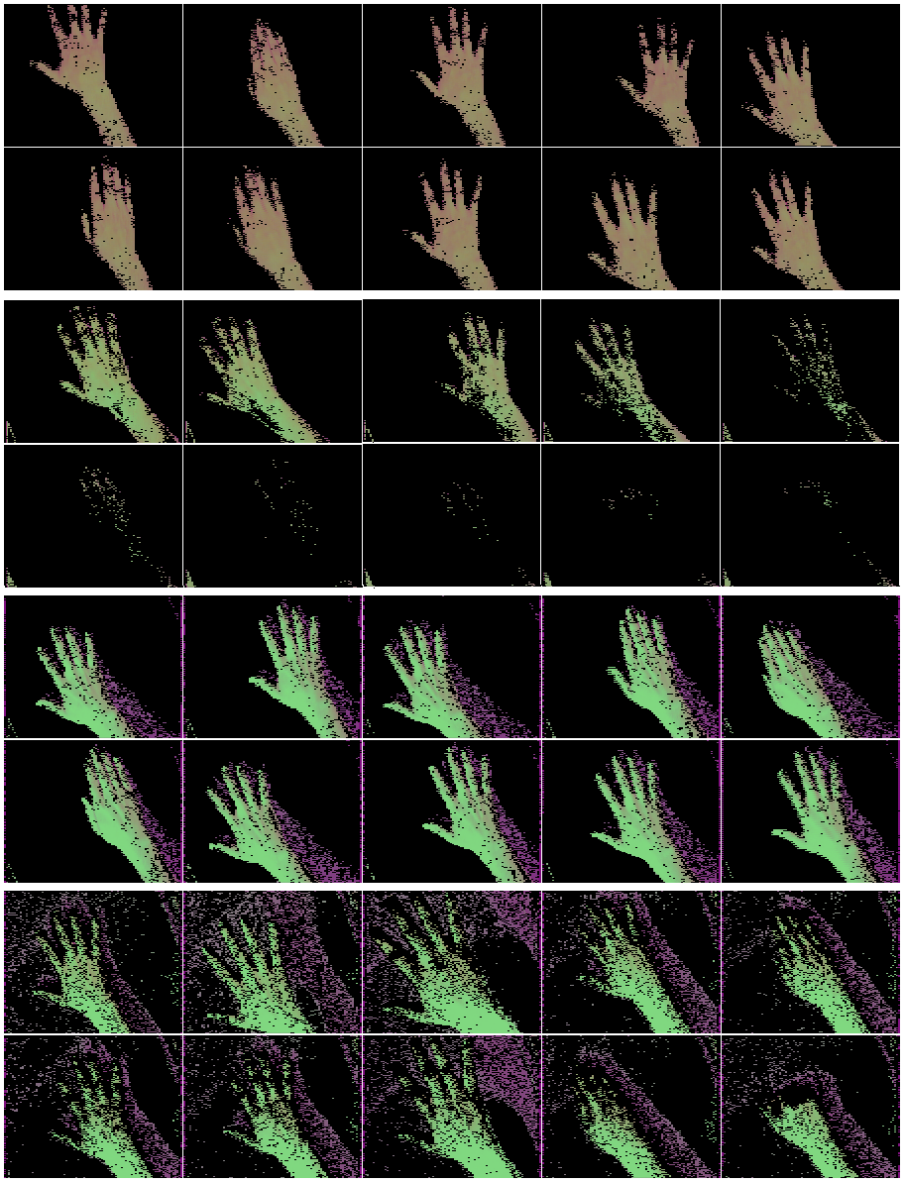

Figure 14: Results of applying color histogram under different lighting conditions. The four sets represent different lighting conditions. These lighting conditions from top to bottom are Fluorescent only, mix of Fluorescent and daylight, daylight only without sun, and sun only.

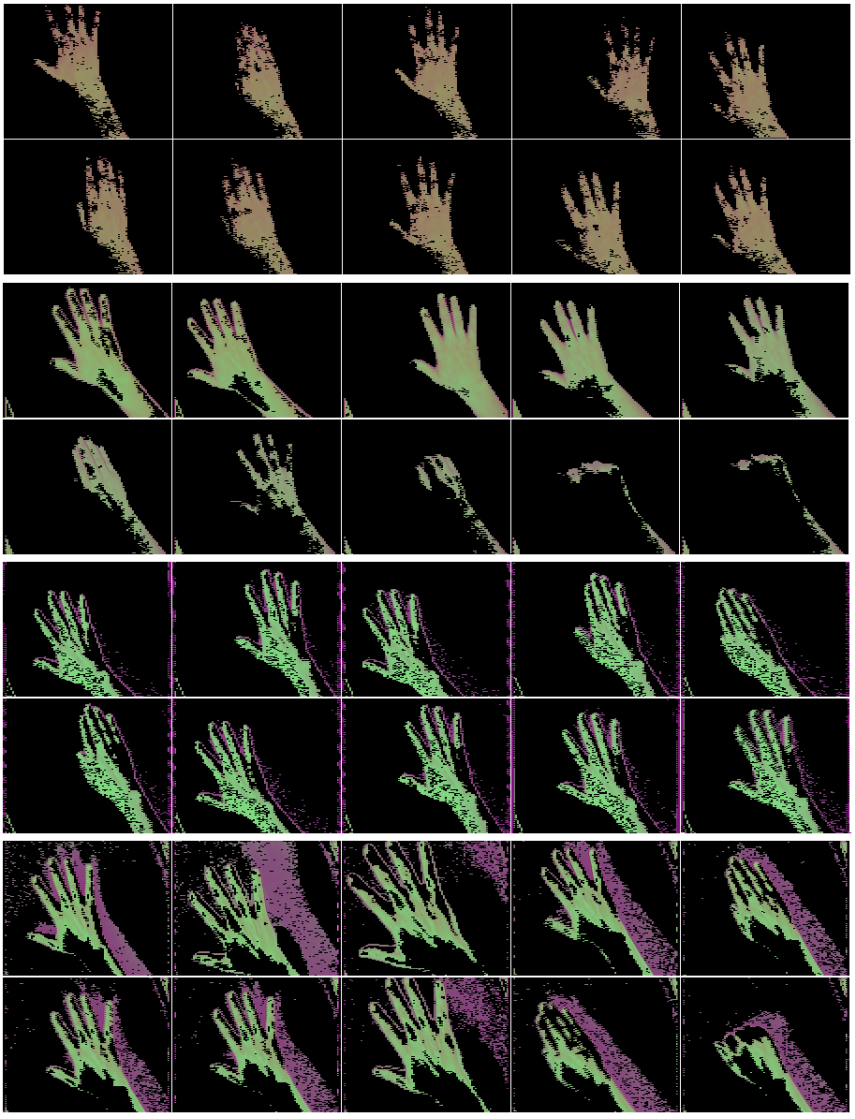

Figure 15: Results of applying color modeling and adaptation under different lighting conditions. The four sets represent different lighting conditions. These lighting conditions from top to bottom are Fluorescent only, a mix of Fluorescent and daylight, daylight only without the sun, and the sun only.

References

[1] B. B Jähne and H. Haussecker, Computer vision and applications, Academic Press, 1999.

[2] A. K. Y. M. a. M. I. S. Tsuruoka, "Extraction of hand region and specification of finger tips from color image," in *Virtual systems and multiMedia*, Geneva, Switzerland, 1997.

[3] J. Yang, W. Lu and A. Waibel, "Skin-Color Modeling and Adaptation," 1997.

[4] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81-96, 2002.

[5] S. Kawato and J. Ohya, "Automatic skin-color distribution extraction for face detection and tracking," in *Signal Processing Proceedings, 2000.. 5th International Conference on*, 2000.

[6] D. A. Forsyth and M. M. Fleck, "Automatic detection of human nudes," *International Journal of Computer Vision*, vol. 32, no. 1, pp. 63-77, 1999.

[7] D. A. Forsyth, M. Fleck and C. Bregler, "Finding naked people," *International Journal of Computer Vision*, 1996.

[8] D. Saxe and R. Foulds, "Toward robust skin identification in video images," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, 1996.

[9] S. J. McKenna, S. Jabri, Z. Duric and H. Wechsler, "Tracking interacting people," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000.

[10] K. Karmann and A. von Brandt, "Moving object recognition using an adaptive background memory," in *Time-Varying Image Processing and Moving Object Recognition,2, 1990.*, 1990.

[11] K. Toyama, J. Krumm, B. Brumitt and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999.

[12] Q. Huang, B. Dom, N. Megiddo and W. Niblack, "Segmenting and representing background in color images," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 1996.

[13] Qinghe Zheng, Xinyu Tian, Shilei Liu, Mingqiang Yang, Hongjun Wang and Jiajie Yang , "Static hand gesture recognition based on Gaussian mixture model and partial differential equation," *IAENG International Journal of Computer Science* , pp. 569-583, 2018.

[14] M. S. a. D. Ballard, "Color indexing," *IJCV*, vol. 7, no. 1, p. 11–32, November 1991.