Indonesian Journal of Informatics Education

Research Paper ISSN: 2549-0389

Hand Detection on HSV Color Space Model and Syntactic Extraction of Fingertip by Thinning Method for Hand Gesture Recognition

Yusfia Hafid Aristyagama
Department of Computer and Informatics Education
Universitas Sebelas Maret
yusfia.hafid@staff.uns.ac.id

Febri Liantoni
Department of Computer and Informatics Education
Universitas Sebelas Maret
febri.liantoni@gmail.com

Nurcahya Pradana Taufik Prakisya
Department of Computer and Informatics Education
Universitas Sebelas Maret
nurcahya.ptp@staff.uns.ac.id

Abstract:

In the discussion of computer vision, detection and recognition is an interesting topic to discuss. Advanced computer vision technology requires a high-level interaction method above the text-based console interaction. Hand detection and gesture recognition is one of the interaction cases in computer vision. In this study, an experiment of hand detection and syntactic hand gesture recognition method are discussed. HSV (Hue Saturation Value) space color model is used as the basis of hand detection and segmentation. Then, the thinning method is used to get the endpoint features of each fingertip.

The proposed design is designed to meet with real-time video processing. The experiment intended to find some issues usually happened when the ZS thinning method is used to gain detection and recognition. The result shows that the proposed design can detect and recognize some gestures, but unstable hand movement may lead to a fault called the extra endpoint. In this research, extra endpoints are considered as a challenge that must be anticipated when using the thinning method, especially the ZS algorithm to perform syntactic hand gesture recognition.

Keywords: Endpoint, Hand Gesture, HSV, Thinning, Recognition

DOI: http://dx.doi.org/10.20961/ijie.v5i2.51693

Introduction

Detection and recognition are interesting topics to discuss in the field of computer vision. An advanced human-computer interaction technology needs a higher level of interaction style than console text-based interaction. In this case, one interesting topic to discuss is hand detection and hand gesture recognition as the base of those interaction models. Both hand detection and hand gesture recognition are important techniques to do an advanced interaction model. In real-time processing, the need for fast detection and recognition method is inevitable. In this case, every implemented algorithm must be able to do fast pixel-based processing. Implementing fast pixel-based processing, the designed system should be able to produce a high frame rate.

Hand detection and recognition can be implemented in various fields of Information and Communication Technology. For example, Sutoyo et al. (2015) utilized the human hand gesture as a method to operate computers for presentation purposes. Makahaube et al. (2021) also used the hand gesture to implement the interaction between human and automated education service kiosks to reduce the direct interaction between humans and the machine during the Covid-19 pandemic era.

In the implementation of the hand gesture recognition process, several methods can be used to do gesture recognition. For example, Silvia & Husni (2014) used hand contour information to do the detection processes. Huu & Phung Ngoc (2021) use a support vector machine and histogram-oriented gradient to do the detection and classification of hand gesture recognition. Eshitha & Jose (2018) used an artificial neural network to do hand gesture recognition. Both researches by Eshitha & Jose (2018) and Huu & Phung Ngoc (2021) used machine learning to implement hand gesture detection and recognition. The syntactic method especially, the thinning method also can be implemented as a method to capture the hand gesture pose due to the output characteristic of the thinning algorithm itself. The thinning algorithm can give a one-pixel width image generated from the original binary image. It can be seen in the example given by Sudarma & Putu Sutramiani (2014) which use thinning method to clarify the curvature of Balinese scripts. Therefore, the thinning algorithm should be able to extract the one-pixel width skeletal information of hand gestures from the proper binary images too. Moreover, some of the thinning algorithms have parallel nature such as the one proposed by Zhang & Suen (1984). It can be executed in parallel which may result in a faster process.

In this research, we proposed the use of a combination of several techniques that can be implemented to do hand detection and some hand gesture recognition in real-time video processing in detail. In section I, the research background detail has been explained already. Based on the literature review, every related work is discussed in section II. Meanwhile, section III discusses the contribution of this research. Section IV discusses the method to perform the research. Section V describes the design of the system. The results of the experiment are discussed in Section VI. Section VI discusses the conclusion of the research.

Related Work

In the past research, several background-foreground segmentation algorithms were introduced. These algorithms can be used as a hand detection method. Kim et al. (2005) proposed a codebook segmentation method. This algorithm needs a training phase in every pixel to do detection. This training phase takes advantage of the background's quasi-periodic movements to build a background model. ROI (Region of Interest) detection can be done by comparing every pixel to those background models. A clear example of quasi-periodic movements is the periodic movement of leaves on the tree. Periodically they are going up and down. Sometimes, they move to the left and right because of the wind. Sobottka & Pitas (1998) describes the use of HSV (Hue Saturation Value) color space to do detection. It can be implemented by defining the search region. Proper parameter initialization is needed to define the best search region in a fixed range. In this case, ROI must be in a fixed range of HSV color too. Meanwhile, Santosh et al. (2013) discuss the use of GMM (Gaussian Mixture Model) to detect a moving object. So, GMM can be implemented to detect pedestrians or running a car on the toll road. It is because they never stop to move, so those movements can be detected by GMM. In the case of hand detection, skin color has a fixed range of search regions. Those fixed ranges of search regions are the main feature of skin that can be detected through proper search region parameter initialization. Based on those skin color properties, in this research, we would like to use a method discussed by Sobottka & Pitas (1998). Compared to the other segmentation method mentioned earlier, HSV color space segmentation is the most proper method to do hand detection. Sobottka & Pitas (1998) prove that HSV color space segmentation is able to detect the skin color of the face in a fixed range region. By that evidence, there is a high possibility that it can be used to detect the skin color of the hand too. In that case, HSV color space segmentation will be used as a hand detection method in the next discussion.

Thinning algorithms can be classified into three categories (L. Lam & Suen, 1992). There is sequential, parallel, and non-iterative thinning. The most cited work of thinning method can be found in literature about a fast parallel algorithm for thinning digital patterns (Zhang & Suen, 1984). In this research, we will use the term of Zhang Suen (ZS) algorithm to mention those aforementioned works. ZS algorithm is belonging to the parallel thinning algorithm. It means that every pixel can be processed in parallel. If the hardware supports parallel programming, pixel processing can be done quickly. In this case, the hardware resources can be exploited to get a good performance. ZS algorithm is simple, efficient, and resistant to noise. Given a binary image, this algorithm is able to give a one-pixel width horizontal and vertical skeleton as the result. But on diagonal shape, it can't produce a pure one-pixel width skeleton. Based on the previous experiment which was done by Lam & Suen (1995), the ZS algorithm can perform fast image processing. The research that was done by Shimizu et al. (2000) and Qin et al (2013), proposes the use of thinning algorithm to get some representative skeleton pattern of character. The thinning algorithm can be used not only for the binary image of character pattern recognition but also for the other binary images. For example, literature by Abu-Ain et al (2013) proposed a thinning algorithm and use it to perform thinning for some binary images.

In the case of hand gesture recognition, the use of the thinning method is supported by the existence of the endpoint feature. By a thinning method, a representative skeleton can be produced from a representative binary image. A representative skeleton will give correct information about the existence and the position of an object's endpoints. A classification of an object can be deduced from the existence of an endpoint pattern. From the previous explanation, the use of thinning algorithm gives several advantages, especially for the ZS algorithm. Based on the aforementioned information, the ZS algorithm will be used to do a thinning on the binary image of the hand gestures as the ROI. The only thing needed from the skeleton is the endpoints information that will be used to do classification. So, the diagonal pixel width problem of ZS thinning will be ignored. As long as the algorithm can produce the endpoint and do fast image processing, the other feature can be ignored. In this research, endpoint features can represent fingertips' position.

Thinning method alone cannot be used to perform hand gesture detection and recognition. Some several methods and algorithms can help to improve the thinning result during the thinning process for example HSV color space segmentation, ZS algorithm itself, morphological operation, and gaussian blur.

HSV Color Space Segmentation

HSV color space Hexa core cone-shaped as illustrated in (left). Hue (H) is represented as angle, the purity of colors is represented as saturation (S), and the degree of grayscale is represented as value (V) (Sobottka & Pitas, 1998). As illustrated in Figure 1, H value varies from 0^0 to 360^0, S varies from 0 to 255. It can be normalized into 0 to 1. V also varies from 0 to 255 too. So, its range can also be normalized into 0 to 1. Search region represents a fixed range region that is constrained by the maximum and minimum value of each needed parameter variable.

(Sobottka & Pitas, 1998) did an extensive experiment on HSV segmentation. $S_{min} = 0.23$, $S_{max} = 0.68$, $H_{min} = 0^0$, and $H_{max} = 50^0$ are the result of the experiments. Bvv values signing those values as input parameters, white skin as well as human sallow skin of human can be detected. V_{min} and V_{max} values are ignored. But those parameters are not tested yet on the other skin color.

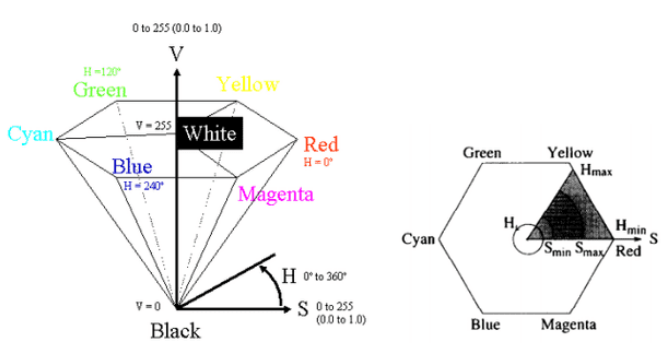


Figure 1. HSV Color Space (left) and search region (right)

Based on our experiment, those assigned parameter values cannot handle brown skin color perfectly. Given that condition, in the next discussion, all of the defined parameters will be assigned manually depending on the skin color to get a better detection result of ROI. The overall detection process can be illustrated as a flowchart in Figure 2. The input image must be in the form of HSV color space.

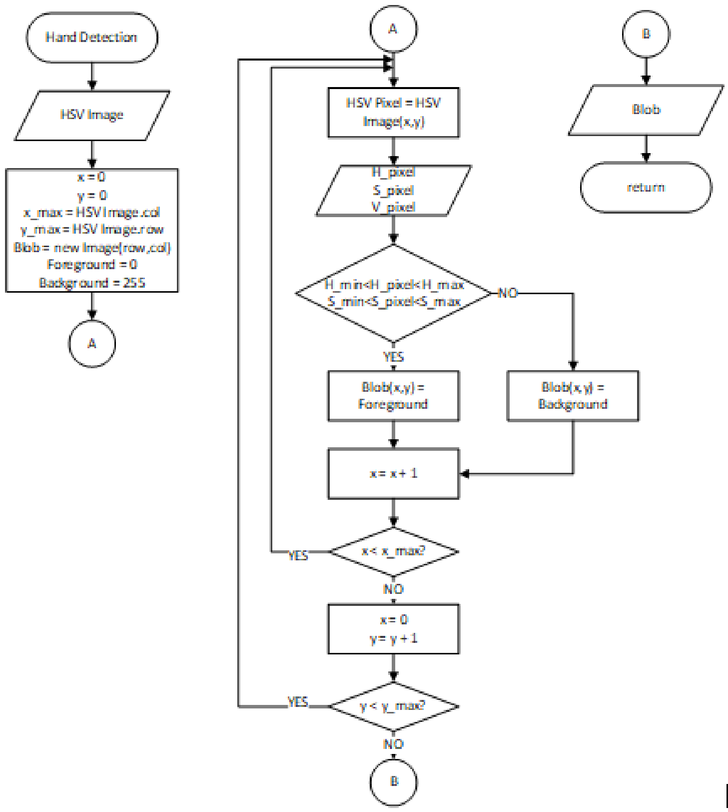


Figure 2. Hand segmentation flowchart in HSV color space

Zhang Suen (ZS) Algorithm

ZS is an algorithm that can be applied to do a thinning pattern of the digital image. The objective of thinning is to obtain a skeleton pattern of a binary image. To preserve the pixel connectivity, ZS does two sub-iterate processes access (Zhang & Suen, 1984). ZS operates on a black pixel $P1$ having 8 connected neighbors. The 8 neighboring pixels structure can be illustrated in Figure 3.

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figure 3. The 8 neighboring pixels structure of the ZS algorithm

$A(P1)$ is the transition number of a white to a black (0 to 1) pixel in a sequential order $P2$, $P3$, $P4$, $P5$, $P6$, $P7$, $P8$, $P9$, and then back to $P2$. The transition number from the example in Figure 5 is 3, so $A(P1) = 3$.

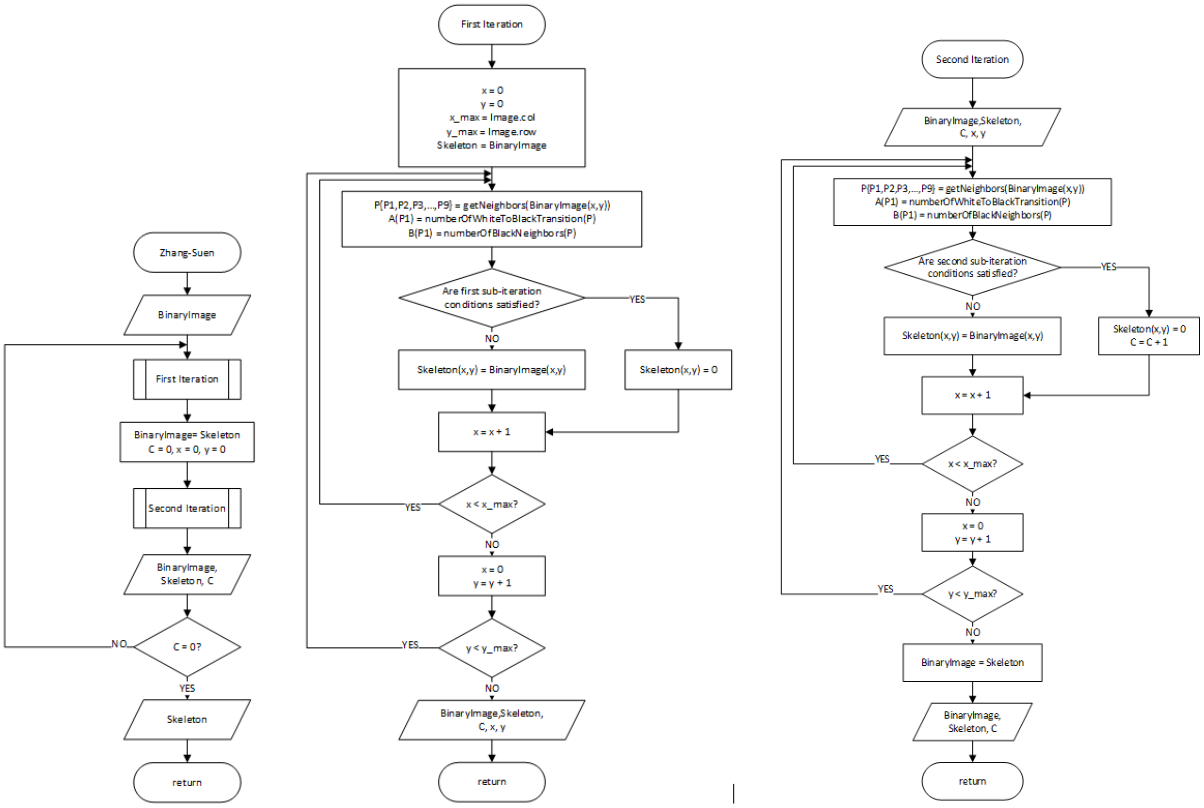


Figure 4. ZS thinning algorithm flowchart

$B(P1)$ denotes the number of black (openseighbors of $P1$. For the example, $B(P1) thea$ of given example in Figure 5 is 4 because $P1$ has 4 black neighbors.

1	1	0
0	1	0
1	0	1

Figure 5. The example of 8 neighboring pixel sequence

The overall process of the ZS algorithm can be illustrated by the flowchart in Figure 4. To process a binary image, the ZS algorithm divides the operation into two sub-iterations. Each of the sub-iterations is applied for every pixel in the mentioned binary image. Those sub-iterations can be done by processing all of each pixel in a sequential or parallel way. But, the result of that operation should be stored in a different image, so the value of the processed image will not be changed during the iteration. In the first sub-iteration, five conditions must be considered.

1. $P1 = 1$
2. $2 \leq B(P1) \leq 6$
3. $A(P1) = 1$
4. $P2 * P4 * P6 = 0$
5. $P4 * P6 * P8 = 0$

If a pixel met all those conditions, then change it into white (0). Vice versa, its value will not be changed. In the second sub-iteration, the following conditions must be considered.

1. $P1 = 1$

2. $2 \leq B(P1) \leq 6$
3. $A(P1) = 1$
4. $P2 * P4 * P8 = 0$
5. $P2 * P6 * P8 = 0$

If a pixel met all of those conditions, then change it into white (0). Otherwise, its value will not be changed. Both of the sub-iterations must be done until no pixel can be changed anymore.

Morphological Operations

The morphological operation has two kinds of basic operation. Those two operations are dilation and erosion (Young et al., 1998). Dilation can be denoted by the following equation:

$$D(A,B) = A \oplus B = \bigcup_{\beta \epsilon B} (A + \beta)$$

end erosion can be denoted by the following equation:

$$E(A,B) = A \ominus \tilde{B} = \bigcap_{\beta \epsilon B} (A - \beta)$$

where A denotes a binary image and B denotes the structuring element. The principle of dilation is expanding the area of ROI using a structuring element. Otherwise, the principle of erosion is to narrow the area of ROI using a structuring element. Let A an example of binary image shown in Figure 6 be (a). Then, let B be an example element structuring element shown in Figure 6 (b). The result from dilation of B on A can be shown in Figure 6 (c), otherwise the result from erosion of B on A can be shown in Figure 6 (d). Those two mentioned operations can be combined to form another morphological operation. The open morphological operation can be done by doing erosion first, then dilation later. Vice versa, the close morphological operation can be done by doing dilation first, then erosion later. Based on the literature by Jamil et al. (2008), the morphological operation can be used to do noise removal on a 2D binary image.

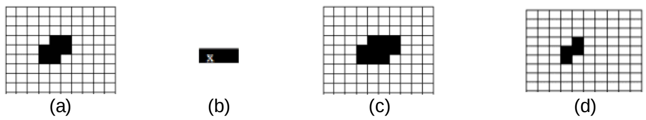


Figure 6. The illustration of morphological operations

Gaussian Blur

Trabelsi & Savaria (2013) and Ritter & Wilson (2000) describe that a Gaussian filter is one of several noise removal or smoothing techniques. In 2D image processing, a Gaussian filter uses a convolution technique to process 2D images (Trabelsi & Savaria, 2013).

Originality

In this research, HSV color space segmentation and ZS Algorithm are used to create a syntactic recognition model. This research includes detailed information about the way to implement the logic from scratch and includes possible fault behavior during the detection and recognition process. HSV color space segmentation is used to separate the ROI and the background. In this case, the ROI is a human hand bare skin. ZS algorithm is used to get the skeleton model of the ROI. Using the aforementioned model, the number of fingertip feature points existing in the ROI can be extracted.

Research Method

This research was conducted in several phases. They are literature review, design, development, and observation on the result in sequence. The literature review phase is focused on finding possible algorithms, system design possibilities, and the theories related to the research. The design phase is focused on designing the sequence of the system's input, process, and output, constructing the several algorithms into sequences, and documenting the sequences into some diagrams based on the theoretical information conducted during the literature review phase. The development phase focused on implementing the design phase through coding activities. The observation phase aims to find the fault behavior possibilities in the implementation of the system.

Proposed Design

To do detection and recognition, several parts should be done. Sequentially, several parts of the process can be shown in a block diagram shown in Figure 7. Every part of them will be elaborated by a flowchart in Figure 8.

Figure 7. Block diagram of the proposed design

In the data acquisition block, the sample of data will be captured from a video or camera with a fixed resolution. The captured image sample usually uses an RGB color space as the default format.

In preprocessing block, the RGB data sample will be converted into HSV color space. The Gaussian filter will be used to do a smoothing task and noise removal of those HSV images, so the application produces an HSV image with a more clear color region. By those clear differences in the color region, the segmentation (hand detection) process may produce a better quality of the binary image. The hand detection steps have been explained already in Figure 2. The result of this hand detection process is a binary image consisting of black and white pixels only. Those binary images should be refined more through morphological operation, so the application can produce noise-free binary images. After that ZS algorithm will be used to do a thinning to the existing binary image. The thinning process can be seen through a flowchart in Figure 4. A skeleton will be produced from the previous process. So, a skeleton will be the final result of this preprocessing block.

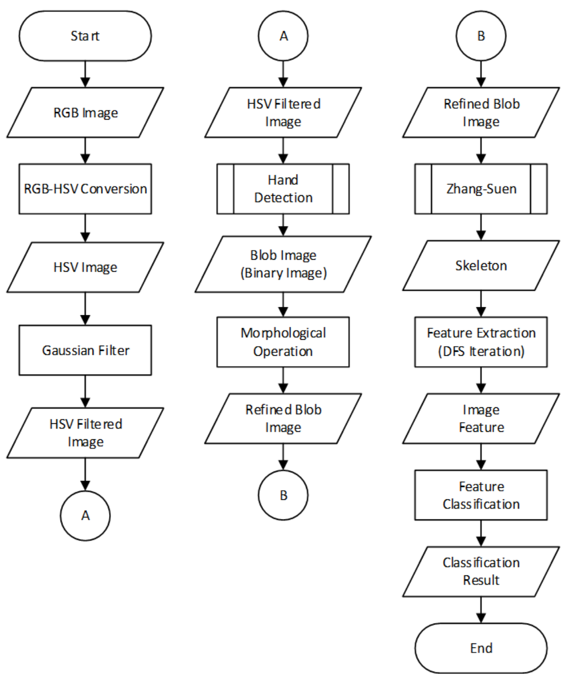


Figure 8. Flowchart design of hand detection and gesture recognition

In the feature extraction block, a specific skeleton will be selected. The skeleton will be iterated using a DFS algorithm. In every iteration, the algorithm will verify whether the inspected pixel is an endpoint or not. If an endpoint detected, then the pixel coordinate will be collected in a fingertip coordinate list. Otherwise, it will be ignored. So, a list of endpoint coordinate is the final result of this feature extraction block. This list represents the coordinate location of every fingertip detected by the application in the input image.

Syntactic classification block will classify the detected gesture based on the number of endpoints detected in a skeleton. The number of a detected endpoint can be counted by calculating the size of mentioned endpoint list earlier.

In the result block, the classification result will be shown by the application. The endpoint location will be printed on the default input image. Those printed marks are a sign of detected fingertip.

All those mentioned blocks will be executed sequentially. In each frame change, those blocks of operation will be executed again until the frame is stopped.

Result and Analysis

An experiment is conducted to validate the detection and recognition result process. The experiment also intended to find several kinds of faults usually happened when the ZS thinning method is used to gain the detection and recognition result.

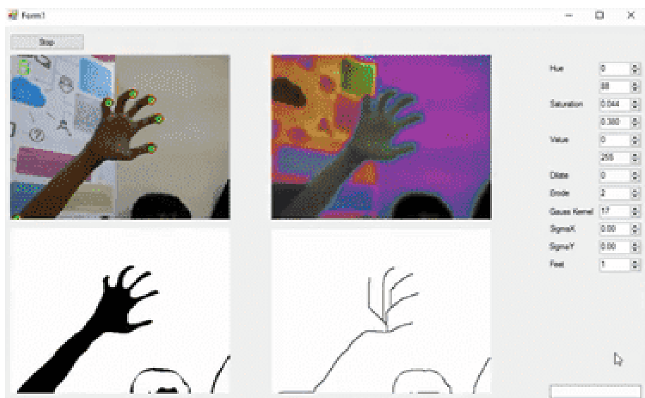


Figure 9. Application UI Result

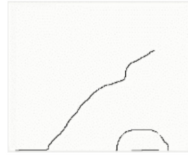
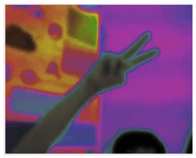
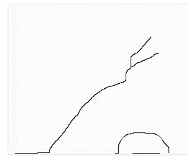
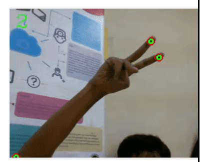
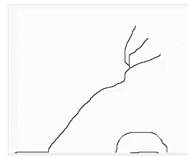
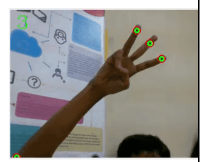
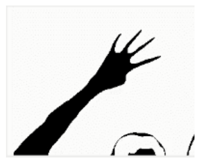
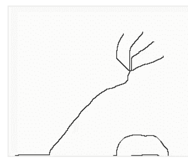
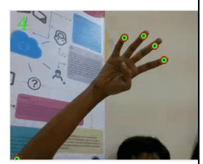
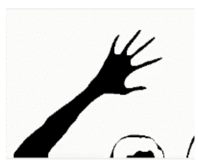
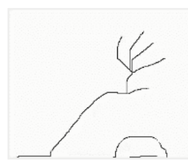
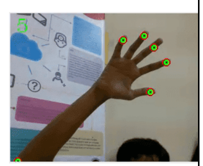
Figure 9 describes the application UI result in general. The experiment is conducted by verifying some gestures described in Figure 10. The tested data is an image with a size of 320×240 pixels which is captured frame by frame from a webcam in real-time. This experiment is conducted to know the capability of the proposed design in detection and recognition cases.

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

Figure 10. Gesture Classification (a) one, (b) two, (c) tree, (d) four, (e) five

Results of the first conducted experiment can be seen in table 1 and table 2. By ZS thinning, each gesture can be detected and recognized in an ideal condition. The ideal condition here means that the segmentation algorithm on HSV color space can produce a representative binary image of the detected hand. Hand segmentation on the HSV color space model can work if the background image color is simple and contrasts with the ROI color. Imperfect detection may result in a skeleton that is different from the actual expectations. A bad skeleton will lead to several fault results. From the observation, two kinds of fault can happen during the detection and recognition process. This fault is caused by unstable hand movement.

Table 1. Result of the experiment

Gesture	HSV	Binary	Skeleton	Result
One				
Two				
Tree				
Four				
Five				

The first type of fault is caused by an imperfect detection of hand. A hole with an irregular shape and angle inside the detected region is the result of this imperfect detection. This irregular shape of the hole may result in some unexpecendpointsints. An unexpected endpoints called by ethe Xtra endpoint, as illustrated in Figure 11. In this case, those extra endpoints are the result of some sharp angle inside the detected region preserved by the thinning algorithm.

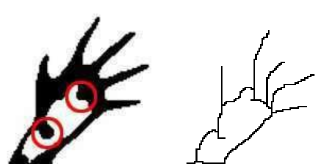
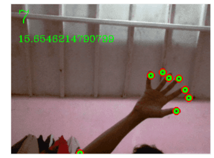


Figure 11. The cause of extra endpoints

The second type of fault is caused by a combination of unexpected background noise with the binary image of the detected hand. This combination may result in a bad shape binary image. Some regions of a binary image with a sharp angle resulting from background noise will be preserved by the thinning algorithm. This condition led to the formation of an extra endpoint too.

Table 2. Result of the experiment

Fault Type	**HSV**	**Binary**	**Skeleton**	**Result**
1st fault				
2nd fault				

Both the first and second faults produce some extra endpoints. In the case of fingertip counting, these extra endpoints lead to the wrong result, because these extra endpoints will be added to the list of the endpoint. In the end, these extra endpoints will be counted as fingertips too. To prevent this condition, the formation of extra endpoints should be anticipated in the first experiment.

Conclusion

This research proposes a design of syntactic method on hand detection and gesture recognition using a sequence of some algorithm. HSV color space segmentation is implemented as a hand detection method based on skin color in the HSV color space model. ZS algorithm is implemented as a thinning method to get a skeleton from an object. Unstable hand movement may lead to a fault called by the extra endpoint. Extra endpoints are considered a challenge that must be anticipated when using the thinning method to perform syntactic recognition.

References

Abu-Ain, W., Abdullah, S. N. H. S., Bataineh, B., Abu-Ain, T., & Omar, K. (2013). Skeletonization Algorithm for Binary Images. *Procedia Technology*, *11*(Iceei), 704–709. https://doi.org/10.1016/j.protcy.2013.12.248

Eshitha, K. V., & Jose, S. (2018). Hand Gesture Recognition Using Artificial Neural Network. *2018 International Conference on Circuits and Systems in Digital Enterprise Technology, ICCSDET 2018*. https://doi.org/10.1109/ICCSDET.2018.8821076

Huu, P. N., & Phung Ngoc, T. (2021). Hand Gesture Recognition Algorithm Using SVM and HOG Model for Control of Robotic System. *Journal of Robotics*, *2021*, 1–13. https://doi.org/10.1155/2021/3986497

Jamil, N., Sembok, T. M. T., & Bakar, Z. A. (2008). Noise removal and enhancement of binary images using morphological operations. *Proceedings - International Symposium on Information Technology 2008, ITSim*, *3*. https://doi.org/10.1109/ITSIM.2008.4631954

Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, *11*(3), 172–185. https://doi.org/10.1016/j.rti.2004.12.004

L. Lam, S. W. L., & Suen, C. Y. (1992). Thinning Methodologies -A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*, 869–885.

Lam, L., & Suen, C. Y. (1995). An Evaluation of Parallel Thinning Algorithms for Character Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 914–919. https://doi.org/10.1109/34.406659

Makahaube, S. S., Sambul, A. M., & Sompie, S. R. U. (2021). *Implementation of Gesture Recognition*

Technology for Automated Education Service Kiosk. *16*(4), 465–472.

Qin, X. Z., Lei, L., Yuan, Q. D., Qi, W., & Ying, M. (2013). An algorithm of thinning on character skeleton. *Proceedings - 2013 International Conference on Computational and Information Sciences, ICCIS 2013*, 730–733. https://doi.org/10.1109/ICCIS.2013.197

Ritter, G. X., & Wilson, J. N. (2000). Handbook of computer vision algorithms in image algebra: Second edition. In *Handbook of Computer Vision Algorithms in Image Algebra: Second Edition*.

Santosh, D., Venkatesh, P., & Poornesh, P. (2013). Tracking Multiple Moving Objects Using Gaussian Mixture Model. *Ijsce.Org*, (2), 114–119. Retrieved from http://www.ijsce.org/attachments/File/v3i2/B1453053213.pdf

Shimizu, M., Fukuda, H., & Nakamura, G. (2000). A thinning algorithm for digital figures of characters. *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, *2000-Janua*, 83–87. https://doi.org/10.1109/IAI.2000.839576

Silvia, A., & Husni, N. L. (2014). *Hand Contour Recognition In Language Signs Codes Using Shape Based Hand Gestures Methods*. 65–68.

Sobottka, K., & Pitas, I. (1998). A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing: Image Communication*, *12*(3), 263–281. https://doi.org/10.1016/S0923-5965(97)00042-8

Sudarma, M., & Putu Sutramiani, N. (2014). The Thinning Zhang-Suen Application Method in the Image of Balinese Scripts on the Papyrus. *International Journal of Computer Applications*, *91*(1), 9–13. https://doi.org/10.5120/15844-4726

Sutoyo, R., Prayoga, B., Fifilia, Suryani, D., & Shodiq, M. (2015). The Implementation of Hand Detection and Recognition to Help Presentation Processes. *Procedia Computer Science*, *59*(Iccsci), 550–558. https://doi.org/10.1016/j.procs.2015.07.539

Trabelsi, A., & Savaria, Y. (2013). A 2D Gaussian smoothing kernel mapped to heterogeneous platforms. *2013 IEEE 11th International New Circuits and Systems Conference, NEWCAS 2013*. https://doi.org/10.1109/NEWCAS.2013.6573641

Young, I. T., Gerbrands, J. J., Vliet, L. J. Van, Theodore, I., Jacob, J., Vliet, V., & Jozef, L. (1998). *Fundamentals of image-processing*. Delft University of Technology.

Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, *27*(3), 236–239. https://doi.org/10.1145/357994.358023