WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Hand Gesture Recognition- A Parallel Implementation

Hai T. Nguyen[1], Binh A. Nguyen[2], Ngoc T. Le[2], Hanh T. Pham[2], and Giao N. Pham[3*]

[1]HUTECH Institute of Engineering, HUTECH University, Ho Chi Minh City, Vietnam
[2]ICT Department, FPT University, Hanoi, Vietnam
[3]Dept. of Computing Fundamentals, FPT University, Hanoi, Vietnam
E-mail:nt.hai75@hutech.edu.vn,binhnase04865@fpt.edu.vn, ngoclthe131028@fpt.edu.vn,hanhpthe130014@fpt.edu.vn,giaopn@fe.edu.vn
(*Corresponding author)

ABSTRACT

Hand gesture recognition is one of the very active research areas in the Computer Vision field. It provides the easiness to interact with machines without using any extra device and if the users don't have much technical knowledge about the system, they still will be able to use the system with their normal hands. Gestures communicate the meaning of statement said by the human being. They come naturally with the words to help the receiver to understand the communication. It allows individuals to communicate feelings and thoughts with different emotions with words or without words. This paper presents a parallel implementation of hand gesture recognition. Gesture made by human being can be any but few have a special meaning. Human hand can have movement in any direction and can bend to any angle in all available coordinates

Keywords: Hand Gesture Recognition; Open-CV; Feature Extraction; Contour Extraction; Aforge.net;

1. Introduction

This hand gesture recognition algorithm is based on a chapter from the book by K. Kraiss[1] and uses the AForge.Net[2] framework for capturing frames from a webcam in real time.Using a webcam frames are captured, processed for detection of the hand, features extracted and using the features the gestures are identified. There are 3 types of gestures that can be recognized: stop, right and left as seen in figure 1. The program was designed in C# with .Net Framework 4 and uses it's threading support[3] to implement a parallel version of the algorithm.

Fig.1: The three gestures: Stop, Right, and Left.

2. The Proposed Method

The algorithm steps are presented below and in Fig. 2: Frame capture using A Forge .Net Framework; Binarization of the image using a threshold method; Threshold filtering of the binary image using one upper and lower value; Contour extraction; Feature extraction; and Gesture recognition [4, 5].The two features that are used for identifying the three gestures are compactness and protrusion ration. Based on the values of these two parameters the hand gesture can be identified and the result shown to the user.

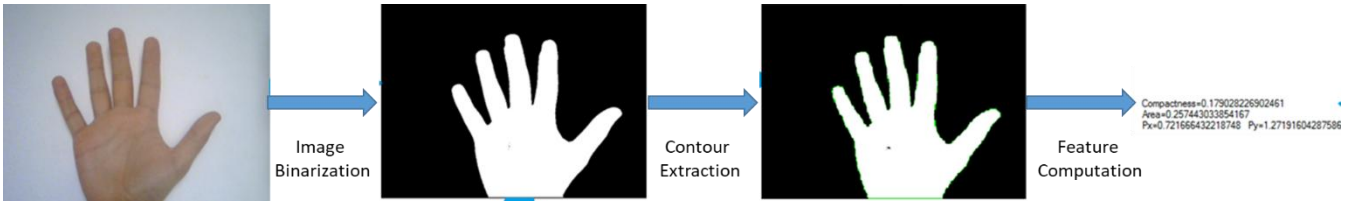

Fig.2. Gesture algorithm steps

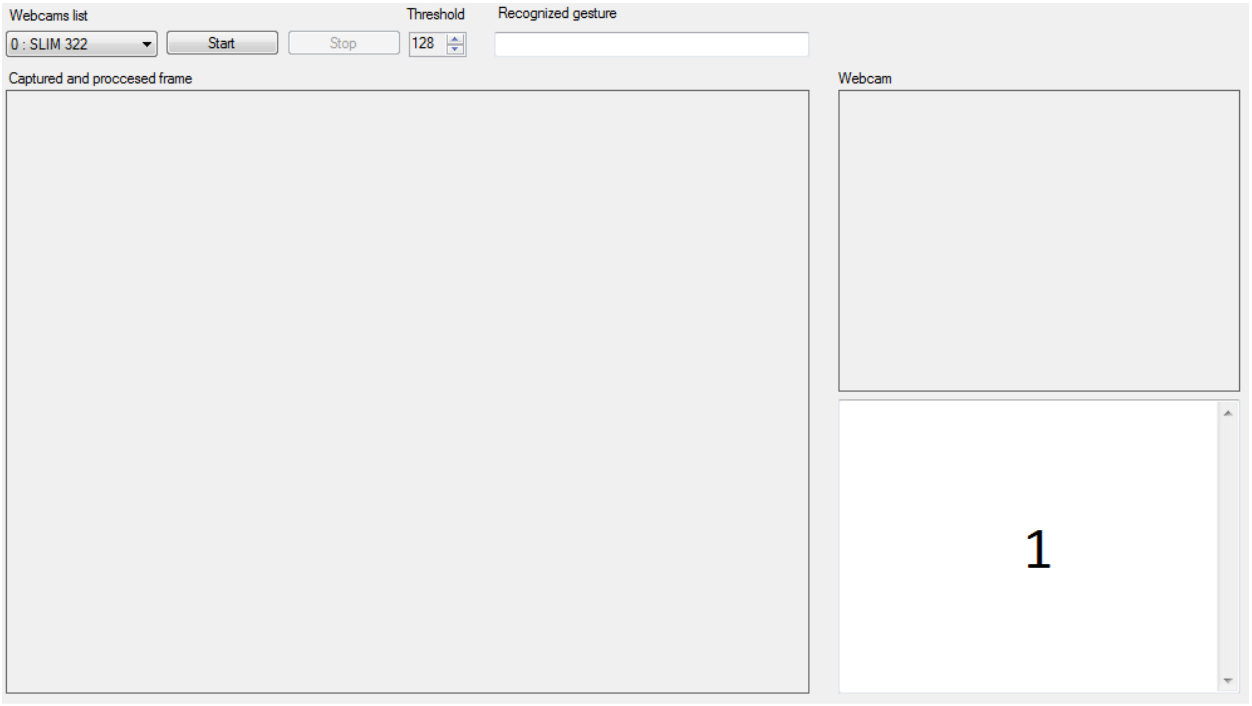

Fig.3. Graphical interface of the application

An overview of the graphical interface can be seen in figure 3 and contains the following sections [6-8]:

- ✓ **Webcam list:** Displays the video capture devices detected on the system by AForge .Net.
- ✓ **Start and stop buttons:** Are used for starting and stopping the acquisition process.
- ✓ **Threshold:** The control is used for setting the threshold for the binarization process permitting the user to accommodate for different conditions.
- ✓ **Webcam:** The frames captured from the webcam, binarized and filtered are shown in that area
- ✓ **Captured and processed frame:** The area is used for displaying the current captured frame and the contour is highlighted. The frames are captured at a rate of one frame per second.
- ✓ **Recognized gesture:** The text box is used for displaying the current gesture that has been recognized. Values are: Stop, Right, Left, not recognized.
- ✓ **Area 1:** The text box is used to display information about the current process and the features extracted.

3. Experimental Results

Four steps of the algorithm have been chosen to be implemented in a parallel form to meet the systems real time constraint. Three of the steps are configured to detect the number of logical processors on the system and to start threads according to that number [.To implement a parallel version of the binarization process, which is an operation carried out on single pixels, the matrix containing the image was divided into lines and each thread processes one line as shown in Fig. 4. When a thread completes its task another is started and the process continues until all of the image has been converted. The same technique is used to copy the resulting image back to the original bitmap and displayed on screen.

For the task of filtering some of the noise present in the binary image a two threshold filter was used. A similar process to the one above was used to implement this step the only difference was that a window for the filter is used. For the result of a thread to not be influenced by another thread, two

images were used, one for temporary storage of the result.

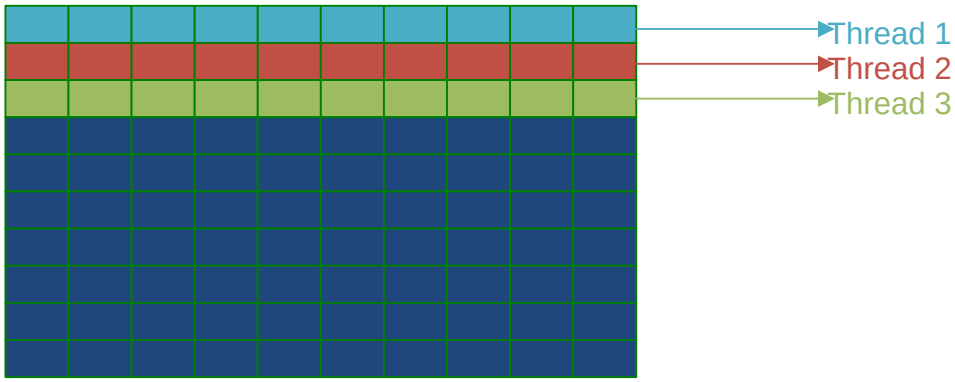

Fig. 4. Thread allocation for image binarization and back copy.

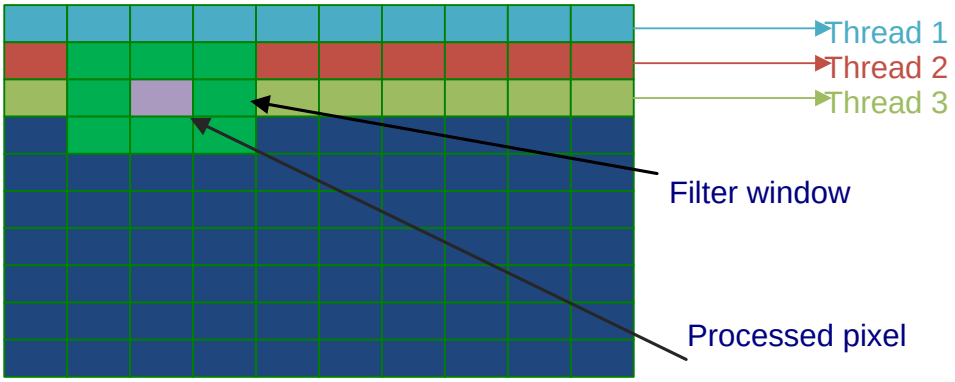

Fig. 5. Thread allocation for binary image filtering.

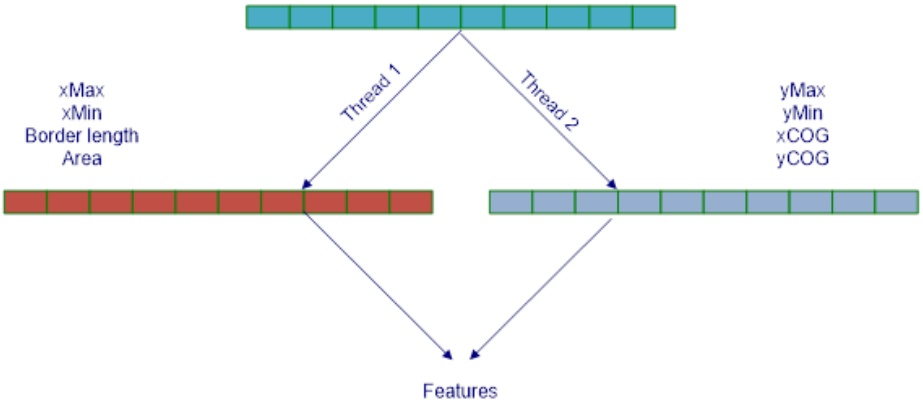

Fig. 6. Thread allocation for feature computation.

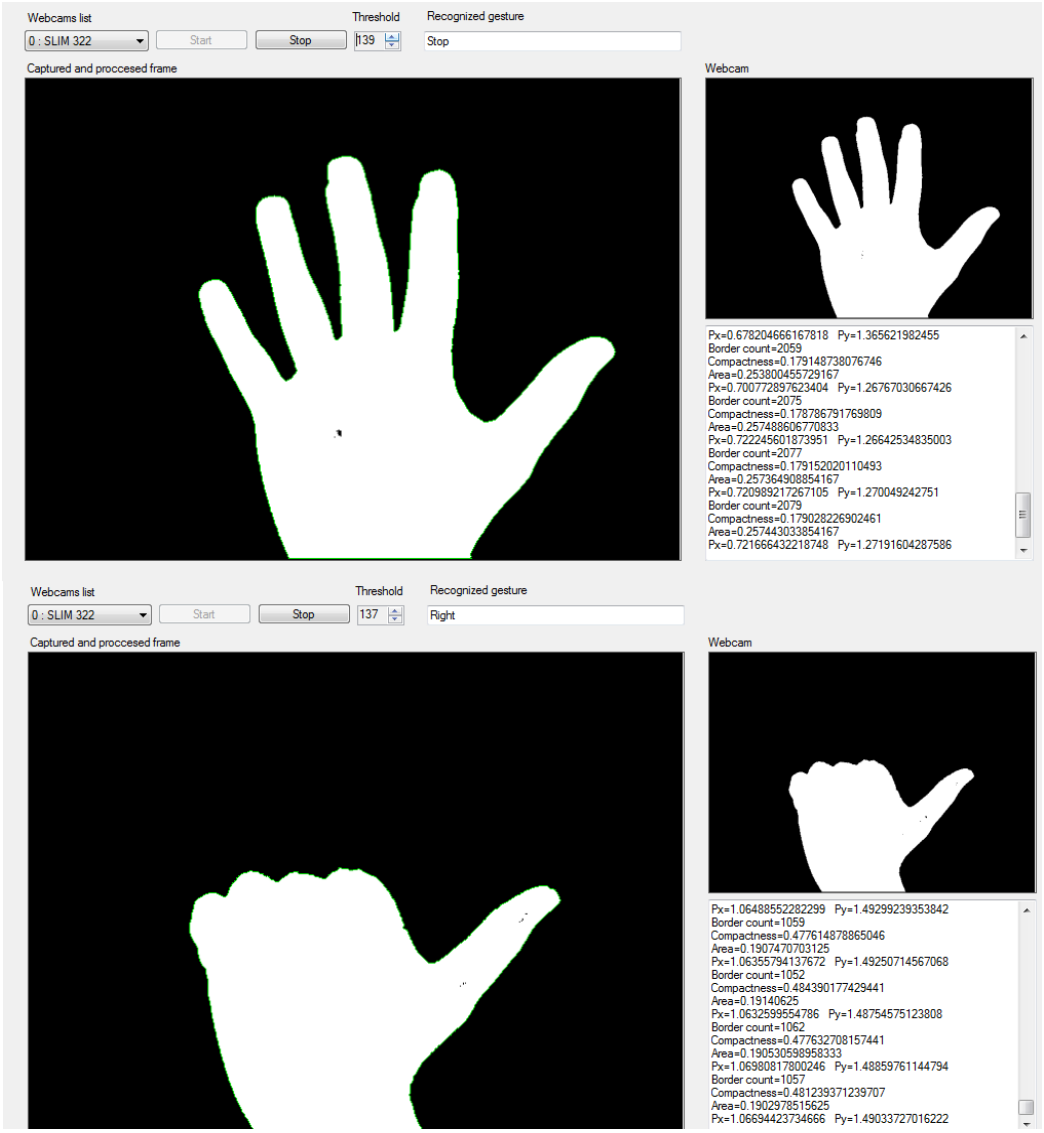

Fig. 7. Gestures recognized by the software.

The step to use parallel implementation is the feature computation step. The features that are needed for the gesture recognition process: compactness and protrusion ratio. To compute these features another set of features must be calculated first: xMax (maximum coordinate on the X axis), xMin, yMax, yMin, area, length of the border, center of gravity position on the X axis and center of gravity position on the Y axis. To accelerate this task, the set was split into groups of equivalent complexity and a thread was allocated for each one as described in Fig. 6.

The application was written in C# using .Net Framework 4 and utilizes AForge .Net Framework for interfacing with the webcam. To run the algorithm, select a webcam from the list. If no webcam is present try another webcam, reinstalling webcam drivers or search the AForge website for possible solutions. The start button starts the acquisition process, the image binarized, filtered and displayed in the webcam area. Once per second a frame is captured, the region is identified, the features are computed and a hand gesture is identified, if no gesture can be found a message will be displayed. Stopping the acquisition process is done with the stop button as shown in Fig. 7

4. Conclusion

Different applications of hand gesture recognition have been implemented in different domainsfrom simply game inputs to critical applications. Hand gesture recognitions is the natural tointeract with vision enabled computers and other machines. This paper primarily focused on thestudy of work done in the area of natural hand gesture recognition using Computer VisionTechniques. In the future we will work in the area of individual finger position bendingdetection and movements, as work done in this area are very few. Mostly researchers workedwith full hand position detection or the fingertip position to write virtual words.

Acknowledgement

This research is supported by FPT University, Hanoi, Vietnam; and HUTECH Institute of Engineering, HUTECH University, Ho Chi Minh City, Vietnam.

REFERENCES

[1] K. Kraiss, Advanced Man-Machine Interaction Fundamentals and Implementation, Srpinger, Berlin, 2006 available online at http://www.springerlink.com/content/978-3-540-30618-4#section=458355&page=1&locus=0

[2] http://www.aforgenet.com

[3] Parallel Programming with Microsoft .NEThttp://msdn.microsoft.com/en-us/library/ff963553

[4] Kendon A. Gesture: Visible Action as Utterance, Cambridge University Press. UK, 2004.

[5] Kroeker K.L., Alternate interface technologies emerge, Communications of the ACM, Vol. 53, No. 2, Feb 2010, pp. 13-15.

[6] Nolker C., Ritter H., Visual Recognition of Continuous Hand Postures, IEEE Transactions on neuralnetworks, Vol 13, No. 4, July 2002, pp. 983-994.

[7] Sturman D., Zeltzer D., A survey of glove-based input, IEEE Transactions on Computer Graphicsand Applications, Vol. 14, No. 1, Jan. 1994, pp. 30-39.

[8] Stefan A., Athitsos V., Alon J., Sclaroff S., Translation and scale invariant gesture recognition incomplex scenes, Proceedings of 1st international conference on pervasive technologies related toassistive environments, Greece, July 2008.