GESTURE RECOGNITION USING HISTOGRAM AND STATISTICS

Akila Victor[1], Niraj Kumar Jain[2] and Shresth Sethi[3]

1. Assistant Professor, Vellore Institute of Technology, Vellore, Tamil Nadu, India, Phone No.09994081412
2. Student, Vellore Institute of Technology, Vellore, Tamil Nadu, India, Phone No.08870683101
3. Student, Vellore Institute of Technology, Vellore, Tamil Nadu, India, Phone No.08870521882

ABSTRACT

This paper is about gesture recognition technique using video data as input. An unknown movement of the body in front of the camera should be detected by the application and then the type of gesture is to be recognized. Our algorithm has to be adaptive so in this approach we use different types of filtering algorithms to abstract the body image, several filtering techniques like difference filter, threshold filter, opening filter etc are used in order to improve the image complexity and noise reduction. The idea of our gesture recognition algorithm is quite simple, which is based on histograms and statistics, thus makes this algorithm quite easy, in terms of implementation and understanding, it is quite efficient in performance, and does not require a lot of computational resources, which is quite important in cases where we need to process a lot of frames per second.

Keyword: Gesture Recognition, histograms, statistics, filters, image complexity, noise reduction.

Corresponding Author: Akila Victor

INTRODUCTION

Image processing not only is a very vast field but is a field that is expanding at a very fast pace. Everyday a new application for image processing crops up. It can be applied to a spectrum of applications ranging from surveillance to image enhancement. There is virtually no measure of how many computer applications would be crippled without image processing. Now images when represented in digital format can be simply understood as a 2-D array of pixels, with associated values of intensity. For all practical purposes it is this array of pixels that is required to deal with the image, and most of the image processing applications can be easily accomplished by modifying and transforming the intensity values. Many applications require that the intensities be treated as statistics. This statistics can essentially be treated as a histogram. The application of histogram technique thus comes in here. Histogram is a very innovative science itself. It has widespread application in the field of computer science, and one the applications in the field of image processing is discussed in this paper.

HISTOGRAM

Histogram's in image processing is the graphical representation of the tonic distribution in the digital imaging. It represents the number of pixels for every tonic value. By viewing the histogram of a specific image a viewer can be able to determine the entire tonic distribution. The histogram represents the count of pixels of an image (vertical axis) with a particular sharpening value (horizontal axis). Methods used in the digital editor permit the viewer to opticaly balance the sharpness value of each pixel and to vigorously display the output as the balance are made. Improvements in picture sharpness and difference can thus be obtained. In the branch of computer science, image histograms can be useful aid for thresholding. Because the data presented in the graph is a description of pixel arrangement as a function of tonic difference, image histograms can be studied for peaks and valleys which is used to allot a threshold value. This threshold value is used for edge detection, image segmentation.

METHODOLOGY

As acknowledged earlier, any image can be described as an array of pixels. This array of pixels that is required to deal with the image, and most of the image processing applications can be well accomplished by modifying and transforming the intensity values. Many applications require that the intensities be treated as statistics. This statistics can essentially be treated as a histogram. The step by step procedure to do this would involve a number of processes.

3.1 MOTION DETECTION AND OBJECT EXTRACTION

We have used difference filter to extract the object image from the background image. On the difference image, it is possible to see the actual difference between two images. Now we are using threshold filter to the difference image so that each pixel may be classified as a significant change i.e, caused by moving object or as a non-significant change. After getting the threshold difference image we need to reduce noise so we are using opening filter, by this filter the left alone pixels, which is due to deafening camera and other circumstances, will be eliminated, so we'll have an image which describes only the more or less relevant areas of changes. We need to shift our background image in the areas of our switching by applying the MoveTowards filter, As we have an object image which carry the background plus the target on it, then by applying the MoveTowards filter successively on the background image that will make it alike the target image. The more we use the MoveTowards filter to the background image, the more visible is the presence of the object. So, we account the size of the largest object on the present frame and, if it is not that large, we ponder the object as trivial, and we just renew our background frame to adjust to the changes. Now the conflict between frames is obtained by examining the changed pixel variable, which is used to compute the aggregated changes in percent, and then the value is matched with the actual motion limit to check if we have a motion or not. Now we keep a counter which evaluate the amount of sequential frames without motion. And, unless the magnitude of sequential frames with no motion attains some specific value, we now move the object to the hand gesture recognition module.

3.2 HAND GESTURE RECOGNITION

When we have discovered an object to process, we can examine it, trying to acknowledge a hands gesture. The hands gesture recognition algorithm infer that target object captures the entire image, but not part of it. The aim of our algorithm is simple, and completely based on statistics and histograms which makes this algorithm easy, in terms of execution and understanding. To begin with the process we are going to find areas of the image which are captured by hands, and the area, which is captured by the torso. This can be obtained by using two type of histogram vertical and horizontal histogram, which is determined by using the Horizontal_Intensity_Statistics and Vertical_Intensity_Statistics classes. Now we can infer from the histogram that the torso area is discribed by a peak of high values but, the hands areas have comparatively low values on the histogram. As we know the dimensions of the human body, we may say that the hand thickness can never go beyond 30 percent of the body height. By implementing a threshold on the horizontal histograms, we can classify torso and hand area. From the horizontal threshold histogram, we can obtain the body torso thickness and hands length, the length of the left hand is the thickness of the free area from the left, the right hands length is the thickness of the free area upon the histogram from the right, and the torso's thickness is the thickness of the area between the free areas. As we have obtained the hand length and the torso thickness, we can judge if the hand is raised or not raised. For both the hands, the algorithm identifies weather the hand is raised, not raised, straight, raised diagonally up, or diagonally down. Now to obtain the accurate position of the hand we apply Vertical_Intensity_Statistics class, but in some cases we may get deafening histograms so we have to follow two small steps on vertical histogram.

3.2.1 First of all, we are removing the low values from the histogram, which are lower than 10% of the histogram’s maximum value.

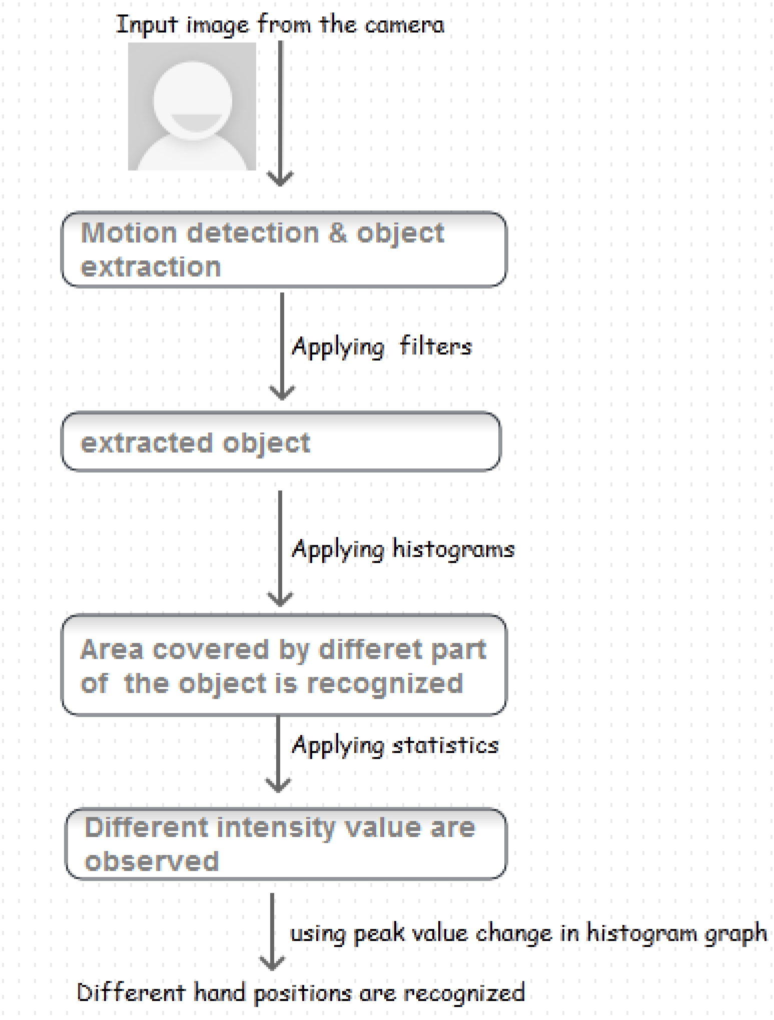
3.2.2 Another type of issue is a "twin" hand, which is actually a shadow. This could be solved by moving along the histogram and eliminating all peaks which are not the highest peak.

If the hand is raised straight then the hands length is much more than its thickness and, its histogram must have a high value, but thin peak, so we apply this property to classify weather the hand is raised straight or not. Now we are going to judge if the hand is diagonally raised down or diagonally up. The peak value of diagonally down hand is switched to the center of the histogram and the peak value of the diagonally up hand is switched to the beginning of the histogram, so we can determine the position of hand. Thus our algorithm provides four different positions of both the hands. Finally the algorithm will give the following results:

- left hand not_raised; right hand not_raised.
- left hand not_raised; right hand raised_diagonally_down.
- left hand not_raised; right hand raised_straight.
- left hand not_raised; right hand raised_diagonally_up.

FLOW CHART

The image is taken from the video camera as the input for this process, now our task is to detect whether there is any moment or not to know if there is any object in our frame and then we need to extract the object image from the whole image, so we use different types of filtering algorithms to extract the object. As we have got the extracted image, we can apply histogram technique to recognize the area covered by different parts of the object. Then by applying vertical intensity statistics and horizontal intensity statistics we got to know about the different intensity values for the extracted image, finally by using the peak value changes in the histogram graph we can determine different hand positions.


Input image from the camera
Motion detection & object extraction
Applying filters
extracted object
Applying histograms
Area covered by differet part of the object is recognized
Applying statistics
Different intensity value are observed
using peak value change in histogram graph
Different hand positions are recognized

CONCLUSION AND FUTURE WORK

In this paper, we have the algorithms which, first allow us to extract moving objects from a video camera, and, second, to successfully identify hands gesture illustrated by the object. The algorithm which we used is very smooth and easy for the implementation process. Also, as it depends only on the output given by histograms, it is very efficient in performance, and does not demand a lot of computational resources, which is important in cases where we require to access allot of frames per second. In the future, we will try to implement the project for full body gesture like recognizing hand and leg gestures together.

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2,pp. 257–286, 1989.

[2] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe, "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems," Computer Music Journal,vol. 24, no. 1, pp. 57–69, 2000.

[3] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis, "A gesture-driven multimodal interactive dance system," in Proceedings of the IEEE International Conference on Multimedia and Expo, 2004, pp. 1579–1582.

[4] B. Peng, G. Qian, and S. Rajko, "View-invariant full-body gesture recognition from video," in Proceedings of the International Conference on Pattern Recognition, 2008, pp. 1–5.

[5] "View-invariant full-body gesture recognition via multilinear analysis of voxel data," in Proceedings of International Conference on Distributed Smart Cameras, 2009, pp. 1–8.

[6] B. Peng and G. Qian,"Online gesture spotting from visual huall data," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[7] J. Kjolberg,"Designing full body movement interaction using modern dance as a starting point," in Proceedings of the Conference on Designing Interactive Systems, 2004, pp. 353 – 356.