International Journal of Technology and Engineering System (IJTES)
Vol 8. No.1 – Jan-March 2016 Pp. 138-142

available at : www.ijcns.com
ISSN: 0976-1345

SWARM INTELLIGENCE FOR DETECTING INTERESTING EVENTS IN CROWDED ENVIRONMENTS

R.Sudhandira[1], Dr.Smitha Elsa Peter[2],
[#]PG Scholar &PRIST University, Assistant Professor&PRIST University
Associate.Professor &PRIST University-Tanjavur Campus Tamilnadu, India

***Abstract*— Crowd counting and density estimation is still one of the important tasks in video surveillance. Usually a regression based method is used to estimate the number of people from a sequence of images. This Project describes segmentation, detection, and counting of objects (people) as blobs in a real scenario. In this paper we investigate to estimate the count of people in a crowded scene. Blob detection is designed with binary level images. So that, scene image have to be thresholded to the certain level, its become binary image. Experiments on these databases show good performance of our method for people counting. This proposed system has been simulated using Matlab R2013a.**

Keywords— Blob analysis, Gaussian Mixture Model, MATLAB

I. INTRODUCTION

Real-time people flow estimation can be very useful information for several applications like security or people management such as pedestrian traffic management or tourists flow estimation. The use of video cameras to track and count people increase considerably in the past few years due to the advancement of image processing algorithms and computers' technology. Several attempts have been made to track people but all those different ways can be classify in three categories of different complexity:

1. Methods using region tracking features. To improve this methods some adding a classification scheme of pixels based on colour or textures.
2. Methods using 2D appearance of humans (using different models of humans)
3. Methods using multiple cameras to make a full 3D modelling.

The third category is more accurate than the two others because it rebuilds precisely the scene (so it deals in a better way the occlusion problems) but it is also the most difficult with complex algorithms. Some of the time, this system required a complex camera set-up (calibration) and cannot operate in real-time because the 3D models are too slow. This is why most of the systems used the two other categories.

The main goal of this thesis was to make different research about existing people counting system based on video camera and build a prototype of this using, in a first time, Matlab-Simulink programming tool. Then, link this system to the centralized database build by Yorrick. And finally discuss and compare it with other existing people counting system based on laser beam build by the two electronics students.

Automatic detection of blobs from image datasets is an important step in analysis of a large-scale of scientific data. These blobs may represent organization of nuclei in a cultured colony, homogeneous regions in geophysical data, tumor

locations in MRI or CT data, etc. This paper presents several approaches for blob detection and applications.

Before going into detail on blob detection, first some definitions of a blob are given. Lindeberg defines a blob as being a region associated with at least one local extremum, either a maximum or a minimum for resp. a bright or a dark blob. Regarding the image intensity function, the spatial extent of a blob is limited by a saddle point, a point where the intensity stops decreasing and starts increasing for bright blobs and vice versa for dark blobs. A blob is represented as a pair consisting of one saddle point and one extremum point.

Hinz just describes a blob as a rectangle with a homogeneous area, i.e. a constant contrast, which becomes a local extremum under sufficient amount of scaling. Rosenfeld et al. defines a blob as a crossing of lines perpendicular to edge tangent directions, surrounded by 6 or more directions, like in the following picture:

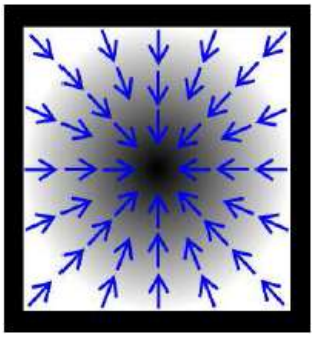


Fig: 1 A blob surrounded by 8 different directions

A third definition of a blob is given by Damerval [4], describing blobs as the largest modulus maxima of the continuous wavelet transform (CWT, see Appendix) along some maxima of interest. The CWT is able to construct a time-frequency representation, offering a good localization of frequencies and time (scale). The exact meaning of modulus maxima and maxima of interest is explained later in the section of the concerning method.

To this point only 2D blob definitions are mentioned. Other definition of a 3D blob as being elliptic features in scale-space portioned by a convex hull (boundary of the minimal convex set containing a set of voxels belonging to a blob).

II. PROPOSED SYSTEM

In this paper we have analysed the existing pre-processing filters and proposed a method that best ensembles to enhance the delectability of objects (people) in a real time scenario.

A. Pre-processing Filter

The Sobel operator does this in a rather clever way. An image gradient is a change in intensity (or colour) of an image (I'm over simplifying but bear with me). An edge in an image occurs when the gradient is greatest and the Sobel operator makes use of this fact to find the edges in an image. The Sobel operator calculates the approximate image gradient of each pixel by convolving the image with a pair of 3×3 filters. These filters estimate the gradients in the horizontal (x) and vertical (y) directions and the magnitude of the gradient is simply the sum of these 2 gradients.

B. Blob Detection

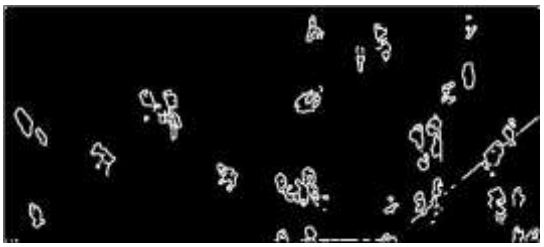
Blob detection is an algorithm used to determine if a group of connecting pixels are related to each other. This is useful for identifying separate objects in a scene, or counting the number of objects in a scene. Blob detection would be useful for counting people in an airport lobby, or fish passing by a camera. Blob area – Area of a particle expressed in real units (based on image spatial calibration). This value is equal to Number of pixels when the spatial calibration is such that 1 pixel represents 1 square unit. Middle mass would be useful for a baseball catching robot, or a line following robot.

To find a blob, threshold the image by a specific range as shown in fig. 4(b), the centroid has been detected after labeling each region in an image. That represents the middle mass, or the average location of all pixels of the selected color. Steps involved in Blob detection algorithm:

1. If the pixel is a blob colour, label it '1' otherwise label it 0
2. Go to the next pixel if it is also a blob colour and if it is adjacent to blob 1 label it '1' else label it '2' (or more)
3. repeat until all pixels are done

What the algorithm does is labels each blob by a number, counting up for every new blob it encounters. Then to find middle mass, also we can just find it for each individual blob. Blob detection is useful in finding blobs sizes, these are approximated values in pixels of an image. If we know difference between real size of starfish or shark, that will leads to predict original size of the objects.

The Blob Processing block helps to find the number of objects (starfish) visible in the image. So an additional MATLAB function block is added to count the total number of cars that are passing. The result is displayed using the inbuilt display block as shown in figure 5(c).



(a)

(b)

Fig. 2 (a) Edge extraction form gray scale using sobel edge detection. (b) Image filling by Matlab function to close object region

C. VHDL Design of Blob Detection Algorithm

The proposed blob recognition algorithm is aiming at processing a gray-scale image in VGA size, and can be divided into three major steps as below:

- Image scale down: the incoming video stream is in VGA (640x480) size. Inλ order to decrease the processing time and to reduce the resources for computation, this incoming VGA video image is downscaled by factors of four to QQVGA (160x120).
- Candidate location: given a downscaled QQVGA image in which the position of the target is unknown, this step is supposed to search for every possible candidate. Multiple candidates may be chosen according to a certain criterion, which exerts strong influence upon the performance of the whole system.
- Object identification: given each candidate, the related image data will be captured from the main memory to process and analyse. An identification algorithm inspired by the unique feature of the blob face is adopted to extract the explicit information of each face. The latter two steps can be divided into finer processes, and are illustrated in Figure 3.

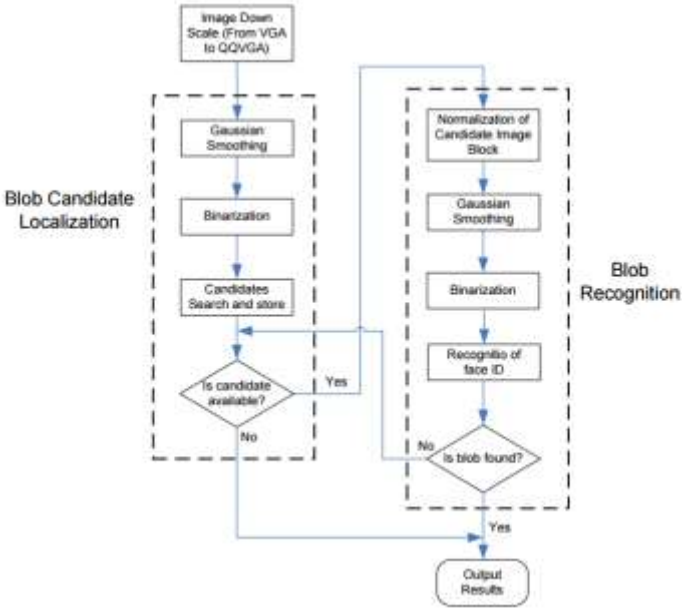


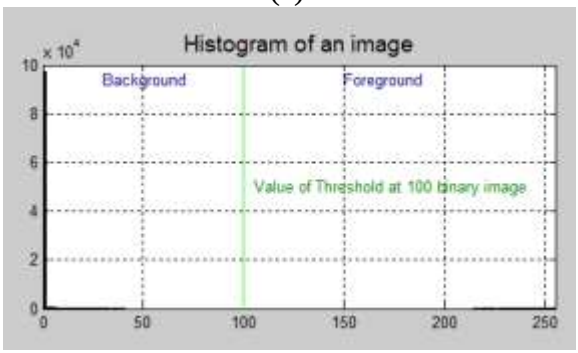
Figure 3 Flow Chart of Proposed Algorithm

III. RESULTS

Proposed work has useful in segmentation, detection, counting of people in crowded places. A typical blob analysis process scans through an entire image and detects all the particles, or blobs, in the image and builds a detailed report on each particle. This report usually contains approximately 50 pieces of information about the blob, including the blob's location in the image, size, and shape, orientation to other blobs, longest segment, and moment of inertia which is shown in fig. 3. Input image and thresholding values are shown in Fig. 5(a) and 5(b).

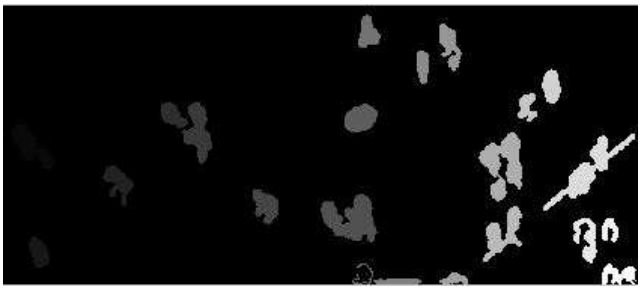
Fig: 4 shows number of objects found in a scene

(a)

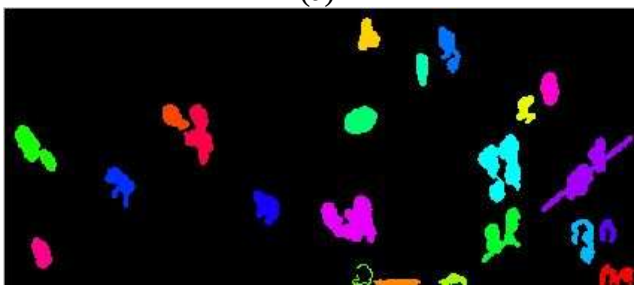


(b)

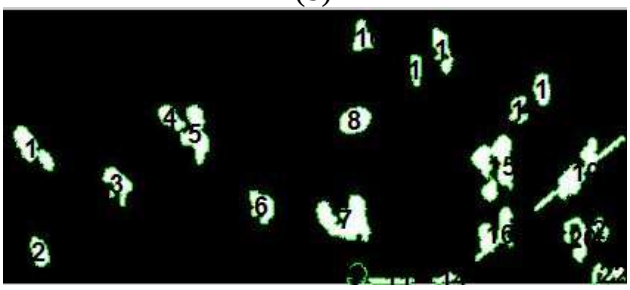
Fig. 5 (a) shows input rgb image (b) Shows threshold value used to know object region in an image



(a)



(b)



(c)

Fig. 6 (a) Converting binary to gray scale image after thresholding, (b) Applying pseudo colour for detected blobs, (c) showing numbers on each blobs

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	316	9,312	3%
Number of 4 input LUTs	450	9,312	4%
Number of occupied Slices	311	4,656	6%
Number of Slices containing only related logic	311	311	100%
Number of Slices containing unrelated logic	0	311	0%
Total Number of 4 input LUTs	516	9,312	5%
Number used as logic	450		
Number used as a route-thru	66		
Number of bonded IOBs	10	232	4%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	2.51		

Fig. 7 shows the area utilized by FPGA for people counting

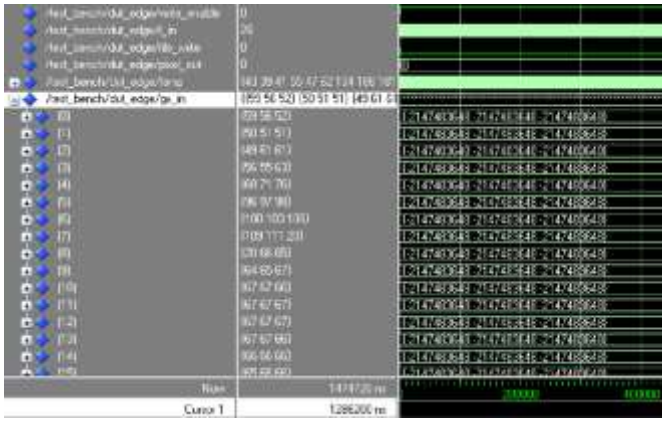


Fig. 8 Shows edge detection of input image

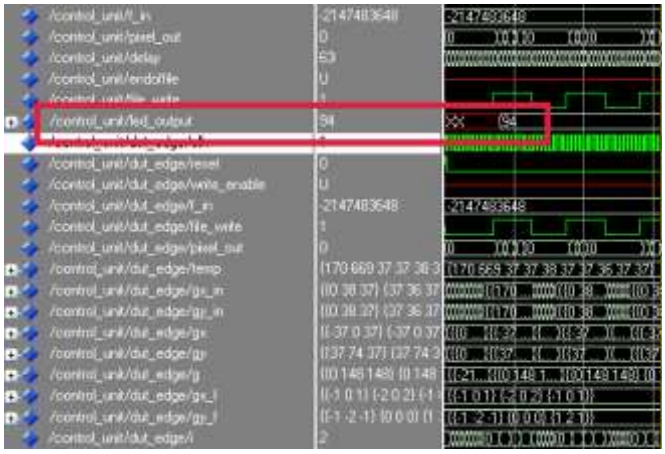


Fig. 9 Red boxed variable results number of objects detected in VHDL implementation

IV. FUTURE WORK

Performance improved with applying preprocessing filters before detecting blobs (people) on the image. It tends to detect people on clear consistency of the image taken from real life scenarios with binary format because blob detection is only applicable for binary images. If we want to count and track object passes through a frame or image only once in a time means, feature based on color segmentation could be added to address this issue. In future feature extraction based detection is helpful in differentiating people and other 'objects'.

V. CONCLUSIONS

This proposed work presents a system with optimized function for detection, segmentation, counting and of people and other objects like vehicle, animals, etc. Detection and counting is based on blob detection with combination of preprocessing filtering technique to exactly extract particular objects as being people or other small objects in the surface area. This system achieves optimized detection, segmentation and counting of people in social places which is very useful in creating a traffic model from this crowd people counting technique.

REFERENCES

1. X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007, pp. 1-8.
2. S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, pp. 1915-1926, 2012.
3. R. Clement, M. Dunbabin, and G. Wyeth, "Toward robust image detection of crown-of-thorns starfish for autonomous population monitoring," in Australasian Conference on Robotics and Automation 2005, 2005.
4. C.Stauffer, W.E.L. Grimson. "Adaptive Bacckground Mixture Models for Real-Time Tracking," in Proc. Computer Vision and Pattern Recognition Conf., vol. 2, Fort Collins, CO. USA, June 1999,pp.246-252.
5. Object Tracking: A Survey, Alper Yilmaz, Omar Javed, Mubarak Shah. A. D. Bue, D. Comaniciu, V. Ramesh, and C. Regazzoni. Smart cameras with real-time video object generation. In Proceedings of the IEEE International Conference on Image Processing, volume 3, pages 429–432, June 2002
6. Prakash Chockalingam. Non-rigid multi modal object tracking using Gaussian mixture model, The Graduate School of Clemson University, PhD thesis, 2009.
7. Ballard, D.H. and Brown, C.M. (1982) Computer Vision, Prentice Hall, New York.