Ateneo de Manila University, Manila, Philippines

Abstract: Microscopic pedestrian field requires large amount of trajectory data of individual pedestrian. Such data collection, if done manually, needs tremendous efforts and very time consuming. Though many literatures had asserted possible automation of such task using video camera, most of them had not been performed for a very crowded situation. Our research has shown a breakthrough to automatically track many pedestrians in crowded pedestrian experiments where occlusion is clearly a problem. The result of such tracking is a database of pedestrian trajectories over time and space.

Key words : Pedestrian tracking

## 1 INTRODUCTION

Measurement of individual pedestrian trajectories, which is part of microscopic pedestrian field of study, is useful in a variety of situations. In business environments, information about customer's movement derives better knowledge on shopping behavior and preferences. In crowded locations (such as sports venues, public transport station, religious events), pedestrians' movement influences safety concerns.

One major approach to obtain pedestrian trajectories is to use video camera and track each individual pedestrian from the video camera from the time he/she is detected on the scene until he/she leaves the scene. However, given the normal video frame rate of 25-30 images per second, to track hundreds of individual pedestrians manually would requires herculean effort despite its accuracy. Better approach is to automate the tracking system.

While such automation in tracking in video camera is not new in the literatures of computer vision (e.g. see [1], [2], and [3] for good survey on the field), most of them are performed only for a relatively few pedestrians. Relatively few recent papers (e.g. [4]), however, reported to track pedestrian on a crowded situation in short-medium range sometime up to maximum distance of 8 meters from the camera.

Most of the person detection technique works better in a well controlled environment such as laboratory setting with a small amount of people [2] whereas other techniques require more than one camera with overlapping fields of view in order to cope with the pedestrian segmentation problem in the a dense crowd [5].

In the realistic environments, while walking, the people form groups and then separate from one another, and more over they have shadows [1] and the object, especially, people undergo a change in their shapes while moving and their motion is not constant that cause the tracking is a difficult problem [3].

This paper describes our attempt to automate such tracking of more than a hundred pedestrians at once as illustrated in Fig. 1. The scene was taken during pedestrian experiment in Ateneo de Manila University, Philippines for the study of microscopic pedestrian. The video was taken from the third floor of a building (about 10 meter height) at unknown angle. The camera was not calibrated.

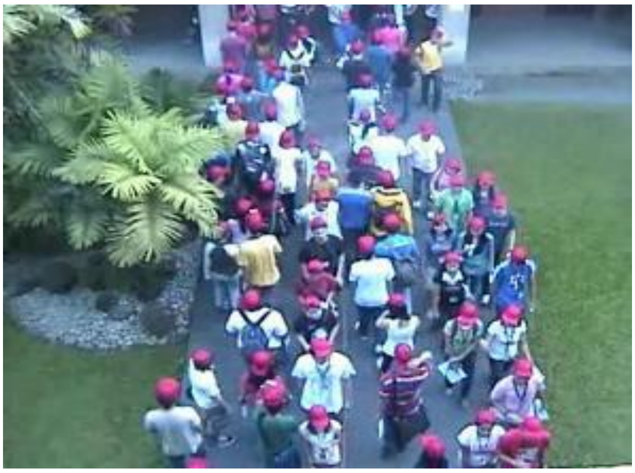


**Fig. 1** Challenging scene of a crowded situation where more than 120 pedestrians walk in a narrow passage

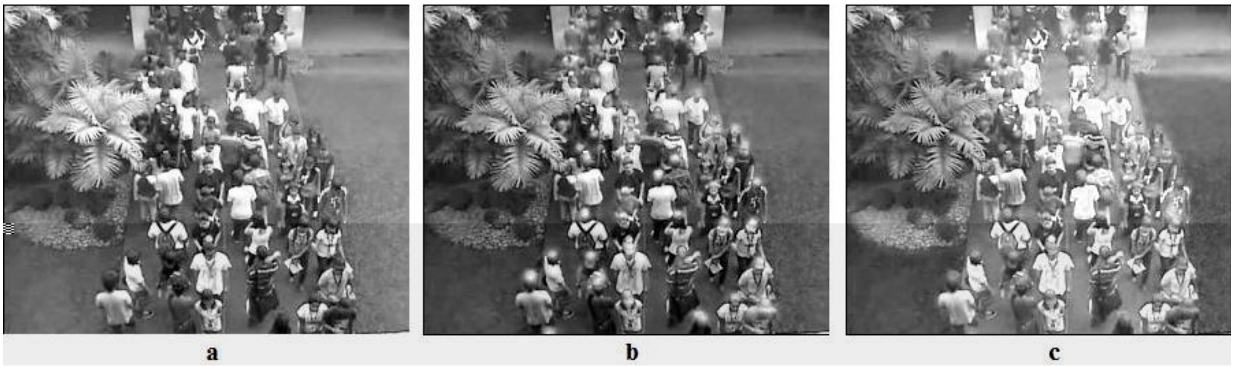


**Fig. 2** Splitting channel into a) green, b) red and c) blue channels indicates that the red hats are clearly marked in red channel

Tracking pedestrians in a crowded situation is a hard problem because pedestrian does not move as a rigid body. Human body goes through a large range of variation during walking. In crowded situation, due to the angle of the camera, the bodies of pedestrians which are farther from the camera are occluded by other nearer pedestrians. Furthermore, in outdoor scene, the lighting condition is uncontrolled and it may create shadow that hinders correct detection of pedestrians.

Our goal is to track all pedestrians in the scene and save the data of tracking as a NTYX table where, N is the pedestrian number, T is time in video, and X and Y are the coordinate position of the pedestrian in the image that readily converted into scene coordinate.

To solve the occlusion problem, we use a priory knowledge that all pedestrians are wearing red hat during the experiment and the camera are located a position that is high enough to cover the head of all pedestrians.

The rest of the paper will describe our algorithm to detect the object, to analyze its features and to track them.

## 2 OBJECTS DETECTION

Unlike most of the previous approaches that the first step of tracking is removing background [3], we do not remove the background to detect the pedestrian. Using the priory knowledge that pedestrians are wearing red hat, the detection system is directly related to this color information.

Our system is working on an AVI movie; this type of movie is recorded in RGB format. First, we split the video sequence into three channels of Red, Green and Blue (RGB) based on the color information as illustrated in Fig. 2.

High value of red channel is associated with reddish color ranging from yellow, pink to red while greenish and bluish color have low values in the red channel. Similarly, high value of green channel is associated with greenish color ranging from yellow to cyan. Our goal is to extract only the red hat information by removing other unnecessary color information. Subtracting green channel from the red channel will produce color of high reddish value and very low greenish value at the same time. That is equivalent to the range of violet to red. Since we also have prior knowledge that the road color was dark, the Blue channel does not carry much information to be subtracted from the red.

Therefore our object detection algorithm is simply a subtraction of green channel from the red channel and put some threshold to make them binary image as shown in equation (1) and (2).

$$i(x,y) = r(x,y) - g(x,y) \qquad (1)$$

$$f(x,y) = \begin{cases} 1 & if\ i(x,y) > \phi \\ 0 & otherwise \end{cases} \qquad (2)$$

Higher threshold value in equation (2) will produce cleaner blob but it will also remove small blobs. Thus, higher threshold produces higher false negative because some farther object will not be detected. Smaller threshold, on the other hand, yield bulkier blobs that introduce false positive that several noises are detected as object and some blobs are merged. Thus, the value of the threshold value was chosen in such a way so that it minimizes the total of false positive and false negative. After several experiments, we determined that the threshold of 70 (out of 255) yield sufficiently close to the optimum value.

After some noise reduction, the binary results are illustrated in Fig. 3. The black blobs are the red objects from the scene.

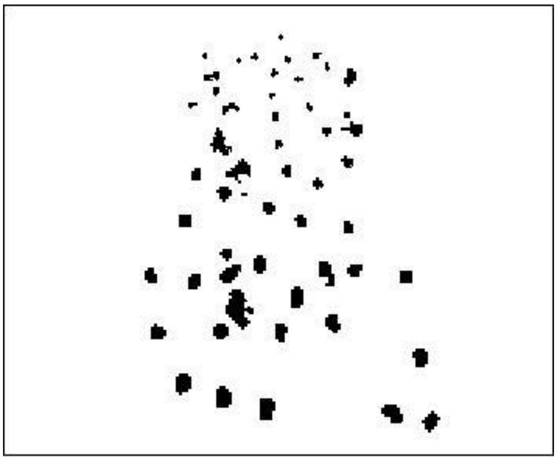


**Fig. 3** Binary Image of detected red objects

## 3 OBJECTS ANALYSIS

After all objects in the frame are detected using connected component algorithm, we analyze each detected object for their area and centroid coordinates. We developed macro for ImageJ to run the particle analysis for this purpose and the results of such analysis are shown in Fig. 4 where each detected object are outlined and numbered with unique ID throughout the image sequences.

Fig. 5 shows example of table results from the particle analysis. The table consist of unique identification number for each blob, centroid coordinates (X,Y), slice or frame number, and area of the blob (in pixels). Blobs with too large area than normal head are truncated. The object ID in this table is per frame. The same object in different frame will have a different identification number.

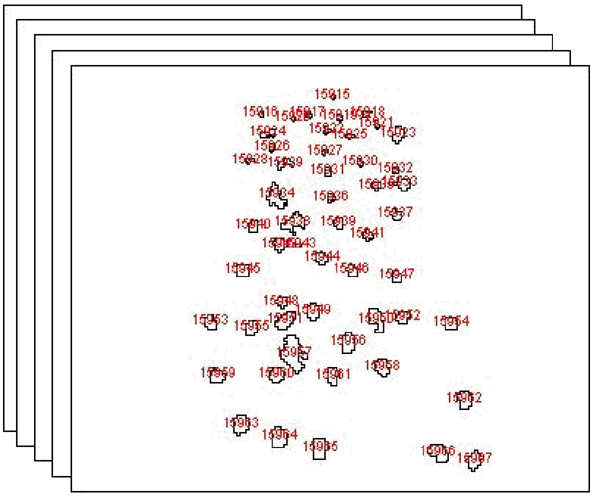
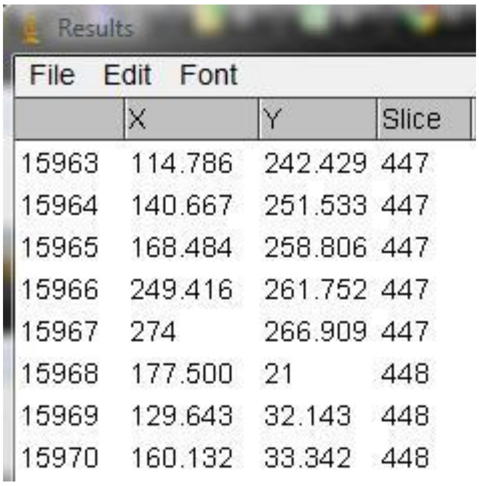


**Fig. 4** Results of particles analysis of the binary image. Notice that each detected object is outlined and numbered



| | X | Y | Slice |
|---|---|---|---|
| 15963 | 114.786 | 242.429 | 447 |
| 15964 | 140.667 | 251.533 | 447 |
| 15965 | 168.484 | 258.806 | 447 |
| 15966 | 249.416 | 261.752 | 447 |
| 15967 | 274 | 266.909 | 447 |
| 15968 | 177.500 | 21 | 448 |
| 15969 | 129.643 | 32.143 | 448 |
| 15970 | 160.132 | 33.342 | 448 |

**Fig. 5** Tabulated result of particle analysis

## 4 TRACKING

The last part of our system is to track the red hat base on the tabulated results of the particle analysis. The results of object analysis in the last section produce only unique identification number for each detected object. However, the result of object analysis does not distinguish red hat from any other red moving object such as red bag or red shirt of the pedestrians. It is the task of the tracking system to detect the object correctly. The purpose of our tracking system is to process the particle analysis table such that we can find same pedestrian in different slides and give them same unique pedestrian identification number. To perform such tracking, we developed our own tracking algorithm which summarized in Fig. 6.

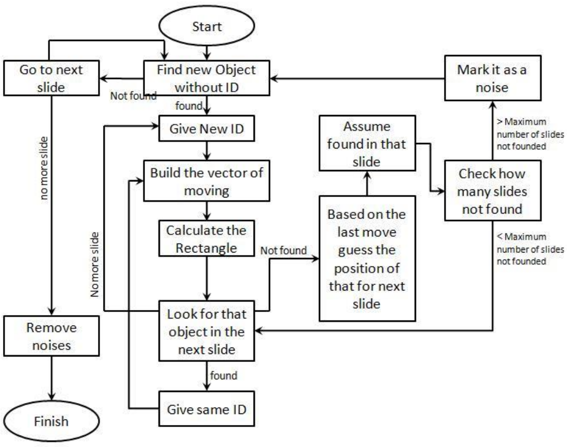

Start
Go to next slide
Find new Object without ID
Mark it as a noise
Not found
found
Give New ID
Assume found in that slide
> Maximum number of slides not founded
Build the vector of moving
Check how many slides not found
no more slide
Calculate the Rectangle
Based on the last move guess the position of that for next slide
Not found
< Maximum number of slides not founded
No more slide
Remove noises
Look for that object in the next slide
found
Give same ID
Finish

**Fig. 6** Flowchart of tracking algorithm

First, our tracking system will search for the detected objects that have not been track yet. Once it found, new pedestrian identification number is given and it continues to search for the same pedestrian in the next slides. The search is based on the position of object in the last frame and the velocity vector of moving that object between last two frames. The system will search within the neighborhood of the previously detected pedestrian. The neighborhood is a fixed size rectangle with the centroid of estimated position for that pedestrian in the center of the rectangle. The estimated position will calculate based on the vector of moving. Vector of moving will be calculate in the following formulas.

$$\mathbf{v}_t = \begin{bmatrix} x_t - x_{t-1} \\ y_t - y_{t-1} \end{bmatrix} \tag{3}$$

$$\mathbf{v}_t = \alpha \mathbf{v}_{t-1} \log y + \beta \tag{4}$$

Once system calculates the estimated position, it search all detected blobs in the neighborhood rectangle as illustrated in Fig. 7.

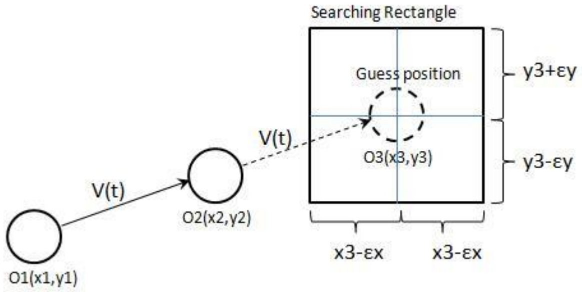


**Fig. 7** Finding the estimated position and searching neighborhood rectangle

If more than one objects are found in the neighborhood, the system will select based on nearest neighbor, nearest area and similarity of the velocity vector.

When the system cannot find any objects within the vicinity of the rectangular neighborhood, the search are continued into the next several frames based on similarity of the average velocity vector that has been projected (due to the inclination of the angle of uncalibrated camera). The velocity vector is used to estimate the next position of the pedestrians. Using this estimated position, the neighborhood search is performed in the next frame. If the system cannot find the similar object for a maximum number of frames, then it concluded that detected object was not a pedestrian and, then it marks this object as a noise. The control flow goes back to the first frame where the object was found and looks for the next object.

When all the objects in all slides are tracked, then the algorithm will report the results.

With this simple tracking scheme, we were able to find the correct red hats and removing the noise. After tracking the entire objects in all frames, out of 128 pedestrians, our algorithm was able to detect with false positive of 145 pedestrians which gives 86.7% correctly detected.

## 5 CONCLUSION

We have described simple algorithms to detect, to analyze and to track pedestrian from uncalibrated angle camera for outdoor crowded situation where occlusion is clearly a problem. The system was able to recognize pedestrians at reasonable rate of misclassification. The result of such tracking is a database of pedestrian trajectories over time and space. Improvement of such system using multiple views and to increase the rate of correctly classified would be subject to further research study.

## REFERENCES:

McKenna, Stephen J. (2000). Tracking Groups of People. Computer Vision and Image Understanding Vol. 80, pp. 42–56.

Philip Kelly, B.A. (2007). Pedestrian Detection and Tracking Using Stereo Vision Techniques. PhD. Dissertation, Dublin City University, Dublin.

Javed, Omar, and Shah, Mubarark (2008). Automated Multi-Camera Surveillance: Algorithm and Practice, Springer, New York.

Kelly, Philip, O'Connor, Noel E., and Smeaton, Alan F. (2009). Robust pedestrian detection and tracking in crowded scenes, Image and Vision Computing Vol. 27, pp. 1445-1458.

Eshel, Ran, and Moses, Yael. (2008). Homography Based Multiple Camera Detection and Tracking of People in a Dense Crowd. IEEE Conference on Computer Vision and Pattern Recognition.