Estimating Speeds and Directions of Pedestrians in Real-Time Videos: A solution to Road- Safety Problem

Sultan Daud Khan

Complex Systems and Artificial Intelligence Research Center
Department of Informatics, Systems and Communication
University of Milano-Bicocca
sultan.khan@disco.unimib.it

Abstract. Pedestrian injuries and fatalities are one of the most significant problems related to travel and road safety. Pedestrians are vulnerable users of roads and due to the very different velocities and mass when compared to vehicles like cars and trucks, and very often they undergo serious injuries in case of collisions. Older pedestrians are even more vulnerable to injuries and fatalities due to (i) their reduced mobility and reflexes and (ii) their increased fragility when compared to young individuals. Crosswalks are the point where pedestrians face lower level of safety because they have to cross the street and must be aware of the incoming traffic. Such kind of awareness becomes difficult in case of old pedestrians because of their reduced physical and perceptive capabilities. Besides other factors, lower speed of an old pedestrian is an important factor that limits the mobility of old pedestrians and it also increases the risk of fatalities while crossing the road. In this paper, we developed vision based intelligent system that can detect low speeds and directions of pedestrians and can help him/her by (a) increasing the time associated to a green light for pedestrians, (b) using audible signals to help the pedestrians understanding that there are cars approaching the crossing.

1 Introduction

Pedestrian injuries and fatalities are one of the most significant problem related to travel and road safety. Pedestrians are the vulnerable users of road and due to the very difference in speed and mass when compared to vehicles like cars and trucks, are very often they undergo serious accidents. Studies have shown that more than one fifth of the pedestrians killed while crossing the roads [1]. About 273000 pedestrian were killed in road traffic crashes in 2010 [2]. 15% of the total numbers of people killed were pedestrians in European Countries [3]. Walking is a basic mode of transport in all societies around the world. Walking is good for health and particular for the cardiovascular patients. Due to increase in number of vehicles and population, people prefer to walk rather than taking a car particularly in the cases when their destinations are near. Of course, these pedestrian will use the roads but due to some risk factors like difference speed, alcohol, lack of road infrastructure for pedestrian and lack of other perceptive capabilities will lead them to injuries and sometimes to fatalities. It is also stated that most of these accidents took place in urban areas. Urban areas are most populated places relative to rural areas. Urban areas are equipped with wide road and huge traffic flow which makes it difficult for pedestrians to move. Pedestrian injuries and fatalities also have psychological, socioeconomic and health costs. Although there is no estimation for economic impact of pedestrian injuries but road traffic crashes consumes 1% to 2% of gross national product [4]. Also the survivors of traffic crashes, their families and friends often suffer intense social, physical and psychological effects. Pedestrians form mixed group of people in terms of age, gender and socioeconomic status. Studies have showed that pedestrian crashes are related to risk factors and road geometrical factors [5]. These factors include: 1) age and gender 2) pedestrian crossing time 3) pedestrian crossing speed 4) crossing the street with red or green traffic light. Pedestrian crashes affect the people from different age group. Studies have shown that in United States in 2009, the fatality rate for pedestrians older than 75 years, higher than the fatality rate of any other age group [6]. Literature suggests that pedestrians of age older than 65 years have high accident risk than any other age group. There are many reasons involved in the high fatality rate for pedestrian older than 65 years. These include: 1) deficits in their physical abilities 2 sensory and perceptual abilities 3) cognitive abilities. The aged population of most of developed countries like japan and European Countries like Italy etc. are increasing. The rate of old population is expected to increase by a 20% by year 2031 [7]. Older pedestrians face major problems and accidents may occur as a result of age-related decline skills used while crossing the road. These include 1) motion perception 2) memory capacity 3) reaction time and physical mobility such as the ability to rotate neck, walking and muscle control, balance and postural control [8] . There are two components of motion; distance travelled and speed and it is believed that age differences in motion perceptions are the cause of certain accidents. Crosswalks are the point where pedestrians face lower level of safety because they have to cross the street and must be aware of the incoming traffic. Many studies have examined the behavior of pedestrians crossing the road by analyzing several factors. [9] Analyzed the behavior of 1392 pedestrian in signalized crosswalks. They made hypothesis that pedestrians are more optimistic of crossing the road with red traffic light if another pedestrian crossed before him. Moreover, men are more optimistic to cross the road with red traffic light than woman most unsafe choices were taken by old pedestrians in [10].Old pedestrians face many problems as other participants in the community, particularly in transport domain. Older pedestrians often report inability to complete crossings in the time given by pedestrian light. Keeping in view the above discussion, our work is motivated by two factors. One is that the proposed problem has a great social meaning. Secondly, our proposed solution, which makes it different from existing approaches, is focused on detecting the old pedestrians crossing on the basis of the low speeds and help him/her by (a) increasing the time associated to a green light for pedestrians, (b) using audible signals to help the pedestrians understanding that there are cars approaching the crossing. Previous research on traffic signal control was mainly focused on vehicle monitoring, and very little literature can be found on pedestrians side. An approach to detect and count pedestrians at an intersection using fixed camera is proposed in [11]. Background subtraction is employed for motion segmentation and median filtering and erosion/dilation operations are performed to reduce noise. Connected components are extracted and information about the size and coordinates of each connected component is used to compute the number of people in the scene. Computer vision based multi-agent approach is presented in [12], where each agent makes decisions according to local variables and information received from other agents. The problem of object tracking in an uncontrolled urban environment is discussed in [13]. A specific motion detection algorithm was used to detect objects such as pedestrians, vehicles, etc. The motion detection algorithm proposed is based on construction of a reference edge image of the background, composed of all stationary edges in the scene. A single camera looking at an intersection point is used in [14]. The authors focused on motion tracking. Motion segmentation is performed using an adaptive background model that can gain robustness with respect to the changes in illumination while tracking of objects is performed by computing the overlap between bounding boxes. In the above works frame difference method are used which can not completely extract all the information regarding foreground area, the central part of the target will be lost which ultimately result in bad tracking. Alternative to other approaches a vision-based intelligent pedestrian crossing system is developed in [15] using stereo vision approaches which detects the pedestrians and help those who need longer to cross the road. In this paper, we propose a vision-based approach that can detect the pedestrians efficiently and automatically. The proposed system is not only applicable to road safety problem but can also be applicable to security systems inside building. For robust foreground segmentation we use Lucas-Kanade optical flow [16] and Gaussian Mixture Model [17]. The perfect background can not be obtained by optical flow and GMM methods individually. After foreground segmentation, we apply Lucas- Kanade Tracker that track the points of pedestrian from frame to frame and calculate the instantaneous velocity and average speed of pedestrian is determined by calculating the scale factor. Also, in this paper, we estimate the direction of the pedestrians. The rest of the paper is organized as follows. In Section 2, we shall describe motion detection techniques. In Section 3, we discuss tracking. In Section 4, we shall discuss our proposed framework and in Section 5, we shall discuss experimental results.

2 Motion Segmentation

Motion segmentation is the most important pre-processing step for detecting the moving objects from the video. Traditionally in video surveillance with a fixed camera, researchers tend to find some sort of motion in the video. There are two part of such of videos, background and foreground part. The object in motion is the foreground part of the video and the rest static part is the background.

Motion detection is used to extract foreground part from the video. Such kind of extraction is useful for detecting, tracking and understanding the behavior of the object. A survey on motion detection techniques can be found in [18]. There are two types of motion: 1) large-scale body movements like movement of head, legs and arms [19], and 2) small scale body movements like hand gestures and facial expressions [20], [21].

A. Foreground Segmentation: A popular and traditional foreground object segmentation method is a background subtraction. It calculates the difference between current image and background image and detects the foreground by setting up the threshold value. Pixels in the current frame that deviate significantly from the background are considered to be moving objects. These foreground pixels are further processed for object localization and tracking. This technique is prone to errors when there is a change in illumination in the video.

B. Approximate Median: median filter is another technique for motion segmentation. It buffers N number of frames and median of these frames are calculated which will be the background reference frame. This method is effective in some cases but many frames have to be stored for calculating median frame which makes it not suitable in most of cases. Also this method will end up with errors if there is change in illumination.

C. Gaussian Mixture Model: The GMM is one of the most commonly used methods for background subtraction in most of visual surveillance applications. A mixture of Gaussians is maintained for each pixel in the image. with time to time, new pixel values updates the mixture of Gaussians using an online K-means approach. This updating of mixture of Gaussians is used to account for illumination changes, sensor movements and noise.

D. Temporal Differencing: Another common approach for motion segmentation is the temporal differencing. In temporal differencing, video frames are separated by a constant time interval and compared to find the regions that are changed. A small time interval between the frames can increase the robustness to illumination changes. Temporal differencing approach is computationally inexpensive but in some cases it fails to extract the shape of the object and cause small holes.

E. Optical Flow: optical flow estimates the motion by matching points on objects over multiple frame using vectors. Optical flow technique gives more accurate estimation of motion if the frame rate is high. Horn and Schunck [21], Lucas and Kanade [15] , and Szeliski and Couglan [22] are popular techniques for calculating optical flow. A comparison of these methods can be found in [23].

3 Object Tracking

Tracking is defined as the problem of estimating the trajectories of objects in image plane. Video surveillance has motivated many researchers to explore tracking techniques. Tracking moving objects in a video sequence is a difficult job. Occlusion makes tracking a difficult problem for the researchers. In video surveillance, normally some features of the moving objects are extracted and tracking that object using those features. Selecting good features that can be used for tracking is very important, since the object appearance, color and orientation may change from frame to frame. So we need to extract those features that can be tracked for a long period of time.

4 Proposed Methodology

Pedestrian injuries and fatalities are most significant problems related to road safety. Crosswalks are the point where pedestrians face lower level of safety and most of time they end up with serious accidents. Therefore, as a solution an automatic monitoring system based on computer vision technology is needed that can automatically detect the behaviors of pedestrians on crosswalks and alarm the system in order to prevent pedestrian vehicle collisions. Figure 1 shows the methodology, video is streamed from the camera to the system and system processes the video frame by frame. From each frame, foreground is segmented which represents the objects of interest (pedestrians). Blob analysis is performed to find out the independent blobs of a particular size. Corner points are extracted from each bounding box. Later on, these points are tracked through number of frames using Luckas Kanade point tracker. Instantaneous velocity of each point related to bounding box is calculated and average speed of the object is determined by equation discussed in the following section.

4.1 Foreground Segmentation

Identifying moving objects in video sequence is a fundamental and critical task in video surveillance and gesture recognition in human-machine interface. Foreground segmentation is an important pre-processing step for detecting moving objects from the video. Traditionally, background subtraction method is used for extracting moving objects from the video frame where pixels in the currents frame that deviate significantly from the background are considered as part of moving objects. Such kinds of methods are usually prone to errors due to unpredicted and changing behavior of the pixels. In addition, this method can not accurately detect fast moving or slow moving as well as multiple objects. Also these methods are affected by change in illumination in the video frame. Some time change in illumination in static background will be detected as part of moving object. Such errors and noise must be removed from the foreground objects before applying blob analysis and tracking.In order to extract valid and accurate foreground objects, we employed both Gaussian mixture model and Lucas Kanade optical flow as in [24]. There are five popular optical flow methods: gradient based algorithms, block based algorithm, energy based, phase-based algorithm and neuro-dynamic algorithms [25].As optical flow can not get rid of change in illumination so we use Gaussian Mixture Model in combination with optical flow to extract an accurate foreground objects. GMM is more robust to light changes and slight movements for small image sequence. In order to obtain a foreground object without noise, we make use of both LK optical flow and

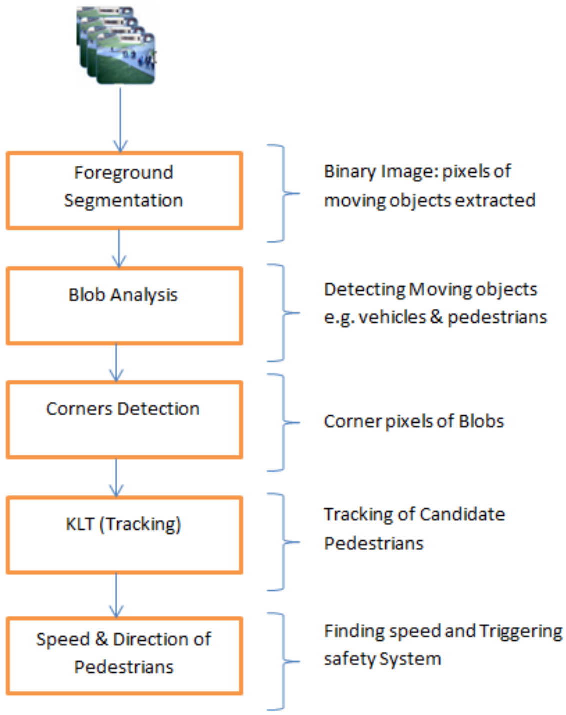


Fig. 1. Flow Chart of Proposed Framework

GMM method as show in Figure 2.As shown in Figure 2, Lucas Kanade optical flow and GMM are applied in parallel to the input image.LK optical flow is applied to two adjacent images of video i.e. $f\ (x,\ y,\ t_1)$ and $f\ (x,\ y,\ t)$ and output of LK optical flow will be in the form of magnitudes. LK optical flow tends to find the corresponding points on $f\ (x,\ y,\ t_1)$ on next frame $f\ (x,\ y,\ t)$.we discuss LK optical flow in more detail in the following section. After calculating LK optical flow between two frames, we use threshold Thof to segment motion from static background. Thof value can be calculated experimentally. The range of T_hof is [0.005 0.002] which is calculated experimentally. The range of T_hof for different videos is different. For slow moving objects the value of T_hof will be low and for fast moving objects the range of T_hof will be high. It is matter of fact that fast moving objects generates optical flow vectors with high magnitudes and slow moving objects generate flow vectors of lower magnitudes. So for the videos, where objects moving with different speeds, mean value of all flow vectors should be taken as T_hof. In extracting moving part from the image, the pixel with large magnitude than Thof will be classified as foreground while the pixels whose magnitudes are less than T_hof will be the part of background. In the same way, we get foreground objects by applying GMM, but the GMM ends up with errors as shown in Figure 2. So in order to extract accurate fore-

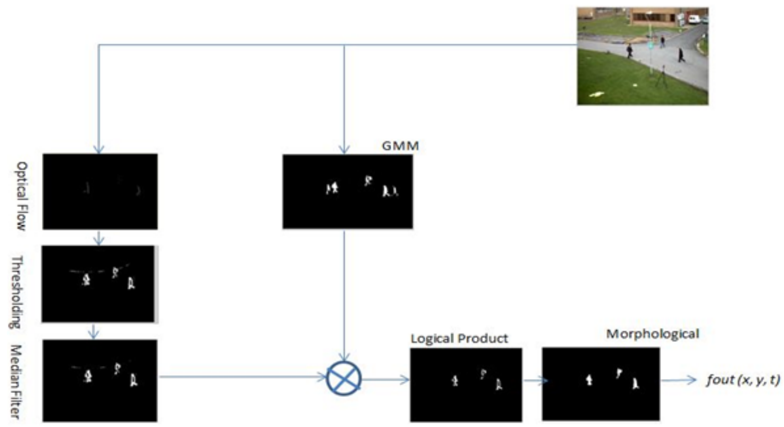


Fig. 2. Framework of extracting accurate background

ground we apply logical product of foreground mask generated by LK optical flow and GMM. Later on, we apply Morphological processes like morphological opening and closing on the binary image generated by logical product of LK optical flow and GMM. The morphological open operation is erosion followed by dilation eliminates smooth contours and protrusions while morphological close is dilation followed by erosion smooth the section of contours, eliminates small holes and fills gaps in contours. These operations are dual to each other. Later on, flood fill algorithm is applied to fill small holes. The output image $fout$ (x, y, t) from Morphological processing block contains accurate foreground objects while will applied to Blob analysis block for detecting moving objects.

4.2 Blob Analysis

Blobs are the connected regions in the binary image. The purpose of blob analysis is to detect those points or regions in binary image that are different from other part of image in terms of brightness or area etc. Following are the steps for finding connected components in the binary image.
1. Search for unlabeled pixel
2. Label all the pixels in connected region containing p by flood fill algorithm
3. Repeat step 1 and step 2 until all pixels are labeled.
In the next step, we measure the area of each connected component. Area of connected component is the number of pixels in the region. There may be different moving objects in the video frame with different area sizes. In Transportation surveillance system, video frame contains multiple objects like pedestrians, vehicles of different sizes. In this paper, we are interested only in pedestrians whom we want to detect and track over multiple frames. The area of pedestrians is normally contains less number of pixels than vehicles and trucks in other words

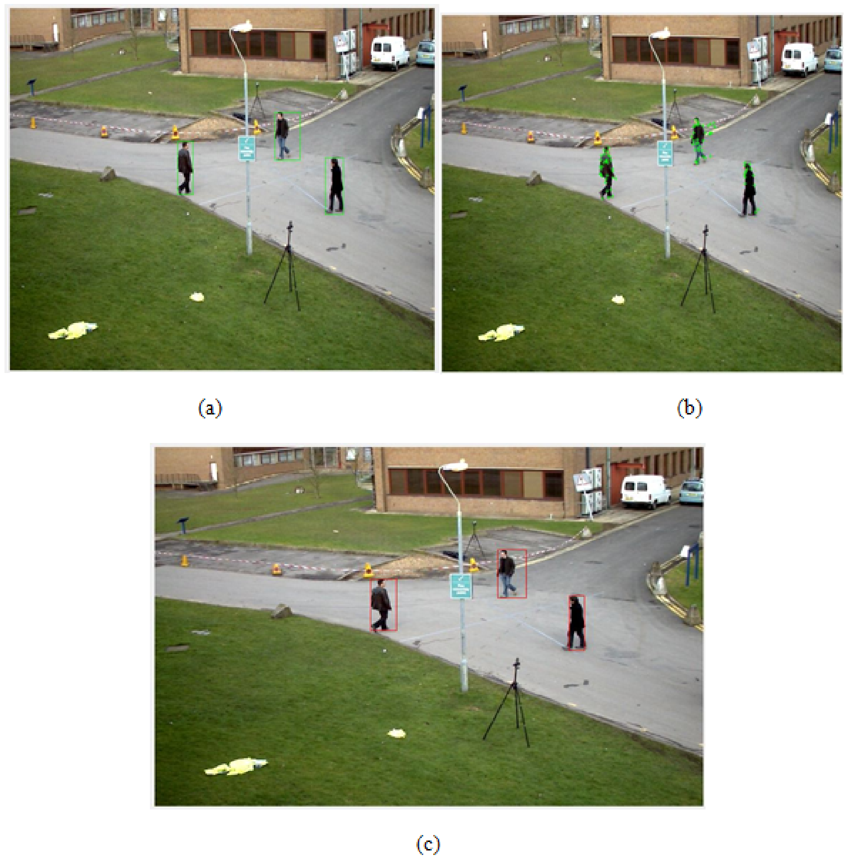
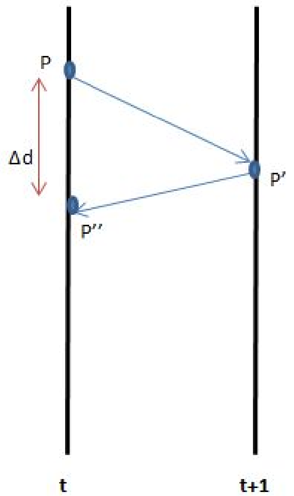


Fig. 3. : (a) shows bounding boxes with detected pedestrians (b) shows the corner points extracted from bounding boxes (c) shows tracked pedestrian with updated bounding boxes

the size of pedestrian is less than vehicle and trucks. On the basis of this assumption, we set the upper and lower bound of blobs area. The upper and lower bound of blobs area can be found experimentally. The size of blob also depends on the resolution of frame. In this paper, we use videos of resolution 576 x 768 pixels and lower and upper bound of blobs area is [1000 2500] pixels. The connected component (object) will be classified as pedestrian if its area lies within the upper and lower bound otherwise it will be discarded.

4.3 Corner Points Detection and Lucas Kanade Point Tracker

In order to track objects detected in blob analysis step, we use KLT feature tracker [26] where motion is detected by using pyramidal Lucas- Kanade optical flow method, using Shi and Tomasi feature detection algorithm [27]. They pyramidal implementation of Lucas- Kanade gives more robustness against huge movement with different speeds. There are two categories of optical flow algorithms 1) sparse optical flow 2) dense optical flow. Dense optical flow methods like Horn- Schunck [21] estimates the displacement of all pixels of image while sparse optical flow algorithms, such as Lucas-Kanade approach estimates the displacement for selected number of pixels. Sparse optical flow gives more robustness towards noise. In sparse optical flow methods, the selection of pixels should be done automatically and done wisely. There are various methods in literature for selecting features to be traced correctly. Robust features can be found and tracked over multiple frames using scale invariant feature transform (SIFT) algorithm [28]. Speeded-Up Robust Features (SURF) [29] also presents a method for extracting interest points and describe these points for fast comparison. Canny edge detector [30] detects edges in image. Then image is divided into small blocks and we search for the closest edge pixel from the center of the block. If the pixel is found, it will be regarded as feature point for that block. Harris Corner detector [31] detects the points where two edges are detected. One of the disadvantage of SIFT and SURF algorithms is that they are computationally expensive. The comparison of all these feature detection techniques can be found in [32]. In this paper, we use Shi and Tomasi algorithm for extracting corner points. Figure 3 (a) shows the pedestrian detected during blob analysis step. In the next, we detect corner points of each bounding box using Shi and Tomasi corner detector as shown in Figure 3 (b). But through our experiments we realize that points detected in the first frame may not be tracked over multiple frames. This is due to dramatically change in the appearance of the objects and change in intensity values of pixels. Such kind of change always results in tracking failure. Here, it should be noted that our aim is to find the instantaneous velocity of valid pixels. A pixel will be valid pixel if its forward and backward trajectory does not differ significantly. [33] Purposes a method that can automatically detect the tracking failure by forward and backward tracking of pixels. As shown in Figure 4 (a) , point p on frame t is to be tracked on frame $t+1$. Let p is a point detected on frame $t+1$. This is a forward trajectory of point p. in order to check the validity of the pixel, the point location p will track back the point on frame t. this is backward trajectory. Let p is the point location tracked during backward trajectory. The difference between the two trajectories is Δd. In ideal case, the value of Δd is zero. So here, we define a threshold value for d. experimentally the value of d is in range of [1 3]. The higher the value of Δd, the more the pixels will be in errors and hence speed of pedestrian will be erroneous. It should be noted that before calculating the speed of pedestrian, invalid pixels must be removed. The pixel will be invalid if its related Δd is greater than a threshold value otherwise it will a valid pixel. Through our experiments, we have observed that among group of valid pixels there are some pixels which appear to be valid but in actual are invalid pixels. Such group of pixels are static and do not move with the object. So for accurate result, we remove all invalid and static pixels and consider only valid pixels.



Vectors	Validity	Vectors	Validity
1	1	16	1
2	1	17	1
3	0	18	1
4	1	19	1
5	1	20	1
6	0	21	1
7	1	22	0
8	1	23	0
9	1	24	1
10	1	25	0
11	0	26	1
12	0	27	0
13	0	28	1
14	1	29	1
15	1	30	1

Fig. 4. (a) Forward Backward error. (b) Validity of all points.

Figure 4 (b) shows the points of one of pedestrian in frame that were detected by Shi Tomasi corner detector. *0* represents the points (pixels) whose Δd is greater than threshold range when tracked over next frame and hence regarded as invalid pixels while *1* represents the valid pixels. We consider only valid pixels for finding the speed, direction and understanding the behavior of pedestrian.

4.4 Estimation of Speed

In order to find the speed of pedestrian, we must use the valid points. To find the speed of pedestrian, instantaneous speed can be found by using successive image frames of the video. This instantaneous speed for each pixel can be found by Equation 1.

$$v = \Delta d / \Delta t \tag{1}$$

Where v is the instantaneous velocity of a valid point. d is the change in displacement of valid point over successive frames.Δt is time interval between two successive video frames and is equal to the frame capture rate of the camera. In our experiments, Δt is 33.33 milliseconds. To find the accurate speed of pedestrian, only one valid point is not enough. We need to find out valid points for the same pedestrian, calculate the instantaneous velocities of those points. Then by averaging instantaneous velocities of all valid points. Lets assume that n valid points are selected from the pedestrian and let vi represents the instantaneous velocity of point i. where $i = 1....n$. By using those instantaneous velocity vectors, we can find the instantaneous velocity of pedestrian by Equation 2:

$$Viv(t) = \sum_{i=1}^{n} vi(t) \tag{2}$$

Where Viv is the instantaneous velocity of a pedestrian at time t, $vi(t)$ is the instantaneous velocity vector of ith point and n is the number of valid points that tracked. $vi(t)$ in Equation 2, is calculated in pixels per second because we measure the displacement Δd between two frame in pixels which has no correspondence with real displacement. As we know, that objects closer to the camera will have large pixel displacement than the objects far from the camera although both objects are moving with same speed in real world. Thus, it is very difficult to find the accurate speed of pedestrians using only optical flow. In order to overcome this limitation, we calculate the displacement of each pedestrian in the world coordinates. In our case, we calculate the distance between entry and exit point of pedestrians. Later on, we calculate the displacement in pixels and derive a linear scale factor to relate displacement in images to the motion in the world.

4.5 Estimation of Orientation

At intersections and pedestrian crossings, pedestrians frequently change their speed and directions. It is even more dangerous and prone to pedestrian vehicle collisions if the pedestrians change their walking directions in the middle of the road. Therefore to avoid such collisions, driver must know when the pedestrian is going to change his/her walking direction to dangerous area. Therefore, estimating the orientation of pedestrian on pedestrian crossing becomes very important. In this paper, we estimate the walking direction of pedestrian by making use of optical flow vectors of the valid points. Let vel (x, y, u, v) is the velocity vector of a pixel. Where (x, y) is the coordinate of a pixel and u and v are the horizontal and vertical movements of the pixel. As the optical flow field of the foreground image contains all the velocity vectors vel (x, y, u, v), therefore it is easy to get the magnitude r (x, y) and angle. Let δ (x, y) be a direction of optical flow vector at pixel (x, y) in frame t and is given by Equation 3.

$$\delta(x, y) = \tan^{-1}(u/v) \tag{3}$$

As we are interested in finding the orientation of the pedestrian, therefore we consider angle information.

5 Experimental Results and Discussions

We carried out our experiments on a PC of 2.6 GHz (Core i5) with 4.0 GB memory and data set from UCF. As shown in Figure 5 (a), pedestrians cross the road in the opposite directions while the car is moving towards the pedestrians. Studying such kind of scenario becomes very important for understanding the pedestrian/vehicle interactions. And in order to avoid collisions we develop a

Fig. 5. (a)Sample video frame. (b) Foreground segmentation. (c) Tracking using KLT

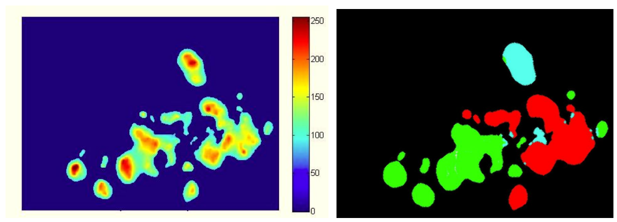


Fig. 6. (a)Estimated speed of pedestrians. (b) Dominant direction of pedestrians.

system that automatically finds the speeds and directions of pedestrians and vehicles. Figure 5 (a) shows a sample frame taken from a video sequence. Figure 5 (b) and (c) shows the objection detection and tracking results after applying our algorithm mentioned in section 4. The most important result are shown in Figure 6. Figure 6(a) shows the different speeds of pedestrians. The color bar shows amount of speed a pedestrian is travelling with. The dark regions shows that pedestrians are moving with high speed while cyan and yellow regions represents relatively low speed. Figure 6(b) shows the directions of pedestrians. As shown in Figure 6 (b) there are two dominant flows, one towards the East and other towards west. Red color shows that pedestrians are moving towards west while green color shows pedestrians moving towards east. The vehicle in cyan color is moving towards south

6 Conlcusions

In this paper we proposed a framework to find the speed and direction of people moving in the video. Foreground objects (people) are extracted by applying Gaussian mixture model and optical flow. Blob analysis is performed to detect pedestrians. Lucas Kanade Tracker is used to track each point across multiple frames. Later on, average speed (pixels/second) of each pedestrian is computed. It is observed that proposed framework worked very well in low density scenarios. We are developing methods and techniques that can automatically map image coordinates to world coordinates and find the actual speed(meter/seconds) of pedestrians and is a part of our future works.

References

1. W. H. Organization *et al.*, "Pedestrian safety: A road safety manual for decision-makers and practitioners." *Geneva and Switzerland*, 2013.
2. J. Whitelegg, *Quality of life and public management: Redefining development in the local environment.* Routledge, 2012.
3. E. Commission, E. Commission *et al.*, "Care community road accident database," *Brussels, EC: http://europa. eu. int/comm/transport/care*, 2005.
4. M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, C. D. Mathers *et al.*, "World report on road traffic injury prevention," 2004.
5. A. Dernellis and R. Ashworth, "Pedestrian subways in urban areas: some observations concerning their use," *Traffic engineering & control*, vol. 35, no. 1, pp. 14–18, 1994.
6. W. H. Organization *et al.*, "Global status report on road safety 2013: supporting a decade of action: summary," 2013.
7. S. for Seniors Working Group (WA), *Safety for Seniors: Final Report on Pedestrian Safety*, 1988.
8. T. Rosenbloom, "Crossing at a red light: Behaviour of individuals and groups," *Transportation research part F: traffic psychology and behaviour*, vol. 12, no. 5, pp. 389–394, 2009.
9. C. Holland and R. Hill, "Gender differences in factors predicting unsafe crossing decisions in adult pedestrians across the lifespan: A simulation study," *Accident Analysis & Prevention*, vol. 42, no. 4, pp. 1097–1106, 2010.
10. V. Bhuvaneshwar and P. B. Mirchandani, "Real-time detection of crossing pedestrians for traffic-adaptive signal control," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on.* IEEE, 2004, pp. 309–313.
11. C. Conde, A. Serrano, L. Rodrıguez-Aragón, J. Pérez, and E. Cabello, "An experimental approach to a real-time controlled traffic light multi-agent application," in *Proceedings of AAMAS-04 Workshop on Agents in Traffic and Transportation*, 2004, pp. 8–13.
12. P. Vannoorenberghe, C. Motamed, J.-M. Blosseville, and J.-G. Postaire, "Monitoring pedestrians in a uncontrolled urban environment by matching low-level features," in *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, vol. 3. IEEE, 1996, pp. 2259–2264.
13. H. Vceraraghavan, O. Masoud, and N. Papanikolopoulos, "Vision-based monitoring of intersections," in *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on.* IEEE, 2002, pp. 7–12.
14. A. Fascioli, R. I. Fedriga, and S. Ghidoni, "Vision-based monitoring of pedestrian crossings," in *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on.* IEEE, 2007, pp. 566–574.
15. B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.
16. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.
17. C. Cedras and M. Shah, "A survey of motion analysis from moving light displays," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on.* IEEE, 1994, pp. 214–221.

18. T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.
19. V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 677–695, 1997.
20. B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.
21. B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1, pp. 185–203, 1981.
22. R. Szeliski and J. Coughlan, "Spline-based image registration," *International Journal of Computer Vision*, vol. 22, no. 3, pp. 199–218, 1997.
23. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.
24. W. Li, X. Wu, K. Matsumoto, and H.-A. Zhao, "Crowd foreground detection and density estimation based on moment," in *Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on.* IEEE, 2010, pp. 130–135.
25. M. Cristani, R. Raghavendra, A. Del Bue, and V. Murino, "Human behavior analysis in video surveillance: a social signal processing perspective," *Neurocomputing*, vol. 100, pp. 86–97, 2013.
26. C. Tomasi and T. Kanade, *Detection and tracking of point features.* School of Computer Science, Carnegie Mellon Univ., 1991.
27. J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on.* IEEE, 1994, pp. 593–600.
28. D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.
29. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006.* Springer, 2006, pp. 404–417.
30. J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
31. C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.
32. N. Nourani-Vatani, P. Borges, and J. Roberts, "A study of feature extraction algorithms for optical flow tracking," in *Australasian Conference on Robotics and Automation*, 2012.
33. Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Pattern Recognition (ICPR), 2010 20th International Conference on.* IEEE, 2010, pp. 2756–2759.