The Study of the Detection and Tracking of Moving Pedestrian Using Monocular-Vision*

Chang Hao-li, Shi Zhong-ke, and Fu Qing-hua

College of Automation, Northwestern Polytechnical University,
Xi'an 710072, China
Chl8201@sohu.com, {zkeshi, fuqinghua}@nwpu.edu.cn

Abstract. To ensure the safety and efficiency of the pedestrian traffic, this paper presents a real-time system for moving pedestrian detection and tracking in sequences of images of outdoor scenes acquired by a stationary camera. The self-adaptive background subtraction method and the dynamic multi-threshold method were adopted here for background subtraction and image segmentation. During the process of tracking, a new method based on gray model GM(1,1) was proposed to predict the motion of pedestrians. And then a template for tracking pedestrian continuously was presented by fusing several characters of targets. Experimental results of two real urban traffic scenes demonstrate the efficiency of this method, then the application of this method is discussed in real transportation system.

1 Introduction

Pedestrian is one of the most principle participants in transportation system, ensuring the safety and smoothness of the pedestrian traffic is an important goals to construct the city transportation system. But in present research, only vehicles are considered[1][3], while pedestrians are mostly neglected. When designing the traffic control scheme, the delay of vehicles and the length of vehicles' queue are the most important targets to evaluate the performance of the system. The obstacles of intelligent vehicles are thought mostly vehicles too. But in real traffic scenes, the influence of pedestrians can not be neglected, especially in the traffic environment of China which is a typical mixed traffic. The information of pedestrians can not only be used to realize a traffic control application such as a pedestrian control scheme at intersection, but also can be used to design the safety system and navigation system of Intelligent Vehicles. Pedestrian detection is the most common approach to obtain the pedestrian information in traffic scenes. It is the foundation to analysis and comprehend pedestrian behavior.

Tracking moving pedestrians in video streams has been an active area of research in computer vision. In recent years, many researchers have begun to do research in this area. Stereovision is the most common method to detect pedestrians[4][5]. It is efficient in middle and short distance, but is too computationally expensive and the system is complex. Neural network[4][5] and the step rhythm[6][7] are used to recognize pedestrians in some systems. This method is complex and not credible. To detect the

* Supported by the national natural science foundation of China (Grant NO. 60134010).

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part IV, LNCS 3994, pp. 878 – 885, 2006.

pedestrian the legs must be detected exactly, so this method will be disabled when the legs can not be detected for some reason. So an approach to detect moving pedestrians in congested traffic scenes based on monocular vision was presented. It includes two course of processing: segmentation and tracking. The self-adaptive background subtraction method and the dynamic multi-threshold method are adopted here for background subtraction and image segmentation. A new method based on gray model GM(1,1) was proposed to predict the motion of pedestrians, which will improve the accuracy. A simple criterion based on multi-features is used for classification and template matching, guided by motion prediction for tracking. Tested using real traffic video sequences, the system are able to track multiple isolated pedestrians robustly.

2 Moving Pedestrian Extraction

Reliable tracking requires that the pedestrians can be segmented out reliably. To be useful, the segmentation method needs to accurately separate pedestrians from the background, be fast enough to operate in real time, be insensitive to lighting and weather conditions, and require a minimal amount of initialization. This can be done by using either models describing the appearances of the targets or a model describing the appearance of the background.

At the images level, we perform background subtraction and thresholding to produce difference images. On the assumption that the camera is still, the background difference method is comparatively efficient and simply to detect moving objects. The moving object can be segmented out according to the following equation.

$$dif\ (i,\ j) = I_k(i,\ j) - B(i,\ j)\ . \tag{1}$$

$I_k(i,\ j)$, $B(i,\ j)$ represent the current image and current background respectively.

2.1 Adaptive Background Segmentation

A self-adaptive background subtraction method is used for segmentation. This method is much simpler and more robust to update the background. In addition, this method is also insensitive to lighting conditions and has the further advantage of not requiring initialization with a background image.

Take the first frame as the initial background. For each frame of video sequence, we take the difference between the current image and the previous image giving the difference image BW_k:

$$BW_k = \begin{cases} 1 & \text{if abs}\ (\ \mathrm{I_k} - \mathrm{I_{k-1}}) \geq T \\ 0 & \text{if abs}\ (\ \mathrm{I_k} - \mathrm{I_{k-1}}) < T \end{cases}\ . \tag{2}$$

Here, I_k, I_{k-1} represent the two continuous images of the image serial, the value of T is 10% of the peak value.

We update the background by taking a weighted average of the current background and the current frame of the video sequence. However, the current image also contains foreground objects. Therefore, before we do the update we need to classify the

pixels as foreground and background and then use only the background pixels from the current image to modify the current background. The binary object mask is used to distinguish the foreground pixels from the background pixels.

$$B_k = \begin{cases} B_{k-1} , & BW_k = 1 \\ (1-\alpha) I_k + \alpha B_{k-1} , & BW_k = 0 \end{cases} . \tag{3}$$

B_k , B_{k-1} and I_k represent the new background, the instantaneous background and the current image respectively. The weight assigned to the current and instantaneous background affect the update speed. We want the update speed to be fast enough so that changes in illumination are captured quickly, but slow enough so that momentary changes do not persist for an unduly long amount of time. The weight α has been empirically determined to be 0.1. We have found that this gives the best tradeoff in terms of update speed and insensitivity to momentary changes.

2.2 Select the Two Thresholds Dynamically

After subtracting the current image from the current background, the resultant difference image has to be thresholded to get the binary object mask. The resulting connected regions are then grouped into pedestrians and tracked. Since the object mask itself is used to update the current background, a poorly set threshold would result in poor segmentation. So the choice of the threshold is critical. The pedestrians' appearances change dynamically, so a static threshold cannot be used to compute the object mask. Therefore we need a way to update the threshold as the current background changes. The difference image is used to update the threshold.

The reason using two different thresholds is for detecting the nature of occlusion. In our images, a major portion of the image consists of the background. Therefore the difference image would consist of a large number of pixels having high values. And the histogram contains mainly two parts: the noise and the statistical characters of the motion object. We use this observation to decide the thresholds. On assumption that the noise parameters obey the Gaussian model, which can be described as follows:

$$f = \begin{cases} f_l = (\sqrt{2\pi}\sigma_l)^{-1} e^{-(x-\mu)^2/2\sigma_l^2} & x \le \mu \\ f_r = (\sqrt{2\pi}\sigma_r)^{-1} e^{-(x-\mu)^2/2\sigma_r^2} & x > \mu \end{cases} . \tag{4}$$

Among these, μ represents the mean value, σ_l, σ_r represents the parameters of the two parts of noise at the both sides of the mean value.

The histogram of the difference image will be filtered using the middle-value filter. Then find the two segmentation thresholds at the filtered histogram. The histogram of the difference image will have high values for low pixel intensities and low values for the higher pixel intensities. To set the left threshold, we look for the first dip in the histogram that occurs to the left of mean value, starting from the pixel value $x = \mu - 3\sigma_l$ corresponding to the histogram. The corresponding pixel value is used as the left threshold T_l. To set the right threshold T_r, the method is similar as looking for the left threshold. We need to look for the first dip in the histogram that occurs to the right of the mean value, and start from $x = \mu + 3\sigma_r$.

2.3 Segmentation

Moving entities are then extracted as follows:

$$BW(i,j) = \begin{cases} 1, & dif(i,j) \le T_l \ \ or \ \ dif(i,j) \ge T_r \\ 0, & others \end{cases} . \tag{5}$$

In order to eliminate noise from being classified as foreground, a threshold is used so that any blob with area smaller than the threshold is deleted from the foreground. Several measures were taken to further reduce the effect of noise. A single step of erosion followed by a step of dilation is performed on the resulting image and small clusters are totally removed. And also, the background image is updated using a very slow recursive function to capture slow changes in the background.

3 Pedestrian Tracking

The purpose of tracking pedestrians is to obtain the tracks of the pedestrians. The key is to detect and track pedestrians continuously. In every frame, a relation between the blobs in the current frame is sought with those in the previous frame. Then the pedestrian in the detection region can be tracked. Achieving robust tracking in outdoor scenes is a hard problem owing to the uncontrollable nature of the environment. Furthermore, tracking in the context of an intersection should be able to handle non free-flowing traffic and arbitrary camera views.

3.1 Detection and Recognition

The individual regions are computed using a connected components extraction method. The various attributes of the blob such as centroid, area, and elongation that can be computed during the connected component extraction.

Although the pedestrians are much different, the shapes of them are similar and the inverse proportion of the height and the width comply with certain criterion. We model the pedestrian as a rectangular patch with a certain dynamic behavior. The area of the pedestrian is smaller and the dispersibility of the shape is larger comparing with vehicle. There we can define the dispersibility as follows:

$$dis = P^2/A \ . \tag{6}$$

Here the equation (6) represents the shape dispersibility; the l, w, P and A stand for the height, width, perimeter and the area. According to the two characters, we can distinguish pedestrians from vehicles. We found that this simple model adequately resembles the pedestrian's shape and motion dynamics for the purpose of tracking.

3.2 Motion Prediction

To improve the efficiency of the approach, we need to have an estimation of where to place the box corner which represents the detected pedestrian with respect to the distance of one bounding box corner and the centroid of the pedestrian are obtained.

Hence, the parameters estimated are the distance of a corner point from the centroid, the length and the height of the pedestrian.

The motion of the pedestrian is random and difficult to depict. Therefore, we choose the grey model GM(1,1) as the motion model to predict the motion of the pedestrian. The future state of the pedestrian is predicted by processing the records using the GM(1,1). The definition of the GM(1,1) is as follows:

Supposed that there is a date serial $X^{0} = \{x_1^0, x_2^0, \cdots, x_n^0\}$, we can get a new date serial $X^{1} = \{x_1^1, x_2^1, \cdots, x_n^1\}$ by accumulating all these dates. In this equation, $X_k^1 = \sum_{i-1}^{k} x_k^0$, so the differential equation of the GM(1,1) is as follows:

$$\frac{dX^{1}}{dt} + aX^{1} = b \cdot \tag{7}$$

The equation $\hat{a} = (a,b)^T$ represents the predicting parameter serial. The following result will be obtained:

$$\hat{a} = (BB^{T})^{-1} BX^{0} \cdot \tag{8}$$

$$B = \begin{bmatrix} -\frac{1}{2}(x_1^1 + x_2^1) & -\frac{1}{2}(x_2^1 + x_3^1) & \cdots & -\frac{1}{2}(x_{n-1}^1 + x_n^1) \\ 1 & 1 & \cdots & 1 \end{bmatrix} \cdot \tag{9}$$

Using these estimated parameters, request the difference equation, and the following prediction model can be obtained:

$$\hat{x}_{k+1}^0 = \left(\frac{b}{1+0.5a} - \frac{a}{1+0.5a} x_1^0\right) e^{-a(k-1)} \cdot \tag{10}$$

Using the model, the previous three positions can be used to predict pedestrians' next position. The new data will be used to update the record. The process is quickly enough and the date needed is few. So the pedestrians' up-to-the-minute movement law can be tracked and the position conditions can be predicted exactly.

3.3 Matching

Tracking is then performed by finding associations between the pedestrians in the current image with those in the previous frames based on the proximity of the blobs. The pedestrian in the current image inherit the timestamp, label and other attributes such as velocity from a related pedestrian. So, the parameters such as condition, shape, area and statistical characteristics are computed for each pedestrian.

Because of the association of the pedestrian motion in temporal and spatial, the position can be considered as the most important feature. First search the area of around the prediction position, if detect nothing then the tracking is failed. Otherwise we can continue judge according to other features.

Although blobs obtained from difference images can be sufficient to decide the location of pedestrians in many cases, the intensity information may be useful to resolve

certain ambiguities. The use of such information in the form of statistical distribution of intensities may add to the robustness of the current system and is worth pursuing. We use the following two statistical characteristics: the mean gray value and the coefficient of the consecutive frames which are computed out for matching.

Then the template of the detected objects can be build according to these dates and used to track moving pedestrians. During the process of tracking, the matching model is build up for each object, so we can judge whether the object appears around the detecting region and the template are matching or not. If succeed, then track the object and the template is updated by the object's new characters. Otherwise, the template remains, and the next frame will be processed. A blob is considered at the pedestrian level only if it is tracked successfully for a certain period of time. This eliminates blobs that appear then disappear momentarily, such as most blobs that appear due to tree motion back and forth.

4 Results and Application

We test the proposed method using the two video sequences acquired by a stationary camera. One is that there is only single people in the traffic scene, the other is there are many pedestrians. The size of the image is 320×240, the image collection rate is 0.1fra/s. The velocity of process is 11fra/s, so it is quickly enough for the system.

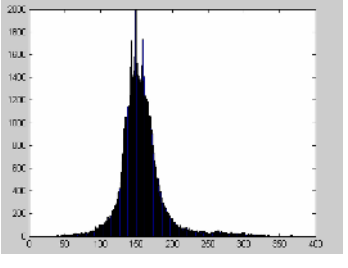

Fig. 1a. The histogram of the difference image

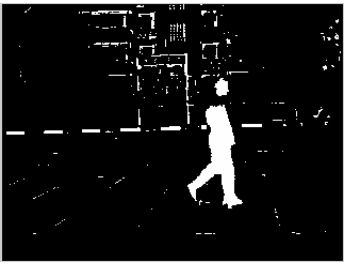

Fig. 1b. The segmented image

The Fig 1.a shows the histogram of the difference image. The calculated thresholds are $T_l = -34$, $T_r = 65$. The Fig 1.b shows the segmentation result using the thresholds.

As is shown in Fig.2, the pedestrian is modeled as a rectangular patch whose dimensions depend on its location in the image. The dimensions are equal to the projection of the dimensions of an average size pedestrian at the corresponding location in the scene. As the following figures shown, pedestrians close to each other and several pedestrians appearing at the same time can be detected exactly.

The results of the GM(1,1) position estimation for a pedestrian are shown in Fig.3. The position estimations of the GM(1,1) are presented against the actual measurements in Fig 3.a. The two tracks are similar and the result showed that this approach can track the pedestrian perfectly. Fig 3.b is the real image showing the detection result, the thinner line rectangle represents the predicted position of the pedestrian, and the wider one is the real position.

Fig. 2. The result of pedestrian detection

Fig. 4. The result of pedestrian tracking

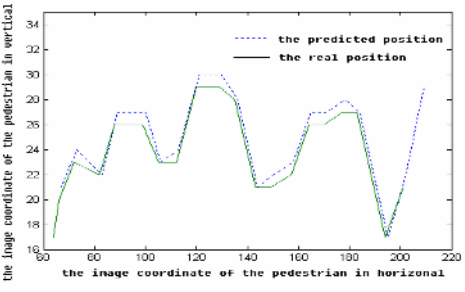

Fig. 3a. The result of tracks comparing

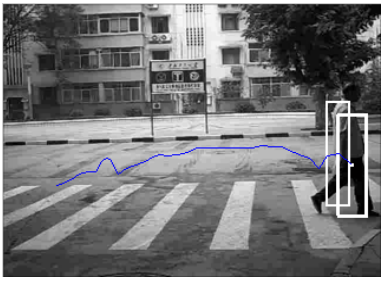

Fig. 3b. The result of position comparing

Fig 4 is the results of tracking. The lines behind the pedestrians show the trajectories of the pedestrians. And the time used to pass the region and the velocity can be calculated out given the frame rate. By analyzing the trend of the coordinate, we can know the moving direction of the pedestrian. At last, the outputs of the system are all these results. For example, in the Fig.4, the pedestrian come into and leave the region at the 231th and the 367th frame. The time is 13.6s. The real length of the region is 15.5m, and the velocity is 1.14m/s. The direction is towards the camera.

In most cases, pedestrians were tracked correctly throughout the period they appeared in the scene. For each pedestrian in the view of the camera, the system produces location and velocity information when the pedestrian is visible. And the speed of each pedestrian can be recorded successfully. It also gave periodic averages of speeds of pedestrians that belong to one of several categories. The system can be applied reliably. The system has a peak performance of 15fra/s. In a relatively cluttered image, the processing rate dropped down to about 11fra/s.

There is a wealth of potential applications of pedestrian tracking. The data can be used in a scheduling algorithm to control walk signals at an intersection. And it can detect and track not only humans in front of or around vehicles but it can also be employed to track several diverse traffic objects of interest. One should note that the reliable detection and tracking of traffic objects is important in several vehicular applications. In this paper, we are mainly interested in applications related to traffic control with the goal of increasing both safety and efficiency of existing roadways. For example, information about pedestrians crossing the streets would allow for automatic control of traffic lights at an intersection. Pedestrian tracking also allows the use of a warning system, which can warn drivers and workers at a work zone from

possible collision risks. The proposed approach also has a large number of potential applications, such as security monitoring, event recognition, pedestrian counting, traffic control, and traffic-flow pattern identification applications emphasize tracking on a coarser level.

5 Conclusions

This paper presents a real-time system for moving pedestrian detection and tracking in images sequences of outdoor scenes acquired by a stationary camera. This approach was tested using two traffic video images, and the potential application is discussed. This approach can detect and track pedestrians exactly and robustly in most cases. The system outputs the spatio-temporal coordinates of each pedestrian during the period when the pedestrian is in the scene.

To improve the precision and stability of the system, the approach should be ameliorated in many aspects. For example, how to track several pedestrians in the clustered traffic scenes; and also how to track pedestrians at night; these are all problems should be solved in the next research.

References

1. Chu J W, Ji L S, Guo L, et al, C: Study on Method of Detecting Preceding Vehicle Based on Monocular Camera. IEEE Intelligent Vehicles, Italy(2004), June 14-17:750–755
2. Surendra G, Osama M, Robert F, et al, J: Detection and Classification of Vehicles. IEEE Transactions on Intelligent Transportation Systems Vol.3(1) 37–47(2002)
3. Osama M, Nikolaos P P, J: A novel method for tracking and counting pedestrians in real-time using a single camera. IEEE Transactions on Vehicles Technology(2001)
4. Wohle J C, Anlauf J, Pörtner T, et al. A time delay neural network algorithm for real-time pedestrian recognition[C], IEEE Intelligent Vehicles(1998) 247–252
5. Zhao L, Thorpe C,C: Stereo and neural network-based pedestrian detection, ITSC(1999)289–303
6. Pai C J, Tyan H R, Liang Y M, J: Pedestrian detection and tracking at crossroads. IEEE (2003)101–104
7. Cristovbal C, Johann E, Thomas K, J: Walking pedestrian recognition, IEEE Transactions on Intelligent Transportation Systems Vol.3(1) 155–163(2000)
8. Alan J L, Hironobu F, Raju S P,J: Moving target classification and tracking from real-time video, IEEE(1998) 8-14