Dhaka Univ. J. Eng. & Tech. Vol. 1(1) 1-5, 2010 (July)

Real-time Vision-Based Pedestrian Detection by Contour Matching

Mohsin Khan, Emon Kumar Dey and Md. Hasanuzzaman

Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh
e-mail: mohsincsedu@yahoo.com, emonkd@yahoo.com, hzamancsdu@yahoo.com

Abstract

Real time Pedestrian detection is a challenging and important task for computer vision-based researchers. This paper presents a novel method for real-time pedestrian detection. In this method motion areas are segmented from real-time image frames captured by a CCD camera. From the segmented and filtered motion area contour is detected using proposed scan conversion algorithm. The normalized contour is then compared with predefined pedestrian shape contour templates and if satisfactory match is found then the pedestrian is localized and bounded using a rectangular box. The system is tested using 4960 image frames and achieved average precision and accuracy rates are 95.72 % and 93.05% respectively.

Keywords : Motion areas segmentation, scan conversion algorithm, contour, pedestrian detection.

I. Introduction

Intelligent machines are playing important roles in different aspects of life now-a-days. Its importance is increasing as natural human-intelligence machine interaction systems have been studied extensively. A **pedestrian** is a person traveling on foot, whether walking or running. There are millions of vehicles in the world and the number is increasing day by day. As the explosive growth in the number of vehicles increases, a large number of road accidents happen every year that causes millions of deaths in a year. But we must not reduce our vehicles because they are important for us. So we should take proper measures to reduce the accidents or number of deaths. Suppose, if a vehicle can detect a pedestrian automatically in the road and can take proper actions to save the pedestrian by its artificial intelligence then the death statistics of the pedestrians will be significantly reduced. The aim of the researchers in this field is to develop a robust computer vision-based pedestrian detection system for assisting intelligent machine such as mobile robot, intelligence vehicle, etc.

Various approaches have been proposed and implemented for pedestrian detection [1]. Many researches are going on to improve the limitations of existing systems. A system proposed by *Philomin and Gavrila can* detect pedestrians in static images by first extracting edges and then matching to a set of exemplars or templates [2]. *Papageorgiou et. al* proposed a pedestrians detection system using support vector machine trained on an over complete wavelet basis [3][4]. But false positive rate of this system was significantly higher. *Mohan et. al.* extended the work of *Papageorgiou and Poggio* and used four components classifiers for detecting Heads, Legs, Left and Right arms separately [5]. The individual results are then combined to detect the pedestrian. Paul Viola *et. al.* described pedestrians detection system that integrates image intensity information with motion information [6]. They used a detection algorithm that scans a detector over two consecutive frames of a video sequence. The detector was trained to take advantage of motion and appearance information to detect a walking person.

A fusion of an automotive radar sensor and a monocular vision system is proposed by Milch [7]. His method used two steps approach for the object detection. The first step generates a target list by using Radar sensor. The list is then hypotheses for presents of pedestrians. In the second step the vision system is used for proving the hypotheses. *Yasuno et. al.* proposed a technique for detecting pedestrian in far infrared images [8]. They set up the height of a camera so that people's head might be present at the vertical central part of images. In infrared images pedestrians belong to the upper range in the grey-level. Therefore, detects the core (especially face) of a head using the P-tile method. In this approach detects the head area candidate using binarization variable threshold in the search region around core, lowering a threshold and it makes the search area for detection body and legs. Another method proposed by Bertozzi *el. al.* described a pedestrian detection technique in infrared stereo image [9]. It is based on the localization and distance estimation of warm areas in the scene. The algorithm groups areas with similar position and considers only results with specific size and aspect ratio. The aim of our research is to develop a system that can detect pedestrian in real-time from a mobile robot or a roaming vehicle.

II. Proposed System Description

Figure-1 shows the proposed pedestrian detection system. This system uses a CCD camera to capture the real-time image. As pedestrians are moving object the first step of this work is to segment the motion areas from the captured image frames. From the segmented and filtered image, contour or boundary is detected using propose scan conversion method and normalized. The normalized boundary or contour is compared with predefined contour using proposed point-fitting algorithm for matching. The system captures 30 image frames per second. Since human eye cannot observe changes made within 0.1 second, this system process 10 frames per second, whose difference are not significant for human observer.

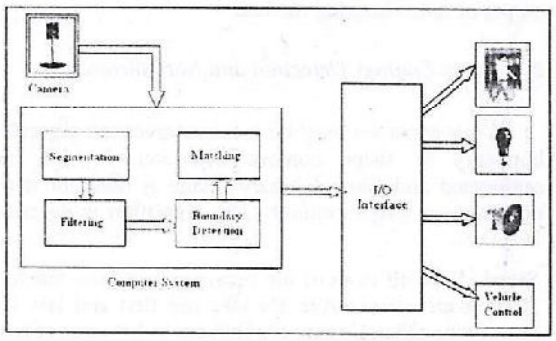


Fig. 1 : Proposed pedestrian detection system

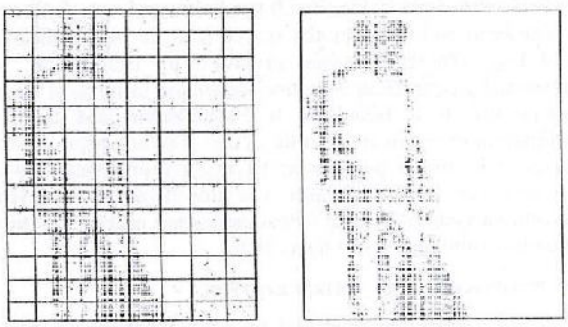
A. Motion Area Segmentation and Filtering

This system uses subtraction of two consecutive 24-bit color image frames for detecting motion areas since this system considers both pedestrian and camera are moving. Equation 1 is used to subtract two consecutive color image frames,

$$D(M \times N) = F_t(M \times N) - F_{t-1}(M \times N) \qquad (1)$$

Where $F_t(MXN)$ is the current image frame represent using RGB (Red, Green, Blue) color coordinate system and $F_{t-1}(MXN)$ is the immediate previous image frame in the same color coordinate system. Subtracted image is converted as binary image base on motion intensity of every pixel. If the color intensity difference of a particular pixel on current image frame and the previous image frame is greater than the predefined threshold then we consider that the pixel is moving significantly and mark the point using 1 else we mark it by 0. The motion areas are converted as binary image using equation 2.

$$V_t B(M \times N) = \begin{cases} 1 & when \ (R_t + G_t + B_t) \geq Th \\ 0 & Otherwise \end{cases} \qquad (2)$$



(a) Segmentation output with noise (b) Output after noise filtering

Fig. 2: Sample input and output of filtering method

Figure 2(a) shows the sample output of segmentation method where moving areas are shown using Red color. Sometimes there may be few noises or small blocks in the segmented images. To eliminate the noises we divided the whole image into several small blocks (such as 30X30 size blocks). If total number of 1 in each block is less than a predefined threshold value then the system considers the block as a noise and simply eliminates that block that mean replaces with 0's. Figure 2 shows the sample input and output of noise filtering method.

B. Shape Contour Detection and Normalization

In this paper we propose a scan conversion algorithm for boundary or shape contour detection. In this method segmented and filtered binary image is used for detecting boundary or shape contour. The algorithm is described as bellow:

Step1: Scan all rows of the binary image from left to right. If there are consecutive 1's take the first and last 1's and convert the other 1's into 0's that are in between of first and last 1 and save the result in a temporary matrix H_s.

Step2: Scan all columns from top to button or bottom to top. In case of consecutive 1's takes the first and last 1's and convert the other 1's into 0's that are in between of first and last 1 and save the result in a temporary matrix V_s.

Step 3: Calculate the union of matrix H_s and V_s, $H_s \cup V_s$

Figure 3 shows the example input and output of shape contour or boundary detection method.

(a) Input of contour detection Method

(b) Output of contour detected method

Fig. 3: Example input and output of contour detection

Boundary images are may be various sizes like (M_1xN_1), (M_2xN_2), (M_3xN_3), etc. Normalization method is scales all boundary images into the size of predefined template images. In this paper, training and test images are scaled to 180x200 resolutions. The boundary image B[(0,0), (MxN)] is scaled to image N[(0,0), (180,200)] using the equation 3

$$N(x_i, y_i) = B(x_i \times S_x, y_i \times S_y) \qquad (3)$$

Where, $S_x = \frac{180}{M}$ and $S_y = \frac{200}{N}$. The

C. Shape Contour Matching

After normalizing, the contour image is compared with the predefined shape contour template images. Both the

training and test contour images are prepared and stored in a similar way. This paper proposes point-fitting algorithm to compare test contour with predefined shape contour templates. The detail of our matching algorithm is as follows:

Step 1: Read the test contour image [Tst(x, y)], and record the co-ordinates that hold 1 in the set S,

$$S = \{(p,q),(r,s),(u,v),....,(m,n)\} \quad (4)$$

Where, *(p,q)*, *(r,s)*, *(u,v)* and *(m,n)* represents the location of binary pixels whose value are equal to 1.

Step 2: Read i^{th} shape contour template image [Tmp_i(x, y)] and record the coordinates that hold 1 in the set R.

$$R = \{(p^1,q^1),(r^1,s^1),(u^1,v^1),....,(m^1,n^1)\} \quad (5)$$

Where, $(p1,q^1)$, (r^1,s^1), (u^1,v^1) and (m^1,n^1) represents the location of binary pixels whose value are equal to 1.

Step 3: Compare test contour points with shape contour template points and calculate Hit (H_i) and Miss (M_i) scores.

a) Hit Score: Calculate how many contour points are matches (both the value of Tst(x,y) and Tmp_i(x,y) is equal to 1), that represents the Hit Score. The Hit score (Hi) is equivalent to the cardinality (total number of elements in a set) of H set ($H_i = \overline{\overline{H}}$), where,

$$H = \{S \cap R\} \quad (6)$$

b) Miss Score: Calculate how many contour points do not match (either the value of Tst(x,y)) or Tmp(x,y) is equal to 1 not both) that represents the Miss Score. The Miss score (Mi) is equivalent to the cardinality of M set ($M_i = \overline{\overline{M}}$), where,

$$M = \{\{S \cup R\} - \{S \cap R\}\} \quad (7)$$

Step 4: Calculate the Hit rate.

$$\text{Hit rate, } HR_i = \frac{H_i}{H_i + M_i} \quad (8)$$

Where, H_i = Total Hit-scores for i^{th} template and M_i = Total Miss-scores for i^{th} template. Go to step 2 and do until finish to read all shape contour templates.

Step 5: If Maximum(HR_1, HR_2, HR_3,..... HR_i) is greater than a predefined threshold then Pedestrian is localized and bounded using a rectangular color box.

In case of exact matching the value of HR_i is 1. In this system the threshold is defined through experiment. We defined the threshold value for this system is 0.85. If we reduce the threshold precision and accuracy will be decreased.

III. Experiments, Results and Discussions

The camera captures 30 image frames per second and store as 24-bit color BMP image using RGB color coordinate system. The proposed method is implemented using Microsoft Visual C (6.0) and OpenGL programming language. To process each frames the system takes (capturing, segmentation, filtering, shape contour detection, normalization and matching) approximately 60 milliseconds. To train the pedestrian detection system we collect several known template images and then prepare shape/contour template of human pedestrian. The system takes 400 images of different pedestrian in different poses and in different scenarios for preparing template image. Figure 4 shows the example shape contour template images that are collected in various pose of pedestrian.

We have tested the system in different scenarios. Figure 5 shows several output images of pedestrian detection system. The system uses 10 Video Clips with different durations to analyze the performance of the proposed system and summarized the result in Table 1. In the Table 1, symbol D_S represent duration of a video file in second and T_F represents total number of process frames, P represents total number of pedestrian present in a specific video, C represents total number of correctly detected pedestrians, F represents total number of falsely detected pedestrians, A_p and P_p represent accuracy and precision of the proposed pedestrian detection system. The Accuracy (A_p) of this system is calculated by the ratio of total number of correctly detected pedestrian (C) and total number of pedestrian (P) as presented in equation 9. The Precision (P_p) of this system is calculated by the ratio of total number of correctly detected pedestrian (C) and summation of number of correctly detected pedestrian (C) and number of falsely detected pedestrians (F) as shown in equation 10. Figure 6 shows the performance of pedestrian detection in terms of accuracy and precision.

$$A_p = \frac{C}{P} \times 100 \quad (9)$$

$$P_p = \frac{C}{C + F} \times 100 \quad (10)$$

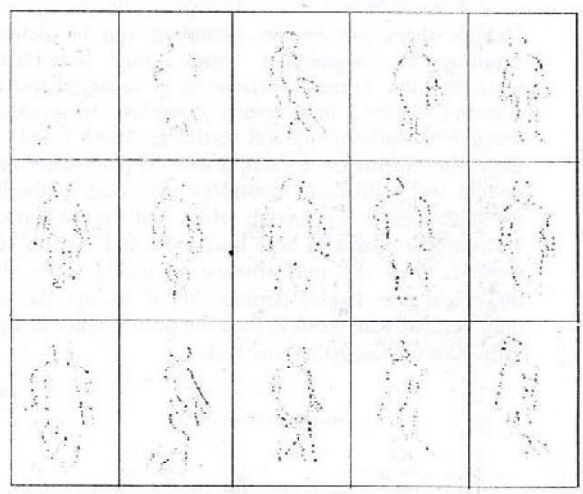


Fig. 4: Examples of shape contour templates

Table 1: Performance Analysis of Pedestrian Detection Method

Video #	D_S	T_F	P	C	F	A_p (%)	P_p (%)
#1	10	100	105	95	5	90.40	95.00
#2	18	180	40	38	0	95.00	100
#3	36	360	300	280	9	93.33	96.88

#4	48	480	411	378	25	91.97	93.79
#5	54	540	509	488	10	95.87	97.99
#6	57	570	397	370	12	93.2	96.85
#7	60	600	539	520	19	96.47	96.47
#8	63	630	557	514	27	92.28	95.00
#9	72	720	500	556	39	91.20	92.12
#10	78	780	446	405	30	90.80	93.10

Average accuracy of this system is 93.05% and average precision is 95.72%. Accuracy for the best case is 96.47 % and worst case is 90.8 %. Precision for the best case is 100 % and for the worst case is 92.12 %. The accuracy and precision rate varied because of variation in lighting conditions, backgrounds, etc.

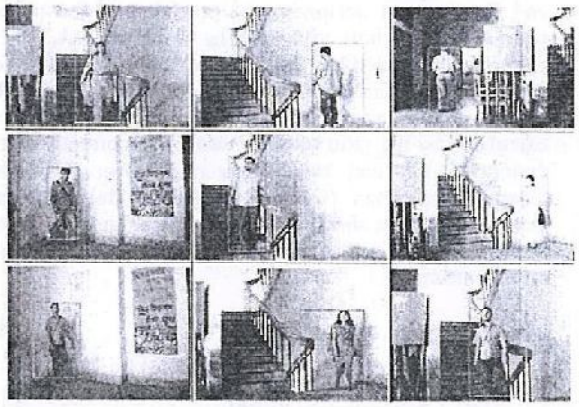
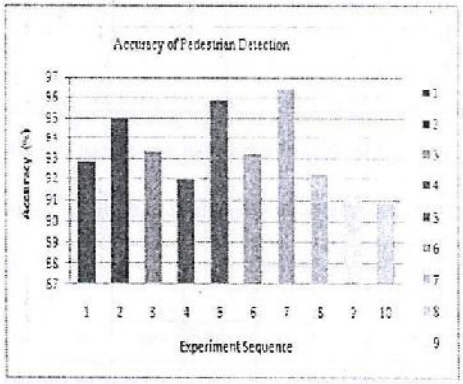
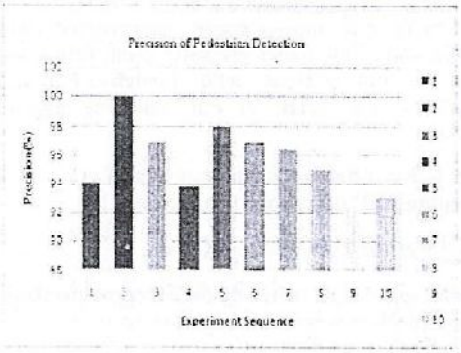


Fig. 5: Sample results of pedestrian detection method

Though shape contour or boundary can be detected by scanning the segmented image only horizontally or vertically, this system proposes a boundary/shape contour detection method that detects boundary by scanning the image both horizontally and vertically. Table 2 and Figure 7 show the performance comparison of pedestrian detection system using different boundary detection methods for 7 video sequences. Following tables and Figure show that if we combine results of both horizontal and vertical scanning methods then the performance is much better than the individual one. It also depicts that if we use the result of only vertical scan method then the performance is very poor even than the horizontal method.



(a) Accuracy for different videos



(b) Precision for different videos

Fig. 6: Performance of pedestrian detection system

Table II: Performance Comparison of Pedestrian Detection System Considering Different Contour Detection Methods

Video Sequences	Accuracy of Pedestrian Detection using Different Contour Detection Methods		
	Horizontal scan	*Vertical scan*	*Both Horizontal and Vertical scan*
#1	80.25	60.12	92.40
#2	87.16	64.25	95.50
#3	86.45	61.31	93.33
#4	78.20	58.65	96.97
#5	87.60	68.35	90.87
#6	79.50	59.31	94.20
#7	81.77	65.42	91.47

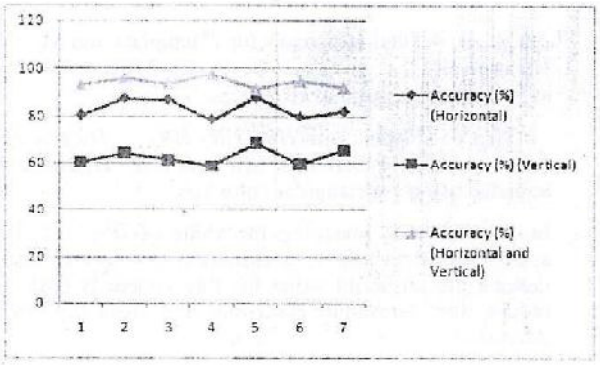


Fig. 7: Performance comparison of pedestrian detection system for different contour detection methods

IV. Conclussion

This paper presents a real-time pedestrian detection system by shape contour/boundary matching. The system proposes a novel scan conversion method for boundary or shape contour detection from segmented motion areas. The

proposed system runs in real-time with satisfactory success rates. Average accuracy and precision of the pedestrian detection system is 93.05 % and 95.72 % respectively. It takes only 60 milliseconds in a 180X200 pixel image on a 1.8 GHz core2 Duo Processor.

One of the limitations of this system is that it can not detect multiple pedestrians. There are some situations such as when a human or pedestrian is just backside of another pedestrian this system can not detect the two pedestrians separately. If the camera can't capture the whole body or shape of the human then this system may fail to detect that the moving object is human or not. Our future work is to make this system more robust to detect multiple pedestrians and tracking pointing pedestrian.

References

[1] M. Enzweiler, D.M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments", IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), Vol. 31(12), pp. 2179-2195, 2001

[2] V. Philomin and D. Gavrila, "Real Time Object Detection for Smart Vehicles", In the proceeding of IEEE international Conference on Computer Vision, pp. 87-93, 1999.

[3] C. Papageorgiou, M. Oren, and T. Poggio, "A General Frame Work for Object Detection" In the Proceeding of International Conference on Computer Vision, pp. 555-562, 1998.

[4] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection", Int. Journal of Computer Vision (IJCV), Vol. 38(1), pp. 15-33, 2000.

[5] A. Mohan, C. Papageorgiou and T. Poggio, "Example-based Object Detection in Images by Components", IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), Vol. 23(4), pp. 349-361, 2001.

[6] Paul Viola, Michael J. Jones, and Daniel Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance", Int. Journal of Computer Vision (IJCV), Vol. 63(2), pp. 153-161, 2005.

[7] S. Milch, "Pedestrian Detection with Radar and Computer Vision", Dissertation Universität Darmstadt: Fachgebiet Lichttechnik, Utz-Verlag, München, 2001.

[8] M. Yasuno, S. Ryousuke, N. Yasuda and M. Aoki, "Pedestrian Detection and Tracking in Far Infrared Images", In the proceeding of 8th International Conference on Intelligence Transportation Systems, pp. 131-136, 2005.

[9] M. Bertozzi, A. Broggi, A. Lasagni, M.D. Rose, "Infrared Stereo Vision-Based Pedestrian Detection", in the proceeding of IEEE Int. conference on Computer Vision and Pattern Recognition, pp. 23-28, 2005.