A Symmetry Search and Filtering Algorithm for Vision Based Pedestrian Detection System

Guanglin Ma and Anton Kummert
University of Wuppertal

Su-Birm Park and Stefan Müller-Schneiders
Delphi Electronics & Safety

Alexander Loffe
Delphi Deutschland GmbH
A Symmetry Search and Filtering Algorithm for Vision Based Pedestrian Detection System

Guanglin Ma and Anton Kummert
University of Wuppertal

Su-Birm Park and Stefan Müller-Schneiders
Delphi Electronics & Safety

I. INTRODUCTION

According to the report of Federal office for statistics Germany, the rate of the crash-related injuries and fatalities are reduced dramatically during the last 30 years by adapting more and more passive safety systems on the car. In spite of such an impressive improvement, the social annual costs are still very high in EU-15. The introduction of preventive safety systems can contribute to further avoidance or mitigation of accidents.

In general there are two big branches of pedestrian protection technologies. One is collision mitigation zone technology like front airbags, battery disconnect system and occupant sensing systems (Figure 1). Another one is the collision avoidance zone technology like pedestrian detection, lane departure warning and traffic sign recognition system (Figure 2).
environment and possible collision with other obstacles around the driving environment.

The vision based pedestrian detection on a moving vehicle is one of the most challenging but also interesting and active research topics in the field of active safety. The challenges of the pedestrian detection on a moving vehicle are: movement of the sensor, unpredictable road situations and background scenarios, variations in the size, appearance and pose of the pedestrians.

In the past years, there were many different approaches have been done for pedestrian detection using different kind of sensors or by fusing of different sensors. The type of sensors can be classified to monochrome camera, color camera, far and near infrared camera and etc. According to the number of the cameras in use, the vision system can be classified to single, stereo or even Tetra-vision system [1]. Although the more cameras (both in type and number) we have, the more information we can acquire, but single camera approach is preferred by car manufacturers because of its lower cost in serial production.

In this paper we present a single monochrome camera based pedestrian candidate detection system, by mainly focus on the position refinement and filtering of detection pedestrian candidate through a real-time symmetry search and filtering algorithm.

Several different symmetry search approaches have been presented in the literature [4], [5], [6]. The algorithm presented in this paper is the extension of the idea of [5] and [6].

The remainder of this paper is structured as follows: The next section gives a brief introduction of our vision based pedestrian detection system and the global structure of the system. Section III presents the previous work of our pedestrian detection system and the necessity of the symmetry search algorithm in the system. The detailed symmetry search and filtering algorithm is described in section IV. Finally section V ends the paper with a brief conclusion of the proposed approaches. The results shown in this method are fast, cost effective and well suited for real-time vision applications.

II. SYSTEM DESCRIPTION

The strategy of our vision system is to develop multiple vision applications which can run on a single camera either simultaneously or stand alone depends on the requirement. These vision applications are Lane departure warning, advanced high beam control, Pedestrian detection and classification, Traffic sign recognition and Night vision (extra near infra-red lights and display needed). Figure 3 shows an example of our vision based pedestrian detection system, which draw a yellow bounding box around the detected pedestrian and displayed on the screen.

![Pedestrian detection system](image1)

Figure 3 Pedestrian detection system

Many vision based pedestrian detection algorithms are using 2 step detection strategy to do pedestrian detection. First, the potential pedestrian candidates are detected and afterward, the detected candidates are fed into the classifier to do final verification. We also use 2 step detection strategy to do our pedestrian detection [2], [3]. Left image in figure 4 shows the detected pedestrian candidates (green bounding boxes), and right image in figure 4 shows the final pedestrian which is classified by our classifier.

![Two step pedestrian detection](image2)

Figure 4 Two step pedestrian detection

Figure 5 shows the detailed system structure of our pedestrian detection system. We get the image frame together with the velocity and yaw rate of the host vehicle as the input and potential pedestrian candidates are detected by the detection and segmentation algorithm. Several features are extracted on each pedestrian candidate and the classifier classifies on these features. In between we have low level and high level trackers to track the pedestrian candidates and classification results. And the classification results are feed into the detection of next frame.
The algorithm presented in this paper is a subset of the pedestrian candidate detection part which consists of detection and segmentation sub blocks.

III. PREVIOUS WORK

DETECTION - The detection of pedestrian candidates is done by utilizing the Inverse perspective mapping (IPM) algorithm. Figure 6 shows the idea of inverse perspective mapping based obstacle detection algorithm [2], [3].

In order to do obstacle detection, we first project two consecutive images to the ground plane. We will get trapezoid shaped transformed image on the ground plane. Using the velocity and yaw rate of the host vehicle, we transform the second image to the time instance of the first image. The idea behind here is that if the image pixels which we acquired are from the ground plane, there will be a linear transformation between two corresponding pixels in two consecutive image frames and these two pixels will have almost the same pixel intensity. If not, they are projection from an object above the ground plane. By subtracting two transformed images on the ground plane and threshold on them, we will get detection on objects above the ground plane.

1D PROFILE- For obstacles from a low contrast images or a very small motion (both from the host vehicle & obstacle itself), there are much smaller amount of detected points compare with normal situations. The detected points are distributed irregularly along the edge and there is no clear cue to distinguish the correct detections from both of the irregularly distributed sensor noise and the detections from small patterned areas like bushes etc. A vertical profile of the binary detection result is created in order to distinguish a desired detection of a pedestrian with other unwanted detections (noise, detection on bushes etc). The idea is based on the fact that pedestrians have highly vertically oriented edges compare with their background. By integrating the detection result vertically, the detection on vertically oriented objects is amplified. Figure 8 illustrates the binary detection (green points) result and their vertical profile (cyan graphs). In his way the detected signal is emphasized and obstacles under bad illumination conditions or under bad weather conditions like rainy, foggy, or snowy are also detected.

REGION OF INTEREST- One of the most stringent requirement of the online vision system is the efficiency on the calculation time. In order to further decrease the
calculation time, we introduced a novel idea termed “pedestrian detection strip” (PDS) which can decrease the calculation time by factor of 6.

Because of the geometrical property, the IPM based detection only works on the region under the horizon, and if we also exclude the region of the car hood, then our detection region will be around 200 pixels in height in the image plane.

By further evaluating the detection results, we found a common factor that the pedestrian always cross the horizon. Based on this observation, we restrict our detection region only on a small strip around the horizon. The height of the strip is 30 pixels and it is selected for a reliable detection of a pedestrian in 50 meters away from the car.

Figure 9 is a side view of our pedestrian detection strip. The red strip is our PDS and the green triangle area is the area where our detection can not be covered.

Figure 9 Side view of Pedestrian Detection Strip

BOUNDING BOX GENERATION – The detection result which we get from PDS is only the horizontal location of the detected objects on a 2D image plane. There is no information on the depth or the size of the detected object available. A foot point is for each detected obstacles in order to get the pedestrian sized bounding box. The searching of the foot point is done in the following way:

1) Vertical edges are detected for each detected point from Figure 10(a) using Sobel edge detector and the edge is binarized. (Figure 10(b))
2) A modified vertical Hough transform is performed on the binary edge image to eliminate small vertical edges which are mainly caused by non-pedestrians. (Figure 10©)
3) The lowest end point of the vertical Hough transform is selected as the foot point of the pedestrian.
4) From each detected foot point, a 1.8meter x 0.9meter bounding box is calculated out and this bounding box is the detected result of our pedestrian candidate detection algorithm. (Figure 10(d))

The detected pedestrian candidates are fed into the classifier and the classifier decides whether the detected candidate is a real pedestrian or not.

Figure 10 Foot point search

Because the initial detection is done on a small horizontal strip, not only pedestrians, but some other obstacles are also detected as pedestrian candidates. Further more, the bounding boxes of the initial detections are generated by the foot point search on the edge image. Because of these reason, the bounding boxes of the initial detection results may be not aligned to the center of the pedestrians well. Another problem of the initial detection is two or more pedestrians walking in parallel may included in a single bounding box. In the contrast, the next stage - the classification step requires high accuracy on the size and position of the detected bounding box to deliver correct classification result.

A symmetry search approach is done on the initial detection result to make the detection result more correctly aligned to the center of the pedestrian and also split a single bounding box around 2 or more pedestrians into several single bounding boxes.

IV. SYMMETRY SEARCH

For each bounding box of the initial detection the searching of the symmetry is done in the following steps:

1) Define the ROI of the symmetry search.
2) Symmetry search in the defined ROI.
3) Verification of the searched symmetry.
4) Split or filter the detected bounding box.

SEARCH REGION – In our algorithm, instead of performing the symmetry search on the whole pedestrian area, we narrowed down the scope for the symmetry search region to the lower leg part. This is based on the observation that normally the background of the lower leg parts is homogeneously distributed ground plane (Figure 9). In the contrast, the background
of upper body parts can be very complicated and can affect the symmetry search a lot. Figure 11 and Figure 12 illustrates the idea. Figure 11 is the original image of an urban traffic scene and Figure 12 is the edge image of Figure 11. In the Figure 12, the green boxes are our ROIs for symmetry search where only small edges of the ground plane are included in the ROI and consequently the correct symmetry of the pedestrian can be found easily. The red box is an example of the conventional pedestrian sized symmetry search box, which contains complicated background inside of it and the symmetry search may be wrong because of the symmetry pattern in the complicated back ground as shown in the figure.

The height of the symmetry search region is 1/3 height of the detected bounding box, where only the lower leg and feet of the pedestrian will be covered inside of the bounding box. The horizontal search region is expended both to left and right side with 1/3 width size of the detected bounding box. The symmetry search is done inside of the defined search region by shifting a 1/3 pedestrian height box (green box in the Figure 12) from left to right pixel wise.

- **Symmetry density** – By comparing the edge points in the left and right side of the symmetry axis (red vertical line in Figure 13), the symmetry of each row is calculated and consequently the symmetry density of the box is also calculated by accumulating the symmetry density of each row.
- **Symmetry distance** – The symmetry distance is also calculated row wise first and afterward the symmetry distance of a search box is calculated by averaging the symmetry distance of each row inside of the search box. The symmetry distance of a row is the mean distance of each symmetry point in the row to the symmetry axis.
- **Symmetry slope** – From the row wise symmetry distance, the symmetry slope angle and direction of a search box is calculated out. The slope is used for the verification and filtering of the symmetry search result which is going to be presented in the following subsection.
- **Edge Density** – The edge density of each sub block is calculated out. The edge density is simply calculated out by counting the binary edge points in each sub block. The sub block edge densities are also used for the verification and filtering.

![Figure 11 Original image of an urban traffic scene](image1)

**Figure 11 Original image of an urban traffic scene**

![Figure 12 Symmetry search on leg](image2)

**Figure 12 Symmetry search on leg**

**SYMMETRY SEARCH** – The searching of the symmetry is done on the vertical edge image. First, the above mentioned green search box is divided into 8 sub-blocks (2 horizontally, 4 vertically) as illustrated in Figure 13. In the figure the red vertical line is the symmetry axis of the bounding box. Several different features of the symmetry are calculated out:

![Figure 13 Symmetry search sub-blocks](image3)

**Figure 13 Symmetry search sub-blocks**

The initial symmetry search axis is searched by only look at the symmetry density histogram. Figure 14 shows a example of initial symmetry search. Figure 14(a) shows the original detected bounding box which is not well aligned to the center of the pedestrian. The above mentioned symmetry features are calculated out on the ROI of the symmetry search first. Figure 14(b) shows the histogram of the calculated symmetry density on the ROI. Within the histogram, local peaks with threshold are searched. As there is only one big local peak on the pedestrian in the right side, the peak position of the symmetry density histogram is selected as the initial symmetry search result. Figure 14(d) shows the refined bounding box which is aligned to the searched symmetry density peak. Figure 14(c) shows the symmetry search sub-blocks on the position of the searched symmetry peak. These sub-blocks are going to be used for the verification of the initial symmetry search result.
Figure 14 Initial symmetry search result: (a) original detection bounding box, (b) symmetry density histogram on the edge image, (c) symmetry search sub-blocks over the peak of the symmetry histogram, (d) initial symmetry search result.

DETECTION SPLIT – For the cases where two or more pedestrians walk in parallel, the detected bounding box sometimes located between two pedestrian as shown in Figure 15(a). The histogram of the symmetry density looks like Figure 15(b): three local peaks appeared in the histogram. One local peak on the left side pedestrian, another local peak on the right side pedestrian and the third local peak in between. As the amplitude of these three local peaks are almost the same. In this case, on each local peak a new bounding box is drawn as illustrated in Figure 15(c). Please note that in order to illustrate three different bounding boxes clearly, three different colors are used to draw the bounding boxes in the Figure 15(c).

Figure 15 Symmetry split

SYMOMETRY VERIFICATION – As shown in the previous examples, for the case where only a single symmetry density peak appeared on the histogram, we can find the symmetry axis easily and correctly. But there are also some cases where local symmetry density peaks deliver wrong symmetry result. A symmetry verification process is necessary in these cases.

Figure 15 shows a typical situation where a symmetry verification is necessary. As explained previously, there are three peaks are appeared in the search ROI and in this case on each peak a bounding box is drawn as shown in Figure 15(c). Among these three bounding boxes, two bounding boxes are drawn correctly on pedestrians, but the third one is drawn the symmetry peak which is caused by the two pedestrians walking in symmetry.

Figure 15 Symmetry split

A symmetry verification process is done on each found symmetry local peak to verify whether there are pedestrian liked symmetry features presented on the found area. The verification is done in the following way. First, the symmetry search sub-blocks are overlapped on the found local peak (Figure 16(b)). Using the previously calculated symmetry features (slope, distance, distribution etc), the found symmetry is verified and if a symmetry is not from a pedestrian, then it is filtered out. (Figure 16(c)).

Following cases are checked and symmetry peaks from non pedestrian area are filter out:

- Group of pedestrians walk in parallel: If the symmetry density is very low in the center area of symmetry search result, they are mostly caused by pedestrians walk in parallel.
- Pedestrians faced to each other: If the shape of the symmetry is in inverse trapezoid form, then they are probably cause by two walking pedestrians face to each other.
- Non-pedestrian area case 1: If the edge density is very low in overall, they are probably from empty area.
- Non-pedestrian area case 2: If the edge density is distributed homogeneously, they are from non pedestrian area like ground plane.
V. CONCLUSION

We have presented a symmetry search and filtering algorithm for a single monochrome camera based pedestrian detection algorithm.

Our vision-based pedestrian detection is done in 2 sub-steps. First the pedestrian candidates are detected and afterward the detected pedestrian candidates are feed into the classifier to classify the detected pedestrian candidate.

The initial pedestrian candidates are detected within a small horizontal strip and a foot point of detected pedestrian candidates are search to create corresponding bounding box. As the foot point is searched on the edge image, sometimes the detected bounding box may not well aligned to the center of the pedestrian. For that reason a symmetry search algorithm is necessary for the refinement of the initial detection.

In our algorithm, we first narrow down the scope of the symmetry search to the lower leg part where the background is relatively simple ground plane. Afterward, the defined ROI is divided into 8 sub blocks and edge density and symmetry density is calculated in each sub block.

Local peaks of the symmetry density histogram are selected as the found symmetry axis and the split and verification process is performed.

The described algorithm has been integrated in our vision based pedestrian protection system.

REFERENCES