3-14

MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

A Rear Vehicle Location Algorithm for Lane Change Assist

Wei Liu[1], Chunyan Song[1], Pengyu Fu[2], Nan Wang[2] and Huai Yuan[2]
[1]Software Center, Northeastern University
[2]Advanced Automotive Electronic Technology Research Center, Neusoft Co.,Ltd., China
AAC, A2 building, Neusoft Park, Hun Nan Industrial Area, Shen yang, China
(lwei, Wangnan, yuanh)@ neusoft.com

Abstract

A monocular vision based location algorithm is presented to detect and track rear vehicles for lane change assist. The algorithm uses the shadow underneath the vehicle to extract regions of interest (ROI), and then locates vehicles with the combination of symmetry, contour and the shadow underneath a vehicle. Three symmetry criterions are proposed to compute the vehicle symmetry axis: contour symmetry, gray symmetry and S component symmetry in HSV color space. Experimental results have shown better accuracy and robustness in different road and environmental conditions.

1 General Instructions

In recent years, because of the increasing of vehicles, traffic problems have become more and more serious worldwide. Intelligent driver assistance is an area of active research, and vehicle detection and tracking based on vision information are particularly interesting for their low cost [1,2,3]. The vision information is often obtained with monocular camera or stereo camera. In this paper, a novel vehicle location algorithm is presented based on monocular camera.

In vehicle detection and tracking systems, determining the exact position of a vehicle in images, especially its bottom edges and width is very important for acquiring distance to vehicles, relative speed, lateral position and time. Many institutions and researchers have done a lot of research in this field. Betke et al. [4] utilized edge maps to detect cars. They proposed a search method looking for vehicles' left/right edge as well as top/bottom edge by calculating vertical and horizontal edge histograms in search regions. Broggi et al. [5] obtained the ROI based on road positions and Inverse Perspective Mapping, then calculated the vehicle symmetry axes in the ROI and looked for whether two corners exist corresponding to the bottom edge of the vehicle via pattern matching technology in the symmetry regions of edge images. Finally, searched the top edge by perspective constraints as well as size constraints. Bertozzi et al. [6] detected the four vehicle corners with templates matching. Huang et al. [7] located vehicles with the features such as the shadow underneath a vehicle, and vertical edges, etc. Kyo et al. [8] utilized an abstract edge segment grouping method to locate vehicles. Kate et al. [9] first determined ROI by the shadow underneath a vehicle and then located vehicles with texture and symmetry. Kim et al. [10] utilized the shadow underneath a vehicle and symmetry for vehicle location.

All the above algorithms use some features of vehicles, such as symmetry, shadows, geometrical features (vertical and horizontal edges), etc. However, there are some limitations for these algorithms. They are easily prone to influences of backgrounds and illuminations, which makes the calculation of symmetry axes and edges imprecise, and as a result, affects the locating accuracy. What's more, most of the above literature is dedicated to front-vehicle locations, and some of them are inapplicable for rear-vehicle location. For example, when determining the left/right edge of vehicles with vertical projection, the rear structure of the vehicle has a rectangular shape with four corners, yet the front structure is streamlined and without strong vertical edges, so we can't obtain the left/right edge simply via vertical projection. Hereby based on summing up the applicability of other algorithms, we propose a novel location algorithm, it can decrease the disadvantageous influences of illuminations, backgrounds and as a result, makes the location more accurate.

2 Vehicle location Based on Features

2.1 ROI extraction

Shadow is an important clue to search for candidate vehicles in the image plane. Tzomakas et al. [11] detected vehicles with the shadow underneath a vehicle. They gave a detailed description about how to detect the shadow as well as how to extract vehicles, namely, ROI extraction method. Some optimization is also given [12]. We adopt the same approach to get the initial ROI to provide a coarse region for precise vehicle location.

2.2 Symmetry axis detection based on symmetry constraints

Symmetry is the vehicle's significant characteristic, it has been often used for vehicle detection. Images of vehicles observed from rear or frontal views are in general symmetrical in the horizontal and vertical directions. Various kinds of symmetry are used in a lot of literature such as contour symmetry, grayscale symmetry, horizontal symmetry and vertical symmetry, etc. However, each of them has its own advantages and disadvantages. For example, the advantage of the binary contour symmetry is that the illumination variety has little influence on it, but the background, especially the symmetrical background, easily affects it, and when the vehicle is partly occluded, its binary contour symmetry will be destroyed. The advantage of the grayscale symmetry is that the various backgrounds hardly affect it, but it is prone to influences of illumination conditions. The HSV space is expressed by the hue (H), saturation (S) and value (V), which is suitable for the human vision characteristic. The S component in HSV space is related to the material property. Therefore, we use the three methods to calculate the vehicle's symmetry axes in ROI: binary contour symmetry, grayscale symmetry and the S component symmetry in HSV space. Then the final vehicle axis is synthesized by integrating the three symmetry axes.

We adopt the algorithm to detect the binary contour symmetry axis as described in [13].It's given as follows:

1) Apply the Sobel Operator on the source image and construct the binary contour image IMG,

2) Select two thresholds (W_min, W_max) to determine the range of possible widths of the vehicle.

3) Construct an accumulator array A [col] and initialize all its elements to zero.

4) Scan the IMG and perform the following steps,

FOR each row

FOR each pair of non-zero pixels (x_1, y_1) and (x_2, y_2) in the current row, Do

{

distance $= |x_1 - x_2|$;

axis $= (x_1 + x_2)/2$;

if (distance>W_min and distance< W_max)

A[axis] = A[axis]+1;

}

5) Find the maximum element in A[coI] , whose index col indicates the most promising candidate for the symmetry axis of the object.

The calculation of gray symmetry axis is the same as that of S component symmetry axis. It's shown as follows:

1) Extract the shadow underneath a vehicle; calculate the bottom edge of the shadow Y_{bc} as well as the left and right terminal X_l , X_r ;

2) Compute the width of the shadow: $W = X_r - X_l$;

3) Measure the symmetry axis $S(j)$ with the following formula:

$$S(j) = \sum_{Y_{bc}}^{Y_{bc}+H} \sum_{\Delta x=1}^{W/2} \sum_{j=X_l-\Delta k}^{X_r+\Delta k} \left| P(j+\Delta x, i) - P(j-\Delta x, i) \right| \quad (1)$$

$$j_{sym} = \arg\min_{j} S(j) \quad (2)$$

Where $H = 0.9W$ and $P(x, y)$ is the gray value or the S-component value. j_{sym} is the symmetry axis we want to get.

The calculation of symmetry axes is shown in Figure 1 and Figure 2.

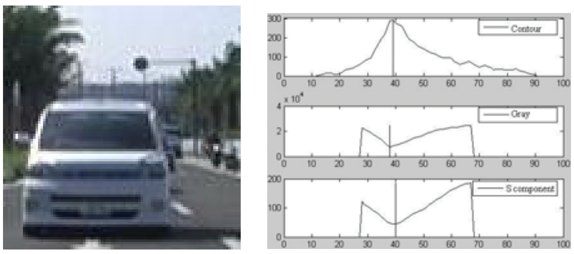

Figure.1. Computing the three kinds of symmetry axes

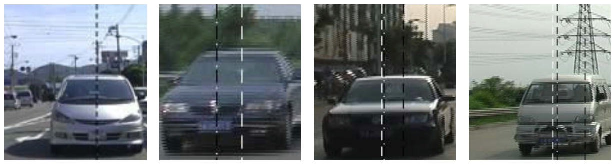

Figure.2 Examples for the symmetry axes. White dashed represents gray symmetry axis; Black dashed represents binary value symmetry axis; Dash-dot represents S component symmetry axis

As is shown in Figure 2,The calculating results can be classified into three cases: 1) the symmetry axes obtained with the three above methods completely(or approximately) overlap, 2)two of them completely(or approximately) overlap, 3) all of them never overlap with each other(It hardly takes place). In order to overcome the influence of illumination conditions and various backgrounds, we adopt the three methods to calculate the vehicle's symmetry axes and synthesize them so that the calculated axis is more accurate. The synthetic strategy is as follows:

1) Calculate the three symmetry axes x_1, x_2, x_3 ;

2) Calculate the synthetic symmetry axes x_{sym} .Three cases should be considered respectively:

Case 1:

If $\max\{x_i\} - \min\{x_j\} \le \Delta$, then

$x_{sym} = x_k$

Where $\min\{x_i\} \le x_k \le \max\{x_j\}$,

$i, j, k = 1,2,3$, $i \ne j$;

Case 2:

If $|x_i - x_j| \le \Delta$ and $\min\{|x_i - x_k|, |x_j - x_k|\} > \Delta$

Then $x_{sym} = \left\lceil \frac{x_i + x_j}{2} \right\rceil$, $i, j, k = 1,2,3$, $i \ne j \ne k$;

Case 3:

Otherwise, $x_{sym} = x_1$;

Where Δ is a threshold to evaluate the similarity of the symmetry axes and its size is related to the width of ROI.

2.3 The left and right edges computation

After finding the symmetry axis of vehicles, we extract vertical edges in ROI with Sobel Operator. Then filter based on symmetry constraints, and those pixels which are not symmetric about the computed axis are eliminated since they are probably noisy pixels. Now we get a much simpler symmetric region compared to the original contour image. Accounting for vehicle skewing in image, we search for the symmetric pixel p_b of pixel p_a in a small region around the overlapped mirror pixel p_a .When getting the vertical edge images after filtered, we compute the left and right edges. The general idea is to calculate the vertical histogram based on the computed symmetric region. Here we project respectively in the left and right regions around the symmetry axis. The point that corresponds to the highest peak in the histogram is regarded as the vertical projection of the side edge of the vehicle, and the other edge can be got via symmetry. So we can locate the left/right edge as well as the width of the vehicle

2.4 The top and bottom edge computation

A limited vertical stripe region is obtained after the calculation of the left and right edges of vehicles. The location of the top and bottom edges can be performed within this stripe. The steps are as follows:

1) Eliminate those horizontal edges whose widths are much wider than the width of the vehicle in the upper part of ROI since they are probably the edges of buildings or bridges.
2) Calculate the centroid of this vertical stripe region
Since the x-coordinate of the centroid is on the symmetry axis, we just need to calculate the y-coordinate as follows:

$$g_y = \frac{\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}(I_{ij} \times j)}{\sum_{j=0}^{n-1}\sum_{i=0}^{m-1} I_{ij}} \tag{3}$$

Here, I_{ij} is the pixels' values of the vertical stripe region; n, m are the height and width of this region respectively.

3) Compute the top/bottom edge of the vehicle

To locate the top/bottom edge, we perform the computation of horizontal histogram on the computed symmetric region. Below the centroid, search the significant horizontal edge from bottom to top in ROI, and the first horizontal edge whose histogram's value is larger than a specific threshold is regarded as the bottom edge of the vehicle. Both the threshold and the histogram are normalized by the width of the vehicle. The location of the top edge is similar to that of the bottom edge.

2.5 The edge adjustment based on the shadow underneath the vehicle

In practical applications, because of the complexity of backgrounds and illuminations, the above method still has some disadvantages. For example, our method may compute a pseudo-symmetry axis; and the left/right edge obtained by vertical projection may be affected by the vertical edges of backgrounds. Furthermore, due to the fact that the lower part of a vehicle is often darker than any other part in the same image, sometimes we couldn't find a significant bottom edge exceeding the specific threshold and as a result, we can't confirm the bottom edge. What's more, the location of the top edge is prone to influences of illumination conditions, especially when the vehicle has arrived at the image boundary, yet only a part of it is in the image, or it's occluded by other vehicles. In these situations, the vehicle will lose symmetry, which restricts the above method. In order to solve these problems, a method of edge adjustment is proposed based on the shadow. It's described as follows:

1) Bottom edge adjustment

If we can't find the bottom edge satisfying the threshold or the calculated bottom edge is quite different from Y_{bc}, replace it with Y_{bc};

2) Left and right edge adjustment

Sometimes the computed Left/right edge is false due to the influence of symmetric background. In this situation we can validate it with the computed shadow width and perspective constraints. If the shadow width is much smaller than the computed vehicle width and it can satisfy the perspective constraints , we can conclude that the calculated left or right edge is not precise, and replace it with X_l or X_r respectively.

3) Top edge modification

After adjusting the bottom edge as well as the left/right edge of the vehicle, we modify the top edge according to its aspect ratio. In general, the aspect ratio is between 0.6 and 1.4.

It's necessary to notice that when modifying the edge with the shadow, we should distinguish which is the real shadow underneath the vehicle and which is the shadow projected by bridges, trees, buildings, etc. Some literature has given a lot of detail about it, so we won't consider it here. Moreover, practically because of various reasons, it's very likely that the real width of the vehicle is wider than that of the shadow. So we should combine the distance information and perspective constraints to modify the left/right edge.

3 Implementation and Experimental Results

In order to evaluate the performance of the algorithm, different videos were taken from a backward-looking camera mounted on a vehicle, as well as on different scenes, including highway, urban road, sparse road, etc at different times of day. Sometimes roads are covered with japanning, smear, snow and so on.

Frames are manually extracted from video snippets as testing images. They are input into the recognition engine and the corresponding results are output. The recognition engine consists of ROI extraction, vehicle location and recognition. The extraction and location results are illustrated in the testing images, which are shown with black and white bounding-boxes respectively. Knowledge-based and appearance-based methods are combined for vehicle recognition, we don't discuss it here. The experimental results are shown in Table 1.

Table 1: Location accuracy

Metrics / Video	NRR	$RA1$	$RA2$
Urban road	1193	92.50%	90.83%
Urban narrow road	1125	94.77%	86.04%
High way	2294	93.72%	90.28%
mountain road	1113	89.13%	88.12%

The meaning of each metrics is as follows:

NRR (Number of Reference Regions) represents the reference number of vehicles appearing in the testing images; $RA1$ and $RA2$ are used to evaluate the accuracy of vehicle location

$$RA1 = \sum^{NRR} \frac{Area(TP)}{Area(RR)}, \quad RA2 = \sum^{NRR} \frac{Area(TP)}{Area(DRR)}$$

Where RR (Reference Region) represents ideal location regions of vehicles; DRR (Detection Result Region) represents the location regions obtained by the algorithm. TP represents the intersection of the above two. As is shown in Figure 3, the larger $RA1$ and $RA2$ are, the more precise the location is. Figure 4 shows some representative results under different conditions.

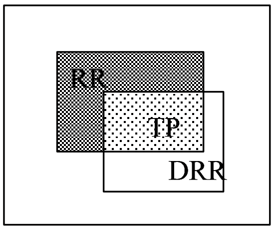

Figure 3 Sketch map of computing RA1 and RA2

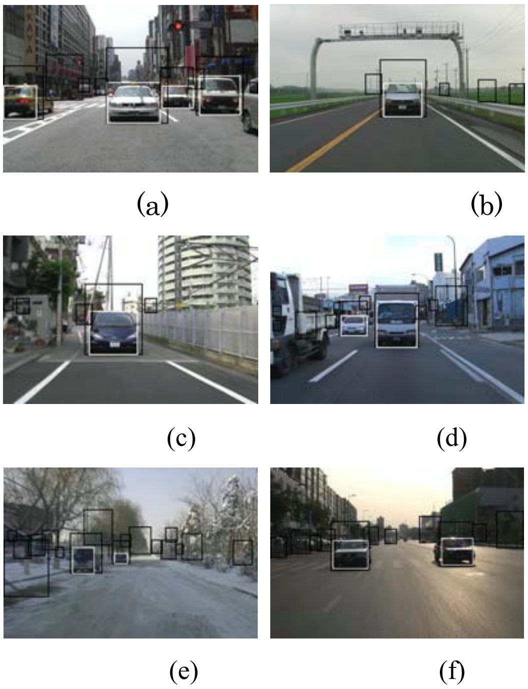

Figure 4 Vehicle location examples in different situations: (a) an urban scene with high traffic,(b) a highway scene with low traffic, (c) a urban narrow road scene, (d) different types of vehicles,(e) different weather condition (snowy) ,(f) bad lightening conditions.

4 Conclusion and future work

A new location algorithm is presented to locate rear vehicles based on monocular camera. It can locate vehicles in different road and environment conditions. The experimental result has shown higher accuracy, so it can completely satisfy the requirement of vehicle detection and tracking.

Currently, it takes about 1ms to deal with every ROI on the standard PC machine (Pentium IV 2.8GHz and 512M of RAM) without performing specific software optimizations. In future, we will do some optimization to satisfy the real-time requirement.

Acknowledgments

The authors would like to thank their colleagues for their help with the data collection.

Reference

[1] Z. Sun, G. Bebis and R. Miller: “On-road Vehicle Detection Using Optical Sensors a Review,”IEEE International Conference on Intelligent Transportation Systems, Washington, DC, October 3-6, pp. 585-590,2004.

[2] S. Han, E. Y. Ahn and N.Y. Kwak: “Detection of Multiple Vehicles in Image Sequences for Driving Assistance System,” Computational Science and Its Applications – ICCSA 2005: International Conference, Singapore, May 9-12, pp. 1122-1128, 2005.

[3] C. Hoffmann, T. Dang and C. Stiller: “Vehicle Detection Fusing 2D Visual Features,” IEEE Intelligent Vehicles Symposium, University of Parma ,Parma, Italy, June 14-17, pp. 280-285, 2004.

[4] M. Betke, E. Haritaoglu and Larry S. Davis: “Real-time Multiple Vehicle Detection and Tracking from a Moving Vehicle,”Machine Vision and Applications, vol. 12, no. 2, pp. 69-83, 2000.

[5] A. Broggi ,M. Bertozzi, A. Fascioli, C. G. Lo Bianco, and A. Piazzi: “Visual Perception of Obstacles and Vehicles for Platooning,” IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 3, pp. 164-176, 2000.

[6] M. Bertozzi, A. Broggi and S. Castelluccio: “A Real-time Oriented System for Vehicle Detection,” Journal of Systems Architecture, no. 43, pp. 317-325, 1997.

[7] S.S. Huang, C.-Jen Chen, P.Y. Hsiao and L.C. Fu: “On-board Vision System for Lane Recognition and Front-vehicle Detection to Enhance Driver's Awareness,” Proceedings of the 2004 IEEE International Conference on Robotics & Automation New Orieans, LA-April 2004.

[8] S. Kyo, T. Koga, et al: “A Robust Vehicle Detecting and Tracking System for Wet Weather Conditions Using the IMAP-VISION Image Processing Board,” IEEE International Conference on Intelligent Transportation Systems, pp. 423-428, 1999.

[9] Ten Kate.T.K.; van Leewen.M.B.; Moro-Ellenberger.S.E.; Driessen,B.J.F: “Mid-range and Distant Vehicle Detection with a Mobile Camera,” IEEE Intelligent Vehicles Symposium, University of Parma, Parma, Italy,June14-17, pp. 72-77,2004.

[10] S.Y. Kim, S.Y. Oh, J.K. Kang and Y.W. Ryu: “Front and Rear Vehicle Detection and Tracking in the Day and Night Times Using Vision and Sonar Sensor Fusion,” International Conference on Intelligent Robots and Systems, 02-06 Aug. pp. 2306-2311, 2005.

[11] C. Tzomakas and W. Seelen: “Vehicle Detection in Traffic using shadows ,” Institution of Neuroinformatik, Ruhr University Bochum in Germany, 1998.

[12] A. Khammari, F. Nashashibi, Y. Abramson and C. Laurgeau: “Vehicle Detection Combining Gradient Analysis and Adaboost Classification,” Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, September 13-16, pp. 1084-1089, 2005.

[13] Y. Du and N. Papanikolopoulos: “Real-Time Vehicle Following through a Novel Symmetry-Based Approach,” IEEE International Conference on Robotics & Automation. Albuquerque, New Mexico-April.pp.3160-3165, 1997.