A Wheel-based Side-view Car Detection using Snake Algorithm

Veerapathirapillai Vinoharan*, Amirthalingam Ramanan†, and Saluka R. Kodituwakku‡
*Computer Unit, University of Jaffna, Sri Lanka
†Department of Computer Science, University of Jaffna, Sri Lanka
‡Department of Statistics and Computer Science, University of Peradeniya, Sri Lanka
{vinoharan, a.ramanan}@jfn.ac.lk, salukak@pdn.ac.lk

Abstract— **Car detection is a challenging research area where the appearance has various changes due to different models, poses, scales, lighting conditions, background, occlusion and various image sizes. Amidst these external challenging factors, car side-views have obvious and consistent characteristics in their structure such as wheels, oblique windows and bumpers, which provide crucial cues for detection. In this paper, we describe a novel side-view car detection technique which constructs an initial contour for the Snake using the detection of wheels. Our approach is subdivided in to three stages: wheel detection, bounding box detection, and segmenting the car from its background. In the first step, we use Hough transformation to detect circles in an image and validate the wheels with a learnt visual vocabulary. In the second step, we construct an initial contour with the aid of the detected wheels and the width-to-height statistics of different types of cars. This initial contour forms the bounding box of the car. In the third step, the Snake algorithm proceeds with the initial contour to fit the boundary closer to the car. Hence, the foreground, car, is separated from the background. We empirically evaluated our method on 100 side-view cars and the testing results shows that this approach has good performance.**

Index Terms— **Car detection, Circular Hough Transform, Snake algorithm, SURF, Visual Vocabulary**

I. Introduction

Visual object detection is the task of finding a specific object in a given digital image or video sequence. Recently, most of the research works [4]–[6] on object detection cover the car detection problem. Car detection is a growing research area which has important applications, such as intelligent traffic surveillance, driver assistant and vehicle parking systems.

In this paper, we address side-view car detection and segmentation not as separate entities, but as two closely collaborating processes. We also address the detection problem in static images that can be gray-level or colour images and focus on local features that describe these structural characteristics in particular. There are few approaches that specially consider the structural characteristics of cars [1]–[3]. All side-view cars have two wheels and similar boundary shapes but little bit of variations. In our approach, bounding boxes of a car in its side-view is detected by means of wheel detection, set of line detection, and then the detected box is passed as the initial contour for the Snake algorithm [10], [11]. We use the Hough circle detection algorithm [9] to find the possible circular-like structures that are present in a given image. We create visual vocabularies using speeded up robust features (SURF) [8] to identify the two wheels and reject the rest of the circles that have been detected in the previous step. Then car statistics such as wheelbase distance, wheelbase-to-hood ratio, and oblique lines that makes up the front and rear windows are used in finding the bounding box of a car. This bounding box is passed as the initial contour for the Snake (or active contour) algorithm for finding the outline of the car by an iterative process. Using the final contour obtained by the snake method is used to perform the foreground-background segmentation in order to locate the car.

The rest of this paper is structured as follows. Section 2 summarises various car detection algorithms that have been carried out in the recent years. Section 3 briefly describes the needed background for our work. Section 4 explains in detail the proposed methodology in achieving detection and segmentation. Section 5 describes the experimental setup and testing results which supports our claim. Finally, section 6 concludes the paper.

II. Related Work

Many related side-view car detection research has been carried out in recent years. This section briefly describes them with their merits and demerits.

In [2], the authors proposed a vehicle detection system for cars with different direction perspectives: front, back, side, and oblique. For the front or back-view cars, the knowledge of horizontal structures are used to generate the hypothesis area, and for oblique car, a template matching technique is applied to generate the vehicle hypothesis. Using these hypothesis, line features are extracted from each sub-part of the designed template. The method used to detect side-view cars in [2] has been revisited in 2011 by the same authors [1]. They reported that side-view cars can be detected based on a template matching technique. The template is constructed using the shape knowledge of a car which mainly consists of wheel positions, horizontal and oblique lines of a car. They use Hough transformation to detect circles of an image and then the wheels are detected by using some assumptions such as the two wheels are of the same size, exist in the same horizontal line, and not in the top of the image. They estimate the approximate width and height of a car, where the height is about two-third and width is about three-fifth of the distance between two wheels. According to this assumption, the initial

978-1-4673-1975-1/12/$31.00 ©2012 IEEE

bounding box is generated. They use the Hough transform method to detect lines that roughly picks out relatively long line segments included in the edge image that makes a line extraction strategy based on a line fitting algorithm. The authors have tested their method on a set of images that has been collected on the World Wide Web. Their method shows an average success rate of 90.7% in detecting cars under the assumptions that wheels exist in the same horizontal line and not in top of the image. The method that we propose here is free from such sort of assumptions which make it more practical.

In [3], the authors proposed a car recognition method using multiple features of contour. The method starts with eliminating the background area from the image by using main colours such as RGB and HSV classification. Image contours are then extracted in the following three steps: (i) finding the region of interest in order to separate the image into foreground objects and background by using different colour model histograms, (ii) extracting the image edges by the usage of Canny operator, and (iii) obtaining single-edged contour by adopting image pyramid. The final contour obtained after the step (iii) is then described as features using Fourier descriptor, round rate, direction ratio, and the circumference ratio of the car wheel to car body. Finally, the similarity between the feature vectors are computed to a test image and the car class is predicted accordingly. Assessing the overall performance of the authors' method is difficult, since it depends on the feature vector as it is composed with the choice of multiple descriptors.

In [4], a novel set of image strip features was proposed for detecting cars. The proposed set of image strip features describes the appearances of common structural components such as wheels, pillars, bumpers, etc. Image strip features are described by back-to-back regions considered as a template of a curve segment with a certain strip pattern. The algorithm is based on integral image method that builds a full set of image strip features with different curve segments, strip patterns, and positions. These features represent various types of lines and arcs with edge-like and ridge-like strip patterns, which enrich the simple features such as Haar-like features and edgelet features. For detection of cars, the authors use a sliding window strategy and the mean-shift clustering algorithm to merge the positive responses of the classifier and obtain the bounding boxes of the objects. The proposed method is evaluated on the UIUC single-scale car dataset.

In [5], the authors have proposed a method for learning the appearance and spatial structure of a visual object category in order to recognise objects of that category, localise them in cluttered scenes, and automatically segment them from the background. The sub part of their approach is a learned representation of object shape that can combine the information observed on different training examples in a probabilistic extension of the generalised Hough transform. They extract local features around interest points and grouping them with an agglomerative clustering scheme to constructing a codebook of local appearances that are characteristic for its member objects. Based on this codebook, they learn an implicit shape model that specifies where on the object the codebook entries may occur. They formulated the multi-scale object detection problem from which hypotheses are found by a scale-adaptive mean-shift search. By back projecting the contributing votes, the authors retrieve the hypothesis's support in the image, which shows that the system's reaction. They integrate learned knowledge of the recognised category with the supporting information in the image to derive a probabilistic formulation for the top-down segmentation problem. The result supplies a pixel-wise figure-ground segmentation and extension of recognition. The authors have evaluated their method on the UIUC single-scale and multi-scale car datasets.

III. Background

A. Hough Transform for Circle Detection

The Circular Hough Transform (CHT) [9] finds circular formations, of a given radius *R*, within an image. CHT is almost identical to the Hough transform for lines, but uses the parametric form for a circle: $x = x_0 + R\cos\alpha$ and $y = y_0 + R\sin\alpha$, where the locus of (x_0, y_0) points in the parameter space fall on a circle of radius *R* centred at (x, y).

The operation of CHT can be summarised in the following steps: Edges are detected in an image and then each edge point is taken as a centre of a circle of radius *R* drawn onto an accumulator array which is raised with one. The values in this accumulator are compared to a preset threshold those values higher than the threshold are the centres of fixed radius circles. That is, many constructed circles intersect leading to a large intensity peak in the accumulator array at, or near, the centre of the circle. The CHT can be formulated as a convolution applied to an edge magnitude image whose binary mask coefficients are set on the circle boundary and are zero elsewhere.

B. Speeded Up Robust Feature

Speeded up robust feature (SURF) [8] is partly inspired by scale invariant feature transform (SIFT) [7] that makes use of integral images. The scale space is analysed by upscaling the integral image-based filter sizes in combination with a fast Hessian matrix-based approach. The detection of interest points is selected by relying on the determinant of the Hessian matrix where the determinant is maximum. The Hessian matrix is roughly approximated using a set of box-type filters and no smoothing is applied when going from one scale to the next. Image convolutions with these box filters can be computed rapidly by using integral images independently of their size. Next, the descriptors are computed based on orientation using 2D Haar wavelet responses calculated in a 4×4 sub region around each interest point, resulting in a 32 dimensional vector. When information about the polarity of the intensity changes is considered, this in turn results in a 64 dimensional vector. The extended version of SURF, e-SURF, has the same dimension as SIFT which is 128. SURF features can be extracted faster than SIFT using the gain of integral images and yield a lower dimensional feature descriptor resulting in faster matching and less storage space. Furthermore, SURF shows slightly better performance to SIFT in illumination changes, but performs less when the rotation is large.

C. Snake Algorithm

Snake method is an iterative process used in image segmentation tasks, particularly for objects whose edges are not well-defined. A snake or an active contour is a set of points which aims to enclose a target feature, the feature to be extracted. An initial contour is placed outside the target feature, and is then evolved so as to enclose it. In other words, snakes are energy-minimising splines guided by external forces and influenced by image forces that pull it in the direction of distinct features such as edges and lines.

The energy function for a snake is usually represented by a vector, $v(s) = (x(s), y(s))$ having arc length, s, as parameter:

$$E_{Snake} = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))\, ds$$

where E_{int} represents the internal energy of the snake due to bending or discontinuities, E_{image} is the image force, and E_{con} is the external constraint.

The iterative process of the snake algorithm starts with a set of initial points. For each point, its m×m neighbourhood of pixels is considered and the total snake energies for each of the neighbourhood points are considered. The position in the neighbourhood with minimum total snake energy is considered as the new optimum position, and the current point is moved to this position. This iterative process repeats for all the points of the snake until a stopping condition is met. A typical stopping condition is that zero snake points have been moved in a single iteration.

The internal image energy E_{int} is defined to be a weighted summation of first and second -order derivatives around the contour.

$$E_{int} = \alpha(s)\left|\frac{dv(s)}{ds}\right|^2 + \beta(s)\left|\frac{d^2v(s)}{ds^2}\right|^2$$

where, $dv(s)/ds$ measures the *elastic energy* and $d^2v(s)/ds^2$, measures the *curvature energy*.

Choice of the values of α and β control the shape of the snake method aims to attain. Low values for α imply the points can change in spacing greatly, whereas higher values imply that the snake aims to attain evenly spaced contour points. Low values for β imply that curvature is not minimised and the contour can form corners in its perimeter whereas high values predispose the snake to smooth contours.

IV. Methodology

Our approach considers side-view car detection and foreground-background segmentation as two interleaved processes that closely collaborate towards a common goal. In Figure 1, subparts (a) to (f) indicate the detection part and subparts (g) to (h) indicate the segmentation part.

A. Vocabulary construction

In the first stage, we use a small number of side-view car images to construct a visual vocabulary that will help in separating the wheels of a car from the circles detecting by CHT in its background. SURF descriptors are used in constructing a vocabulary for wheels and for the background. We extracted e-SURF interest points from manually segmented bounding boxes that consist wheels only. Thereafter we extract interest points from the entire image and exclude interest points that are of wheels to constructing a vocabulary for background. The goal of cluster analysis should not be to obtain the smallest possible number of clusters, but to ensure that the resulting clusters are visually compact and contain the same kind of structure. The construction of vocabulary in this work is achieved by using the traditional K-means method. The basic flow of vocabulary construction is depicted in Figure 2.

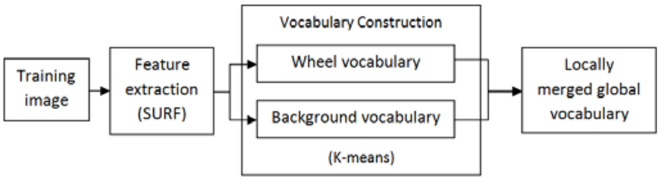

Fig. 2. The process of constructing visual vocabularies using patch-based descriptors in order to verify the two wheels of a car from the detected circles.

B. Wheel Detection

Most of the side-view cars have two wheels that are visible in an image and those wheels are nearly round. Due to the roundness property of a wheel, one can find the wheels by using a circle detection algorithm. In our method, circles are detected by using Hough transform [9]. However, this method would detect all circles some of which may not be the wheels (see Figure 1d). To address this problem, after detecting the circles, we cast votes to each circle with the interest points that match the vocabulary of wheels. The votes are cast by finding similarity between the spatial location of an interest point that matches the vocabulary of wheels, and the centre of a circle, whose similarity is below a cut-off threshold. Thereafter two circles are choosen to be the wheels of the car using the hypothesis that those circles are of near radius, they are disjoint, and have the majority votes among the circles excluding the ones that are rejected in the previous steps. In this case, we do not assume that these two car wheels are in the same line.

C. Bounding box detection

Following the wheel detection step, we tentatively identify a horizontal line, L_1, which makes part of the roof of a car by assuming that the height of a car is about 0.55 of the distance between two wheels. We then detect a set of horizontal lines using Sobel edge detection method and adjust L_1 to its nearest line detected by the Sobel edge detector. We now estimate a horizontal line, L_2, which nearly passes through the top

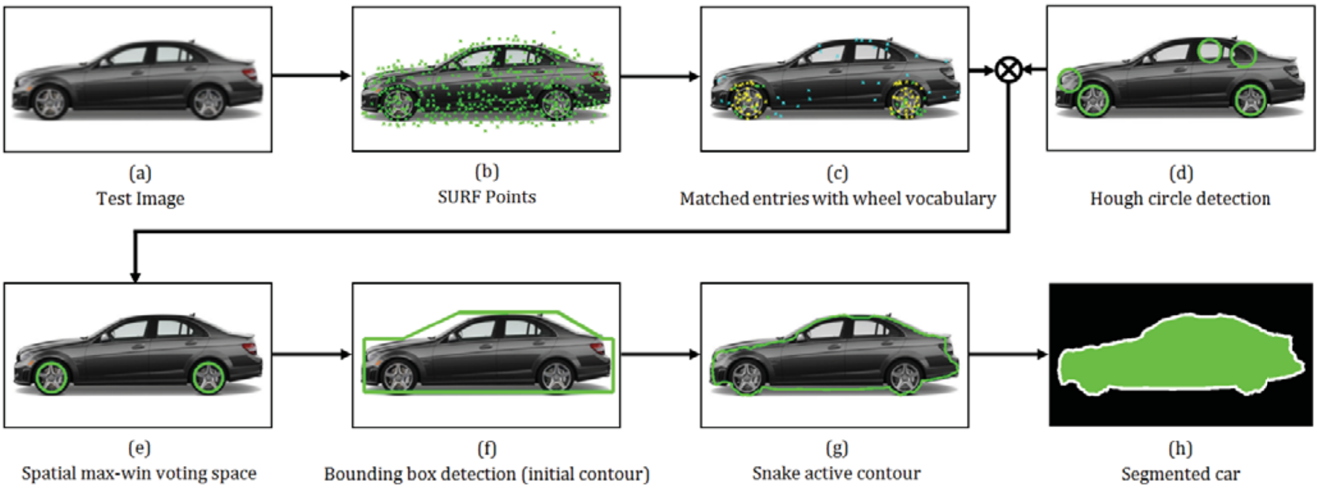
(a)
Test Image
(b)
SURF Points
(c)
Matched entries with wheel vocabulary
(d)
Hough circle detection
(e)
Spatial max-win voting space
(f)
Bounding box detection (initial contour)
(g)
Snake active contour
(h)
Segmented car

Fig. 1. The side-view car detection and segmentation process. Hough circles are detected and compared to the vocabulary of wheels. Matching patches then cast votes to the circles, which lead in identifying wheels of the car. Based on edge detection and length statistics of side-view cars the bounding box (initial contours) is found. Finally, the snake algorithm progress with the initial contour and fits to the outline of the car. Thus, foreground-background segmentation is achieved.

edge of the bonnet, side-windows, and boot of a car which is estimated to be two-thirds of the height of a car. In addition to this, two oblique lines, L_3 and L_4, are chosen about windows such that they intersect L_1 and L_2. Finally, by using the information that the front and rear tracks have an average length of 950mm, we formulate the bounding box of the car using L_1, sub parts of L_2, L_3, L_4, and bottom line of wheels (see Figure 1f). This bounding box serves as the initial contour for the Snake method.

D. Segmentation

In order to segment a side-view car from its background the initial contour is been iterated by active contour or snake method that fits better the boundary of the car. Thereafter, we segment the car area from the background using the active contours achieved by the snake method.

V. Experimental setup

We evaluate the performance of our approach on four different types of side-view car images obtained from Google images. The image collection consists of coupe, hatchback, sedan, and wagon type of cars consisting of 25 images per category. Figure 3 shows some example images of different types of cars presented in our collection of data. The length statistics of side-view cars in this work were learnt using the following make of different types of cars: Alfa Romeo, Audi, BMW, Hyundai, Mercedes Benz, Mitsubishi, Peugeot, Saab, Vauxhall, Volkswagen, and Volvo.

For constructing visual vocabularies, we manually labelled the wheels in 20 car samples and extracted e-SURF descriptors. The descriptors from wheel images were clustered using the K-means method with K = 100 to constructing a vocabulary for wheels. Similarly a background vocabulary was also constructed using the descriptors from background images with K = 100. These 20 car samples are excluded from our 100 image of collections.

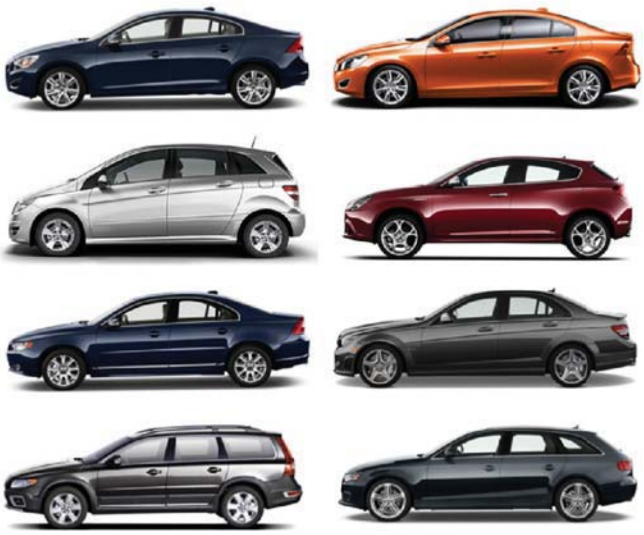

Fig. 3. Example images of side-view cars. (row-1) coupe, (row-2) hatchback, (row-3) sedan, and (row-4) wagon types of cars

For detecting circles in car images, we used Hough circle transform method with a radius range of 20 to 50 and the threshold on the radian magnitude with 10 pixels. We iterate the snake method for 500 times with the standard settings.

VI. Discussion and Conclusion

This paper addresses the problem of side-view car detection in static images together with figure-background segmentation. Our system starts to progress in identifying the bounding box of the car by using the Hough circle transform that we verify for the two wheels by means of a learnt visual vocabulary.

The bounding box is improved by using statistics of side-view cars. This bounding box is then fed to the snake algorithm that iteratively fits to the boundary of the car that segments the object from its background. Our system correctly detected and segmented 95 cars out of the 100 side-view cars that were tested. Our approach is more practical as there are no assumptions made about the two wheels that should exist in the same horizontal line and not in the top of the image.

References

[1] F. Tan, L. Li, B. Cai, and D. Zhang, "Shape Template Based Side-View Car Detection Algorithm", in *Third International Workshop on Intelligent Systems and Applications*, pp. 01–04, 2011.

[2] B. Cai, F. Tan, Y. Lu, and D. Zhang, "Knowledge Template Based Multi-perspective Car Recognition Algorithm", in *International Journal of Information Engineering and Electronic Business*, Vol. 2, No. 2, pp. 38–45, 2010.

[3] Y. Lu, B. Cai, and D. Zhang, "Contour Based Car Recognition Algorithm", in *International Symposium on Computer Network and Multimedia Technology*, pp. 01–04, 2009.

[4] W. Zheng, and L. Liang, "Fast Car Detection using Image Strip Features", in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2703–2710, 2009.

[5] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation", in *International Journal of Computer Vision*, Vol. 77, Numbers 1-3, pp. 259–289, 2008.

[6] Z. Sun, G. Bebis, and R. Miller, "On-road Vehicle Detection: A Review", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, pp. 694–711, 2006.

[7] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", in *International Journal of Computer Vision*, Vol. 60, pp. 91–110, 2004.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features", in *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346–359, 2008.

[9] T. J. Atherton, and D. J. Kerbyson, "Size Invariant Circle Detection", in *Image and Vision Computing*, Vol. 17, Issue 11, pp. 795–803, 1999.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models", in *International Journal of Computer Vision*, pp. 321–331, 1988.

[11] M. S. Nixon, and A. S. Aguado, *Feature Extraction and Image Processing*, Second Edition, Academic Press, pp. 217–245, 2008.