2012 Intelligent Vehicles Symposium
Alcalá de Henares, Spain, June 3-7, 2012

Detection & Classification of Arrow Markings on Roads using Signed Edge Signatures

S. Suchitra, R. K. Satzoda and T. Srikanthan

Abstract— **In this paper, we propose a novel method to robustly identify and classify arrow markings in road images. In the proposed method, simple and unique signatures are first derived for the various arrow types, based on signed edge maps and decomposing the arrows into smaller parts. The signed edge maps are processed using Hough Transform (HT), and the resulting Hough spaces are analyzed systematically, using a set of simple rules. The signatures are rotation-invariant and scale-invariant, thereby making the approach robust to variations in the appearance of the arrow markings. It is shown that the method yields a high detection and classification accuracy, of as high as 97% in the test images considered.**

I. Introduction

Detection of arrow markings on roads is a task of interest for various advanced driver assistance (ADAS) appplications, such as assisted navigation, lane change assistance etc [1], [2]. The crucial information provided by arrow markings, can also be of significant help to the driver in averting many dangerous situations [3]. Although substantial work has been done and reported on detection of lane markings using on-board cameras [4], there is very limited literature available today on detection of arrows in road scenes, as reported in [1] & [2], as recently as 2011.

The limited recent work on arrow detection in on-board camera images includes techniques like inverse perspective mapping (IPM) [2], [5], [3] of input images, using connected components [1], [2], applying wavelets [3] and matching the extracted features using curve/spline fitting models [1] and geometrical pattern matching [5]. For example, Maier et al. [1] presented a general geometric approach using curve based prototype fitting. Connected components was first used to extract arrows and then the extracted features are analyzed using protype fitting models like splines etc. Classification of arrows is done using measures like Hausdorff distance.

In [2] by Philippe et al., an algorithm was proposed, which is based on the following processing steps: marking pixel extraction, detection using connected components before IPM, and recognition based on the comparison with a single pattern or with repetitive rectangular patterns. IPM was also used by Nan Wang et al. in [3] and the extracted features were matched using an improved Haar wavelet feature extraction approach. A LabVIEW based system was developed in [5] for detection, measurement and classification of painted objects on road surfaces. This also used IPM and geometrical pattern matching.

In this paper, we propose a novel method to effectively detect arrow markings on the road, using simple signatures based on customized signed edge maps and Hough Transform (HT). Though HT seems to be a straightforward choice to detect lines that form the arrow boundaries, analyzing the Hough accumulation space is a challenging task, to conclusively detect arrows. We propose effective techniques to deal with these challenges using simple scale-invariant, and rotation-invariant edge signatures, derived from the proposed signed edge maps. A set of rules are also proposed to analyze the Hough accumulation space to successfully detect and classify the arrow markings on roads.

The paper is organized as follows. In Section II, the challenges of analyzing peaks in Hough transform are discussed. The proposed method is then elaborated in Section III with the main steps described in Sections III-A to III-D. The results are presented and discussed in Section IV before concluding the paper in Section V.

II. Challenges in Arrow Detection using Hough Transform

In this section, we take a look at the arrows in greater detail and show the complexities that are involved in detecting the arrows effectively. Typically, arrows on the road surface are of the following types: (a) Simple Forward Arrow, (b) Simple Right Arrow, (c) Simple Left Arrow, (d) Left/Right Arrow, (e) Forward/Left Arrow, (f) Forward/Right Arrow and (g) Forward/Left/Right Arrow, as shown in Fig. 1.

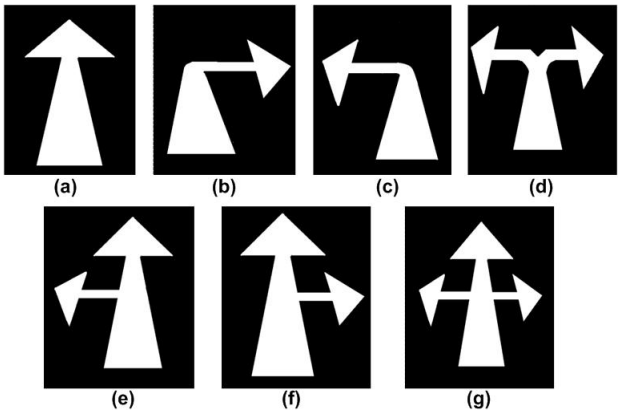

Fig. 1. Arrow types on roads

From Fig. 1, an arrow can be defined by a set of straight lines that are placed at certain relative positions and angular orientations with respect to each other. This set of lines and their properties that define an arrow can be called as the '*signature*' of the arrow. Different types of arrows (Fig. 1) have different signatures based on the number of component lines, their positions and orientations etc. For example, the forward arrow in Fig. 1(a) has two horizontal lines where

978-1-4673-2118-1/$31.00 ©2012 IEEE

as the right arrow in Fig. 1(b) has three horizontal lines. Similarly there are different slant lines at varying angles in each of the arrows. Apart from differences in the types of arrows, the arrow size itself varies as the vehicle moves towards or away from the arrow. The angular orientation of the arrow also changes with horizontal drifts of the vehicle on the lane. The same arrow marking will then have straight lines of different lengths and relative orientations. This is illustrated in Fig. 2 showing forward arrow marking in different orientations Fig. 2(a) to (c), and scales Fig. 2(d) & (e). The arrows may even have slight differences in the thickness or may appear more bright or more faded at different places.

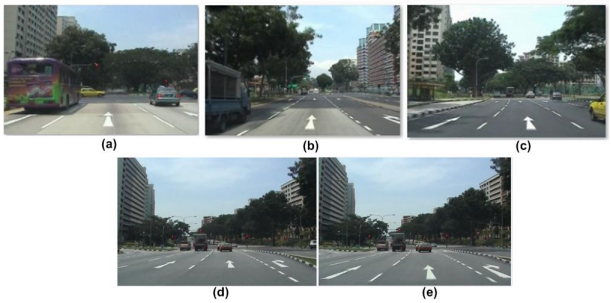

Fig. 2. (a),(b) and (c) show the forward arrow in different orientations; (d) and (e) show the different scales of the same arrow as the vehicle moves forward

Hough Transform (HT) is one of the most common line detection methods [6]. The straight line signature of an arrow in an image can be analysed by studying the peaks that the straight line edges of the arrow form in its Hough accumulation space. The (ρ, θ) values corresponding to the detected peaks indicate the spatial position of the lines in the image and their orientations. The relationship between the various (ρ, θ) pairs of the peaks detected, is indicative of the arrow signature. However, directly analysing the Hough space to detect the arrow signature is neither conclusive nor straightforward. This is because of the following reasons:

1) Firstly, the different (ρ, θ) pairs of the arrow edges, for a given arrow type, do not always occur at the same pre-defined (ρ, θ) positions. This is because the arrows can vary in position, scale and orientation. Therefore, it is more than just a simple check for the presence or absence of peaks in pre-defined (ρ, θ) positions. Rules need to be generated, that associate peaks with each other, based on 'relative' position and orientation of peaks w.r.t other peaks.
2) The number of lines forming an arrow is atleast six. This means a sizeable number of rules to associate the peaks with each other.
3) The (ρ, θ) values of the arrow edges can span the entire range of angles in Hough space. Because of the large number of allowable (ρ, θ) combinations, stray lines can cause false positives. Any stray line falling in that angle range could be easily mistaken for a valid edge of an arrow.
4) Moreover, peak detection in Hough space inherently suffers from the problem of clutter. For these reasons, the Hough space alone is not often used to detect patterns.

Fig. 3(a) & (b) show Hough accumulation spaces of forward arrows corresponding to the forward arrows in Fig. 2(a) & (c) respectively. It can be seen from Fig. 3 that the two Hough spaces look substantially different with different number number of peaks, locations of peaks and relative assocation between them.

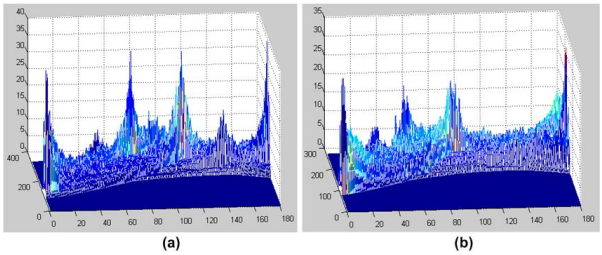

Fig. 3. Edges of the forward arrow in the host lane in Fig. 2(a) and (c) giving different Hough spaces in (a) and (b) respectively.

Defining simple and unique signatures for each arrow type and being able to detect these signatures effectively in the image, forms the basis of our proposed method. The signatures should be scale-invariant and rotation-invariant.

III. Proposed Arrow Road Marking Detection Method

In this section we describe in detail the proposed method to detect arrow markings on road surface. The proposed method comprises three main steps: (1) Signed Edge Maps Generation (2) Hough Transform Computation (3) Arrow Signature Detection in Hough Accumulation Space. Before we propose the different methods, we first define the region of interest (RoI) where we look for the arrow markings. Firstly, it is to be noted the top half of a road scene usually consists of features such as sky, trees etc., that are not of interest to us [6]. The lower half, referred to as 'near-view' is where we expect the arrows to appear. Secondly, in this paper, we are interested to detect arrow markings only in the host lane, as this is what is most relevant to the driver. And thirdly, arrow detection, as a task, is usually integrated into bigger applications, where lane markings are already detected. Given the above, the lane marking positions are used to extract the host lane region in the near view, and this forms the RoI for further detection of arrow markings.

A. Proposed Use of Signed Edge Maps

In this section, we describe the first step in the proposed arrow marking detection. In this step, we generate what we call as '*signed edge maps*' from the edge detection process. The conventional edge detection process involves applying gradient kernels such as Sobel [6] on the RoI I to give gradient maps G_x and G_y in the x and y directions. G_x and G_y are then used to compute the gradient magnitudes G. The edge map E is computed by thresholding G with

threshold Th. Sometimes, just the horizontal or vertical gradient maps are thresholded to yield partial edge maps E_x or E_y respectively. The following equations put the entire conventional edge map generation in a nutshell:

$$\begin{aligned} &G_x = I \otimes S_x \qquad G_y = I \otimes S_y \\ &G = \sqrt{G_x^2 + G_y^2} \\ &E = G > Th, \qquad E_x = |G_x| > Th, \quad E_y = |G_y| > Th \end{aligned} \tag{1}$$

It can be seen that conventionally, the sign of gradient value will not affect the decision to qualify that pixel as an edge pixel in the edge map. We now propose to use the gradient signs to decompose the horizontal edge map E_x further into two signed edge maps called positive and negative edge maps, E_{x+}, and E_{x-}, such that:

$$\begin{aligned} &E_{x+}(x,y) = 1 \; if \; G_x(x,y) \geq 0 \wedge E_x(x,y) = 1 \\ &E_{x-}(x,y) = 1 \; if \; G_x(x,y) < 0 \wedge E_x(x,y) = 1 \\ &E_x = E_{x+} \cup E_{x-} \end{aligned} \tag{2}$$

Let us now understand what these signed edges mean. We identify an arrow marking as a light colored closed shape, bounded by linear edges, having a dark background (road surface). If we scan the image from left to right, then we encounter one set of bounding edges of the arrow, which have dark $\rightarrow$ light intensity transitions ($\rightarrow$ is used to denote direction of intensity transition), and another set of edges, which have light $\rightarrow$ dark intensity transitions. This direction of the intensity transition is captured by the sign of the edge map. E_{x+} has the edge pixels that are formed by dark $\rightarrow$ light transition when we scan the image in from left to right, i.e. x-direction. E_{x-} comprises edges that are formed by light $\rightarrow$ dark transition in the x-direction. This also implies that E_{x+} contains the edges on the left of the arrow, and E_{x-} has the right edges of the arrow, if the image were to be scanned from left to right. For ease of understanding and explanation, we call E_{x+} as the left edge map E_l and E_{x-} as the right edge map E_r, in the rest of the paper. The decomposed edge maps are processed further in subsequent steps to ascertain the presence of arrow markings.

The rationale behind the decomposition of E_x into the signed edge maps is two-fold. Firstly, we bring in an additional constraint in the line search, by specifying the exact direction of the intensity transitions. By doing so, we can, not only look out for a given position or angular orientation of the lines, but also whether they are bounding edges in the left or right of the arrow. In other words, these signed edge maps help us to propose unique signatures to identify each type of arrow. Secondly, they help in reducing the complexity of Hough accumulation space, thereby reducing the effort needed in its analysis. Since the edge map is split into simpler edge maps, the Hough accumulation space is also split into simpler spaces that can be analyzed faster and effectively.

Fig. 4 shows an example of the two signed edge maps E_l and E_r for an arrow image. It can be seen that the left edges of the arrow are captured in E_l and the right edges in E_r in Fig. 4(b) and (c) respectively.

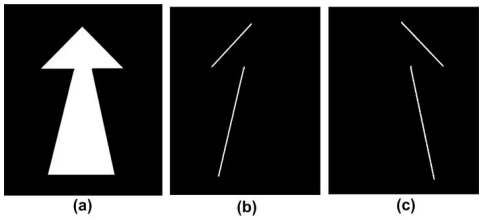

Fig. 4. (a) Sample forward arrow (b) E_l (c) E_r

B. Applying Hough Transform

After the edge map is decomposed into E_l and E_r, we apply Hough transform (HT) on both the edge maps to generate Hough accumulation spaces H_l and H_r respectively. The different linear edges in E_l and E_r are captured as peaks in H_l and H_r. The (ρ, θ) tuples corresponding to the peaks in the Hough spaces are stored separately in P_l and P_r, i.e. the linear edges in the left edge map E_l are now captured in $P_l = \{(\rho_{l1}, \theta_{l1}), (\rho_{l2}, \theta_{l2}), \ldots, (\rho_{ln}, \theta_{ln})\}$. Similarly the other peaks' set P_r is generated. These values will now be analyzed to look for the signatures of the arrow markings.

C. Defining and Detecting Arrow Signatures

The next step is to search for arrow signatures in the generated peak sets P_l and P_r. In order to do this, we first define arrow signatures that are unique to each of the arrow types, and also scale/rotation invariant, using the decomposed edge maps E_l and E_r. We break the arrow into its component parts and build the arrow signature using signatures of each of its parts. From Fig. 1, it can be seen that an arrow of any type comprises of a main arrow tail and one or more of the following types of arrow heads (1) Forward facing arrow head, (2) Left arrow head and (3) Right arrow head.

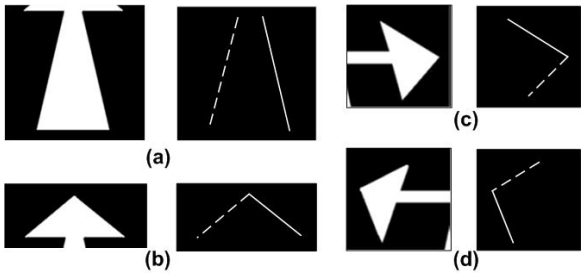

Fig. 5. Left-slant (solid) and right-slant (dashed) lines are indicated on different arrow parts: (a) Main arrow tail, (b) Forward facing arrow head, (c) Right arrow head, and (d) Left arrow head

It can be observed that each of these arrow parts can be defined by a pair of lines - one slanting to the left (referred to as left-slant line) and the other to the right (right-slant line). Fig. 5 shows the left and right slant lines as solid and dashed lines respectively, that define each of the arrow parts. A line is tagged as a right-slant or a left-slant line based on the θ value of the line. If a line l_1 is given by (ρ_1, θ_1), then l_1 is tagged as left-slant or right-slant based on the equations below:

$$l_1 = \left\{ \begin{array}{l} right \; slant \; if \; 15^o \leq \theta_1 \leq 85^o \\ left \; slant \; if \; -85^o \leq \theta_1 \leq -15^o \end{array} \right\} \tag{3}$$

These right and left slant lines are captured in E_x and in turn reflected in the decomposed signed edge maps E_l and E_r, based on the nature of intensity transitions (defined by equation (2)). Let us take the example of the arrow head in Fig. 5(b). There is one right-slant line and one left-slant line. The right-slant lines forms the left edge of the arrow head. Hence it is captured in E_l. Likewise, the left-slant line is captured in E_r.

Similarly, for each part of the arrow, the left-slant and right-slant lines of the pair occur either both in E_l, or both in E_r, or one in E_l and the other in E_r. For example, the main arrow tail for all arrow types will have one right-slant line that is captured in the left edge map E_l and one right-slant line that is captured in the right edge map E_r. Whereas, the right arrow head will have both the left-slant and right-slant lines occuring in E_r. Table I lists the edge maps in which the right and left slant lines are captured for each part of the arrows. Based on this table, we first extract the

TABLE I
LOCATION OF RIGHT AND LEFT SLANT LINES FOR EACH ARROW PART

Arrow Part	Right Slant	Left Slant
Main Arrow Tail	1 in E_l	1 in E_r
Forward Arrow Head	1 in E_l	1 in E_r
Left Arrow Head	1 in E_l	1 in E_l
Right Arrow Head	1 in E_r	1 in E_r

candidate line pairs for each arrow part and subject them to a set of simple association rules to confirm the presence of that part. The order in which this is done is the following: (1) Main arrow tail, (2) Forward arrow head, and (3) Side arrow heads. These steps are explained below.

1) Detecting the Main Arrow Tail: The main arrow tail is a common part of all arrow types. Hence, we first detect the presence of the main arrow tail to infer the presence of an arrow marking. In order to do this, all line pairs that are potential candidates representing the main arrow tail are first extracted. From Table I, the main arrow tail is supposed to have one right-slant line in E_l and one left-slant line in E_r. Therefore, all right-slant lines in E_l and all left-slant lines in E_r are selected and paired up in various combinations to yield the candidate line pairs. An illustration of all the candidate line pairs extracted in this step for arrow tail detection, is shown in Fig. 6 (the top row), for a simple forward arrow. As there are two left-slant lines in E_r and two right-slant lines in E_l, in the case of a simple forward arrow, this results in a total of four line pairs, as shown in Fig. 6. These lines are given by the (ρ, θ) tuples in P_l and P_r, i.e. $P_l = \{(\rho_{l1}, \theta_{l1}), (\rho_{l2}, \theta_{l2})\}$ and $P_r = \{(\rho_{r1}, \theta_{r1}), (\rho_{r2}, \theta_{r2})\}$.

But only one of these pairs is a valid pair that actually forms the bounding edges of the arrow tail. If (ρ_{l1}, θ_{l1}) and (ρ_{r1}, θ_{r1}) correspond to the right and left slant lines of the main arrow tail, then we need associate these two pairs together as the bounding edges of the main arrow tail, i.e. (ρ_{l1}, θ_{l1}) must be paired with (ρ_{r1}, θ_{r1}) only, and not with (ρ_{r2}, θ_{r2}), in P_r.

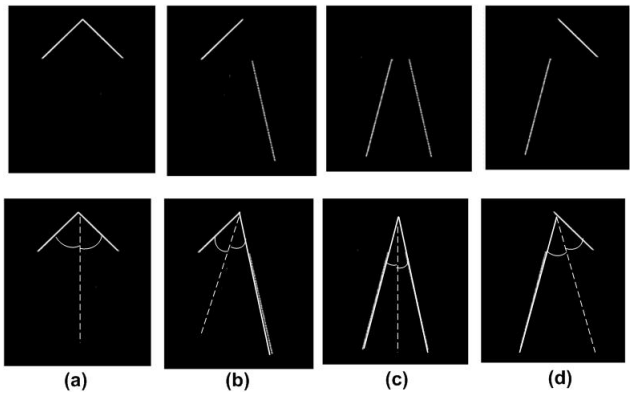

Fig. 6. (a)-(d): Top row shows the candidate line pairs for main arrow tail of a simple forward arrow; Bottom row shows the axis for each of the line pairs

This is accomplished by a simple association rule based on the '*axis*' of each candidate line pair. Axis is defined as the line that equally divides the angle of intersection of the two component lines. The association rule that we set on each candidate pair, to confirm the presence main arrow tail is that, the axis of the valid line pair passes through the vanishing point. This is because the main arrow tail is always pointed towards the vanishing point, as illustrated in Fig. 7 (a). The vanishing point is obtained by the intersection of the left and right lane markings, as shown by the solid black lines in Fig. 7 (a). It can be seen that the axis of the main arrow tail (shown as dashed line) passes through the vanishing point.

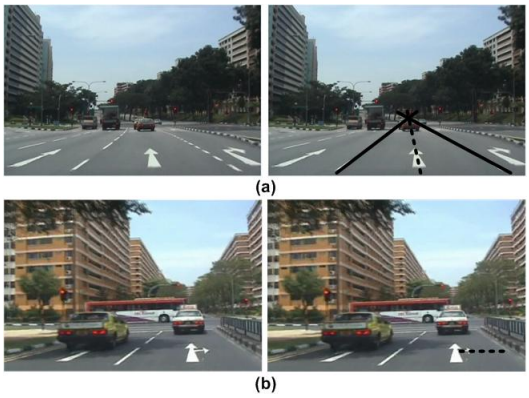

Fig. 7. Sample road scene showing: (a) axis of main arrow tail passing through the vanishing point, (b) axis of side arrow head being a horizontal.

Now applying this constraint on the four candidate line pairs shown in the top row of Fig. 6, the line pairs Fig. 6(b) & (d) are eliminated. This is because, for the forward arrow that is considered in Fig. 6, the vanishing point is vertically in front of the arrow head, which is satisfied by line pairs in Fig. 6(a) & (c). Since the main arrow tail is invariably longer than the arrow head, we can further confirm that the line pair in Fig. 6 (c) is the main arrow tail, using the Hough peak heights.

Mathematically, this constraint with the axis of the main

arrow tail is checked in the following way, for every pair of right and left slant lines. Referring to Fig. 8, given the (ρ, θ) tuples of right and left slant lines as (ρ_l, θ_l) and (ρ_r, θ_r) respectively, the equations of right and left-slant lines, rs and ls, are given by:

$$\begin{aligned} right-slant(rs): &\quad xcos\theta_l + ysin\theta_l = \rho_l \\ left-slant(ls): &\quad xcos\theta_r + ysin\theta_r = \rho_r \end{aligned} \tag{4}$$

Consider any two horizontal lines drawn at y_1 and y_2 such that they intersect two slant lines rs and ls at points P_1, P_2, P_3 and P_4, as shown in Fig. 8. The points of intersection are computed by substituting y_1 and y_2 in the equations in (4). If P_{m1} and P_{m2} are the mid points of P_1 and P_2, and P_3 and P_4, respectively, then the axis is defined by the line $P_{m1}P_{m2}$. Now in order to verify if this axis passes through the vanishing point P_{vp}, we check the collinearity of P_{m1}, P_{m2} and P_{vp}.

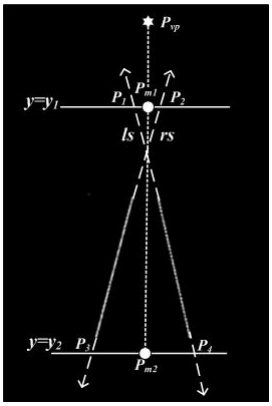

Fig. 8. Figure showing how the axis of main arrow tail is computed.

2) Detecting Arrow Heads: Once the detection of main arrow tail is confirmed, we now look for the other arrow parts in exactly the same way as the main arrow tail. The only difference is that: (1) the line pairing will vary for each part according the listing in Table I, and (2) the orientation of the axis is dependent on the direction of the arrow head. The axis is pointed towards the vanishing point in the case of forward arrow head, whereas the axis is found to be a horizontal line for right and left arrow heads. An example is shown in Fig. 7(b) for right arrow head using a dashed line.

D. Arrow Detection and Classification based on Detected Arrow Parts

Once we infer the presence or absence of the various arrow parts based on their signatures, as explained in the previous sections, we can confirm the presence of the arrow marking and classify it. Table II summarizes the various combinations of arrow parts that define each arrow type. The entire algorithm to detect different arrow markings is illustrated in Fig. 9, showing all the processing steps.

IV. Results & Discussion

In this section, we present the results of arrow marking detection using the proposed method. The proposed method

TABLE II

Arrow Classification using Presence/Absence of Arrow Parts

Type of Arrow	Main Arrow Tail	Forward Arrow Head	Left Arrow Head	Right Arrow Head
Forward Only	✓	✓		
Left Only	✓		✓	
Right Only	✓			✓
Forward & Left Turn	✓	✓	✓	
Forward & Right Turn	✓	✓		✓
Left & Right Turn	✓		✓	✓
Anywhere Turn	✓	✓	✓	✓

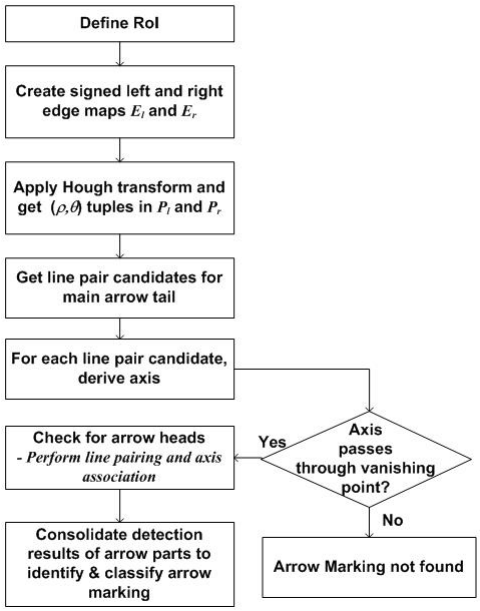

Fig. 9. Block diagram showing the proposed method to detect different arrow markings

was applied on images that were extracted from videos taken on Singapore highways and major roads, during day time and late evening time. The image resolution of the input images is set to 480×720 pixels and more than 1100 images were used to evaluate the proposed method. The set of images had the different kinds of road markings as listed in Fig. 1. It was found that the proposed method has detected the arrows correctly in 97% of the image test cases.

Fig. 10 shows the some of test cases and the detection result (as inset) using the proposed method. It can be seen that the proposed method is successfully able to detect the different types of arrows shown in Fig. 10.

The scale and rotation invariance of the proposed method is illustrated in Fig. 11. The two images in the left column (Fig. 11(a)) show the detection of the forward arrow marking of two different scales as the vehicle moved forward. In two images in the right column (Fig. 11(b)), the forward-right turn arrow varies in scale as well as orientation. The proposed method is able to detect the above arrows successfully, showing both rotation and scale invariance.

False/Mis-Detections: There are a few cases when the

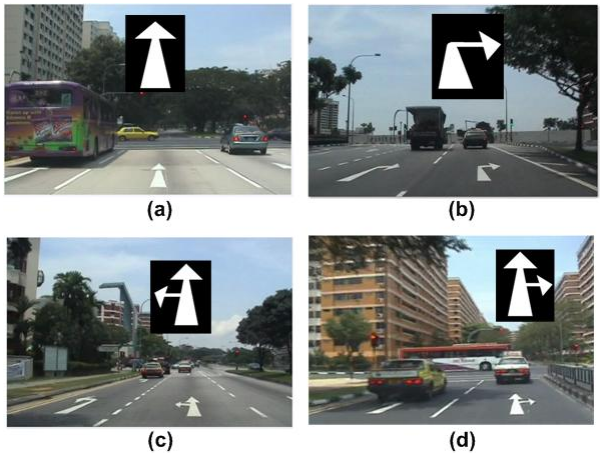

Fig. 10. Different types of arrow markings are detected using the proposed method.

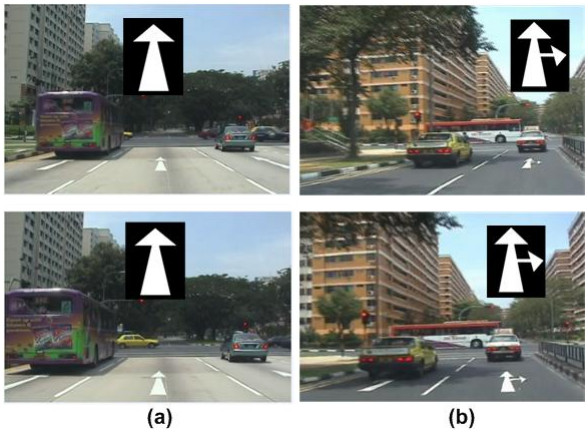

Fig. 11. Scale and rotation invariance of proposed method.

proposed method failed to detect the arrow markings. Fig. 12 shows some of these cases. One of the common cases for mis-detection is when the distance between the arrow and the vehicle is beyond a certain detectable range. Fig. 12(a) & (b) show two such cases, where the edges of the arrow marking are not distinctive enough to yield correct signatures. For example in Fig. 12(a) the right turn arrow is not detected because the number of pixels are too few to yield anything meaningful in the Hough space. Similarly, in Fig. 12(b), there is a forward arrow marking but the edges of the arrow head merge with the edges of the main arrow tail. This results in an incomplete signature. This leads us to define a maximum distance from the vehicle beyond which the arrow markings cannot be conclusively detected.

The other cases where the proposed method fails to make complete signatures are when the arrow marking is either partially present in the RoI (Fig. 12(c)) or when the arrow marking is occluded by a vehicle (Fig. 12(d)). The arrow markings were also mis-detected in the cases of poor illumination because the thresholding in the edge detection missed out some edges of the arrow, resulting in partial signatures.

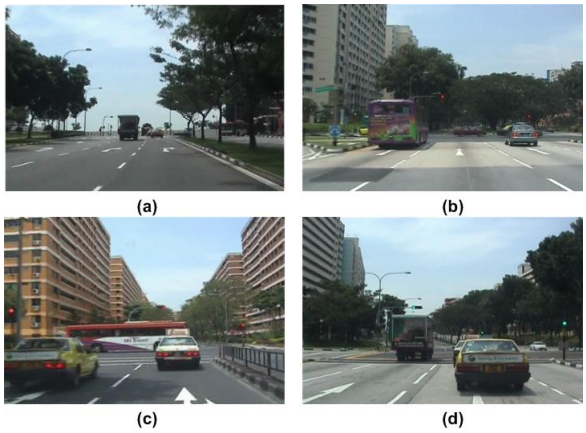

Fig. 12. Mis-detection due to: (a)&(b) the distance between arrow marking and the vehicle, (c)&(d) presence of partial/occluded arrow markings

V. Conclusion

It is shown that the techniques proposed in this paper successfully detect and classify arrows in road scenes. The decomposition of the arrows into their component parts and the use of signed edge maps, enable generation of simple signatures that are scale-invariant and rotation-invariant, to uniquely define each arrow type. We have shown how the left and right edge maps, coupled with axis direction, suffice to constrain the lines belonging to arrows and reject all non-arrow candidates. The method has been validated on test image sequences containing more than 1100 real images and it is shown that the results are of high detection/classification accuracy. The mis-detections are analysed separately and form scope for future work.

References

[1] G. Maier, S. Pangerl, and A. Schindler, "Real-time detection and classification of arrow markings using curve-based prototype fitting," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, june 2011, pp. 442 –447.

[2] P. Foucher, Y. Sebsadji, J.-P. Tarel, P. Charbonnier, and P. Nicolle, "Detection and recognition of urban road markings using images," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, oct. 2011, pp. 1747 –1752.

[3] N. Wang, W. Liu, C. Zhang, H. Yuan, and J. Liu, "The detection and recognition of arrow markings recognition based on monocular vision," in *Control and Decision Conference, 2009. CCDC '09. Chinese*, june 2009, pp. 4380 –4386.

[4] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system and evaluation," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, march 2006, pp. 20 – 37.

[5] I. Chira, A. Chibulcutean, and R. Danescu, "Real-time detection of road markings for driving assistance applications," in *Computer Engineering and Systems (ICCES), 2010 International Conference on*, 2010, pp. 158 – 163.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall, 2002.