Proceedings of the 16th International IEEE Annual Conference on
Intelligent Transportation Systems (ITSC 2013), The Hague, The
Netherlands, October 6-9, 2013

Identifying Lane Types: A Modular Approach

S.Suchitra[1], R.K.Satzoda[1] and T.Srikanthan[1]

Abstract— **Lane detection is a problem that has been extensively studied by the research community in the past two decades. However limited literature can be found on techniques to distinguish the various types of lane markings - such as solid, dashed, single, double, zigzag etc. In this paper, we present a modular approach to detect and distinguish a wide range of lane markings. The fundamental processing module for detecting basic lane markings (BLM) is first proposed, after which we show how this can be deployed for distinguishing the various lane marking types. The underlying principle is that any lane marking can be broken down into one or more BLMs. A modular architecture is presented to detect and distinguish the various lane markings using the proposed modules. The techniques are evaluated on the road marking dataset in [8] and is shown to yield a high detection accuracy.**

I. INTRODUCTION

Surround scene interpretation with respect to the host vehicle is of growing importance as advanced driver assistance systems (ADAS) are heading towards providing the driver with comprehensive information of the current road scene. Information about lane positions has been widely used for a number of ADAS applications over the past two decades [4], [1], [5]. Insufficient literature, however, is found on distinguishing the various lane markings such as dashed versus solid, single versus double, zigzag markings etc. [4] summarizes many existing lane detection methods, and it is evident that the related research has primarily been in the context of basic lane markings (BLM), where we define BLM as single linear markings that separate lanes on the road. Limited existing techniques to distinguish dashed versus solid lane markings are either based on frequency analysis [2], [3] or employ scan lines [9], [6].

A computationally efficient method to detect BLM in complex road scenarios was previously proposed by us in [7] that uses gradient angle histograms and Hough Transform. In this paper, the basic lane detection module is reused in a modular fashion to construct a framework that can detect road markings of different types. Evaluation performed on [8] database shows that the algorithm yields high accuracy and the underlying architecture is shown to have attractive properties of modularity and computational simplicity.

II. BLM DETECT MODULE

In this section, we first briefly recap the basic lane detection algorithm previously proposed in [7], and then describe the proposed BLM detect module.

The region of interest (RoI) is taken as the near view corresponding to the bottom of the image. Firstly, the RoI is divided into blocks of size $m \times n$, and are subjected to generation of straight line edge maps (SLEM) using gradient angle histograms (GAH), described in [10]. The resultant edge maps contain mainly straight line edges and thus capture the lane marking features. Next, in order to eliminate lines that do not belong to lane markings, only those line pairs are chosen which exhibit edges of opposite nature, and separated by a known distance apart. To check for these efficiently, only the positive edges are first sent into the HT. The angle ranges where peaks are detected are now searched for presence of peaks in the negative edge map. The detected peaks give the position and angular orientation of the line pairs corresponding to the lane markings. A history based tracking is employed to remove temporal outliers. Angle range for left and right lane markings were set to be (20° to 70°) and (-20° to -70°) respectively. The expected thickness for lane markings was taken to be in the range of 5 to 15 pixels.

Therefore, the inputs to the algorithm are the RoI, expected angle range $\Delta\theta$, and expected thickness Δd. The outputs (ρ, θ) correspond to position and angular orientation of the lane markings.

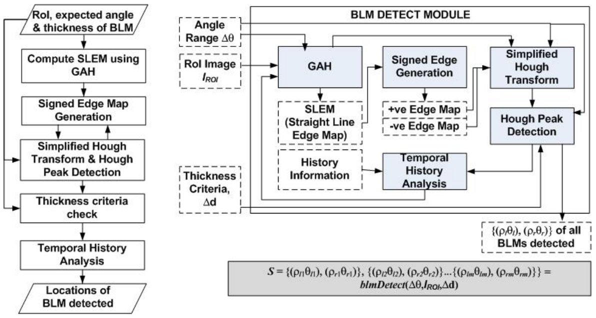
RoI, expected angle & thickness of BLM
Compute SLEM using GAH
Signed Edge Map Generation
Simplified Hough Transform & Hough Peak Detection
Thickness criteria check
Temporal History Analysis
Locations of BLM detected
BLM DETECT MODULE
Angle Range Δθ
RoI Image I_{ROI}
Thickness Criteria, Δd
GAH
SLEM (Straight Line Edge Map)
History Information
Signed Edge Generation
+ve Edge Map
-ve Edge Map
Temporal History Analysis
Simplified Hough Transform
Hough Peak Detection
$\{(\rho_l\theta_l), (\rho_r\theta_r)\}$ of all BLMs detected
$S = \{(\rho_{l1}\theta_{l1}), (\rho_{r1}\theta_{r1})\}, \{(\rho_{l2}\theta_{l2}), (\rho_{r2}\theta_{r2})\}...\{(\rho_{lm}\theta_{lm}), (\rho_{rm}\theta_{rm})\}\} =$
$blmDetect(\Delta\theta, I_{ROI}, \Delta d)$

Fig. 1. Flowchart summarizing the steps in BLM Detection, BLM Detect Module

The above set of steps can be generalized to detect any similar lane-like artifacts oriented in other angles or having different thicknesses, by adjusting the region of interest, expected angle range and expected thickness to appropriate values. Based on Fig. 1, we formulate eq. (1):

$$S = \texttt{blmDetect}(\Delta\theta, I_{ROI}, \Delta d) \quad (1)$$

where S corresponds to set of m valid line pairs detected, given by
$\{\{(\rho_{l1}, \theta_{l1}), (\rho_{r1}, \theta_{r1})\}, \ldots \{(\rho_{lm}, \theta_{lm})\}\}$.

[1] The authors are from the Center for High Performance Embedded Systems at Nanyang Technological University, Singapore `assuchitra, rksatzoda, astsrikan at ntu.edu.sg`

978-1-4799-2914-613/$31.00 ©2013 IEEE

III. Deploying BLM Detect Module to Distinguish Lane Types

A. Solid versus Dashed Lane Markings

The BLM detection algorithm can be deployed to distinguish lane markings as solid or dashed by relying on the continuity properties of the solid and dashed markings. A dashed lane marking is visibly characterized by gaps in the BLM occurring at intervals, as we move along the BLM in a given image. A solid lane marking on the other hand has a continuous presence of BLM when we traverse along the marking. This discontinuity of dashed markings within an image frame translates into a temporal discontinuity across frames at a given fixed window. In other words, if one were to be viewing the near view through a fixed window, then the presence and absence of the BLM would be seen in an alternating fashion. On the other hand, a solid marking shows a continuous presence of BLM in the near view window with no gaps. Based on this, we propose a two-pronged approach to distinguish dashed from solid lane markings using (a) spatial continuity check within an image frame and (b) temporal continuity check across time frames. It is shown that having both the spatial and temporal checks strengthens the accuracy of the proposed approach.

1) Approach 1: Spatial Continuity Analysis: The BLM detect module is first used to get the position and angular orientation of the lane marking. The block in which the BLM is detected is selected, along with its neighboring blocks on either side horizontally. We explain the algorithm for the left lane and the same can be extended for the right lane also. Let the left and right edges of the left host lane determined by `blmDetect` be given as $L^l_L(\rho^l_L, \theta^l_L)$ and $L^r_L(\rho^r_L, \theta^r_L)$ respectively. Let the block in which the lane was found be $B(i,j)$, where i is the last row (i.e. $i = 8$). Therefore, in order to detect the spatial continuity check, we consider blocks $B(i,j-1)$ to $B(i,j+1)$. As an example, in the image shown in Fig. 3 (a), the left lane marking is detected at $j = 3$; therefore, blocks 2,3 and 4 are considered for processing the left lane marking. Similarly, blocks 5, 6 and 7 are considered for the right host lane. The reason why these adjacent blocks are considered is because the lane marking is expected to extend into these blocks. Therefore, checking for the presence of the lane features in these blocks will determine the type of the lane marking.

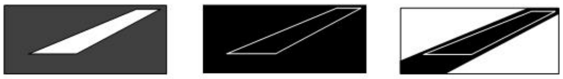

Fig. 2. Generating mask for spatial continuity check (a) Sample image block containing left lane marking (b) Edge map (c) Result of AND-ing the edge map with mask

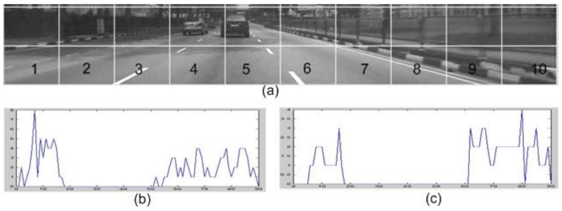

Fig. 3. Using spatial continuity analysis for distinguishing solid vs dashed lane markings (a) Image divided into blocks

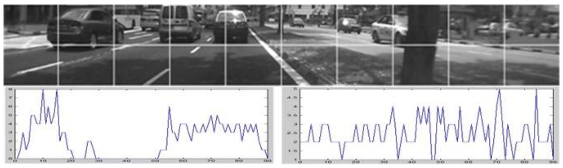

Fig. 4. Using spatial continuity analysis for distinguishing solid vs dashed lane markings

Next, the positions of the left and right edges of the lane marking is used to generate a mask M_L, such that it contains the lane marking. This mask is generated so that we can probe the edges along the lane marking only and eliminate all other edges. We consider a small allowance to the positions of the left and right edges to capture the lane markings within the mask. This is illustrated in Fig. 2. Fig. 2 (a) shows an image block with the lane marking, (b) shows the edge map of the marking and (c) shows the mask that needs to be considered that can encapsulate the lane marking. Given $L^l_L(\rho^l_L, \theta^l_L)$ and $L^r_L(\rho^r_L, \theta^r_L)$, the mask in block $B(i,j)$ is generated by checking if the coordinates (x,y) in $B(i,j)$ satisfy the following:

$$(\rho^l_L - \delta_\rho < (x\cos\theta^l_L + y\sin\theta^l_L)) \wedge ((x\cos\theta^r_L + y\sin\theta^r_L) < \rho^r_L - \delta_\rho) \quad (2)$$

In the above equation, δ_ρ is the small allowance we set to capture the lane marking within the mask. All coordinates (x,y)s in $B(i,j)$ that satisfy the above equation will result in M_L. In other words, all coordinates that are located to the right of the left edge and left of the right edge, satisfy the above equation resulting in M_L. This mask is then used to selectively get the lane edges along the left lane. Given the straight line edge map (SLEM) for block $B(i,j)$, represented by $E_{SL(i,j)}$, we get the selective edge map along the lane $E_{ML(i,j)}$ as:

$$E_{ML(i,j)} = E_{SL(i,j)} \bigodot M_L \quad (3)$$

where $\bigodot$ represents elementwise multiplication. The same procedure is repeated for all the three blocks, i.e. $B(i,j-1)$ to $B(i,j+1)$ and the E_{ML}s of the three blocks are concatenated in the same order to get the overall E_{ML}.

We generate a vertical projection P_L of E_{ML} by summing all edge pixels along the horizontal direction. We traverse row by row in the edge map contained within the mask region and the sum of edge pixels is computed for each row and these values are projected into a linear array, which is referred as projection array P_L. The projection arrays are shown in Fig. 3 (b) and (c) for the left and right lane markings respectively. The projection array has 90 values corresponding to the block height of 90, where each row yields an entry in the array.

A continuous string of 0's in this array indicates a gap in the marking, and hence a dashed marking. In the illustration in Fig. 3, which is a case of dashed markings in both left and right sides of the host lane, the projection arrays are shown in Fig. 3 (b) and (c). Continuous strings of zeros, of lengths more than 30 and 40 are seen for the left and right lane markings respectively, clearly indicating the presence of dashed markings on either side. An example of a host lane with solid markings on one side and dashed markings on the other is shown in Fig. 4. The left side of the host lane contains a dashed marking, which is reflected by a stretch of more than 20 zeros in its corresponding projection array. On the other hand, the right side shows a solid marking, which is reflected by the absence of any zero-stretches in the projection array. The stray zeros that are present due to breaks in the edges and are discarded as outliers. Similarly, if there are stray edge pixels in the gap regions, they are ignored as outliers. The steps for spatial continuity check are summarized as a flowchart in Fig. 6 after the following section on temporal continuity check.

2) Approach 2: Temporal Continuity Analysis: The above spatial continuity analysis can be used to distinguish the lane markings as dashed or solid in a static frame, and this method can be applied even if the vehicle is stationary. However, for moving vehicles, temporal history information can serve as an additional cue to confirm the finding, through a continuity analysis of the markings across time frames is performed. This is based on the observation that a solid marking not only exhibits spatial continuity, but also a temporal continuity. On the other hand, a dashed marking exhibits gaps in the spatial domain (as shown in Sec. III-A.1), as well as across time frames, as explained below.

In order to perform temporal continuity check, a small slice of the image from the bottom most row of blocks (near view), 10 pixels in height is chosen as the window for viewing the lane markings. The presence or absence of lane markings in this image slice is checked in every image frame, by applying a mask around the lane position that is output as (ρ,θ) by the proposed lane marking detection method. The same mask that was generated for the spatial continuity check in the previous approach is used. The lowermost 10 pixel window from bottom is used to check if there are any edge pixels that are captured in the mask. The total number of edge pixels in this window N_L and N_R are determined for the left and right host lanes respectively. If this count is greater than a given threhsold, a temporal variable $R(k)$ is updated, where k represents the time instant. In other words, $R(k)$ is used to keep track of whether or not the lane marking was detected in the image slice in the k-th frame. The R values of the previous M frames in the history are checked. When a minimum number of zeros (marked by Z_{min}) is found in this array, it indicates a gap in the marking, and hence presence of a dashed marking. Also, when the number of zeros exceeds a certain threshold value Z_{max}, it means the vehicle has entered a zone with no lane markings. The temporal continuity check is summarized in the equation below:

$$Z_{max} > M - \left[\sum_{i=0}^{M} R(k-i)\right] > Z_{min} \tag{4}$$

where $M = f(FR,V)$, FR being the frame rate per second and V being the velocity of the vehicle in meters per second. The number of frames M that are stored and checked for temporal continuity depends on the frame rate and the velocity of the vehicle. For a stationary vehicle of a very slow moving vehicle, only spatial continuity check is employed. The temporal continuity check can be triggered for moving vehicles.

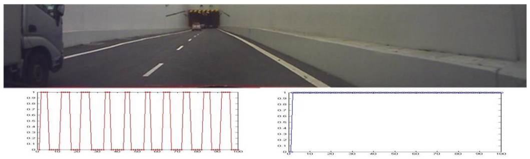

Fig. 5. Using temporal continuity check for differntiating solid vs dashed lane markings

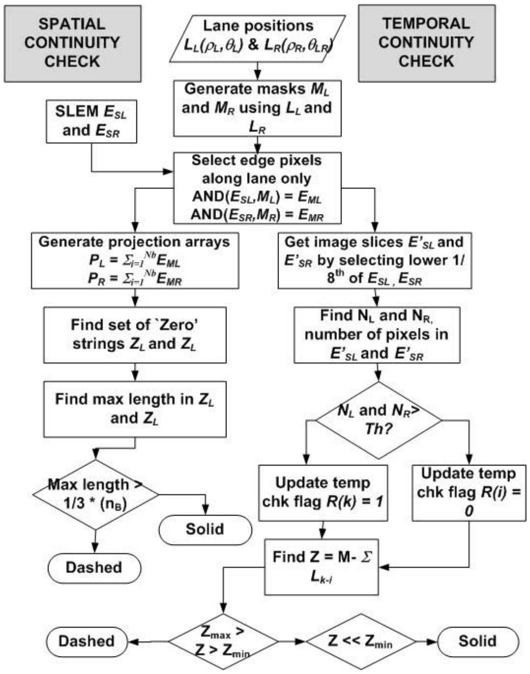
SPATIAL CONTINUITY CHECK
Lane positions $L_L(\rho_L,\theta_L)$ & $L_R(\rho_R,\theta_{LR})$
TEMPORAL CONTINUITY CHECK
Generate masks M_L and M_R using L_L and L_R
SLEM E_{SL} and E_{SR}
Select edge pixels along lane only AND(E_{SL},M_L) = E_{ML} AND(E_{SR},M_R) = E_{MR}
Generate projection arrays $P_L = \Sigma_{i=1}^{Nb} E_{ML}$ $P_R = \Sigma_{i=1}^{Nb} E_{MR}$
Get image slices E'_{SL} and E'_{SR} by selecting lower 1/8th of E_{SL}, E_{SR}
Find set of `Zero' strings Z_L and Z_L
Find N_L and N_R, number of pixels in E'_{SL} and E'_{SR}
Find max length in Z_L and Z_L
N_L and N_R> Th?
Max length > 1/3 * (n_B)
Solid
Dashed
Update temp chk flag $R(k) = 1$
Update temp chk flag $R(i) = 0$
Find Z = M- Σ L_{k-i}
Dashed
Z_{max} > Z > Z_{min}
Z << Z_{min}
Solid

Fig. 6. Flowchart summarizing steps in spatial and temporal continuity analysis for distinguishing lane markings

The two approaches of spatial and temporal continuity analysis for distinguishing lane markings between solid and dashed, are summarized in a flowchart in Fig. 6.

B. Detecting Double Linear Markings

In order to check whether the marking type is single or double, we first detect all the BLMs that are present in the region of interest. The output of the BLM detect algorithm is fed into a simple step to distinguish single versus double markings, based on the number of parallel BLMs that are detected. Spatial and temporal continuity checks are performed on the detected BLMs in order to classify lane markings into the following four categories - single solid, double solid, single dashed and double dashed. In cases of double lane markings, it is sufficient to perform the continuity checks on any one of the BLMs.

C. Detecting Zigzag Markings

A zigzag marking has multiple BLMs placed in a zigzag orientation. It can be observed that since the zigzag markings are drawn along the lane direction, the overall marking runs towards the vanishing point. However, the individual strokes of lines that make the zigzag marking do not converge into the vanishing point. This property of multiple BLM in the region of lane markings, that are oriented in directions that do not converge towards the vanishing point, are employed in the proposed zigzag marking technique.

One important observation regarding zigzag road markings is that they occur only at road boundaries or carriageway boundaries, not in between two lanes. Considering that the default marking at road boundaries or carraigeway boundaries is a solid lane marking (be it single or double), the most common scenario for a zigzag marking is when a solid marking precedes a zigzag, as shown in Fig. 7 (a). However, sometimes, due to a cross roads merging into the host lane, dashed markings (single or double) are seen replacing solid markings at those stretches, followed by zigzags. This scenario is illustrated in Fig. 7 (b). It is also possible that the zigzag marking starts from a region of no lane markings, as shown in Fig. 7 (c). Finally, in these different scenarios, the zigzag marking can be either single or double. An example of a double zigzag marking is shown in in Fig. 7 (d).

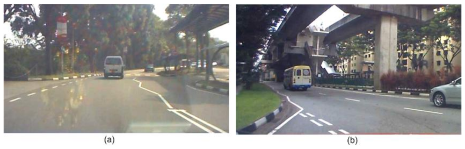

Fig. 7. Possible zigzag marking scenarios (a) Most common with solid marking preceding zigzag (b) Case with dashed marking at cross road preceding zigzag

Deriving trigger conditions for zigzag marking detection: When the recent history shows the presence of a dashed or solid lane marking, single or double, but the current frame does not detect the same marking in the expected direction, then it could be due to one of the three cases below:

1) The most likely case is that the host vehicle is in a zone of dashed linear markings, or algorithm failed to detect the edges due to poor illumination etc. The presence of dashed lane markings can be confirmed by the proposed method.
2) If dashed markings are not detected, then the other possibility is that the host vehicle has entered a zone of zigzag markings. In this case, the angular orientation of the BLMs forming the zigzag would be different from the angle of the expected lane markings (as per history), and hence the detection shows a negative for the marking in the current frame.
3) If it is neither the case of zigzags, nor the case of dashed lines, then it is possible that the host vehicle has entered a zone of no lane markings.

Algorithm 1 summarises the proposed zigzag detection algorithm, including the trigger condition. Consider k to be the index of the current image frame; inputs to the algorithm are the current frame, I_k, and detection history of past N frames. $R(k)$ is the binary variable introduced in Sec. III-A.2 to store the detection outcome of the k^{th} frame. When this is a 0 and when the history shows a dashed marking, then spatial continuity check is done to confirm the presence of dashed markings. If not, the frame is checked for presence of zigzag marking.

Algorithm 1 Algorithm for Zigzag marking detection

```
Input: current frame $I_k$, history of past $N$ frames
if $R_k == 0$ & no dashed linear marking found then
    start zigzag detection
    $\rho_k^l = 1/N \left[\sum_{i=1}^{N} \rho_{k-i}^l\right]$, $\theta_k^l = 1/N \left[\sum_{i=1}^{N} \theta_{k-i}^l\right]$
    $\rho_k^r = 1/N \left[\sum_{i=1}^{N} \rho_{k-i}^r\right]$, $\theta_k^r = 1/N \left[\sum_{i=1}^{N} \theta_{k-i}^r\right]$
    $\begin{bmatrix} cos\theta_k^l & sin\theta_k^l \\ cos\theta_k^r & sin\theta_k^r \end{bmatrix}^{-1} \begin{bmatrix} vpx \\ vpy \end{bmatrix} = \begin{bmatrix} \rho_k^l \\ \rho_k^r \end{bmatrix}$
    $M$:create mask bound by lines $l_1(\rho_k + \Delta\rho_k, \theta_k)$ and $l_2(\rho_k - \Delta\rho_k, \theta_k)$
    $I_M := I_k \otimes M$
    $\{l_1(\rho_1, \theta_1), \ldots, l_P(\rho_P, \theta_P)\} = blmDetect(I_m, \theta_k \pm \Delta\theta)$
    if $(P \geq 2)$ then
        for $i := 1$ to $P$ do
            $vpcheck = ptLine(\rho_i, \theta_i, vp_x, vp_y)$
            if $vpcheck == 0$ then
                z = z+1
            end if
        end for
        if $z \geq 1$ then
            Output: $zzDetect := 1$
        else
            Output: $zzDetect := 0$
        end if
    end if
end if
```

The region of interest for finding the zigzag marking is derived based on the host lane boundaries, considering that zigzag markings are drawn in the region of lane boundaries, along the road. From history, the expected lane boundaries on the left $((\rho_k^l, \theta_k^l))$, and right $((\rho_k^l, \theta_k^l))$ of the host lane are inferred from history by an averaging operation of the (ρ, θ) values of the past N frames. Next, a mask is created around the expected lane boundary position, by considering a parallelogram bound by lines $l_1(\rho_k + \Delta\rho_k, \theta_k)$ and $l_2(\rho_k - \Delta\rho_k, \theta_k)$. This is illustrated in Fig. 8, with the RoI represented by the shaded parallelogram.

Once the RoI is set, BLM detection algorithm described in Sec. II is applied on the image within the RoI, for a wider angle range. In terms of the function `blmdetect()`, the input parameter $\delta\theta$ is set to a large value to accommodate linear segments of the zigzag, that have angles of orientation that are more spread as compared to straight lane markings. The output of the `blmdetect()` function are subject to the condition that the lines do not align towards the vanishing point. As per the vanishing point criteria described above, although the overall zigzag marking is drawn in the general direction towards the vanishing point, the individual BLMs that constitute the zigzag are not aligned towards the vanishing point. If atleast two such BLMs are detected, then the presence of the zigzag marking is confirmed.

IV. Modular Architecture for Detecting Lane Types

It was shown in the previous sections of this paper that the BLM detect module was deployed to detect var-

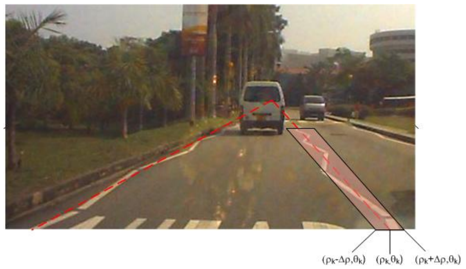
$(\rho_k-\Delta\rho,\theta_k)$
(ρ_k,θ_k)
$(\rho_k+\Delta\rho,\theta_k)$

Fig. 8. Region of interest for zigzag detection

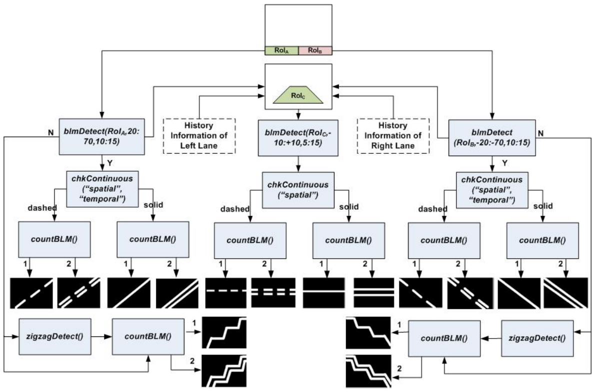
RoI_A
RoI_B
RoI_C
History Information of Left Lane
History Information of Right Lane
blmDetect(RoI_A,20:70,10:15)
blmDetect(RoI_C,-10:+10,5:15)
blmDetect (RoI_B,-20:-70,10:15)
N
Y
chkContinuous ("spatial", "temporal")
chkContinuous ("spatial")
chkContinuous ("spatial", "temporal")
dashed
solid
countBLM()
1
2
zigzagDetect()
countBLM()
zigzagDetect()
countBLM()

Fig. 9. Overall block architecture for detection of simple and complex linear road markings

ious kinds of linear markings. This is summarized as a modular architecture in Fig. 9, with BLM detection being the core computational module, together with supporting modules, namely `countBLM()`, `zigzagDetect()` and `chkContinuous()`. `countBLM()` analyses the output set *S* of `blmDetect()`, and generates a binary output *n* that can either take the values of '0' or '1' depending on whether the markings are found to be 'single' or 'double'. This is done by simply checking the number of parallel BLMs detected that satisfy the properties of lane markings. Based on the algorithm that was presented for zigzag marking detection in 1, function `zigzagDetect()` outputs a binary value *z*, depending on whether a zigzag is detected on not. Function `chkContinuous()` takes the parameters 'spatial' or 'temporal', and determines whether the marking is solid or dashed. The horizontal lines, such as stop lines can also be detected by using the BLM detect module after setting appropriate parameters. Although horizontal markings are not lane markings, they are included in Fig. 9 for the sake of completeness.

V. Results and Discussion

The proposed approach to distinguish lane markings using temporal continuity check is applied on the 20 test sequences from the dataset in [8]. The speed of the vehicle in the test videos considered for this experimentation was in the range of 10m/s to 25m/s. Given that the frame rate in the camera set up used was 30 frames per second, and given that the gaps in the dashed markings are around 2m to 3m long on Singapore roads, a maximum of around 10 frames is required to fully capture the gap. Accordingly, M and Z_{max} in 4 are set to 20 and 10 respectively. Fig. 5 shows an illustration of the *R* array (explained in III-A.2) for lane markings on either side of the host lane.

Fig. 10 shows sample results of the temporal continuity analysis for some of the sequences in the dataset. The temporal arrays for the left and right lane markings are shown along with the classification outcome marked as 'Dashed' or 'Solid'. It can be seen that the arrays conclusively show an alternating waveform pattern in the case of dashed markings, but is a constant waveform set to 1 in the case of solid markings.

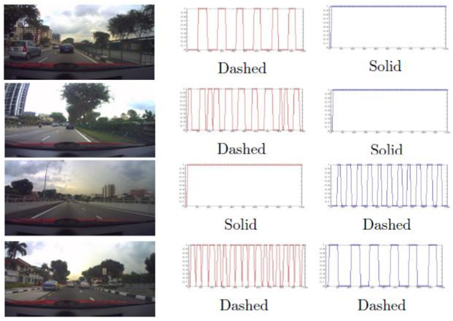

Fig. 10. Sample results for distinguishing solid versus dashed markings

Further, dashed markings can have varying gaps between every two dashes to indicate the degree of allowance given to vehicles to change lanes. Closely spaced dashed marking indicate a more restricted lane change, and sparse dashes indicate a free movement of vehicles between lanes. The temporal array generated through this analysis can also throw light on whether the dashed markings are closely or sparsely spaced. The troughs in the waveform correspond to the gaps, and hence the length of troughs in the alternating waveform is studied more closely. It was observed that the length of the trough is dependent on two parameters: (i) the speed of the vehicle and (ii) the length of gaps between the dashes. For the same lane stretch, if the vehicle speed is increased, then it can be expected that the frequency of the waveform increases, thereby decreasing the trough length, and vice versa. Also, for a given speed, if the troughs are longer, then it can be inferred that the gaps are relatively larger. This can be seen in the fourth image sequence in Fig. 10. The lane marking on the left host lane are closely spaced, while the markings on the right are sparse. This is reflected in the corresponding waveforms. Since the vehicle speed is fixed for the analysis of the left and right lane markings, the relative gaps can be easily inferred. In order to infer the exact length of gaps, this analysis can be extended further to an automated model that incorporates the actual vehicle speed as an input parameter.

The proposed algorithm to classify lane markings as single-dashed, single-solid, double-dashed and double-solid was applied on 40 image sequences that were selected from the dataset in [8], such that they represent all the four types of lane markings. It was seen that the algorithm leads to 95 % accuracy in detecting the type of lane marking. The misdetections were primarily in cases with road boundaries and pavement edges contributing to multiple edges that sometimes get detected as double linear markings.

One of the same sequences is shown in Fig. 11, which shows an image sequence with a double solid lane marking on the left and a single-dashed lane marking on the right. The classification output has been embedded as inset images on the top-left and top-right corners of the images in the sequence, for validation purposes.

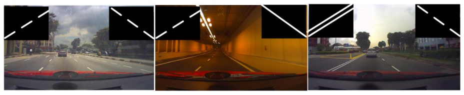

Fig. 11. Classification results of single/double and solid/dashed

In order to evaluate the proposed zigzag detection algorithm, 35 image sequences containing zigzag markings and 35 of them with straight lane markings on either side were chosen. Some sequences had zigzags on both sides, while some had it on one side. It was observed that the all sequences with zigzag markings were detected successfully using the method. Also, the negative samples without the zigzag marking were shown to correctly output a negative. Sample images corresponding to 8 of these different sequences are shown in Fig. 12.

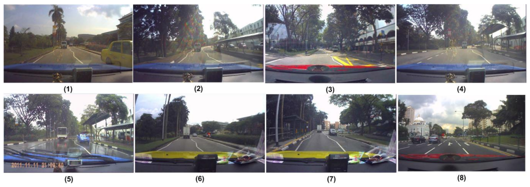

Fig. 12. Images on which Algo. 1 is applied for zigzag detection.

The results for these images are shown in Table I. It was seen that the proposed algorithm is able to detect the presence of zigzag marking. For every image in Fig. 12, the vanishing point, the (ρ, θ) pairs for left and right edges of the left and right host lanes are shown in Table I. Furthermore, the ρ value that is obtained by using the vanishing point and the θ values of each line segment is also shown. This value is shown under ρ^L_{vpt} and ρ^R_{vpt} for the left and right lanes respectively. These values are used to find the flag that is set to $yes(Y)$ if the lane marking passes through vanishing point and vice versa. The flags are denoted as f^L_{Vpt} and f^R_{Vpt} for left and right lanes respectively. It can be seen from Table I that in the cases where there is a zigzag marking, multiple linear edges are detected by the proposed technique and most of the linear edges do not pass through the vanishing point. In the case of non-zigzag marking cases, only one linear edge is detected which passes through the vanishing point. This is consistently shown in all the test cases that were considered.

VI. Conclusion

In this paper, we have proposed a modular approach for detecting and distinguishing various kinds of lane markings including dashed versus solid, single versus double and zigzag markings. The core computational module was identified to be BLM detect module. A two-pronged approach to distinguish solid and dashed lane markings was proposed which involves performing spatial and temporal continuity analyses on the BLMs detected. Further, the BLM detect module was deployed to detect double lane markings. Finally, a novel algorithm to detect zigzag markings was presented. A modular architecture was presented with BLM detect and other supporting modules. The proposed techniques were validated on road markings dataset [8] and was shown to yield high detection rates.

TABLE I

Results of Zigzag Detection

#	V_{pt}	Left Lane			Right Lane		
		E_{left}/E_{right}	ρ^R_{vpt}	f^L_{Vpt}	E_{left}/E_{right}	ρ^R_{vpt}	f^R_{Vpt}
1	(612, 546)	(90,-38)/(140,-34)	51	N	(765,33)/(771,34)	794	N
		(694,-90)/(-654,-85)	-612	N	(758,22)/(750,18)	736	N
		(339,-20)/(232,-27)	296	N	(771,51)/(791,43)	818	N
		(894,48)/(896,48)	820	N	(739,19)/(751,23)	720	N
2	(635, 562)	(318,-23)/(330,-22)	264	N	(844,37)/(850,36)	833	N
		(-348,-62)/(-369,-63)	-302	N	(763,63)/(748,66)	819	N
		(280,-24)/(305,-22)	248	N	(789,21)/(789,21)	757	N
		(825,53)/(722,74)	844	N			
3	(552, 581)	(311,-27)/(394,-21)	262	N			
		(-183,-56)/(-181,-56)	-145	N			
		(233,-30)/(281,-27)	214	N			
4	(681, 546)	(-65,-41)/(-21,-38)	-43	N	(749,19)/(756,21)	740	Y
		(282,-22)/(303,-21)	243	N			
		(-486,-68)/(-529,-71)	-440	N			
		(280,-21)/(239,-24)	254	N			
5	(624, 559)	(-132,-47)/(5,-39)	-87	N	(731,18)/(731,17)	724	Y
		(244,-27)/(248,-26)	213	N			
		(-632,-83)/(-554,-77)	-563	N			
		(289,-23)/(203,-28)	267	N			
6	(515, 536)	(170,-36)/(188,-35)	123	N	(730,22)/(734,22)	692	N
		(-505,-81)/(-352,-69)	-431	N	(-719,348)/(718,38)	734	N
		(272,-27)/(258,-28)	235	N	(712,22)/(711,21)	693	N
		(820,45)/(740,69)	743	N	(724,42)/(710,49)	743	N
7	(521, 505)	(-89,-47)/(-32,-43)	-43	N	(669,49)/(657,53)	723	N
		(291,-23)/(290,-23)	254	N	(705,23)/(712,26)	670	N
		(-340,-67)/(-384,-70)	-287	N	(497,83)/(-329,-79)	573	N
		(302,-21)/(310,-20)	275	N	(678,20)/(693,23)	653	N
8	(545, 574)	(31,-44)/(53,-42)	26	Y	(772,31)/(776,31)	772	Y

References

[1] D. L. Aharon Bar Hillel, Ronen Lerner and G. Raz, "Recent progress in road and lane detection: a survey," in *Mach. Vis. and App., Springer-Verlag*, feb 2012.

[2] J. Collado, C. Hilario, A. de la Escalera, and J. Armingol, "Detection and classification of road lanes with a frequency analysis," in *IEEE Intel. Veh. Symp.*, june 2005, pp. 78 – 83.

[3] T. Gavrilovic, J. Ninot, and L. Smadja, "Frequency filtering and connected components characterization for zebra-crossing and hatched markings detection," 2010, p. A:43.

[4] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Trans. on Intel. Trans. Sys.*, vol. 7, no. 1, pp. 20 –37, march 2006.

[5] H. B.-A. Nadra Ben Romdhane, Mohamed Hammami, "A comparative study of vision-based lane detection methods," in *Advanced Concepts for Intel. Vis. Sys., Lecture Notes in Comp. Sci.*, vol. 6915, 2011, pp. 46–57.

[6] V. Popescu, R. Danescu, and S. Nedevschi, "On-road position estimation by probabilistic integration of visual cues," in *IEEE Intel. Veh. Symp.*, june 2012, pp. 583 –589.

[7] R. Satzoda, S. Suchitra, and T. Srikanthan, "Robust extraction of lane markings using gradient angle histograms and directional signed edges," in *IEEE Intel. Veh. Symp. (IV)*, june 2012, pp. 754 –759.

[8] S. Suchitra and R. Satzoda, "Chipes road markings dataset," Online: https://docs.google.com/folder/d/0B6eAY2O8msWGRm5mODBXckVPTXc/edit, [Online; accessed 25-Jan-2013].

[9] S. Vacek, C. Schimmel, and R. Dillmann, "Road-marking analysis for autonomous vehicle guidance," in *Eur. Conf. on Mobile Rob.* , 2007, pp. 1–6.

[10] R. Satzoda, S. Suchitra, and T. Srikanthan, "Gradient Angle Histograms for Efficient Linear Hough Transform," in *IEEE Int. Conf. on Image Proc. (ICIP)*, nov 2009, pp. 3273 - 3276.