Zebra Crosswalks Detection Based on Geometric Features

Buyi Zhang, Chunan Ye and Yutian Chen
Instructor: Jason Corso
EECS 504 Foundations of Computer Vision
University of Michigan

I. INTRODUCTION

Zebra crosswalks are important visual signs in urban areas. It would help visually impaired people and potentially mobile robots to safely go across the streets if there is an application that can inform them when a zebra crosswalk is in front of them. It would be better that the locations of the zebra crosswalks can be detected and highlighted to them. Therefore, we propose an algorithm that can detect zebra crosswalks under limited occlusion and normal light conditions using images from a pedestrian's point of view.

In this project, we use the definition that a stripelet [1] is one of the rectangles filled with bright color, usually white or yellow, that forms the zebra crosswalk. We assume that the stripelets of the zebra crosswalks are seen from the side of the road. In other words, they are more likely to be horizontal than vertical. We assume that the occlusion is limited so that for each given stripelet, a sufficiently large portion of it is still visible. For normal light condition, we assume that the light condition is weak and uniform so that it doesn't affect the color of the stripelets.

Our algorithm is a complete implementation with variations of the algorithm proposed by J. Coughlan and H. Shen [1]. It is a graphical model-based approach that uses the *geometric features* of the zebra crosswalks. We detect the top and bottom boundaries of the stripelets as line segments by doing edge detection. We then group those line segments to form candidates of potential stripelets using their color, orientation and perspective property. We consider each stripelet as a node of a graph and solve the segmentation problem of assigning each node either as true zebra crosswalk stripelets or false zebra crosswalk stripelets. The segmentation is solved using belief propagation on the graph of stripelets, which is assumed to be an Markov Random Field.

Our implementation shows promising results on the test samples we selected online. The efficiency of the algorithm is high and indicates the algorithm's good potential to be turned into real-time applications.

II. RELATED WORK

A. Line segment detection

The most recent edge detection technique is based on gradient. To obtain characterization of line segments from edge points, we see two approaches being proposed. One approach is based on Hough transform which does a majority vote in the parameter space [4]. Another approach is a greedy approach based on local search [2]. Parameters are then calculated from each detected regions.

We use the second one in our algorithm. One advantage of it is that it works better in estimating multiple line segments. Another advantage is that it makes good use of the property of the line segments and makes the estimation very efficient.

B. Graphical model for figure-ground segmentation

For the zebra crosswalk detection problem, J. Coughlan and H. Shen propose a graphical model with stripelets as nodes. They further assume that this graph is a Markov Random Field, and segmentation problem is solved using either graph cut or belief propagation [1].

We use the belief propagation approach in our implementation. Based on this method, we extended the connectivity of the graph model and reduced the effect of noisy line segment estimations on the final segmentation results.

III. METHODS

A. Line segment detector

Our line segment detector is an implementation of the algorithm proposed by R. Grompone von Gioi, etc [2]. The gradient of a pixel location (i, j) is defined as

$$g_x(i,j) = \frac{I(i,j+1) + I(i+1,j+1) - I(i,j) - I(i+1,j)}{2} \tag{1}$$

$$g_y(i,j) = \frac{I(i+1,j) + I(i+1,j+1) - I(i,j) - I(i,j+1)}{2} \tag{2}$$

where I is the original image, g_x, g_y are the horizontal gradient and vertical gradient. We can then compute the level-line angle and gradient magnitude of each pixel by

$$\theta(i,j) = \arctan 2(-g_y(i,j), g_x(i,j)) \tag{3}$$

$$G(i,j) = \sqrt{g_x(i,j)^2 + g_y(i,j)^2} \tag{4}$$

We are more interested in horizontal edge points than vertical edge points. For every edge point, a threshold test is done on both its gradient magnitude G and the absolute value of its vertical gradient g_y. The thresholds are set loosely so that only unwanted edge points are removed.

We sort the points by their gradient magnitude. Starting from points with larger gradient magnitudes, we do a greedy local search for connected points with sufficiently small difference to the level-line angle. The level-line angle used to perform the search is initially set as the level-line angle of the starting point. It is then updated using the level-line angle of added points. The update rule is defined as

$$\theta = \arctan\left(\frac{\sum_j \sin(\theta_j)}{\sum_j \cos(\theta_j)}\right) \tag{5}$$

A center of the the region (c_x, c_y) is computed as

$$c_x = \frac{\sum_j G(j) \cdot x(j)}{\sum_j G(j)} \tag{6}$$

$$c_y = \frac{\sum_j G(j) \cdot y(j)}{\sum_j G(j)} \tag{7}$$

The main angle can be computed as the angle of the eigenvector associated with the smallest eigenvalue of the matrix

$$M = \begin{pmatrix} m^{xx} & m^{xy} \\ m^{xy} & m^{yy} \end{pmatrix}$$

where

$$m^{xx} = \frac{\sum_j G(j) \cdot (x(j) - c_x)^2}{\sum_j G(j)}$$

$$m^{yy} = \frac{\sum_j G(j) \cdot (y(j) - c_y)^2}{\sum_j G(j)}$$

$$m^{xy} = \frac{\sum_j G(j) \cdot (x(j) - c_x)(y(j) - c_y)}{\sum_j G(j)}$$

The main angle plus $\pi/2$ is used as θ of the angle of the normal vector of the line segment. The parameter ρ of the line segment by solving

$$c_x \cos(\theta) + c_y \sin(\theta) = \rho$$

x_{min}, x_{max} are recorded as the minimum x coordinate and maximum x coordinate achieved by the points in this region during the search. The vector $[\rho, \theta, x_{min}, x_{max}]$ gives the parametric representation of our line segments.

Additionally, we compute the *polarity* of each line segments by examining whether the gradient of this region sums to positive or negative values. According to our implementation, positive line segments will always be bottom boundaries of the stripelets and negative line segments will always be top boundaries of the stripelets.

Since we only care about the top and bottom boundaries of the stripelets, we remove the line segments whose slope is larger than $\frac{1}{8}\pi$.

B. Line segment grouping

Due to the noise in the image and the distortion, the boundary of stripelets may be broken into multiple segments. In order to extract a complete stripelet, we group those segments into one.

We sort the line segments by their length in descending order. Then we iterate over all line segments and check its similarity with the rest of them based on the following criterion:

- The angle between two line segments is less than $\frac{1}{32}\pi$.
- The distance from the center of the shorter segment to the line of the longer segment is less than 5.

If the condition is satisfied, we group the related segments into one. We take out all the points belong to those line segments in the original image and then use the Hough transform to estimate the new line segment.

C. Stripelets grouping

For all the remaining line segments, a matching relationship is computed to produce candidates of potential stripelets. We iterate over all the pairs of positive line segments and negative line segments and only accept pairs with the following properties.

- The negative line segment is above the positive line segments
- The two line segments are relatively parallel
- the vertical width of the candidate stripelet is within [5,40] pixels
- The two line segments have sufficiently large overlap in length
- The bright pixels occupies at least 65% of the area formed by the candidate stripelet

The first 4 properties are ideas proposed by J. Coughlan and H. Shen [1]. We added the 5th property as an additional mean to remove noisy line segments.

D. Graphical model and belief propagation

The stripelets generated in the previous step mutually form the graph we needed to do the segmentation. A stripelet is characterized by its centroid's coordinate (x, y), vertical width w, parameters of the two line segments and its range on x axis. The graph is assumed to be a Markov Random Field which only depends on unary cues and binary cues to determine the segmentation. A connection of two stripelets is rejected if the higher one has longer vertical width than the lower one, which contradicts the perspective property. A connection is also rejected if the distance between the centroids of the two stripelets are too long. If a connection is not rejected, a numerical value of binary potential indicating "how much" the two stripelets are connected is then computed using the characterization parameters. The unary cue uses the perspective property that the true zebra crosswalk stripelets' vertical width linearly decays as we increase their centroid's y coordinate.

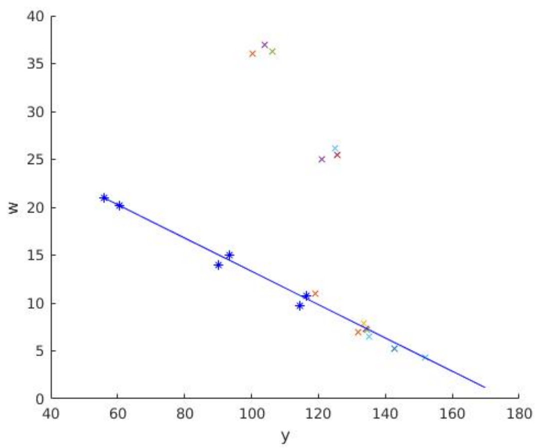

Figure 1: The scatter plot of the (y,w) points of stripelets and the envelope

This idea is proposed by J. Coughlan and H. Shen [1]. We interpreted this idea as a fundamental assumption of our input image. The fundamental assumption is that the lowest stripelet found is almost sure to be a true zebra crosswalk stripelet. Therefore, passing through the stripelet's corresponding point in the $y-w$ space, we estimate a linear envelop that is likely to be close to the corresponding points of all the true zebra crosswalk stripelets. For any given stripelet, the distance from its corresponding point in the $y-w$ space to this envelop is used to compute the unary cue of this stripelet, representing its unary potential to be a true zebra crosswalk stripelet. The unary function is defined as

$$\phi_i(i=1) = 2\min(1, L(1-\frac{E}{\hat{w}}))$$

where $i=1$ means considering stripelet i being a true zebra crosswalk stripelet. L is defined as the square root of the geometric mean of the length of the top line segment and the length of the bottom line segment [1]. E is defined as the distance from the point to the envelop. $\hat{w}$ is defined as the w value along the envelop with same y of the point. We corrected the error made in J. Coughlan and H. Shen's statement to correctly reflect the idea of limiting the unary potential to emphasize the binary cue.

The binary cue is based on the cross ratio test between the two stripelets. True zebra crosswalk stripelets are of equal width and are placed with equal width intervals. Regardless of the perspective, the cross ratio of any two neighboring stripelets are invariant and theoretically $1/4$. We combine this fact and another fact that the cross ratio is invariant to the intersection line used to compute this cross ratio. Two "probing" lines are generated for each two stripelets and the cross ratios are computed as r_1 and r_2. The binary cue is embedded in the cross ratio error measure [1]

$$R = (|r_1 - 1/4| + |r_2 - 1/4|)/2 + 2 * |r_1 - r_2|$$

$$\phi_{ij}(i=1, j=1) = 100e^{-10R}$$

In addition, we connect stripelets with similar y coordinates and similar length. This is intended to deal with the case when imperfect line segment estimation produces "broken" stripelets that partitions a complete true zebra crosswalk stripelet. Figure 4 of our test sample shows many such "broken" line segments from the estimation.

Then, for belief propagation a message update algorithm is run on this pairwise MRF [3].

$$m_{ij}^{new}(j) = \sum_i \phi_{ij}(i,j)\phi_i(i) \prod_{k \in Nbd(i)\setminus j} m_{ki}^{old}(i)$$

To update message from i to j, obtain the product of all messages $m_{ki}^{old}(i)$ flowing into i (except for the message from j), binary potentials from i to j and unary potentials of node i.

Once the message converge, we use belief read-out equation to get the final belief of corresponding node $i=1$.

IV. Results

We selected two successful samples to show the whole process. Note that the red line segments are of negative polarity and the green line segments are of the positive polarity.

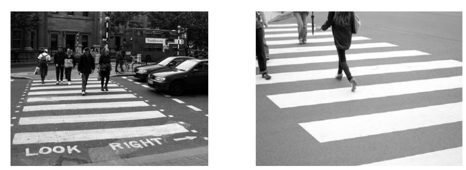

Figure 2: The original image

Figure 3: The line segments extraction

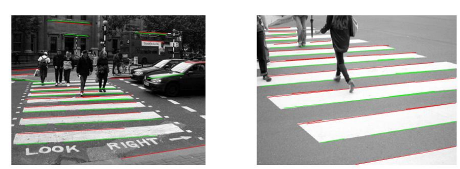

Figure 4: The line segments grouping

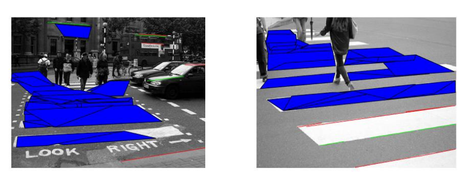

Figure 5: The stripelet grouping

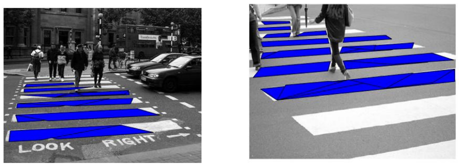

Figure 6: The final result

Some samples demonstrate inaccurate segmentation of the stripelet.

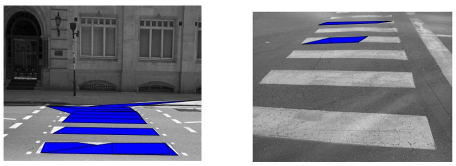

Figure 7: The samples of inaccurate segmentation

Empirically, our algorithm has promising performance on most of samples we have tested so far.

Additionally, we tested our algorithm on images of roads without zebra crosswalk from a pedestrian's point of view.

Figure 8: False positive case of our algorithm

The worst case runtime for the algorithm on all our test samples is within 3 seconds.

V. Discussion

Through this project, we verified the feasibility of using geometric features of the zebra crosswalks in its detection problem [1]. The efficiency of the algorithm is sufficiently good to produce real time response so that real time applications can be made upon it. Our algorithm is limited by the light condition and integrity of stripelets. For the left image of the inaccurate samples given above, the shading effect on the farther stripelets causes the inaccurate detection. For the right image of the inaccurate samples, both shading and color fading of the closer stripelets causes the failure of detection. The algorithm is not sensitive to resolution. All images are re-sized to 300x400, which is small compared to the highest resolution cameras can achieve nowadays.

Although the algorithm works well on true positive cases under our assumptions, it has unpredictable result on false positive cases. In false positive cases, our fundamental assumption that the stripelets with low y coordinates are likely to be true zebra crosswalk stripelets is violated. However, our algorithm still enforces the estimation of such an envelope and uses that envelope to do the other steps, which is unreasonable and will cause unpredictable results. Ideally speaking, the binary cue of the cross ratio test will be able to reject them since they are unlikely to hold that property. We believe that eventually, for the decision of the existence of zebra crosswalks, additional care needs to be taken to deal with the false positive cases.

The performance of our algorithm is consistent with both true positive cases and false positive cases. Taking N to be the total number of pixels in the re-sized image (currently set as 300x400), all the subroutines of the algorithm have linear worst case complexity with respect to N. This ensures that the algorithm's worst case complexity is $O(N)$ which won't grows too fast if higher resolution is attempted to achieve more accurate result.

VI. Acknowledgment

All test samples are collected from Google image search engine. The latex template is provided by Overleaf [5].

References

[1] J. M. Coughlan, Huiying Shen, A Fast Algorithm for Finding Crosswalks using Figure-Ground Segmentation; 2nd Workshop on Applications of Computer Vision, in conjunction with European Conference on Computer Vision(ECCV 2006).

[2] R. Grompone von Gioi, J. M. Morel and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.32, no.4, pp. 722-732, April 2010.

[3] J. Coughlan, A Tutorial Introduction to Belief Propagation, Aug. 2009, slides. Retrieved from online: http://computerrobotvision.org/2009/tutorial_day/crv09_belief_propagation_v2.pdf

[4] X. Zhang, H. Burkhardt, Grouping edge points into line segments by sequential Hough transformation, in: ICPR, vol. 3, 2000, pp. 676-679

[5] H. Kwakernaak and P. Misra, Preparation of Papers for IEEE Sponsored Conferences and Symposia, https://www.overleaf.com/latex/templates/preparation-of-papers-for-ieee-sponsored-conferences-and-symposia/zfnqfzzzxghk#.WsqMaK0sEUE