

A Geometric Framework for Rectangular Shape Detection

Qi Li

Abstract—Rectangular shape detection has a wide range of applications, such as license plate detection, vehicle detection, and building detection. In this paper, we propose a geometric framework for rectangular shape detection based on the channel-scale space of RGB images. The framework consists of algorithms developed to address three issues of a candidate shape (i.e., a connected component of edge points), including: 1) outliers; 2) open shape; and 3) fragmentation. Furthermore, we propose an interestingness measure for rectangular shapes by integrating imbalanced points (one type of interest points). Our experimental study shows the promise of the proposed framework.

Index Terms—Outliers, open shape, fragmentation, interest point detection.

I. INTRODUCTION

RECTANGULAR shape detection has a wide range of applications, such as license plate detection [7], [10], vehicle detection [22], [27], and building detection [6], [26]. One strategy for rectangular shape detection is based on analyzing, splitting, and merging line segments [16], [18], [25]. The method proposed by Lin and Nevatia [16] starts from line segments that are detected by applying the USC LINEAR linking method [23] to Canny edge images [3]. L-junctions and T-junctions are then detected from these line segments to break them into smaller segments. A colinearization process is applied to group small segments into longer ones to partially address the fragmentation issue in Canny edge images. The method proposed by Tao et al. [25] first extracts straight line segments by applying contour tracking and straight-line splitting techniques to an edge image. A graph is then introduced to analyze and merge straight line segments to construct rectangles. Liu et. al [18] proposed a MRF model based method. This method makes use of color information to compute an edge map. Line segments are merged in order to reduce the time complexity of MRF computation. Merging and splitting criteria used in the above-mentioned strategy have to engage heuristics on the proximity of line segments and different junction or bifurcation structures, which limits its adaptability.

The Hough transform (or its continuous version), as a voting strategy, provides a more unified and adaptive framework to

Manuscript received February 27, 2013; revised January 10, 2014, March 17, 2014, and May 11, 2014; accepted May 12, 2014. Date of publication July 25, 2014; date of current version August 18, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bulent Sankur.

The author is with the Department of Computer Science, Western Kentucky University, Bowling Green, KY 42101 USA (e-mail: qi.li@wku.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2343456



Fig. 1. Windowed Hough transform [9] is applied to a license plate image with dimension 240×320 . The method detects a number of small rectangles (colored by red), as shown in (b), when the range of radius is set to the interval $[4, 14]$ (a default setting used in the publicly available software package). The method did not detect any rectangle when the range of radius is set to the interval $[14, 64]$ (used in the paper [9]), or larger ranges, e.g., $[14, 100]$, and $[14, 200]$. (a) Input. (b) Detected rectangles.

analyze edges of an input image [1], [6], [9], [18], [29]. Zhu et al. [29] proposed an efficient Hough transform method with an assumption that the dimensions of rectangular objects in an image are identical and known. Jung and Schram [9] proposed a so-called windowed Hough transform that first applies a Hough transform to detect all line candidates, and then detects rectangles by enforcing rectangular properties (i.e., perpendicular relation of two intersecting lines) to the candidates of lines. The method contains several sensitive parameters to tune in, which limits its applications. For example, the method uses a ring with a pre-defined internal radius D_{\min} and external radius D_{\max} as a search region, where the internal and external radii are approximately equal to the smallest and largest diagonal of any possible rectangles, respectively. Fig. 1 illustrates an application of windowed Hough transform on a license plate image. Fig. 1 (a) shows the input image, and (b) shows detected rectangles (colored by red) on top of an edge image. Note that the edge image is relatively clean. The range of radius $[D_{\min}, D_{\max}]$ is set to the interval $[4, 14]$, one setting suggested by Jung and Schram in their software package. We also observed that the method did not detect any rectangle when the range of radius is set to the interval $[14, 64]$ (used in previous works [9]), or larger ranges, e.g., $[14, 100]$ and $[14, 200]$. Observing the problem that the Hough transform is sensitive to image noise, clutter and light conditions, Bhaskar et al. [1] proposed a Radon transform (a continuous version of Hough transform) based method that generalized Jung and Schram's method, and obtained clear improvement. However, the method of Bhaskar et. al also contains various parameters to tune in, which motivated the authors to consider learning methods in their future work.

In this paper, we propose a geometric framework for rectangular shape detection using candidate shapes in RGB channels and scale spaces. We call a connected component

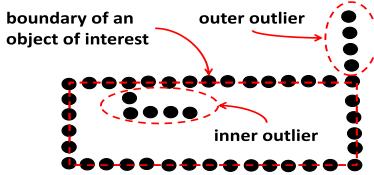


Fig. 2. Two types of outliers with respect to the boundary of an object of interest. Outer outliers are image points outside the boundary, and inner outliers are image points inside the boundary.

(in terms of 8-connectivity) in an edge image a *candidate shape*. Note that a connected component may not have a pure line structure, e.g., a point in a component may have more than two neighbors. In other words, a candidate shape is basically a set of edge points that have the connectivity property but no ordering property. To our best knowledge, no existing methods in the literature of rectangle detection rely on candidate shapes in channel-scale spaces. In the context of salient region detection, candidate shapes in channel-scale spaces have close relationship to scale-space primal sketch proposed by Lindeberg [17], i.e., both of them are based on scale-space and low-level image structures. However, image structures they used are different. Image structures used in a primal sketch are Gaussian blob regions, while image structures used in candidate shapes are edges.

There are various difficulties in the application of candidate shapes to detect objects/regions of interest. These include: i) *outliers*, ii) *open shape*, and iii) *fragmentation*. Specifically, a candidate shape may contain image points that do not lie on the boundary of an object of interest. For convenience, we call these points *outliers* with respect to a boundary. (Note that a boundary can be formulated as a statistical model.) In this paper, we further refine outliers as two types based on their spatial relationship with a boundary: i) outer outliers, i.e., image points outside the boundary, and ii) inner outliers, i.e., image points inside the boundary, as illustrated in Fig. 2. A candidate shape may be open, i.e., the set of edge points does not form a closed shape. If an open shape has significant amount (e.g., $\geq 50\%$) of overlapping with the boundary of a simple shape object (such as a license plate), it may be possible to design an algorithm to recover the shape, as shown in Section III. However, if an open candidate shape only has a small amount of overlapping with the boundary of an object of interest, i.e., the candidate shape is a small fragment, it will be unreasonable to recover the target shape based on the single fragment.

Our first strategy to address the above-mentioned three difficulties of candidate shapes is to introduce the channel-scale space of RGB images, aiming to enhance the chance of detecting good candidates of rectangles, where a channel can be red, green, blue, or gray. However, the application of a channel-scale space can not overcome the three difficulties thoroughly, due to image noises, illumination variation, and cluttered scenes. Moreover, small scales tend to produce fragmented candidate shapes, while large scales tend to produce candidate shapes with outliers. Thus, we further address the three difficulties individually. Specifically, we propose: i) an algorithm to remove outer outliers in a rectangular candidate

shape using spatial likelihood, ii) algorithms to detect four vertex candidates that implicitly addresses the possibility of open rectangular candidate shape, and iii) algorithms to check if a candidate shape is open or not and if two rectangular candidate shapes are complementary or not to synthesize candidate shapes. The synthesized candidate shape can address the fragmentation difficulty by augmenting the space of rectangle candidates effectively.

Many rectangular shapes detected by the above methods may be false positive. In order to quantify the likelihood of a rectangular shape being a true positive, we introduce an interestingness measure of a rectangular shape by the integration of imbalanced points [11], [13], [14]. The intuition on imbalance is to distinguish different nature of interest points and edge points. (More details on imbalanced points will be reviewed in Section V.) In experiments, we present visual and quantitative evaluations of proposed algorithms.

As a summary, the main contributions of this paper include:

- the introduction of channel-scale space of RGB images,
- algorithms for rectangularity verification,
- algorithms to address fragmentation issues of candidate shapes, and
- an interestingness measure of rectangular candidate shapes by the integration of interest points.

The rest of this paper is organized as follows: Section II describes channel-scale spaces of an RGB image. Section III illustrates how to verify rectangular candidate shapes in the channel-scale space. In Section IV, we propose algorithms to address the fragmentation issue of candidate shapes. In Section V, we propose an interestingness measure of rectangles. The experimental study is presented in Section VI. Finally, a conclusion is presented in Section VII.

II. CANDIDATE SHAPES IN CHANNEL-SCALE SPACES

Given a grayscale image I , we denote $I_\sigma = I * G_\sigma(x, y)$ the convolution of I and the 2d Gaussian filter with a scale σ . Denote E_{I_σ} an edge image of I_σ . Assume that E_{I_σ} consists of a number of disjoint connected components, that is, $E_{I_\sigma} = \bigcup_i C_{i,\sigma}$ and $C_{i,\sigma} \cap C_{j,\sigma} = \emptyset, i \neq j$, where $\{C_{i,\sigma}\}$ is a set of connected components (in terms of 8-connectivity) in scale σ , and \emptyset indicates an empty set. Small connected components need to be removed in order to control the computation cost in the follow-up rectangularity verification. For convenience, a connected component C is called a *candidate shape* if its size (in the unit of pixel) is above a threshold t . The choice of t is adaptive to applications. For example, in the context of plate license detection and recognition, t can be relatively large, i.e., large enough to enclose 6 to 7 recognizable characters or digits. In our experiment, t is set to 50.

We denote S_σ a set of *candidate shapes* in the edge image E_{I_σ} , i.e.,

$$S_\sigma = \{C_{i,\sigma} : |C_{i,\sigma}| > t\}, \quad (1)$$

where $|C|$ denotes the size of a connected component C .

A RGB image includes multiple channels, e.g., red, green, blue, and gray, which are indexed by $r = 1, \dots, 4$, respectively. Given a RGB image, we extend a scale space $\{I_\sigma\}$ to

the channel-scale space $\{I_\sigma^r\}$. Similarly, we extend the notation $C_{i,\sigma}$ to $C_{i,\sigma}^r$, and the notation S_σ to S_σ^r . Candidate shapes of a RGB image is then defined as the union of candidate shapes in each channel and each scale, i.e.,

$$S = \bigcup_r \bigcup_\sigma S_\sigma^r. \quad (2)$$

The rationale behind the union strategy is to increase the chance of locating a clean edge of an object of interest under various appearance and imaging disturbance.

Note that a candidate shape of an image is a connected components of edge points that have not been spatially ordered. In the rest of the paper, we denote $P = \{p_i = (x_i, y_i)\}$ a candidate shape of an input image.

III. DETECTION OF RECTANGULAR CANDIDATE SHAPES

In this section, we first propose a method to remove outer outliers, then two methods to detect four vertex candidates in a candidate shape, and last a verification method of the rectangularity of a candidate shape based on the four vertex candidates.

A. Removal of Outer Outliers

Our basic strategy to remove outer outliers is to use a 1-d projection of a rotated candidate shape, where the rotation is applied to align the principal axis of the shape with the horizontal axis.

For simplicity of illustration, we first assume that the orientation of a candidate shape P is horizontal. (This assumption will be removed in the proposed algorithm.) We first construct a vertical projection of P .

$$h(j) = |\{p_i \in P | p_{i,y} = j\}|, \quad (3)$$

where $|\cdot|$ denotes the cardinality of a set. The projection $h(\cdot)$ aims to reveal a likelihood of “vertical” line segments if they exist. In the ideal case, there are two single-pixel-wide peaks in the projection function $h(\cdot)$, i.e., two peaks are associated with the y-location of vertical line segments. Fig. 3 illustrates this phenomenon. However, under perspective distortions, image noise or other factors, two peaks of a projection may be wider than a single pixel, which makes the detection of two local maxima not reliable. A popular method to address noisy and flat peaks is to smooth the projection h with a Gaussian kernel. However, this method contains a sensitive parameter, i.e., the size of Gaussian kernels. Note that a very small size may not be sufficient to smoothen flat “peaks”, while a very large size may bring serious biased boundary estimation.

We propose an alternative method without any parameter as follows. We first approximate the absolute derivative of $h(\cdot)$ by the following operation:

$$h'(j) = |h(j+1) - h(j)|. \quad (4)$$

Then, we locate the four largest h' values as follows:

$$j_1 = \text{argmax}_j h'(j), \quad (5)$$

$$j_2 = \text{arg2ndmax}_j h'(j), \quad (6)$$

$$j_3 = \text{arg3rdmax}_j h'(j), \quad (7)$$

$$j_4 = \text{arg4thmax}_j h'(j). \quad (8)$$

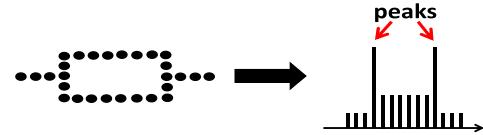


Fig. 3. In the ideal case, the vertical projection of a candidate shape that contains a perfect rectangular shape has two peaks.

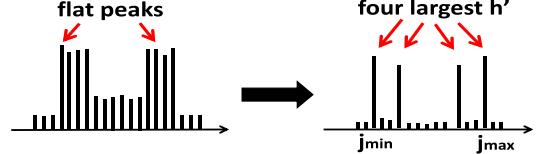


Fig. 4. For a projection whose peaks are noisy and flat, we first detect four largest h' values, then j_{\min} and j_{\max} .

Algorithm 1 Removal of Outer Outliers

input: a candidate shape $P = \{p_i\}_{i=1}^n$ // represented as a $n \times 2$ matrix

output: a subset $P' \subset P$

1. $m(P) = \frac{1}{n} \sum_{i=1}^n p_i$ // mean of P , a 1×2 matrix
 2. $P_c = P - \text{ones}(n, 1)m(P)$ // centralization
 3. $A = P_c^T P_c$ // covariance
 4. $[U, D, V] = \text{svd}(A)$
 5. $P_{c,r} = P_c U$ // rotate centralized points
 6. construct the projection $h(\cdot)$ by Eq. 3
 7. compute $h'(\cdot)$ by Eq. 4
 8. compute j_1, \dots, j_4 by Eq. 5
 9. $j_{\min} = \min(j_1, \dots, j_4)$
 10. $j_{\max} = \max(j_1, \dots, j_4)$
 11. $P'_c = \{p \in P_c | j_{\min} \leq p_{i,y} \leq j_{\max}\}$
 12. $m = |P'_c|$
 13. $P' = P'_c U^{-1} + \text{ones}(m, 1)m(P)$ // recover
-

Denote $j_{\min} = \min(j_1, j_2, j_3, j_4)$ and $j_{\max} = \max(j_1, j_2, j_3, j_4)$. Then $P' = \{p_i \in P | j_{\min} \leq p_{i,y} \leq j_{\max}\}$ is a subset of P without outer outliers. Fig. 4 illustrates the above method.

We summarize the above processing in Algorithm 1. The first part of the algorithm (steps 1 to 5) is the normalization of P . The time complexity of Algorithm 1 is in $O(|P|)$, where $|P|$ indicates the cardinality of set P . Fig. 5 illustrates the intuition of the proposed method.

B. Detection of Four Vertex Candidates

The key step in the proposed framework on rectangular shape detection is the determination of the best four candidates of vertices. We propose two methods, Bounding-Box and Diagonal-First, to detect four vertex candidates of a candidate shape (without outer outliers). The term *bounding box* used in this paper refers to *axis-aligned bounding box*.

1) *Bounding-Box Vertex Detection*: Given a candidate shape $P = \{p_i = (x_i, y_i)\}_{i=1,\dots,n}$, we denote $x_{\min} = \min_{i=1,\dots,n} x_i$, $x_{\max} = \max_{i=1,\dots,n} x_i$, $y_{\min} = \min_{i=1,\dots,n} y_i$, $y_{\max} = \max_{i=1,\dots,n} y_i$. The bounding box of P is constructed by the following four vertices:

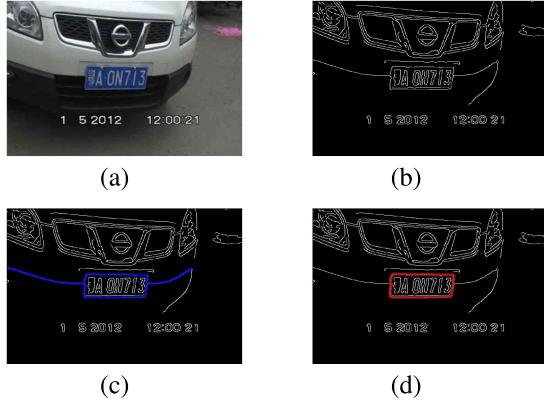


Fig. 5. An example of the effectiveness of the algorithm of removing outer outliers: (a) an input image, (b) an edge image of the input image, (c) a candidate shape in the edge image that contains outer outliers, and (d) the output of the Algorithm 1.

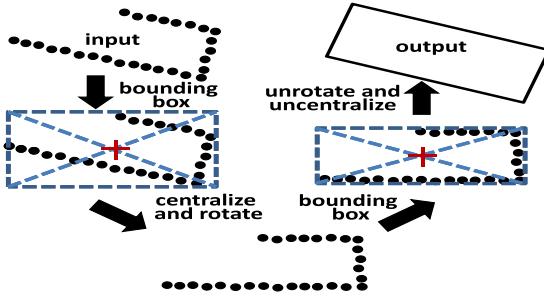


Fig. 6. Determination of four vertex candidates of a candidate shape by the Bounding Box method.

$v_1 = (x_{\min}, y_{\max}), v_2 = (x_{\min}, y_{\min}), v_3 = (x_{\max}, y_{\min}), v_4 = (x_{\max}, y_{\max})$. The centroid of P , i.e., the centroid of the bounding box, is computed by $c(P) = (\frac{x_{\min}+x_{\max}}{2}, \frac{y_{\min}+y_{\max}}{2})$.

Since the orientation of a given candidate shape P may not be horizontal, its bounding box may not match P very well, as illustrated in the first bounding box in Fig. 6. But this bounding box provides us a reliable estimation of $c(P)$ when P is an open shape. In terms of $c(P)$, we can centralize P , then extract the orientation information of P , and last rotate P so that the orientation of the rectangular shape formed by the points is horizontal, based on which we can obtain a bounding box that fits P . In general, the orientation of P can be conveniently estimated by applying SVD (or eigen-decomposition) to the covariance matrix of P . But in the case of missing points (i.e., open rectangular shape), such estimations are not accurate. Instead, we use an exhaustive search strategy to estimate the orientation of P . Fig. 6 illustrates the idea of using the bounding box technique to detect four vertex candidates of a candidate shape.

Given a degree θ , we denote

$$U_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

its rotation matrix. We summarize the Bounding-Box method in Algorithm 2. The time complexity of Algorithm 2 is $O(|P|)$.

2) *Diagonal-First Vertex Detection*: The Diagonal First method starts from the detection of a diagonal of a rectangular candidate shape P . Specifically, denote v_1 and v_2 a pair of points in P such that their Euclidean distance is the maximum

Algorithm 2 Bounding-Box Vertex Detection

input: a candidate shape $P = \{p_i\}_{i=1,\dots,n}$

Output: v_1, v_2, v_3 and v_4

1. compute the centroid $c(P)$
 2. $P_c = P - \text{ones}(n, 1)c(P)$ // centralization
 3. Search θ^* to minimize area of the bounding box of $P_c U_\theta$
 4. $P_{cr} = P_c U_{\theta^*}$ // rotate centralized points
 5. $[v_{cr,1}, v_{cr,2}, v_{cr,3}, v_{cr,4}]$ = the bounding box of P_{cr}
 6. for each vertex $v_{cr,i}, i = 1, \dots, 4$
 7. $v_i = v_{cr,i} U_{\theta^*}^{-1} + c(P)$ // unrotate and uncentralized
-

among all possible pairs of points, i.e.,

$$(v_1, v_2) = \operatorname{argmax}_{p_i \neq p_j} \|p_i - p_j\|, \quad (9)$$

where $\|\cdot\|$ denotes the norm of a vector. v_1 and v_2 are then considered as the end points of a diagonal of P , and thus two vertex candidates.

Next, we compute the third vertex v_3 of P . We propose two criteria to compute v_3 . One criterion is the minimization of the dot product of normalized $v_1 p$ and $v_2 p$, i.e.,

$$\begin{aligned} v_3 &= \min_{p \in P - \{v_1, v_2\}} \left| \frac{v_1 - p}{\|v_1 - p\|} \frac{v_2 - p}{\|v_2 - p\|} \right| \\ &= \min_{p \in P - \{v_1, v_2\}} |\cos \theta(v_1 p, v_2 p)|, \end{aligned} \quad (10)$$

where $\theta(v_1 p, v_2 p)$ is the cross angle between the lines $v_1 p$ and $v_2 p$. In other words, Eq. 10 is equivalent to the maximization of the perpendicularity of two lines $v_1 p$ and $v_2 p$. (Note that $\cos 90^\circ = 0$.)

The second criterion is the maximization of the sum of distances from a point p to v_1 and v_2 , i.e.,

$$v_3 = \max_{p \in P} \|p - v_1\| + \|p - v_2\|. \quad (11)$$

In our experiment, we found that the second criterion is more robust with respect to shape distortion than the first criterion. Specifically, given a set of points forming a perfect rectangular shape as a ground truth, we distort the shape with a certain amount of displacement of these points. We observe that rectangles obtained by the second criterion are consistently closer (in terms of the overlapping area) to the ground truth than rectangles by the first one.

To compute the fourth vertex v_4 , we first compute the centroid of the rectangle by $C = \frac{v_1 + v_2}{2}$. Based on the geometry property of a rectangle, the direction vector $v_3 v_4$ has the same direction as the direction vector $v_3 C$ and $\|v_3 v_4\| = 2\|v_3 C\|$. Thus, we have

$$v_4 = v_3 + 2 \times (C - v_3). \quad (12)$$

Fig. 7 presents an intuitive illustration of the proposed algorithm. We summarize the above processing in Algorithm 3. The time complexity of Algorithm 3 is dominated by Step 1 that takes $O(|P|^2)$. Step 2 is in $O(1)$, and Step 2' is in $O(|P|^2)$. So the choice of the two criteria has no effect on the time complexity of the Diagonal-First detection method. Similar to the Bounding-Box method,

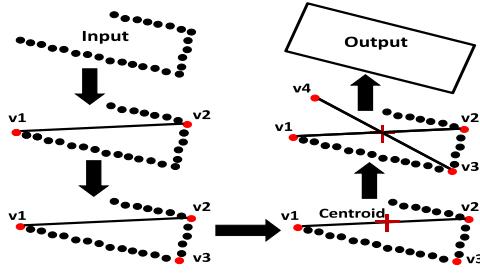


Fig. 7. Determination of four vertex candidates of a candidate shape by Method 1 (Diagonal First).

Algorithm 3 Diagonal-First Vertex Detection

input: a candidate shape $P = \{p_i\}_{i=1,\dots,n}$
output: v_1, v_2, v_3 and v_4 if yes. no otherwise.

1. compute v_1 and v_2 by Eq. 9
2. compute v_3 by Eq. 10
- 2'. compute v_3 by Eq. 11 // alternative criterion
3. compute v_4 by Eq. 12

the Diagonal-First method is robust with respect to candidates of open shapes and the existence of inner outliers. However, they are both sensitive to outer outliers.

C. Rectangularity Verification

After four vertex candidates are detected, we verify the rectangularity of a candidate shape by estimating the percentage of points “on” the boundary of a rectangle. There are two parameters used here: one ϵ_1 is used to control the displacement of a point from the boundary of a rectangle, and the other ϵ_2 is used to control the tolerance of a percentage of outliers. Specifically, given a point p and two vertices v_1 and v_2 , we compute the distance from p to the line segment v_1v_2 by

$$dist(p, v_1v_2) = \left| (p - v_1) \cdot \frac{v_1v_2}{\|v_1v_2\|} \right|,$$

where $\|v_1v_2\|$ is the norm of the vector v_1v_2 , $(a, b)^\perp$ is defined as $(-b, a)$, and \cdot is the dot product. Given four vertices v_1, \dots, v_4 , we call p *on the boundary* of a rectangle or an *inlier* if $\min\{dist(p, v_1v_3), dist(p, v_1v_4), dist(p, v_2v_3), dist(p, v_2v_4)\} \leq \epsilon_1$, where ϵ_1 is the displacement threshold (in pixel unit). In the literature, this threshold is usually set to 2 or 3 pixels [16]. We slightly increase the threshold to 5 in order to adapt to potential distortions of edges in a relatively large scale space. The second parameter ϵ_2 is used to control the tolerance of existence of inner outliers. Precisely speaking, a candidate shape consisting n points is verified as a rectangular shape if and only if $\frac{\#\text{inliers}}{n} \geq 1 - \epsilon_2$. In our experiments, we fix it to be $\epsilon_2 = 0.2$. Algorithm 4 gives a summary of the process. The time complexity of Algorithm 4 is in $O(|P|)$.

Given an input image of size $r \times c$, an edge detector takes $O(krc)$ time to get edge points, where k is a factor equal to the product of the number of channels and the number of

Algorithm 4 Verification of the Rectangularity of a Candidate Shape

input: a candidate shape $P = \{p_i\}_{i=1,\dots,n}$, and four vertices v_1, v_2, v_3 and v_4
output: yes or no

1. count = 0
 2. for $p \in P$
 3. if p “on” v_1v_3, v_1v_4, v_2v_3 , or v_2v_4 (in terms of ϵ_1)
 4. count++
 5. if $\frac{\text{count}}{|P|} > (1 - \epsilon_2)$
 6. return yes
 7. else
 8. return no
-



Fig. 8. Rectangular candidate shapes extracted from different channels under different scales. Each row represents a channel. From top to bottom, channels are red, green, blue, and gray.

scales. The formulation of candidate shapes (i.e., connected components of edge points) takes $O(krc)$ too. Denote m the number of all candidate shapes and P_{\max} the largest candidate shape, the overall time complexity of the detection of rectangular candidate shapes is $O(krc + m|P_{\max}|^2)$. Fig. 8 shows an example of rectangular candidate shapes (colored in red) in channel-scale space of a RGB image, verified by Algorithm 4. Note that the correct candidate of the license plate is extracted in the red channel with scale $\sigma = 3$, and all other candidates are false positives. This shows that the union of candidate shapes increases the chance of locating a rectangular shape by increasing the pool of candidate shapes. Note that we repeat the same process for each candidate shape.

IV. ADDRESS THE FRAGMENTATION ISSUE

In this section, we first propose an algorithm to check if a rectangular candidate shape is open or not. Then we propose

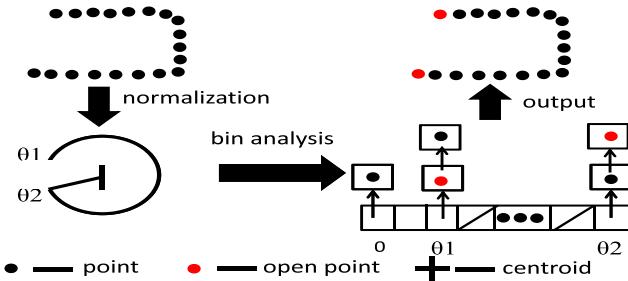


Fig. 9. Illustration of verification of an open shape.

two schemes to check if two open rectangular candidate shapes are fragments of another (bigger) rectangular shape or not, i.e., complementary or not.

A. Verification of Open Shapes

We propose an approach to verify open shapes as illustrated in Fig. 9. The basic idea of the proposed approach is the following. First, we normalize each point to a unit vector based on the centroid. Then, we accumulate normalized points to an angular bin according to their orientations. Based on the characteristic that the angular bin (θ_1) of one side of an open point should be empty and the angular bin (θ_2) of the other side should not be empty, we can locate two angular bins associated with two open points. Finally, we compute pair-wise distances of points in bins θ_1 and θ_2 to find out the shortest distance, and the two points that contribute the shortest distance are output as open points. Alg. 5 describes the details of the procedure of detection of open shapes. The time complexity of Alg. 5 is $O(|P|)$.

B. Verification of the Complementarity of Two Open Rectangular Shapes

Given two open rectangular candidate shapes $P_i, i = 1, 2$, we denote $\theta_l(P_i)$ the angular position of the (first) “left-open” point of a candidate shape P_i in the angular space $[0, 359^\circ]$, and $\theta_r(P_i)$ the angular position of the (first) “right-open” point of the candidate shape P_i . An open point is called left/right-open if the bin of its left/right neighbor in the angular space is empty. In Fig. 9, the point associated with θ_1 is right-open, while the point associated with θ_2 is left-open. Recall that a candidate shape is a connected component (of edge pixels), and there is at most one left-open point and one right-open point for a candidate shape. However, the union of two candidate shapes can have up to two left-open or right-open points.

We propose two schemes to verify the complementarity of two open candidate shapes—one is based on the relation of open degrees of two candidate shapes, and the other is based on the relation of their orientations.

1) *Complementarity Verification via Open Degrees:* Fig. 10 illustrates the basic idea of our first scheme to verify the complementarity of two open candidate shapes. Two rectangular shapes illustrated in the left side of Fig. 10 are complementary; two rectangular shapes illustrated in the right side of Fig. 10 are non-complementary. Note that the centroid we use is the bounding-box centroid of the union $P_1 \cup P_2$.

Algorithm 5 Verification of an Open Shape

input: a candidate shape $P = \{p_i\}_{i=1,\dots,n}$
output: open points if yes

1. compute the centroid $c(P)$
2. $P_c = P - \text{ones}(n, 1)c(P) // \text{centralization}$
3. for each $p = (x, y) \in P_c$
4. $p = \frac{p}{\|p\|}$
5. for $\theta = 1 : 360 // \text{init each angular bin}$
6. $\text{bins}[\theta] \leftarrow \text{null};$
7. for each $p = (x, y) \in P_c$
8. $\theta_p = \text{atan}(\frac{y}{x})$
9. insert p into $\text{bins}[\theta_p]$
10. for $\theta = 1 : 360$
11. $\text{front} \leftarrow \theta + 359 \bmod 360$
12. $\text{rear} \leftarrow \theta + 1 \bmod 360$
13. if $(\text{length}(\text{bins}[\text{front}]) == 0 \&\& \text{length}(\text{bins}[\theta]) > 0)$
14. $\text{idx1} \leftarrow \theta$
15. if $(\text{length}(\text{bins}[\text{rear}]) == 0 \&\& \text{length}(\text{bins}[\theta]) > 0)$
16. $\text{idx2} \leftarrow \theta$
17. find $(p^*, q^*) = \text{argmin } \|p - q\|$, for p and q are nodes of $\text{bins}[\text{idx1}]$ and $\text{bins}[\text{idx2}]$
18. return p^* and q^*

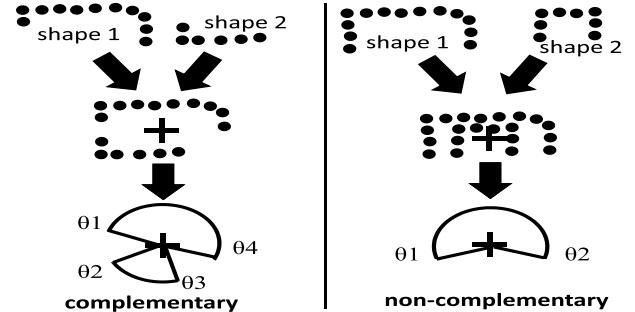


Fig. 10. The first proposed scheme to distinguish the complementarity and non-complementarity of open rectangular candidate shapes.

The “open degrees” of a candidate shape P can be calculated as follows:

$$\text{opendegrees} = \begin{cases} 360 - (\theta_r - \theta_l) & \text{if } \theta_r \geq \theta_l \\ \theta_l - \theta_r & \text{otherwise} \end{cases} \quad (13)$$

For the union $P_1 \cup P_2$, there are three possible cases of open points: 1) $P_1 \cup P_2$ has no open point; 2) $P_1 \cup P_2$ has one left-open point θ_l and one right-open point θ_r ; 3) $P_1 \cup P_2$ has two left-open point θ_{l1}, θ_{l2} and two right-open points θ_{r1}, θ_{r2} . In the third case, we calculate the open degrees of a candidate shape P as follows:

$$\text{opendegrees} = \begin{cases} 360 - [(\theta_{r1} - \theta_{l1}) + (\theta_{r2} - \theta_{l2})] & \text{if } \theta_{r1} \geq \theta_{l1} \\ (\theta_{l1} - \theta_{r1}) + (\theta_{l2} - \theta_{r2}) & \text{otherwise} \end{cases} \quad (14)$$

Fig. 11 presents an intuitive illustration of the calculation of open degrees of a candidate shape with two open points and a union of two candidate shapes with four open points.

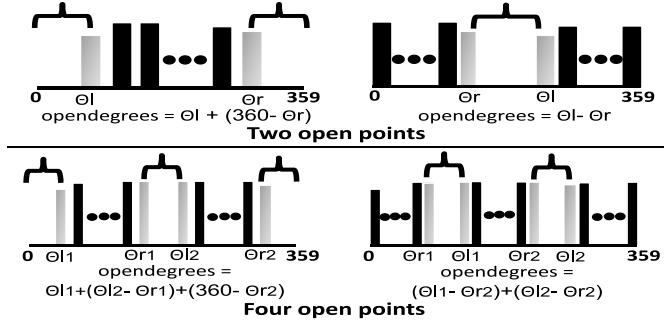


Fig. 11. Illustration of the calculation of “open degrees” based on left-open and right-open points. First row: Case of two open points. Second row: Case of four open points.

Formally, two open rectangular shapes P_1 and P_2 are called *complementary* if

$$\begin{aligned} \text{opendegrees}(P_1) + \text{opendegrees}(P_2) \\ - \text{opendegrees}(P_1 \cup P_2) = 360^\circ. \end{aligned} \quad (15)$$

Otherwise, P_1 and P_2 are called *non-complementary*.

A union $P_1 \cup P_2$ forms a synthesized shape. For convenience, we call it a *2-piecewise* candidate shape. We then apply a vertex detection method (Alg. 2 or Alg. 3) and the rectangularity verification method (Alg. 4) to all (synthesized) 2-piecewise candidate shapes.

We can recursively define a *n-piecewise* candidate shape by testing if a $(n - 1)$ -piecewise candidate shape and a 1-piecewise candidate shape (i.e., a connected component) are both open shapes and verifying their complementarity. It is also worth noting that the open shape verification and the complementarity verification are not limited to rectangular shapes. However, we empirically observed that *2-piecewise candidate shapes are effective enough* to detect rectangles robustly due to the low complexity of a rectangle.

2) *Complementarity Verification via Orientations*: Denote $c(P)$ the centroid of a candidate shape P , which is estimated by the centroid of the bounding box of P . Given two candidate shapes $P_i, i = 1, 2$, denote a directional vector $\Delta c(P_i) = c(P_1 \cup P_2) - c(P_i)$ whose orientation is dependent on the “context” $P_1 \cup P_2$. Denote $\theta_{cb}(P_i)$ the orientation of $\Delta c(P_i)$. We call $\theta_{cb}(P_i)$ the *context-based* orientation of a shape P_i . Intuitively, an open rectangular candidate shape also has its own orientation, i.e., independent on other candidate shapes. We call this *context-free* orientation of a shape.

Fig. 12 illustrates a comparison of the spatial context of two candidate shapes. Two rectangular shapes shown in the left side of Fig. 12 are complementary; two rectangular shapes shown in the right side of Fig. 12 are non-complementary. The basic idea of the second complementarity verification scheme is to check the “consistency” between the context-based orientation and its context-free orientation.

We formally define the *context-free orientation* of a candidate shape P as follows:

$$\theta_{cf}(P_i) = \begin{cases} \frac{\theta_l + \theta_r}{2} & \text{if } |\theta_r - \theta_l| \leq 180 \\ \left(\frac{\theta_l + \theta_r}{2} + 180\right) \bmod 360 & \text{otherwise} \end{cases} \quad (16)$$

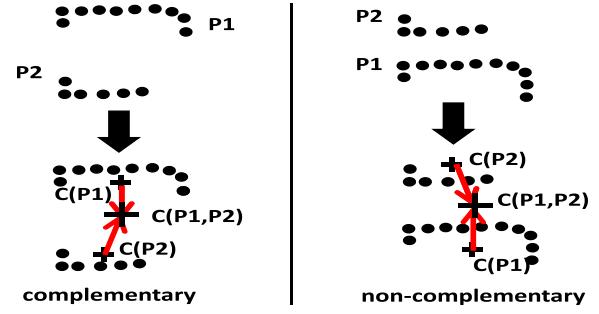


Fig. 12. The second proposed scheme to distinguish the complementarity and non-complementarity of open rectangular candidate shapes. $c(P_i)$ is the centroid of the bounding box of the candidate shape $P_i, i = 1, 2$. $c(P_1, P_2)$ is the centroid of the bounding box of $P_1 \cup P_2$. A red arrow indicates the *context-based orientation* of a candidate shape.

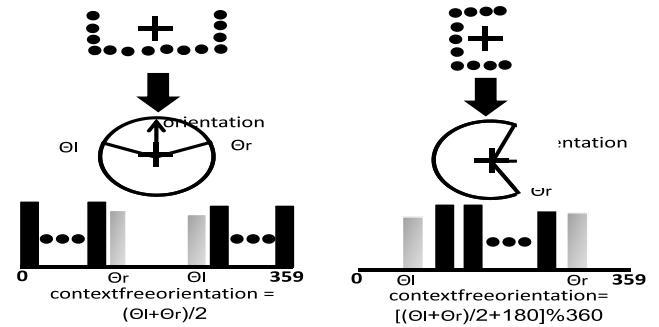


Fig. 13. Illustration of the estimation of the context-free orientation of an open candidate shape based on its left-open and right-open points. First column: Case of $|\theta_r - \theta_l| \leq 180$. Second column: Case of $|\theta_r - \theta_l| > 180$.

Fig. 13 presents the intuition of the computation of the orientation of an open candidate shape. Under the second scheme of complementarity verification illustrated in Fig. 12, we now propose the second complementarity verification method: Two open shapes $P_i, i = 1, 2$ are *complementary* if

$$\cos \theta_{cf}(P_i) \cos \theta_{cb}(P_i) + \sin \theta_{cf}(P_i) \sin \theta_{cb}(P_i) \geq 0, \quad i = 1, 2, \quad (17)$$

i.e., the similarity (dot product) between $(\cos \theta_{cf}(P_i), \sin \theta_{cf}(P_i))$ and $(\cos \theta_{cb}(P_i), \sin \theta_{cb}(P_i))$ is greater than or equal to 0. Otherwise, they are called *non-complementary*.

Overall, the time complexity to address the open shape issue is $O(|P_1| + |P_2|)$, where P_1 and P_2 are two candidate shapes.

V. AN INTERESTNESS MEASURE OF RECTANGULAR SHAPES

The above proposed methods may detect many false positive rectangular shapes. To quantify the likelihood of a rectangular shape being a true positive, we propose an interestness measure of rectangular shapes by integrating imbalanced points.

Imbalanced point detection aims to minimize the occurrences of edge points [14]. Since edge points have similar local appearances (i.e., not distinctive to each other), they increase the chance of mismatching in the higher-level applications. Edge points can be characterized as points of balanced local appearances. Specifically, denote I a grayscale image, p an image point, $\theta_i = (i - 1) * \Delta\theta$, and $l_i = (\cos \theta_i, \sin \theta_i)$,

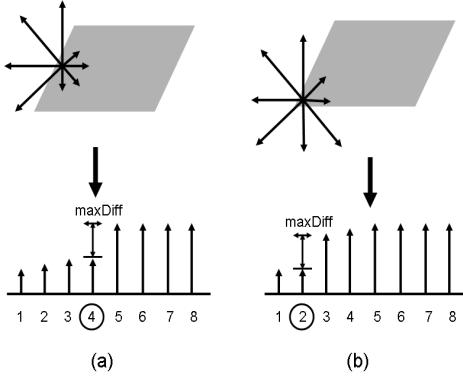


Fig. 14. Illustration of the imbalance oriented selection. Eight arrows are sorted in term of their magnitudes, as shown in the second row. (a) An edge point of balanced local appearance, where the index of maximum difference is 4 (half of 8 directions); (b) An imbalanced point, where the index of maximum difference is 2 [14].

for $i = 1, \dots, n$, where $\Delta\theta = \frac{360}{n}$. Denote $\frac{\partial I}{\partial l_i}(p)$ a directional derivative of p along l_i direction. We cluster $\frac{\partial I}{\partial l_i}(p)$, $i = 1, \dots, n$, into two classes in terms of their cardinality $|\frac{\partial I}{\partial l_i}(p)|$. If two clusters have the same size, the image point p is called *balanced*. Otherwise, p is *imbalanced*.

A sorting method [14] was proposed to group $\frac{\partial I}{\partial l_i}(p)$, $i = 1, \dots, n$, as follows: by sorting the magnitudes of the derivatives, we have

$$\left| \frac{\partial I}{\partial l_{i_1}}(p) \right| \leq \dots \leq \left| \frac{\partial I}{\partial l_{i_n}}(p) \right|,$$

where i_j , $j = 1, \dots, n$, is a permutation of numbers $1, \dots, n$. Denote

$$\text{maxDiff} = \max_{j=1, \dots, n-1} \left(\left| \frac{\partial I}{\partial l_{i_{j+1}}}(p) \right| - \left| \frac{\partial I}{\partial l_{i_j}}(p) \right| \right),$$

and denote the index of maximum difference as

$$j^* = \operatorname{argmax}_{j=1, \dots, n-1} \left(\left| \frac{\partial I}{\partial l_{i_{j+1}}}(p) \right| - \left| \frac{\partial I}{\partial l_{i_j}}(p) \right| \right). \quad (18)$$

If the index of maximum difference is equal to $\frac{n}{2}$, p is considered as a balanced point. Otherwise, p is imbalanced. In [14], the authors further consider an image point p whose index of maximum difference is larger than $n/2$ as a redundant point because of its co-occurrence of a certain point whose index of maximum difference is less than $n/2$. It is clear that a larger n (more directions) gives more precise characterization of the imbalance of an image point. However, a larger n also increases computational cost. We apply Bresenham's line algorithm [2] to efficiently extract line segments within a neighborhood of an image point for the computation of the first-order directive along each direction in order to reduce the computational cost. In our study, the radius of the neighborhood is set to 5.

Fig. 14 illustrates the basic idea of imbalance oriented selection given $n = 8$. A long (short) arrow indicates a directional derivative on p of a large (small) magnitude. The number of long arrows is equal to the number of short arrows, which indicates the balance nature of an edge point. Fig. 14(a)



Fig. 15. (a) Imbalanced points detected in a vehicle image; (b) License plate detected by integrating rectangular shapes and imbalanced points.

shows a case of a balanced point. Fig. 14(b) shows a case of an imbalanced point, where the number of long arrows is not equal to the number of short arrows.

Given an RGB image I , denote R a rectangular shape and P the entire set of imbalanced points detected in the grayscale of I . We propose an interestingness measurement of a rectangular shape as follows:

$$\text{interestingness}(R) = |\{p \in P : p \text{ inside } R\}|, \quad (19)$$

where $|\cdot|$ refers to the cardinality of a set. The R with the highest interestingness is output as the best candidate of a license plate.

Fig. 15 (a) shows an example of imbalanced points detected in a vehicle image. Note that a significant amount of imbalanced points are detected in the plate region. Fig. 15 (b) shows the rectangular shape with the highest interestingness.

VI. EXPERIMENTS

In this section, we will present visual and quantitative comparisons of different components in the proposed framework. We use two different types of images: i) license plate images, and ii) road sign images.

License plate detection plays an important role in many applications such as vehicle monitoring and parking lot monitoring. Several works were proposed for license plate detection recently [5], [8], [28]. Donoser et al. [5] proposed a detection method using Maximally Stable Extremal Region (MSER) [20]. Miao and Yue [21] proposed a detection method by analyzing connected components in binary images that are converted from input images by Niblack's adaptive thresholding method [24]. Miao and Yue's method was motivated by the success of Niblack adaptive thresholding method in document image processing. Based on a Niblack binary image, Miao and Yue's method first extracts connected components, then applies heuristics on aspect ratio, height, and area of connected components to remove connected components that do not have the shape of a character or a digit. (It is interesting to note that connected components corresponding to the boundary of license plates are also intentionally removed.) To locate a license plate, Miao and Yue's method finally applies heuristics on the distance, height difference, centroid level of each pair of character/digit-like connected components to group all characters or digits together. Hsieh et al. [8] proposed a real-time detection method that uses the locations of two tail lights to determine the location of a license plate. Chen et al. [4] proposed a method that integrates probability models of

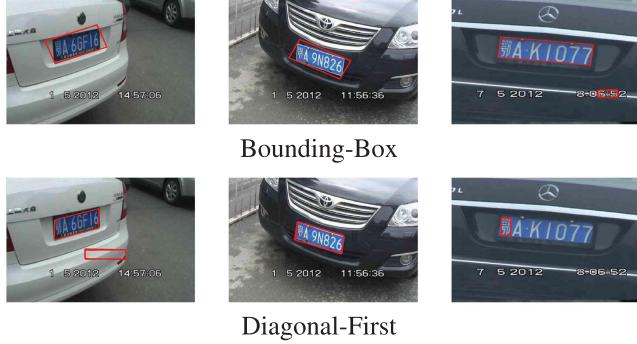


Fig. 16. A comparison between the Bounding-Box method and the Diagonal-First method. The best three rectangles in terms of the proposed interestness measure are displayed. The Bounding-Box method suffers from a serious perspective distortion of a license plate. But the Bounding-Box method is somehow complementary to the Diagonal-First method when a license plate has no serious perspective issue.

shape feature, texture feature and color (HSI) features, and their experiments showed that the method achieved 97% detection accuracy. Li [15] proposed a SVM learning method using HoG (Histogram of Gradient) features. Zhou et al. [28] proposed a Principal Visual Word (PVW) method that integrates the classic bag-of-word model with a geometric context.

In this experiment, We will compare the proposed method with two most related methods: i) Donoser et al. [5] and ii) Miao and Yue [21]. All three methods are based on the analysis of geometry information. But a difference between the proposed method and the other two is that the proposed method makes use of edge information, while the other two methods make use of region information.

Note that there are two sets of parameters in the proposed framework. The first set of parameters is used in the detection of rectangular candidate shapes. There are two essential parameters in this set, i.e., ϵ_1 and ϵ_2 . Recall that these parameters are used together to verify the rectangularity of a candidate shape. ϵ_1 is the threshold used to decide whether a point is on a line or not, and ϵ_2 is the threshold on the tolerance of shape distortion. We set $\epsilon_1 = 5$ and $\epsilon_2 = 0.8$, respectively. The threshold t in the definition of candidate shape (Eq. 1) is set to 50 (pixels). The range of scale space in candidate shape extraction is set to the interval [1, 5]. Both parameters have more influence on the running time than the detection accuracy. Other parameters in canny edge detection are default parameters set in Matlab. The second set of parameters contains one essential parameter, which is n (the number of discretized directions in an angular space. Consistent with our previous work [11], [12], we set $n = 64$. The size of our test images is 1536×418 .

A. Bounding-Box vs. Diagonal-First

First, we present a visual comparison of the Bounding-Box method and the Diagonal-First method, as shown in Fig. 16. The best three rectangles in terms of the proposed interestness measure are displayed. As shown by the first two images in Fig. 16, the Bounding-Box method did not perform very well if a license plate is subjected to a significant

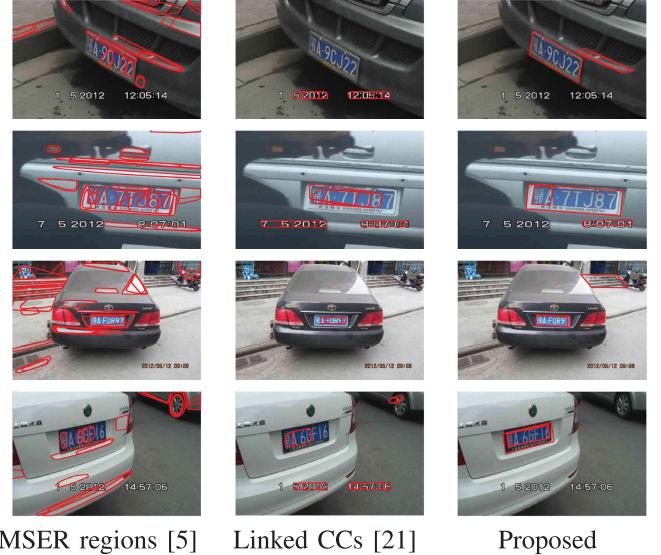


Fig. 17. A comparison between the proposed method and two region-based methods [5], [21].

amount of perspective distortion as the rectangularity of the shape does not hold any more. Instead, the Diagonal-First method can tolerate a serious perspective distortion (up to 25° or so). However, if there is no serious perspective distortion, the Bounding-Box method can outperform the Diagonal-First method, as shown by the last two images in Fig. 16. This comparison shows the benefit of integrating the two methods in the context of license plate detection.

B. More Visual Results

Next, we give a visual comparison between the proposed method and two region-based methods [5], [21], as shown in Fig. 17, to get an intuition on the behaviors of these methods. For Miao and Yue's method [21], we draw the convex hull of linked connected components as a plate candidate. For the proposed method, the three rectangles with highest interestness are output. From Fig. 17, we observe that MSER regions extracted in an image (e.g., in the first or fourth column) do not always include a plate region. We also observe two main shortcomings in Miao and Yue's method: 1) it fails to address the case that a plate character consists of multiple connected components, such as the first character (a Chinese character) in the test images; 2) it is sensitive to a perspective view (see the worst two cases are the first and the fourth images where plates are under larger perspective views than other two). It is mainly caused by the heuristic on the width and height of a plate character or digit.

Fig. 18 shows some positive samples of the proposed method. The rectangles displayed are the best candidate rectangular shapes in terms of the ratio of the rectangle's area to the number of imbalanced points enclosed in the rectangle.

Fig. 19 shows a comparison between Lowe's keypoints [19] and imbalanced points [14]. The numbers of keypoints and imbalanced points are 1921 and 289, respectively. A large

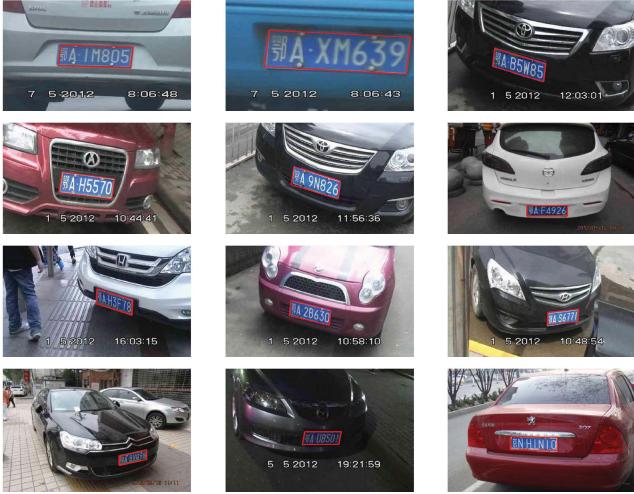


Fig. 18. Positive samples. The best rectangular shape (in terms of proposed criterion) encloses the license plate in an input image.



Lowe's keypoints [19]

Imbalanced points [14]

Fig. 19. A comparison between Lowe's keypoints and imbalanced points.

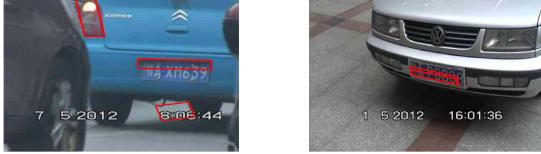


Fig. 20. Negative samples. None of the best three candidates (in terms of the proposed interest measurement) can localize the license plate.

number of keypoints increase the cost of computing the rectangle-over-point ratio. Moreover, keypoints appear much more noisy than imbalanced points, i.e., they are less correlated with a license plate region.

Fig. 20 shows two negative samples, where none of the best candidates (in terms of the proposed criterion) encloses the license plate. By analyzing the intermediate results of the framework, we found that the failure is generally caused by distorted rectangular shapes. Specifically, the rectangular shape of a license plate is merged with some enclosed rectangular shapes, due to a low contrast between a license plate and its background, an appearance disruption of a license plate, illumination, or other factors. Simply speaking, this is caused by the limitation of edge detection. So a possible strategy to enhance a detection rate is to integrate region information (such as MSER and linked connected components in Niblack



Fig. 21. Examples on road sign images. The best rectangle (in terms of the interestness measure) is shown.

binary images). In contrast to edge detection, imbalanced point detection is more stable with respect to the image noise, illumination, and transformation, which has been shown in repeatability evaluation [14]. Thus, the imbalanced point based interestness measurement works stably in the detection of license plates.

We also tested the proposed framework on road sign images. Fig. 21 shows some examples of road sign images.

C. Quantitative Evaluation

In the following, we present a quantitative evaluation of the proposed framework based on detection rates. Given an image, the result is defined as a *successful detection* if one of the first best three rectangles (in terms of interest measure) has larger than 95% overlapping regions with the ground truth. We test the methods on a dataset consisting of 1000 vehicle images that contain license plates. The dimension of these images is 512×401 .

In the context of license plate detection, imbalanced points have two appealing properties. First, imbalanced points contain corner structures [12]. Note that many bold capital characters or digits contain vertices that are essentially corners. Second, imbalanced points have localities, i.e., an imbalanced point may be continuous to another imbalanced point. In our previous study, we observed that stronger localities are demonstrated in a scene with stronger line structures, e.g., bold capital characters or digits.

Table I shows detection rates obtained by i) two region-based methods (Donoser et al. [5] and Miao & Yue [21]), and ii) different combinations of components in the proposed framework, where imbalanced points are used to measure the interestness of a rectangle. The first three rows show the detection rates of the method of Donoser et al. (67%), Miao & Yue's method (82%), and Zhou et al. (94%), which are all lower than the proposed entire framework (97%), as shown in the last row. The next three rows show a comparison of the Bounding-Box Rectangularity Verification (BBRV) method, the Diagonal-First Rectangularity Verification (DFRV) method, and the integration of both methods. (The integration means that a candidate shape is output as a rectangular shape if one of these methods has a successful output.) This result is consistent with the visual comparison of BBRV and DFRV shown in Fig. 16—BBRV is more sensitive to perspective distortion, while BBRV can help improve DFRV. The 7th row shows the

TABLE I

DETECTION RATES OF TWO REGION-BASED METHODS AND DIFFERENT COMBINATIONS OF COMPONENTS IN THE PROPOSED FRAMEWORK. RO: REMOVAL OF OUTLIERS; BBRV: BOUNDING-BOX RECTANGULARITY VERIFICATION; DFRV: DIAGONAL-FIRST RECTANGULARITY VERIFICATION; B-RV: BOTH RECTANGULARITY VERIFICATION; DCV: DEGREE-BASED COMPLEMENTARITY VERIFICATION; OCV: ORIENTATION-BASED COMPLEMENTARITY VERIFICATION

Method	Detection Rate (%)
Donoser et al. [5]	67
Miao & Yue [21]	82
Zhou et al. [28]	94
BBRV	68
DFRV	88
BBRV&DFRV	90
RO+BBRV&DFRV	93
BBRV&DFRV+DCV	93
BBRV&DFRV+OCV	95
RO+BBRV&DFRV+OCV	97

effectiveness of the method on the removal of outliers (RO). The 8th and 9th rows show a comparison of Degree-based Complementarity Verification (DCV) method and Orientation-based Complementarity Verification (OCV) method. We can observe that the later method outperforms the former method. We also test the framework with interestness measured by Lowe's keypoints, and the obtained result is 92%.

The average running time to process an image by the proposed framework is 1.1 seconds, which is higher than while comparable to the running time of the method of Donoser et al. (0.59 second) and Miao & Yue's method (0.91 second). The hardware setup for the estimation of running time is: CPU 2.10G and Memory 2G. Matlab is the platform of code implementation.

VII. CONCLUSION

In this paper, we proposed a geometric framework for rectangular shape detection. The proposed framework addresses three issues of a candidate shape: i) outer outliers, ii) open shape, and iii) fragmentation. Furthermore, we propose an interestness measure of a rectangular shape by the integration of imbalanced points. Our experimental study demonstrates the promising results of the proposed framework.

REFERENCES

- H. Bhaskar, N. Werghi, and S. A. Mansoori, "Combined spatial and transform domain analysis for rectangle detection," in *Proc. 13th Conf. Inf. Fusion (FUSION)*, Jul. 2010, pp. 1–7.
- J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.
- J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- Z.-X. Chen, C.-Y. Liu, F.-L. Chang, and G.-Y. Wang, "Automatic license-plate location and recognition based on feature salience," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3781–3785, Sep. 2009.
- M. Donoser, C. Arth, and H. Bischof, "Detecting, tracking and recognizing license plates," in *Proc. 8th Asian Conf. Comput. Vis. (ACCV)*, vol. 2, 2007, pp. 447–456.
- I. Elouedi, A. Hamouda, and H. Rojbani, "Rectangular discrete radon transform towards an automated buildings recognition from high resolution satellite image," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2012, pp. 1317–1320.
- D.-S. Gao and J. Zhou, "Car license plates detection from complex scene," in *Proc. 5th Int. Conf. Signal Process.*, vol. 2, 2000, pp. 1409–1414.

- C.-T. Hsieh, L.-C. Chang, K.-M. Hung, and H.-C. Huang, "A real-time mobile vehicle license plate detection and recognition for vehicle monitoring and management," in *Proc. Joint Conf. Pervas. Comput.*, Dec. 2009, pp. 197–202.
- C. R. Jung and R. Schramm, "Rectangle detection based on a windowed Hough transform," in *Proc. 17th Brazilian Symp. Comput. Graphics Image Process. (SIBGRAPI)*, Oct. 2004, pp. 113–120.
- S. Karungaru, M. Fukumi, N. Akamatsu, and T. Akashi, "Detection and recognition of vehicle license plates using template matching, genetic algorithms and neural networks," *Int. J. Innovative Comput., Inf., Control*, vol. 5, no. 7, pp. 1975–1985, Jul. 2009.
- Q. Li, "Interest points of general imbalance," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2536–2546, Nov. 2009.
- Q. Li, "Scale assignment for imbalanced points," in *Proc. 23rd IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2011, pp. 197–204.
- Q. Li and Z. Xia, "Detecting image points of diverse imbalance," in *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2008, pp. 2592–2595.
- Q. Li, J. Ye, and C. Kambhamettu, "Interest point detection using imbalance oriented selection," *Pattern Recognit.*, vol. 41, no. 2, pp. 672–688, 2008.
- X. Li, "Vehicle license plate detection and recognition," M.S. thesis, Dept. ECE, Univ. Missouri, Columbia, MO, USA, Dec. 2010.
- C. Lin and R. Nevatia, "Building detection and description from a single intensity image," *Comput. Vis. Image Understand.*, vol. 72, no. 2, pp. 101–121, 1998.
- T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention," *Int. J. Comput. Vis.*, vol. 11, no. 3, pp. 283–318, 1993.
- Y. Liu, T. Ikenaga, and S. Goto, "An MRF model-based approach to the detection of rectangular shape objects in color images," *Signal Process.*, vol. 87, no. 11, pp. 2649–2658, 2007.
- D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.
- M. Ligang and Y. Yongjuan, "Automatic license plate detection based on connected component analysis and template matching," in *Proc. 2nd Int. Conf. Intell. Control Inf. Process.*, Jul. 2011, pp. 1090–1093.
- H. Moon, R. Chellappa, and A. Rosenfeld, "Performance analysis of a simple vehicle detection algorithm," *Image Vis. Comput.*, vol. 20, no. 1, pp. 1–13, 2002.
- R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comput. Graphics Image Process.*, vol. 13, no. 3, pp. 257–269, 1980.
- W. Niblack, *An Introduction to Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986, pp. 115–116.
- W.-B. Tao, J.-W. Tian, and J. Liu, "A new approach to extract rectangular building from aerial urban images," in *Proc. 6th Int. Conf. Signal Process.*, vol. 1, Aug. 2002, pp. 143–146.
- Y. Wang, F. Tupin, and C. Han, "Building detection from high-resolution PolSAR data at the rectangle level by combining region and edge information," *Pattern Recognit. Lett.*, vol. 31, no. 10, pp. 1077–1088, 2010.
- T. Zhao and R. Nevatia, "Car detection in low resolution aerial image," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, vol. 1, 2001, pp. 710–717.
- W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, Sep. 2012.
- Y. Zhu, B. Carragher, F. Mouche, and C. S. Potter, "Automatic particle detection through efficient Hough transforms," *IEEE Trans. Med. Imag.*, vol. 22, no. 9, pp. 1053–1062, Sep. 2003.



Qi Li is an Associate Professor with the Department of Computer Science, Western Kentucky University, Bowling Green, KY, USA. He received the Ph.D. degree in computer science from the University of Delaware, Newark, DE, USA, in 2006. His current research interests include pattern recognition, computer vision, machine learning, and bioinformatics. He is an Associate Editor of *Neurocomputing*, the *International Journal of Data Mining, Modelling, and Management*, and the *International Journal of Data Analysis Techniques and Strategies*.