Hindawi Publishing Corporation
International Journal of Antennas and Propagation
Volume 2013, Article ID 357379, 9 pages
http://dx.doi.org/10.1155/2013/357379

Research Article

Junction Point Detection Algorithm for SAR Image

Jun Zhang, Tingjin Luo, Gui Gao, and Lin Lian

Department of Information System and Management, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Jun Zhang; aiya0215@263.net

Received 19 December 2012; Revised 4 February 2013; Accepted 21 February 2013

Academic Editor: Deren Li

In this paper, we propose a novel junction point detector based on an azimuth consensus for remote sensing images. To eliminate the impact of noise and some noncorrelated edges of SAR image, an azimuth consensus constraint is developed. In addition to detecting the locations of junctions at the subpixel level, this operator recognizes their structures as well. A new formula that includes a minimization criterion for the total weighted distance is proposed to compute the locations of junction points accurately. Compared with other well-known detectors, including Forstner, JUDOCA, and CPDA, the experimental results indicate that our operator outperforms them both in location accuracy of junction points and in angle accuracy of branch edges. Moreover, our method possesses satisfying robustness to the impact of noise and changes of the SAR images. Our operator can be potentially used to solve a number of problems in computer vision, such as SAR image registration, wide-baseline matching, and UAV navigation system.

1. Introduction

Synthetic Aperture Radar is able to provide high resolution ground data or images even under heavy weather conditions, and it is widely used in areas of military reconnoiter, topographical mapping, resource exploration and vegetation analysis, and so on. Therefore, it is very important for understanding and analyzing SAR images. Nevertheless, due to the impact of image-forming mechanism and wide dynamic range, SAR images consist of massive coherent macula noises and make the accurate extractions of corners very difficult at the same time, they also hinder the development of SAR image application systems. Therefore, it is crucial to detect and extract the identifiable, invariant, and information-intensive features for SAR image application systems.

Corner feature detection in images is a fundamental problem in computer vision and has been successfully used in visual tracking, panoramic image stitching, and motion estimation [1–3] among other applications. In these systems, detecting corner features is the first critical step toward many more complicated processes. Over the past decades, there have been many corner and junction detectors proposed in published literatures. These detectors compete with each other in terms of localization accuracy, speed, and information they provide. Common corner detection algorithms can be divided into three categories: (1) algorithms based on gray level statistics, which detect the pattern of the center point by calculating the numbers of similar approximation (or different) points between the local neighborhood (or boundary) and the center of the gray scale, such as SUSAN [4] and FAST [5]; (2) algorithms based on the second-order structure tensor, which builds the autocorrelation matrix of the corner around its local neighborhood region and determines whether the point is the corner by eigenvalue analysis of the matrix, such as the typical algorithms of Förstner and Gülch [6], Harris and Stephens [7], and KLT [8] and some other methods based on improved structure tensors [9, 10]; (3) algorithms based on curvature analysis, that is, the detection of the large curvature points on the edge at the corners, such as CSS [11], ECSS [12], and CPDA [13].

Although corner detectors have been found to perform very well in many areas, a number of inherent weaknesses have been exposed in practical application. Most often, the criterion on which the detection is based is neglected (e.g., local structural information). In particular for SAR images, because they include massive coherent noises, the traditional corner detectors will extract pseudopoints and affect the efficiency and results of SAR image matching and application. By contrast, junction detectors are concerned with both the locations and structures. Due to multibranches topology stability, during matching detectors can effectively filter out the external interference by using richer information and thereby ensure better matching results.

Although McDermott [14] observes that junction point detection is very difficult, even a well-developed human visual system is not an exception. However, due to the abovementioned advantages of junction detectors, there are many junction detectors proposed in published literatures. The approach described in [15] detects junctions using a piecewise constant function that partitions a circular template into wedge-shaped regions and introduces a minimum description length principle and dynamic programming algorithm to compute the optimal parameters of a model. Chabat et al. [16] introduce a junction detector based on the analysis of local anisotropy and identifying corners as points with a strong gradient but not oriented in a single dominant direction. Cazorla et al. [17, 18] propose two Bayesian methods for junction classification that evolved from the Kona method: a region-based method and an edge-based method. Bergevin and Bubel [19] propose a junction characterization and a validation method where junction branches of volumetric objects are extracted at points of interest in a 2D image using a topologically constrained grouping process and a binary split tree. Perwass [20] proposes a method to extract the intersections between the conic curves and to determine all possible linear support domains, then to determine the edges from the image gradients and to determine the type of extracted junction points by the local geometry structural analysis of edges.

Recently, Elias and Laganière [21] propose JUnction Detection Operator based on Circumferential Anchors (JUDOCA), which represents the latest research result on junction point detection algorithms. JUDOCA has been successfully used to solve many problems, such as 3-D reconstruction, camera parameter enhancing, and indoor and obstacle localization [22–24]. However, JUDOCA also has some drawbacks; for example, it only computes integer-valued junction points and cannot achieve subpixel position precision and uses the path directions instead of the dip angles of junction branches, which brings extra errors, and its algorithm is sensitive to fractured edges.

In this paper, we present a novel branch-point detection algorithm based on azimuth consensus. At the same time, compared with previous methods, our proposed algorithm can extract junction points at the level of sub-pixel accuracy and low contrast change and build a set of characteristic descriptions for recognition. In addition, experimental results indicate that our proposed algorithm provides improved positioning accuracy for junction points and angle accuracy for branch edges, and it has improved the robustness to noises especially for SAR images.

The contribution of this paper is summarized as follows. Section 2 provides the definition of the azimuth consensus and groups the edge points that satisfy the azimuth consensus constraints. Section 3 describes the accurate calculation of junction point locations and the characteristic description for a junction point and its branch edge structure. Experimental results are present in Section 4, where proposed algorithm is compared with previous algorithms with respect to accuracy, contrast change, noise, and SAR images. Section 5 concludes the paper and discusses future work.

2. Junction Point Detection and Branch Edge Grouping

2.1. Subpixel Location of Edge Points. For a given SAR image $f(\mathbf{x})$, where $\mathbf{x} = [x, y]^{\mathrm{T}}$ is the coordinate vector, which is corresponding to a pixel in the image, the gradient of $f(\mathbf{x})$ in the location $\mathbf{x}$ is defined as

$$\overrightarrow{f}(\mathbf{x}) = \left[f_x(\mathbf{x}), f_y(\mathbf{x})\right]^{\mathrm{T}}, \tag{1}$$

where $f_x(\mathbf{x})$ and $f_y(\mathbf{x})$ are the first-order partial derivatives of $f(\mathbf{x})$ with respect to the x- and y-directions, respectively.

The gradient magnitude and orientation in $\mathbf{x}$ are given by

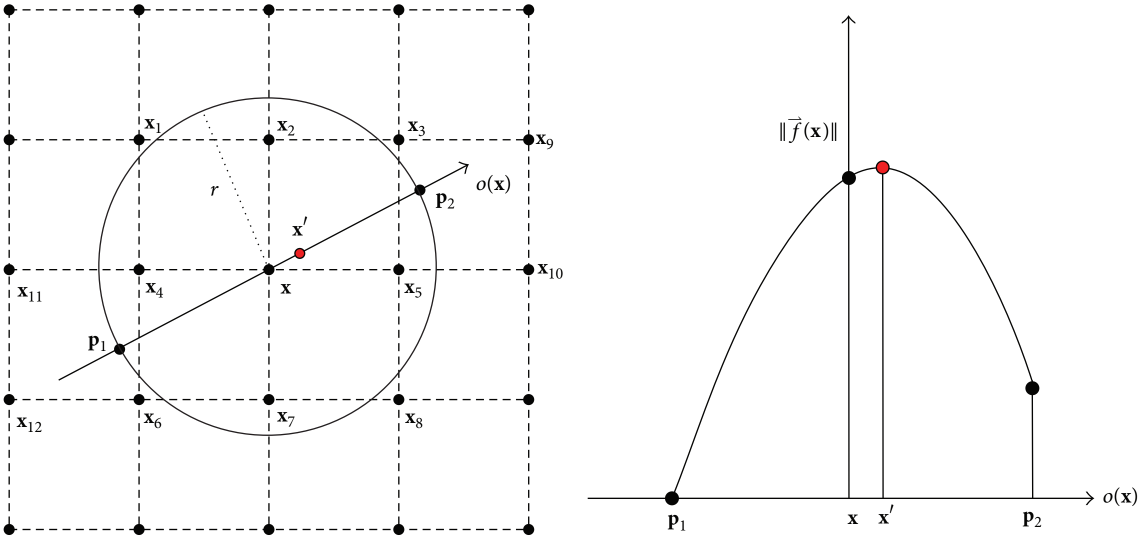
$$\begin{gathered}\left\|\overrightarrow{f}(\mathbf{x})\right\| = \sqrt{f_x^2(\mathbf{x}) + f_y^2(\mathbf{x})}, \\ o(\mathbf{x}) = \tan^{-1}\left(\frac{f_y(\mathbf{x})}{f_x(\mathbf{x})}\right)//180^\circ,\end{gathered} \tag{2}$$

where // is the modular operation which limits the range of gradient orientation to $[0^\circ, 180^\circ)$, and the vertical direction of gradient orientation is defined as $o^{\perp}(\mathbf{x}) = (o(\mathbf{x}) + 90^\circ)//180^\circ$.

To find the edge points of an image, our method utilizes non-maxima suppression [25] to determine the gradient magnitude image in the direction of the gradient orientation (see Figure 1(a)). Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be two solutions of the following equation (6) in the local neighborhood circle region that has $\mathbf{x}$ as its center and has a radius of r:

$$\begin{gathered}\overrightarrow{f}(\mathbf{x})^{\mathrm{T}}(\mathbf{p} - \mathbf{x}) = 0, \\ \|\mathbf{p} - \mathbf{x}\| = r.\end{gathered} \tag{3}$$

Because it is possible that coordinate values of $\mathbf{p}_1$ and $\mathbf{p}_2$ are nonintegers, their gradient magnitude values of $v(\mathbf{p}_1)$ and $v(\mathbf{p}_2)$ are calculated by the bilinear interpolation method. Let $\|\overrightarrow{f}(\mathbf{x})\|_{\mathrm{NMS}}$ be the gradient magnitude image after performing non-maximal suppression. If $\|\overrightarrow{f}(\mathbf{x})\| < \max(v(\mathbf{p}_1), v(\mathbf{p}_2))$, then$\|\overrightarrow{f}(\mathbf{x})\|_{\mathrm{NMS}} = 0$. If $\|\overrightarrow{f}(\mathbf{x})\| \geq \max(v(\mathbf{p}_1), v(\mathbf{p}_2))$, then $\|\overrightarrow{f}(\mathbf{x})\|_{\mathrm{NMS}} = \|\overrightarrow{f}(\mathbf{x})\|$. In the direction $o(\mathbf{x})$, using the points $(\mathbf{x}, \|\overrightarrow{f}(\mathbf{x})\|)$, $(\mathbf{p}_1, v(\mathbf{p}_1))$ and $(\mathbf{p}_2, v(\mathbf{p}_2))$ fitting a parabolic curve, $\mathbf{x}'$—the real-valued sub-pixel location of $\mathbf{x}$—is the peak



(a) Nonmaximal suppression in the direction of the gradient orientation

(b) Subpixel location of local maxima by a parabolic curve

Figure 1: Non-maximal suppression and sub-pixel location of an edge point.

of the constructed parabolic curve (as shown in Figure 1(b)). The formula of the sub-pixel location $\mathbf{x}'$ is

$$\mathbf{x}' = \mathbf{x} + r \cdot \frac{v(\mathbf{p}_1) - v(\mathbf{p}_2)}{2\left(v(\mathbf{p}_1) + v(\mathbf{p}_2) - 2\left\|\vec{f}(\mathbf{x})\right\|\right)} \begin{bmatrix}\cos(o(\mathbf{x})) \\ -\sin(o(\mathbf{x}))\end{bmatrix}. \tag{4}$$

2.2. Azimuth Consensus. Let Ω be the local circular region for which $\mathbf{x}_0 = [x_0, y_0]^{\mathrm{T}}$ is its center and its radius is λ in the image $\|\vec{f}(\mathbf{x})\|_{\mathrm{NMS}}$ and define N that is an assembly consisting of nonzero intensity value, that is,

$$\mathrm{N} = \left\{\mathbf{x} \mid \left\|\vec{f}(\mathbf{x})\right\|_{\mathrm{NMS}} > 0, \mathbf{x} \in \Omega, \mathbf{x} \neq \mathbf{x}_0\right\}. \tag{5}$$

The set N$'$ is defined as the sub-pixel position of all points in N. For each point, let $\mathbf{x} = [x, y]^{\mathrm{T}} \in \mathrm{N}$, and then compute the angle $s_{x_0}(\mathbf{x})$ of its corresponding center point $\mathbf{x}_0$, that is,

$$s_{\mathbf{x}_0}(\mathbf{x}) = \begin{cases} \tan^{-1}\left(\dfrac{y' - y_0}{x' - x_0}\right), & x' > x_0, y' > y_0 \\ \tan^{-1}\left(\dfrac{y' - y_0}{x' - x_0}\right) + 360^\circ, & x' > x_0, y' < y_0 \\ \tan^{-1}\left(\dfrac{y' - y_0}{x' - x_0}\right) + 180^\circ, & x' < x_0 \\ 90^\circ, & x' = x_0, y' > y_0 \\ 270^\circ, & x' = x_0, y' < y_0, \end{cases} \tag{6}$$

where x' and y' are the coordinates of sub-pixel position $\mathbf{x}' = [x', y']^{\mathrm{T}} (\mathbf{x}' \in \mathrm{N}')$. $\Delta_{x_0}(\mathbf{x})$ is defined as the angle between $s_{x_0}(\mathbf{x})$ and $o^{\perp}(\mathbf{x})$:

$$\Delta_{x_0}(\mathbf{x}) = \sin^{-1}\left(\left|\sin\left(s_{x_0}(\mathbf{x}) - o^{\perp}(\mathbf{x})\right)\right|\right). \tag{7}$$

Formula (7) ensures that the range of the angle $\Delta_{x_0}(\mathbf{x})$ belongs to $[0^\circ, 90^\circ]$. If a certain point $\mathbf{x} = [x, y]^{\mathrm{T}}$, which belongs to the set N defined by the formula (9), satisfies the formula (12) condition, and then $\mathbf{x}$ has azimuth consensus for $\mathbf{x}_0$(ε is a choosing angle threshold):

$$\Delta_{x_0}(\mathbf{x}) < \varepsilon. \tag{8}$$

From the geometrical presentation of azimuth consensus, we find that if point $\mathbf{x}$ is located in one of the straight lines that cross over the center $\mathbf{x}_0$, then $\Delta_{x_0}(\mathbf{x}) = 0$; otherwise $\Delta_{x_0}(\mathbf{x}) > 0$. Azimuth consensus consists of the relative position and the edge orientation information, so it is able to filter out the distortion of noise and unrelated edge points effectively.

2.3. Junction Point Classification and Branch Edge Grouping. The set $\mathrm{M} = \{\mathbf{x} \mid \Delta_{x_0}(\mathbf{x}) < \varepsilon, \mathbf{x} \in \mathrm{N}\}$ consists of all points in N that satisfy the azimuth consensus constraints, and the set M$'$ is a set of sub-pixel locations corresponding to M. From (6), the range of angles $s_{x_0}(\mathbf{x})$, which is relative to the center $\mathbf{x}_0$, of the points in M belongs to $[0^\circ, 360^\circ)$. The points in M are classified based on the distribution of angles to determine all the junction branches of $\mathbf{x}_0$. Simultaneously, based on the classified results of the junction point classification algorithm, which is shown in Algorithm 1, the algorithm decides whether $\mathbf{x}_0$ is a junction point or not.

In Algorithm 1, to eliminate the wrap-around effect of the angles and ensure that the junction points corresponding to the angles close to 0° and 360° are divided into the same

```
Input: M = $\{x_1, x_2, \ldots, x_m\}$, the dip angle set $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ corresponding to M, and the threshold angle $\tau$
Output: The classification result set E of branch edges' points
Initialize: Set the set E = NULL
Main steps of algorithm:
(1) Define a $m \times m$ matrix H and initialize H = null; and then compute all the elements of the matrix
    according to the equation
    $\mathrm{H}(i, j) = \begin{cases} \cos(\phi_i - \phi_j), & \text{if } \cos(\phi_i - \phi_j) \geq \cos(\tau), \\ \text{null}, & \text{otherwise}, \end{cases} \quad 1 \leq i < j \leq m$
(2) Calculate the position of the maximal value of H; assuming the maximum is at $(i, j)$, set $\mathrm{H}(i, j)$ = null,
    $\upsilon_1 = \sin(\varphi_i) + \sin(\varphi_j)$ and $\upsilon_2 = \cos(\varphi_i) + \cos(\varphi_j)$, and then, compute the classification center $\varphi$
    $\phi = \begin{cases} \tan^{-1}\left(\frac{\upsilon_1}{\upsilon_2}\right), & \upsilon_1 > 0, \upsilon_2 > 0 \\ \tan^{-1}\left(\frac{\upsilon_1}{\upsilon_2}\right) + 360^\circ, & \upsilon_1 > 0, \upsilon_2 < 0 \\ \tan^{-1}\left(\frac{\upsilon_1}{\upsilon_2}\right) + 180^\circ, & \upsilon_2 < 0 \\ 90^\circ, & \upsilon_1 > 0, \upsilon_2 = 0 \\ 270^\circ, & \upsilon_1 < 0, \upsilon_2 = 0 \end{cases}$
(3) Update $\upsilon_1, \upsilon_2$ and $\varphi$ by all angles that satisfy the condition $\cos(\varphi_k - \varphi) \geq \cos(\tau)$ $(k = 1, 2, \ldots, m)$ searched
    from the set $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$;
(4) Repeat step (3) until the angle set that satisfies the condition $\cos(\varphi_k - \varphi) \geq \cos(\tau)$ never change and place
    the junction points corresponding to the angle set into E;
(5) Repeat steps (2), (3) and (4) until all the elements of H are null, and then return E.
```

Algorithm 1: Junction point classification algorithm.

branch edge, we construct the matrix H by the cosine value of the D value between the two angles. By executing the algorithm in Algorithm 1, all junction edge points E of the current point $\mathbf{x}_0$ are computed. If $\mathbf{x}_0$ is a valid junction point, then E must satisfy the following conditions.

(i) The size of set E, that is, the number of branch edges, is larger than 2.

(ii) If the size of E is equal to two, the intersection angle must be larger than a fixed threshold to avoid *Colinearity*.

(iii) The junction and edge points in its local neighborhood area often violate the azimuth consensus constraints. So for every junction edge point, the gradient magnitude of all pixels that are in the Bresenham path [26] between $\mathbf{x}_0$ and its closest point is nonzero.

Using the above-described detection algorithm, many junction points are often detected in the neighborhood of the actual junction points. A response function as (13) is to determine the actual location of the junction points:

$$c(\mathbf{x}_0) = \sum_i \left(\prod_{x \in \mathbf{X}_i} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\Delta_{\mathbf{x}_0}(\mathbf{x})\right)^2}{2\sigma^2} \right) \cdot \left\| \overrightarrow{f}(\mathbf{x}) \right\|_{\mathrm{NMS}} \right), \tag{9}$$

where $\mathbf{X}_i$ is a set composed of the ith branch points and σ is the standard deviation of the Gaussian function.

3. Accurate Junction Point Localization and Characterization

3.1. Accurate Junction Point Localization. To improve the accuracy of the location of junction points, we compute the accurate location of junction points based on the minimal distance criterion [6, 27]. Assume that $\mathbf{x}_0$ is the integer-valued location of a junction point, that $\mathbf{x}_e$ is its related junction edges' point, and that $\mathbf{x}'_e$ is the sub-pixel location corresponding to $\mathbf{x}_e$. From (1), the gradient of $\mathbf{x}_e$ is $\overrightarrow{f}(\mathbf{x}_e)$, so the equation of line crossover $\mathbf{x}'_e$ is represented by

$$\overrightarrow{f}(\mathbf{x}_e)^{\mathrm{T}} \left(\mathbf{x} - \mathbf{x}'_e\right) = 0. \tag{10}$$

The optimal locations of the junction points satisfy the condition that the total weighted distance from all line segments is the shortest, that is,

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} \sum_{\mathbf{x}_e \in \mathrm{E}} \left(D_{\mathbf{x}_e}(\mathbf{x})\right)^2 \\ &= \arg\min_{\mathbf{x}} \sum_{\mathbf{x}_e \in \mathrm{E}} \left| \overrightarrow{f}(\mathbf{x}_e)^{\mathrm{T}} \left(\mathbf{x} - \mathbf{x}'_e\right) \right|^2 \\ &= \arg\min_{\mathbf{x}} \sum_{\mathbf{x}_e \in \mathrm{E}} \left(\mathbf{x} - \mathbf{x}'_e\right)^{\mathrm{T}} \overrightarrow{f}(\mathbf{x}_e) \overrightarrow{f}(\mathbf{x}_e)^{\mathrm{T}} \left(\mathbf{x} - \mathbf{x}'_e\right) \\ &= \arg\min_{\mathbf{x}} \left(\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - 2\mathbf{x}^{\mathrm{T}}\mathbf{b} + \mathbf{c}\right), \end{aligned} \tag{11}$$

where **A**, **b**, and **c** are shown as the following equation (12), respectively:

$$\begin{aligned}\mathbf{A} &= \sum_{\mathbf{x}_e \in E} \overrightarrow{f}(\mathbf{x}_e)\overrightarrow{f}(\mathbf{x}_e)^{\mathrm{T}},\\ \mathbf{b} &= \sum_{\mathbf{x}_e \in E} \overrightarrow{f}(\mathbf{x}_e)\overrightarrow{f}(\mathbf{x}_e)^{\mathrm{T}}\mathbf{x}'_e,\\ \mathbf{c} &= \sum_{\mathbf{x}_e \in E} \mathbf{x}_e'^{\mathrm{T}}\overrightarrow{f}(\mathbf{x}_e)\overrightarrow{f}(\mathbf{x}_e)^{\mathrm{T}}\mathbf{x}'_e.\end{aligned} \quad (12)$$

To minimize formula (11), the optimal location of junction point $\mathbf{x}^*$ is determined by taking the derivative of the right-side function with respect to $\mathbf{x}$ and setting it to zero, yielding $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$.

3.2. Accurate Branch Edge Orientation and Characterization. After calculating the accurate location of junction point $\mathbf{x}^*$, update the current center location $\mathbf{x}_0$ as $\mathbf{x}^*$, and then, use (6), and compute the dip angles of branch edges points that fulfill the azimuth consensus constraints. With the dip angle set of a certain junction point labeled as $\{\theta_1, \theta_2, \ldots, \theta_u\}$, the optimal orientation of this branch edge is described as

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{u} \cos(\theta - \theta_i). \quad (13)$$

In (13), $\sum_i \cos(\theta - \theta_i)$ is considered to be the object function for the following reasons: first, the distribution of the edge angles is relatively centered on a small area, so it decreases the search range of the optimal solution for the best angle; second, the cosine function is not susceptible to the sign of the D-value between two angles; third, the cosine function can eliminate the warp-around effect and map the angles close to 0° and 360° into the same value.

Next, calculate the value of $\sum_i \cos(\theta - \theta_i)$ according to every value θ, find the maxima from a series of $\sum_i \cos(\theta - \theta_i)$, and set the maxima to θ^*. After executing the above steps for the branch edges corresponding to all the junction points, the characterization description set for all junction points is created as

$$J = \left\{J_i \mid J_i = \left\langle x_i, y_i, \theta^*_{i1}, \theta^*_{i2}, \ldots, \theta^*_{im_i}\right\rangle, i = 1, 2, \ldots, n\right\}, \quad (14)$$

where n is the number of junction points detected, m_i is the number of branch edges corresponding to a junction point, and every element J_i, which is represented as $m_i + 2$ tuple, describes the ith junction point location and branch edges angle information corresponding to the ith junction point.

4. Experimental Results Comparison and Analysis

4.1. Experiment Design and Parameter Setting. The experiments included four parts: comparisons with the Forstner, CPDA, and JUDOCA algorithms in terms of the location accuracy, junction edge orientation accuracy, contrast changes, and the impact of noise. We chose these algorithms to be compared with our proposed method because of the error control and local optimization mechanism of these algorithms.

Table 1: Other experimental parameters setting.

Public parameter	Value
Standard deviation of Gaussian partial derivatives	1.5
Radius of nonmaxima suppression	3
Range of branch edges' angle (JUDOCA and our proposed method)	(30°, 150°)

Note that in the accuracy experiment, because CPDA is unable to detect all the junction points using the default parameter values, during the edge extraction step, the high and low thresholds are modified to be 0.2 and 0.05, respectively, and the gap connection length is set to two, while the other three algorithms maintain their default values. In addition, in the following experiments, the parameters shown in Table 1 are equal, the other parameters use default values.

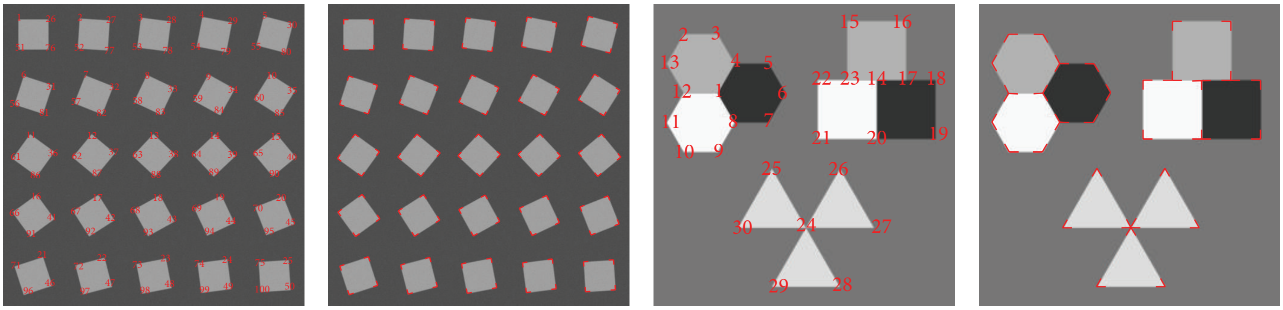
4.2. Accuracy of the Location of the Junction Points and the Orientation of the Branch Edges. To quantify the precision of our chosen methods, the ground truth data are required for all the locations of the junction points and the orientations of the branch edges. Therefore, we construct two artificial images for testing (as shown in Figures 2(a) and 2(c), where the digits in the images number represent the junction points in consecutive order).

The First Test Image: quadrate image [28] of size 500 ∗ 500, including 25 51 ∗ 51 square boxes, as shown in Figure 2(a). These 25 squares have the following characteristics: (1) the four edges of the 1st square (the square in the upper left corner) are parallel with the corresponding horizontal and vertical lines; (2) the nth square is rotated $3.6(n-1)$ degrees ($n = 1, 2, \ldots, 25$) along the 1st square in a clockwise direction; (3) the distance between two centers of adjacent squares is 100 pixels; (4) the center coordinates of the 1st square are (51, 51). The other test image: polygon image [28] of size 256 ∗ 256, including nine regular polygons as shown in Figure 2(c). These 9 polygons have the following characteristics: (1) the height of all regular polygons is 50 pixels; (2) the adjacent polygons form the public edges and vertices; (3) the coordinates of the junction points labeled as 1, 14, and 24 are (55, 76), (190, 65), and (130, 190), respectively.

Forstner, CPDA, JUDOCA, and our proposed algorithm are used to determine the junction points from the two images described above and compared with the known baseline data; the smaller the error of the result, the higher the detection accuracy. The formula of the location error of the junction points is defined as

$$\mathrm{LE} = \sqrt{\left(x - x_g\right)^2 + \left(y - y_g\right)^2}, \quad (15)$$

where (x, y) and (x_g, y_g) are the location of junction points extracted by the above-mentioned algorithms and the actual-location of junction points, respectively.



(a) (b) (c) (d)

Figure 2: The results of junction points and junction branches detection for two images: quadrate image and polygon image. (a) Quadrate image. (b) The junction detection result of our proposed method. (c) Polygon image. (d) The junction detection result of our proposed method.

Table 2: The mean error of the junction points' position among the Forstner algorithm, CPDA, JUDOCA, and our proposed method for the quadrate image and polygon image (unit: pixel).

	Forstner	CPDA	JUDOCA	Proposed method
Quadrate	**0.2437**	1.3270	0.6525	0.2876
Polygon	0.4046	1.4206	1.4222	**0.3475**

The location error curves for the above two test images determined by using the four detection operators are shown in Figure 3, where the horizontal axes represent the labeled junction points and the vertical axes describe the location error of the junction points determined by formula (15). The mean location error of all the junction points is listed in Table 2. The performance of the Forstner algorithm is the best option; for location accuracy, our proposed method can maintain the error to under-one pixel, just as the Forstner algorithm does, and achieve sub-pixel location accuracy (as shown in Figures 2(b) and 2(d)). The error of CPDA and JUDOCA are both larger than that of the Forstner algorithm and the mean error is greater than one pixel. For some junction points in Figure 2(c), the error of CPDA is even over 3 pixels. This is why our proposed algorithm introduces the same accurate location method to detect junction points and extract their accurate locations, while CPDA and JUDOCA only extract integer-valued junction point locations and cannot achieve sub-pixel location accuracy because the edge orientation information is unused to optimize junction point location.

4.3. Noise Impact. In this section, we test the impact of noise on our proposed algorithm. Random noises are added to the original image, and all operations are applied to this noisy version. Figure 4 from row 1 to row 2 shows the images combined with 1% and 1.2% noise, respectively, and each noisy image includes 26 junction points. When the test image includes the noise, the detection result often contains both the correct junction points and a certain number of pseudopoints (i.e., false alarms).

Assuming the number of the real junction points in the test image is N_g and the number of the detected junction points is N_t, which includes N_c correct junction points, we can compute the recall rate ($\mathrm{RR} = N_c/N_g$) and precision rate ($\mathrm{PR} = N_c/N_t$) to examine the operator performance. In the experiment, we chose contrast measure criterion combined with RR and PR [29]:

$$\mathrm{ACU} = \frac{\mathrm{RR} + \mathrm{PR}}{2} \times 100\%. \tag{16}$$

The results of the four algorithms for the two noisy test images are shown in Figure 4 columns from (a) to (d). The results for N_t, N_c, RR, PR, and ACU determined from these test images are listed in Table 3. The experimental results indicate that RR of Forstner operator is the highest. However, the Forstner operator ignores the more stable edge information in the local area, thereby leading to the extraction of more pseudojunction points; therefore, PR is the lowest, and ACU only is approximately 50%. CPDA exhibits strong robustness to the impact of noise and results in a higher PR and a value of ACU that is able to reach 70%. This result is mainly because CPDA extracts junction points by using discrete curvature estimates and by using fracture-edge connectivity technology.

However, the noises can lead to parted branch edges and thereby cause RR to decrease. The robustness of JUDOCA to noise is weaker than that of CPDA, especially its ACU, which is under high noise conditions and reduced by half, compared to the value of ACU under low noise conditions. PR and ACU of our method are the highest among the four methods; that is, it has the best robustness to the impact of noise. This excellent performance using our method is due to the introduction of azimuth consensus constraints to filter out the noise and the lack of a strict requirement for a connective path of branch edges.

4.4. Experiment Results for SAR Images. This experiment is different from the first four experiments on artificial test images, as it is usually very difficult to extract the ground truth data from the natural images. This difficulty is primarily because junction point detection for SAR images is not only relative to the local structural pattern but is also often related to the observation view and the observer's subjective judgment [14]. Therefore, in this section, we adopt

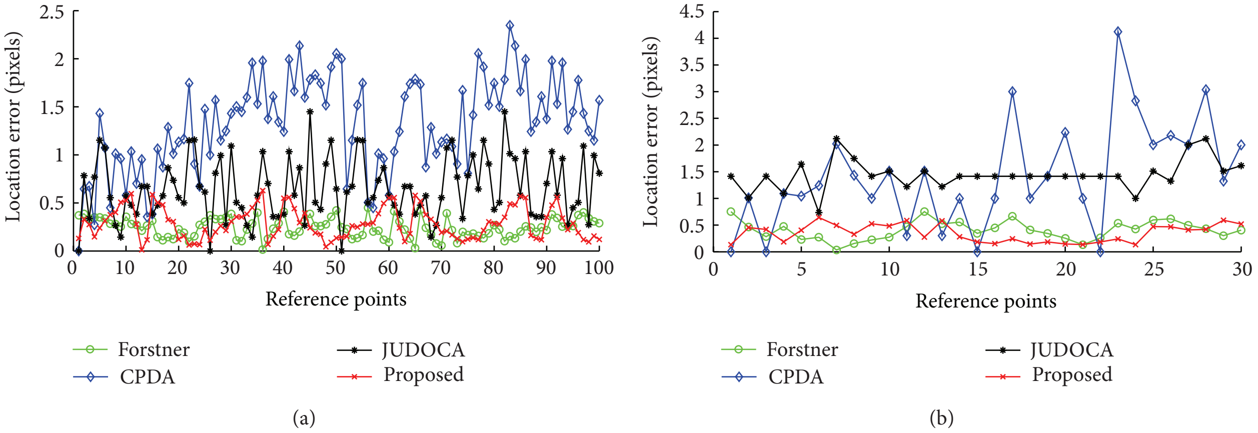
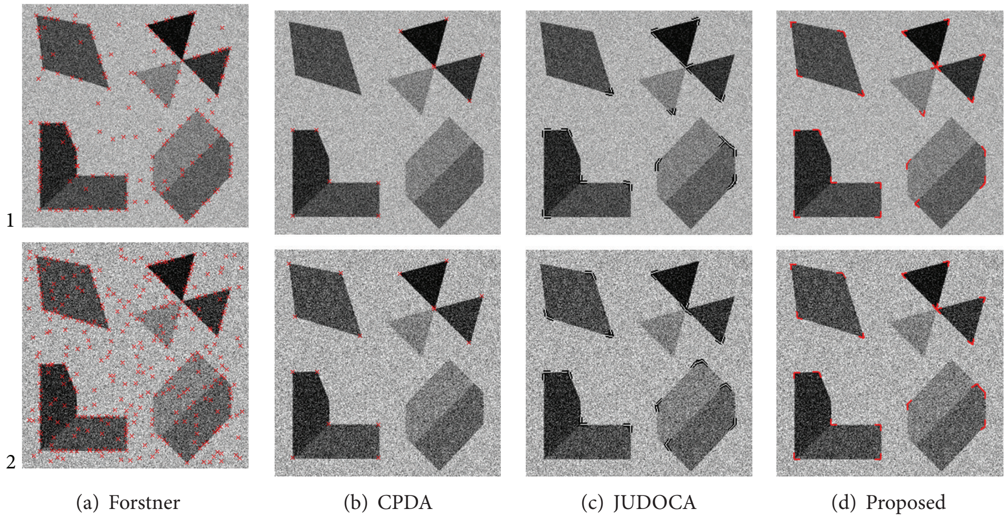


FIGURE 3: Location accuracy contrast among the Forstner algorithm, CPDA, JUDOCA, and our proposed method for detected junction points. (a) The horizontal axis represents the labeled number of 1 ~ 100 junction points in Figure 2(a) and the vertical axis describes the location error of the junction points detected by the Forstner algorithm, CPDA, JUDOCA, and our proposed method. (b) The horizontal axis represents the labeled number of 1 ~ 30 junction points in Figure 2(c) and the vertical axis describes the location error of the junction points detected by the Forstner algorithm, CPDA, JUDOCA, and our proposed method.



(a) Forstner (b) CPDA (c) JUDOCA (d) Proposed

FIGURE 4: Junction points detection results under different noise conditions. From row 1 to row 2, the random noises added is 1% and 1.2%, respectively. The results of the four methods are shown as the images from column (a) to column (d).

the qualitative evaluation criterion and choose two SAR images, which have different revolutions (as shown in Figure 5). The results of the two images are shown in Figures 5(a) and 5(b), respectively. The results in Figure 5(a) indicate that, for some obvious junction points, JUDOCA missed more points than our proposed method and it lost a part of the branch edges; the results for the SAR image in the City of Maoming are shown in Figure 5(b) and indicate that the two algorithms achieve comparable detection results.

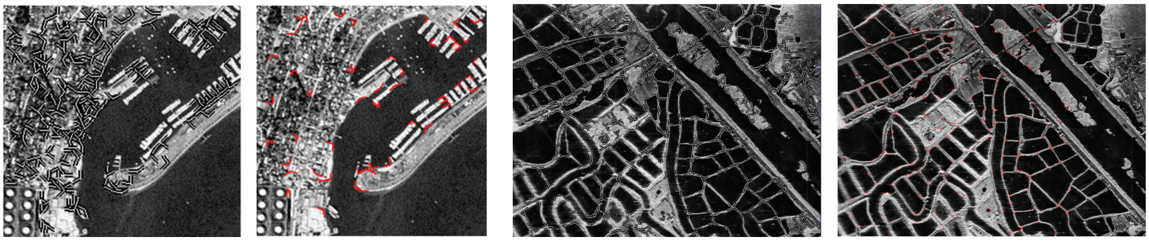
Note that under the situation consisting of massive noises, JUDOCA extracts more pseudojunction points around the circumference region than our method (especially in Figure 5(a)), but the points in the circle easily introduce significant location error because they are inappropriately detected as junction points. Thus, during the detection of a junction point, JUDOCA chooses and filters out the branch edges and noise impact only by the connective paths. By contrast, our proposed algorithm can effectively filter out the noise impact and the circumference points by using azimuth consensus and extract more useful structural information.

5. Conclusion

This paper presents a novel method for junction point detection to detect junctions accurately in SAR images. The proposed algorithm uses an azimuth consensus to filter out the impact of noise and pseudo-junction points, such as those from a homogeneous region and 1D-edge points. The experimental results demonstrate that our proposed algorithm

Table 3: Performance of four methods with different levels of noise.

	Forstner					CPDA					JUDOCA					Proposed				
	N_t	N_c	RR	PR	ACU	N_t	N_c	RR	PR	ACU	N_t	N_c	RR	PR	ACU	N_t	N_c	RR	PR	ACU
Image 1	173	24	0.92	0.14	53%	12	11	0.42	0.92	67%	15	14	0.54	0.93	74%	21	20	0.77	0.95	86%
Image 2	363	21	0.81	0.06	44%	16	14	0.54	0.88	71%	16	15	0.58	0.94	76%	19	19	0.73	1.00	87%



(a)

(b)

Figure 5: Comparison of the detection results for four SAR images [29]. (a) The detection results of the SAR image (size: 256 ∗ 256). Left: JUDOCA; right: proposed algorithm. (b) The detection results of the SAR image in the City of Maoming (size: 1358 ∗ 1036). Left: JUDOCA; right: proposed algorithm.

exhibits improved performance in the detection accuracy and is less susceptible to contrast change and noise impact than JUDOCA and CPDA. One area of our future work is to apply the junction detector to computer vision. Possible applications include content-based image registration, SAR image stitching, and multisensory image matching.

Acknowledgment

This research was partially supported by the National Natural Science Foundation of China under Grants 61170159 and 60902093.

References

[1] I. Golightly and D. Jones, “Corner detection and matching for visual tracking during power line inspection,” *Image and Vision Computing*, vol. 21, no. 9, pp. 827–840, 2003.

[2] I. Zoghlami, O. Faugeras, and R. Deriche, “Using geometric corners to build a 2D mosaic from a set of images,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 420–425, June 1997.

[3] Y. Liu, X. Zhang, and T. S. Huang, “Estimation of 3D structure and motion from image corners,” *Pattern Recognition*, vol. 36, no. 6, pp. 1269–1277, 2003.

[4] S. M. Smith and J. M. Brady, “SUSAN—a new approach to low level image processing,” *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[5] E. Rosten, R. Porter, and T. Drummond, “Faster and better: a machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2009.

[6] W. F. Förstner and E. Gülch, “A fast operator for detection and precise location of distict point, corners and centres of circular features,” in *Proceedings of the ISPRS Conference on Fast Processing of Photogrammetric Data*, pp. 281–305, 1987.

[7] C. Harris and M. Stephens, “A combined corner and edge detection,” in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.

[8] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Tech. Rep. CMU-CS-91-132, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pa, USA, 1991.

[9] U. Köthe, “Edge and junction detection with an improved structure tensor,” in *Proceedings of the DAGM-Symposium (DAGM ’03)*, vol. 2781 of *Lecture Notes in Computer Science*, pp. 25–32, 2003.

[10] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek, “Nonlinear structure tensors,” *Image and Vision Computing*, vol. 24, no. 1, pp. 41–55, 2006.

[11] F. Mokhtarian and R. Suomela, “Robust image corner detection through curvature scale space,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1376–1381, 1998.

[12] F. Mokhtarian and F. Mohanna, “Enhancing the curvature scale space corner detector,” in *Proceedings of the 8th Scandinavian Conference on Image Analysis*, pp. 145–152, Bergen, Norway, 2001.

[13] J. H. Han and T. Poston, “Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature,” *Pattern Recognition Letters*, vol. 22, no. 10, pp. 1133–1144, 2001.

[14] J. McDermott, “Psychophysics with junctions in real images,” *Perception*, vol. 33, no. 9, pp. 1101–1127, 2004.

[15] L. Parida, D. Geiger, and R. Hummel, “Junctions: detection, classification, and reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 687–698, 1998.

[16] F. Chabat, G. Z. Yang, and D. M. Hansell, “Corner orientation detector,” *Image and Vision Computing*, vol. 17, no. 10, pp. 761–769, 1999.

[17] M. Cazorla, F. Escolano, D. Gallardo, and R. Rizo, “Junction detection and grouping with probabilistic edge models and Bayesian A,” *Pattern Recognition*, vol. 35, no. 9, pp. 1869–1881, 2002.

[18] M. A. Cazorla and F. Escolano, "Two Bayesian methods for junction classification," *IEEE Transactions on Image Processing*, vol. 12, no. 3, pp. 317–327, 2003.

[19] R. Bergevin and A. Bubel, "Detection and characterization of junctions in a 2D image," *Computer Vision and Image Understanding*, vol. 93, no. 3, pp. 288–309, 2004.

[20] C. Perwass, "Junction and corner detection through the extraction and analysis of line segments," in *Proceedings of the 10th InternationalWorkshop Combinatorial Image Analysis (IWCIA '04)*, vol. 3322 of *Lecture Notes in Computer Science*, pp. 568–582, 2004.

[21] R. Elias and R. Laganière, "JUDOCA: junction detection operator based on circumferential anchors," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2109–2118, 2012.

[22] R. Elias, "Sparse view stereo matching," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1667–1678, 2007.

[23] R. Elias, "Enhancing sensor measurements through wide baseline stereo," *ELCVIA*, vol. 7, no. 3, pp. 36–53, 2008.

[24] H. Hajjdiab, R. Elias, and R. Lagani'ere, "Wide baseline obstacle detection and localization," in *Proceedings of the International Symposium on Signal Processing and Its Applications (ISSPA '03)*, vol. 1, pp. 21–24, 2003.

[25] P. Kovesi, *Invariant measures of image features from phase information [Ph.D. thesis]*, The University of Western Australia, 1996.

[26] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[27] T. Lindeberg, "Junction detection with automatic selection of detection scales and localization scales," in *Proceedings of the 1st International Conference on Image Processing (ICIP '94)*, vol. 1, pp. 924–928, Austin, Tex, USA, November 1994.

[28] http://www.ipb.uni-bonn.de/softwarefoerstneroperator/.

[29] http://www.crisp.nus.edu.sg/~research/tutorial/opt_int.htm# multispectral.

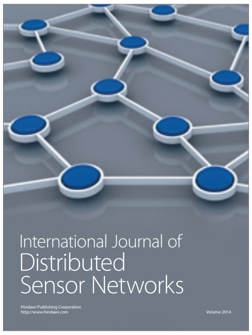
The Scientific World Journal

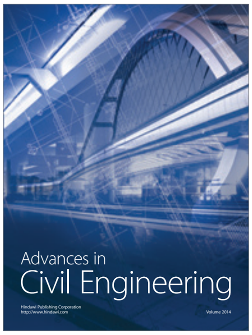
International Journal of
Rotating Machinery

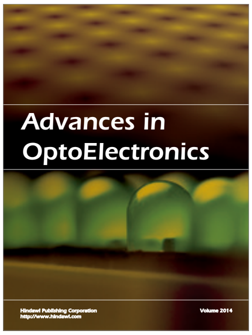
Journal of
Engineering

Journal of
Sensors

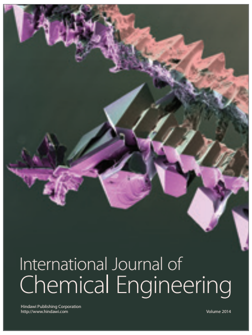
Advances in
Mechanical Engineering


International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering

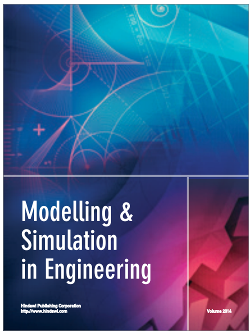

Advances in
OptoElectronics

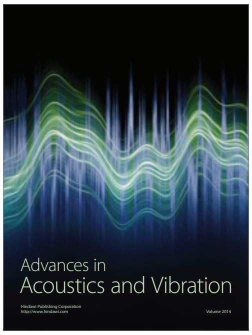
Journal of
Robotics


International Journal of
Chemical Engineering

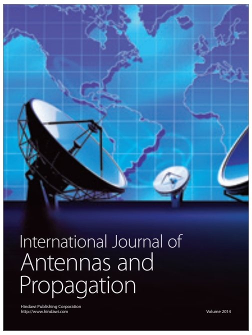
VLSI Design

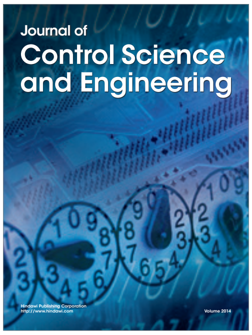
International Journal of
Navigation and Observation

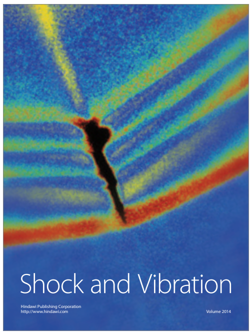

Modelling &
Simulation
in Engineering


Advances in
Acoustics and Vibration

Active and Passive
Electronic Components


International Journal of
Antennas and Propagation


Journal of
Control Science and Engineering


Shock and Vibration

Journal of
Electrical and Computer Engineering