

JUDOCA: JUnction Detection Operator Based on Circumferential Anchors

Rimon Elias, *Senior Member, IEEE*, and Robert Laganière, *Member, IEEE*

Abstract—In this paper, we propose an edge-based junction detector. In addition to detecting the locations of junctions, this operator specifies their orientations as well. In this respect, a junction is defined as a meeting point of two or more ridges in the gradient domain into which an image can be transformed through Gaussian derivative filters. To accelerate the detection process, two binary edge maps are produced; a thick-edge map is obtained by imposing a threshold on the gradient magnitude image, and another thin-edge map is obtained by calculating the local maxima. Circular masks are centered at putative junctions in the thick-edge map, and the so-called circumferential anchors or *CA points* are detected in the thin map. Radial lines are scanned to determine the presence of junctions. Comparisons are made with other well-known detectors. This paper proposes a new formula for measuring the detection accuracy. In addition, the so-called junction coordinate systems are introduced. Our operator has been successfully used to solve many problems such as wide-baseline matching, 3-D reconstruction, camera parameter enhancing, and indoor and obstacle localization.

Index Terms—Corner, corner detection, feature detection, interest-point detection, junction, junction detection.

I. INTRODUCTION

FEATURE detection in images is a fundamental problem in computer vision. In many vision systems, detecting features can be used as a first step toward a more complicated stream of processes. Hence, the reliability of such a step can greatly affect the overall outcome of the vision system.

During the past decades, many feature detectors were proposed. Those compete with each other in terms of localization accuracy, speed, and the information they provide.

Different types of features are found useful for recognition (e.g., edges and corners). An *edge* can be defined as a gradual transition in the intensity level. On the other hand, a corner may not have a single universal definition. A *corner* can be defined as a point with low self-similarity or a location where variations of the intensity function in both directions are high. Alternatively, a corner may be defined as an image point where two or more edges meet. Different corner detectors are built on top of different definitions, which lead to different outcomes that vary not only in the number of corners detected but in the location accuracy and speed of the process.

Manuscript received December 27, 2009; revised July 27, 2010, November 26, 2010, February 14, 2011, and April 11, 2011; accepted October 24, 2011. Date of publication November 11, 2011; date of current version March 21, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sharath Pankanti.

R. Elias and R. Laganière are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: relias@ieee.org; laganier@site.uottawa.ca).

Digital Object Identifier 10.1109/TIP.2011.2175738

Sometimes, the term *interest point* may be used instead of a *corner*. However, if the interest point is a position that can be robustly detected, an end of a line can be considered as an interest point as it may have local intensity maximum or minimum, or low self-similarity. Similarly, isolated points or curve points with local maximum may be detected as interest points.

It is worth noting that most of the corner or interest-point detectors are concerned with detecting only the locations of those points. Most of the time, the criterion upon which the detection is based is neglected (e.g., local dissimilarity based on intensity variations).

Junction detectors do the job of identifying both locations and the information used to make the identification. For example, if a corner is considered as the intersection of two edges, then a junction can be identified by its location and the orientations of the edges forming it. This results in richer information that can be important for recognition.

In this paper, we propose an operator that has the advantage of providing accurate junction localization and characterization. These feature characteristics can be exploited to make recognition more viewpoint invariant. The proposed operator has been successfully used in many applications; e.g., wide-baseline matching [5], [7], [16], 3-D reconstruction [9], camera parameter enhancing [6], [8], indoor localization [10], [11], and obstacle localization [13].

The rest of this paper is organized as follows: Section II presents a literature review of some junction and corner detectors. Section III presents the theory upon which we based our detector. Section IV describes the proposed algorithm. Section V introduces the so-called junction coordinate systems that can be used to facilitate junction transformation. Section VI presents experimental results testing various aspects of the proposed operator, while Section VII compares it to other well-known operators. Finally, Section VIII is a conclusion.

II. CORNER/JUNCTION DETECTION

There are many corner and junction detectors proposed in the literature. However, the most famous detectors frequently used are corner detectors, which have better complexity.

The Moravec corner detector [20] calculates the sum of squared differences (SSD) between a rectangular patch surrounding a candidate corner and other same-size patches around this patch. Then, the corner response is the smallest SSD obtained. Harris and Stephens [14] enhanced the idea by using a circular window to ensure isotropic response.

Smallest univalue segment assimilating nucleus (SUSAN) [25], [26] is used to detect corners and edges. It uses a circular mask with a central pixel that is called a *nucleus*. Pixel intensities within the mask are compared with the nucleus' intensity.

Hence, an area within the mask can be defined to have the same or similar intensity like the nucleus. This area is maximum in case the nucleus lies in a homogeneous region. It gets reduced at corners.

Rosten and Drummond [22], [23] proposed the Feature from Accelerated Segment Test (FAST) corner detector. This operator is designed to respond as fast as possible to corner detection requests. A circle is centered at putative corner points. The center point is identified as a corner if a contiguous set of circumferential pixels is all brighter than the central pixel plus a threshold or darker than the central pixel minus a threshold. Corners can be also detected where image edges have their maximum curvature. This is the basic idea for the curvature scale space (CSS) detector [19]. Another version is presented in [15]. Other techniques have been also based on scale evolution differences [28] and gradient correlation matrices [29].

Methods for characterizing junctions have been proposed. However, their computational complexity does not generally allow them to be used for junction detection and localization. They would instead be applied on selected interest points having been previously detected by some efficient corner detector as in [27], which is based on the successive application of rotated wedge-averaging filters around a given keypoint. The approach described in [21] detects junctions using a piecewise constant function that partitions a circular template into wedge-shaped regions. The detection and the characterization are achieved through energy minimization in order to find the best approximating junction model; however, the required computational effort remains important. In [24], a model representing a neighborhood by color distribution is used, which makes it suitable for color images.

Another category of approaches handles the detection by looking at the edge image. In [2], a junction detector uses the local properties of the gradient magnitude near junctions to reconnect broken edges. In [4], junctions are detected by measuring the rate of the orientation vector change along lines within a given neighborhood. The detector in [1] uses a two-pass Hough transform to identify points of intersection in an edge map. More discussion can be found in [23] and [25].

III. DETECTING JUNCTIONS

Junctions in images occur when several uniform regions join at one prominent point, i.e., the junction location. When a junction is produced by only two regions, this definition assumes maximal curvature at the junction.

Assuming a symmetrical transition between adjacent regions, then points of junctions will correspond to points where ridges in the gradient domain intersect. This suggests that the junction detection can be accomplished in the gradient domain. Moreover, we may assume that joining edges can be roughly approximated by straight lines in the vicinity of the junctions. The detector proposed in this paper is built around these ideas. It defines the vicinity of a putative junction as a circular neighborhood of radius λ that is centered at the point of the junction as illustrated in Fig. 1. In addition, two properties should hold.

- 1) The gradient values at the intersections between the circle bounding the junction's neighborhood and each of the

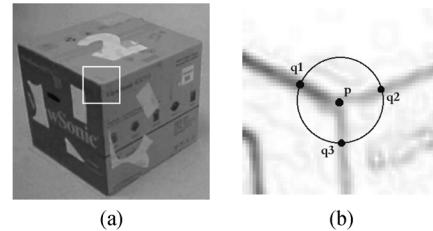


Fig. 1. (a) Y-junction. (b) Junction at p with three CAs q_1 , q_2 , and q_3 (superimposed on the gradient image).

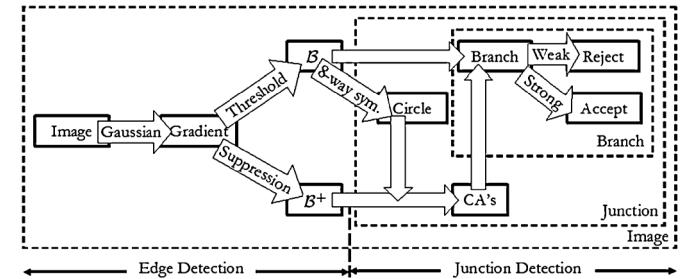


Fig. 2. Main steps of the JUDOCA algorithm.

junction radial lines must correspond to a local maximum in the direction of the gradient (a ridge point).

- 2) The value of the gradient along the ray on which each junction branch lies must always be greater than a predetermined value (although not necessarily on a ridge, in order to permit some minor deviations from the straight-edge model).

Consequently, a junction exists at a given image point if two or more rays obeying the aforementioned properties are emitting from this point. If only two rays are found, then the angle between both rays must not be 180° . The strength of such a junction should be proportional to the sum of the gradient magnitudes along each accepted ray.

Properties 1 and 2 form the basis upon which junction detection can be performed. An operator that makes use of these properties is presented in the next section. It is the junction detection operator based on circumferential anchors (JUDOCA).

IV. JUDOCA ALGORITHM

The main steps of the JUDOCA detector are shown in Fig. 2. The algorithm can be split into two main phases, i.e., edge detection, as discussed in Section IV-A, and junction detection, as discussed in Section IV-B.

Nonmaximal junction suppression is added after detecting junctions to help remove clouds of junctions throughout a user-specified circular area. This is controlled by parameter ϵ that defines the radius of this area.

A. Edge Detection

The first step in our algorithm is to detect edges as the proposed operator relies on the intensity gradient. This phase requires two parameters, i.e., σ and t_B , and it proceeds to two procedures.

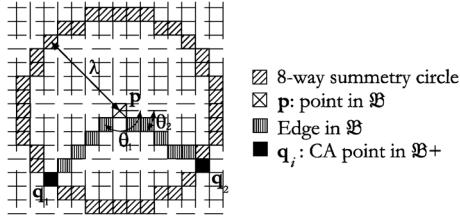


Fig. 3. Eight-way symmetry circle is placed at every point p in \mathcal{B} . CA point q is detected as the intersection between the circle and \mathcal{B}^+ .

- 1) Vertical and horizontal Gaussian derivative filters are applied to the input image \mathcal{I} . The size of the Gaussian kernel, which is specified by a variance parameter σ also offers the advantage of allowing edge detection at different scales. In other words, only more prominent edges are detected for larger σ values; however, if good localization is required, smaller σ values should be used.
- 2) The gradient magnitude is computed, and two binary images \mathcal{B} and \mathcal{B}^+ are created from it.
 - a) \mathcal{B} is created by imposing threshold $t_{\mathcal{B}}$ on the gradient image. Hence, \mathcal{B} contains thick edges.
 - b) \mathcal{B}^+ contains \mathcal{B} points that are local maxima in the direction of the gradient; i.e., nonmaximal suppression is applied to get thin edges.
- Those images are the basis of detecting junctions as discussed below.

B. Junction Detection

This phase takes as inputs the two binary images \mathcal{B} and \mathcal{B}^+ created in the previous phase. It also takes four parameters λ , s , θ_{\min} and θ_{\max} . This phase proceeds to six procedures.

- 1) For each point $p \in \mathcal{B}$, a circle of radius λ centered at this point is considered. The set of points $\mathcal{P}_{\mathcal{B}}$ on the circumference of the circle can be precomputed as a discretized circular contour of radius λ using the eight-way symmetry algorithm [12], as depicted in Fig. 3.
- 2) The list of candidate points on the circumference of this circle $\mathcal{Q}_{\mathcal{B}}^+ = \{q_1, q_2, \dots\} | \mathcal{Q}_{\mathcal{B}}^+ \in \mathcal{B}^+ \text{ (i.e., } \mathcal{Q}_{\mathcal{B}}^+ = \mathcal{P}_{\mathcal{B}} \cap \mathcal{B}^+\text{)}$ are obtained. These so-called *circumferential anchor* (CA) points are the extremities of potential radial lines for the putative junction. (The CA points are the black dots q_1 , q_2 , and q_3 that appear in Fig. 1 and the solid black squares q_1 and q_2 shown in Fig. 3.)
- 3) For each CA point $q_i \in \mathcal{Q}_{\mathcal{B}}^+$, the set of points along each line segment $\langle p, q_i \rangle$ (inclined at θ_i with respect to the horizontal direction) are scanned in \mathcal{B} using Bresenham's algorithm [3], [12]. To be accepted as a junction line, a continuous path must exist. If not, then this radial line is rejected, and the scanning operation is repeated with the next CA point.
- 4) Strength S_i of the radial line $\langle p, q_i \rangle$, i.e., $S_i(\langle p, q_i \rangle)$, is defined as the sum of the squared distances from $\langle p, q_i \rangle$ to the \mathcal{B}^+ points in the currently considered set. This strength is normalized by the length of this segment.
- 5) If the strength of a radial line is smaller than a predetermined threshold s , then this radial line is rejected. Otherwise, $\langle p, q_i \rangle$ becomes one of the branches of the putative junction at p .

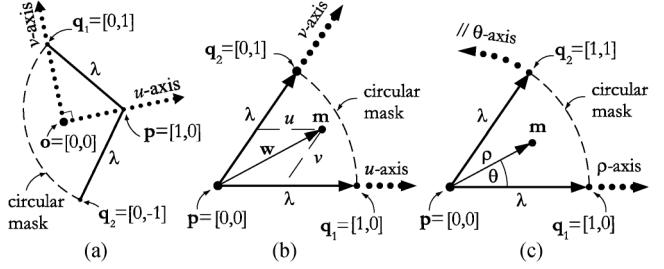


Fig. 4. (a) Local coordinates of a two-edge JUDOCA junction. (a) Cartesian coordinates. (b) Parametric coordinates. (c) Polar coordinates.

- 6) If the number of branches found at p is less than two, then there is no junction at this location. Otherwise, the orientations of the accepted radial lines are recorded, and strength $S(p)$ of this junction is set to the minimum across all strengths $S_i(\langle p, q_i \rangle)$. In other words

$$S(p) \Leftarrow \arg \min_i S_i(\langle p, q_i \rangle). \quad (1)$$

In case of the two-edge junction having edge orientations θ_1 and θ_2 , an extra step must be undertaken in order to ensure that this junction is not, in fact, a simple line. This can be verified by ensuring that $|\theta_1 - \theta_2| \neq 180 \pm \varepsilon$, where ε is a margin. In practice, a threshold is set on the maximal and minimal acceptable angles for the two-edge junctions. This is controlled by parameters θ_{\min} and θ_{\max} .

Additionally, the strength of the junction can be used to perform a nonmaximal-suppression postprocessing phase to eliminate clusters of junction that could arise, particularly if a permissive threshold $t_{\mathcal{B}}$ is used.

Note that different definitions for the junction strength could have been adopted. However, the one previously given is based on an assertion stating that a junction is as weak as its weakest branch. This definition also implies that weak N -junctions can be transformed into a stronger ($N - 1$)-junction.

V. JUNCTION COORDINATES

Here, we introduce the notion of junction coordinates. Each two-edge junction defines its own local two-axis coordinate system. Many coordinate systems (e.g., Cartesian, parametric, and polar) are applicable. Fig. 4(a)–(c) shows different coordinate systems for a two-edge junction. Of course, any m -edge junction can be split into a series of two-edge junctions. These coordinate systems can play a key role in defining corresponding junctions in multiview vision systems.

In all the junction coordinate systems below, the locations of the CA points q_1 and q_2 are $[x_p + \lambda \cos(\theta_1), y_p + \lambda \sin(\theta_1)]^T$ and $[x_p + \lambda \cos(\theta_2), y_p + \lambda \sin(\theta_2)]^T$, respectively, where the location of the junction p is $[x_p, y_p]^T$, λ is the radius of the JUDOCA neighborhood, and θ_1 and θ_2 are the orientation angles of the two edges forming the junction.

A. Cartesian Coordinates

Fig. 4(a) shows what can be regarded as the junction local Cartesian coordinate system. The origin of this system, i.e., o , is at $[(q_1/2) + (q_2/2)]$. Two vectors can be estimated, i.e., $\mathbf{u} =$

$\mathbf{p} - \mathbf{o}$ and $\mathbf{v} = \mathbf{q}_1 - \mathbf{o}$. In this case, $\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 = \lambda^2$. Hence, any point $\mathbf{m} = [x, y]^T$ can be expressed as $[u, v]^T$ such that

$$u = \frac{(\mathbf{m} - \mathbf{o}) \bullet \mathbf{u}}{\|\mathbf{u}\|^2} \quad \text{and} \quad v = \frac{(\mathbf{m} - \mathbf{o}) \bullet \mathbf{v}}{\|\mathbf{v}\|^2} \quad (2)$$

where \bullet represents the dot product and $\|\cdot\|$ represents the norm. Note that $\|\mathbf{u}\| = |\lambda \cos((\theta_2 - \theta_1)/2)|$ and $\|\mathbf{v}\| = |\lambda \sin((\theta_2 - \theta_1)/2)|$. The Cartesian coordinates produced for any point \mathbf{m} in the circular sector bounded by \mathbf{p} , \mathbf{q}_1 , and \mathbf{q}_2 have uv coordinates such that $u \in [-\lambda + \|\mathbf{u}\|, 1.0]$ and $v \in [-1.0, 1.0]$. In case of stereo images, if \mathbf{p} and \mathbf{p}' are corresponding junctions, then for each point \mathbf{m} in the vicinity of \mathbf{p} , there is point \mathbf{m}' in the vicinity of \mathbf{p}' that can be estimated as

$$\mathbf{m}' = \mathbf{o}' + u[\mathbf{p}' - \mathbf{o}'] + v[\mathbf{q}'_1 - \mathbf{o}'] \quad (3)$$

where \mathbf{o}' is the origin of the \mathbf{p}' coordinate system and \mathbf{q}'_1 is a CA point associated with \mathbf{p}' . Note that \mathbf{m} and \mathbf{m}' share the same uv coordinates, although they may have different xy coordinates.

B. Parametric Coordinates

As depicted in Fig. 4(b), the u - and v -axes are along the edges intersecting at the junction location $\mathbf{p} = [x_p, y_p]^T$, which is the origin of the system in this case. Two vectors can be estimated, i.e., $\mathbf{u} = \mathbf{q}_1 - \mathbf{p}$ and $\mathbf{v} = \mathbf{q}_2 - \mathbf{p}$. Note that $\|\mathbf{u}\| = \|\mathbf{v}\| = \lambda$. Consequently, any point $\mathbf{m} = [x, y]^T$ can be expressed as $[u, v]^T$ such that

$$\begin{aligned} u &= \frac{(\mathbf{u} \bullet \mathbf{v})(\mathbf{w} \bullet \mathbf{v}) - (\mathbf{v} \bullet \mathbf{v})(\mathbf{w} \bullet \mathbf{u})}{(\mathbf{u} \bullet \mathbf{v})^2 - (\mathbf{u} \bullet \mathbf{u})(\mathbf{v} \bullet \mathbf{v})} \\ v &= \frac{(\mathbf{u} \bullet \mathbf{v})(\mathbf{w} \bullet \mathbf{u}) - (\mathbf{u} \bullet \mathbf{u})(\mathbf{w} \bullet \mathbf{v})}{(\mathbf{u} \bullet \mathbf{v})^2 - (\mathbf{u} \bullet \mathbf{u})(\mathbf{v} \bullet \mathbf{v})} \end{aligned} \quad (4)$$

where $\mathbf{w} = \mathbf{m} - \mathbf{p}$ and \bullet represents the dot product. The parametric coordinates produced are normalized so that any point \mathbf{m} in the circular sector bounded by \mathbf{p} , \mathbf{q}_1 , and \mathbf{q}_2 has uv coordinates such that $u \in [0.0, 1.0]$ and $v \in [0.0, 1.0]$. In case of stereo images, if \mathbf{p} and \mathbf{p}' are corresponding junctions, then for each point \mathbf{m} in the vicinity of \mathbf{p} , there is a point \mathbf{m}' in the vicinity of \mathbf{p}' that can be estimated as

$$\mathbf{m}' = \mathbf{p}' + u[\mathbf{q}'_1 - \mathbf{p}'] + v[\mathbf{q}'_2 - \mathbf{p}']. \quad (5)$$

Note that \mathbf{m} and \mathbf{m}' share the same uv coordinates, although they may have different xy coordinates.

C. Polar Coordinates

In Fig. 4(c), the origin of the local polar system is at the location of junction \mathbf{p} . Coordinates $\rho\theta$ of point \mathbf{m} in the circular sector bounded by \mathbf{p} , \mathbf{q}_1 , and \mathbf{q}_2 are

$$\rho = \frac{\|\mathbf{m} - \mathbf{p}\|}{\|\mathbf{q}_1 - \mathbf{p}\|} \quad \text{and} \quad \theta = \left| \frac{\cos^{-1} \left(\frac{(\mathbf{m} - \mathbf{p}) \bullet (\mathbf{q}_1 - \mathbf{p})}{\|\mathbf{m} - \mathbf{p}\| \|\mathbf{q}_1 - \mathbf{p}\|} \right)}{\theta_2 - \theta_1} \right| \quad (6)$$

where \bullet represents the dot product, $\|\cdot\|$ represents the norm, and $\lambda = \|\mathbf{q}_1 - \mathbf{p}\|$. Note that $\rho \in [0.0, 1.0]$ and $\theta \in [0.0, 1.0]$. In case of stereo images, if \mathbf{p} and \mathbf{p}' are corresponding junctions,

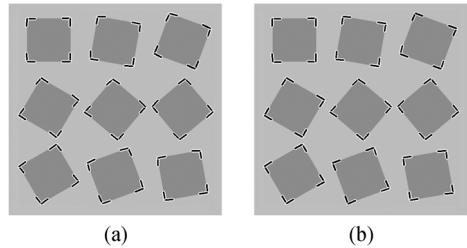


Fig. 5. JUDOCA operation is applied to a synthetic image of a rotating square having an intensity of 140 on a background of 190. The parameters used are $\sigma = 1$, $t_B = 10$, $s = 80\%$, $\theta_{\min} = 40^\circ$, $\theta_{\max} = 140^\circ$, and $\epsilon = 3$.

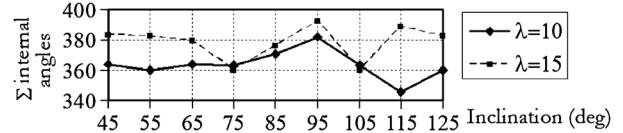


Fig. 6. Sum of internal angles of the rotating squares in Fig. 5 measured for different values of λ .

then for each point \mathbf{m} in the vicinity of \mathbf{p} , there is point \mathbf{m}' in the vicinity of \mathbf{p}' that can be estimated as

$$\mathbf{m}' = \mathbf{p}' + \begin{bmatrix} \rho \cos(\theta + \theta_1) \\ \rho \sin(\theta + \theta_1) \end{bmatrix}. \quad (7)$$

Note that \mathbf{m} and \mathbf{m}' share the same $\rho\theta$ coordinates, although they may have different xy coordinates.

VI. EXPERIMENTAL RESULTS

Here, we will test our operator against orientation accuracy, location accuracy, noise and smoothing impact, repeatability/consistency, and timing issues. We will show how the concept of junction coordinates can be utilized in transformation to facilitate junction matching. In the following section, we will compare our detector against well-known operators.

A. Orientation Accuracy

In order to test the sensitivity of JUDOCA to different rotation angles, we created the image shown in Fig. 5. The images contain a rotating square on top of a gray background. The step angle of rotation is 10° . The JUDOCA operation is applied to this image, and the results are shown Fig. 5(a) and (b) when $\lambda = 10$ and $\lambda = 15$, respectively.

The accuracy of the JUDOCA response to different orientations can be measured by summing the internal angles of each square that, ideally, should be 360° regardless of the rotation angle. Fig. 6 shows the sum for nine different square inclination angles. The inclination angle of a square is considered to be the inclination angle of the diagonal joining the upper left and lower right corners. The experiment is done twice for $\lambda = 10$ and for $\lambda = 15$. It is evident that the accuracy increases with larger values of λ . Moreover, this experiment shows that the operator tends to slightly overestimate the value of the angles. This bias can be explained by the smoothing effect of the Gaussian filter that blunts the tip of sharp corners.

In the same synthetic image, the bisector of any of the angles should ideally be one of the angles

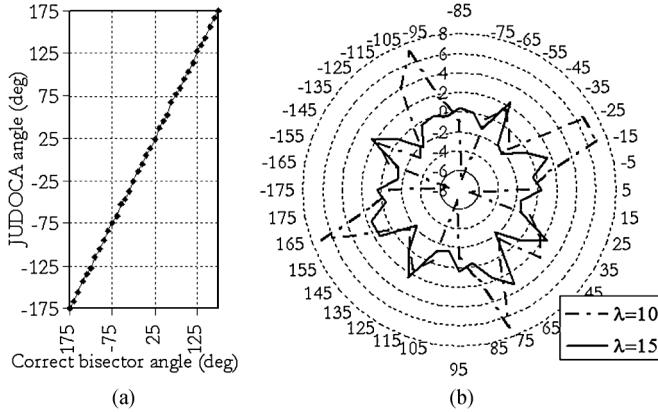


Fig. 7. (a) The angles measured by JUDOCA with respect to their ideal values. The parameters used are those of Fig. 5(b). (b) Starplots showing the differences measured between the JUDOCA angles and the ideal angles when $\lambda = 10$ and $\lambda = 15$ for the image shown in Fig. 5.

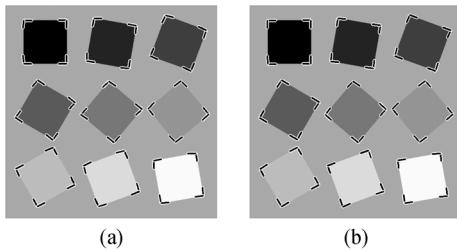


Fig. 8. “Squares” image. The intensities of squares range from 0 to 250. The parameters used are $\sigma = 1$, $t_B = 6$, $s = 80\%$, $\theta_{\min} = 40^\circ$, $\theta_{\max} = 140^\circ$, and $\epsilon = 3$. (a) $\lambda = 10$. (b) $\lambda = 15$.

$\{-175^\circ, -165^\circ, -155^\circ, \dots, 5^\circ, 15^\circ, \dots, 175^\circ\}$. Fig. 7(a) shows the angles measured by JUDOCA with respect to their ideal values.

The curve in Fig. 7(a) is close to the ideal curve, which should be a straight line joining points $[-175^\circ, -175^\circ]$ and $[175^\circ, 175^\circ]$. The differences between the real and ideal curves are shown as starplots in Fig. 7(b). The concentric circles represent difference levels in degrees, whereas radial lines represent inclination angles within range $[-180^\circ, 180^\circ]$. The error margin inferred for $\lambda = 15$ lies in interval $[-2.4^\circ, 3.2^\circ]$, while the interval changes to $[-6.9^\circ, 6.9^\circ]$ for $\lambda = 10$. This shows that the accuracy increases with larger values of λ .

The previous experiments are repeated for the “squares” image shown in Fig. 8. Fig. 8(a) shows the results when $\lambda = 10$, whereas Fig. 8(b) shows the results when $\lambda = 15$. Note that a strict t_B value of 6 (as opposite to 10 in the previous experiments) is used to detect edges where the foreground and background intensities get closer.

The differences between the JUDOCA angles and the ideal angles are shown in Fig. 9. The error margin inferred for $\lambda = 15$ lies in interval $[-2.4^\circ, 3.2^\circ]$, while the interval changes to $[-2.9^\circ, 6.9^\circ]$ for $\lambda = 10$. Again, this shows that the differences get smaller with larger values of λ .

B. Noise Impact

Random noise is added to the “squares” image, and the JUDOCA operation is applied to this noisy version. Fig. 10(a) shows the \mathcal{B} and \mathcal{B}^+ images combined where the effect of noise

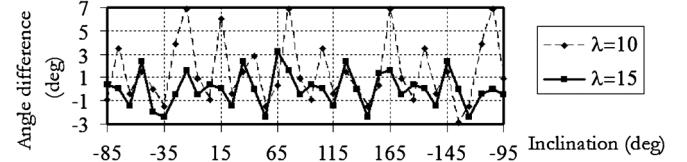


Fig. 9. Differences measured between the JUDOCA angles and the ideal angles when $\lambda = 10$ and $\lambda = 15$ for the image shown in Fig. 8.

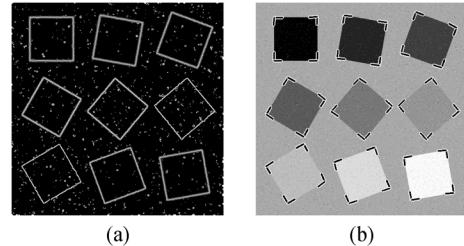


Fig. 10. JUDOCA operation is applied to the “squares” image after adding noise. The parameters used are $\sigma = 1.5$, $t_B = 4$, $\lambda = 15$, $s = 50\%$, $\theta_{\min} = 60^\circ$, $\theta_{\max} = 120^\circ$, and $\epsilon = 3$. (a) Edges detected. (b) Junctions detected.

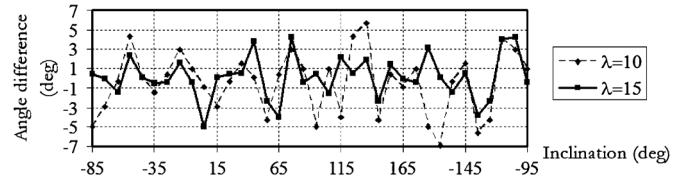


Fig. 11. Differences measured between the JUDOCA angles and the ideal angles when $\lambda = 10$ and $\lambda = 15$ for the noisy version of the “squares” image.

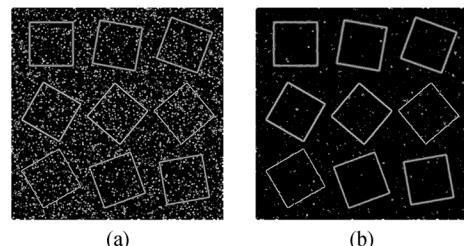


Fig. 12. Noise is reduced by using larger values of σ . (a) $\sigma = 1.2$. (b) $\sigma = 1.6$.

clearly appears. Fig. 10(b) shows the output of the operator. The angle differences in the case of the noisy image are shown in Fig. 11. The differences range from -6.9° to 5.7° when $\lambda = 10$ and from -5° to 4.2° when $\lambda = 15$.

Note that parameter σ plays an important role regarding the noise robustness of the operator at the edge detection step. This is illustrated in Fig. 12 where the detector has been tested on the previous noisy image. By increasing the value of σ , it has been possible to considerably reduce the impact of noise.

C. Smoothing Impact

In order to test the impact of smoothing on the outcome of the JUDOCA operator, we have applied a 5×5 Gaussian filter to the “squares” image for five times. The JUDOCA operation is performed afterward. The result is shown in Fig. 13(a). In this case, the obvious impact is in the edge detection step. Increasing the smoothing increases the width of edges in image \mathcal{B} , as shown

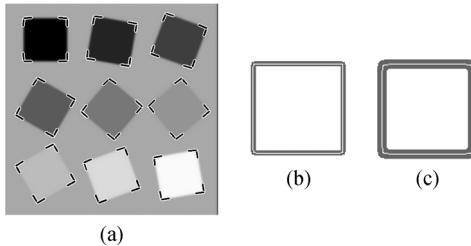


Fig. 13. “Squares” image smoothed by a 5×5 Gaussian filter applied five times. (a) The JUDOCA outcome. The parameters used are $\sigma = 1.0$, $t_B = 2$, $\lambda = 15$, $s = 75\%$, $\theta_{\min} = 40^\circ$, $\theta_{\max} = 140^\circ$, and $\epsilon = 2$. (b) B^+ is superimposed on B for the upper left square of the original image. (c) Response of the same square in the smoothed image.

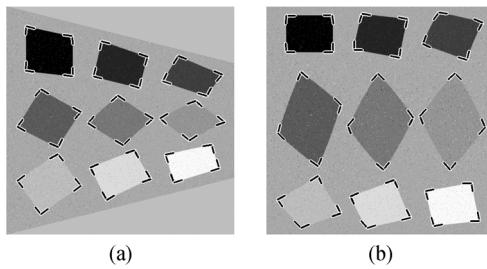


Fig. 14. JUDOCA junctions have been successfully detected when the noisy “squares” image is affine transformed and distorted. The parameters used are (a) $\sigma = 1.5$, $t_B = 4$, $\lambda = 15$, $s = 50\%$, $\theta_{\min} = 60^\circ$, $\theta_{\max} = 120^\circ$, and $\epsilon = 3$. (b) $\sigma = 2.2$, $t_B = 3$, $\lambda = 11$, $s = 60\%$, $\theta_{\min} = 60^\circ$, $\theta_{\max} = 120^\circ$, and $\epsilon = 3$.

in Fig. 13(b) and (c). Needless to say, increasing thickness results in more processing time as more pixels are investigated for the presence of junctions. More discussion on timing issues is included in Section VI-F.

D. Repeatability/Consistency

In [17] and [18], it is stated that corner locations and numbers should not get affected by noise, rotation, uniform or nonuniform scaling, and affine transformation. A criterion for testing the consistency of corner numbers is proposed as

$$\text{CCN} = 100 \times 1.1^{-|N_t - N_o|} \quad (8)$$

where CCN stands for “consistency of corner numbers,” N_o is the number of corners in the original image, and N_t is the number of corners in the transformed image.

In Section VI-A, we tested the effect of rotation, and in Section VI-B, we tested the effect of noise. According to (8), the CCN value of the JUDOCA outcome in both cases reaches 100% as the number of corners/junctions detected N_t is the same number of presumed/original corners/junctions N_o .

Affine transformation and nonaffine distortion are applied to the noisy “squares” image, and JUDOCA operation is performed. Fig. 14(a) and (b) show the results. Again, according to (8), CCN reaches 100%.

E. Number and Location Accuracy

A measure of the number accuracy is applied as [17], [18]

$$\text{ACU} = 100 \times \frac{\frac{N_a}{N_o} + \frac{N_o}{N_g}}{2} \quad (9)$$

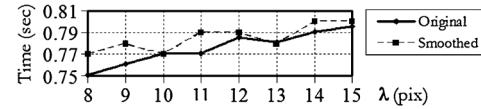


Fig. 15. Time increases with larger values of λ as junctions are detected for the “squares” image. Time also increases with the addition of smoothing.

where N_g is the number of corners in the ground-truth set \mathcal{N}_g (or the number of corners that ideally should be detected), N_o is the number of corners detected in the original image, and N_a is the number of corners that are detected and that belong to \mathcal{N}_g and \mathcal{N}_o at the same time (i.e., $N_a = \mathcal{N}_g \cap \mathcal{N}_o$).

If this measure of accuracy is applied to the previous results, each value of N_g , N_o , and N_a will be 36, giving an accuracy of 100%. Moreover, we measured the location differences as Euclidean distances between the results of Figs. 8(b) and 10(b). The average difference was 0.75 pixel.

F. Timing Issues

The complexity of the JUDOCA algorithm is $O(n)$ if n is the number of pixels. Generally, JUDOCA takes around 0.8 s to process a 640×480 image. Moreover, the response of the operator is almost linear with respect to the value of λ . As shown in Fig. 15, a difference of 46 ms is recorded between $\lambda = 8$ and $\lambda = 15$ when junctions are detected for the “squares” image.

As a comparison, we measured the time consumed for the smoothed version of the same image where the same parameters are used in this case. It is evident from Fig. 15 that the time increases with the addition of smoothing in general.

G. Junction Parameters

The choice of a parameter value depends on the potential effect of this parameter. For example, in case of noisy images, a large value of σ (greater than 1.0) reduces the effect of noise, and consequently, the erroneous detection of junctions will be reduced. On the other hand, smaller σ values (usually less than 1.0) are preferred in case of smoothed or blurred images so that correct junctions would not be missed.

Values greater than 10 for the edge parameter t_B result in neglecting not-very-strong edges. This leads to missing some junctions. On the contrary, decreasing the value of t_B results in avoiding edge discontinuity; however, decreasing this value severely toward 0 results in detecting weak edges (and hence, erroneous junctions may be identified).

Choosing large values for λ and s detects strong junctions only. Reducing those values increases the number of junctions at the cost of picking erroneous junctions for very small λ and s . In our experiments, ranges of 8–12 for λ and 50%–75% for s produce a reasonable outcome for most cases.

Finally, θ_{\min} and θ_{\max} affect the detection of junctions at points of maximum curvature. This choice becomes effective in cases of small values of s .

H. Junction Transformation

Utilizing any of the junction coordinate systems of Section V with a stereo pair of images can transform a region surrounding a junction into another region surrounding its corresponding junction. This transformation is useful in case of wide-baseline

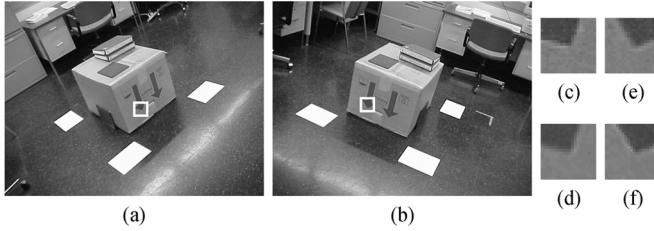


Fig. 16. (a) and (b) Original images with corresponding junctions identified. (c) Original left junction magnified. (d) Right junction transformed. (e) Original right junction magnified. (f) Left junction transformed.

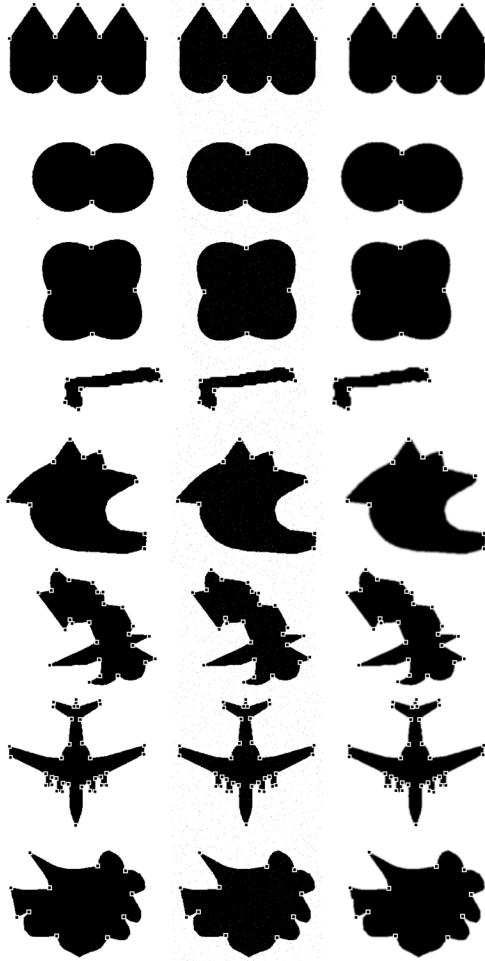


Fig. 17. JUDOCA operator is applied to different shapes [15]. The first column represents the original images. The second column includes the same images after adding 3% random noise. The third column consists of the same images after smoothing with a 5×5 Gaussian filter.

pairs where corresponding regions are most likely to be substantially different. Fig. 16(a) and (b) shows a wide-baseline pair of images where two corresponding junctions are identified. Fig. 16(c) and (e) shows the original regions. Utilizing the outcome of our operator, transformation can be done as shown in Fig. 16(d) and (f).

I. Response to Different Curvatures

In order to show how our operator performs with different curvature situations, JUDOCA is applied to a number of shapes,



Fig. 18. JUDOCA junctions: The parameters shown are σ , t_B , λ , and s . (a) 1, 7, 10, 75. (b) 1.8, 6, 15, 70. (c) 1.1, 7, 10, 60. (d) 1.1, 6, 10, 65.

as shown in Fig. 17. In all cases (except for the lowest shape), both CCN and ACU reach 100% (i.e., $N_o = N_t = N_a = N_g$). The numbers of corners/junctions detected are 9, 2, 4, 6, 10, 25, and 36, respectively. For the lowest shape, $N_o = N_t = N_a = 11$, while $N_g = 12$ (i.e., CCN = 100% and ACU = 95.83%).

Fig. 18 shows the results obtained when applying JUDOCA to more complex real-world images.

VII. COMPARISONS WITH OTHER OPERATORS

Here, we compare among JUDOCA, Harris/Plessey [14], SUSAN [25], [26], FAST [22], [23], and modified CSS [15]. The comparisons will be made in case of noisy and smoothed images. Moreover, the repeatability of features, the location accuracy, and the time consumed are compared.

A. Noise Impact

The operators under comparison are used with the noisy versions of the “squares,” “blocks,” and “lab” images (after adding 5%, 5%, and 3% random noise, respectively). The results are shown in Fig. 19. It is obvious that the Harris, SUSAN, and FAST operators do not have good response in the presence of noise. On the contrary, the modified CSS shows a better performance; however, some instances are incorrect. The response is better with JUDOCA where the Gaussian derivative filters play an important role to neutralize the noise effect.

B. Smoothing Impact

The 5×5 Gaussian-smoothed versions of the “squares,” “blocks,” and “lab” images are used with each of the operators under consideration. The results are shown in Fig. 20. It is clear that the JUDOCA operator could handle the presence of smoothing contrary to the other operators. In many instances, the FAST operator results in a cluster of corners at one location

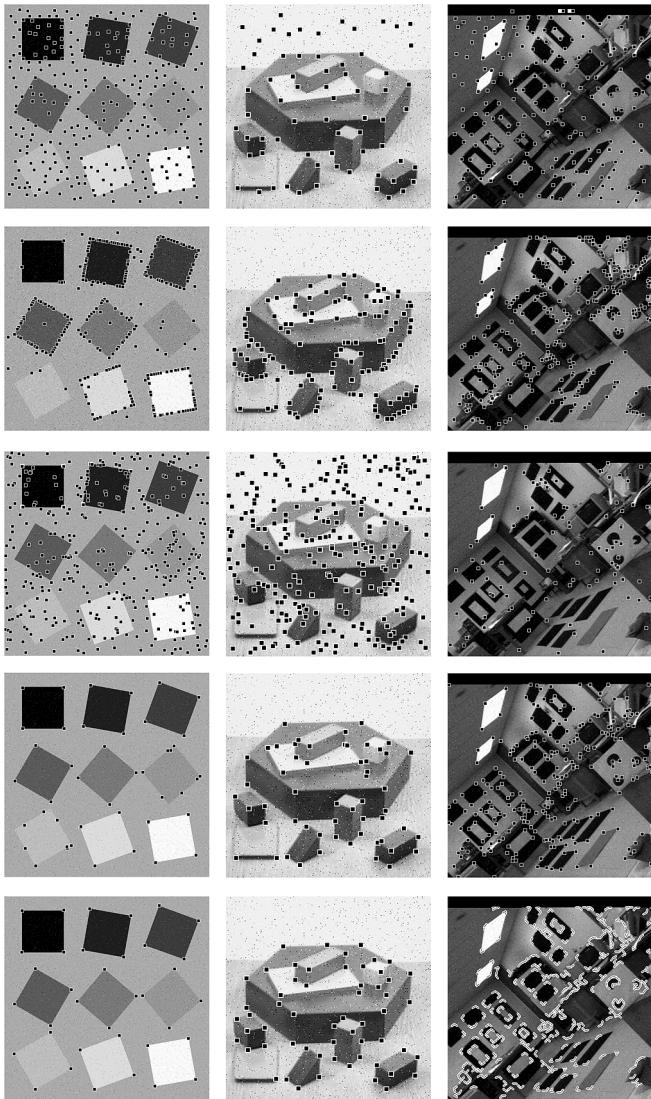


Fig. 19. Noisy “squares,” “blocks,” and “lab” images are operated on using the Harris (first row), SUSAN (second row), FAST (third row), modified CSS (fourth row) and JUDOCA (fifth row) detectors. The parameters are block size, aperture size, and k for Harris and thresholds for SUSAN and FAST. The parameters for the modified CSS detector are the minimum ratio of the ellipse axes, the maximum obtuse angle, the standard deviation, the high and low thresholds of the Canny detector, and the gap size. The JUDOCA parameters are σ , t_B , λ , and s . The values are as follows. Harris: (9, 5, 0.04), (9, 5, 0.02), and (15, 7, 0.05). SUSAN: (30), (25), and (35). FAST: (120), (100), and (100). Modified CSS: (1.5, 162, 3, 0.15, 0, 1), (1.5, 162, 3, 0.35, 0, 1), and (1.5, 162, 3, 0.35, 0, 1). JUDOCA: (1.5, 4, 15, 50), (1.5, 6, 10, 63), and (1.75, 6, 10, 67).

(after applying its nonmaximal step). Moreover, some features are missing in case of the modified CSS.

C. Repeatability/Consistency

The results of applying the operators under consideration to our original test images are shown in Fig. 21. To test the corner consistency, (8) is used with the noisy, smoothed, and original images. Table I shows that JUDOCA outperforms the other operators.

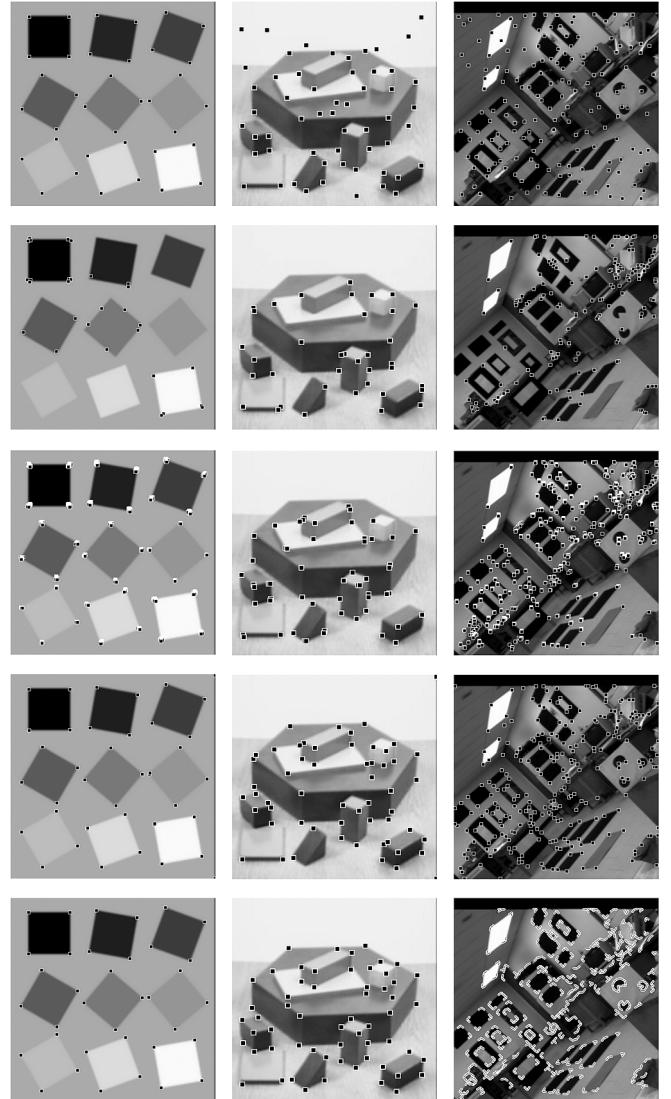


Fig. 20. Different detectors applied to the Gaussian-smoothed test images. The order and parameters are those listed in Fig. 19. The values are as follows. Harris: (9, 5, 0.05), (9, 5, 0.02), and (15, 7, 0.05). SUSAN: (3), (20), and (20). FAST: (2), (20), and (20). Modified CSS: (1.5, 162, 3, 0.1, 0, 1), (1.5, 162, 3, 0.35, 0, 1), and (1.5, 162, 3, 0.35, 0, 1). JUDOCA: (1, 2, 15, 75), (0.8, 7, 10, 60), and (0.65, 8, 10, 69).

D. Number and Location Accuracy

Equation (9) is applied to the outcome of each of the operators, and the results are shown in Table I. The number of corners in the ground-truth sets N_g are 36, 63, and 275 for the “squares,” “blocks,” and “lab” images, respectively. The results show the superiority of the JUDOCA operator with respect to the other operators. In order to visually compare the location accuracy between detectors, we overlaid the outcomes of JUDOCA and Harris operators [see Fig. 21 (top right and bottom right)] onto the same “lab” image, as shown in Fig. 22.

Notice that (9) does not specify if the corner detected at a small distance from the correct location (e.g., 2 pixels) should be considered a correct corner or not. Hence, we are proposing a modification to (9) that assigns a weight to each of the detected corners following the distance to its optimal location. Thus, each

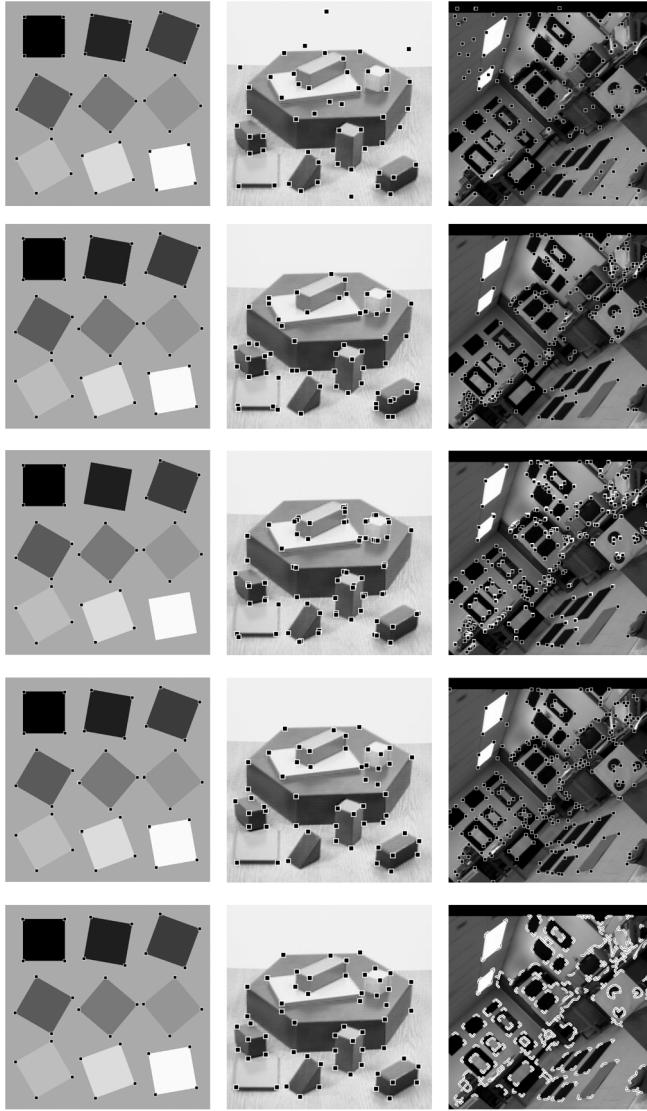


Fig. 21. Different detectors applied to the original test images. The order and parameters are those listed in Fig. 19. The values are as follows. Harris: (9, 5, 0.04), (9, 5, 0.02), and (15, 7, 0.05). SUSAN: (12), (25), and (30). FAST: (10), (30), and (35). Modified CSS: (1.5, 162, 3, 0.1, 0, 1), (1.5, 162, 3, 0.35, 0, 1), and (1.5, 162, 3, 0.35, 0, 1). JUDOCA: (1, 6, 15, 80), (1.1, 6, 10, 65), and (1.2, 7, 10, 60).

corner can contribute a value from 0.0 to 1.0 to N_a . Consequently, (9) can be rewritten as

$$M - ACU = 50 \times \left(\frac{\sum_{c \in \mathcal{N}_o \cap \mathcal{N}_g} c}{N_o} + \frac{\sum_{c \in \mathcal{N}_o \cap \mathcal{N}_g} c}{N_g} \right) \quad (10)$$

such that

$$c = \begin{cases} 1, & 0 \leq d \leq \sqrt{2} \\ 1 - ((d - \sqrt{2})/(\delta - \sqrt{2})), & \sqrt{2} < d \leq \delta \\ 0, & d > \delta \end{cases}$$

where d is the actual distance to the optimal location and δ is the maximum distance accepted. Our formula takes into account not only the number of detected corners but how far they are from the correct positions. Table II lists the modified ACU values ($M - ACU$) for the “squares” image when $\delta = 4$.

TABLE I
CCN AND ACU VALUES FOR DIFFERENT OPERATORS

Image	Operator	N_o	N_t	CCN	N_a	ACU
Squares (original)	Harris	36	-	-	36	100.00
	SUSAN	36	-	-	36	100.00
	FAST	32	-	-	32	94.44
	CSS (He)	40	-	-	36	95.00
	JUDOCA	36	-	-	36	100.00
Squares (noisy)	Harris	36	304	8.07E-10	33	91.67
	SUSAN	36	265	3.32E-08	25	69.44
	FAST	32	405	3.64E-14	10	29.51
	CSS (He)	40	46	56.45	35	92.36
	JUDOCA	36	36	100.00	36	100.00
Squares (smoothed)	Harris	36	36	100.00	36	100.00
	SUSAN	36	29	51.32	14	38.89
	FAST	32	599	3.39E-22	32	94.44
	CSS (He)	40	46	56.45	35	92.36
	JUDOCA	36	36	100.00	36	100.00
Blocks (original)	Harris	56	-	-	47	79.27
	SUSAN	61	-	-	46	74.21
	FAST	76	-	-	49	71.13
	CSS (He)	53	-	-	47	81.64
	JUDOCA	59	-	-	59	96.83
Blocks (noisy)	Harris	56	68	31.86	41	69.15
	SUSAN	61	130	0.14	43	69.37
	FAST	76	264	1.65E-06	3	4.35
	CSS (He)	53	56	75.13	43	74.69
	JUDOCA	59	58	90.91	58	95.18
Blocks (smoothed)	Harris	56	56	100.00	31	52.28
	SUSAN	61	37	10.15	32	51.63
	FAST	76	57	16.35	42	60.96
	CSS (He)	53	57	68.3	42	72.96
	JUDOCA	59	59	100.00	59	96.83
Lab (original)	Harris	173	-	-	74	34.84
	SUSAN	214	-	-	122	50.69
	FAST	706	-	-	183	46.23
	CSS (He)	271	-	-	211	77.29
	JUDOCA	278	-	-	263	95.12
Lab (noisy)	Harris	173	213	2.21	70	32.96
	SUSAN	214	283	0.006	111	46.12
	FAST	706	106	3.81E-35	23	5.81
	CSS (He)	271	267	68.3	208	76.19
	JUDOCA	278	277	86.96	224	81.02
Lab (smoothed)	Harris	173	164	28.43	74	34.84
	SUSAN	214	154	0.02	66	27.42
	FAST	706	1019	1.00E-17	193	48.76
	CSS (He)	271	265	56.45	212	77.66
	JUDOCA	278	278	100.00	246	88.97

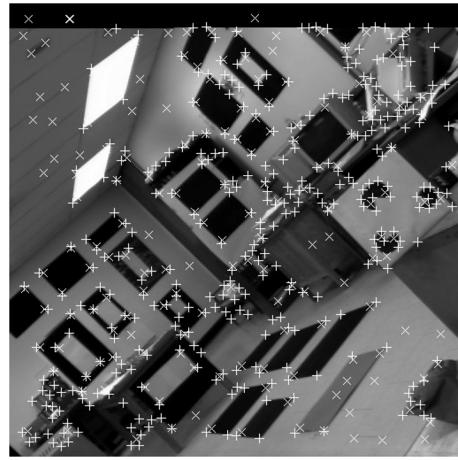


Fig. 22. JUDOCA locations are indicated by the + signs, whereas Harris locations are indicated by the x signs.

E. Timing Issues

Time is measured for different resolutions using different operators. Experiments are performed on a P4 processor running

TABLE II
M-ACU VALUES FOR THE “SQUARES” IMAGE

Operator	N_o	M-ACU
Harris	36	26.70
SUSAN	36	100.00
FAST	32	67.49
CSS (He)	40	92.31
JUDOCA	36	100.00

TABLE III
TIME IN SECONDS CONSUMED BY DIFFERENT OPERATORS

Resolution	Harris	SUSAN	FAST	CSS (He)	JUDOCA
512 × 512	0.13	0.0805	0.01	4.329	0.726
640 × 640	0.186	0.1255	0.02	6.075	1.1265
800 × 800	0.28	0.19	0.0305	7.853	1.757
1000 × 1000	0.466	0.29	0.04	13.417	2.694

at 1.8 GHz. The results listed in Table III show that our operator consumes more time than the Harris, SUSAN and FAST operators but less time than the modified CSS.

VIII. CONCLUSION

This paper have presented a technique to accurately detect junctions in the image. The proposed JUDOCA algorithm uses Gaussian derivative filters. Two binary images have been obtained, i.e., \mathcal{B} by imposing a threshold on the gradient magnitude image and \mathcal{B}^+ by applying nonmaximal suppression. At each point in \mathcal{B} , a circle has been positioned. The so-called CA points have been determined as the intersection between the circle and \mathcal{B}^+ . The radial lines connecting the center and the CA points have been scanned. This scanning process determines whether the radial line is a branch of a putative junction. A junction is detected if at least two non-collinear branches are detected. Experimental results have been performed showing the accuracy of the proposed solution. This paper has proposed a new formula for measuring the location accuracy. Comparisons with other well-known operators show the superiority of the JUDOCA operator in many aspects.

REFERENCES

- [1] W. Barrett and K. Petersen, “Houghing the hough: Peak collection for detection of corners, junctions and line intersections,” in *Proc. IEEE Comput. Vis. Pattern Recog.*, 2001, vol. II, pp. II-302–II-309.
- [2] D. Beymer, “Finding junctions using the image gradient,” in *Proc. IEEE Comput. Vis. Pattern Recog.*, 1991, pp. 720–721.
- [3] J. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.
- [4] F. Deschenes and D. Ziou, “Detection of line junctions in gray-level images,” in *Proc. ICPR*, 2000, vol. 3, pp. 754–757.
- [5] R. Elias, “Wide baseline matching through homographic transformation,” in *Proc. 17th ICPR*, 2004, vol. 4, pp. 130–133.
- [6] R. Elias, “Enhancing accuracy of camera rotation angles detected by inaccurate sensors and expressing them in different systems for wide baseline stereo,” in *Proc. QCAV*, 2007, vol. 6356, pp. 635 617–1–635 617–8.
- [7] R. Elias, “Sparse view stereo matching,” *Pattern Recog. Lett.*, vol. 28, no. 13, pp. 1667–1678, Oct. 2007.
- [8] R. Elias, “Enhancing sensor measurements through wide baseline stereo,” *ELCVIA*, vol. 7, no. 3, pp. 36–53, 2008.
- [9] R. Elias, *Modeling of Environments: From Sparse Views to Obstacle Reconstruction*. Saarbrücken, Germany: LAP Lambert Acad. Publ., 2009.
- [10] R. Elias and A. Elnahas, “An accurate indoor localization technique using image matching,” in *Proc. 3rd IET IE*, 2007, pp. 376–382.
- [11] R. Elias and A. Elnahas, “Fast localization in indoor environments,” in *Proc. IEEE CISDA*, 2009, pp. 1–6.
- [12] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*. Reading, MA: Addison-Wesley, 1996.
- [13] H. Hajjdiab, R. Elias, and R. Laganière, “Wide baseline obstacle detection and localization,” in *Proc. ISSPA*, 2003, vol. 1, pp. 21–24.
- [14] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proc. Alvey Vision Conf.*, 1988, pp. 147–151.
- [15] X. He and N. Yung, “Corner detector based on global and local curvature properties,” *Opt. Eng.*, vol. 47, no. 5, pp. 057 008-1–057 008-13, May 2008.
- [16] S. Kamel, M. Moharram, and R. Elias, “Wide baseline stereo matching through junction parametric and polar warping,” in *Proc. CGIM*, 2010, pp. 59–64.
- [17] F. Mohanna and F. Mokhtarian, “Performance evaluation of corner detection algorithms under similarity and affine transforms,” in *Proc. BMVC*, 2001, pp. 353–362.
- [18] F. Mokhtarian and F. Mohanna, “Performance evaluation of corner detectors using consistency and accuracy measures,” *Comput. Vis. Image Understand.*, vol. 102, no. 1, pp. 81–94, Apr. 2006.
- [19] F. Mokhtarian and R. Suomela, “Robut image corner detection through curvature scale space,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1376–1381, Dec. 1998.
- [20] H. Moravec, “Visual mapping by a robot rover,” in *Proc. IJCAI*, 1979, pp. 598–600.
- [21] L. Parida, D. Geiger, and R. Hummel, “Junctions: Detection, classification, and reconstruction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 7, pp. 687–698, Jul. 1998.
- [22] E. Rosten and T. Drummond, “Fusing points and lines for high performance tracking,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1508–1515.
- [23] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *Proc. ECCV*, 2006, pp. 430–443.
- [24] M. Ruzon and C. Tomasi, “Edge, junction and corner detection using color distributions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1281–1295, Nov. 2001.
- [25] S. Smith and J. Brady, “SUSAN—A new approach to low level image processing,” Oxford Univ., London, U.K., Tech. Rep. TR95SMS1c, 1995.
- [26] S. Smith and J. Brady, “SUSAN—A new approach to low level image processing,” *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, May 1997.
- [27] W. Yu, K. Daniilidis, and G. Sommer, “Rotated wedge averaging method for junction characterization,” in *Proc. IEEE Comput. Vis. Pattern Recog.*, 1998, pp. 390–395.
- [28] X. Zhang, H. Wang, M. Hong, L. Xu, D. Yang, and B. Lovell, “Robust image corner detection based on scale evolution difference of planar curves,” *Pattern Recog. Lett.*, vol. 30, no. 4, pp. 449–455, Mar. 2009.
- [29] X. Zhang, H. Wang, A. Smith, X. Ling, B. Lovell, and D. Yang, “Corner detection based on gradient correlation matrices of planar curve,” *Pattern Recog.*, vol. 43, no. 4, pp. 1207–1223, Apr. 2010.



Rimon Elias (M’98–SM’09) received the M.C.S. and Ph.D. degrees in computer science from the University of Ottawa, Ottawa, ON, Canada, in 1999 and 2004, respectively.

He is the author of “Modeling of Environments” published by Lambert Academic Publishing, and of several book chapters, encyclopedia, journal, and conference papers. His interests embrace different image-related fields including computer vision, image processing, computer graphics, and visualization.

Dr. Elias is listed in *Who’s Who in the World* and *Who’s Who in Science and Engineering*.



Robert Laganière (M’97) received the Ph.D. degree from INRS-Telecommunications, Montreal, QC, Canada, in 1996.

He is a Full Professor and a Faculty Member with the VIVA Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. He is the author of “OpenCV Computer Vision Application Programming” published by Packt Publishing, and the coauthor of “Object-Oriented Software Engineering” published by McGraw Hill. His research interests are in computer vision and image processing with applications to visual surveillance, driver assistance, image-based modeling, and content-based video interpretation.

are in computer vision and image processing with applications to visual surveillance, driver assistance, image-based modeling, and content-based video interpretation.