Extracting Line Junctions from Curvilinear Structures

François Deschênes and Djemel Ziou
Département de mathématiques et d'informatique
Université de Sherbrooke, Québec, Canada, J1K 2R1
(1)(819)821-8000 / (1)(819)821-8200 / {deschene, ziou}@dmi.usherb.ca

ABSTRACT

This paper describes an efficient approach for the detection of line junctions in gray-level images. The algorithm is divided into two steps. First, given the lines extracted from the original image, local line curvature is estimated. For this purpose, two different measures of curvature are proposed: The rate of change of direction of the orientation vector along the line and the mean of the dot products of orientation vectors within a given neighborhood. The second step involves the localization of junctions. Examples are provided based on experiments with remotely sensed images containing road intersections.

1. INTRODUCTION

The intersection of several edges (e.g. steps, lines) constitutes a junction or a corner. Junctions prove to be useful in many computer vision and image-understanding tasks such as optical flow estimation, stereo matching and detection of road intersections in remotely sensed images. The problem of detecting gray-level line junctions has not been tackled thoroughly in previous work. Traditional junction detectors are devoted to the detection of step-edge corners. Such corners are usually located using a Laplacian operator (zero-crossings or extrema) or a curvature measure based on the gradient direction. However, the gradient direction of a line does not exist, and a line junction does not correspond to a zero-crossing of the Laplacian. Consequently, step-edge corner detectors are not suitable for line junctions. Their response to a single line junction is not unique [2]. As far as we know, no corner detectors have been proposed yet for this type of junctions.

This paper proposes a detection method for gray-level line junctions which is suitable for L, X, Y and T junctions. The method described here is divided into two stages. First, given the lines extracted from the original image, the local curvature is computed. Two different measures of curvature have been developed and tested: The projection of the change of direction of the orientation vector along the line, and the mean of the dot products of orientation vectors within a local neighborhood. The second stage involves the localization of line junctions. Detailed descriptions of both stages are presented in Sections 2 and 3, respectively. The resulting algorithm is summarized in Section 4. Section 5 is devoted to experimental results. The conclusion follows in Section 6.

2. ESTIMATION OF LOCAL LINE CURVATURE

The rate of change of line direction is commonly called line curvature. Based on this definition, an ideal line junction corresponds to local maximum of line curvature. Assuming that lines (plausibility, orientation and location) have been extracted from the original image using a line detector, a local line curvature can be estimated. As mentioned before, two curvature measures were developed and tested. The first one is derived from a measure proposed by Kitchen and Rosenfeld [3] for step-edge corners, which is defined as the rate of change of gradient direction along the edge, multiplied by the gradient magnitude. Based on this work, a curvature measure for lines can be defined. Let $\vec{d}(x,y) = (u,v)$ be the non-normalized orientation vector perpendicular to the line at any point (x,y). The direction θ of $\vec{d}$ is given by:

$$\theta(x,y) = \begin{cases} \arctan\left(\frac{v}{u}\right) & \text{if } u \neq 0 \\ \frac{\pi}{2} & \text{otherwise.} \end{cases} \quad (1)$$

The partial derivatives of θ are:

$$\theta_x = \frac{v_x u - u_x v}{u^2+v^2} \quad \text{and} \quad \theta_y = \frac{v_y u - u_y v}{u^2+v^2}, \quad (2)$$

where u_x and v_x are the partial derivatives of u and v. Projecting the vector (θ_x, θ_y) along the line and multiplying the result by the magnitude of $\vec{d}$ provides a measure of curvature given by:

$$C(x,y) = \frac{u_x v^2 + v_y u^2 - uv(u_y + v_x)}{u^2+v^2}. \quad (3)$$

This quantity is large for points belonging to the area surrounding a junction. This is due to the significant change of direction of θ within such an area.

The second measure is based on the dissimilarity between the orientation vectors of the line pixels within a given neighborhood. Lacroix and Acheroy [4] proposed a measure of dissimilarity for step-edge corners based on the cross product of gradient vectors. By replacing cross product with dot product and adapting it to lines, the measure of dissimilarity between any pair of orientation vectors $(\vec{d}(x_1,y_1), \vec{d}(x_2,y_2))$ is given by:

$$s(\vec{d}(x_1,y_1), \vec{d}(x_2,y_2)) = 1 - \cos(\vec{d}(x_1,y_1), \vec{d}(x_2,y_2)). \quad (4)$$

$s(\vec{d}(x_1,y_1), \vec{d}(x_2,y_2))$ equals 1 for perpendicular vectors and is null for parallel vectors. The second measure of curvature may be estimated by computing the mean of $s(\vec{d}(x,y), \vec{d}(x+\Delta x, y+\Delta y))$ over a local $(2M+1)\times(2M+1)$ neighborhood. This measure is given by:

$$C(x,y) = \frac{1}{N}\sum_{\Delta x=-M}^{M}\sum_{\Delta y=-M}^{M}\beta(\Delta x,\Delta y)\times s(\vec{d}(x,y),\vec{d}(x+\Delta x,y+\Delta y)), \quad (5)$$

where

$$\beta(\Delta x,\Delta y) = \begin{cases} e^{-(\Delta x^2+\Delta y^2)} & \text{if } (x+\Delta x, y+\Delta y) \text{ and } (x,y) \text{ are linked by a line} \\ 0 & \text{otherwise} \end{cases}$$

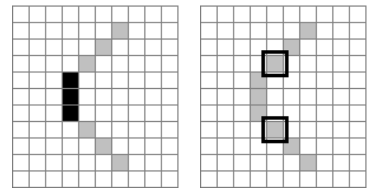
and N represents the number of points $(x+\Delta x, y+\Delta y)$ such that $\beta \neq 0$. β ensures that the influence of faraway neighbors of (x,y) on $C(x,y)$ is weaker than the influence of closer neighbors. This function also ensures that only line points are taken into account, which means that line strength does not need to be considered explicitly in 5. Thus $C(x,y)$ is large for line points lying in the neighborhood of a junction.

3. LOCALIZATION OF JUNCTIONS

We mentioned before that local maxima of line curvature correspond to line junctions. However, in practice this property is not always valid. First, smoothing and numerical differentiation affect the estimation of curvature in the vicinity of a junction. They also influence line location and orientation. Secondly, a line junction is formed by the intersection of several lines. Hence, the orientation of a junction point is not unique. However, the line detector provides only one orientation vector per pixel. Therefore, there is no guarantee that local maxima of curvature coincide with junction locations [2]. In order to circumvent these problems, orientation vectors of junctions are extrapolated from those of surrounding line points. Since smoothing modifies orientation vectors, especially in high curvature regions such as the vicinity of a junction, the orientation vectors of the immediate neighbors of a junction must not be taken into account. For this purpose, two classes of line points are created: Line points having low curvature value and those having high curvature value. Let low curvature endpoints (LCE) be defined as line points having low curvature value and belonging to the neighborhood of a high curvature point, as shown in Fig. 1. Based on this definition, the junction detector extracts low curvature endpoints in order to predict junction locations. Since lines are assumed to be one pixel wide, we consider only one LCE per line within a local neighborhood. For this purpose, all non-maximum LCE within a 3×3 neighborhood are removed using a measure of continuity $p(x,y)$ given by:

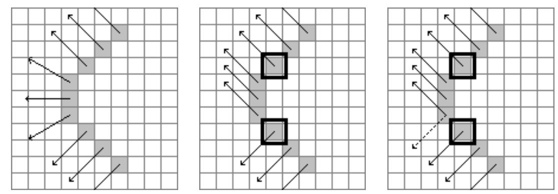
$$p(x,y) = [1 - C(x,y)] \times L(x,y), \quad (6)$$

where $C(x,y)$ represents the estimated curvature and $L(x,y)$ corresponds to the line strength. $p(x,y)$ is maximum for LCE having the lowest curvature value and the highest line strength.



a) b)

Figure 1: Example of low curvature endpoints. a) High curvature line pixels (black pixels). b) Low curvature endpoints (black squares).



a) b) c)

Figure 2: Example of propagated orientation vectors and curvature update. a) Original vectors. b) Propagated vectors from the first endpoint (upper black square). c) Given the propagated vectors, the curvature estimate is updated using the orientation vector of the second endpoint (lower black square).

Notice that the line strength has already been considered explicitly in (3), but in this case considering it again does not influence the final result.

Using low curvature endpoints (LCE), the orientation vectors of every junctions are recovered in order to update the curvature estimate. Starting from every LCE, high curvature neighbors in the line direction are visited. Each time the pixel (x,y) is visited, the orientation vector of the current LCE ($\vec{d}_i(x,y)$) is assigned to it, as shown in Fig. 2b and 2c. Then, the curvature $C(x,y)$ of every high curvature pixels is updated according to the following rule:

$$C(x,y) = C_o(x,y) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} s(\vec{d}_i(x,y), \vec{d}_j(x,y)) + k, \quad (7)$$

where n is the number of orientation vectors $\vec{d}_i(x,y)$ assigned to the pixel (x,y), $C_o(x,y)$ corresponds to the original curvature estimation, $s(\vec{d}_i(x,y), \vec{d}_j(x,y))$ is defined in (4) and $k \geq 1$ is a constant that corresponds to the minimum curvature increase. k is used to distinguish line points that have been visited at least once from those that have not been. Such a rule ensures that maximum values of line curvature correspond to line junctions [2]. Consequently, junctions can be located by extracting line points corresponding to maximum curvature value within a local neighborhood.

4. SUMMARY OF THE ALGORITHM

Let us assume that the orientation, plausibility and location of lines have been extracted from the original image. The implemented algorithm is straightforward and is divided into two main steps:

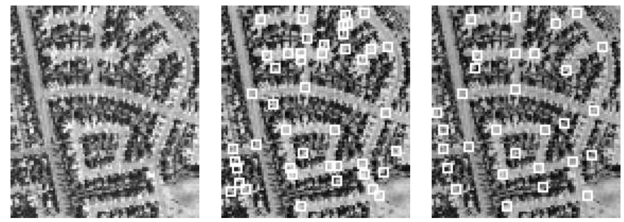
1. Estimation of local line curvature using either (3) or (5).
2. Localization of line junctions (Section 3): a) Extraction of low curvature endpoints (LCE); b) Updating of curvature estimate using (7); c) Extraction of local maxima of curvature.

5. EXPERIMENTAL RESULTS

The detection method for line junctions was tested using several synthetic and real images containing L, T, Y, and X line junctions. Lines were extracted from the original images using an algorithm presented by Steger [5]. Notice that this line detector is not the central part of the technique and it may be replaced by another detector that provides the location, plausibility and orientation of the lines [6]. The following results were obtained with two different remotely sensed images containing road intersections. First, Fig. 3 shows the results of the algorithm for an urban road network. The line detector parameters are the scale $\sigma = 2.2$ and the plausibility threshold $t = 30.0$. The first curvature estimate (given by (3)) was thresholded using $t_c = 0.2$, while the second estimate (given by (5)) was computed within a 5×5 neighborhood ($M = 2$) and thresholded using $t_c = 0.8$. Additional information about the influence of the parameters (σ, t, t_c and M) can be found in [2]. The results obtained from a remotely sensed image containing rural road intersections are shown in Fig. 4 (for line detector: $\sigma = 2.5, t = 55.0$; for (3): $t_c = 0.2$ and for (5): $M = 2, t_c = 0.4$). As shown in these figures, all evident road intersections have been localized using a curvature estimate from either (3) or (5). However, (5) seems to provide more accurate results than (3). The response to a given junction is unique and localization is more accurate. This difference is mainly due to the imprecision of the numerical differentiation used in (3), which increases the localization error. In (5), there is no numerical differentiation. Furthermore, $\beta(\Delta x, \Delta y)$ ensures that only contiguous line points are taken into account. Consequently, (5) is superior to (3).

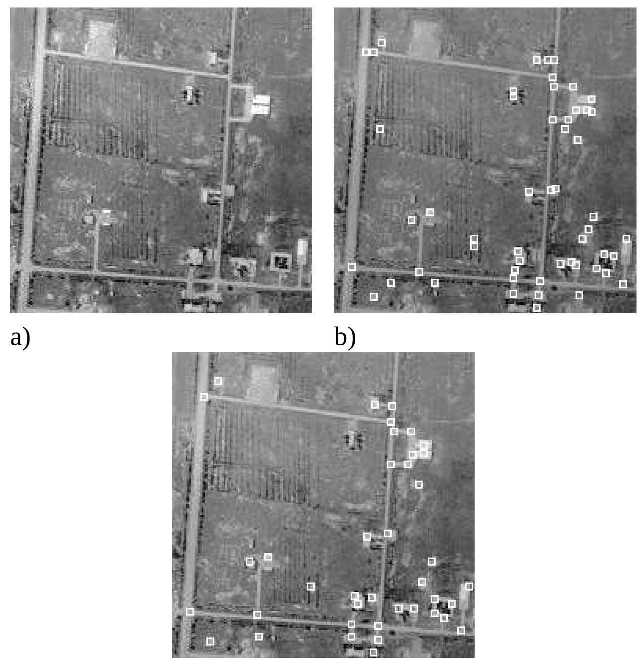
6. CONCLUSION

A detection method for gray-level line junctions was described in this paper. This method is divided into two steps. First, given the lines extracted from the original image, a local measure of curvature is computed. For this purpose, two different measures were tested: The rate of change of direction of the orientation vector along the line (given by (3)) and the mean of the dot products of orientation vectors within a given neighborhood (given by (5)). Then, additional processing based on local curvature and low curvature endpoints is done in order to localize junctions. Experimental results show that the algorithm localizes line junctions accurately. Those also show that (5), the second measure of curvature, provides better localization accuracy. Up to now, this detector has been integrated into systems that deal with remotely sensed road networks [1].



a) b) c)

Figure 3: Real image containing road intersections. Courtesy of CTI, Geomatics Canada. a) Original image. b) Extracted junctions using (3). c) Extracted junctions using (5).



c)

Figure 4: Real image containing rural roads. a) Original image. b) Extracted junctions using (3). c) Extracted junctions using (5).

References

[1] M. F. Auclair Fortier, D. Ziou, C. Armenakis, and S. Wang. Automated Correction and Updating of Roads from Aerial Ortho-Images. In *Proceedings of the International Society Photogrammetry and Remote Sensing Conference*, Amsterdam, Holland, July 16-23 2000.

[2] F. Deschênes and D. Ziou. Detection of Line Junctions and Line Terminations Using Curvilinear Features. Technical Report 235, DMI, Université de Sherbrooke, Canada, J1K 2R1, 1999.

[3] L. Kitchen and A. Rosenfeld. Gray-level Corner Detection. *Pattern Recognition Letters I*, pages 95–102, 1982.

[4] V. Lacroix and M. Acheroy. Feature Extraction Using the Constrained Gradient. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53:85–94, 1998.

[5] C. Steger. Extracting Curvilinear Structures: A Differential Geometric Approach. In *Proceedings of the Fourth European Conference on Computer Vision*, volume 1, pages 630–641, 1996.

[6] D. Ziou. Finding Lines in Grey-Level Images. In *Proceedings of the International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8 2000.