TR－H－036　　　0095

Junctions Analysis on the Projected Image of 3D Objects ~Detecting Y-and arrow-Junctions from Local Image Derivatives~

Toshiki ISO　Masahiko SHIZAWA

1993. 10. 21

ATR人間情報通信研究所

〒619-02 京都府相楽郡精華町光台2-2　☎07749-5-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
Telephone: +81-7749-5-1011
Facsimile: +81-7749-5-1008

Junctions Analysis on the Projected Image of 3D Objects

∼ Detecting Y- and arrow-Junctions from Local Image Derivatives ∼

Member Toshiki Iso
Member Masahiko Shizawa

ATR Human Information Processing Laboratories

Hikaridai 2-2, Seika-cho Soraku-gun Kyoto 619-02 Japan
Tel: 07749-5-1065 Fax: 07749-5-1008
E-mail: iso@hip.atr.co.jp

Editorial Index: Image Processing

Abstract

Analyzing projected images is important for interpreting 3D objects. In particular, it is necessary to analyze such junctions as the Y- and arrow-junctions which are composed of multiple edges. This paper proposes a one-shot algorithm for detecting the Y- and arrow-junctions of projected images. This algorithm includes a stage to detect orientations of triple edges and another stage to classify junctions by using information on local image derivatives. Experimental results obtained by using our algorithm show that Y- and arrow-junctions can now be detected easily.

1 INTRODUCTION

It is important to detect specified objects from among several objects for pattern recognition and image understanding[1]. To do this, it is necessary to analyze the shapes of junctions generated by overlapping objects and to know the relative positions of these objects. detection of multiple orientations at junctions has required the use of many filters tuned to different orientations, and the selection of appropriate orientations has been done by searching for significant energy responses[2]. Freeman and Adelson detected multiple orientations by using steerable filters which can be tuned to arbitrary orientations by continuously varying the weights of the linearly-combined basis filters[3]. This method, however, requires that the energy extrema be found in the energy distribution of the filter responses. Furthermore, fourth-order Gaussian differentiation filters are required to detect two orientations at junctions like a corner. Perona proposed steerable-scalable kernels for junction analysis[4], but these kernels have complicated shapes. On the contrary, we proposed a one-shot algorithm that can obtain double orientations without finding the energy extrema and without using complicated filters. The algorithm uses a parametrized filter based on the principle of superposition[10].

To detect the contours of objects, however, it is necessary not only to get the multiple orientations of junctions, but also to classify the junctions. This is because the junction types provide important clues as to the arrangement of objects. Freeman and Adelson detected junctions by using energy responses obtained from filtering with a quadrature pair filter, but their method did not include clear criteria for the classification of the junction shapes[5]. Perona also did not provide clear criteria for classifying junctions[4]. Harris proposed a corner detector by using first-order image derivatives, but it could only detect the L-junctions, and its performance was inadequate for higher-order structures[6]. Noble's method of considering the geometric properties of the image surface did not produce good results for real images[7]. Rosenthaler, Heitger et al. proposed a detection scheme for keypoints, based on an analysis of oriented energy channels using differential geometry, but their method could only localize keypoints (junction points); it could not classify junctions[8]. Matas and Kittler proposed an algorithm to obtain globally consistent junctions with probabilistic relaxation from lines detected by Hough transform[9]; the algorithm was complex. A final disadvantage: The above methods were not effective for detecting junctions like Y- or arrow-junctions.

With the above background in mind, we proposed a one-shot algorithm that requires only low-order image derivatives in local neighborhoods to detect double orientations and to classify L- ,T- , and X-junctions[12]. Moreover, we showed the results for artificial images and real images. In this paper, we describe an extended version of that algorithm; the new algorithm enables us to detect Y- and arrow-junctions. The algorithm has two stages: the first stage detects triple orientations by using local image derivatives[10]. The other classifies Y- and arrow-junctions by using two criteria that can be checked with first-order image derivatives in local neighborhoods. Finally, we show the results of experiments obtained by using this algorithm.

2 THEORY OF DETECTING TRIPLE ORIENTATIONS

In our method, first, it is necessary to detect triple orientations of edges that compose Y- or arrow-junctions. In this chapter, we explain the theory of detecting triple orientations[10],[12]. Any point (x, y) on the edge along $\vec{p_1}$ in the image $f_1(x, y)$ must satisfy the constraint:

$$\{(\vec{p_1} \cdot \nabla) g(x, y)\} * f_1(x, y) = 0 \tag{1}$$

$(\vec{p_1} \cdot \nabla)$ is the inner product of $\vec{p_1}$ and ∇. $\nabla = \frac{\partial}{\partial x}\vec{e_x} + \frac{\partial}{\partial y}\vec{e_y}$ ($\vec{e_x}$ and $\vec{e_y}$ are the unit vectors of x and y). $g(x, y)$ is a window function, and $\{(\vec{p_1} \cdot \nabla) g(x, y)\} * f_1(x, y)$ is the convolution of $(\vec{p_1} \cdot \nabla) g(x, y)$ and $f_1(x, y)$. Next, we assume that the image $f(x, y)$ containing orientations of the edge along $\vec{p_1}$, $\vec{p_2}$ and $\vec{p_3}$ can be equal to the image generated by additive superposition of image $f_1(x, y)$ with the edge along $\vec{p_1}$, image $f_2(x, y)$ with the edge along $\vec{p_2}$,and image $f_3(x, y)$ with the edge along $\vec{p_3}$ (see Fig. 1)[11],

$$f(x, y) = f_1(x, y) + f_2(x, y) + f_3(x, y) \tag{2}$$

Then, the constraint of triple orientations can be written as follows:

$$\{(\vec{p_1} \cdot \nabla)(\vec{p_2} \cdot \nabla)(\vec{p_3} \cdot \nabla) g(x, y)\} * f(x, y) = 0 \tag{3}$$

This constraint can be proved as follows;

$$
\begin{aligned}
&(\vec{p_1}\cdot\nabla)(\vec{p_2}\cdot\nabla)(\vec{p_3}\cdot\nabla)g(x,y)*f(x,y) \\
&= (\vec{p_1}\cdot\nabla)(\vec{p_2}\cdot\nabla)(\vec{p_3}\cdot\nabla)g(x,y)*(f_1(x,y)+f_2(x,y)+f_3(x,y)) \\
&= (\vec{p_2}\cdot\nabla)(\vec{p_3}\cdot\nabla)[(\vec{p_1}\cdot\nabla)g(x,y)*f_1(x,y)] \\
&\qquad +(\vec{p_3}\cdot\nabla)(\vec{p_1}\cdot\nabla)[(\vec{p_2}\cdot\nabla)g(x,y)*f_2(x,y)] \\
&\qquad +(\vec{p_1}\cdot\nabla)(\vec{p_2}\cdot\nabla)[(\vec{p_3}\cdot\nabla)g(x,y)*f_3(x,y)] \\
&= 0 \qquad (4)
\end{aligned}
$$

To obtain $\vec{p_1}$, $\vec{p_2}$ and $\vec{p_3}$ so as to satisfy the constraint, the squared residual on the left hand side of equation (3) is optimized by using Lagrange's method of undetermined multipliers with the condition $|\vec{p_1}|^2|\vec{p_2}|^2|\vec{p_3}|^2 = 1$. ($|\vec{p_1}| = \sqrt{p_{1x}^2+p_{1y}^2}$, $|\vec{p_2}| = \sqrt{p_{2x}^2+p_{2y}^2}$, $|\vec{p_3}| = \sqrt{p_{3x}^2+p_{3y}^2}$)
We define

$$
\begin{aligned}
E(\vec{p_1},\vec{p_2},\vec{p_3},\lambda) &= \sum_{(x,y)\in\varepsilon} |\{(\vec{p_1}\cdot\nabla)(\vec{p_2}\cdot\nabla)(\vec{p_3}\cdot\nabla)g(x,y)\}*f(x,y)|^2 \\
&\qquad +\lambda(1-|\vec{p_1}|^2|\vec{p_2}|^2|\vec{p_3}|^2) \qquad (5)
\end{aligned}
$$

where ε is the area of energy integration. Then, the conditions for the optimal estimates of $\vec{p_1}$, $\vec{p_2}$ and $\vec{p_3}$ are

$$
\frac{\partial E}{\partial P_1}=0,\quad \frac{\partial E}{\partial P_2}=0,\quad \frac{\partial E}{\partial P_3}=0,\quad \frac{\partial E}{\partial P_4}=0,\quad \frac{\partial E}{\partial \lambda}=0 \qquad (6)
$$

where

$$
P_1 = p_{1x}p_{2x}p_{3x} \qquad (7)
$$

$$P_2 = p_{1y}p_{2y}p_{3y} \tag{8}$$

$$P_3 = p_{1x}p_{2x}p_{3y} + p_{1x}p_{2y}p_{3x} + p_{1y}p_{2x}p_{3x} \tag{9}$$

$$P_4 = p_{1x}p_{2y}p_{3y} + p_{1y}p_{2x}p_{3y} + p_{1y}p_{2y}p_{3x} \tag{10}$$

This yields an eigenvalue problem:

$$AX = \lambda BX \tag{11}$$

where

$$A = \begin{pmatrix} a_{30,30} & a_{30,03} & a_{30,21} & a_{30,12} \\ a_{30,03} & a_{03,03} & a_{03,21} & a_{03,12} \\ a_{30,21} & a_{03,21} & a_{21,21} & a_{21,12} \\ a_{30,12} & a_{03,12} & a_{21,12} & a_{12,12} \end{pmatrix}, B = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \frac{2}{3} \end{pmatrix}, X = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix}$$

$$a_{ij,kl} = \sum_{(x,y)\in\varepsilon} (W_{ij}W_{kl}^* + W_{kl}W_{ij}^*) \tag{12}$$

$$(i,j,k,l = 0,1,2,3; \quad i+j=3; \quad k+l=3)$$

$$W_{30} = \frac{\partial^3 g}{\partial x^3} * f, \; W_{03} = \frac{\partial^3 g}{\partial y^3} * f, \; W_{21} = \frac{\partial^3 g}{\partial x^2 y} * f, \; W_{12} = \frac{\partial^3 g}{\partial x y^2} * f$$

W_{ij}^* is a conjugate complex of function W_{ij}.

Therefore, by solving the eigenvalue problem (11) , we can obtain P_1, P_2, P_3, P_4.

Next, we will try to compute triple orientations from P_1, P_2, P_3, P_4. To do this, we solve simultaneous equations (7) $\sim$ (10) under the conditions $|p_1|^2 = 1$, $|p_2|^2 = 1$ and $|p_3|^2 = 1$. If the solutions are identical to each other in the following cases, we define them as independent solutions.

Case (I): $P_1 = p_{1x}p_{2x}p_{3x} \neq 0$

When both sides of equations (7) $\sim$ (10) are divided by P_1, we can obtain

$$\frac{P_2}{P_1} = k_1k_2k_3 \tag{13}$$

$$\frac{P_3}{P_1} = k_1 + k_2 + k_3 \tag{14}$$

$$\frac{P_4}{P_1} = k_1k_2 + k_2k_3 + k_3k_1 \tag{15}$$

where

$$k_1 = \frac{p_{1y}}{p_{1x}}, \quad k_2 = \frac{p_{2y}}{p_{2x}}, \quad k_3 = \frac{p_{3y}}{p_{3x}} \tag{16}$$

By using the relations between solutions and coefficients of algebraic equations, we can get the following third-degree algebraic equation with respect to t.

$$t^3 - \left(\frac{P_3}{P_1}\right)t^2 + \left(\frac{P_4}{P_1}\right)t - \left(\frac{P_2}{P_1}\right) = 0 \tag{17}$$

Now, k_1, k_2, and k_3 are solutions of this equation. Therefore, by solving the equation, we can obtain the following orientations of triple edges.

$$\theta_1 = \tan^{-1}(k_1), \quad \theta_2 = \tan^{-1}(k_2), \quad \theta_3 = \tan^{-1}(k_3) \tag{18}$$

Case (II): $P_1 = 0,\ P_3 \neq 0$

Our problem is symmetrical with respect to k_1, k_2, and k_3. Therefore, without loss of generality, we can assume

$$p_{1x} = 0. \tag{19}$$

Then, by using $|\vec{p_1}|^2 = 1$, we can assume

$$p_{1y} = 1. \tag{20}$$

By using the conditions in equations (19) and (20),

$$k_1 = \frac{1}{0} = \infty \quad (\theta_1 = \frac{\pi}{2}) \tag{21}$$

Then, by dividing both sides of equations (8) $\sim$ (10), we can get as follows;

$$\frac{P_2}{P_3} = k_2 k_3 \tag{22}$$

$$\frac{P_4}{P_3} = k_2 + k_3 \tag{23}$$

We can obtain k_2 and k_3 as real solutions of a second-degree algebraic equation.

$$t^2 - \left(\frac{P_4}{P_3}\right) t + \left(\frac{P_2}{P_3}\right) = 0 \tag{24}$$

Then, we can get the following triple orientations.

$$\theta_1 = \frac{\pi}{2}, \quad \theta_2 = \tan^{-1}(k_2), \quad \theta_3 = \tan^{-1}(k_3) \tag{25}$$

Case (III): $P_1 = 0,\ P_3 = 0,\ P_4 \neq 0$

Similarly, by using equations (7) $\sim$ (10) under the conditions $P_1 = 0$, $P_3 = 0$, $|\vec{p_1}|^2 = 1$ and $|\vec{p_2}|^2 = 1$, we can obtain as follows;

$$k_1 = \frac{1}{0} = \infty, \quad k_2 = \frac{1}{0} = \infty, \quad k_3 = \frac{P_2}{P_4} \tag{26}$$

Then, we can get the following triple orientations.

$$\theta_1 = \frac{\pi}{2}, \;\; \theta_2 = \frac{\pi}{2}, \;\; \theta_3 = \tan^{-1}(k_3) \tag{27}$$

In each of the above cases, by solving algebraic equations, we can obtain orientations of triple edges.

3 THEORY OF CLASSIFYING Y- AND ARROW-JUNCTIONS

Now, we have triple orientations $\vec{p_1}$, $\vec{p_2}$ and $\vec{p_3}$. However, $-\vec{p_1}$, $-\vec{p_2}$ and $-\vec{p_3}$ can also satisfy the constraint of triple orientations. Therefore, we can obtain eight candidates for Y- and arrow-junction shapes (see Fig. 2). In this chapter, we explain our theory of classifying Y- and arrow-junctions.

3.1 Two Criteria for Classifying Junctions

To detect Y- and arrow-junctions, we use two criteria for classifying junctions:

[Criterion 1] *Whether edges are real borders.*

The aim of this criterion is to choose real borders from candidate borders.

[Criterion 2] *Whether a cross-section of the intensity profile around the borders is concave/convex.*

The aim of this criterion is to check for contradiction in intensity values in the areas around the real borders.

3.2 Local Image Derivatives Information

We check the above criteria by using information on local first-order image derivatives (see Figures 3 and 4).

[*Sum of directional derivatives along border* $\vec{v}$]

$$D_{\vec{v}} = \sum_{(x,y)\in\delta(\vec{v})} [\{\vec{v}\cdot\nabla g(x,y)\} * f(x,y)] \qquad (28)$$

$$(\vec{v} = \vec{p_1}, \vec{p_2}, \vec{p_3}, -\vec{p_1}, -\vec{p_2}, -\vec{p_3})$$

[*Sum of directional derivatives along border* $\vec{v_\perp}$]

$$D_{\vec{v_\perp}} = \sum_{(x,y)\in\delta(\vec{v})} [\{\vec{v_\perp}\cdot\nabla g(x,y)\} * f(x,y)] \qquad (29)$$

$$(\vec{v_\perp} = \vec{p_{1\perp}}, \vec{p_{2\perp}}, \vec{p_{3\perp}}, -\vec{p_{1\perp}}, -\vec{p_{2\perp}}, -\vec{p_{3\perp}})$$

where $\delta(\vec{v})$ is the local area for evaluating borders. We define $\vec{v_\perp}$ to be the direction rotated $\frac{\pi}{2}$ counterclockwise from the direction of $\vec{v}$. The local area does not contain the neighborhoods of junctions because first-order image derivative information on the neighborhoods of junctions is not suitable for analyzing the shape of junctions.

3.3 Judgement of Two Criteria

Consequently, we check two criteria by using expressions (28) and (29).

[Judgement of Criterion 1]

For example, whether or not the direction $\vec{p_1}$ is a border depends on expressions (28) and (29). If $D_{\vec{v}}$ and $D_{\vec{v_\perp}}$ are nearly equal to zero, that is,

$$\begin{array}{lcl} |D_{\vec{p_1}}| & \simeq & 0 \\ |D_{\vec{p_{1\perp}}}| & \simeq & 0 \end{array} \tag{30}$$

then all directional first-order image derivatives in the local area along $\vec{p_1}$ are zero. Therefore, $\vec{p_1}$ cannot be a border. Otherwise, when

$$|D_{\vec{p_{1\perp}}}| > threshold \tag{31}$$

$\vec{p_1}$ can be a border.

[Judgement of Criterion 2]

Suppose $\vec{p_1}$ and $\vec{p_2}$ are borders. Under this condition, there are two areas divided by the two borders. To avoid contradiction in intensity values in the two areas, the cross-section of the intensity profile should be concave or convex (see Fig. 5). Therefore, it is necessary to check the signs of $D_{\vec{p_{1\perp}}}$ and $D_{\vec{p_{2\perp}}}$ since the signs can determine the intensity profile around the borders (see Fig. 5).

3.4 Classifying Y- and Arrow-junctions

Now, we can classify Y- and arrow-junctions as follows;

(i) By checking whether six candidate borders can be real or not, we determine three real borders.

(ii) By using (28) and (29) defined in Subsection **3.2**, we check the contradiction in intensity values in the areas around the real borders chosen above. Then, with three directions $\vec{p_1}$, $\vec{p_2}$,and $\vec{p_3}$, we can get six reasonable patterns of Y-junctions (see Fig. 6). False patterns satisfy the following conditions;

$$\begin{array}{rl} & (\ D_{\vec{p_1}\perp} > 0 \quad \wedge \quad D_{\vec{p_2}\perp} > 0 \quad \wedge \quad D_{\vec{p_3}\perp} > 0\ ) \\ \vee & (\ D_{\vec{p_1}\perp} < 0 \quad \wedge \quad D_{\vec{p_2}\perp} < 0 \quad \wedge \quad D_{\vec{p_3}\perp} < 0\ ) \end{array} \tag{32}$$

The same criterion can be used for classifying arrow-junctions.

4 EXPERIMENTS

We carried out computer simulations for our algorithm to detect junctions. Figure 7 shows the flow of our algorithm for the experiments. This flow has two steps: to detect multiple orientations, and to classify junctions.

4.1 Orientation Detection Stage

We detect orientations at each point (x, y) by using the following procedure.

[*Detecting multiple orientations*]

(i) We convolve original image $f(x, y)$ ($256[pixel] \times 256[pixel]$, 256 *grayscale—levels*) with the third-order, the second-order and the first-order image derivatives of window function $g(x, y)$.

Window function $g(x, y)$ is given as

$$g(x, y) = \frac{1}{2\pi\sigma^2} exp[-\frac{x^2 + y^2}{2\sigma^2}] \tag{33}$$

where the σ is scale parameter.

(ii) We solve each eigenvalue problem obtained from the optimization problem of the constraints for the triple (described in Section **2**), double and single orientations in ref.[12].

[Determining the number of orientations]
At each point, we determine the number of orientations, by using a criterion based on the ratio of minimum eigenvalue to second minimum eigenvalue.

4.2 Junction Classification Stage

We compute each sum of directional derivatives along $\vec{v} = \vec{p_1}$, $\vec{p_2}$, $\vec{p_3}$, $-\vec{p_1}$, $-\vec{p_2}$, $-\vec{p_3}$ and $\vec{v_\perp} = \vec{p_1}_\perp$, $\vec{p_2}_\perp$, $\vec{p_3}_\perp$,$-\vec{p_1}_\perp$, $-\vec{p_2}_\perp$, -$\vec{p_3}_\perp$ by using the first-order image derivatives obtained in Subsection **4.1** (i). Then, we classify junctions by using the criteria described in Subsection **3.1**.

4.3 Parameter setting for the experiments

In our experiments, we use the following parameter.
(1) Radius r_ε (see Fig. 3) of the area of energy integration ε is 30[*pixel*].
(2) Radius r_O (see Fig. 3) of the junction region is 2[*pixel*].
(3) The local area for evaluating borders $\delta(\vec{v})$ ($\vec{v} = \vec{p_1}, \vec{p_2}, \vec{p_3}, -\vec{p_1}, -\vec{p_2}, -\vec{p_3}$) contains a point (x, y) within 15[*pixel*] along $\vec{v}$ in r_O away from the junction point.
(4) The threshold value of inequality (31) for evaluating borders is 5.
(5) The scale parameter σ in equation (33) is 1[*pixel*].

4.4 Results of the Experiments

The results for artificial images are shown in Figure 8.

Only junctions are shown at every 32[*pixel*] in Figure 8(a) and at every 16[*pixel*] in Figures 8(b) and (c). In Figures 8(a)~(c), dots represent points having a single orientation or no orientation, and void areas represent points that cannot be classified as junctions.

5 CONCLUSION

We have proposed a one-shot algorithm that has a stage to detect triple orientations with the constraint of triple edges and another stage to classify Y- or arrow-junctions by using information on local image derivatives. Furthermore, it has been demonstrated that our method can be applied to artificial images. From results of experiments with our algorithm, criteria previously used for detecting L-, T- and X-junctions, were found to be equally effective in the detection of Y- and arrow-junctions. Our method has the following advantages:

- it is possible to detect Y- and arrow-junctions by using two criteria for classifying junctions.
- we can check the two criteria by using only first-order image derivatives.

Therefore, by using the criteria, we may soon be able to detect more complicated junctions like K- or ψ-junctions.

In the future, to make our algorithm robust, we will investigate how to determine the optimal size of the area of energy integration or local area for evaluating borders. Since real images have independent intensity variations in local areas, it is necessary to achieve adaptive processing in independent local areas. Moreover, we will examine how to localize junctions more accurately. Then, we will apply the proposed method to contour detection and the extraction of specified objects from among many objects.

ACKNOWLEDGMENT

The authors would like to thank Dr. Y. Tohkura, President of ATR Human Information Processing Laboratories, Dr. S. Akamatsu, and their colleagues in ATR Human Information Processing Laboratories.

References

[1] D.G. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, 1985.

[2] S.W. Zucker, "Early Orientation Selection: Tangent Fields and the Dimensionality of Their Support", *Comput. Vision, Graphics, and Image Process.*, Vol. 32, pp. 74–103, 1985.

[3] W.T. Freeman and E.H. Adelson, "The Design and Use of Steerable Filters", *IEEE Trans.on Pattern Anal. and Mach Intell.*, Vol. PAMI-13, No. 9, pp. 891–906, 1991.

[4] P. Perona, "Steerable-Scalable Kernels for Edge Detection and Junction Analysis", *Image and Vision Computing*, Vol. 10, No. 10, pp. 663–672, 1992.

[5] W.T. Freeman, "Steerable Filters and Local Analysis of Image Structure", *Ph.D. Thesis, M.I.T. Media Lab.*, 1992.

[6] C.G. Harris, "Determination of ego-motion from matched points", *Proc. Alvey Vision Conf. Cambridge, UK*, pp. 189–192, 1987.

[7] J.A. Noble, "Finding corners", *Image and Vision Computing*, Vol. 6, No. 2, pp. 121–128, 1988.

[8] L. Rosenthaler, F. Heitger, O.Kubler and R. v. d. Heydt, "Detection of General Edges and Keypoints", *In Computer Vision - ECCV'92,* Springer-Verlag, pp. 78–86, 1992.

[9] J. Matas and J. Kittler, "Junction detection using probabilistic relaxation", *Image and Vision Computing,* Vol. 11, No.4, pp. 197–202, 1993.

[10] M. Shizawa and T. Iso, "Direct Representation and Detection of Multi-Scale, Multi-Orientation Fields Using Local Differentiation Filters", *Proc. IEEE CVPR'93,* pp. 508-514, 1993.

[11] M. Shizawa and K. Mase, "Principle of Superposition: A Common Computational Framework for Analysis of Multiple Motion", *Proc. IEEE Workshop on Visual Motion,* pp. 164-172, 1991.

[12] T. Iso and M. Shizawa, "Junction Analysis based on Local Image Derivatives $\sim$ Detecting L-, T-, X-Junctions from Low-order Image Derivatives $\sim$ ", *IEICE Transactions on Information and Systems (in Japanese)* (to appear).

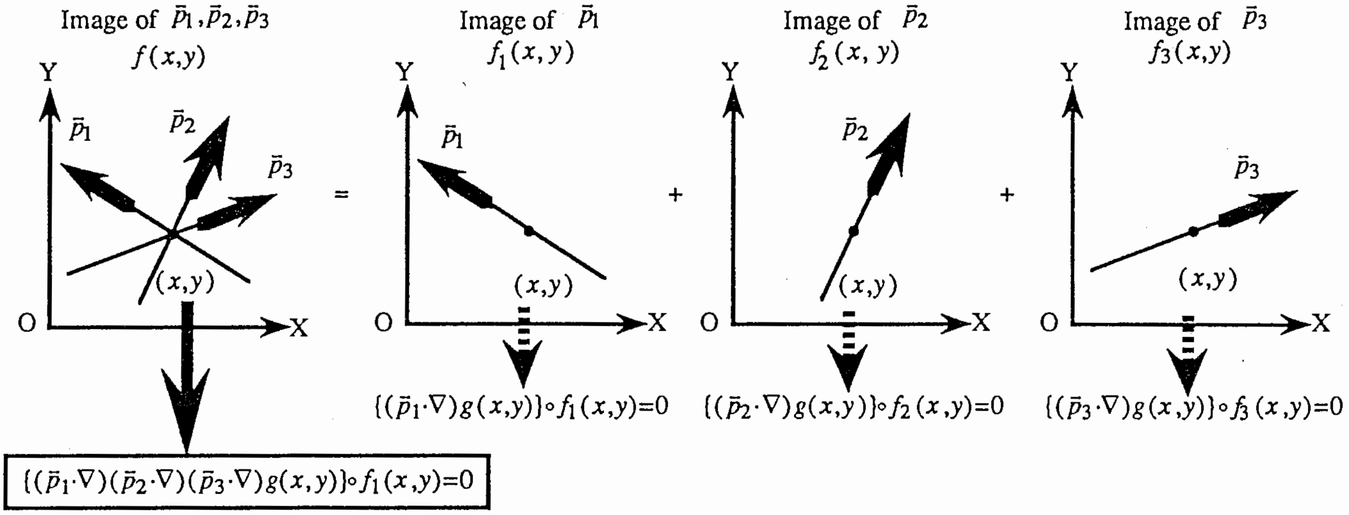


Image of $\bar{p}_1, \bar{p}_2, \bar{p}_3$
$f(x,y)$
Image of $\bar{p}_1$
$f_1(x,y)$
Image of $\bar{p}_2$
$f_2(x,y)$
Image of $\bar{p}_3$
$f_3(x,y)$
Y
X
O
$\bar{p}_1$
$\bar{p}_2$
$\bar{p}_3$
(x,y)
=
+
+
$\{(\bar{p}_1 \cdot \nabla) g(x,y)\} \circ f_1(x,y)=0$
$\{(\bar{p}_2 \cdot \nabla) g(x,y)\} \circ f_2(x,y)=0$
$\{(\bar{p}_3 \cdot \nabla) g(x,y)\} \circ f_3(x,y)=0$
$\{(\bar{p}_1 \cdot \nabla)(\bar{p}_2 \cdot \nabla)(\bar{p}_3 \cdot \nabla) g(x,y)\} \circ f_1(x,y)=0$

Fig 1 : Constraint of triple edges based on the Principle of Superposition

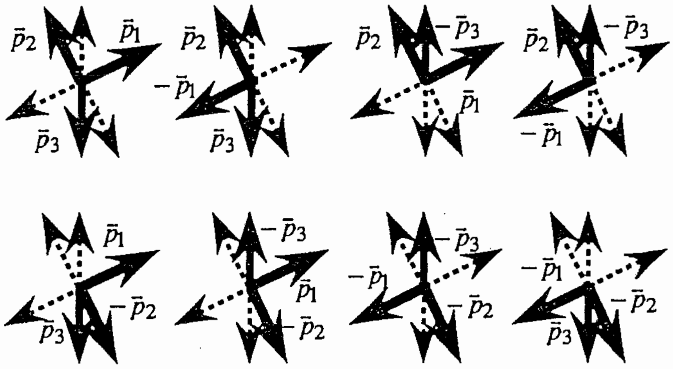
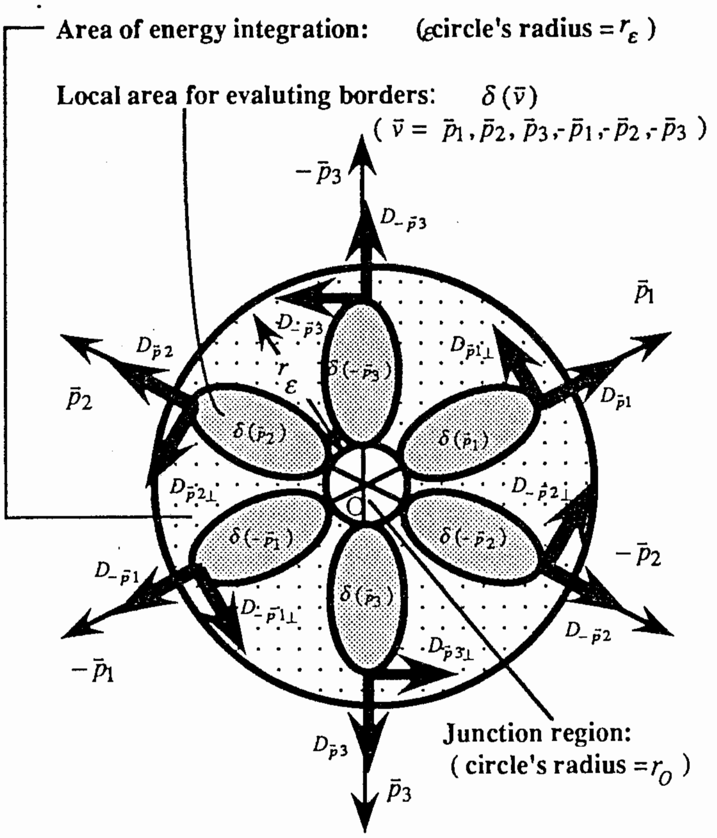


Fig 2 : Types of Y- and arrow -junctions



Area of energy integration: (εcircle's radius = r_ε)

Local area for evaluting borders: $\delta(\bar{v})$ ($\bar{v} = \bar{p}_1, \bar{p}_2, \bar{p}_3, -\bar{p}_1, -\bar{p}_2, -\bar{p}_3$)

Junction region: (circle's radius = r_O)

Fig 3 : Local derivative information for determining the junction

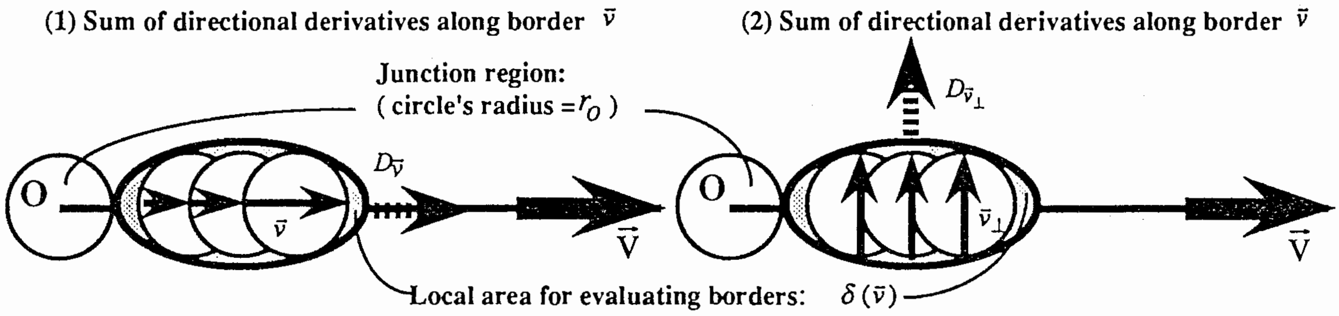
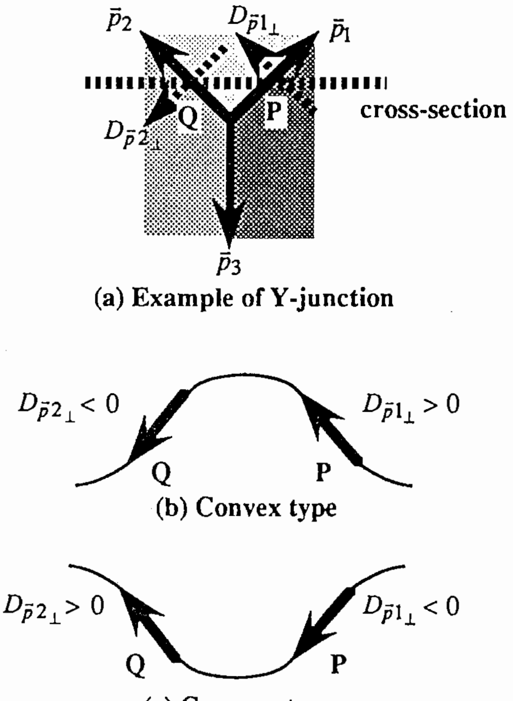

(1) Sum of directional derivatives along border $\vec{v}$
(2) Sum of directional derivatives along border $\vec{v}$
Junction region:
(circle's radius $=r_O$)
O
$D_{\vec{v}}$
$\vec{v}$
$\vec{V}$
O
$D_{\vec{v}_\perp}$
$\vec{v}_\perp$
$\vec{V}$
Local area for evaluating borders: $\delta(\vec{v})$

Fig 4 : Derivative information in the local area for evaluating borders

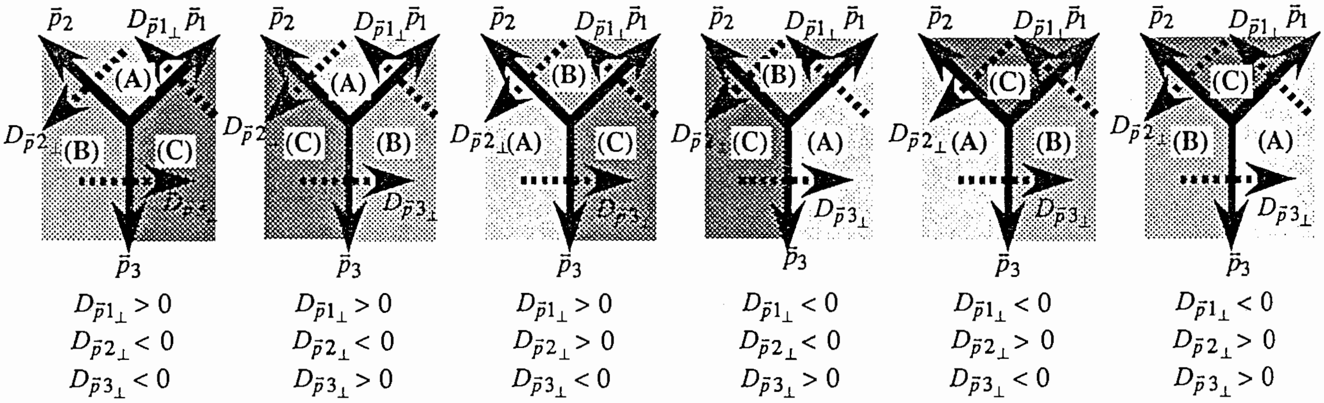


(a) Example of Y-junction

(b) Convex type

(c) Concave type

Fig 5 : A cross-section of an intensity profile in an area between border edges


$\bar{p}2$ $D_{\bar{p}1_\perp}$ $\bar{p}1$ (A) $D_{\bar{p}2_\perp}$ (B) (C) $D_{\bar{p}3_\perp}$ $\bar{p}3$
$D_{\bar{p}1_\perp} > 0$ $D_{\bar{p}2_\perp} < 0$ $D_{\bar{p}3_\perp} < 0$
$\bar{p}2$ $D_{\bar{p}1_\perp}$ $\bar{p}1$ (A) $D_{\bar{p}2_\perp}$ (C) (B) $D_{\bar{p}3_\perp}$ $\bar{p}3$
$D_{\bar{p}1_\perp} > 0$ $D_{\bar{p}2_\perp} < 0$ $D_{\bar{p}3_\perp} > 0$
$\bar{p}2$ $D_{\bar{p}1_\perp}$ $\bar{p}1$ (B) $D_{\bar{p}2_\perp}$ (A) (C) $D_{\bar{p}3_\perp}$ $\bar{p}3$
$D_{\bar{p}1_\perp} > 0$ $D_{\bar{p}2_\perp} > 0$ $D_{\bar{p}3_\perp} < 0$
$\bar{p}2$ $D_{\bar{p}1_\perp}$ $\bar{p}1$ (B) $D_{\bar{p}2_\perp}$ (C) (A) $D_{\bar{p}3_\perp}$ $\bar{p}3$
$D_{\bar{p}1_\perp} < 0$ $D_{\bar{p}2_\perp} < 0$ $D_{\bar{p}3_\perp} > 0$
$\bar{p}2$ $D_{\bar{p}1_\perp}$ $\bar{p}1$ (C) $D_{\bar{p}2_\perp}$ (A) (B) $D_{\bar{p}3_\perp}$ $\bar{p}3$
$D_{\bar{p}1_\perp} < 0$ $D_{\bar{p}2_\perp} > 0$ $D_{\bar{p}3_\perp} < 0$
$\bar{p}2$ $D_{\bar{p}1_\perp}$ $\bar{p}1$ (C) $D_{\bar{p}2_\perp}$ (B) (A) $D_{\bar{p}3_\perp}$ $\bar{p}3$
$D_{\bar{p}1_\perp} < 0$ $D_{\bar{p}2_\perp} > 0$ $D_{\bar{p}3_\perp} > 0$

Intensity in area (A) > Intensity in area (B) > Intensity in area (C)

Fig 6 : Shapes of intensity at neighborhoods of Y-junctions

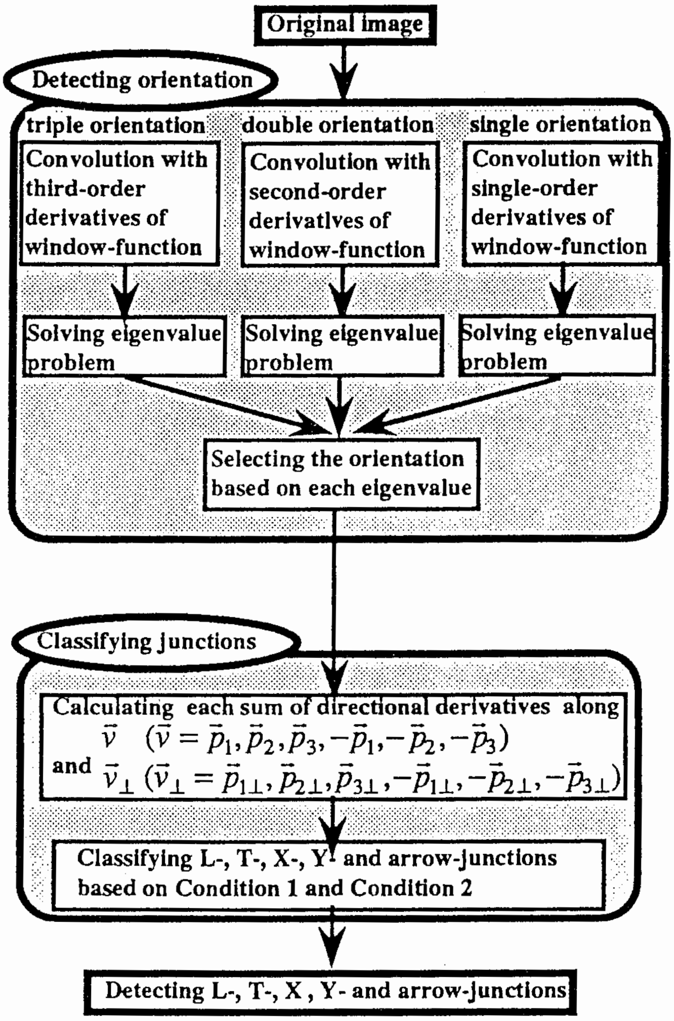
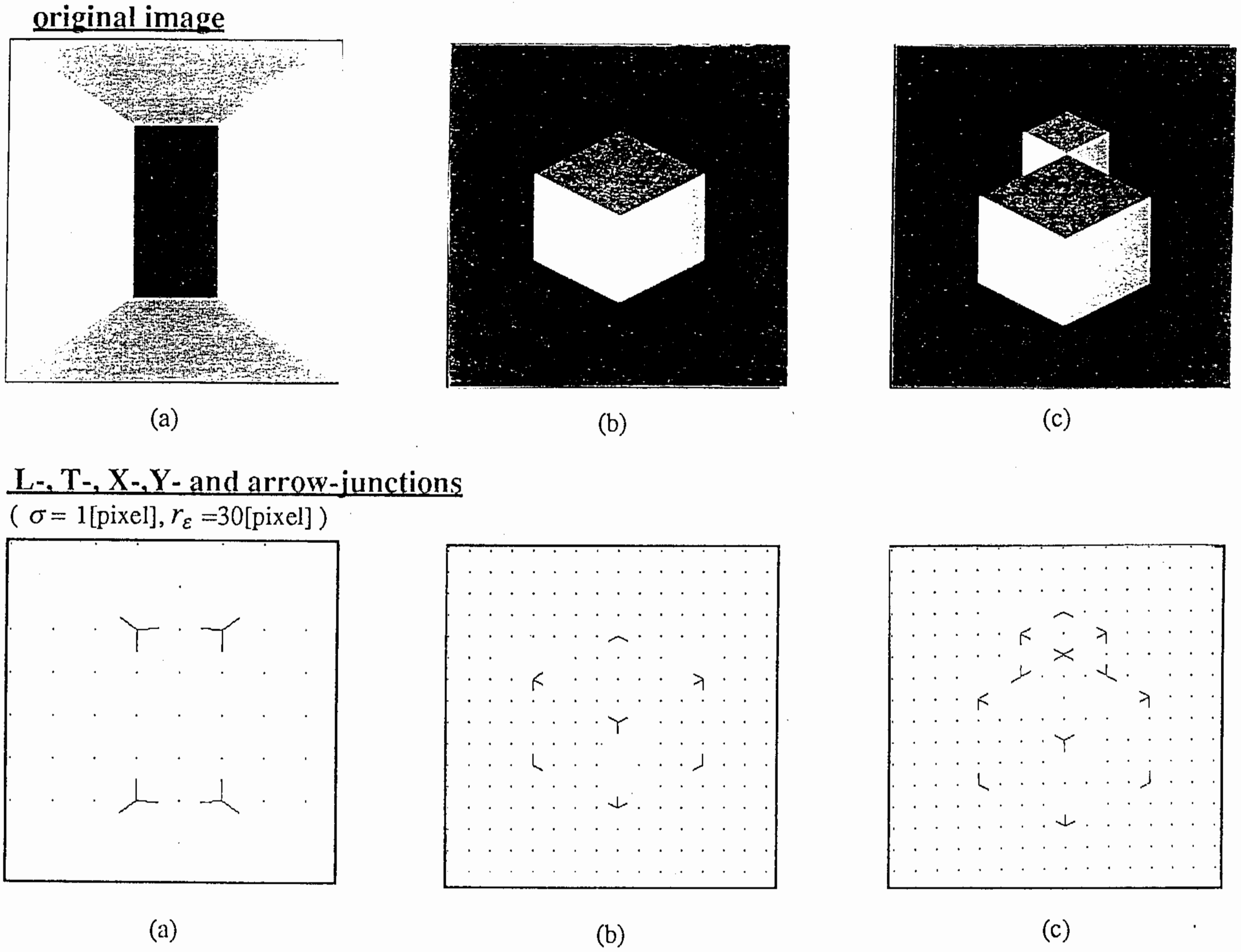

Original image
Detecting orientation
triple orientation
double orientation
single orientation
Convolution with third-order derivatives of window-function
Convolution with second-order derivatives of window-function
Convolution with single-order derivatives of window-function
Solving eigenvalue problem
Solving eigenvalue problem
Solving eigenvalue problem
Selecting the orientation based on each eigenvalue
Classifying junctions
Calculating each sum of directional derivatives along
$\vec{v}$ $(\vec{v} = \vec{p}_1, \vec{p}_2, \vec{p}_3, -\vec{p}_1, -\vec{p}_2, -\vec{p}_3)$
and $\vec{v}_\perp$ $(\vec{v}_\perp = \vec{p}_{1\perp}, \vec{p}_{2\perp}, \vec{p}_{3\perp}, -\vec{p}_{1\perp}, -\vec{p}_{2\perp}, -\vec{p}_{3\perp})$
Classifying L-, T-, X-, Y- and arrow-junctions based on Condition 1 and Condition 2
Detecting L-, T-, X , Y- and arrow-junctions

Fig 7 : Flow of the one-shot algorithm for detecting L- , T-, X-, Y- and arrow-junctions



original image

L-, T-, X-,Y- and arrow-junctions

(σ= 1[pixel], r_ε =30[pixel])

Fig 8 : Detecting L-, T-, X-, Y- and arrow-junctions from artificial images