Log-Polar Coordinate Image Space for the Efficient Detection of Vanishing Points

Kyung-Seok Seo, Chang-Joon Park, and Heung-Moon Choi

ABSTRACT—Log-polar coordinate image space is proposed as a solution for the problem of unbounded accumulator space in the automatic detection of vanishing points. The proposed method can detect vanishing points at high speed under small memory requirements, as opposed to conventional image space based methods.

Keywords—Vanishing point detection, log-polar coordinate, image space.

I. Introduction

The vanishing point (VP), caused by the perspective effects of 3D-to-2D projection, is one of the invariant features in the estimation of image depth, object dimension, and 3D structure. Since these features are widely used in the fields of computer vision, various techniques regarding precise VP detection (VPD) are rigorously studied. In general, architectural objects have enough parallel lines, stretched in vertical or horizontal directions in 3D space that these lines are clustered and the cross points in 2D projection are accumulated in space in order to detect VPs [1]. Accumulator spaces for the VPD are Gaussian sphere [2]-[4], Hough space [5], [6], and image space [7], [8].

Barnard [2] proposed a Gaussian sphere as a bounded space into which lines in the 2D image space are projected, and thereby, both the finite and infinite VPs can be detected effectively. Tuytelaars and others [6] proposed a space-bounding technique for the unbounded parameter space of a cascade Hough transform in order to detect VPs and lines.

Manuscript received May 03, 2006; revised Aug. 31, 2006.

Kyung-Seok Seo (phone: + 82 53 940 8613, email: ksseo@pdp.knu.ac.kr) and Heung-Moon Choi (email: pdpcs@pdp.knu.ac.kr) are with the Department of Electronics, Kyungpook National University, Daegu, Korea.

Chang-Joon Park (email: chjpark@etri.re.kr) is with Digital Content Research Division, ETRI, Daejeon, Korea.

Gaussian sphere and Hough space based methods [2]-[6] require camera calibration or lose some important geometrical information because of space transformations [8].

In methods based on image space, VPs accumulate onto an image space in order to solve problems of camera calibration requirements and information loss. Rother [8] used image space as an accumulator space to detect VPs without any information loss. In this method a set of cross points between lines is used as the accumulator space. The size of the accumulator space, however, depends on the number of detected lines and it causes a complex search step in order to find the VPs from the accumulator space.

In this letter, log-polar coordinate image space (LPCIS) is proposed as a bounded accumulator space for the efficient detection of VPs, using small memory requirements, at high speed and with high detection accuracy.

II. Line Error for VPD

In a digital image, the detected lines inherently have quantization errors as shown in Fig. 1 [9]. The line error E_l is defined as

$$E_l = d_l \tan \alpha_l , \tag{1}$$

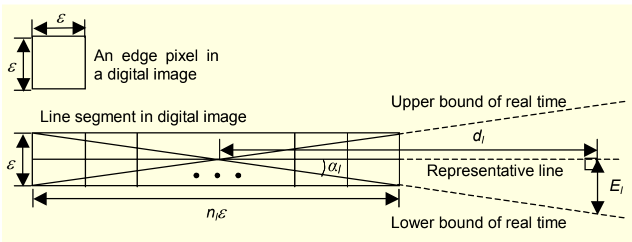

Fig. 1. Line error in a digital image.

where d_l is the distance from the center of a line segment and α_l is the angle between a representative digital line and an upper or lower boundary of analog lines. The line length parameter $\tan\alpha_l$ is in inverse proportion to line length:

$$\tan\alpha_l = \frac{\varepsilon}{n_l\varepsilon} = \frac{1}{n_l}, \tag{2}$$

where ε is the length of a pixel square and n_l is the number of pixels that belong to a line segment in a digital image. Line error E_l increases in direct proportion to distance d_l and in an inverse proportion to line length n_l. Thus, the cell size of the accumulator space for VPD should grow in proportion to the distance between the image origin and the position of a cross point.

III. VPD Based on LPCIS

For VPD, lines are fitted and clustered and then cross points of each line cluster accumulate on LPCIS. The most voted point is selected as the VP for each line cluster.

1. Log-Polar Coordinate Image Space

Each cell of a log-polar coordinate has a fixed angle and its length is in proportion to the distance from the origin. Those properties of the coordinate resemble the properties of line error. In the proposed method, the quantization intervals of the angle and distance axis of LPCIS are defined from the error of the detected lines. The angle interval $\Delta\alpha$ is derived from (2) as

$$\Delta\alpha = \tan^{-1}\frac{1}{\overline{n_l}}, \tag{3}$$

where $\overline{n_l}$ is the average length of detected lines. The distance interval Δd is derived from the average line error of (1) as

$$\Delta d = \frac{d}{\overline{n_l}}, \tag{4}$$

where d is the distance to the origin. The LPCIS is implemented in discrete fashion by a linked list structure. A cell is allocated when a cross point between lines occurs and then is added to the list structure.

For cross point $p_c(x_c, y_c)$, the angle α_c and the distance d_c are computed as

$$\alpha_c = \tan^{-1}(y_c / x_c), \tag{5}$$

$$d_c = \sqrt{x_c^2 + y_c^2}\ . \tag{6}$$

The distance number η_c in the discrete log-polar coordinate is computed as

$$\eta_c = \frac{\ln d_c}{\ln(\overline{n_l}+1) - \ln(\overline{n_l}-1)}. \tag{7}$$

The denominator of this equation is constant.

2. Line Detection and Clustering

For line detection, edges are obtained by the Canny operator. Fitted lines are determined by linking the edge pixels with a similar orientation. The detected line is parameterized on the polar line model

$$\rho = x\cos\theta + y\sin\theta, \tag{8}$$

where ρ is the distance of the line and the origin of the image, and θ is the line orientation.

Before the cross points of the detected lines have accumulated, the lines that are toward the same VP are clustered. VPs of an architectural environment have three mutually orthogonal directions [7]. In order to cluster lines with orthogonal direction, the dominant direction of a line cluster is determined by the method based on an orientation histogram. The remaining line clusters are determined on the assumption of the orthogonal direction of the VPs.

3. VPD Using the LPCIS

Cross points p_c of the lines in each cluster are computed respectively as

$$p_c = \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \frac{1}{\Delta}\begin{bmatrix} \sin\theta_j & -\sin\theta_i \\ -\cos\theta_j & \cos\theta_i \end{bmatrix}\begin{bmatrix} \rho_i \\ \rho_j \end{bmatrix}, \tag{9}$$

where Δ with zero value is excluded from computation:

$$\Delta = \begin{vmatrix} \cos\theta_i & \sin\theta_i \\ \cos\theta_j & \sin\theta_j \end{vmatrix} \neq 0\ . \tag{10}$$

The VP candidates are obtained by voting of all cross points. Finally, the most voted point in the accumulator space is selected as the VP for each line cluster.

The size of an accumulator space is determined by the variation of cross point positions, resulting in the low time/memory complexities of the proposed method. Moreover, cross points can be detected without noticeable loss of precision because the definition of LPCIS is based on line error properties.

IV. Results and Discussion

The LPCIS-based VPD was applied to real images that contained buildings. From the input image, lines were precisely detected and clustered as shown in Fig. 2. Each group is distinguished by a different color.

The percentage error E_p of a detected VP is defined as

$$E_p = \frac{\|v_T - v_D\|_2}{\|v_T\|_2} \times 100 , \tag{11}$$

where, v_T is the manually obtained true VP and v_D is the detected VP. The average percentage error $\overline{E_p}$ of the detected VPs, memory requirements, and processing time for various real images are compared in Table 1. Experiments were conducted on 1200×900 sized images containing about a thousand lines using a Pentium 4, 2.8 MHz CPU.

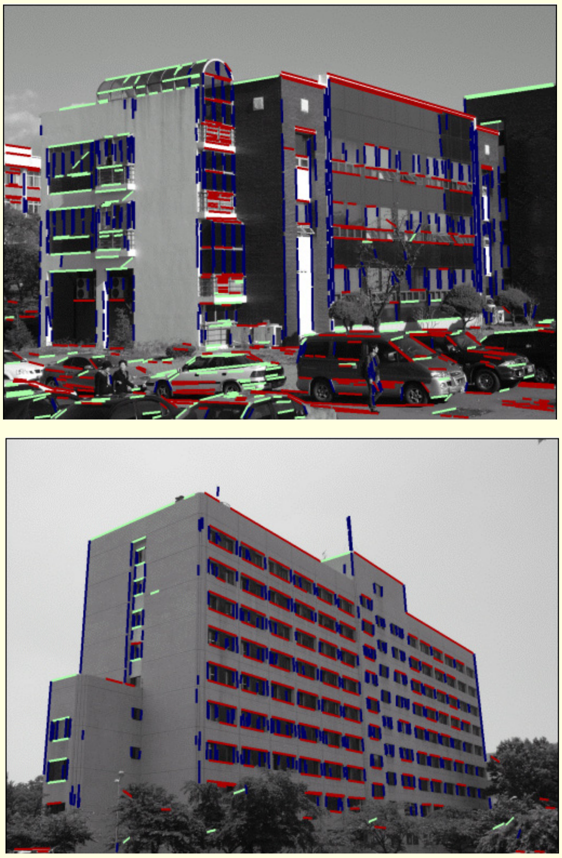

Fig. 2. Two examples of line detection and clustering.

Table 1. A comparison of the average percentage error, memory requirements, and processing time.

	$\overline{E_p}$	Memory requirements	Processing time
Proposed method	0.9%	27 kbyte	0.6 s
Method based on image space [8]	0.7%	1.2 Mbyte	33 s

The conventional image space based method [8] can detect VPs with high accuracy; however, images which contain a lot of lines require high time/memory complexities as shown in Table 1. The proposed method can detect VPs with low time/memory complexities and without noticeable increments in detection error.

V. Conclusion

VPs can be efficiently detected, at high speed, using the proposed LPCIS as a bounded accumulator space which requires only a small amount of memory. Since LPCIS is based on the properties of digitized line error, it can efficiently detect VPs. Moreover, the proposed method can be conducted at high speed and implemented with only a small amount of memory even when an input image has many lines. The proposed method can be applied to embedded system-based VPD, image-based modeling (IBM), and camera calibrations.

References

[1] M. Straforini, C. Coelho, and M. Campani, "Extraction of Vanishing Points from Images of Indoor and Outdoor Scenes," *Image and Vision Computing*, no. 11, 1993, pp. 91-99.

[2] S.T. Barnard, "Interpreting Perspective Images," *Artificial Intelligence*, no. 21, 1983, pp. 435-462.

[3] L. Quan and R. Mohr, "Determining Perspective Structures Using Hierarchical Hough Transform," *Pattern Recog. Lett.*, no. 9, 1989, pp. 279-286.

[4] M. Magee and J. Aggarwal, "Determining Vanishing Points from Perspective Images," *Computer Vision and Graphics and Image Proc.*, no. 26, 1984, pp. 256-267.

[5] H. Nakatami, S. Kimura, and O. Saito, "Extraction of Vanishing Point and its Application to Scene Analysis Based on Image Sequence," *Proc. of the Fifth Inter. Con. on Patern Recog.*, 1980, pp. 370-372.

[6] T. Tuytelaars, L. Gool, M. Proesmans, and T. Moons, "The Cascaded Hough Transform as an Aid in Aerial Image Interpretation," *ICCV*, 1998, pp. 67-72.

[7] V. Cantoni, L. Lombardi, M. Porta, and M. Sicard, "Vanishing Point Detection: Representation Analysis and New Approaches," *ICIAP*, 2001, pp. 90-94.

[8] C. Rother, "A New Approach to Vanishing Point Detection in Architectural Environments," *Image & Vision Computing*, no. 20, 2002, pp. 647-655.

[9] E.J. Kim, M. Haseyama, and H. Kitajima, "Fast Line Extraction from Digital Images Using Line Segments," *J. IEICE*, vol. J84-D-II, no. 8, Aug. 2001, pp. 1566-1579.