

Accurate Corner Detection using 4-directional Edge Labeling and Corner Positioning Templates

Eun-Jin Park* and Doo-Hyun Choi†

Abstract – Corner positioning templates are proposed in order to detect the accurate positions of corners that are extracted using 4-directional edge labeling. Top-down and bottom-up directional labeling are used to label the edge segments with four kinds of labels according to their directions. The points whose labels have changed are then determined as corners. The exact positions of the missing corners due to the disconnected edges are detected through the corner positioning templates that are determined according to the labels of start-points and end-points after the two-pass edge labeling. Experiment results show that the proposed method can detect the exact positions of the real corners.

Keywords: Corner detection, 4-directional edge labeling, Positioning template

1. Introduction

Corner detection in an image plays an important role in computer vision because corners in images include useful information and are invariant to geometric deformation. Corners are used as feature points in many applications, such as image registration, moving object tracking [1], object recognition, and so on. They are also important as the geometrically invariant image features, particularly in the watermarking techniques that are generally sensitive to geometric distortions, such as rotation, scaling, and translation [2], [3]. For these reasons, many corner detection methods have already been explored.

Previous works on corner detection can be categorized into three kinds, namely, edge-based corner detectors, gray level derivative-based corner detectors, and template-based corner detectors [4]. Edge-based detectors look for the maximum curvature points or search for the intersection points of the edge of the polygon line segments. Mokhtarian and Suomela [5] proposed a curvature scale-space (CSS) method, in which corners are detected, tracked, and localized through the curvature analysis based on the multiple scales. This method is highly sensitive to the local variation and noise on the curve. In addition, the CSS corner detector requires appropriate Gaussian smoothing-scale selection, which is a difficult task [6].

Gray level derivative-based detectors calculate the distribution of curvature of a grayscale image and take the point of maximum curvature as the corner. These detectors can avoid the errors existing in the algorithms based on the edge. Kitchen and Rosenfeld [7] proposed a cornerness measure based on the change of gradient direction along an

edge contour multiplied by the local gradient magnitude. Meanwhile, Harris and Stephens [8] proposed a method, in which corners are detected by identifying curvature changes by differential analysis. The corner features obtained by the Harris corner detection method have relatively reasonable uniformity, and the feature points can be quantitatively extracted. However, it has greater redundancy and poorer positioning accuracy when it comes to corner detection. In extreme cases it may even miss some real corners [9].

Template-based detectors, on the other hand, determine the similarity or correlation between a given template and all sub-windows in a given image [10]. However, using these methods, it is impossible to design the templates that can cover all orientations and corner angles [11]. Although the templates have been designed, the computational load required for matching is still very high.

In this paper, a template-based corner detecting method not in the grayscale image but in the edge image is combined with the edge based method. The aim of the proposed method is to detect the exact positions of extracted corner candidates from the proposed 4-directional edge labeling. In the experimental results for the widely used test images, the proposed corner detection method based on 4-directional edge labeling, followed by corner positioning templates, can efficiently detect the exact positions of the real corners.

2. An Overview of the Proposed Algorithm

The flowchart of the proposed corner detection method is shown in Fig. 1. The first step was the extraction of the edge image with 1-pixel width edge using the Canny edge operator, followed by the top-down labeling for the extracted edge. Top-down labeling detects start-points and

† Corresponding Author: School of Electronics Engineering, Kyungpook National University, South Korea. (dhc@ee.knu.ac.kr)

* Dept. of Electronics, Graduate School, Kyungpook National University, South Korea. (cloudzeenee@iisl.knu.ac.kr)

Received: September 17, 2010; Accepted: April 8, 2011

end-points as corner candidates; it also detects junction points as corners. The start-point and end-point comprised the first and last pixels of each edge segment, respectively. The junction points are the pixels whose labels are changed after the top-down labeling. The next step was the bottom-up labeling, which was exactly the same as the top-down labeling except the labeling order. Bottom-up labeling also detects start-points, end-points, and junction points. Similar to top-down labeling, start-points and end-points are corner candidates, while junction points are corners. After conducting the two-pass 4-directional edge labeling, the changes of labels were checked for all the start-points and the end-points. The points whose labels changed were determined as corners. After this step, there were some start-points and end-points whose labels were unchanged because of the disconnected edges caused by the edge operators. For these remaining start-points and end-points, the exact corner positions were determined in the last step using the proposed corner positioning templates.

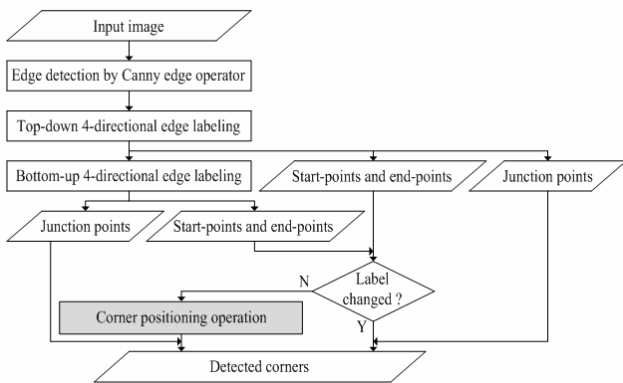


Fig. 1. Flowchart of the proposed corner detection method

3. 4-directional Edge Labeling

Corners are defined as the end-points or intersection points of line segments; however, performing line extraction from edge images is a complicated and time consuming process, especially in the case of non-parametric curves and planar curves. Given that each one of the edge segments has its own direction, corners are the points at which two or more edge segments with different directions encounter. In order to use this property, one of four different directional labels was assigned to each pixel of the edge segments, in accordance with the direction of the edge segment.

In the top-down labeling, finding and labeling edge pixels from the edge image are generally processed in a left-to-right and top-to-bottom manner. Therefore, only four kinds of directions are enough to label all edge segments. The proposed four kinds of directions are the east, southeast, south, and southwest. The four directional labels for top-down pass and bottom-up pass are shown in

Fig. 2. The priority of searching the next pixel in order to determine its label is the same as that in Fig. 2.

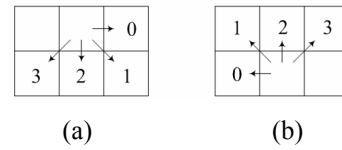


Fig. 2. 4-directional labels: (a) for top-down labeling; (b) for bottom-up labeling

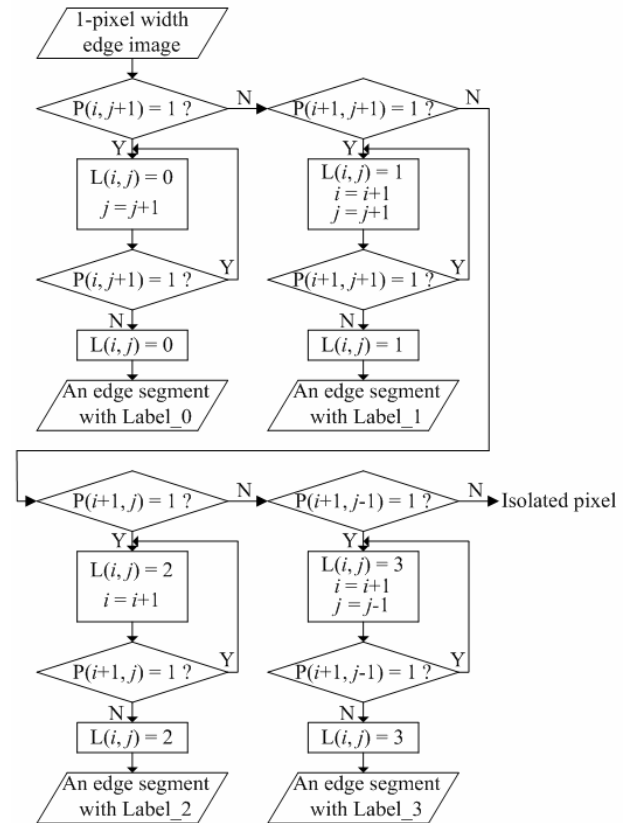


Fig. 3. Flowchart of the top-down labeling for an edge segment

The top-down labeling for just one edge segment is shown in Fig. 3. The first pixel $P(i, j)$ of an arbitrary edge segment was registered as a start-point. The label allotted to a start-point $[L(i, j)]$ was determined by the next pixel found according to the searching priority. Once the label of the start-point was determined, labeling for the next pixels was continued in the same direction until there was no pixel left in the same direction. The last pixel of the same edge segment was registered as an end-point. After the labeling for one edge segment, the edge image was scanned to find another edge segment. This process was repeated until all pixels of the edge image are labeled. If the next pixel was already labeled during the labeling process, it was registered as a junction point.

After labeling each edge segment, corner detection was accomplished by finding only the pixel positions whose

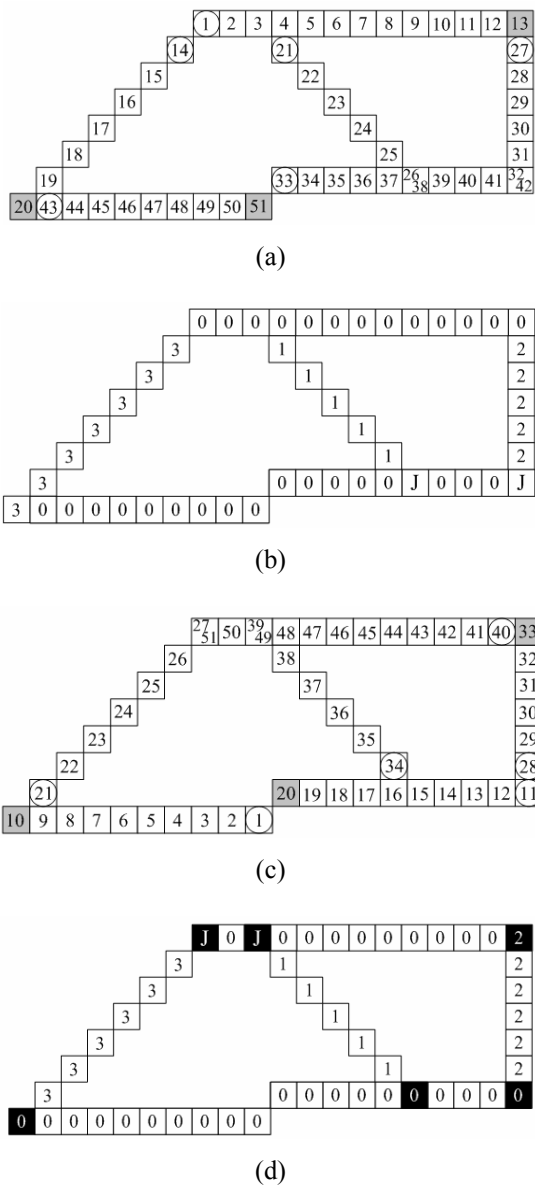


Fig. 4. (a) Labeling sequences for top-down labeling, (b) labeled edge image of (a); (c) labeling sequences for bottom-up labeling; (d) labeled edge image of (c)

directional labels changed. Given that only the start-points, end-points, and junction points can be considered as corners, the changes of label for the rest of the edge pixels need not be checked. In addition, the changes of labels were checked for only the start-points and the end-points except the junction points because the latter must be corners. This method can detect corners rapidly and efficiently without calculating the curvature and angle.

As an example, the labeling sequences of the proposed method are shown in Fig. 4(a) for top-down pass and Fig. 4(c) for bottom-up pass. The circled and shaded pixels stand for the start-points and the end-points, respectively. The start-points are pixels 1, 14, 21, 27, 33, and 43 for the top-down pass and 1, 11, 21, 28, 34, and 40 for the bottom-

up pass. The end-points are pixels 13, 20, and 51 for the top-down pass and 10, 20, and 33 for the bottom-up pass. The label of pixel 26 (or 38) changed from 1 to 0, and the label of pixel 32 (or 42) changed from 2 to 0 during the top-down pass. The label of pixel 27 (or 51) changed from 3 to 0, and the label of pixel 39 (or 49) changed from 1 to 0 during bottom-up pass. These points are registered as junction points. The junction points and the labeling results are shown in Figs. 4(b) and 4(d) for the top-down pass and bottom-up pass, respectively. The labels of pixels 13 and 20 in Fig. 4(a) changed from 0 to 2 and from 3 to 0, respectively, after the bottom-up labeling. These points whose labels changed and junction points are considered as corners. Pixels 1, 3, 13, 20, 26, and 32 are considered as corners, while pixels 33 and 51 are not considered as corners in Fig. 4(a). Given that the labels of these two pixels were not changed after two pass labeling, they were excluded from detected corners. The final detected corner pixels are represented with black background in Fig. 4(d).

4. Corner Positioning Templates

Most of the edge segments detected from the practical images are not perfectly connected to others. For this reason, detecting the exact positions of the corners is burdensome.

Although the intersection point of two different edge segments does not appear due to the edge operators, it can be calculated by the line equations of two edge segments. This means that a corner can be determined using only the directions of two edge segments comprising the corner. In order to use this property, the corner positioning templates based on the results of the 4-directional edge labeling method are proposed in this paper. After conducting the 4-directional edge labeling, each pixel in the image has its own label. Due to the fact that the label of each pixel stands for the direction of an edge segment in the proposed method, the positions of corners can be determined without finding the line equations. As mentioned above, only the start-points and the end-points were considered for the corner positioning templates. The types encountered by two edge segments according to the kinds of start-point or end-point and their respective labels are shown in Fig. 5. In

S(0), S(1)	S(0), E(1)	E(0), S(1)	E(0), E(1)	S(1), S(2)	S(1), E(2)	E(1), S(2)	E(1), E(2)
S(0), S(2)	S(0), E(2)	E(0), S(2)	E(0), E(2)	S(1), S(3)	S(1), E(3)	E(1), S(3)	E(1), E(3)
S(0), S(3)	S(0), E(3)	E(0), S(3)	E(0), E(3)	S(2), S(3)	S(2), E(3)	E(2), S(3)	E(2), E(3)

Fig. 5. 24 types that two edge segments encounter

the figure, S and E refer to the start-point and the end-point respectively; 0, 1, 2, and 3 in brackets are the respective labels of the segments.

Some possible positions of corners exist in each type, and these can be easily determined using the corner positioning templates. For the case of $[S(0), S(1)]$, a start-point with label 0 and a start-point with label 1, the possible positions of two edge segments are depicted in Fig. 6. The white circles represent the start points of each edge segments. As shown in the figure, each corner position can be determined with only the positions of the two circled pixels. The C stands for the determined corner position of each case in Fig. 6. With this property, the exact positions of corners due to the disconnected edges can be determined using the proposed method. Given that the positions and the labels of all start-points and end-points are listed on a look-up table in the proposed method, the positioning for missing corners can be accomplished rapidly and efficiently.

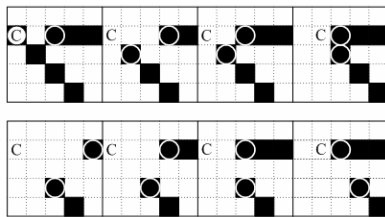


Fig. 6. Determined corners according to the possible positions of two edge segments in the case of $S(0)$, $S(1)$

The conventional template matching methods compare portions of images against one another. The matching process moves the template images to all possible positions in a larger source image. Afterwards, it computes the numerical indices that indicate how well the templates match the image in that position. Therefore, these methods require many different templates in one position for each corner. In contrast, in order to determine a corner position, the proposed method used only a single template, which was assigned to that corner according to the relevant start-point(s) or end-point(s) and their labels. Thus, the time required to obtain accurate corner positions was shorter.

5. Experimental Results

The experimental results on the widely used house image are shown in Fig. 7. For the Kitchen and Rosenfeld detector, some false positives and false negatives exist as shown in Fig. 7(a). The detected corners by the CSS detector are shown in Fig. 7(b). This method sometimes fails to detect the missing corners when the edge segments composing corners are broken due to the edge operator. These false negatives appear in several points. Due to the

smoothing operation of the CSS detector at the high scale space, this method tends to ignore the ambiguous corners on the discontinuous short edge segments, as shown in the right-bottom area of Fig. 7(b). The result of the proposed method with only 4-directional edge labeling is also shown in Fig. 7(c). There are some false negatives due to the disconnected edges. In contrast, most of the true corners are detected correctly in the proposed method with 4-directional edge labeling and corner positioning templates as shown in Fig. 7(d). The missing corners that are composed by the disconnected two different edges are also detected and accurately located using the proposed corner positioning templates.

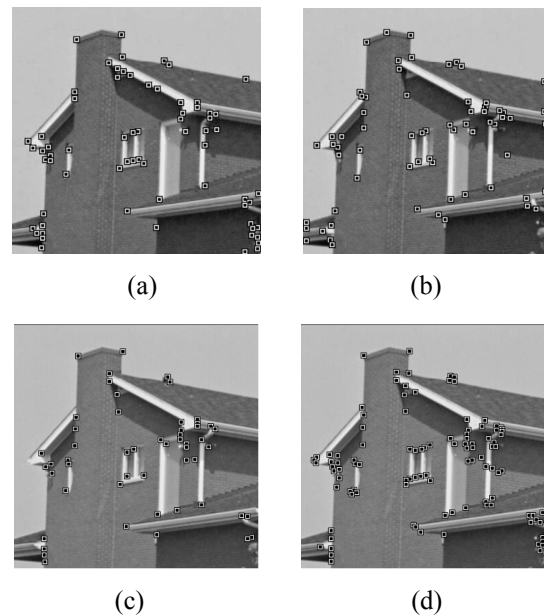


Fig. 7. Detected corners of the house image: (a) Kitchen and Rosenfeld; (b) CSS; (c) 4-directional edge labeling; (d) 4-directional edge labeling with corner positioning templates

The results on the lab image are shown in Fig. 8. For the Kitchen/Rosenfeld detector and CSS detector, several false positives and false negatives are observed, as shown in Figs. 8(b) and (c). The result of the proposed method is shown in Fig. 8(d). The false positives and false negatives of the proposed method are less than both Kitchen/Rosenfeld detector and CSS detector. The positions of the detected corners using the proposed method are more accurate than the conventional methods.

6. Conclusion

An accurate corner detection algorithm using corner positioning templates suitable for the 4-directional edge labeling is proposed in this paper. After labeling all edge segments according to their directions with four directional labels, the corners are detected where the labels changed.

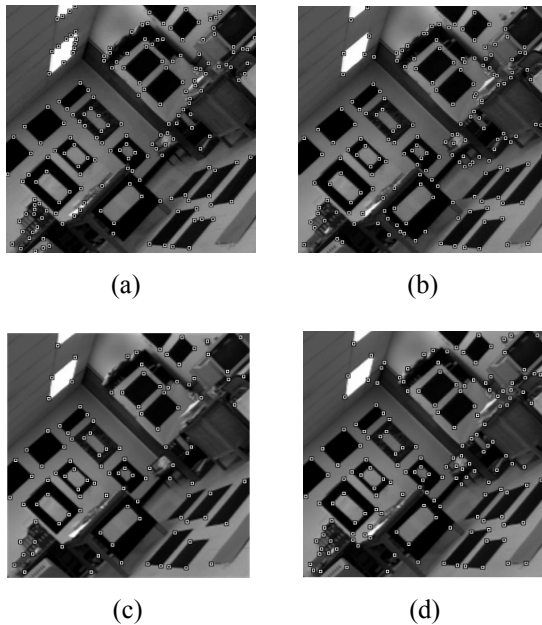


Fig. 8. Detected corners of the lab image: (a) Kitchen and Rosenfeld; (b) CSS; (c) 4-directional edge labeling; (d) 4-directional edge labeling with corner positioning templates

The exact positions of the missing corners resulting from the discontinuous edges brought about by the edge operators can also be determined using the corner positioning templates. Experimental results for the well-known test images show that the proposed method not only eliminates false corners effectively but also localizes the real corner pixels exactly. Furthermore, the proposed method can detect the exact corner positions despite the discontinuous edge contours without any calculation of curvatures and cosine angles.

References

- [1] In Su Kim, Hong Seok Choi, Kwang Moo Yi, Jin Young Choi, and Seong G. Kong, "Intelligent visual surveillance – a survey," *International Journal of Control, Automation, and Systems*, vol.8, no.5, pp.926-939, 2010.
- [2] H. Tian, Y. Zhao, R. Ni, and G. Cao, "Geometrically robust image watermarking by sector-shaped partitioning of geometric-invariant regions," *Opt. Express*, vol.17, no.24, pp.21819-21836, 2009.
- [3] L. Zhang, G. Qian, W.Xiao, and Z. Ji, "Geometric invariant blind image watermarking by invariant Techebichef moments," *Opt. Express*, vol.15, no.5, pp.2251-2261, 2007.
- [4] W. Xu, X. Huang, Y. Wang, and S. Bai, "A biologically motivated corner detection method based on the oriented receptive fields of simple cortical cells," in *Proceedings of IEEE Int. Conf. on Biomedical Engineering and Computer Science*, 2010.
- [5] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Trans. Pattern Anal. Machine Intell.* vol.20, no.12, pp.1376-1381, 1998.
- [6] A. Mohammad and L. Guojun, "Robust image corner detection based on the chord-to-point distance accumulation technique," *IEEE Trans. Multimedia*, vol.10, issue 6, pp.1059-1072, 2008.
- [7] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Lett.* vol.1, no.2, pp.95-102, 1982.
- [8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of Fourth Alvey Vision Conference*, 1988.
- [9] Z. Jin-ping, L. Yong-xiang, J. Chun-wang, G. Da-meng, and L. Jie, "An improved Harris corner distraction method based on B_spline," in *Proceedings of IEEE Int. Conf. on Information management and Engineering*, 2010.
- [10] A. Singh and M. Sheier, "Gray level corner detection a generalization and a robust real time implementation," *Computer Vision, Graphics, and Image Processing*, vol.51, pp.54-69, 1990.
- [11] Z. Zheng, H. Wang, and E. K. Teoh, "Analysis of gray level corner detection," *Pattern Recognition Lett.*, vol.20, issue 2, pp.149-162, 1999.
- [12] J. Cooper, S. Venkatesh, and L. Kitchen, "Early jump-out corner detectors," *IEEE Trans. Pattern Anal. Machine Intell.* vol.15, no.8, pp.823-828, 1993.
- [13] H. T. Sheu and W. C. Hu, "A rotationally invariant two-phase scheme for corner detection," *Pattern Recognition*, vol.29, issue 5, pp.819-828, 1996.



Eun-Jin Park received his B.S. and M.S. degrees in Electronics Engineering from Kyungpook National University, Daegu, South Korea in 1992 and 1999, respectively. His research interests include parallel distribution processing and signal processing for industrial applications.



Doo-Hyun Choi received his B.S. degree in Electronics Engineering from Kyungpook National University, Daegu, South Korea in 1991, He then received his M.S. and Ph.D. degrees in Electrical and Electronic Engineering from Pohang University of Science and Technology (Postech), Pohang, South

Korea in 1993 and 1996, respectively. He is currently a Professor at the School of Electronics Engineering of Kyungpook National University. His research interests include soft-computing algorithm and applications, bioinformatics, and signal processing for industrial applications.