ICARCV '90

INTERNATIONAL CONFERENCE ON AUTOMATION, ROBOTICS AND COMPUTER VISION

19-21 September 1990

PROCEEDINGS

ORGANISED BY

School of Electrical and Electronic Engineering
Nanyang Technological Institute, Singapore

The Institution of Engineers, Singapore

McGraw-Hill Book Co
Singapore New York St. Louis San Francisco Auckland Bogotá
Caracas Colorado Springs Hamburg Lisbon London Madrid Mexico
Milan Montreal New Delhi Oklahoma City Panama Paris San Juan
São Paulo Sydney Tokyo Toronto

Extracting and Combining Edges using Gradient Vector

Mutsuhiro Terauchi, Noriaki Yokota, Koji Ito and Toshio Tsuji

Faculty of Engineering, Hiroshima University
Shitami, Saijo, Higashi-hiroshima 724 Japan

Abstract: This paper presents an approach to extract and combine straight or curved line edges in gray level images. Pixels are grouped into some line-support regions. Then straight or curved edges are assigned on the regions according to an estimation of their curvature. Finally the extracted edges are combined to form a boundary line. In our method fitting of the straight or curved line can be performed by the same algorithm. Some simulation experiments will be carried out for a TV image.

1.Introduction

Edges are the primitive in low level vision but have quite important cues for image understanding and reconstruction of 3D scene [1]. Many researchers have been using the edge as a basic information of hierarchical image structures. Edges are usually defined as local discontinuities or steep changes in some image features, such as image intensity or texture. These edges have been extracted by signal processing methods closely related to the human visual processing. For example, D.Marr and E. Hildreth showed that DOG (the difference of two Gaussian) zero-crossing operator was the best engineering approximation to $\nabla^2 G$ filter which gave the basis for the psychophysically determined channels [2],[3]. These edges are emphasized by local spatial differential operator with respect to the image, eg. Roberts, Sobel, Laplacian operator etc [4],[5]. Thus there have been presented a lot of researches in the literature [6],[7]. However the line extraction has been still a difficult problem.

Most of the edge and line extraction algorithms uses the magnitude of the intensity change as a measure of the local edge. While edge orientation information may be used to modulate the grouping process applied to the strong edges, the edge magnitude usually has the central and dominating influence. We propose a new approach to extract and combine the straight and curved lines in intensity images. J.B.Burns.et.al. grouped pixels into line-support regions of similar gradient orientation, and according to dual overlapping-partition technique they solved the over-merging and fragmentation tradeoff [8].

After extracting edges from gray level image, next step necessary is to group them. It is called integration or combining of edges, where roughly similar edges and/or line segments are grouped into larger lines or boundaries between sets of lines that differ in certain way, which reflects the structure of the scene. The Gestaltists thought of grouping as being subject to various types of rules, the principle of closure, good continuation, regularity, symmetry, simplicity, and so forth, which were summarized as the Gestalt law of Pragnanz [9]. In general it is pretty difficult to group tokens without knowledge which concerns 3D physical world objects encountered frequently. But to introduce the knowledge to these low level vision is too irrelevant and uninvited. In our research the grouping process is however performed just based on some of the features of edges or line segments.

2.Extraction of Edge Region

Straight or curved edge segments are one of the most important keys for understanding or reconstructing the scene from 2D image. Generally, the edge is set of points where the intensity of pixel (brightness) changes most steeply in its neighborhood. It corresponds to a contour on 2D image and often a ridge or a occluding boundary in 3D space.

2.1 Computation of Gradient Vector

First of all, the 2D image is divided into some regions by gathering the pixels where the gradient vectors have the same direction. The gradient vector is defined as a vector which represents the direction of the steepest change of intensity at a pixel, which has the following properties:

1) It has large value near a boundary,
2) It is perpendicular to the boundary.

The gradient vector is computed using differential operators as follows:

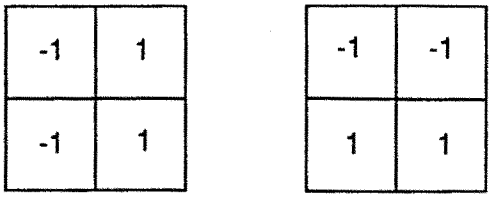
1) Compute the horizontal and vertical elements of the gradient vector, i.e. the intensity differences $G_h(i,j)$, $G_v(i,j)$ between horizontal and vertical neighborhood pairs using the following equations (see Fig.1),

$$G_h(i,j) = -g(i,j) - g(i,j+1) + g(i+1,j) + g(i+1,j+1), \quad (1)$$

$$G_v(i,j) = -g(i,j) + g(i,j+1) - g(i+1,j) + g(i+1,j+1), \quad (2)$$

where $g(i,j)$ denotes gray level intensity at the pixel (i,j).

2) Compute the length and the direction of the gradient vector from following equations.



(a) horizontal (b) vertical

Fig.1 Differential operator.

direction $$\theta = \tan^{-1}\frac{G_v(i,j)}{G_h(i,j)} \quad (3)$$

length $$|\vec{G}| = \sqrt{G_v(i,j)^2 + G_h(i,j)^2} \quad (4)$$

where,

$$G_h > 0 \quad \begin{cases} G_v \geq 0 & \theta = \theta \\ G_v < 0 & \theta = \theta + 2\pi \end{cases}$$

$$G_h = 0 \quad \begin{cases} G_v > 0 & \theta = \pi/2 \\ G_v = 0 & \theta = 0 \\ G_v > 0 & \theta = 3\pi/2 \end{cases}$$

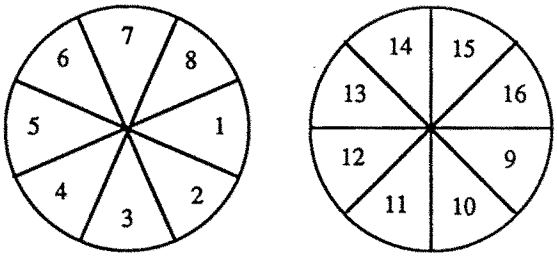
$$G_h < 0 \qquad \theta = \theta + \pi .$$

The above are computed for all pixels.

2.2 Grouping of pixels

The pixels are grouped into some regions based on the directions of the gradient vectors, which is called "gradient region". The direction of the vector will be same along a straight boundary line and change monotonously along a curved boundary line. Therefore, the gradient regions may include some edges. The grouping method proposed by J.B.Burns et.al. is as follows.

1) All the pixels are labeled by one of the eight labels (Ai; i = 1 - 8) shown in Fig.2a according to the direction of the gradient vectors. Furthermore, they are also labeled one of another eight labels (Bi; i = 9 - 16) shown in Fig.2b. That is, each pixel has a couple of labels, say Ai and Bi.
2) Comparing the label of the pixel with its adjacent pixel, if both both have the same label, they are classified into the same group, which forms a region.
3) Although each pixel is included in both the different regions Ai and Bi, the region which consists of more pixels is remained.



a) label (a) b) label (b)

Fig.2 Partion block for gradient vector.

The above algorithm implies that all the edges are approximated by the lines perpendicular to the sixteen different directions with the errors within $\pm \pi/8$.

Further, to reduce a noises mainly caused by digital sampling, we need following low pass filtering,

$$f(i,j) = 0.1(g(i-1,j-1) + g(i,j-1) + g(i+1,j-1) + g(i-1,j) + 2g(i,j) + g(i+1,j) + g(i-1,j+1) + g(i,j+1) + g(i+1,j+1)) , \quad (5)$$

and also the isolated regions which consist of less than 2 pixels are removed.

3.Representation of Edge

3.1.Judgement of linearity

It is assumed that there exists a straight line edge or a curved line edge in the gradient region. Then we can decide whether the straight line or curved line should be fitted using the gradient vectors in the region according to the following algorithm.

1) For each gradient region, compute the center point G and the both side points M and N of the area.
2) Compute the line which runs through the point G and is perpendicular to the direction represented by the label Ai and Bi which is given to the gradient region.
3) The points M and N are projected onto the line, and the projected points are named A and B respectively.
4) A window with $1/10$ width of the length AB is moved along the line AB with the step 1/20. For each step the following cost function is computed.

$$S_i = \frac{1}{\Sigma w_{ij}} \Sigma w_{ij} \ (\theta_{ij} - \theta_k) \quad (6)$$

where,

θij :the direction of the gradient vector of pixel j in sub-region i.

wij : the length of the gradient vector of pixel j in sub-region i.

θk : the direction represented by the label given to the gradient region.

n : the number of pixels involved in sub region

Then the cost function sequence Si (i = 1,..,20) represents how the gradient vectors change along the edges in the image. When the sequence Si is plotted as shown in Fig.3, we can obtain the following remarks from the graph.

a) If $S_i = 0$ for all i like Fig.3a, then fitted line corresponds with the **straight** line edge included in the gradient region.
b) If $S_i \neq 0$ but constant for all i like Fig.3b, the edge is **straight** but has slightly different direction from the fitted line.
c) If Si is not constant, the edge should be fitted by the **curved** line. In this paper, then, Si is approximated by the straight line like Fig.3c, which means that the edge is able to be fitted by the arc.

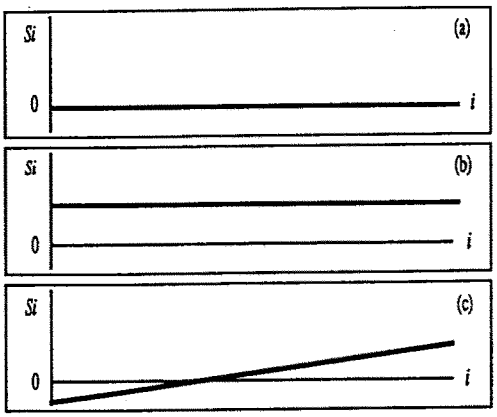


Fig.3 Sequence of Cost function Si

3.2.Curved line fitting

When the edge in the gradient region is fitted by the arc, the radius of the arc is computed using the following equation.

$$R = \frac{d}{2 \sin\left(\frac{S_{max} - S_{min}}{2}\right)} , \qquad (7)$$

where, R : radius of the arc
d : length of the line AB
S_{max} : maximum value of Si
S_{min} : minimum value of Si.

The geometry of the *eq.7* is shown in Fig.4.

The next step is to estimate the position and posture of the arc.

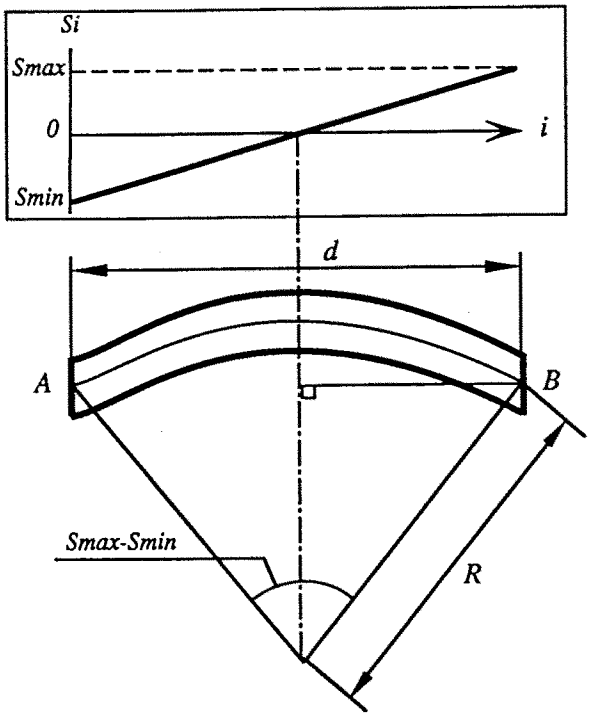


Fig.4 Computation of radius of curvature R

** Depict the straight line q which runs through the point H at $Si = 0$ and has the same direction as the gradient vector at the point H (see Fig.5). Then the curvature center of the fitted arc must lie on the line q and within the width ts of the cross section at H. The exact position Pi of the arc center is determined by the following cost function.

$$\sum_{j=1}^{m} w_j (\psi_{ij} - \theta_j) \rightarrow min , \qquad (8)$$

where, m : the number of pixels in the region.
θ_j : the direction of the gradient vector at the pixel j.
w_j : the length of the gradient vector at the pixel j.
ψ_{ij}: the direction from the pixel j to the position Pi.

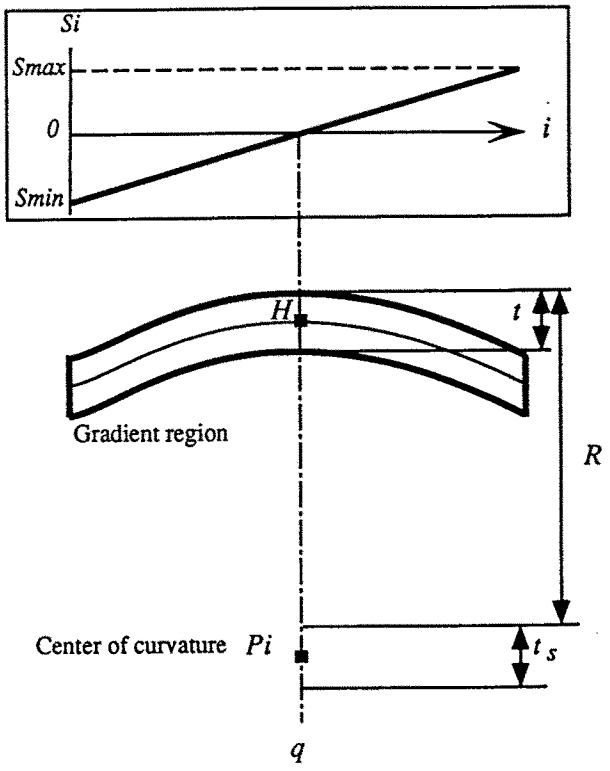


Fig.5 Computation of center of curvature Pi

3.3.Straight line fitting

When $Si = 0$ as shown in Fig.3a, the fitted straight line represents well the edge in the region. When $S_i \neq 0$ and constant, however, the slope of the fitted line is different from the one of the edges in the region. Then the slope should be compensated by the constant value of Si.

Next it is necessary to determine the location in the region of the fitted straight line. Assuming that the intensity level in the region changes monotonously, the middle point of the intensity on the line EF is chosen as shown in Fig.6.

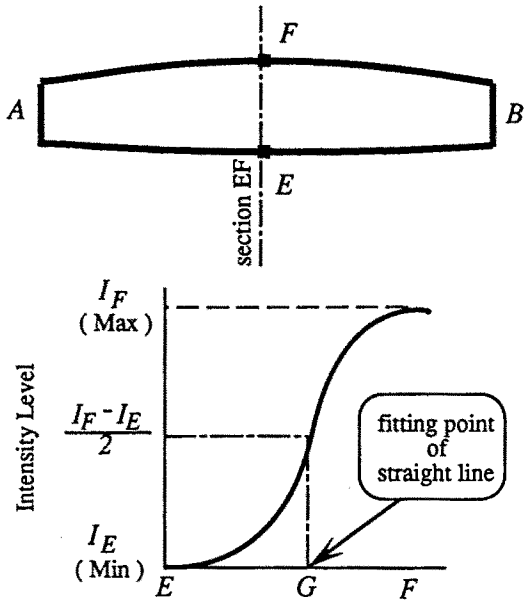


Fig.6 Fitting method for straight line

4. Edge Combining

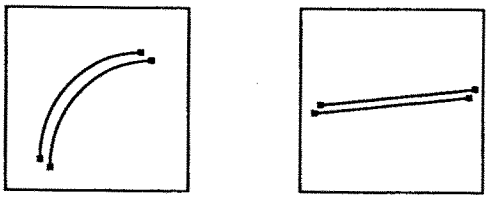
Since the image plane is grouped into the regions based on the gradient vector, fragmentation of the raw line edges is unable to be excluded. Therefore it is necessary to combine the line. In our study the similarity of each feature of line is introduced to judgement for combining. The features used here for straight and curved line are as follows.

Common: terminal positions of line.

For Straight: orientation.

For Curved: curvature, the center of curvature.

When there are two or more lines side by side as shown in Fig.7, then they should not be combined together. Such constraints are implemented by heuristic rules.



(a) curved line (b) straight line

Fig.7 Exceptional case in line combining

5.Experimental Results

Our method is applied for a image of real 3D scene to show the efficiency of it. The process sequence of our method is shown in Fig.8. The input image has 130x130 pixels with 256 level of intensity resolution. The image is filtered by local operator to cut the high frequency noise. Then the gradient vector is computed at every pixels in the image using differential operator. All the pixels are labeled according to the direction of the vector, and the pixels are grouped into the region in two ways. Then two interpretations are integrated to one. After independent regions with less than 2 pixels are rejected, judgement of linearity and fitting of straight and curved line are performed. Finally obtained line segments are combined together by using the feature similarity for each curved and straight line.

The input image used is shown in Fig.9, which contains an indoor scene. The generated region image is shown in Fig.10. In Fig.11 the gradient vectors are superimposed on the region image. The obtained line segments are shown in Fig.11. In Fig.11 there are a few curved line because the input image is sampled data of indoor scene. However curved line is extracted successfully, for example the handle of the board.

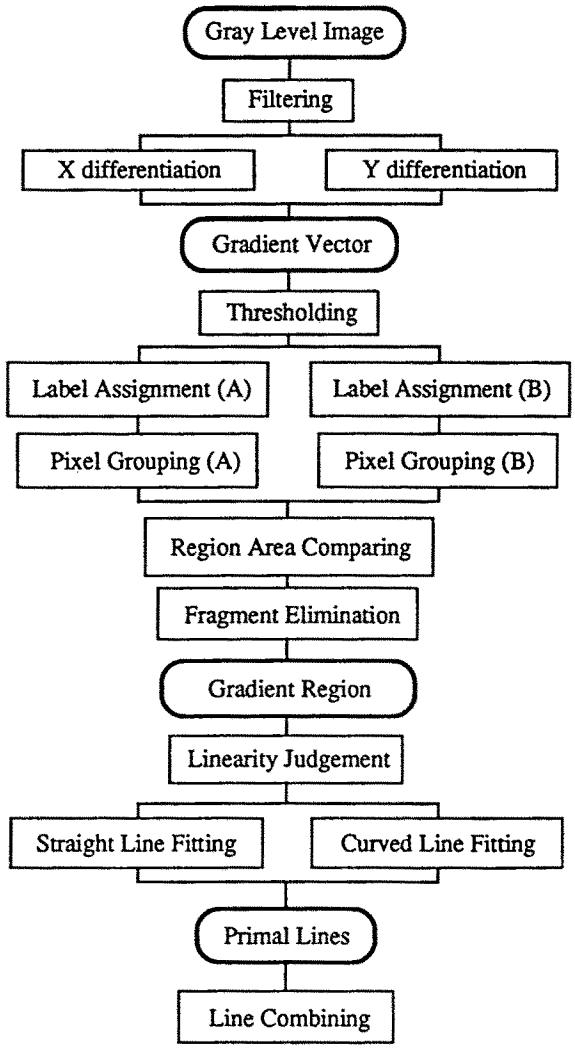


Fig.8 Process Flow Diagram of the System

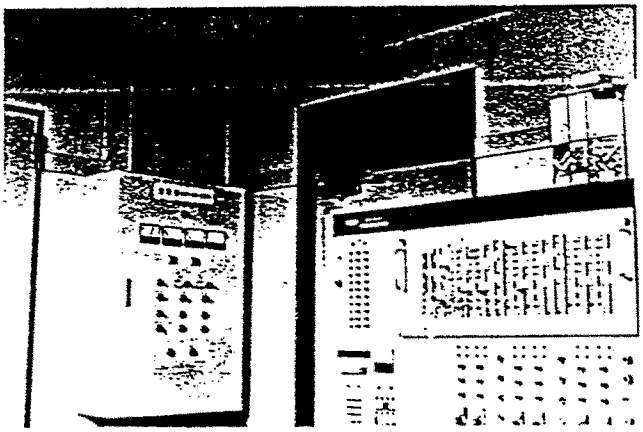


Fig.9 Gray level Input image of indoor scene.

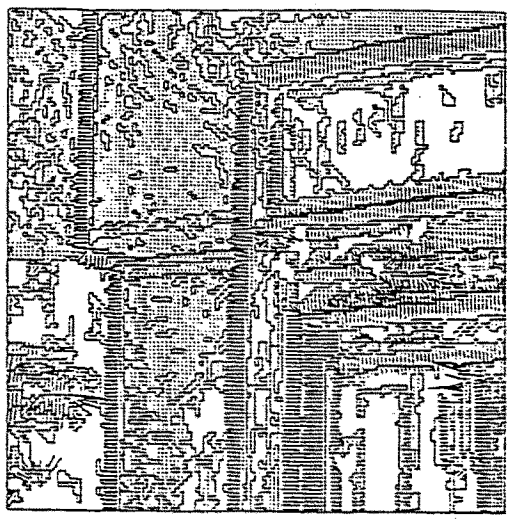


Fig.11 Gradient vectors superimposed on Gradient Region

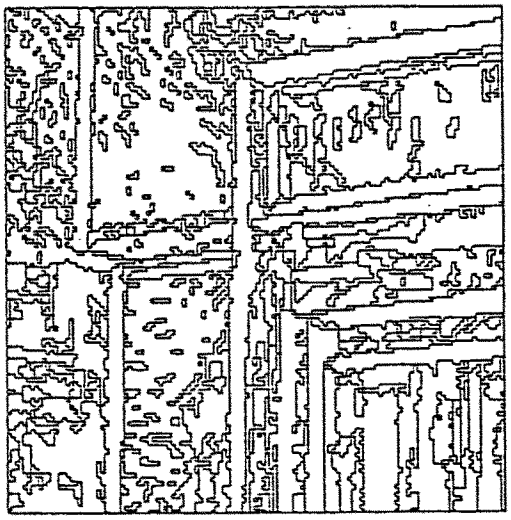


Fig.10 Extracted Gradient Region

Fig.12 Extracted Line Segments

6.Conclusion

In this paper the line segments are extracted from the gray level image by grouping of pixels using the direction of gradient vector, fitting curved and straight line, combining line segments. The method is applied for real indoor scene, and the line segments are successfully extracted.

It is difficult to extract boundary lines for a image where the intensity change is not so steep by using just differential operator which has been using. It is however possible by the method first proposed by J.B.Burns et.al., because this method is not based on the length of gradient vector but mainly on the direction of it.

Furthermore we formulate to fit straight and curved line segments to grouped region and how to combine them. However as we utilize only an arc for curved line, it is necessary to represent it by using more expressive curved line, for example ellipse or parabolic curve defined in differential geometry. Furthermore how to combine or integrate the line segments sophisticatedly is one of open problems.

References

[1] D.Marr, Vision, W.H.Freeman, pp.24-38 (1982)

[2] D.Marr and E.Hildreth, "Theory of edge detection," Proc. R.Soc. London , B207, pp.178-217 (1980)

[3] H.R.Wilson and J.R.Bergen, "A four mechanism model for spatial vision," Vision Res. , vol.19, pp.19-32. (1979)

[4] L.G.Roberts, "Machine perception of three-dimensional solids, in Optical Electro-optical processing of Information, " MIT Press, pp.159-197. (1965)

[5] R.O.Duda and P.E.Hart, Pattern Classification and Scene Analysis, Willey, pp.262-272 (1971)

[6] D.H.Ballard and C.M.Brown, "Computer Vision," Prentice-Hall, pp.79-185. (1982)

[7] S.Yokoi et. al., "A Method for extracting Feature Points and Line Figures from Grey Pictures," Systems Computers Controls, vol.6, pp.77-85. (1975)

[8] J.B.Burns and A.R.Hanson, "Extracting Straight Lines," IEEE Trans. Pattern Anal. Mach. Intell., Vol.PAMI-8, pp.425-455 (1986)

[9] K.Koffka, "Principles of Gestalt Psychology," Harcourt, Brace & World, New York p.110 (1935)