Edge tracing in a priori known direction

Andrzej Nowak, Andrzej Florek, Tomasz Piascik

Laboratory of Automation and Robotics Technical University of Poznan, Poland

Abstract. This paper presents a contour tracing algorithm based on a priori knowledge about searched edges. The algorithm is destined to trace edges having long fragments of approximately constant direction. This enables the implementation of one edge detector mask only in a given area and simplifies the thinning procedure. The way of searching for starting points is discussed as well as choosing and joining fragments of edges assuring the best correspondence between the found edge and the knowledge possessed. The algorithm shows good insensitivity to noise and local edge distortions.

1 Introduction

Edge detection and tracing is a crucial problem in the area of digital image processing. An edge can be defined as a boundary between two homogeneous areas of different luminance. Local luminance changes and edges corresponding to them are one of characteristic image features providing information necessary in the process of scene analysis and objects classification [BB1], [Pr1]. Most of contour extraction algorithms consist of the two basic steps: the edge detection (sometimes with thresholding) and the thinning and linking. They are efficient when applied to the image of nearly homogeneous objects differing significantly from the background (e.g. tools, industrial parts, writing etc.) if the image is not contaminated by noise. When the level of noise increases the obtained contours are often broken and deformed. That makes the process of interpretation and recognition more difficult. More sophisticated methods should be then implemented, e.g. incorporating a feedback path or a local edge enhancement [CS1]. However, all the universal edge tracing algorithms may still fail when implemented to noisy images. In these cases, the use of a priori knowledge about the edge to be traced significantly facilitates construction of an appropriate tracing algorithm.

2 Edge Tracing

The first step of the algorithm is the edge detection. It is made by convolving an image (or a fragment of it) with one mask which is the most sensitive to luminance changes in a chosen direction. The choice of this mask is based on our expectations about the searched edge direction (called here: the assumed edge direction). The obtained edges are thinned then. The edge tracing begins with finding a starting chain (edge fragment). Then the edge is traced, point after point. When a gap is encountered, a procedure for seeking and estimating all the chains passing close to the current boundary point is activated. The chains are estimated according to a criterion examining their usefulness for further tracing. The best chain, fulfiling also some threshold conditions, is accepted as a continuation of the broken boundary. This chain is then connected with the previously found boundary fragment and the tracing procedure continues. In the case when no chains are found (or when none of them fulfils the threshold conditions), searching for a new starting chain begins (it is led in the assumed edge direction.)

2.1 Edge Detection and Thinning

In the present algorithm, the edge direction is quantized into one of the eight directions. This is a common approach assuring good detection of edges in any direction. For detecting edges a set of eight masks, of the size of 3x3 pixels is used, as proposed in [Rol]. The edge direction (from 1 to 8) to which a given mask is the most sensitive, is called here a mask direction. For a chosen image part, one mask which direction is the closest to the assumed edge direction, is implemented. The edge detection is carried out by convolving the considered image fragment with the appropriate mask. The result of this convolution is the edge magnitude image, having "lines" where edges previously existed. Since edges in real images are usually blurred over some area, and after convolution with a 3x3 mask this blurring still increases, then resulting lines are at least a few pixels wide. The point of the maximum edge magnitude on an edge cross-section is assumed a boundary point. Because of blurring and the presence of noise a few local maxima can appear on this cross-section. It is difficult to estimate then if these maxima originate from noise or from a few edges passing close to one another. A simple solution to this problem is to assume a minimum distance between two edges. If a distance between neighbouring maxima is shorter than this minimum, then the bigger maximum is considered as an edge point. It should be aimed to take this minimum distance as short as possible, to prevent attenuating the "weaker" edge by a "stronger," neighbouring one. It was fixed in the implementation that a minimum distance between edges cannot be shorter than three pixels. It also ensures that the obtained chains will not be branched.

Thinning the previously detected edges is achieved in two passes. In the first one all the points of the image fragment are analyzed in rows, from left to right. For each analyzed point the following points (in a direction perpendicular to the assumed edge direction) are checked (see Fig. 1.). If the gradient magnitude of the analyzed point is smaller than the gradient magnitude of one of the two following points, then it is set to zero. This procedure is repeated in the second pass, while moving from right to left.

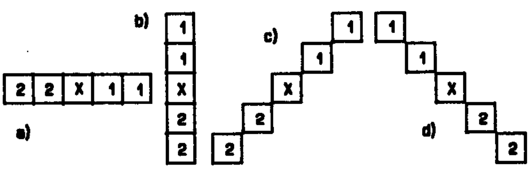

Fig. 1. Neighbourhoods of the analyzed point (x) checked during thinning, for the first (1) and the second (2) pass, for different mask directions

As a result of the thinning procedure fragments of edges (called chains) are obtained. Their direction is close to the direction of the used edge detector mask and they are one pixel wide (measuring in the thinning direction - perpendicularly to the mask direction).

2.2 Edge Tracing

The first step of the edge tracing is searching for a starting chain. This procedure is analogous to the one of finding the best chain in a window, described below. The difference is that the starting chain must fulfil much stronger conditions concerning its length and average gradient magnitude. Also the size of a search window is usually bigger in this case. Additional conditions, as a position of the edge in relation to some characteristic points of the image, derive from the possessed a priori knowledge and must be defined separately for each implementation.

After finding the starting chain, the edge tracing begins from its first point (starting point). For each, already found edge point, the next point is searched in the strictly defined neighbourhood. The choice of the analyzed neighbours depends on a direction of the edge detector mask and on a tracing direction (in accordance or in opposition to the mask direction - see Fig. 2.). If the gradient of any of these neighbours is different from zero, it is accepted as the next edge point. The tracing procedure is continued until a margin of the analyzed area or a break in the traced edge is reached. In the latter case searching for a new chain (which could be assumed as a continuation of the broken edge) is activated. The search is led in the area limited by a window which center is the last found edge point. The window has the square shape and its sides are parallel to the

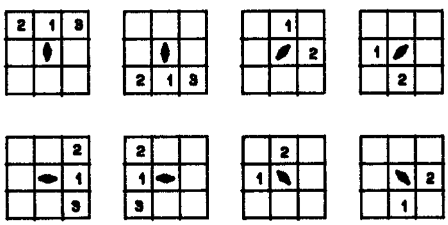

Fig. 2. Neighbours analyzed in searching for the next edge point according to the mask direction

image margins. Its size depends on possible local changes in the edge direction and on a distance from a possible "strong" edge.

In the window all the chains originating in it are searched (except for the already found boundary). All the window points are checked along lines perpendicular to the mask direction, starting from one of the corners. For each point, when its gradient magnitude

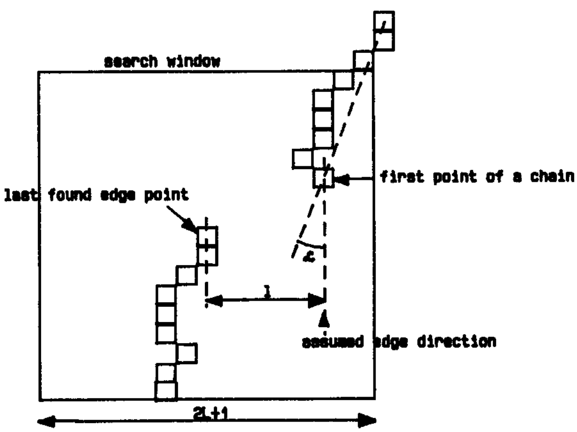

Fig. 3. The way of calculating parameters l and α when the mask of direction 1 or 5 was used

is higher than zero, all its neighbours from the previously checked line are analyzed. If their gradient equals zero then the following (in this window) chain number is assigned to this point and its coordinates are stored. Thanks to the use of one edge detector mask and described thinning procedure, it is not possible that chains beginning in two separate points will join in one chain. Thus, after having the whole window analyzed, the number of chains beginning in it and their starting coordinates are known. Afterwards their properties are analyzed from the point of view of their usefulness for tracing continuation of the broken edge. In order to attain this, each chain is traced from its (already known) starting point to the end (but not further than on a given maximum distance). The total gradient magnitude of all its points and its length (the number of pixels) are counted as well as the average gradient magnitude and deviation from the assumed direction. For counting this deviation only coordinates of the first and the last point of the chain are used. The best chain is chosen from among the chains satisfying specified threshold conditions (concerning their length and average gradient magnitude). The best chain is assumed the one maximizing a given criterion. An exemplifying criterion can be as follows:

$$Q = a_1 \frac{n}{n_{max}} + a_2 \frac{m}{m_{max}} + a_3 \left(1 - \frac{2h}{L}\right) + a_4 \left(1 - |tg\alpha|\right) \tag{1}$$

where:

a_i - not negative weight coefficients,
n - length of a chain in pixels,
m - average gradient magnitude,
l - deviation of the first point of the chain from the assumed direction (in pixels),
L - window size (its side is 2L+1 pixels),
$_{max}$ - maximum values in a window.

The way of calculating parameters l and α is shown in Fig. 3.

The chain maximizing the Q criterion is connected to the previously found edge fragment. In a case when no chains are found in the searching window or no chains fulfil specified threshold conditions, searching for a next starting point begins. The search is carried in the expected direction of the edge. A negative result of the search in the analyzed image part or reaching its margin ends the algorithm in this part of the image.

3 Experimental Results

The presented edge tracing algorithm was implemented to find the outlines of the outer fat layer of halved pigs carcasses. It enables finding the maximum width of this layer as well as other parameters necessary for meat classification. The resolution of analyzed images was 512x512 pixels with 256 grey levels. The images were additionally low-pass filtered (with a 3x3 mask). For edge detection the masks detecting vertical edges were used. The summed up result of implementing both masks is shown in Fig. 4.b. After having analyzed about 100 images the following parameter values were set:
- the minimum length of the best chain in a window was 10 pixels,
- the window size was 21 pixels (41 by 21 for a starting chain)
- the weight coefficients a_1 , a_3 , a_4 were established so that the maximum of each product in (1) was 1 (only a_2 was set bigger).

Exemplary results are shown in Fig. 4. Edges obtained after detection and thinning (Fig. 4.b) are broken in many points. This causes frequent searching for a chain which could be accepted as a continuation of a broken boundary. In the places of joining chains,

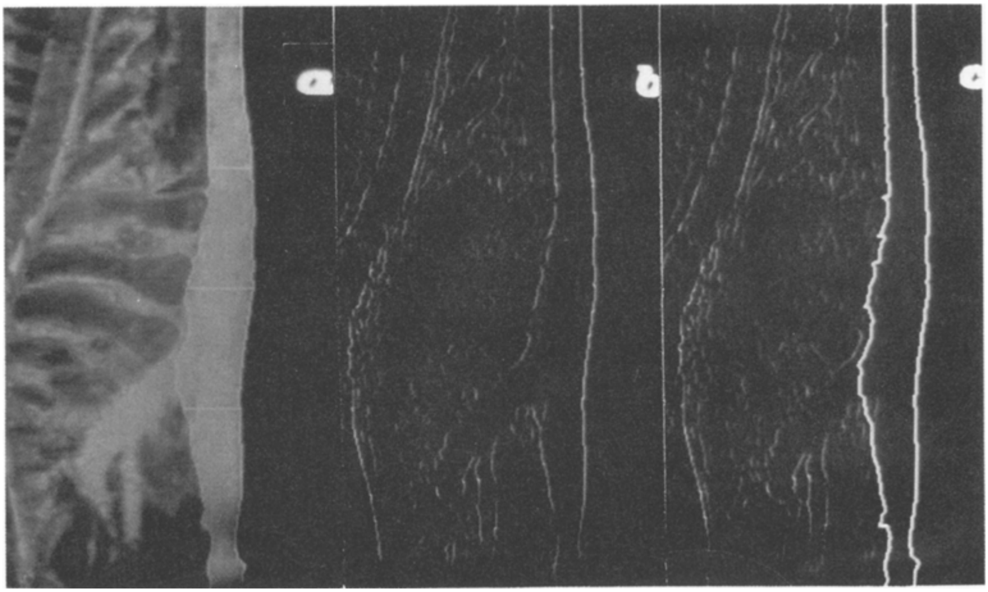

Fig. 4. The result of implementing the edge tracing algorithm to find the outlines of the outer fat layer of halved pig carcass: a) smoothed outlines shown on the original image, b) result of implementing masks detecting vertical edges, c)obtained outlines of the fat layer

local changes in the outline direction can appear (Fig. 4.c). These outlines, after smoothing, were put on a real image (see Fig. 4.a). On the basis of the obtained outlines characteristic parameters of the fat layer were calculated. (Areas of interest are marked with horizontal lines.)

4 Summary

A simple edge tracing algorithm was presented here. Applying universal edge tracing algorithms is unjustified when analyzing images about which some a priori knowledge is accessible. The computational complexity of these algorithms is usually high and their efficiency is low, especially for noisy images. The presented algorithm (based on a priori knowledge about the analyzed scene) is efficient even when applied to noisy images and distorted edges. However, for each implementation it requires setting values of all parameters as well as defining additional conditions (characteristic for a particular implementation) simplifying the tracing procedure.

References

[BB1] Ballard, D., Brown, C.: Computer Vision. Englewood Cliffs, NJ: Prentice-Hall, 1982
[Pr1] Pratt, W.: Digital Image Processing. New York: Wiley-Interscience, 1978
[CS1] Chen, B., Siy, P.: Forward/backward contour tracing with feedback. IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-9, no. 3, pp. 438–446, May 1987
[Ro1] Robinson, G.: Edge detection by compass gradient masks. Comput. Graphics Image Processing, vol. 6, pp. 482–492, 1977

This article was processed using the LaTeX macro package with ECCV92 style