MVA'98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

Contour tracer for a fast and precise edge-line extraction

Christian Robl[1]
Laboratory for Process Control and Real-Time Systems
Prof. Dr-Ing G. Färber
Technische Universität München

Abstract

Feature extraction algorithms delivering accurate results are computational intensive and cannot keep up with the demanded processing times on single processor systems in automated micropart assembly tasks. For this case an improved contour tracer for real-time edge-line extraction, using adaptive local thresholds, is introduced. In combination with an effective start point search, that concentrates on the strongest local contours, and a two stage robust corner detection a fast and precise edge-line extraction comes possible.

1 Introduction

Edge-line extraction, an essential first step in image interpretation, is, in general, computational expensive. Accuracy and processing time are competitive demands. Classical approaches like LoG [1] and Canny [2] extract precisely edges with a huge computational effort. They have the disadvantage of long processing times per image or the need of an expensive, specialized, mostly massively parallel hardware. In these approaches, big convolution matrices (up to 32x32) or complex algorithms are applied to all pixels of an image. In opposite, contour tracing algorithms are very fast, mainly examining pixels lying on a contour. The contour tracer proposed in [3], [4] uses an average 85ms of processing time on a single processor DEC alpha 3000/600 at 175MHz. The accuracy of this algorithm is better than 1 pixel for straight contours, but, due to its properties, the displacements at contour discontinuities (e.g. corner) are quite larger.
In this paper, we present an improved approach of the above mentioned contour tracer, increasing the accuracy and stability without extending the processing time. We describe only changes made to the original approach, the additional other features of this contour tracer implementation, like the real-time capability, the correction of the image distortions, the application dependent change of the processing order and the two stage edge-line melting, are desribed in [3] [4].
After describing the basic contour tracing algorithm in section 2, we introduce our effective start point search in section 3. Contour tracing using adaptive local thresholds is described in section 4 and the improved detection of corners is presented in section 5. The last section shows some experimental results with two different applications: exploration in an indoor environment and localization of a surface mounted device.

2 Contour tracer

This contour tracer is based on an electro-dynamical analogon, in which an electron is driven along a contour by forces caused by electric (Coulomb-Force) and magnetic (Lorenz-Force) fields. Each pixel forms a horizontal and vertical dipole, whose charges are derived from the gray scale values of the regarded pixel and of each adjacent pixel. The resulting electric field affecting the electron is composed by the contributions of the dipoles in the vicinity. The magnetic field is formed by a current flow in the conductor between the inexhaustible charges of each dipole.
However, this analogon is only necessary to derive the operators, three 5x5 matrices, that are similar to the gradient in x and y direction (Kirsch-Matrices [6]) and the second derivative (Marr-Hildreth-Matrix [7]). The size 5x5 has been found to be a good compromise between the precision of the result and the required workload. The electric field causes the electron to follow the contour, but cannot hold it on the ridge of the contour. Therefore the magnetic field is necessary to stabilize the electron's motion and avoids too strong divergences when the contour is strongly curved.
After finding a suitable starting point, the next contour spot can be determined by applying these operators to the current contour spot according to the electron movement equations in [3] . The proposed search method for finding starting points is grid-based and uses a global threshold.

3 Effective start point search

We developed a new starting point search scheme, which is also grid-based, but finds always the strongest and most important contours. The quality of the extracted edge-lines is also better when a

[1] Address: Arcisstr. 21, 80290 Munich, Germany.
E-Mail: robl@lpr.ei.tum.de, Phone: +49-89-289-23578

good starting point was found. Equivalent to the old search scheme, the image is divided into several small commensurate windows. Each window is scanned horizontally and vertically through the center. A new step that we use is to first scan the whole window to determine the averaged electric field for this window as a measure for a local threshold. Then all maxima that are by *eps* percent higher than the mean value E_{mean} are used as starting points. The computed threshold has to be greater than the given lower limit n, the image noise.

$$th_E = E_{mean} \cdot (1 + \frac{eps}{100}) + n;$$

$$E_{mean} = \frac{\sum_{i=1}^{Size_x} E_h + \sum_{i=1}^{Size_y} E_v}{Size_x + Size_y};$$

Figure 1 shows the horizontal electric field profile of a search window and the computed thresholds and starting points. See Figure 2 for the referring gray scale image.

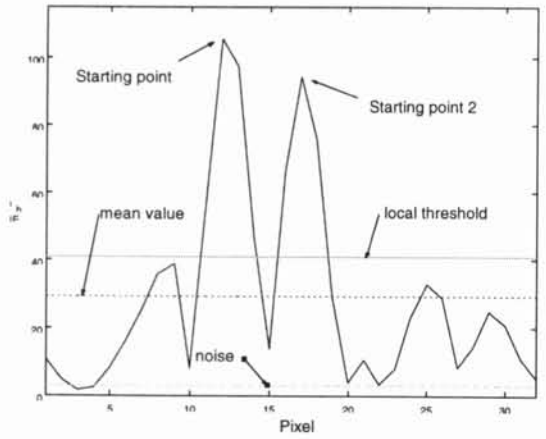


Figure 1: horizontal electric field profile

In contrast to the old algorithm, the electric field in this search scheme is estimated with $E=|E_h|$ and $E=|E_v|$ for horizontal and vertical scanning direction, respectively, instead of the exact computation $E = \sqrt{E_h^2 + E_v^2}$. Additionally the horizontal and vertical electric fields are obtained using the middle horizontal row and middle vertical coloumn of the referring matrix, respectively, instead of applying the whole 5x5 matrix. This saves 45 time consuming multiplication steps per iteration while at the same time obtaining a good estimate for the real absolute value.

4 Adaptive local thresholds

The local threshold principle is also used for contour tracing in the improved approach. Otherwise contours with decreasing contrasts to the neighborhood, caused by illumination effects, cannot be extracted completely (see Figure 2). This would result in too short edge-lines that are not useful for higher level interpretation.

Since all starting points found in a window are processed before the next window is scanned (to ensure real-time capability), the local thresholds for the adjacent windows may be unknown, although a found contour is traced to its end. The mean value of the electric field of the last 16 contour spots is a measure for the threshold similar to the start point search. The next contour spot is determined as long as its electric field value is higher than the actual threshold . The adaptive threshold is computed as follows:

$$th_i = avg_{i-1} \cdot (1 + \frac{eps}{100}) + n;$$

$$avg_i = avg_{i-1} + \frac{E_i - E_{i-16}}{16};$$

with n is the lower limit of the threshold and *eps* the percentage the threshold has to be greater than the mean value; *avg* is the mean value of the last 16 contour spots' electric field.

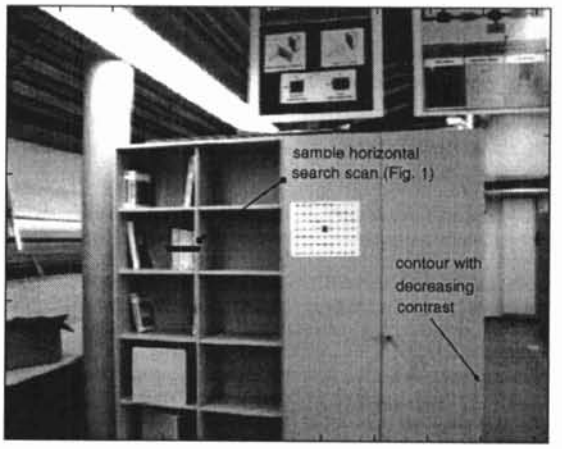


Figure 2 : Sample Image

5 Improved corner detection

The biggest disadvantage of the original approach are the large displacements at contour discontinuities such as corners. Because of the 'electron's' inertia the contour tracer overshoots.

In our new approach we avoid this property with a two stage algorithm for corner detection. The first stage is the origin corner detection with an outlier correction added. A corner is detected if the pixel deviation of the new contour spot to a regression line is greater than the given threshold (see Figure **3**a). The orientation of this regression line is obtained, computing the mean value α_{mean} of the orientations of the straight lines between the starting point and all contour spots lying between starting point and the current contour spot. The regression line starts at the starting point. The pixel deviation ε is computed as follows:

$$\varepsilon = step \cdot i \cdot \tan \Delta\alpha; \text{ with } \Delta\alpha = \alpha_{mean} - \alpha_{reg};$$

with *step* is the step width of the contour tracer and i the i^{th} contour spot; α_{mean} and α_i is the orientation of the regression line and the line between starting point and current contour spot, respectively.
In order to suppress outliers, this condition has to be met for two succeeding contour spots (see Figure 3b) before stage 2 can be entered.

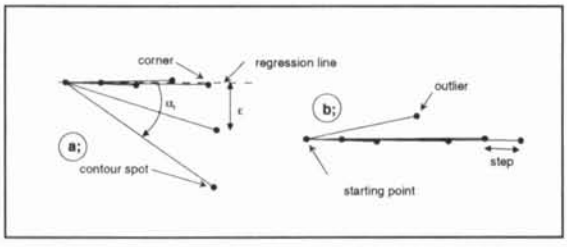


Figure 3 : Corner detection step 1

The second stage resets the contour tracer to the newly found corner point, the starting point for the next edge-line. The electron overshoots, because at a 90° corner the direction of the electric field (similar to the gradient) is 45° and the direction of the electron's velocity is 0°. Since the orientation of the magnetic field is perpendicular to the orientation of the velocity, the ratio between electric and magnetic field for the next step is changed (see Figure 4). This method amplifies the dynamic behavior and therefore the electron's velocity must not be zero. Although the electron's inertia causes a force in the old movement direction, the dynamic behavior is stronger.

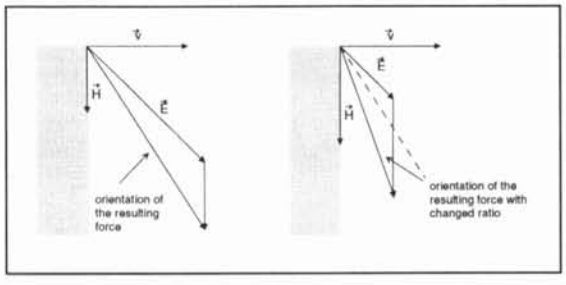


Figure 4 : Electric and magnetic fields at a corner

In order to stay as near as possible to the contour, the next step of the contour tracer is split into two 'half-steps' with the half step width. These two steps are combined and handled as one step in the succeeding regression analysis, further reducing the overshooting of the contour tracer.

6 Experimental results

Our current contour tracer implementation, is not restricted to fast but inaccurate obstacle detection for autonomous mobile robots, but it will also be used for localization, exploration and object and state identification tasks of indoor environments with an autonomous mobile robot [4]. Figure 6 shows the results of the contour tracer for a typical indoor scene. The extracted contour spots are marked with stars and the extracted edge-lines can be seen as dark gray lines. The zoom of the upper left corner of the depicted door shows the good behavior of the contour tracer at corner points.
The possible accuracy with a maximum deviation of 0.8 pixels at discontinuities makes it possible to use the presented improved contour tracer in microparts assembly tasks [5]. Figure 6 shows a mounting sequence of a fine pitch surface mounted device (a;) with an industrial robot. The shown extracted edge-lines (b;) are used for a precise localization (c;) in the range of about 50 microns.
The average processing time for 768x558 gray level images including distortion correction, segmentation and edge-line melting is about 50ms on a single processor DEC AlphaStation 500 at 400 MHz. This is about 30 times faster than the Canny extraction scheme [2], delivering comparable results.
Similar to the original approach the processing order of parts of the image can be changed depending on the application. Therefore real-time processing of images is also possible by aborting the extraction to meet the deadline, but obtaining a suboptimal result.

Acknowledgement

The work presented in this paper was supported by the *Deutsche Forschungsgemeinschaft (DFG)* as a part of the research project SFB331 and by the *Bayerische Forschungsstiftung* as a part of the research project FORMIKROSYS.

References

[1] R. M. Haralick: 'Digital step edges from zero crossings of second directional derivatives', IEEE Trans. PAMI-6, Nr1, 1984.
[2] J. Canny, 'A Computational Approach to Edge Detection' IEEE Trans. PAMI-8, Nr 4, 1986.
[3] G. Magin & C. Robl, 'A Single Processor Real-time Edge-Line Extraction System for Feature Tracking', IAPR MVA '96, pp 422-525, 1996, Tokyo, Japan.
[4] D. Burschka et al., 'Interaction of Perception and Control for Indoor Exploration', IEEE ICRA '98, Leuven, Belgium.
[5] C. Robl et al., 'Micro Positioning System with 3dof for a Dynamic Compensation of Standard Robots', IEEE IROS '97, Grenoble, France.
[6] D. H. Ballard, C. M. Brown, 'Computer Vision', Prentice-Hall, Englewood Cliffs, 1982.
[7] O. Faugeras, 'Three-Dimensional Computer Vision. A geometric Viewpoint', MIT Press, Cambridge, 1993.

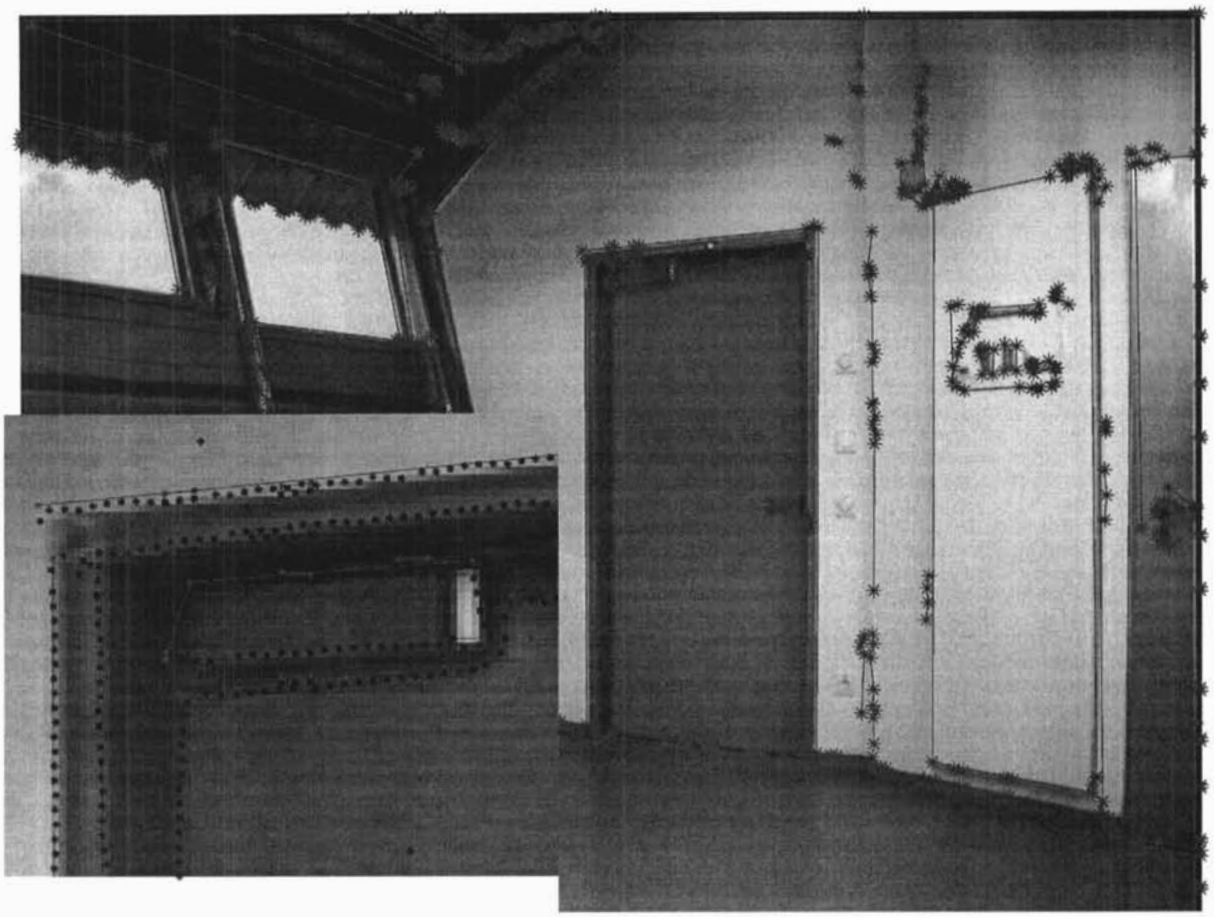
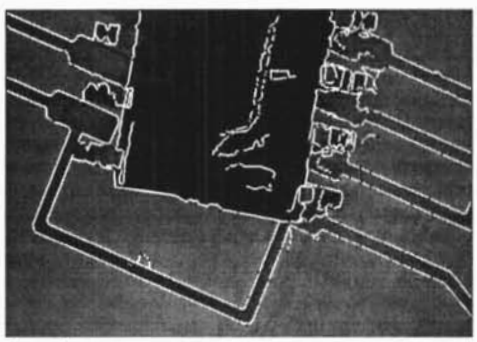
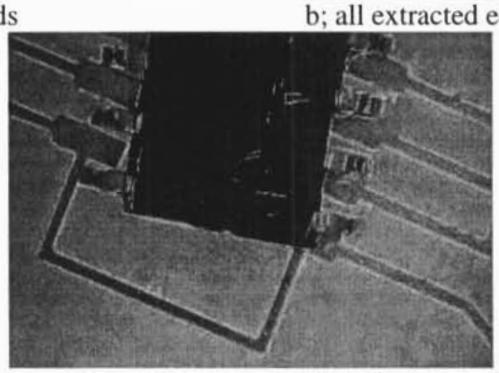


Figure 5 : application - edge_lines for exploration of indoor environments

a; SMD and target pads



b; all extracted edge-lines



c; edge-lines used for determining the distance between pins and pads

Figure 6 : application - mounting of fine pitch surface mounted devices