STRAIGHT LINE DETECTION FOR AUTOMATED MEASUREMENT IN IMAGE SCENES

E. Reetz*, A. Schlegel, P. Werner and G. Linß
*Department of Quality Assurance and Industrial Image Processing
Ilmenau University of Technology, Ilmenau, 98693 / Thuringia, Germany
Email: Edgar.Reetz@googlemail.com

Abstract: When dealing with point cloud data retrieved from optical coordinate measurement machines, a spline filter scale space method as proposed in [1] is applied to contour decomposition problems for automated measurement of geometric primitives/features. The object contour will be split into a more simple structure, using the spline filter scale space to detect corner regions along the contour. This article explains an approach to increase the segmentation results for straight line segments, based on point cloud data only.

Keywords: geometric primitives; contour decomposition; straight line extraction; countour smoothing; spline filter;

I. INTRODUCTION

Searchline based edge detection methods at subpixel-scale precision (see [2]) is a common approach in 2D high-precision measurement using optical measurement devices such as optical coordinate measurement machines. Usually the tester needs to set up the inspection plan manually for the device unter test. This applies especially when there is no CAD data available. Therefore the inspection plan can not be derived from the design data for the device. Thus the tester needs to select the correct ROI[1] setup for each feature to be measured in the image scene. Afterwards each region of interest needs to be put at the correct location, in a way that all searchlines are set perpendicular to the object's contour. Considering a complex inspection plan, this could be a very time consuming procedure. Another disadvantage is the user influence on the measurement results, caused by the manual setup for each region of interest. Improving the objectivity of the results and increasing the level of automation during the user interaction phase, the contour decomposition task should be automated. Therefore edge detection algorithms including subpixeling were used to retrieve contour data from the image scene. Then the contour decomposition could be executed on plain point cloud data. For the problem discussed here, the image preprocessing steps were already completed. Therefore the method's input data is point cloud data only, at subpixel-scaled precision. Figure 1 displays the extracted point cloud of an image showing the contour of a milling tool, containing 1457 contour points. To obtain reliable measurement results, the contour needs to be split at its corner points and the segments between will be approximated by shape primitives.

[1]ROI: region of interest (including the setup for searchline orientation)

Fig. 1. Point cloud representing object contour at subpixel-scaled precision, containing 1457 contour points.

II. STATE OF THE ART

Dividing point clouds representing an object's contour into different homogenous regions of similar contour curvature is a common approach in computer vision. The criterion used to split point clouds into line segments and arcs is:

$$\kappa(t) \approx const. \wedge \kappa(t) \approx 0 \quad \text{for line segments} \quad (1)$$

$$\kappa(t) \approx const. \wedge \kappa(t) \not\approx 0 \quad \text{for arcs} \quad (2)$$

The contour point cloud is retrieved using enhanced edge detection methods at subpixel scale (see [2]). Computing curvature values $\kappa(t)$ along the contour based on the interpolated subpixel scaled point cloud data results in a noisy signal, as shown in figure 2. Methods applied to further smoothen the curvature plot mostly use low pass filtering techniques and $\frac{\pi}{4}$-clockwise smoothing, based on the pixelgrid. Prior set threshold values for contour curvature amplitudes sometimes are used as well to split the curve into candidate regions for straigth line segments and arcs. Often it is difficult to find the correct parameter setup, which

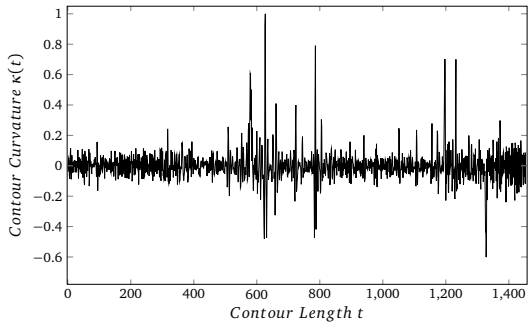

Fig. 2. Computed contour curvature $\kappa(t)$ for the object contour shown in fig. 1

will deliver reliable results for different types of image data, which is the major disadvantage on using parameter based methods for contour split and merge approaches.

III. The Problem Setting

For each object contour there will be a low-pass filter setup for which the best straight line discrimination could be achieved. The underlying point cloud is deviated by the subpixel interpolation method, lens correction algorithms and the object's roughness and waviness. Therefore the threshold based low-pass filter setup will not return similar results for the same object but with higher roughness and waviness of the profile. The initial segmentation results will be highly fragmented or many false positive detections will occur. Different setups for using Savitzky-Golay[2] smoothing filters are shown in [1]. The overall problem to determine the correct filter setup for each situation still remains.

IV. The Novel Approach

For breaking the contour into more simple primitives, spline surface profile filters as in ISO/TS 16610-22 (also see [4]) are used. The filter kernel smoothens the contour point cloud by minimizing the contour its bending energy. Using spline filters with an scale space approach for contour smoothing as proposed in [1], returns different smoothing scales. Figure 3 displays results for the milling tool contour in figure 1, smoothened at different scales and drawn on top of each other. To display the detected corner regions, the binary encoding of the smoothened contour curvature is indicated by dark red color, for regions, where:

$$\kappa(t) > \frac{1}{n-1}\sum_{j=1}^{n-1}\kappa_j. \tag{3}$$

The corner regions displayed in red color increase

[2]see [3] for Savitzky-Golay smoothing filters

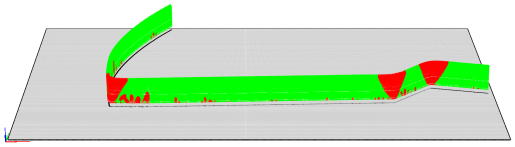

Fig. 3. Spline filter scale space indicating feature points of higher curvature than the profiles average in dark red color.

in width when rising the scale while the contour irregularities caused by roughness and waviness disappear. Regions at smaller curvature are displayed in light green color. Figure 3 depicts that contour irregularities are still present at lower scales. The corner regions remain as major feature points, throughout all scales, therefore they will be used as contour breakpoints. Once the major contour breakpoints are detected, there is two different types of contour segments to be processed. The point subset of the original point cloud representing the corner regions and the contour points between them — under the assumption that there is at least one corner point in the original point cloud detected by the spline filter scale space. Since the corner points show high curvature at every scale, in contrast to non-corner areas, the minimal width of such a corner region is of interest, due to the fact that the corner region width will extend by rising the scale. Constraints to reduce the amount of detected corner regions could be used, e.g. limits for the minimum length of corner regions. For the example given, two corners detected are displayed in figure 4. Vertical

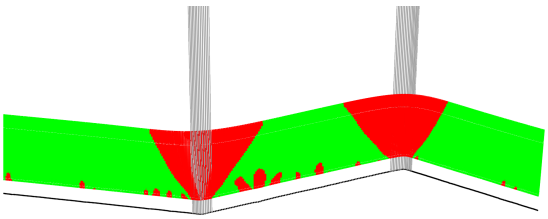

Fig. 4. Profile feature points at minimum width detected at every scale indicated by vertical lines. The original profile is drawn using black dotted line style.

bars intersecting each contour point display the minimum width of interest for the corner region detected, while the curvature scale space colored in red displays possible corner regions at every scale. The subset of contour points located between two corner regions, between beginning and end of the contour or between a start/end point of the contour and a corner region is checked for similarities with arcs or straight line segments in a first coarse step. An approach proposed by [5], comparing the ratio of the distance accumulated along each contour point of interest with the direct distance between the beginning and the end of the segment of interest

returns reliable results, only when a maximum size for the circular segment is known, or when roughness and waviness of the segment of interest is not big enough to cause false positive detections. The method used here proposes the computation of the approximating element and comparing the width of their supporting points zone. The zone of supporting points result from each approximation's shape deviation regarding the contour point cloud. In figure 5 on the left a regression line including its zone of supporting points is drawn. Since the regression circle for same region returns a smaller zone-width, the circular segment is the preferred choice for the approximation element. The remaining segments in figure 5 are approximated by straight line segments. The approximation through-

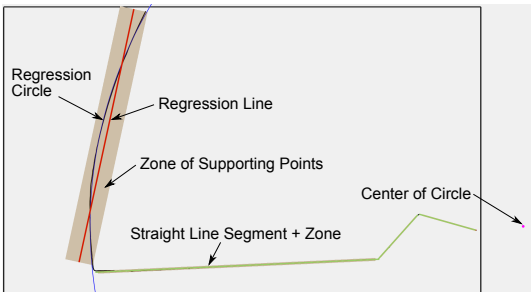

Fig. 5. The initial approximating elements computed for the object contour shown in fig. 1. Each approximation's zone of initial supporting points is indicated as darkened zone surrounding the element.

out all segment points returns an element, computed by two dimensional total least squares regression methods. The approximation's deviation among all segment points, perpendicular to the approximation its orientation, spans the zone of supporting points, or tolerance zone. In figure 5 the tolerance zones are displayed for the initial approximation candidate.
Since the corner regions were determined throughout all filter scales, these regions possibly span over to many points. The reduction of corner regions is achieved by growing their neighbouring elements/approximations. For straight line segements, this "region growing"-like approach, based on the supporting points zone-width, along the approximation's principal direction is executed for all approximations of the same type in the image scene. By now the straight line segments should extend in both possible directions, towards the corner region, but not further than until breaking their tolerance zone. Since most of the straight line segments were set conservatively (a little to short between corner regions), they will extend a little. If the contour roughness/waviness is higher, the zone of supporting points for the approximation will be of higher width as well. Therefore the straight line segments sometimes will be extended too long, since the subsequent contour points have to break the tolerance zone first to stop the region growing process. Due to irregularities along the contour, in some cases the region growing process stops to early. That means, that the line segment will not be extended to its maximum length into the corner region. If the remaining points within the corner region count more than three, two strategies to fill the gap are possible. By iteratively computing the new position of regression lines, considering their tolerance zone-width, both line segments on each side of the corner region aim to extend towards an virtual intersection point (see figure 6). If the straight line segments fail filling the missing gap in the corner region, an arc is used as an alternative approximation (see figure 6). Finally the succesful extracted segments could be used for direct automated measurement tasks as proposed by Werner in [6] or to automatically generate inspections plans, where searchlines need to be set up perpendicular to the object's contour direction. Therefore a new workflow model providing reduced user interaction for optical measurement devices (e.g. coordinate measuring machines) becomes applicable.

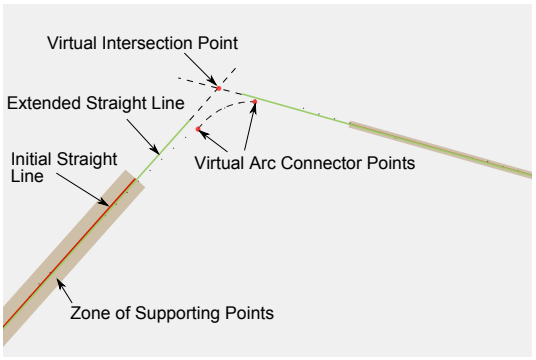

Fig. 6. Corner region in detail: The initial straight line segments were extended until breaking the initial zone of supporting points. The corner region left still contains five points. Possibilities closing the contour gap: a) Extending line segments until intersection point. b) Bridge the gap using an arc.

V. Conclusion

The method presented provides an adaptive approach for segmenting contour point clouds based on spline filter scale space contour smoothing. Corner regions detected throughout all scales were used for coarse contour decomposition. Based on the approximation's shape deviation, the approriate approximation was choosen, while a region growing method was used to extend the straight line segments closely to the corner points. As a result the point cloud contour decomposition could be used to generate inspection plans for 2D-optical measurement devices at reduced user interaction/influence on the measurement results. The inspection plan generated is reusable and could be provided without CAD data. Therefore this approach provides fast and automated generation

of inspection plans reducing the user influenced workflow during the measurement process.

ACKNOWLEDGMENT

The support of the TMBWK ProExzellenz initiative, Graduate School on Image Processing and Image Interpretation at Ilmenau University of Technology is greatfully acknowledged.

REFERENCES

[1] E. Reetz et al., "Application of spline surface profile filters to subpixel contour decomposition problems," in *Advanced Mathematical and Computational Tools in Metrology and Testing*, ser. Series on Advances in Mathematics for Applied Sciences, vol. 84, F. Pavese et al., Ed., vol. 9. Singapore: World Scientific, 2012, pp. 334–342.

[2] O. Kühn, "Ein beitrag zur hochauflösenden geometriemessung mit ccd-zeilensensoren," Dissertation, Technische Universität Ilmenau, Ilmenau, Mai 1997.

[3] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[4] M. Krystek, "Discrete linear profile filters," in *X. International Colloquium on Surfaces Chemnitz (Germany)*, M. Dietsch and H. Trumpold, Eds. Aachen: Shaker Verlag, 2000, pp. 145–152.

[5] T. M. Sezgin, T. Stahovich, and R. Davis, "Sketch based interfaces: early processing for sketch understanding," in *ACM SIGGRAPH 2006 Courses*, ser. SIGGRAPH '06. New York, NY, USA: ACM, 2006.

[6] P. Werner et al., "A novel method for an automated analysis of a measurement scene," in *Advanced Mathematical and Computational Tools in Metrology and Testing*, ser. Series on Advances in Mathematics for Applied Sciences vol. 84, F. Pavese et al., Ed., vol. 9. Singapore: World Scientific, 2012, pp. 334–342.