Line segmentation for interferograms of continuously deforming objects

Torsten Merz Dietrich Paulus Heinrich Niemann

Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung
Martensstr. 3, 91058 Erlangen, Germany

merz@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de

ABSTRACT

A new robust and fast method for non-interactive line segmentation of interferograms is proposed. Fringe contours are represented as a set of polygons using a new technique for contour approximation. The method has been developed for application in interferometry with continuously deforming objects. Its application to real-time holographic interferometry in nondestructive testing is shown.

Keywords: line segmentation, contour approximation, fringe analysis, holographic interferometry, speckle noise, nondestructive testing

1. INTRODUCTION

Line segmentation is often required for phase reconstruction in quantitative interferogram analysis as well as for feature extraction in qualitative interferogram analysis. Qualitative interferogram analysis is applied in the field of nondestructive testing where objects are tested with regard to surface and internal flaws. Flaws appear on interferograms as characteristic fringe irregularities, for instance "bull-eye"-fringes, distorted fringes, local compressed fringes, cutted and displaced fringes.[1] These irregularities can be represented without loss of information by fringe contour lines.

Existing segmentation methods in holographic or speckle interferometry,[3] like e.g. skeletonization or fringe tracking techniques, are either slow or they fail due to high speckle noise and varying background illumination. The proposed method overcomes this problem by analyzing the local intensity change during object deformation. Our approach is based on an article by J.Wang and I.Grant who proposed a phase-mapping method for ESPI using continuously deforming objects[4] which we extend to real-time holographic interferometry.

2. IMAGE FORMATION

In holographic interferometry the intensity $I(x, y)$ of an ideal interferogram can be described by the following expression[2]:

$$I(x, y) = I_0(x, y)\,[1 + V(x, y)\,\cos\delta(x, y)]\,R_s(x, y) \tag{1}$$

where $I_0(x, y)$ describes the local average intensity, $V(x, y)$ the fringe visibility, and $R_s(x, y)$ the multiplicative influence of speckle noise. The argument of the cosine function $\delta(x, y)$ is the phase difference which results from the displacement of the surface between both object states to be compared.

In practice additional disturbances are electronic noise due to recording with a CCD camera and non-uniform background intensity. Electronic noise can be reduced by time averaging. Its influence is relatively small, so we neglect it in the following. We assume that background intensity is not time dependent.

Local average intensity, fringe visibility, and speckle noise do not change considerably during object deformation if the optical system is kept stable and the displacement of the surface with regard to the initial object state is not too big. The latter should be considered anyhow as the fringe visibility vanishes if the displacement of speckle pattern is bigger than speckle size.[2] Thus light intensity $I(x,y)$ is a function of $\delta(x,y)$ only. Considering different deformation states during time, the intensity can be described by the following simplified expression (this is also valid for speckle interferometry[4]):

$$I(x,y,t) = I_d(x,y) + I_e(x,y)\,\cos\delta(x,y,t) \tag{2}$$

3. IMAGE ANALYSIS

3.1. Calculation of cosine map

The fringe pattern is described by the cosine term of Eq. 2, the fringe contour lines are all lines with equal phase difference $\delta = 90^o$. In order to find these lines it is necessary to calculate the cosine map of an interferogram, which is given as

$$\cos\delta(x,y,t) = \frac{2I(x,y,t) - I_{max}(x,y) - I_{min}(x,y)}{I_{max}(x,y) - I_{min}(x,y)}, \tag{3}$$

where $I_{max}(x,y) = I_d(x,y)+I_e(x,y)$ and $I_{min}(x,y) = I_d(x,y)-I_e(x,y)$ have to be determined for each individual pixel (x,y) in the CCD array. We assume that the phase difference during object deformation will be at least 2π, so that the intensity of each pixel will range from $I_{max}(x,y)$ to $I_{min}(x,y)$. A real-time algorithm for obtaining $I_{max}(x,y)$ and $I_{min}(x,y)$ in an image sequence using a special image processing board is shown in reference.[4] After having found $I_{max}(x,y)$ and $I_{min}(x,y)$ it is possible to calculate the cosine map $\cos\delta(x,y,t)$ for each recorded interferogram. Thereafter the following threshold function is applied (binarisation of cosine map):

$$c(x,y,t) = \begin{cases} 0 & : \quad cos(x,y,t) \le 0 \\ 1 & : \quad cos(x,y,t) > 0 \end{cases} \tag{4}$$

It was found that it is useful to improve the map with a binary "spin"-filter in order to close small gaps.[5]

In comparison to segmentation methods based on single interferograms this method reduces speckle noise considerably and it allows to mark regions with uncertain fringe information which are characterized by pixels having a lower contrast $I_{max}(x,y) - I_{min}(x,y)$ than a given threshold T_c.

3.2. Contour approximation

Contour approximation is done for data reduction and noise reduction. It is useful especially as a basis for feature extraction. In the following we consider a single binary cosine map $c(x,y,t_k)$ at the time $t = t_k$. The fringe contour lines are given as the border lines between regions with value 1 and value 0 in the binary map. In a contour following step these lines are traced by searching the image line by line until a point with a change from value 0 to value 1 or vice versa is found. Then the contour is traced considering the eight neighbours of each new found contour point until the starting-point or an image border is reached. In case of reaching an image border the contour is traced from the starting-point in opposite direction and the two parts are combined. For feature extraction it is useful to save the contour points in such a way that the tracing direction is always the same, so it

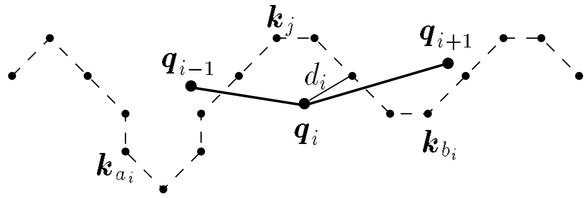


Figure 1: Contour approximation

is later possible to say at which side of the contour the fringe maximum lies. Each contour is saved as a sequence $[\boldsymbol{k}_i]$ of absolute point coordinates. Very short sequences which can result from distorted fringes will be removed.

After having found all contour lines, each contour line is approximated by a polygon. A polygon is represented as a sequence of vertices $[\boldsymbol{q}_i]$. We propose to define each vertex $\boldsymbol{q}_i$ as the center of gravity of a subsequence of contour points:

$$\boldsymbol{q}_i = \frac{1}{s}\sum_{j=a_i}^{b_i} \boldsymbol{k}_j \qquad a_i = n_i - \frac{s}{2} \quad b_i = n_i + \frac{s}{2} \tag{5}$$

where s determines the smoothing size and n_i the length of each polygon segment. For feature extraction it is useful to increase n_i in that way that all segments are about the same length. The smoothing size depends on the noise level of the contour line and on its shape. If s is chosen too small the resulting polygon will be noisy, is it chosen too big it may be possible that parts of a contour with high curvature will be lost.

Fortunately most fringes have the nice property to have a smooth contour, i.e. the curvature of a contour will not change rapidly and does not exceed a certain level. The curvature c_B at a contour point $B = (0,0)$ of a discrete contour can be defined as the reciprocal of the radius r_{ABC} of the circle given by the contour point and its left and right neighbour point $A = (x_a, y_a)$ and $C = (x_c, y_c)$:

$$r_{ABC} = \frac{\sqrt{x_a^2 + y_a^2}\,\sqrt{x_c^2 + y_c^2}\,\sqrt{(x_c - x_a)^2 + (y_c - y_a)^2}}{2\,|x_a\,y_c - x_c\,y_a|} \tag{6}$$

$$c_B = \frac{1}{r_{ABC}} \tag{7}$$

This is used to determine the curvature c_i at each vertex of the approximated contour.

A possibility to verify whether the smoothing size is too big, is to consider the minimal Euclidean distance d_i of a vertex $\boldsymbol{q}_i$ from the corresponding subsequence of contour points $\boldsymbol{k}_j$:

$$d_i = \min\{\|\boldsymbol{q}_i - \boldsymbol{k}_j\|\} \quad j \in [a_i, b_i] \tag{8}$$

It was found that four fixed smoothing sizes are enough to approximate all kind of fringe contours in our application. Each contour is approximated using the four sizes. Then the number of inadmissible vertices with a curvature $c_i > c_{max}$ and a contour line distance $d_i > d_{max}$ are counted for each size. Finally the contour approximation with the lowest number of inadmissible vertices is chosen.

figure	contour following	contour approximation	total time
2	0.7s	3.0s	3.7s
3	1.0s	4.0s	5.0s
4a	0.8s	4.0s	4.8s
4b	1.0s	6.0s	7.0s

Table 1: Computation times for line segmentation of cosine maps on a HP 715/50 66MHz. See text for computation time of cosine maps.

4. EXPERIMENTAL RESULTS

In our experiments we use a 200mm×100mm×20mm acrylic glass plate and an aluminium sheet which are deformed by tension or heating. For demonstration of the described method in nondestructive testing with real-time holographic interferometry, typical material faults are added to the test objects. The interferograms are recorded with a CCD camera with a resolution of 512×512 pixels and 256 grey levels, the computation is done on a HP 715/50 workstation.

It was found that the object shape can be changed continuously by mechanical as well as thermal loading and 10 interferograms which are recorded during changing load are sufficient for calculation of the cosine map. In reference[4] it is shown that the computation of the cosine map can be done in video real-time (i.e. 250ms for 10 images) using a DT2862/2861 image processing board. On our workstation we need 36s seconds mainly because the images have to be loaded from hard disk.

For contour approximation we use a polygon segment length of three to four. The four fixed smoothing sizes are $s \in \{10, 20, 50, 100\}$, the maximum curvature is $c_{max} = 0.2$, the maximum contour line distance is $d_{max} = 3.0$ and the low contrast threshold is $T_c = 10$.

An example of an image sequence of the acrylic glass plate deformed by tension is presented in Fig.2. It shows four of ten interferograms with corresponding cosine maps and segmented fringe contours. Fig.3 shows the results of the proposed method using interferograms of very low quality. Finally two examples of test objects with material faults are given in Fig. 4.

In Table 1 the computation times for the given examples are listed. The computation times depend on the complexity of the fringe patterns and on the noise level of the fringe contours. Note that these times are without the computation times for the cosine maps. The total times are still above video real-time at least using this hardware, but in many applications it is sufficient to do a fast off-line segmentation.

5. CONCLUSION

The described method enables fast line segmentation of interferograms being robust towards speckle noise, background illumination and uncertain fringe information. It is required that the optical system is stable, speckle noise will not change considerably and the phase difference will be at least 2π during object deformation.

Further investigations are directed to feature extraction for automatic fault detection and comparison of interferograms.

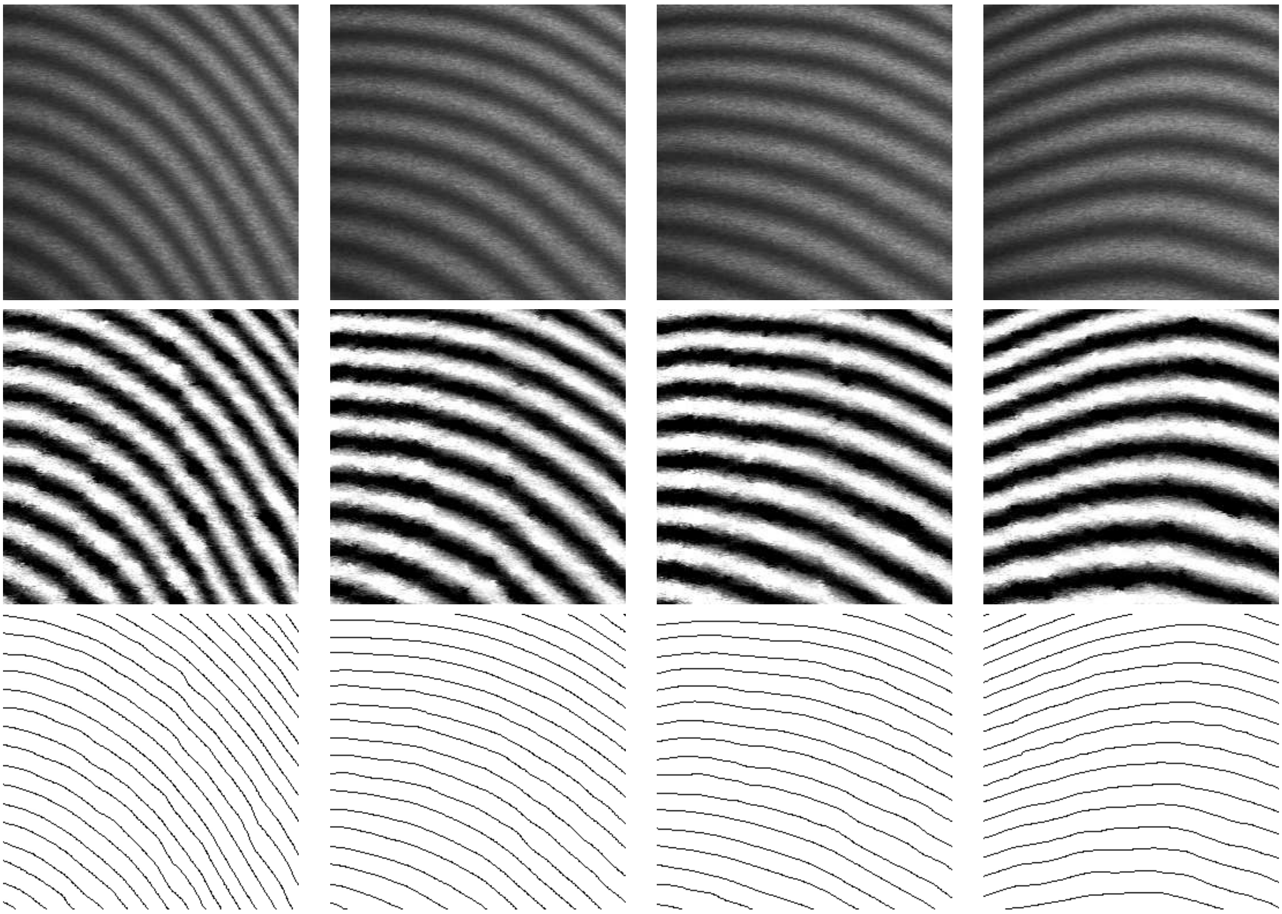


Figure 2: Part of a sequence of holographic interferograms with corresponding cosine maps and segmented fringe contours.

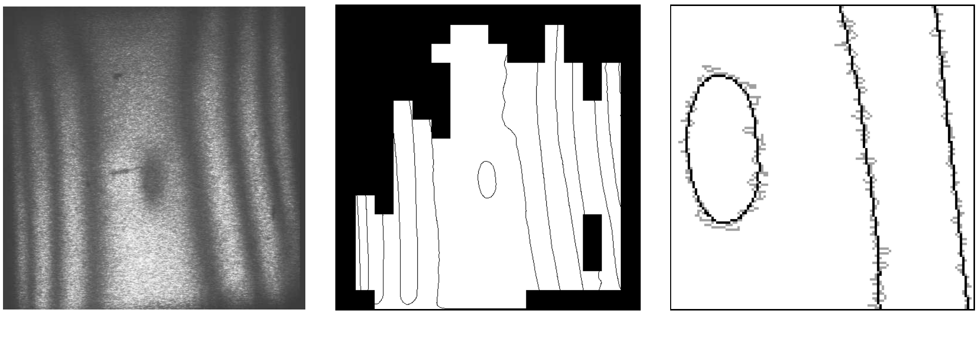
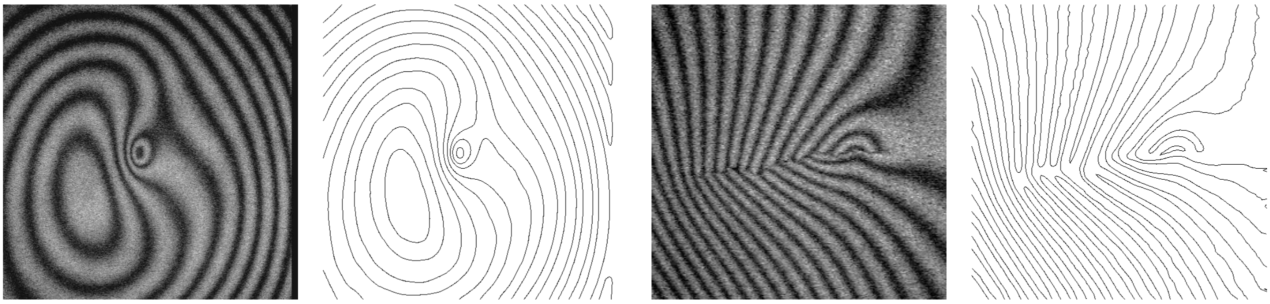


Figure 3: Holographic interferogram, segmented fringe contours with marked regions of low contrast, detail showing approximated (black) and original contour (grey).



(a) (b)

Figure 4: Holographic interferograms and segmented fringe contours of test objects with material faults; (a) acrylic glass plate with inclusion deformed by heating, (b) aluminium sheet with crack on surface deformed by tension.

6. ACKNOWLEDGEMENTS

This work has been supported by the German Research Foundation (DFG) in the project PA 599/1-1. We wish to thank our cooperation partner, the Bremen institute of applied beam technology (BIAS), for producing the interferograms.

7. REFERENCES

[1] U. Mieth, W. Osten, and W. Jüptner. Knowledge Assisted Fault Detection Based on Line Features of Skeletons. In Jüptner W. and W. Osten, editors, *Fringe '93*, pages 367–373, Berlin, October 1993. Akademie Verlag.

[2] W. Osten. *Digitale Verarbeitung und Auswertung von Interferenzbildern.* Akademie Verlag GmbH, Berlin, 1991.

[3] D.W. Robinson and G.T. Reid. *Interferogram Analysis.* IOP Publishing Ltd., London, 1993.

[4] J. Wang and I. Grant. Electronic speckle interferometry, phase-mapping, and nondestructive testing applied to real-time, thermal loading. *Applied Optics*, pages 3620–3627, 1995.

[5] Q. Yu. Spin filtering processes and automatic extraction of fringe centerlines in digital interferometric patterns. *Applied Optics*, pages 3782–3784, 1988.