Estimation of Curvature in 3D Images Using Tensor Field Filtering

H. Bårman G.H. Granlund H. Knutsson
Linköping University
Department of Electrical Engineering
Computer Vision Laboratory
S-581 83 Linköping Sweden

Abstract

This paper describes an algorithm for estimation of directionality in 2D and 3D vector fields and how that feature relates to the curvature of curves in 2D images and surfaces in 3D images.

One of the main properties of the method is that no thresholding is required. It consists of two steps. First the grey level image/volume is filtered with a number of filters to obtain a tensor description of the local orientation. Secondly the tensor image/volume is filtered with a number of filters to achieve the local direction description.

1 Introduction

Earlier papers (e.g [2]) have presented an methodology where a vector data representation is used for curvature estimation in 2D images. Here this algorithm is modified to work on data represented as tensors. This modification enables the generalization from 2D to 3D.

The curvature concept is well-known from vector analysis and differential geometry [7]. We will denote the curvature of a 2D curve with κ and the tangent of the curve with $\vec{t}$. Surfaces have a direction of most curvature as well as a direction of least curvature, and these directions are, apart from being perpendicular to the normal vector of the tangent plane of the surface, also perpendicular to each other. We will denote the two directions (*the principal directions*) as $\vec{k_1}$ and $\vec{k_2}$, while the amounts of curvature (*the principal curvatures*) will be denoted κ_1 and κ_2.

There exists a variety of different curvature estimation and description algorithms, e.g. [1, 3, 5, 6]. The new algorithm presented here differs from standard curvature algorithms in two very important aspects. First no thresholding is required. Secondly the detection is done hierarchically in two steps, where erroneous local orientation information is suppressed (as opposed to 'eliminated' which is the case in other two-step algorithms with thresholding) before the actual curvature estimation takes place.

2 Orientation Estimation

The first step is to achieve a local orientation estimate. The algorithm utilizes the observation that a neighbourhood with one dominant orientation has the energy in the Fourier domain concentrated around a line through the origin orientated at the orientation (or gradient direction) $(x,y)^T$ ($(x,y,z)^T$ for 3D images). A number of *quadrature filters* are applied on the grey level image/volume, where each filter is concentrated in a specific partition of the Fourier domain. The dominant local orientation is achieved by

$$f_1(\xi_x,\xi_y) = \sum_{k=1}^{K} q_k(\xi_x,\xi_y)(\mathbf{T}_k - \alpha\mathbf{I}) \qquad (1)$$

where f_1 denotes the obtained tensor image, ξ_x and ξ_y are spatial coordinates (add ξ_z for 3D), K is the number of filters, q_k denotes the magnitude of the filter response and $\mathbf{T}_k$ is the direction of the filter in the representation domain. This results in a tensor representation which for a dominant orientation $(x,y)^T$ (or $(x,y,z)^T$) equals

$$\mathbf{T}_{2D} = \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix} \mathbf{T}_{3D} = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix} \qquad (2)$$

$\mathbf{T}_k$ in Eq. (1) is computed by expressing the Fourier domain direction of the filter (the direction where the filter is concentrated) as a cartesian vector and put it into Eq. (2). The filters are evenly spaced over a half of the Fourier space. [4] describes the algorithm in detail. It should be noted that this representation is continuous and contains a certainty measure.

3 Curvature Estimation in 2D

The curvature property of curves in the original image is transferred into the orientation image

$f_1(\xi_x, \xi_y)$. It turns out that estimation of curvature direction can be made in the same manner as the estimation of local orientation, i.e. by a summation of the magnitude responses of a number of filters, where each filter is concentrated in a partition of the Fourier domain.

It is in [2] shown that a 2D–neighbourhood with one dominant curvature will have a local Fourier spectra where the centre of gravity is located in the direction of $\vec{t}$ provided that the pixel values are complex and equals

$$(x^2 - y^2) + i2xy. \tag{3}$$

This implies that the following formula (see Fig. 1) can be used

$$f_2(\xi_x, \xi_y) = \sum_{k=1}^{K} q_k(\xi_x, \xi_y)\mathbf{n}_k \tag{4}$$

The vector achieved from Eq. (4) will coincide with the direction of the tangent of the curve in the grey level image. The magnitude of the vector relates to the curvature κ of the curve. The magnitude q_k of the tensor field filtering in Eq. (4) is computed as

$$\begin{aligned} q &= [(h_e * x^2 - h_e * y^2 + 2h_o * xy)^2 + \\ &\quad (h_e * xy - h_o * x^2 + h_o * y^2)^2]^{\frac{1}{2}} \end{aligned} \tag{5}$$

where h_e and h_o are the even and odd parts of the quasi–quadrature filter. Observe that the entire Fourier domain is covered by filters as opposed to the case of orientation estimation, where it is enough to cover half of the Fourier domain. A quasi–quadrature filter is defined in the Fourier domain as

$$H(\mathbf{u}) = \mathbf{H}_\rho(\mathbf{u}) \cos^{2\mathrm{A}} \frac{\phi}{2} \tag{6}$$

where $\mathbf{u}$ is the frequency coordinate vector, u is the length of the vector and

$$\phi = \arccos(\frac{\mathbf{n}_k \cdot \mathbf{u}}{u}) \tag{7}$$

and $\mathbf{n}_k$ is a unit vector determining the main direction of the filter. H_ρ describes the frequency characteristics. The parameter A specifies how concentrated the filter is with respect to its main direction.

Note that the simple formulation of Eq. (4) results in the following features.

- The magnitudes in the neighbourhood are taken into account so that only relevant parts (pixels on the curve) have effect on the computation.
- the gradient of $\varphi(x, y)$ is estimated for those pixels.
- The magnitude of the estimate $f_2(\xi_x, \xi_y)$ contains information about the certainty of the input data, i.e. the quality of the orientation estimates in the neighbourhood, as well as information about the fit to the curvature model.
- The curvature magnitude κ is implicitly reflected through the magnitude of $f_2(\xi_x, \xi_y)$ and the frequency characteristics of the filters used.

The algorithm can be modified to take into account that the tangent and gradient of a curve are perpendicular, i.e.

$$\arg(f_1'(\xi_x, \xi_y)) = 2\arg(f_2(\xi_x, \xi_y)) + \pi \tag{8}$$

Neighbourhoods not fulfilling Eq. (8) are not of curve/curvature type and the direction of the vector field corresponds to another type of event, e.g. line ends.

4 The algorithm in 3D

The interpretation of the orientation tensor as a complex number (Eq. (3) can in the 3D–case be done in three different ways. (Substitute x or y in Eq. 3 with z.) Applying Eq. (4) on the three interpretations results in three different 3D–vectors. It can be shown (proof omitted) that the vectors will point in the principal direction (or in the opposite direction) of most curvature provided that there is one dominant curvature direction in the neighbourhood.

Experiments have shown correct estimation of curvature direction and a reasonable total (the three estimates combined) magnitude invariance of the curvature direction. The scheme has been able to keep track of the weaker 'least curvature direction'. Even the surface of a sphere, with two equal strength curvatures, is handled correctly.

5 The Inverse

The inverse (the transformation from the curvature description to the principal directions $\vec{k_1}$ and $\vec{k_2}$) is obtained by summing the outer products of the three 3D–vectors (denoted $\mathbf{b}_x, \mathbf{b}_y$ and $\mathbf{b}_z$) and computing the eigenvalues of the obtained matrix.

$$\sum_{k=x}^{z} \mathbf{b}_k \mathbf{b}_k^T \tag{9}$$

The eigenvector of the largest eigenvalue determines, apart from the sign, $\vec{k_1}$. The eigenvector of the second largest eigenvalue determines, also apart from the sign, $\vec{k_2}$. The third eigenvalue should for a well defined surface be close to zero and the local normal

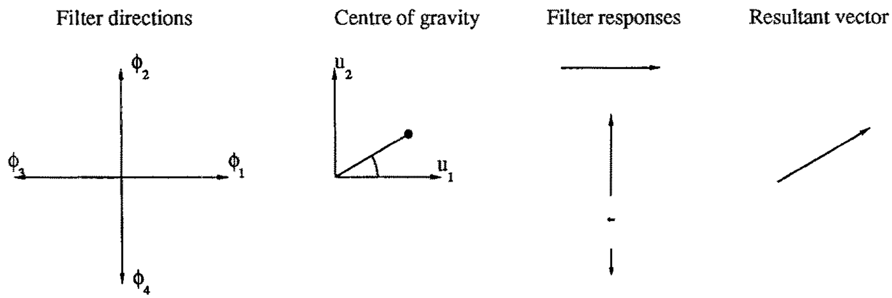

Figure 1: A stylized example of Eq. (4).

vector should be perpendicular to the two curvature directions. The sign of $\vec{k_1}$ and $\vec{k_2}$ is obtained by checking the directions of the three 3D-vectors.

6 Conclusion

A new algorithm for 3D curvature description has been presented. It is a hierarchical non-thresholding method, where the curvature is estimated on a gradient-equivalent image derived from the grey level volume (or time sequence). Both steps are performed without thresholding and with convolution as base operation. The algorithm output consists of a continuous representation constituted by three different 3D vectors. This representation can be translated into the two principal curvature directions.

Acknowledgment

This work was supported by the National Swedish Board for Technical Development.

References

[1] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-8(1):2–14, January 1986.

[2] H. Bårman, G. H. Granlund, and H. Knutsson. A new approach to curvature estimation and description. In *3rd International Conference on Image Processing and its applications*, pages 54–58, Warwick, Great Britain, July 1989. IEE. ISBN 0 85296382 3 ISSN 0537–9989.

[3] E.D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In William J. Wolfe and Nelson Marquina, editors, *Mobile Robots*, pages 161–168. SPIE, Bellingham, 1987. vol. 727.

[4] Hans Knutsson. Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, June 1989. Report LiTH-ISY-I-1019, Linköping University, Sweden.

[5] Jan J. Koenderink. An internal representation for solid shape based on the topological properties of the apparent contour. In Whitman Richards and Shimon Ullman, editors, *Image Understanding 1985–86*, chapter 9, pages 257–285. Ablex Publishing Corporation, 1987.

[6] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *Pattern Analysis and Machine Intelligence*, PAMI–11(8), August 1989.

[7] Michael Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 2. Publish or Perish, Inc., 2nd edition, 1979.