in: M. van Ginkel, J. van de Weijer, L.J. van Vliet, and P.W. Verbeek, Curvature estimation from orientation fields, in: B.K. Ersboll, P. Johansen (eds.), SCIA'99, Proc. 11th Scandinavian Conference on Image Analysis (Kangerlussuaq, Greenland, June 7-11), Pattern Recognition Society of Denmark, Lyngby, 1999, 545-551.

Curvature Estimation from Orientation Fields

M. van Ginkel, J. van de Weijer, L.J. van Vliet, P.W. Verbeek
Pattern Recognition Group,
Department of Applied Physics,
Delft University of Technology
Lorentzweg 1, 2628 CJ, Delft, The Netherlands,
{michael,joostw,lucas,piet}@ph.tn.tudelft.nl

Abstract

We propose a new curvature estimator, which operates on the output of an orientation estimator. Robust orientation estimators have been available for a long time. Some properties of the estimator as well as potential problems and limitations are discussed. The theory is verified by some experiments and practical limitations are investigated. The method is robust and performs well over a wide range of signal to noise ratios, down to about 0 dB.

1 Introduction

Orientation is one of the key features in a variety of image analysis applications. Robust estimators for orientation have been available for some time [7, 4, 11]. In the current paper we describe a method to estimate curvature based on these orientation estimators.

We distinguish between two classes of orientation estimators. The first class consists of estimators for a locally dominant orientation. The second class is able to cope with complex scenes of overlapping oriented patterns. Examples of the latter class are the Hough transform and the orientation space [1, 2, 6, 8] approach. In [3] we have shown how to estimate curvature using the orientation space approach. When an image can be well described using a single locally dominant orientation, the orientation space approach is unnecessarily complex both with respect to the complexity of the method itself as well as the computational cost. In this paper we show a simple method for computing the curvature from the output of an estimator for locally dominant orientation.

The structure of the paper is as follows; we start with a few definitions and a short description of the orientation estimator used in this paper. We then proceed by defining curvature and our method for estimating curvature. Two variants of the estimator are introduced. In the sections that follow, we investigate some of the properties and inherent limitations of the method.

2 Orientation Estimation

Although analysis of curved patterns is our objective, in the following sections we will first review the analysis of oriented straight patterns. At a given point an image is modeled as a straight pattern (curvature zero). Curved patterns can be analysed by considering a neighbourhood of such points, or equivalently a set of slightly rotated, slightly displaced, straight patterns.

2.1 Orientation

We assume that images can locally be modeled as a translation invariant pattern or (paintbrush) stroke. Such a stroke has a one-dimensional intensity profile and an orientation: the profile orientation across the stroke, see figure 1.

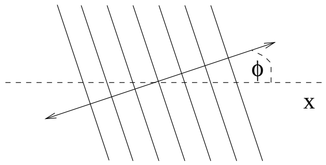


Figure 1: An oriented pattern

We refine our definition of orientation by distinguishing between angles in the interval $[-\pi, \pi\rangle$ and angles in the interval $[-\pi/2, \pi/2\rangle$. The term direction refers to an angle in the interval $[-\pi, \pi\rangle$, thus making a distinction between vectors along the same line, but with a different sign. We reserve the term orientation for angles in the interval $[-\pi/2, \pi/2\rangle$. Vectors in opposite directions have the same orientation.

2.2 The Gradient Square Tensor

Our curvature estimator is applied to an orientation field. To test our curvature estimator we use the Gradient Square Tensor (GST) to perform the orientation estimation. The GST is a well known robust orientation estimator [7, 4, 11]. We briefly review it here, because some of its properties have consequences for the behaviour of the curvature estimator.

The GST approach is essentially a simple gradient based orientation estimator, followed by a regularisation step. The direction of the gradient is an estimator of the local orientation, but is very susceptible to noise. The gradient vectors in a neighbourhood of a straight pattern disturbed by noise have (on average) identical orientation. This suggests using a simple vector averaging scheme, but this is not a viable scheme due to the fact that about half the gradient vectors have an opposite direction with respect to the other half. Averaging results in cancelation of these opposite vectors. The cancelation problem can be solved by embedding the gradient of an image $I(x, y)$ in the following tensor representation:

$$T = \nabla I \; \nabla I^T = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \tag{1}$$

Each of the tensor elements T_{ij} is averaged over the same local neighbourhood. Since this tensor representation is a quadratic form there are no cancelation problems. The final orientation estimate is obtained by performing an eigenvalue analysis of the smoothed tensor. The local orientation is given by the orientation of the eigenvector corresponding to the largest eigenvalue.

2.3 Curved patterns and the GST

The GST gives correct orientation estimates for straight patterns such as depicted in figure 1. Since our interest lies in the analysis of curved patterns, the behaviour of the GST on such patterns must be briefly discussed. A prototypical curved pattern is depicted in figure 2.

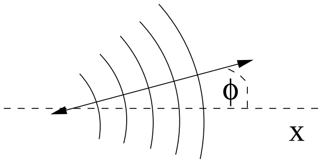


Figure 2: A curved pattern

Consider the local orientation axis in figure 2. As long as the pattern is symmetrical with respect to the orientation axis, the GST will yield an unbiased orientation estimate. Any deviation from this model will result in a biased estimate, and this will also affect the curvature estimator. The estimator that will be introduced in the next section depends only on orientation changes, so a locally constant bias will not influence the result.

2.4 Singularities

It is important to note that both orientation and curvature do not have to exist over the entire image domain. At the centre of a radial pattern, of which we can see a part in figure 2, both orientation and curvature are undefined. Orientation is well behaved in the vicinity of this singularity, but curvature becomes unbounded as the centre is approached. In section 3.2 an upper bound on the curvatures that can be measured is given.

3 Curvature Estimation

3.1 Curvature

Although our interest lies in the curvature of patterns such as depicted in figure 2, we will first give the definition of curvature for a curve. Our definition of curvature for curved patterns is based on the same principle.

The curvature κ at any point along a two-dimensional curve is defined as the rate of change in tangent direction θ of the contour, as a function of arc length s [9].

$$\kappa = \frac{\mathrm{d}\theta}{\mathrm{d}s} \tag{2}$$

It is common practice to apply the previous definition to gray value images by considering isophotes. The curvature of an isophote in a gray-value image $I(x, y)$ is given by the following formula [10]:

$$\kappa = -\frac{I_{cc}}{I_g} = -\frac{I_x^2 I_{yy} - 2 I_x I_y I_{xy} + I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{\frac{3}{2}}} \tag{3}$$

Where I_g is the derivative in the gradient direction, i.e. the gradient magnitude, and I_{cc} is the second derivative in the direction perpendicular to the gradient. Using equation 3 to estimate the curvature is inappropriate for images consisting of the type of patterns shown in figure 2, due to the fact that the gradient vanishes on ridges and in valleys [10]. The isophote curvature also changes sign at these locations, thus giving information on which side of the ridge (closer or farther away from the centre) we are located. The estimator introduced below uses the sign to give more useful information. Finally, isophote curvature is susceptible to noise and there is no easy way to regularise equation 3.

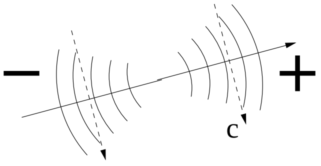


Figure 3: The estimator computes the derivative of the orientation field along the dotted lines.

Since isophote curvature is an inappropriate tool for our type of images, we use the following definition of curvature instead: curvature is the change in orientation in the direction along the strokes, perpendicular to the orientation, as indicated by the dotted lines in figure 3. Our curvature definition is given by:

$$\begin{aligned}\kappa(x,y) = -\frac{\partial\phi(x,y)}{\partial c} = \\ -\sin\phi(x,y)\frac{\partial\phi(x,y)}{\partial x} + \cos\phi(x,y)\frac{\partial\phi(x,y)}{\partial y}\end{aligned} \tag{4}$$

where c is the direction perpendicular to the local orientation. It is apparent from figure 3 that in the right half of the picture the orientation increases along the c axis, while it decreases in the left half. The sign of the curvature will therefore be positive for the pattern in the right half and negative for the pattern in the left half. It shows in which direction to look along the orientation axis in order to find the origin of the pattern.

Equation 4 cannot be implemented directly due to the fact that $\phi(x,y)$ contains jumps, because ϕ lies in the interval $[-\pi/2, \pi/2\rangle$. This and other problems are discussed in the following sections.

3.2 Phase jumps

There are two possible ways to deal with the jumps in the ϕ image: either remove the jumps or make the computation of κ insensitive to them. Getting rid of such jumps is called phase unwrapping and can be easily done for one-dimensional signals. In two dimensions this process is non-trivial [12] and we do not pursue this approach in this paper.

Instead we will make equation 4 insensitive to jumps in ϕ. We start by noting that $\exp(2i\phi(x,y))$ is a continuous function of x and y. Taking the derivative with respect to x yields:

$$\frac{\partial\exp(2i\phi(x,y))}{\partial x} = 2i\phi_x(x,y)\exp(2i\phi(x,y)) \tag{5}$$

Reordering yields an equation that allows us to compute ϕ_x even though ϕ contains jumps:

$$\phi_x(x,y) = -\frac{1}{2}i\exp(-2i\phi(x,y))\frac{\partial\exp(2i\phi(x,y))}{\partial x} \tag{6}$$

In this way ϕ_x and ϕ_y can be computed despite the jumps in ϕ. Equation 4 can subsequently be used to estimate the curvature. The method is strongly related to the work on phase analysis by Jepson and Fleet [5].

It is possible to establish an upper limit on the curvature that can be estimated using this method. Consider the one-dimensional signal $\phi(x) = \kappa x$. If κ is larger than or equal to $\pi/2$, then $\exp(2i\phi(x))$ will be undersampled. The largest curvature allowed is therefore $\pi/2$, corresponding to a radius $(1/\kappa)$ of approximately 0.64. From here on, we will use the convention that the value between brackets behind a curvature value indicates the corresponding radius, for instance 0.2(5).

Implementation of equation 6 for sampled images results in some subtle problems. These will be addressed in the next section and alternative versions of equation 6 will be given.

3.3 Implementation

The derivations in the previous section assume a continuous image. The curvature estimator that was developed involves derivative operations. Great care should be taken whenever an operator involving derivative operators is carried over to the discrete domain.

Equation 6 is potentially sensitive to improper implementation of the derivative operators. It is not possible to create a true discrete derivative. Instead we have to resort to sampled versions of regularised derivative operators. It is indeed the regularisation rather than the dicretisation itself, that causes the problems. Consider an arbitrary regularised derivative operator $D(_xx,y) = (\partial/\partial x)S(x,y)$, where S is the regularisation filter. Applying D to $\exp(2i\phi)$ yields:

$$\begin{aligned}D_x(x,y) * \exp(2i\phi(x,y)) &= \\ &= \frac{\partial}{\partial x}(\exp(2i\phi(x,y)) * S(x,y)) \\ &= (\frac{\partial}{\partial x}\exp(2i\phi(x,y)) * S(x,y) \\ &= (2i\frac{\partial\phi(x,y)}{\partial x}\exp(2i\phi(x,y)) * S(x,y) \\ &\neq 2i\exp(2i\phi(x,y))(\frac{\partial\phi(x,y)}{\partial x} * S(x,y))\end{aligned} \tag{7}$$

The last inequality shows that we cannot simply assume the complex exponential above to cancel with the complex exponential $\exp(-2i\phi(x,y)$ in equation 6, when we replace the derivative in equation 6 by $D_x(x,y)$. In fact, ϕ_x will generally be complex valued.

3.3.1 Modifying the estimator

Despite the arguments above we still expect equation 6 to be approximately correct for discrete images. Two modified versions of equation 6 are introduced below and will be evaluated by the experiments in section 4.

The simplest way of dealing with the complex valued ϕ_x is to simply disregard the imaginary part (since we expect it to be small). The estimate $\tilde{\phi}_x$ for ϕ_x becomes:

$$\begin{aligned}\tilde{\phi}_x(x,y) = Re\Big\{-\frac{1}{2}i\exp(-2i\phi(x,y)) \\ D_x * \exp(2i\phi(x,y))\Big\}\end{aligned} \tag{8}$$

Instead of trying to cancel the phase, it is also possible to directly ignore the phase by only considering the magnitude. The approximation for equation 6 becomes:

$$\tilde{\phi}_x(x,y) = \mathrm{sign}(\phi_x(x,y))\frac{1}{2}\Big|D_x * \exp(2i\phi(x,y))\Big| \quad (9)$$

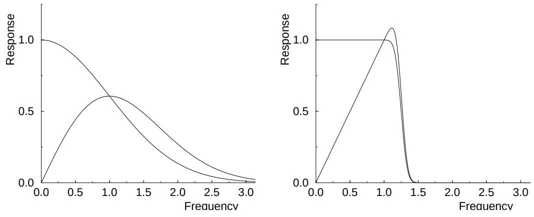
The sign of ϕ_x is taken from equation 8.

The last variant is based on the idea that it may be possible to compensate for the regularisation by also smoothing the cancelation factor:

$$\begin{aligned}\tilde{\phi}_x(x,y) = Re\Big\{-\frac{1}{2}i\Big[S(x,y) * \exp(-2i\phi(x,y))\Big]\\ D_x * \exp(2i\phi(x,y))\Big\}\end{aligned} \quad (10)$$

3.3.2 Discrete derivative operators

Discrete derivative operators are always approximations to the true derivative operators in the continuous domain. A popular class of derivative operators is the family of Gaussian derivative filters. These have a very good localisation in both the spatial and the Fourier domain. Furthermore they are relatively insensitive to noise because of the Gaussian regularisation filter. Examination of the frequency characteristic of the Gaussian and the first derivative of Gaussian filters in figure 4a shows that for low frequency signals a Gaussian derivative filter is nearly identical to a "true" derivative operator (it shows "$j\omega$" behaviour).



(a) (b)

Figure 4: Fourier transforms of a) the Gaussian regularisation filter and its derivative. b) The non-distorting regularisation filter and its derivative.

For many applications the Gaussian derivative family is the ideal set of derivative operators. Despite this, given equation 6's potential sensitivity to the choice of derivative operator, the distortion by the Gaussian regularisation filter may already be too severe. To investigate the influence of the derivative operator, we have also considered another regularisation filter. It has a flat response in the pass band and a smooth transition to the stop band. Its frequency response is shown in figure 4b along with the response of the derivative operator based on it. This non-distorting regularisation filter N is given by:

$$N(\omega, r, \sigma) = \frac{1}{2}(1 - \mathrm{erf}(\sigma(\omega - r))) \quad (11)$$

The position of the transition band is controlled by r and its width by σ. To prevent spatial aliasing the width of the transition band should not be too small.

4 Experiments

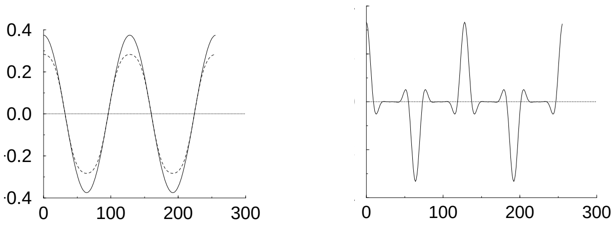
4.1 One-dimensional experiments

In this section we examine how ϕ_x is influenced by the choice of derivative operator and the equation used to implement the estimator. Our first test signal is $\phi(x) = \kappa x$. For this simple signal it is possible to predict the distortion by the derivative filters. Consider $\kappa = \pi/4$; the derivative operator is applied to $\exp(2i\phi(x)) = \exp(i(\pi/2)x)$. The effect of the regularisation filter on this signal is a simple scaling, because the signal has only one frequency component (at $\omega = \pi/2$). In the Gaussian case with $\sigma = 1$ the scaling factor will be $\exp(-(\pi/2)^2/2) \approx 0.291$ resulting in a estimate of 0.229 instead of $\pi/4 \approx 0.785$. Equation 10's smoothing of the cancelation factor will introduce a second scaling and make the results twice as bad. We have therefore not further considered this variant. The non-distorting filter with $r = \pi/2$ and $\sigma = 0.1$ results in a scaling by 0.5, yielding an estimate of 0.393. This can be verified by the results in the following table:

Table 1: The estimated curvature κ for $\phi(x) = \kappa x$. Both derivative operator types and equations 8 and 9 have been used. The values have been obtained by averaging over 128 pixels. The corresponding standard deviations are of the same order as the floating point precision and are therefore not listed.

true κ	estimated κ			
	Gaussian $\sigma = 1$		Non-distorted $r = \pi/2, \sigma = 0.1$	
	eq. 8	eq. 9	eq. 8	eq. 9
0.7854 (= $\pi/4$)	0.2287	0.2287	0.3927	0.3927
0.5	0.3033	0.3033	0.4950	0.4950
0.2	0.1846	0.1846	0.2000	0.2000
0.1	0.0980	0.0980	0.1000	0.1000

The results are in complete agreement with the theory. They show that as long as the signal (the complex exponential) lies in the pass-band of the non-distorting filter, the correct answer is obtained. For curvatures smaller than 0.1(10) the error is less than 2% when using Gaussian derivative filters.



(a) (b)

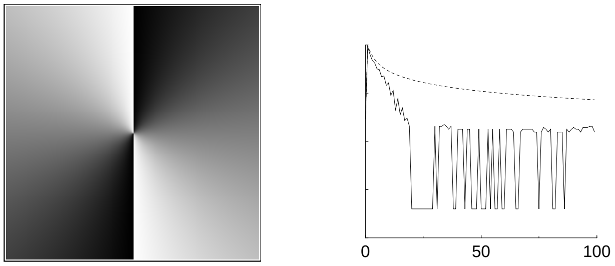
Figure 5: a) The true derivative of $\phi(x, y)$ (solid) and the estimate using Gaussian derivatives (dashed). b) The *difference* between the true derivative and the estimate using non-distorting derivatives.

The linear test signal used in the previous shows that the error depends on the slope of the signal. A second test signal, $\phi(x) = (24/\pi)\sin(\pi x/64)$, with a varying slope has been used to gain further insight into the error. The slope varies between 0 and 0.375. Figure 5a shows the true derivative and the estimate obtained by equation 8 using Gaussian derivatives. The same experiment was done using the non-distorting derivatives with $r = \pi/2$ and $\sigma = 0.1$. The error of the estimated derivative is too small to be visual in a graph such a figure 5a. Instead the difference between the true and the estimated curvature is given in figure 5b. Note the difference in axis scaling between figure 5a and b.

We have also examined the difference in performance between equation 8 and 9. For both the Gaussian and the non-distorting filters the maximum deviation between the two estimates is very small, 1.8 10^{-4} and 8.9 10^{-8} respectively.

4.2 Synthetic orientation data

In the previous section we investigated the behaviour of our derivative estimator on one-dimensional signals. In this section we look at the limitations of the curvature estimator by applying it to noise free, generated, orientation data. The test image is $\phi(x, y) = \text{atan}(y/x)$ and is shown in figure 8a. The curvature of this image is given by $\kappa(x, y) = 1/\sqrt{(}x^2 + y^2)$.
Figure 8b shows the absolute errors of the estimated curvature along a horizontal line through the middle of the test image starting from the centre. Although the errors made by the Gaussian derivative filters are several orders of magnitude larger than those made by the non-distorting filters, they are so small that for all practical purposes both implementations perform without error, assuming that the curvature is not too large. The large errors at the right end of the graph should be ignored; they are caused by border effects.



(a) (b)

Figure 8: a) Synthetic orientation data. b) The error in the curvature estimate using Gaussian derivatives (dashed) and using non-distorting derivatives (solid).

4.3 Noise sensitivity

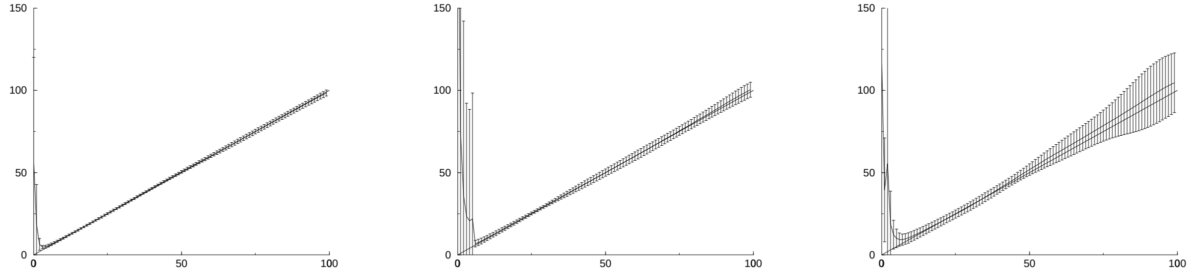
In this section we examine the effect of noise on the curvature estimator. $\phi(x, y)$ is not directly polluted by noise. The noise present in $\phi(x, y)$ is non-additive and non-white, since $\phi(x, y)$ is obtained from an input image using a non-linear estimator, the Gradient Square Tensor. Testing the performance of the curvature estimator on artificial $\phi(x, y)$ data is therefore useless.

Instead we generate noisy images $I(x, y)$ and apply the complete scheme, including the GST stage. The test image is $I(x, y) = \cos(\sqrt{(x^2 + y^2)})$. Gaussian distributed noise is added to $I(x, y)$ and the estimator is applied. For each signal to noise ratio we have repeated the experiment for 25 different noise realisations. The signal to noise ratio is defined by:

$$\text{SNR} = \frac{A}{\sigma_N} \qquad (12)$$

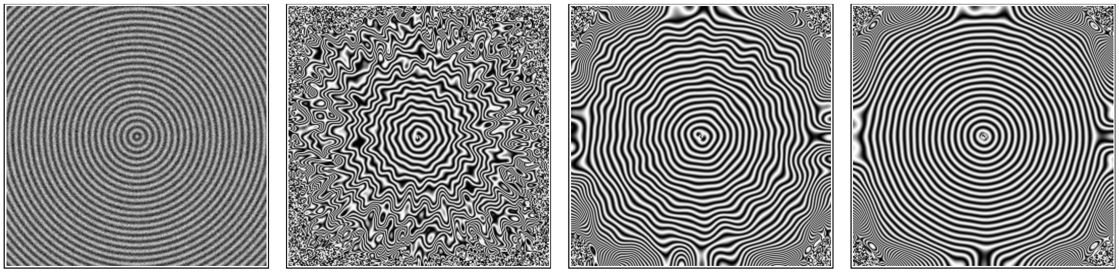
A is the amplitude of the signal (in this case 1, half the peak to peak value) and σ_N is the standard deviation of the noise. Figure 6 shows the results for SNR's of 4, 2 and 1. The GST stage uses Gaussian derivatives with $\sigma = 1$ and the tensor elements are smoothed using a Gaussian with $\sigma_t = 10$.

In figure 7 we have attempted to visualise the influence of noise. We have taken the same test pattern as above with SNR=4. After computing the curvature, we generate the following image: $I(x, y) = \cos(1/\kappa(x, y))$, which should yield an image containing concentric circles. Figure 7b shows the results for tensor smoothing $\sigma_t = 5$. It is clear that the estimator is accurate for large curvatures (except the very large near the centre of the pattern), but fails for smaller curvatures. Note, however, that the cos mapping is very sensitive; good results mean a good estimate, but bad results do not imply a bad estimate. Small curvatures correspond to patterns slowly varying in orientation. To accurately



(a) (b) (c)

Figure 6: Results of the curvature estimation on noisy data. For interpretation purposes we depict $1/|\kappa|$ rather than κ itself. $1/|\kappa|$ was averaged over 25 noise realisations. The standard deviations are indicated. The experiment was repeated for different SNR's: a) SNR=4, b) SNR=2, c) SNR=1.



(a) (b) (c) (d)

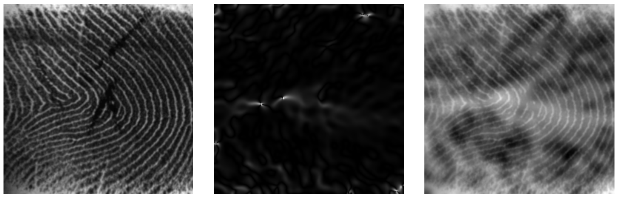
Figure 7: a) A noisy pattern of concentric circles (SNR=4). b) and c) show $\cos(1/|\kappa|)$ for tensor smoothing σ_t is 5 and 10 respectively. d) The same as c), only $1/|\kappa|$ was averaged over 25 noise realisations.

describe the orientation at such locations, we need to use a larger analysis window. Indeed, figure 7c shows that after doubling the tensor smoothing ($\sigma_t = 10$) the results are accurate for small curvatures as well. This indicates that the amount of tensor smoothing should be adjusted to the local curvature, suggesting a two stage estimation. The first to get a rough estimate of the curvature, followed by a more accurate estimation using a spatially variant tensor smoothing.

Some artifacts can be observed in figure 7c. To make sure that these aren't systematic, we have averaged $\kappa(x, y)$ over 25 realisations. The result in figure 7d show that the estimator has no systematic deviations.

4.4 Real data

So far, the estimator has only been applied to various kinds of artificial data to test its limitations and accuracy. In this section we apply the estimator to real data, in particular an image of a fingerprint. There is no ground truth for verifying the results. The evaluation will therefore be strictly qualitative.



(a) (b) (c)

Figure 9: a) A finger print image. b) The absolute value of the estimated curvature. c) finger print image overlayed with a Gaussian ($\sigma = 5$) filtered logarithmic version of the absolute curvature.

Figure 9a shows the finger print image. The (absolute) curvature as estimated by our method is shown next to it. Two hot spots are visible that clearly correspond to topologically important points characterised by a large curvature.

The last image is an attempt to visualise the results using a fair amount of postprocessing. We start by taking the natural logarithm of the image. The resulting image has too many small scale fluctuations that make interpretation difficult. The image is smoothed by a Gaussian filter with $\sigma = 5$ and subsequently added to the finger print image to create the final overlayed image. The nearly horizontally oriented bright blob indicates a region with relatively high curvature. The two regions above and below the bright blob contain relatively straight patterns with a low curvature, which is evident from the dark blobs in the results.

5 Discussion

We have introduced a new curvature estimator which operates on the output of an arbitrary orientation estimator. The implementation of the estimator required careful consideration, because it depends in a subtle way on the quality of the derivative operators used. We have considered three versions of the estimator (equations 8, 9 and 10), as well as two different sets of derivative operators (Gaussian and non-distorting).

The experiments in sections 4.1 and 4.2 show that there is no difference in accuracy between equations 8 and 9. Equation 10 was shown introduce more errors than the other two.

These experiments also indicate that except for large curvatures, the distortion caused by the Gaussian regularisation is not significant. In fact, experiments on noisy data show that this distortion is insignificant even for large curvatures.

The experiments on noisy data show that the estimator is unbiased, rotation invariant and performs well even when a considerable amount of noise is present.

The results can be summarised as follows: equation 8 using Gaussian derivative filters implements a robust curvature estimator. The curvature that is to be measured should be smaller than 0.2(5) or, even better, 0.1(10) for the best results.

We have not compared the performance of the estimator to other curvature estimators, both existing and under development. This issue will be addressed in a separate paper evaluating several curvature estimators.

Acknowledgements

This work was partially supported by the Rolling Grants program 94RG12 of the Netherlands Organization for Fundamental Research of Matter (FOM) and the Royal Netherlands Academy of Arts and Sciences (KNAW).

References

[1] M. van Ginkel, P.W. Verbeek and L.J. van Vliet, Improved Orientation Selectivity for Orientation Estimation, *Proceedings of the 10th Scandinavian Conference on Image Analysis, volume I*, Lappeenranta, Finland, June 9-11, 1997, pp. 533-537.

[2] M. van Ginkel, P.W. Verbeek and L.J. van Vliet, Multi-Orientation Estimation: Selectivity and Localization, in: H.E. Bal, H. Corporaal, P.P. Jonker, J.F.M. Tonino (eds.), *Proceedings of the third annual conference of the Advanced School for Computing and Imaging*, Heijen, The Netherlands, June 2-4, 1997, pp. 99-105.

[3] M. van Ginkel, P.W. Verbeek and L.J. van Vliet, Curvature Estimation for Overlapping Curved Patterns using Orientation Space, in: B.M. ter Haar Romeny, D.H.J. Epema, J.F.M. Tonino, A.A. Wolters (eds.), *Proceedings of the third annual conference of the Advanced School for Computing and Imaging*, Lommel, Belgium, June 9-11, 1998, pp. 173-178.

[4] L. Haglund, Adaptive Multidimensional Filtering, *PhD thesis*, Linköping University, Sweden, 1992.

[5] A.D. Jepson and D.J. Fleet, Phase Singularities in Scale-space, *Image and Vision Computing*, vol. 9, no. 5, October 1991, pp. 338-343.

[6] S.N. Kalitzin, B.M. ter Haar Romeny and Max A. Viergever, Invertible Orientation Bundles on 2D Scalar Images, *Proceedings of the First International Conference on Scale-Space Theory in Computer Vision*, Lecture Notes in Computer Science, Springer-Verlag, July 1997, pp. 77-88.

[7] M. Kass and A. Witkin, Analyzing Oriented Patterns, *Computer Vision, Graphics and Image Processing*, vol. 37, 1987, pp. 362-385.

[8] J.B. Martens, Local Orientation Analysis in Images by Means of the Hermite Transform, *IEEE transactions on Image Processing*, vol. 6, no. 8, August 1997, pp. 1103-1116.

[9] J.J. Stoker, Differential Geometry, *John Wiley & Sons, Inc*, 1969

[10] P.W. Verbeek, A class of sampling-error free measures in oversampled band-limited images, *Pattern Recognition Letters*, vol. 3, 1985, pp. 287-292

[11] L.J. van Vliet and P.W. Verbeek, Estimators for Orientation and Anisotropy in Digitized Images, in: J. van Katwijk, J.J. Gerbrands, M.R. van Steen, J.F.M. Tonino (eds.), *ASCI'95, Proc. First Annual Conference of the Advanced School for Computing and Imaging* (Heijen, NL, May 16-18), ASCI, Delft, 1995, pp. 442-450.

[12] H.A. Vrooman, Quantitative analysis of interferograms *PhD thesis*, Delft University of Technology, The Netherlands, 1991.