Error-Bounds on Curvature Estimation

Sven Utcke

Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Kognitive Systeme
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
utcke@informatik.uni-hamburg.de
http://kogs-www.informatik.uni-hamburg.de/~utcke

Abstract. Estimation of a digital curve's curvature at any given point is needed for many tasks in computer vision, be it differential invariants or curvature scale space. However, curvature estimation is known to be very susceptible to noise on the contour. We shall show how noise on the contour affects the relative accuracy of the curvature computation. One interesting result is that, contrary to intuition, the accurate calculation of the curvature for low-curvature regions is in fact impossible for common image-sizes, while reasonable results may under favourable conditions be obtained for higher-curvature regions.

1 Introduction

In the early years of computer vision, the need to calculate a digital curve's curvature used to arise quite frequently. However, this did turn out to be a rather harder task than one originally appreciated (see [1] which compares several different approaches), and today the knowledge about the difficulty of the task has become so ingrained in the computer vision community that curvature is hardly ever used outside the scale space community, where much of the work is reduced to locating the zero crossings of curvature [2].

We believe, however, that only little attention has so far been given to the question *why* the calculation of curvature is hard. Worring's approach [1] is only relevant for edgels on a pixel-grid; Kovalevsky [3] recently extended this to edgels with sub-pixel accuracy, using his own edge-finder. However, both authors basically follow the assumption that curvature for low-curvature regions can be calculated more accurately than for high-curvature regions, since for the former smoothing will be less invasive than for the latter, and as a consequence concentrate on high-curvature regions with a radius of curvature below 40 pxl.

This paper strives to demonstrate that for real images the curvature of low-curvature regions is in fact just as difficult to compute. This is due to the limited size of outlines in real images, and we shall see that neither very low nor very high curvature can be estimated reliably under real life conditions.

This paper is organised as follows: Section 2 gives the analytic derivation of the expected relative error for a very simple case, this serves mainly to give a feeling for the mathematics involved. Section 3 gives results for a more involved, but also more realistic, case, but without the analytic derivation. Based on these,

L.D. Griffin and M. Lillholm (Eds.): Scale-Space 2003, LNCS 2695, pp. 657–666, 2003.

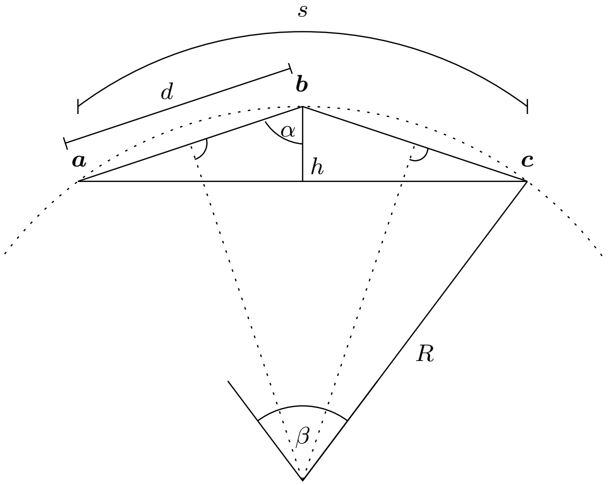

Fig. 1. Simple model for the calculation of curvature

Sec. 4 gives numeric examples of what can be expected from locally operating approaches for the calculation of the curvature, Sec. 5 discusses some additional aspects of curvature calculation, and Sec. 6 summarises our findings.

2 Analytic Derivation of a Simple Case

We will first analyse a very simple case. Given 3 points $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$ on a circle, we can calculate the circle's midpoint as the intersection of the two perpendicular bisectors of the 2 lines connecting point $\boldsymbol{a}$ with $\boldsymbol{b}$ and $\boldsymbol{b}$ with $\boldsymbol{c}$; the circle's radius is then straightforwardly given as the distance from the circle's midpoint to any of the three points on the circle. Figure 1 gives an overview of the construction.

The main source of error using this setup is the uncertainty in the position of the three points. For a digital contour given as edgel-chains, only the edgels' uncertainty perpendicular to the curve is of interest, while the exact location of the edgels along the curve is of no significance.

In order to simplify the example, we will assume that the position of the middle edgel varies with standard deviation σ along the vertical-direction, and that the two other edgels vary with standard deviation σ orthogonal to the line-segments $\boldsymbol{ab}$ and $\boldsymbol{bc}$ (so that d is constant – this only introduces a small error, which will later be ignored anyway). In addition, we will introduce a further simplification by assuming symmetry between the two sides of the construction (this introduces a much bigger error, but gives us a lower bound for the error and a much simpler model). The above simplifications allow us to replace the individual uncertainties in the positions of the edgels by a single uncertainty of the height h of the triangle connecting the three edgels, it is

$$\sigma_h = \sigma\sqrt{1+\sin^2(\alpha)}\ . \tag{1}$$

In doing so we are projecting the edgels' uncertainty onto the vertical axis.

Using similarity triangles, we can calculate the radius R as

$$R = \frac{d^2}{2h} \ . \tag{2}$$

So how does the uncertainty in h propagate to R? Using simple linear error propagation we can approximate the standard deviation of the radius σ_R by multiplying the covariance-matrix of all independent variables – in this case this is simply the scalar σ_h^2 – on both sides with the Jacobian of the dependent variables with respect to all independent variables – in this case this is simply the derivative of R with respect to h,

$$\frac{\partial R}{\partial h} = -\frac{d^2}{2h^2} \ . \tag{3}$$

We therefore get the approximation

$$\sigma_R \approx \sqrt{\frac{d^2}{2h^2} \cdot \sigma_h^2 \cdot \frac{d^2}{2h^2}} = \sigma_h \frac{d^2}{2h^2} \ , \tag{4}$$

for the standard deviation of the radius.

We are, however, interested in the relative accuracy of the radius as a function of the arc-length s between the outer edgels. From Fig. 1 we see that

$$\alpha = \frac{\pi}{2} - \frac{\beta}{4} \tag{5}$$

$$\beta = \frac{s}{R} \tag{6}$$

$$h = d\cos(\alpha) \tag{7}$$

and substituting (1) as well as (5), (6), and (7) in (4) leads to

$$\sigma_R = \frac{\sigma}{2} \frac{\sqrt{1 + \sin^2(\alpha)}}{\cos^2(\alpha)} = \frac{\sigma}{2} \frac{\sqrt{1 + \cos^2\left(\frac{s}{4R}\right)}}{\sin^2\left(\frac{s}{4R}\right)} \ . \tag{8}$$

In order for a result R to be usable we would require its relative error to remain sufficiently small, i. e.

$$\frac{\sigma_R}{R} = \rho \ll 1 \ . \tag{9}$$

Note that, as we only analyse the *relative* error ρ, it does of course not make any difference whether we calculate the radius R or the curvature $\kappa = 1/R$.

We will now calculate the minimum arc-length s necessary to satisfy (9), and from there, assuming a known average distance from edgel to edgel (set to 1 pxl in the following for simplicity), the number of edgels necessary to achieve a desired relative accuracy $\rho = \sigma_R/R$. As

$$0 \leq \cos^2\left(\frac{s}{4R}\right) \leq 1 \tag{10}$$

we get from (8)

$$4R \arcsin\left(\sqrt{\frac{\sigma}{2R\rho}}\right) \leq s \leq 4R \arcsin\left(\sqrt{\frac{\sigma}{\sqrt{2}R\rho}}\right) \tag{11}$$

and for $\sigma \ll R\rho = \sigma_R$, which is almost always the case

$$\frac{4}{\sqrt{2}}\sqrt{\frac{\sigma}{\rho}R} \leq s \leq \frac{4}{2^{1/4}}\sqrt{\frac{\sigma}{\rho}R} \ . \tag{12}$$

Note that the left and right side of (12) differ only by a factor of $2^{1/4} \approx 1.19$; in both cases the minimum required arc-length s for a required relative accuracy $\rho = \sigma_R/R$ is proportional to the same term and approximately

$$s \approx 3\left(\frac{\sigma}{\rho}R\right)^{0.5} \ . \tag{13}$$

The entire model is of course an oversimplification: we only use 3 out of N edgels, the end-edgels' covariance should be taken in radial direction (i. e. orthogonal to the contour), we assume that the distributions of the two end-edgels are not independent of each other (which they are). We also assume that curvature is constant over the entire section of the curve under consideration (which most likely it isn't). However, we do feel that the above model is well suited to give an initial idea of the behaviour of curvature calculation.

In the next section we will analyse a more realistic approach which does away with all of the above simplifications except that of constant curvature, in particular using all edgels along s.

3 Using N Edgels

When looking for a description of a curve-segment, instead of calculating this description directly from the edgels it is not uncommon to fit an algebraic curve to the edgels and use the parameters of the curve with the smallest error of fit to calculate the description. These curves are often polynomials in x and y, and in our particular application, where we are trying to find the radius of curvature of a given curve, it is a natural choice to fit part of a circle to the edgels[1].

Different methods exist to calculate the error of fit between a curve and N edgels. Probably the most intuitive method is to minimise the sum of orthographic, squared distances, which for a circle with midpoint (x_0, y_0) and radius R would be

$$C = \frac{1}{N}\sum_{i=1}^{N}\left(\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - R\right)^2 \ . \tag{14}$$

However, in general no closed form solution exists for (14), and it has therefore become customary to use the so called algebraic distance

$$C = \frac{1}{N}\sum_{i=1}^{N}\left((x_i - x_0)^2 + (y_i - y_0)^2 - R^2\right)^2 \ , \tag{15}$$

[1] This is especially the case as we assume a curve with constant curvature – in general more complicated models will yield better results, although one should be wary of too complex models [4].

for which a closed form solution exists [5]. It is worth noting that minimising either (14) or (15) in their above form will introduce an additional bias into the solution, which we will ignore in the following.

We now once again want to calculate the uncertainty of the radius calculated by minimising (14) or (15) and from there the number of edgels N (or the length s) of the segment to which we need to fit the circle in order to achieve a given relative error $\rho = \sigma_R/R$. The uncertainty σ_R could, in theory, be calculated in direct analogy with the last section as the matrix-product of the edgels' covariance matrix, multiplied on both sides with the Jacobian of R with respect to all edgels, $\boldsymbol{J}_R$. However, since no explicit solution for R exists, at least not for (14), we need a way to calculate the Jacobian without explicit knowledge about R as a function of the edgels $\boldsymbol{e} = (x_1, \dots, x_N, y_1, \dots, y_N)^T$. The tool to do so is the implicit function theorem, it is

$$\boldsymbol{J}_R = -\left(\frac{\partial^2 C}{\partial R^2}\right)^{-1} \left(\frac{\partial^2 C}{\partial R \partial \boldsymbol{e}}\right)^T \Bigg|_{R_{\min}} \tag{16}$$

provided that the Hessian $\frac{\partial^2 C}{\partial R^2}$ is indeed invertible at the point of the solution $R_{\min}$. If we further assume that all edgels $(x_i, y_i)^T$ share the same covariance $\sigma \boldsymbol{I}_2$, where $\boldsymbol{I}_2$ is the 2×2 identity matrix – this is a reasonable model for variance σ perpendicular to the curve – we can calculate the variance of R as

$$\boldsymbol{J}_R \;\sigma \boldsymbol{I}_{2N}\; \boldsymbol{J}_R^T = \sigma \left(\frac{\partial^2 C}{\partial R^2}\right)^{-1} \left(\frac{\partial^2 C}{\partial R \partial \boldsymbol{e}}\right)^T \left(\frac{\partial^2 C}{\partial R \partial \boldsymbol{e}}\right) \left(\frac{\partial^2 C}{\partial R^2}\right)^{-1^T} \Bigg|_{R_{\min}} . \tag{17}$$

Evaluating (17) analytically, a purely mechanical process, leads to rather lengthy (although for (15) still quite manageable) equations outside the scope of this paper. A numerical evaluation is, however, perfectly possible, and we can then fit a function to the numerical results; using the squared orthographic distance (14) the output of these calculations is well approximated by a function

$$N \approx 2 \left(\frac{\sigma}{\rho} R\right)^{0.4} , \tag{18}$$

while for the algebraic distance (15) a good approximation is

$$N \approx 3.7 \left(\frac{\sigma}{\rho} R\right)^{0.4} . \tag{19}$$

This is very similar in structure to (13), but with an exponent of 0.4 rather than 0.5. We will see in the next section what this means for practical applications.

4 Numeric Examples

In Sec. 2 and 3 we have calculated the minimum number of edgels required to compute a radius of curvature R with given relative accuracy $\rho = \sigma_R/R$,

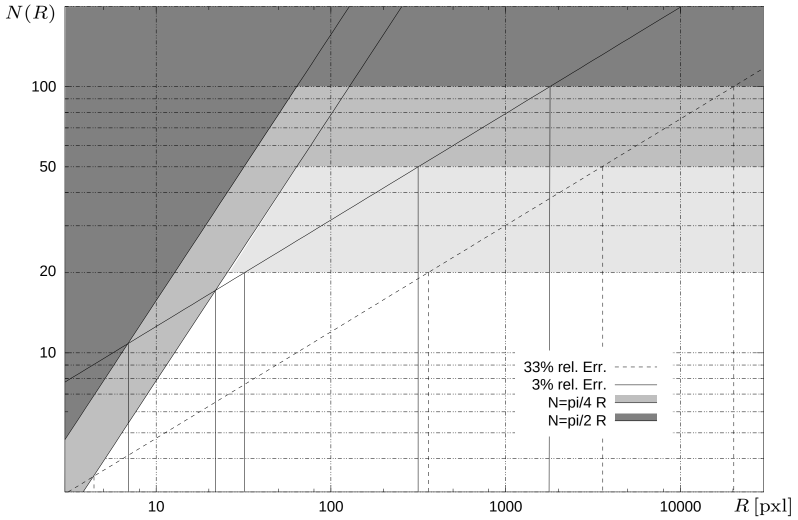

Fig. 2. Number of edgels $N(R, \rho, \sigma = 0.3\,\text{pxl})$ needed to calculate the radius of curvature R with a relative error of less than $\rho = 33\,\%$ (*dashed line*) or $\rho = 3\,\%$ (*solid line*). Inside the *grayed out* regions R can not even be determined with this accuracy if we assume a maximum arc $\beta = \pi/4$ $(\pi/2)$ and a maximum number of edgels $N = 20$ (50, 100)

compare (13), (18), and (19). It is now instructive to see what actual values for s or N are required when dealing with real images. For this it is important to know with what subpixel-precision we can expect to locate edges in real images. As cited in [6] a basic implementation of the Canny algorithm [7] will typically produce edgels with a standard derivation perpendicular to the edge from somewhere about $\sigma \approx 0.1\,\text{pxl}$ to $\sigma \approx 0.3\,\text{pxl}$ depending on the actual CCD (3-chip RGB versus BW versus Bayer RGB) and lens used as well as the contrast between regions. As most affordable colour cameras use only one CCD with a Bayer-filter and often rather cheap lenses, $\sigma = 0.3\,\text{pxl}$ is usually all we can hope for. Together with NTSC or, at best, PAL sized images as they are still predominant in computer vision, we are therefore hampered both by low image quality and small images.

Figure 2 shows how many edgels $N(R)$ are needed to compute the curvature (radius) with a relative accuracy of both $\rho = \sigma_R/R = 3\,\%$ (*solid line*) and $\rho = 33\,\%$ (*dashed line*), using (18). It is worth remembering that the latter corresponds to a worst-case error of about $3\rho \approx 100\,\%$, which means that not even the sign of curvature can be calculated reliably! But even accepting such an error we see from Fig. 2 that for a radius of approx. $R = 400\,\text{pxl}$ we already need to fit a circle to $N = 20$ edgels, by many authors considered the maximum region size which is still sensible to use [3]. The much more realistic (though still big) relative error of $\rho = 3\,\%$ would, for a radius of about $R = 400\,\text{pxl}$ already require a fit to $N > 50$ edgels, and even with $N \approx 100$ edgels could we only calculate radii up to $R \approx 2000\,\text{pxl}$. Similar limits exist for high-curvature regions, in order to calculate a radius of $R = 8\,\text{pxl}$ with a relative error of $\rho = 3\,\%$ we need $N \approx 12$ edgels, that's nearly one quarter of the full circle.

So why is this need for $N > 20$ (or even $N > 100$) a problem? After all, a PAL sized image comes with $768\,\text{pxl} \times 576\,\text{pxl}$, so $100\,\text{pxl}$ isn't really that big. Or is it? In order to understand the problem it is important to realise that so far we have calculated the curvature of something we *knew* to be a circle, i. e. a contour with constant curvature. In real-life, however, we will usually encounter curves with non-constant curvature, and it is important to realise that for such a contour, what we really calculate is not the curvature at the particular point under consideration, but some sort of average curvature. Assuming noise-free data, and as long as the real curvature in the interval under consideration is a monotonic function of the arc-length, we can at least guarantee that the curvature thus calculated would be correct for some point inside the interval – although almost certainly not for the point under consideration. This is called localisation uncertainty in [3]. However, as soon as the curvature over arc-length ceases to be a monotonic function (i. e. contains extrema), arbitrary results are possible even in the case of noise-free data.

From the above it is immediately clear that the region of N edgels from which curvature is calculated should not cross extrema of curvature (whose position we only know after the fact); to keep the localisation error small it is also desirable to keep N small enough so that curvature inside the interval doesn't vary too much. And the fact that the entire object, usually containing several extrema of curvature, has to fit into a PAL-sized ($768\,\text{pxl} \times 576\,\text{pxl}$) image, severely limits the maximum usable number of edgels, [3] cites values around 4 pxl–20 pxl. This is the factor we are most interested in. By contrast, most authors so far have been interested in high-curvature regions [1,3], where visible arc-length is the criterion; as a rule of thumb most authors would limit the arc under consideration to somewhere between $\beta \leq \pi/4$ and $\beta \leq \pi/2$.

In Fig. 2 regions with an arc $\beta > \pi/4$ and $\beta > \pi/2$ as well as $N > 20$, $N > 50$, and $N > 100$ have been grayed-out; only the remaining region (with white background) is suitable for curvature calculation, which clearly limits both the maximum, but also the minimum curvature which can be calculated with a given relative error ρ.

Figure 3 shows an exemplary calculation of curvature for a contour with $\sigma \approx 0.3$ and $N \in \{20, 50, 100\}$. We see that with $N = 20$ we get reasonable (although still quite noisy) results for $20\,\text{pxl} < R < 80\,\text{pxl}$. For smaller radii the calculated results are biased (due to extrema of curvature within the fitting-window) and for bigger radii results are so noisy as to be useless. Using more edgels clearly removes some of that noise, but at the same time introduces a bias into the result which can easily reach $100\,\%$ of the true result or more (note that the scale is logarithmic). Choosing a different scale for different regions on the contour is an obvious way to improve the results, but note Fig. 3 (*bottom left*), where for radii of around $R = 20\,\text{pxl}$ and $N = 50$ we already observe a noticeable bias, while curvature estimates are still quite noisy.

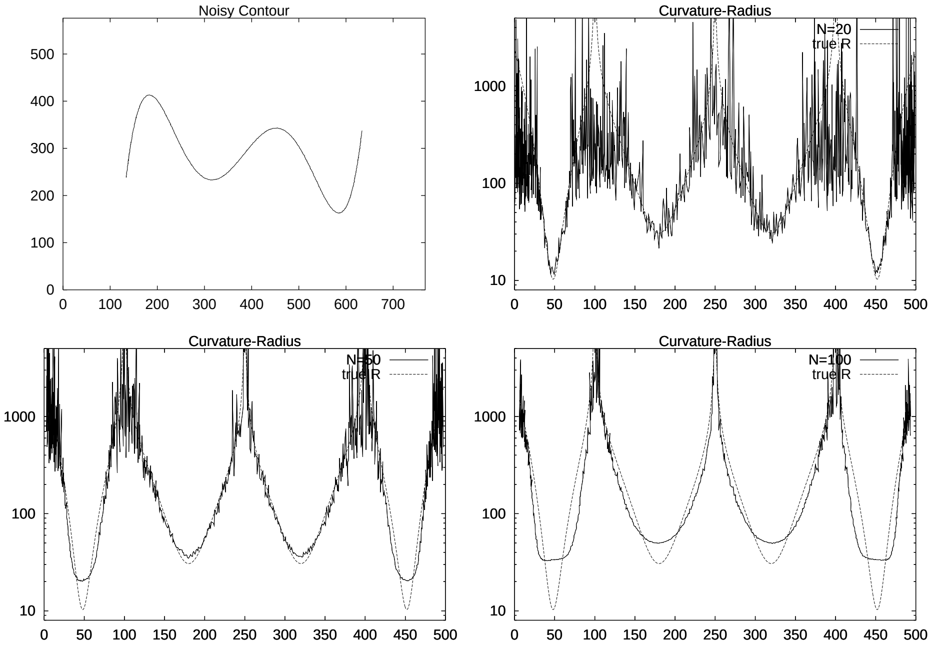

Fig. 3. Gradient calculated for a contour with $\sigma \approx 0.3$ (*top left*) and $N = 20$ (*top right*), $N = 50$ (*bottom left*), and $N = 100$ (*bottom right*)

5 Discussion

In this paper we presented an estimation of the minimum number of edgels required in order to calculate the curvature (-radius) with a given relative error $\rho = \sigma_R/R$. In our calculations we assumed the most simple case possible, that of a contour with constant curvature (essentially part of a circle) and except for the standard deviation of the edgel-positions ignored all additional error-sources (such as the systematic error inherent in (14) and (15)). Even under those optimal conditions did we find that curvature for low curvature regions is essentially impossible to calculate accurately – for real-life, non-constant curve segments the error will increase, due to location inaccuracy for contour segments with non-monotonic curvature (which introduces another systematic error) and the possible presence of curvature extrema within the segment, in which case arbitrary (and arbitrarily wrong) results are possible.

We can of course increase the location accuracy (i. e. the likelihood that the curvature estimated belongs to a point near the one for which we estimated it) by fitting a sufficiently complicated model to the curve; even just fitting a general conic cross-section rather than a circle would improve localisation. It should however be clear that the curves in Fig. 2 already constitute a best-case estimate – curvature calculated from real, non-circular data will always be less correct unless an *exact* model of the curve is known.

So is there no hope at all? An intuitive reaction would be to try and somehow "smooth" the contour – however, there is no better smoothing algorithm available than to try and fit a circle to data which we *know* to be circular. Any other smoothing algorithm can, with the same number of edgels N, only produce inferior results to such a fit. As long as we are calculating curvature based solely on edgel position, we can not get better than (13), (18), and (19)[2]. This means that in order to increase the accuracy of our calculations, we can either decrease σ, or increase the resolution of the image, thus increasing the number of edgels N on any given curve-segment, which would increase the maximum radius of curvature.

Decreasing σ, i. e. increasing the accuracy with which edgels can be located, unfortunately only offers limited room for improvements. Canny's algorithm is already quite good at this, and even algorithms which were specifically tuned to improved location accuracy[8] promise little better than $\sigma = 0.1\,\text{pxl}$. This basically leaves higher resolution as the only way out, and thankfully modern digital cameras offer just that. Using a 6 Mega-pixel camera (i. e. increasing the resolution by a factor of approx. 3.7) means that only about $3.7^{-0.6} \approx 45\,\%$ of the same curve still needs to be visible when compared to PAL, as the number of edgels on any given piece of curve increases, while at the same time their variance decreases.

Finally, I would like to comment on the use of discreet edgels (but with arbitrary coordinates) throughout this paper (rather than pretending that we are dealing with a continuous contour). The reason for this is of course that with real images discreet edgels are all we know how to compute. Using something like the Canny algorithm [7] these edgels are, however, approximately uniformly distributed along the contour (this is different from [3]), and since we are fitting a circle rather than calculating curvature directly from the edgel positions any irregularities would not have much of an effect. However, if you like to pretend that you are dealing with a continuous contour the same arguments still apply; in this case the location of the entire contour is uncertain, you can visualise the contour's error-distribution as that of a 1D-Gaussian distribution swept along the curve in such a way that it is orthogonal to the curve in each point. The radii of the set of circles fully contained within the 1-σ-band around the contour are approximately bounded by a 1-σ_R region around the median value of R, and the same goes for the k-σ-band. This way it might actually be easier to visualise what is happening.

6 Conclusion

In this paper, we have shown that using locally operating methods the curvature of both high-curvature as well as low-curvature regions can not be calculated reliably from standard PAL-sized images even under optimal conditions and

[2] Strictly speaking, this is only true for sufficiently small values of ρ, since we used linear error propagation throughout this paper. For larger values only qualitative statements are possible.

extending the meaning of "locally operating" to rather large regions; only for a possibly quite small range of medium-small curvatures is it at all possible to calculate a meaningful result. Increasing the resolution of the image by a factor u (linearly) will increase the maximum radius of curvature which can be calculated reliably (by $u^{0.6}$), but this does not eliminate the underlying problem.

As did [1,3], this paper only considered local algorithms for the calculation of curvature. It is possible that more global methods (gray-level- or gradient-based, spline-fit, or scale-space based methods, where the biased curvature-estimates in Fig. 3 are somehow propagated back to their true values) could provide additional accuracy. However, this has not been analysed.

Acknowledgements

I wish to thank the anonymous reviewers, especially reviewer two, and in particular Lewis D. Griffin for thankful comments and suggestions.

References

1. Worring, M., Smeulders, A.W.M.: Digital curvature estimation. Computer Vision, Graphics and Image Processing: Image Understanding **58** (1993) 366–382
2. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 789–805
3. Kovalevsky, V.: Curvature in digital 2d images. International Journal of Pattern Recognition and Artificial Intelligence **15** (2001) 1183–1200
4. Kanatani, K.: Geometric information criterion for model selection. International Journal of Computer Vision **26** (1998) 171–189
5. Pratt, V.: Direct least-squares fitting of algebraic surfaces. Computer Graphics **21** (1987) 145–152
6. Utcke, S.: Grouping based on projective geometry constraints and uncertainty. In: Proceedings of the Sixth International Conference on Computer Vision, Bombay, IEEE Computer Society, Narosa Publishing House, New Delhi (1998) 739–746
7. Canny, J.F.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698
8. Overington, I.: Computer Vision: A unified, biologically-inspired approach. Elsevier, Amsterdam (1992)