Multigrid Analysis of Curvature Estimators

Simon Hermann
Angewandte Informatik, Göttingen University,
D-37083 Göttingen, Germany
and Reinhard Klette
CITR, University of Auckland, Tamaki Campus, Building 731,
Auckland, New Zealand

Abstract

This article explains a new method for the estimation of curvature of plane curves and compares it with a method which has been presented in [2]. Both methods are based on global approximations of tangents by digital straight line segments. Experimental studies show that a replacement of global by local approximation results in errors which, in contrast to the global approximation, converge to constants > 0. We also apply the new global method for curvature estimation of curves to surface curvature estimation, and discuss a method for estimating mean curvature of surfaces which is based on Meusnier's theorem.

Keywords: curvature, digital curves, digital surfaces, multigrid analysis

1 Introduction

Curvature estimation is of interest in the context of 2D and 3D image analysis [4], e.g. for the determination of high curvature points on digital curves ("corners") or surfaces ("landmarks"). Curvature estimation has been widely studied for 2D applications, but research on surface curvature is still at its beginning [5]. Corner detectors (e.g. early methods from around 1970-1980), designed for high curvature point detection on plane curves, also provide values which can be used as a measure of curvature. Typically these methods are based on analyzing changes of angles between neighboring points. See [6, 7, 9] for surveys and comparisons. Let p and q be points on a plane curve and δ the angle between their (positively oriented) tangents. The curvature κ at p is defined as the limit of the ratio of δ and the arclength $|pq|$ for $|pq| \to 0$,

$$\kappa(p) = \lim_{pq \to 0} \frac{\delta}{|pq|} \ . \quad (1)$$

In this article we consider curvature as a positive quantity $|\kappa(p)|$. The radius $r(p)$ of the osculating circle at p is the reciprocal value of the curvature, $r(p) = \frac{1}{\kappa(p)}$. Figure 1 illustrates a way for estimating curvature.

Let ρ be an 8-curve in $\mathbb{Z}^2$. Tangents are approximated by DSS (digital straight segment) approximation; see [8] for a review on digital straightness. We apply the linear on-line DSS recognition algorithm **DR1995** which has been proposed in [3]. This algorithm is based on arithmetic geometry.

Let $\mu, \omega, a, b \in \mathbb{Z}$, a, b relatively prime and non-zero, and $\omega = \max\{a, b\}$. The set

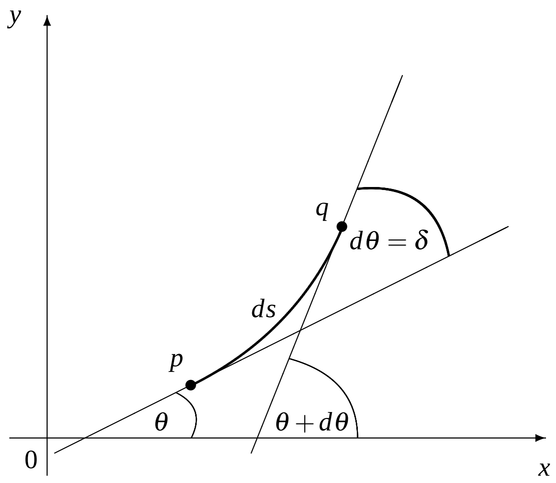

Figure 1: Assume a point q "close" to p: $\tilde{\kappa}(p) = \left|\frac{\delta}{ds}\right|$ provides a curvature estimate.

$$\rho = \{(a,b) : a,b \in \mathbb{Z} \wedge \mu \le ax - by < \mu + \omega\}$$

is an 8-DSL (digital straight line). Let $p, q \in \mathbb{Z}^2$ be start and end point of a DSS $[p,q]$, and let φ be a grid point closest to the center point of the real line segment pq. We define the coordinates of vector $\vec{v} \in \mathbb{Z}^2$ as

$$\vec{v} = (|p_x - q_x|, |p_y - q_y|) = (\delta x, \delta y) \ .$$

An estimated tangent τ on φ is now uniquely defined by the angle θ between vector $\vec{v}$ and the x-axis. We have

$$\tan(\theta) = \frac{\delta x}{\delta y}, \text{ i.e. } \theta = \arctan\left(\frac{\delta x}{\delta y}\right)$$

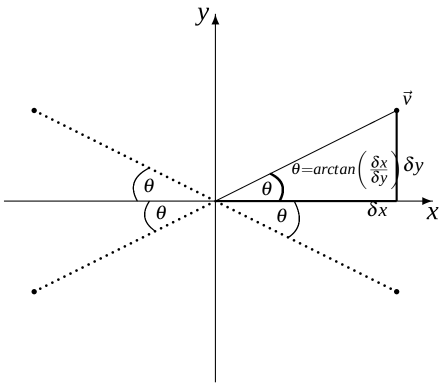

Figure 2: Tangent computation.

is the angle of the tangent with the x-axis [we use the radian measure throughout this article]. Note that it makes no difference, taking the positive values for δx and δy. In every case we have the same θ (see Fig. 2). From now on, an estimated tangent segment on, and centered at φ is defined by an angle θ, i.e. we have $\tau = \tau(\theta)$. Its length is $l(\tau) = \| \vec{v} \|_2$. We speak of a tangent segment τ at φ, or of a DSS $[p,q]$ starting at p.

2 Methods for Curvature Estimation

We present two curvature estimation methods, one based on approximating osculating circles, and one following the original definition in Eq. 1. Both methods use algorithm **DR1995**. The first method estimates curvature based on a single straight line at point $p \in \gamma$, and the second one uses two.

[2] proposes the computation of the maximum-length DSS $\tau(p) = [p,q]$ which begins at point p of γ. Let $\ell = \frac{l(\tau(p))}{2}$ and let

$$r_{inf} = \left\lceil \left(\ell - \frac{1}{2}\right)^2 - \frac{1}{4} \right\rceil \text{ and } r_{sup} = \left\lfloor \left(\ell + \frac{1}{2}\right)^2 - \frac{1}{4} \right\rfloor .$$

Return

$$\left(\frac{2}{r_{inf} + r_{sup}}\right)$$

Compute-curvature(Curve γ)
For point p in γ **do**
compute maximum-length DSS's $[q,p]$ and $[p,s]$
compute $\theta_b = arctan\left(\frac{|q.x - p.x|}{|q.y - p.y|}\right)$
compute $\theta_f = arctan\left(\frac{|p.x - s.x|}{|p.y - s.y|}\right)$
compute $\theta = \frac{1}{2} \cdot \theta_b + \frac{1}{2} \cdot \theta_f$
compute $\delta_b = |\theta_b - \theta|$ and $\delta_f = |\theta_f - \theta|$
return $\frac{1}{2} \cdot \frac{\delta_b}{d_2(q,p)} + \frac{1}{2} \cdot \frac{\delta_f}{d_2(p,s)}$

Figure 3: Method **HK2003** for curvature estimation.

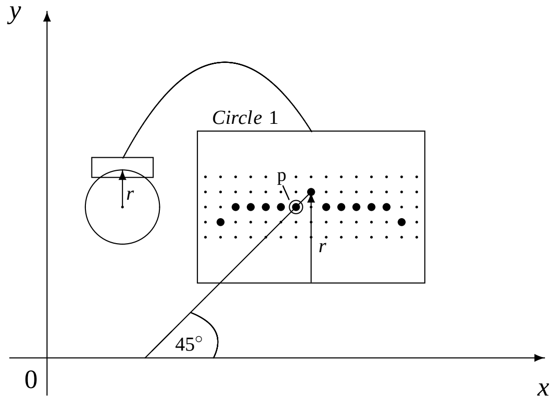

Figure 4: DSS approximation problem for disk 1.

as the estimated curvature $\tilde{\kappa}(p)$. This algorithm **CMT2001** uses only the length of the tangential segment, and not the angle.

The value returned by this function is actually the estimated curvature at midpoint φ of $[p,q]$, and not at p. It would be more appropriate to calculate the maximum-length DSS which is centered at p. For example, let p be the point with index i in γ. Start at point $i-1$, and stop with DSS recognition at point $i+1$. Then start at point $i-2$ and stop at $i+2$. Proceed as far as possible, which finally produces the longest DSS centered at p. However, the time complexity of this algorithm would be superlinear.

In the implemented version of **CMT2001** we process through γ in one direction, computing curvature estimates for midpoints φ of DSS's $[p,q]$. This way we estimate curvature at some points of γ, possibly not just once, and we may omit some of the points on γ.

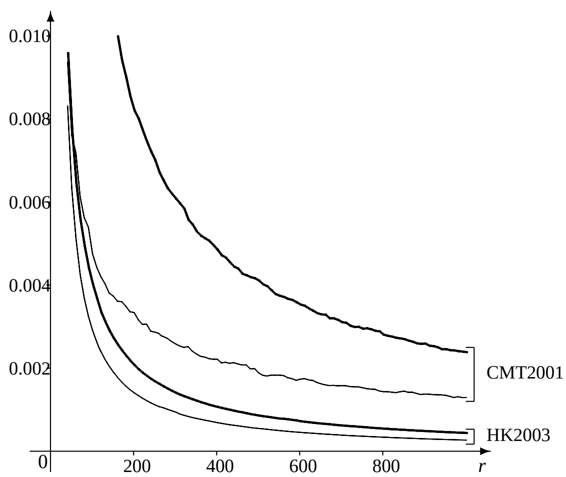

Figure 5: Mean errors of both methods, for both digital circles. The bold lines are for disk 1, and the thin lines for disk 2.

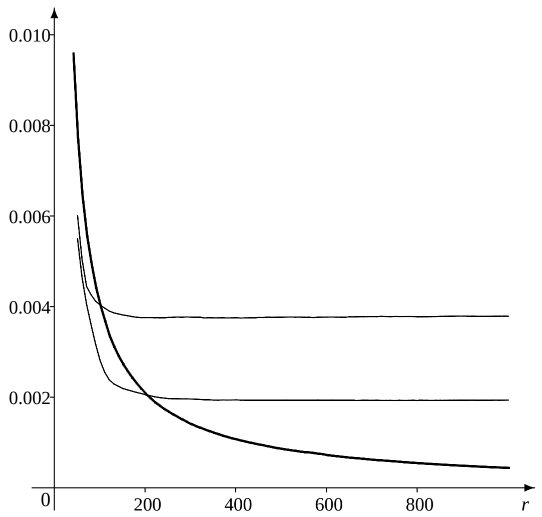

Figure 6: Mean errors of **HK2003** for disk 1: bold line for unlimited application, followed by lines for $k = 7$, and $k = 5$ (from bottom to top, on the right of the figure).

However, the labeled points on γ proved to be sufficient for our experiments.

We propose a new method (let us call it **HK2003**) which also approximates tangents by DSS's, but we assign two DSS's to each point p on γ, one forward DSS $[p,s]$ starting at p, and one backward DSS $[q,p]$ ending at p. Now we estimate the curvature for both midpoints φ_1 and φ_2 of these two DSS's, based on orientations of these DSS's (see Fig. 3). The mean of both values defines the estimated curvature at p.

3 Multigrid Analysis of Curvature Estimation

We tested both methods on two digital circles with about the same radius. Consider the real disks

- disk 1: $\mathbf{x^2 + y^2 \leq r^2}$
- disk 2: $\mathbf{x^2 + y^2 \leq r^2 + r}$

These disks are digitized (Gauss digitization). The 4-borders are traced by 8-curves, defining digital circles.

In our experiments we run r within the interval $[1, \ldots, 1040]$ and generate digital circles. We estimate the curvature for all grid points of these digital circles, and determine the mean and the maximum error. Let $\max_r$ be the maximum and mean_r the mean error of the curvature estimates for all grid points on these digital circles. The curves in Figs. 5 - 8 are drawn in steps of 10. The values are determined as sliding means as $\frac{1}{39}\sum_{i=-19}^{19} mean_{r+i}$ for the mean error, and $\frac{1}{79}\sum_{i=-39}^{39} max_{r+i}$ for the maximum error.

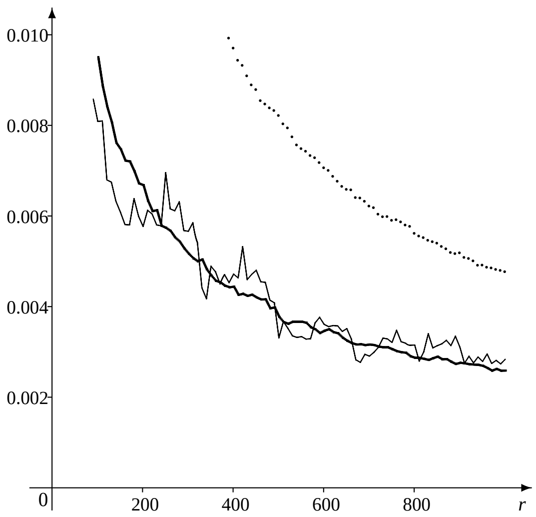

Figure 7: Multigrid analysis for **CMT2001**: mean error for disk 1 (dotted line), maximum error for disk 2 (thin line), and mean error for disk 2 (bold line).

Figure 5 shows that errors are bigger for the first disk than for the second. This effect is explained in Fig. 4: the Gauss digitization of disk 1 produces "single peaks"; if we process the digital circle clockwise then we obtain the worst approximation of tangents at the shown point p (with respect to length and angle, which is in this case 45°). The estimated curvature at such a "single peak" has a nearly constant error of ≈ 0.998 for **CMT2001**, and ≈ 0.1380 for **HK2003**. Such an error does not occur in disk 2.

Figure 6 shows errors of **CMT2001** applied to disk 1 for increasing resolution (bold line). We replaced the global tangent approximation in **CMT2001** by local

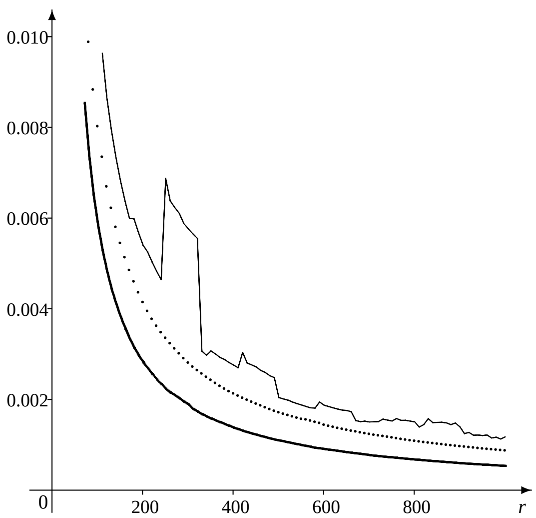

Figure 8: Multigrid analysis for **HK2003**: mean error for disk 1 (dotted line), maximum error for disk 2 (thin line), and mean error for disk 2 (bold line).

methods, where tangents are simply approximated by either 5 (upper thin curve) or 7 consecutive grid points on the digital circle. These local approximations work fine for low resolution digital circles.

Figures 7 and 8 show results of both methods separately for a larger error scale. Note, that the maximum error in Fig. 7 is approximately ten times larger than the mean error. **HK2003** produces the more accurate results, paid by about double computing time compared to **CMT2001**.

4 Curvature Estimation for Surfaces

Assume a surface which is cut by normal slices incident with surface point M, and where such slices intersect the surface in curves with defined curvature at M. (Slices are planes, defining curves on a surface by intersection. Normal slices are slices which contain the surface normal. The surface normal at M is defined as the cross product of two tangents of two curves, defined by any two slices, which have a point of intersection at M .)

Let r_1 and r_2 be the radii of the osculating circles of those curves, which are defined by normal slices C_1 and C_2 producing the maximum and minimum curvature value at M; see Fig. 9 on the left.

The curves are orthogonal and have their point of intersection in M. The mean curvature of a surface at point M is

$$H = \frac{1}{2}\left(\frac{1}{r_1} + \frac{1}{r_2}\right) ,$$

and the Gaussian curvature of a surface at point M is

$$K = \frac{1}{r_1 r_2} .$$

The main normal at a point of plane curves is orthogonal to the corresponding tangent. Meusnier's theorem states: Between the radius r of the osculating circle of a

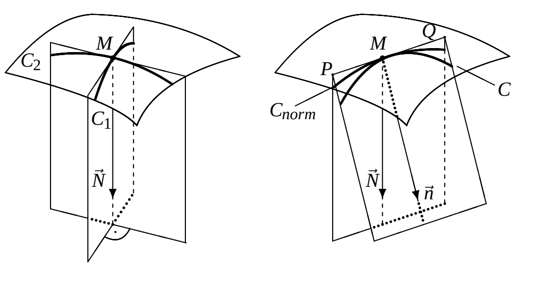

Figure 9: Left: a surface is cut by two normal slices, forming an angle of 90° degrees. Right: A surface is cut by one normal slice and another slice, both intersecting in a straight line which is incident with M.

plane slice C and the radius R of the osculating circle of a normal slice C_{norm}, where both slices have the same tangent PQ in M, exists the relation

$$\rho = R \cdot cos(\vec{n}, \vec{N}) ,$$

where $\vec{N}$ is the unit surface normal and $(\vec{n}, \vec{N})$ the angle between the unit vector $\vec{n}$ of the main normal of curve C and $\vec{N}$. ; see Fig. 9 on the right. The Curvature $\frac{1}{R}$ is called the normal curvature of C at M.

We intersect a surface at point M by exactly two slice contours. This produces two curves "around M". We can compute the normal curvature of these curves by applying Meusnier's theorem. For both slice contours $SlC_{i,b}$ which are crossing the surface face s, we define for $k \in \{1,2,3\}$ the unit tangent t with its coordinates $t_k = 0$ if $k = i$, $cos(\theta)$ for the remaining coordinate with the smaller index and $sin(\theta)$ for the last one. Algorithm 2 explains how to compute θ. For a slice contour $SlC_{2,b}$, we have the coordinates $(cos(\theta), 0, sin(\theta))$. By rotation of these vectors in the mathematical positive direction we get the main normals of both curves. In the example it would result in $(-sin(\theta), 0, cos(\theta))$. The surface normal of unit length can be computed by a vector cross product of the two tangents. Now we can apply Meusnier's theorem by using the new method for the estimation of the curvature of the curves defined by the plane Slices and computing the cosine of $(\vec{N}, \vec{n})$ by the inner product of the two unit vectors $\vec{n}$ and $\vec{N}$. The mean value of both curvatures is used as an approximation for the mean curvature of the surface!

Note that the directions of the computed normals do not correspond with the directions of the real surface normal of the object! But this is not necessary, since we are only interested in the angle between the surface normal and the main normal.

5 Multigrid Analysis of Surface Curvature Estimation

We generate Gauss digitizations of spheres $x^2 + y^2 + z^2 \leq r^2 + r$ and apply the described "3D extension" of a plane curve curvature estimator. Figure 10 compares results for $r \in [1, \ldots, 100]$, obtained for digital circles (in the plane) with those obtained for digital spheres. The Artzy-Herman surface tracking algorithm [1] is applied for tracing all frontier faces of the 6-connected 3D digital ball. Each surface face f is non-parallel to two coordinate planes. These planes are supporting planes for two slices defining intersecting digital curves γ_1 and γ_2 on the surface, where both intersect at f. The tangents of these curves allow to estimate curve curvature (we use **HK2003**) and the tangential plane, i.e. the surface normal. Meusnier's theorem allows to estimate the normal curvatures $\tilde{\kappa}_1$ and $\tilde{\kappa}_2$ for these two curves. Based on the assumption that γ_1 and γ_2 are about orthogonal cuts of the surface we can estimate

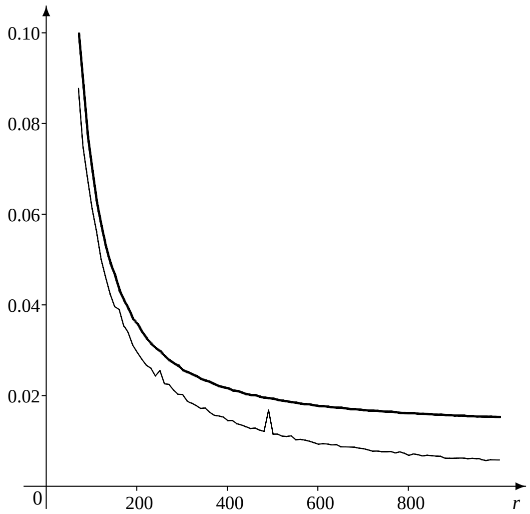

Figure 10: Comparison of **HK2003** applied to digital circles (in the plane) and digital spheres: thin line for circles, and bold line for spheres.

the mean curvature at f by taking the mean of $\tilde{\kappa}_1$ and $\tilde{\kappa}_2$.

Figure 10 shows calculated values without applying any sliding mean. The error for digital circles (for corresponding r-values) decreases faster than the error for digital spheres. This has a simple explanation: since we are slicing the sphere parallel to coordinate planes we obtain

digital circles γ_1 and γ_2 which have a smaller radius than $\sqrt{r^2+r}$, and thus produce a bigger error for curvature estimation..

6 Conclusions

In this article we presented a new global method for the exact estimation of curvature along digital curves. We showed in experimental studies that the error converges to zero for increasing grid-resolution and that there is only a minor difference (small variance) between the mean and the maximum error. We replaced the global tangent approximation by local approximations and showed by experiment that local methods do not converge to the correct curvature value.

Meusnier's theorem allows to apply curve curvature estimators for surface curvature estimation. First results show the feasibility of this approach, and studies on surfaces of "higher complexity" should follow.

References

[1] E. Artzy, G. Frieder, and G.T. Herman. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Vision, Graphics, Image Processing,* **15**:1–24, 1981.

[2] D. Coeurjolly, S. Miguet and L. Tougne. Discrete curvature based on osculation circle estimation. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Visual Form 2001*, pages 303–312. Springer, Berlin, 2001.

[3] I. Debled-Rennesson and J.-P. Reveillés. A linear algorithm for segmentation of digital curves. *Intl. J. Pattern Recognition Artificial Intelligence*, **9**:635–662, 1995.

[4] R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modelling, Morgan Kaufmann Publishers, San Francisco, 2004 (ca. 650 pages, to appear).

[5] A. Lenoir. Fast estimation of mean curvature on the surface of a 3D discrete object. *ISMRA*, **6**:303–312, 2001.

[6] H.-C. Liu and M.D. Srinath. Corner detection from chain code. *Pattern Recognition*, **23**:51–68, 1990.

[7] M. Marji. On the detection of dominant points on digital planar curves. Ph.D. thesis, Wayne State University, Detroit, 2003.

[8] A. Rosenfeld and R. Klette. Digital straightness. In *Electronic Notes in Theoretical Computer Science* **46**, 2001.

[9] M. Worring and A.W.M. Smeulders. Digital curvature estimation. Computer Vision, Graphics, Image Processing, **58**:366–382, 1993.