

Curvature and Bending Energy in Digitized 2D and 3D Images

Lucas J. van Vliet and Piet W. Verbeek

Pattern Recognition Group Delft, Faculty of Applied Physics, Delft University of Technology,
Lorentzweg 1, 2628 CJ Delft, The Netherlands

Abstract

Existing curvature estimators of planar curves are applied to a binary representation of the object. The parametric curve description is 1D-smoothed to overcome quantization errors. In this paper we estimate object curvature directly from a properly sampled grey-scale image using 2D isotropic derivative-of-Gaussian filters. Three times oversampling or a Gaussian σ_K of 2.7 yields sampling-error free results. The algorithm was extended to find the principal curvatures of iso-grey surface patches in 3D. The Gaussian and mean curvatures can easily be computed from the principal curvatures.

Integrated curvature and bending energy of a closed object in 2D or 3D is frequently used as shape discriminator. The binary methods sum the estimated curvature/energy values over a chain-code description of the contour (as in binary length estimators). We estimate bending energy through grey-volume measurement. Volume is measured without thresholding and does not introduce a sampling error. Edges are transformed into volumes by giving them a constant height after which they are shifted perpendicular to the edge over a small distance. Subtraction of the two images shifted in opposite direction produces a volume that is proportional to the edge length [1, 2]. This volume is weighted using the obtained energy values. This method produces very precise ($CV < 0.01\%$) and accurate measures.

1. Introduction

An important contour descriptor in arbitrary dimension is the curvature. It can be used to characterize 2D contour points as convex, concave and flat/inflection points. Peak finding in curvature values along image contours is used for corner detection [3] and detection of significant concavities in several automated segmentation methods. Points of

an iso-grey surface patch in 3D can be classified into elliptic, hyperbolic, parabolic and flat points.

Worring and Smeulders [4] compared several curvature estimation methods applied to a binary representation of a curve. The measured bias is inversely proportional to the smoothing of the contour which is embedded in the analysis technique. Their resampling method performed best. The maximum bias decreases from 10% to 1% when the Gaussian smoothing increases from $\sigma=3$ to $\sigma=16$.

Stokely and Wu [5] compared five methods for curvature measurement applied to 3D binary spheres. The errors decrease when the surface patch considered increases. Their surface triangulation method performed best. The error decreases from 100% to 1% when the patch size increases from 75 voxels to 600 voxels.

We believe that in the thresholding that usually produces the binary image valuable information is lost that should be preserved (and used) to improve the curvature estimation. The method presented in this paper works on grey value images and is founded on sampling theory. Apart from this method Barman [6] estimates curvature without thresholding. He uses quadrature filters applied to a vector field image to estimate direction, curve/straight ratio's, and curvature.

Young [7] measured the bending energy of a 2D object contour for the characterization of biological objects. This measure quantifies the energy stored in the shape of the contour. A circle minimizes this energy for all simply-connected closed curves of equal length. His method sums the energy contributions per chain-code element. Duncan [8] applied the same model for measurement of cardiac shape deformity. Our new method combines recent work on estimators for edge length in 2D and surface area in 3D [1, 2] with curvature measurements to calculate the energy stored in the shape of the object. Apart from a few

plausible assumptions it is founded on sampling theory.

2. Theory

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 .

$$\kappa = \frac{d\theta}{ds} \quad (1)$$

For analytical functions $y(x)$ curvature is given by

$$\kappa(x) = \frac{|y''(x)|}{\left(1 + (y'(x))^2\right)^{3/2}} \quad (2)$$

The curves (or edges) that occur in digitized images can not be described using analytical functions. A widely accepted method uses a parametric description of the curve to be analyzed. The spatial coordinates are given as function of a position parameter t ($x(t)$, $y(t)$). Using

$$y'(x) = \frac{dy(x)}{dx}$$

gives the following equation for the curvature at point t

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{\left(\dot{x}(t)^2 + \dot{y}(t)^2\right)^{3/2}} \quad (3)$$

with

$$\dot{x}(t) = \frac{dx}{dt}$$

To regularize the differentiation process and allow the construction of curvature scale spaces [3, 9, 10] the straight differentiation is replaced by the convolution with the derivative of a Gaussian [11].

$$\dot{x}(t) \Rightarrow x(t) * \dot{G}(t, \sigma) \quad (4)$$

Differentiation with respect to s produces better results than obtained by differentiation with respect to t . The above formulas require a list of x - and y -coordinates. These are usually obtained by thresholding or edge detection after which an 8-connected contour is obtained. To prevent aliasing we propose to measure the object curvature directly from a properly sampled grey-value image. Verbeek [12] presented the idea to measure the curvature of an isophote. In this paper we examine the performance of this technique and extend it to 3D.

3. Isophote curvature in 2D

Let $f(x, y)$ be a grey-value image and f_x and f_y respectively the derivatives in the x and y -direction. Following the same argument as in the previous section, the derivatives are computed by a convolution with the derivative of a 2D Gaussian.

At any point (x, y) in the image we have a gradient vector \mathbf{g} , a contour vector \mathbf{c} (in isophote direction) and a contour direction θ .

$$\mathbf{g} = (f_x, f_y) \quad (5)$$

$$\mathbf{c} = (-f_y, f_x) \quad (6)$$

$$\theta = \arccos(-f_y / \|\mathbf{g}\|) = \arcsin(f_x / \|\mathbf{g}\|) = \arctan(-f_x / f_y) \quad (7)$$

with

$$\|\mathbf{g}\| = \sqrt{f_x^2 + f_y^2}$$

To differentiate along the curve with respect to the arc length we use the operator

$$\frac{d}{ds} = \cos\theta \frac{\partial}{\partial x} + \sin\theta \frac{\partial}{\partial y} = \frac{-f_y}{\|\mathbf{g}\|} \frac{\partial}{\partial x} + \frac{f_x}{\|\mathbf{g}\|} \frac{\partial}{\partial y} \quad (8)$$

Applying eq. (8) to eq. (7) we get

$$\kappa = \frac{d\theta}{ds} = \frac{-(f_{xx}f_y^2 - 2f_xf_yf_{xy} + f_{yy}f_x^2)}{(f_x^2 + f_y^2)^{3/2}} \quad (9)$$

Note that the following relationship exists between the curvature κ and the second derivative in the contour direction ($f_{cc} \equiv \text{SDCD}$).

$$\kappa = \frac{d\theta}{ds} = \frac{-f_{cc}}{\|\mathbf{g}\|} = \frac{-\text{SDCD}}{\|\mathbf{g}\|} \quad (10)$$

3.1. Sampling requirements

The discrete versions of all derivative filters produce a sampling-error-free result when applied to a properly sampled image. The numerator of eq. (9) does not exhibit aliasing if the image $f(x, y)$ is sampled at three times the Nyquist rate. Alternatively, we can reduce the bandwidth of a critical sampled image by choosing a sufficiently large σ_κ in the discrete Gaussian derivatives ($\sigma_\kappa \approx 2.7$) [13, 14].

4. Isophote curvatures in 3D

Curvature in any dimension is defined along a line. Considering an iso-grey surface patch dS centered around the point (x, y, z) we are interested in the principal curvatures κ_1 and κ_2 ($\kappa_1 \geq \kappa_2$). The corresponding surface vectors are \mathbf{c}_1 and \mathbf{c}_2 . They

share the same gradient vector \mathbf{g} that is aligned with the surface normal vector. Given the principal curvatures one can compute well-known surface descriptors such as the Gaussian curvature ($\kappa_1 \kappa_2$) and the mean curvature ($\frac{1}{2}(\kappa_1 + \kappa_2)$).

4.1. Principal curvatures of an isophote surface patch

This section extends the 2D isophote curvature to three dimensions. A three step algorithm is given below.

step 1: Calculate the gradient vector and the Hessian matrix

$$\mathbf{g} = (f_x, f_y, f_z) \quad (11)$$

and the Hessian matrix

$$\mathbf{H} = \begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{xy} & f_{yy} & f_{yz} \\ f_{xz} & f_{yz} & f_{zz} \end{pmatrix} \quad (12)$$

with

$$f_{xy} = \frac{\partial f(x, y, z)}{\partial x \partial y} \Rightarrow f(x, y, z) * G_{xy}(\sigma_\kappa)$$

step 2: Rotate the Hessian to align the first axis with the gradient. The resulting matrix \mathbf{H}' can be written as

$$\mathbf{H}' = \begin{pmatrix} f_{gg} & & \\ & f_{uu} & f_{uv} \\ & f_{uv} & f_{vv} \end{pmatrix} = \begin{pmatrix} f_{gg} & & \\ & & \\ & & \mathbf{H}'_t \end{pmatrix} \quad (13)$$

with f_{gg} the second derivative in gradient direction (SDGD) and \mathbf{H}'_t a 2D Hessian in the touching plane \mathbf{T} perpendicular to \mathbf{g} .

step 3: Compute the eigenvalues of \mathbf{H}'_t . These eigenvalues λ_1 and λ_2 are respectively the maximum and the minimum second derivatives in the plane \mathbf{T} . The eigenvectors are the surface vectors \mathbf{c}_1 and \mathbf{c}_2 . Intersecting the surface patch with the planes $(\mathbf{c}_1, \mathbf{g})$ and $(\mathbf{c}_2, \mathbf{g})$ respectively shows that as in 2D there exists a relationship between the second derivatives in contour direction and the corresponding curvatures.

$$\kappa_1 = \frac{-\lambda_1}{\|\mathbf{g}\|} = \frac{-\text{SDC}_1 \text{D}}{\|\mathbf{g}\|} \quad (14a)$$

$$\kappa_2 = \frac{-\lambda_2}{\|\mathbf{g}\|} = \frac{-\text{SDC}_2 \text{D}}{\|\mathbf{g}\|} \quad (14b)$$

The eigenvalue analysis is computed using the Jacobi technique [15]. The speed of this algorithm can be increased when the Hessian and subsequent

eigenvalue analysis is only applied to those voxels where the gradient exceeds a certain threshold.

5. Bending energy

The bending energy of an object denotes the energy stored in its shape. Using Elasticity theory [16] one can show that the 2D bending energy [7] is directly proportional to the bending energy of a deformed circular rod. Similarly, we will show the relation between our 3D bending energy and the bending energy of a deflected thin plate. In 2D as well as 3D, we have only considered isotropic bodies whose deformations follow Hook's law to a good approximation. That is, the deformation is proportional to the applied forces.

5.1. Bending energy of a circular rod

A bent rod is stretched at some points (convex side) and compressed at others (concave side). There exists, however, a neutral surface (or line) which undergoes neither extension nor compression. The length of the neutral line is unaffected by bending. Large bending deflections will cause a deformation that is a combination of pure bending and torsion. For small deflections we can assume that the bending occurs in a single plane. The deviation of a slightly bent curve from a plane (its torsion) is at least one order of a magnitude smaller than its curvature.

The bending energy per unit length of the rod is

$$\frac{1}{2} EI (\mathbf{r}_{ll})^2 \quad (15)$$

with E Young's modulus, I the moment of inertia of the circular cross section, and \mathbf{r}_{ll} is the second derivative of the position vector \mathbf{r} of a point on the curve with respect to its length. The total bending energy over a rod of length L is therefore

$$\frac{1}{2} EI \int_0^L (\mathbf{r}_{ll})^2 dl \quad (16)$$

For 2D objects, the integrated squared curvature along its contour is directly proportional to the bending energy of a deformed circular rod.

$$BE_{2D} = \int_{\text{contour}} \kappa^2 ds \quad (17)$$

Young [7] proved that the simply-connected, closed contour with the minimum average bending energy is the circle.

5.2. Bending energy of a thin plate

Unlike a bent rod, it is impossible to bend a plate without stretching. An exception is the deformation of a flat plate into a cylindrical surface. Closed surfaces

can never be deformed without stretching. Here we only consider the bending component of the total energy. The bending energy per unit volume of the plate is given by [16]

$$z^2 \frac{E}{1+p} \left\{ \frac{1}{2(1-p)} (\zeta_{xx} + \zeta_{yy})^2 + \left[(\zeta_{xy})^2 - \zeta_{xx}\zeta_{yy} \right] \right\} \quad (18)$$

with E Young's modulus, p Poisson's ratio, the xy -plane is that of the undeformed plate (neutral surface), the z -axis is normal to the surface, and ζ the displacement of a point on the neutral surface, i.e. its z coordinate. The total bending energy of a deformed plate of thickness h is

$$\frac{Eh^3}{24(1-p^2)} \iint_{\text{surface}} \left[(\zeta_{xx} + \zeta_{yy})^2 + 2(1-p) \left\{ (\zeta_{xy})^2 - \zeta_{xx}\zeta_{yy} \right\} \right] dx dy \quad (19)$$

Since we are interested only in the shape, and not in the distribution of deformations inside, we regard the plate as being of infinitesimal thickness.

By taking the x and y axes along the directions of principal curvatures u and v we get

$$\frac{Eh^3}{24(1-p^2)} \iint_{\text{surface}} (\zeta_{uu}^2 + \zeta_{vv}^2 + 2p\zeta_{uu}\zeta_{vv}) dudv \quad (20)$$

Note that $\zeta_{uu} \equiv \kappa_1$ and $\zeta_{vv} \equiv \kappa_2$. In practice, Poisson's ratio varies between 0 and $\frac{1}{2}$. A value close to $\frac{1}{2}$ corresponds to a medium (e.g. rubber) having a modulus of rigidity that is small compared to the modulus of compression.

For small values of Poisson's ratio we obtain an approximation of the bending energy of a deformed plate in which the integrand is independent of the medium.

$$\frac{Eh^3}{24} \iint_{\text{surface}} (\zeta_{uu}^2 + \zeta_{vv}^2) dudv \quad (21)$$

If Poisson's ratio is close to $\frac{1}{2}$ we get for the bending energy

$$\frac{Eh^3}{18} \iint_{\text{surface}} (\zeta_{uu}^2 + \zeta_{vv}^2 + \zeta_{uu}\zeta_{vv}) dudv \quad (22)$$

After substitution of $\zeta_{uu} \equiv \kappa_1$ and $\zeta_{vv} \equiv \kappa_2$, the bending energy per unit surface is directly proportional to a function of the principal curvatures.

$$BE_{3D} = \int_{\text{surface}} (\kappa_1^2 + \kappa_2^2 + 2p\kappa_1\kappa_2) dS \quad (23)$$

The above theory applies to deformations of flat plates, i.e. the undeformed state. Deformations of shells have properties which are fundamentally different from those of the deformations of flat plates. For shells, stretching is a first order effect and thus is

important even for small bending deflections. Here, we are only interested in bending energy. We can think of our closed surfaces composed of a set of small rectangular surface patches satisfying the above requirement. The sum of energy contributions for each patch gives the energy stored in the entire surface.

For a family of closed 3D objects a sphere is the shape of minimal energy. Note that the bending energy in 3D is dimensionless and therefore scaling invariant, e.g. a sphere of radius R has the same bending energy as a sphere of radius aR ($a > 0$). The proposed bending energy of eq. (23) with Poisson's ratio $p=0$ has already been proposed as a fairness criterion in geometric surface modelling [17, 18].

5.3. Measurement of bending energy

Existing binary methods in 2D [7, 8] sum the estimated energy contributions (κ^2) over a chain-code description of the contour (as in binary length estimators).

We estimate bending energy through grey-volume measurement. Volume is measured without thresholding and does not introduce a sampling error. For a properly sampled signal, the sum of the samples is directly proportional to the volume under the grey-scale landscape. When skipping the curvature terms in the integrands of eqs. (17,23), these eqs. reduce respectively to a 2D edge length estimator and a 3D surface area estimator. In earlier work [2] we developed a 2D edge length / 3D surface area estimator based on volume measurements. We developed a technique (called the GCL technique) to transform signals with edges into signals with a volume proportional to the edge length (surface area). Edges are transformed into volumes by giving them a constant height after which they are shifted perpendicular to the edge over a small distance. Subtraction of the two images shifted in opposite direction produces a volume that is proportional to the edge length [2]. This technique can also measure bending energy when the each point of the signal is weighted by the bending energy contribution at that point before integration.

In practice, the GCL technique [2] yields unbiased edge length measurements for an object in image f

$$\text{length} = \iint \text{GCL}(f) dx dy \quad (24)$$

This estimator for 2D bending energy is

$$BE_{2D} = \iint \kappa^2(x,y) \text{GCL}(f) dx dy \quad (25)$$

and in 3D we get

$$BE_{3D} = \iiint \left(\kappa_1^2(x,y) + \kappa_2^2(x,y) + 2p\kappa_1(x,y)\kappa_2(x,y) \right) GCL(f) dx dy dz \quad (26)$$

In the remainder of this paper we will set Poisson's ratio equal to zero ($p=0$).

6. Experiments

We have performed experiments to test the theory presented here. First we develop a recipe to construct bandlimited test objects. In order to compare our results with [4, 5], we give the bias and CV for the measured curvature of individual pixels/voxels.

6.1. Test Images

The test images contain a simulated image of a step edge object imaged through an optical system and sampled at the Nyquist frequency (1N) or at twice the Nyquist frequency (2N). Randomly positioned bandlimited discs/ellipses/spheres/ellipsoids are constructed as test object. To construct an arbitrary bandlimited object, we start from its Fourier transform, ensure proper bandlimitation by multiplying with the perfect in-focus OTF (optical transfer function) [19, 20], and apply an inverse Fourier transform to obtain the desired image. In 3D the OTF is replaced by a Gaussian function that shows behavior comparable to the corresponding OTF. In earlier work we found that the 2D point spread function of an in-focus optical system is well described by a 2D Gaussian; $\sigma_{PSF}=0.9$ [13, 14].

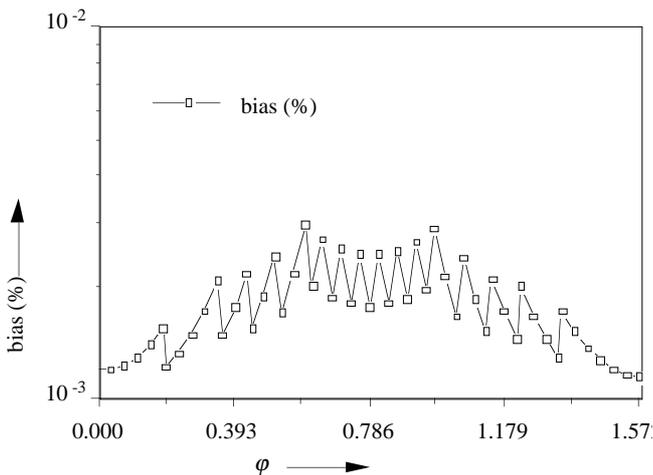


Figure 1: Curvature bias as function of φ for a disc of radius 25. The disc was sampled at three times the Nyquist rate; curvature calculation was done with $\sigma_\kappa=1.5$; and only pixels close to the true contour are taken into account. Note that interpolation of the result image is allowed to find the curvature at sub-pixel resolution. In the edge region the CV was smaller than 0.001 %.

6.2. Isophote curvature in 2D

As in [4] we used a disc of radius 25. Our image, however, is sampled at three times the Nyquist frequency. We studied the bias as a function of the orientation. In contrast to the binary methods we would not expect a dependency between the bias and the local orientation of the curve. However, repeated measurements show that an extremely small bias differs with a small dynamic range (cf. figure 1). It is smallest when the curve is aligned with the sampling grid. It is largest at $3\pi/8$. The CV is smaller than 0.001 %. The radial error is the same as for the measured principal curvatures in 3D.

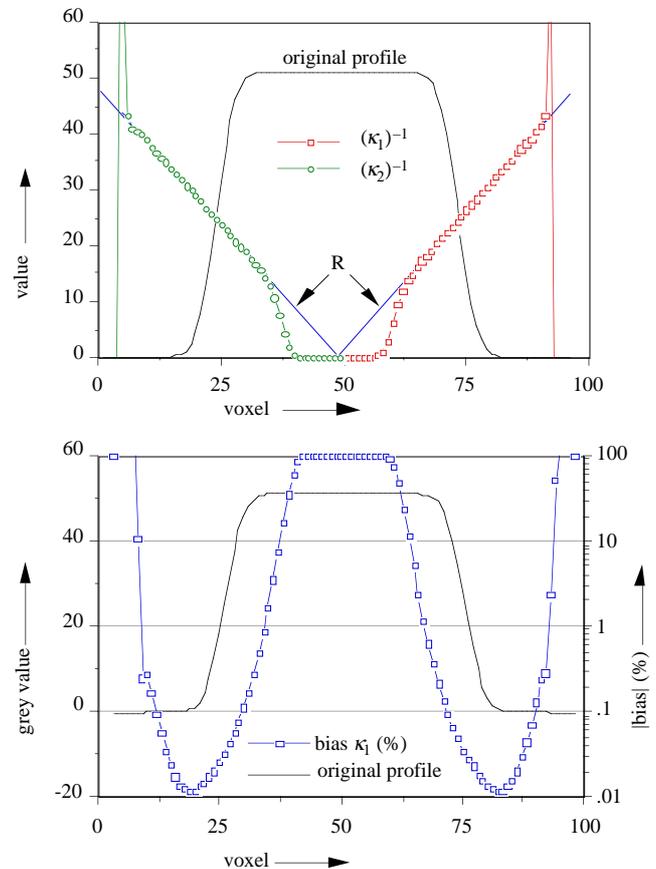


Figure 2: Cross-sections through the center of a sphere sampled at three times the Nyquist rate. **a)** Original grey-scale profile, prediction of κ^{-1} (for a sphere $\kappa_1=\kappa_2=r^{-1}$), at the right side the measured κ_1^{-1} and at the left side the measured κ_2^{-1} . **b)** Original grey-scale profile and the absolute bias as function of the distance to the center of the object. It is clear that in the edge region the error becomes very small ($<0.1\%$). In the edge region the CV was smaller than 0.01 %.

6.3. Principal curvatures of iso-grey surface patch in 3D

The non-isotropy of the bias measured over a sphere with radius 25 is again a factor of 2. The minimum error occurs when the surface normal is aligned with one of the axes of the sampling grid and the maximum occurs around the surface point with surface normal in the direction of the voxel diagonal. The radial error for κ_1 and κ_2 are similar and displayed in figure 2. From figure 2 we may conclude that the curvature measurements in the edge region yield a bias smaller than 0.1 %. The CV's were smaller than 0.01 %.

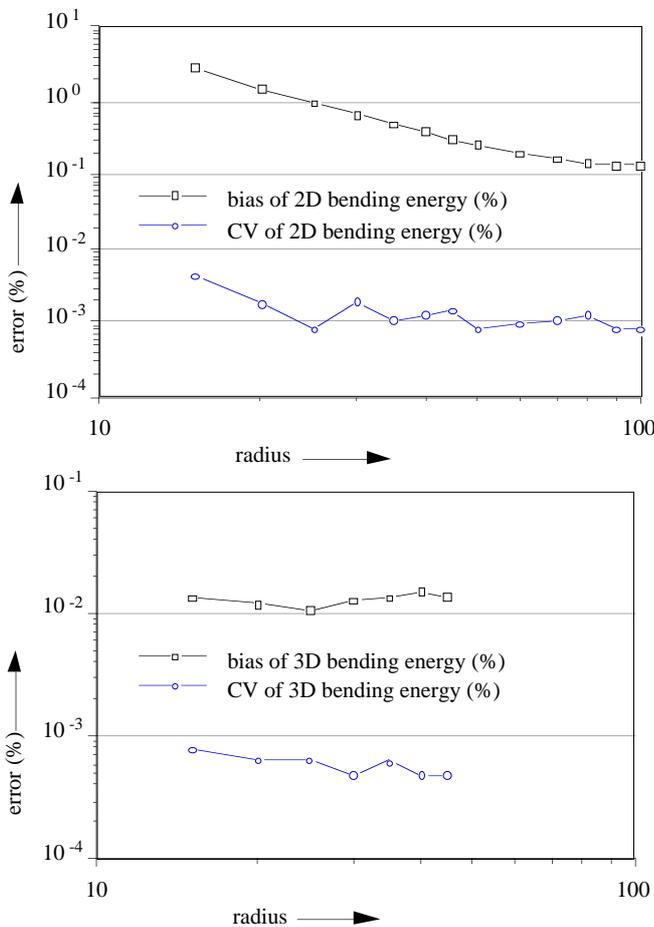


Figure 3: Bending energy measurements from 2D discs and 3D spheres sampled at twice the Nyquist rate. The curvature estimator used a $\sigma_\kappa=5.4$ to produce an aliasing free result. **a)** Errors in bending energy for 2D discs as function of the disc radius. For each size 10 randomly positioned discs were measured. **b)** Errors in bending energy for 3D spheres as function of the sphere radius. For each size 10 randomly positioned spheres were measured.

6.4. Bending energy

We have argued that three times oversampling is needed to avoid aliasing in the curvature measurement. Consequently, bending energy (κ^2) requires six times oversampling of the original image, or sampling near the Nyquist rate and a large σ_κ in the discrete Gaussian derivatives ($\sigma_\kappa=6 \times 0.9=5.4$). Due to the low energy contribution in the highest frequencies, a somewhat smaller discrete Gaussian ($\sigma_\kappa \approx 5.4 \times \sqrt{2} \approx 3.8$) will ensure enough bandlimitation to avoid serious aliasing effects. The discrete Gaussian filters have to be a few times (≈ 3) smaller than the smallest edge radius of the object to allow accurate curvature measurements.

Bending energy for 2D discs and 3D spheres were evaluated as a function of the object radius. The results are shown in figure 3. The CV's in 2D and 3D are about the same. The bias of the 3D energy measure is significantly smaller than that of its 2D counterpart.

We have also measured the bending energy of 2D ellipses. For each eccentricity, the a and b ($a > b$) values have to satisfy the minimum edge radius condition, $R_{min} > 3\sigma$ with $\sigma^2 = \sigma_{PSF}^2 + \sigma_\kappa^2$

$$(b^2/a) > 3\sigma$$

Table 1 shows the bending energy for ellipses with the same minimum edge radius. The relative error in the bending energy is mainly governed by error of the point with the highest curvature (smallest edge radius). Similar for 3D ellipsoids with principal axes a , b , and c ($a > b > c$). The minimum edge radius condition is

$$(c^2/a) > 3\sigma$$

Table 2 shows the 3D bending energy for ellipses. Notice that indeed a sphere has the lowest amount of free energy.

7. Conclusions

In this paper we measure the isophote curvature in 2D and 3D images. The theory is based on sampling theory and yields very small errors: bias $< 0.1\%$ and CV $< 0.01\%$. These errors are significantly smaller than those produced by binary measurement methods [4, 5]. The theory requires oversampling by a factor three, or a critically sampled image and a discrete Gaussian derivatives with $\sigma_\kappa=2.7$.

Using the estimated curvatures we can calculate the energy stored in the elastic rod or sheet that follows the shape of the object. Analogous to our edge

Table 1: Relative error (in %) of the bending energy for a set of ellipses having the same smallest edge radius of 16.2 pixels (3×5.4). The measurements are averaged over 50 ellipses of different position and orientation. The ellipses are sampled at twice the Nyquist rate, the σ_k is either 5.4 or 3.8. The CV's are on the order of 10^{-3} %. The theoretical values are calculated using the Mathematica script of appendix A.

a (Δ)	b (Δ)	b/a	theory		experiments: bias(BE)	
			L (Δ)	BE (Δ^{-1})	$\sigma_k=5.4$ (%)	$\sigma_k=3.8$ (%)
16.2000	16.2000	1.0	101.788	0.38785	2.2996	2.3460
20.0000	18.0000	0.9	119.463	0.33461	1.3428	1.5430
25.3125	20.2500	0.8	143.581	0.29060	0.1022	0.7285
33.0612	23.1428	0.7	177.948	0.25491	-1.2204	-0.0239
45.0000	27.0000	0.6	229.743	0.22661	-2.4111	-0.6946
64.8000	32.4000	0.5	313.906	0.20482	-3.4499	-1.2577
101.250	40.5000	0.4	466.016	0.18865	-4.5085	-2.0753
180.000	54.0000	0.3	789.464	0.17727	-4.4672	-1.1897

Table 2: Bending energy of ellipsoids that all have a smallest edge radius of 11.4 pixels (3×3.8). The measurements are averaged over 10 ellipsoids of different position and orientation. The ellipsoids were sampled at twice the Nyquist rate, the σ_k is set at 3.8. The CV's are on the order of 10^{-3} %. The theoretical values are calculated using the Mathematica script of appendix A.

scale (Δ)	a	b	c	theory		experiment	
				A (Δ^2)	BE	BE	bias (%)
11.4000	1.0	1.0	1.0	1633.13	25.1327	25.1200	-0.0504
14.0741	1.0	0.95	0.9	2245.63	25.3567	25.2983	-0.2302
14.0741	1.0	0.9	0.9	2167.12	25.4221	25.3484	-0.2898
17.8125	1.0	0.9	0.8	3224.30	26.1493	25.9621	-0.7157
17.8125	1.0	0.8	0.8	2986.36	26.3973	26.1852	-0.8035
23.2653	1.0	0.85	0.7	4894.35	27.7918	27.4698	-1.1585
23.2653	1.0	0.7	0.7	4317.98	28.2883	27.9515	-1.1905

length/surface area estimator, we accomplish this by measuring the volume of a grey value landscape. The volume produced by the GCL method [2] is directly proportional to the edge length of objects in 2D or the surface area of objects in 3D. A point-by-point multiplication of the GCL landscape with the energy contribution for the underlying pixel/voxels yields a volume proportional to the bending energy of the object present in the image.

The bending energy estimation of 2D ellipses improves with increasing edge radii. The bias depends on the point of the contour having the highest curvature (smallest edge radius). Ellipses with various eccentricities but equal minimum edge radius show that bending energy changes from a small overestimate (2%) for eccentricity = 1 to a small underestimate (-2%) for ellipses with an eccentricity near 3. The corresponding CV's are smaller than 0.01%.

The bending energy for spheres remains accurate over the entire range of edge radii. An experiment applied to 3D ellipsoids show that the bias increases from -0.05% for spheres to -1% for ellipsoids with 'eccentricities' near 1.5. The CV's are around 0.001%.

Our method can be applied using simple derivatives-of-Gaussian filters, some point operations

and a little help from Numerical Recipes in C [15] to do eigenvalue analysis.

The current method is sensitive to shading in the image. This can easily be eliminated by applying the curvature algorithm to the output of a second-derivative filter.

8. Acknowledgment

This work was partially supported by The Netherlands Foundation for Biomedical Research NWO-MEDIGON, the Dutch Government as part of the SPIN-FLAIR II program "Delft Intelligent Assembly Cell", and the SPIN-3D program for 3D biomedical image analysis.

Appendix A

The circumference of a 2D ellipse is given by an elliptical integral. In order to evaluate the proposed measures for bending energy we need to calculate the true bending energy of ellipses in 2D and ellipsoids in 3D. These values are obtained using numerical integration in Mathematica™ [21] based upon the theory of Differential Geometry. The function `EllipseLengthBE` computes the perimeter and bending energy of 2D ellipses. The function

EllipsoidAreaBE computes the surface area and bending energy (note that we have set Poisson's ratio equal to zero) of 3D ellipsoids. For readability purposes we have adopted the exact notation used by [22, 23].

```

EllipseLengthBE[a_, b_] :=
(* a, b: principal axes of 2D ellipse *)
Module[{
  t, (* curve parameter *)
  x, (* position vector x(t) *)
  xt, (* 1st derivatives of x to t *)
  xtt, (* 2nd derivatives of x to t *)
  d, (* dummy vector *)
  ds, (* length element *)
  K, (* curvature *)
  L, (* total length *)
  BE}, (* 2D bending energy *)
x = {a*Cos[t], b*Sin[t], 0};
xt = D[x,t]; xtt = D[x,{t,2}];
ds = Sqrt[xt.xt];
d = CrossProduct[xt,xtt];
K = Sqrt[d.d]/(xt.xt)^(3/2);
L = 4 * NIntegrate[ds, {t,0,Pi/2}];
BE = 4 * NIntegrate[K^2 * ds, {t,0,Pi/2}];
{L,BE}]

EllipsoidAreaBE[a_, b_, c_] :=
(* a,b,c: principal axes of 3D ellipsoid *)
Module[{
  u, (* surface parameter 1 *)
  v, (* surface parameter 2 *)
  x, (* position vector x(u,v) *)
  xu, xv, (* 1st derivatives of x *)
  xuu, xuv, xvv, (* 2nd derivatives of x *)
  E, F, G, (* first fundamental form *)
  dA, (* area element *)
  n, (* surface normal vector *)
  L, M, N, (* second fundamental form *)
  Km, (* mean curvature *)
  Kg, (* Gaussian curvature *)
  A, (* total area *)
  BE}, (* 3D bending energy *)
x={a*Sin[v]Cos[u],b*Sin[v]Sin[u],c*Cos[v]};
xu = D[x,u];
xv = D[x,v];
xuu = D[x,{u,2}];
xuv = D[x,{u,v}];
xvv = D[x,{v,2}];

E = xu.xu; F = xu.xv; G = xv.xv;
dA = Sqrt[E*G - F*F];
n = CrossProduct[xu,xv]/dA;
L = xuu.n; M = xuv.n; N = xvv.n;

Km = (E*N - 2*F*M + G*L)/(2(E*G - F*F));
Kg = (L*N - M*M)/(E*G - F*F);
A = 8*NIntegrate[dA,{u,0,Pi/2},{v,0,Pi/2}];
BE = 8*NIntegrate[(4*Km^2 - 2*Kg)*dA,
{u,0,Pi/2},{v,0,Pi/2}];
{A,BE}]

```

9. Literature

- [1] P.W. Verbeek and L.J. van Vliet, "An estimator of edge length and surface area in digitized 2D and 3D images," 11th ICPR, pp. 1-5, 1992.
- [2] P.W. Verbeek and L.J. van Vliet, "Estimators of 2D edge length and position, 3D surface area and position in sampled grey images," *BioImaging*, **1**, in press, 1993.
- [3] A. Rattarangsi and R.T. Chin, "Scale-space detection of corners of planar curves," *PAMI-14*(4), pp. 430-449, 1992.
- [4] M. Worring and A.W.M. Smeulders, "The accuracy and precision of curvature estimation methods," 11th ICPR, pp. 139-142, 1992.
- [5] E.M. Stokely and S.Y. Wu, "Surface parametrization and curvature measurement of arbitrary 3D objects: five practical methods," *PAMI-14*(8), pp.833-840, 1992.
- [6] H. Barman, "Hierarchical Curvature Estimation in Computer Vision," Ph.D. thesis no. 253, Linköping University, Sweden, 1991.
- [7] I.T. Young, J.E. Walker, and J.E. Bowie, "An analysis technique for biological shape I," *Info Control*, **25**, pp. 357-370, 1974.
- [8] J.S. Duncan, F.A. Lee, A.W.M. Smeulders, and B.L. Zaret, "A bending energy model for measurement of cardiac shape deformity," *PAMI-10*(3), pp. 307-319, 1991.
- [9] J.J. Koenderink and W. Richards, "Two-dimensional curvature operators," *Journal of the Optical Society of America A*, **5**(7), pp. 1136-1141, 1988.
- [10] F. Mokhtarian and A.K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *PAMI-14*(8), pp. 789-805, 1992.
- [11] D.M. Wuescher and K.L. Boyer, "Robust contour decomposition using a constant curvature criterion," *PAMI-13*(1), pp. 41-51, 1991.
- [12] P.W. Verbeek, "A class of sampling-error free measures in oversampled band-limited images," *Pattern Recognition Letters*, **3**, pp. 287-292, 1985.
- [13] P.W. Verbeek and L.J. van Vliet, "On the location error of curved edges in low-pass filtered 2D and 3D images I: theory," internal report, 1991.
- [14] L.J. van Vliet and P.W. Verbeek, "On the location error of curved edges in low-pass filtered 2D and 3D images II: application aspects," internal report, 1991.
- [15] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, "Numerical Recipes in C," Cambridge University Press, 1990.
- [16] L.D. Landau and E.M. Lifshitz, "Theory of Elasticity," Pergamon Press, 1959.
- [17] H. Walter, "Numerical representation of surfaces using an optimum principle," Ph.D. thesis (in German), University of Munich, 1971.
- [18] H. Nowacki and D. Reese, "Design and fairness of ship surfaces," in: *Surfaces in computer aided geometric design*, R. Barnhill and W. Boehm (ed), North-Holland, pp. 121-134, 1982.
- [19] M. Born and E. Wolf, "Principles of Optics, sixth edition," Pergamon Press, 1959.
- [20] C.S. Williams and O.A. Becklund, "Introduction to the Optical Transfer Function," John Wiley, 1989.
- [21] S. Wolfram, "Mathematica, second edition," Addison-Wesley, Redwood City, CA, 1988.
- [22] W. Boehm, "Differential Geometry I," in: *Curves and Surfaces for Computer Aided Geometric Design*, G. Farin (ed), Academic Press, pp. 173-183, 1990.
- [23] W. Boehm, "Differential Geometry II," in: *Curves and Surfaces for Computer Aided Geometric Design*, G. Farin (ed), Academic Press, pp. 267-283, 1990.