ESTIMATION OF CURVATURES IN POINT SETS BASED ON GEOMETRIC ALGEBRA

Helmut Seibert[a], Dietmar Hildenbrand[b], Meike Becker[b], Arjan Kuijper[a]

[a]*Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany*
[b]*Graphics Systems Group, Technical University of Darmstadt, Germany*
helmut.seibert@igd.fraunhofer.de
dietmar.hildenbrand@gris.informatik.tu-darmstadt.de
meike.becker@gris.informatik.tu-darmstadt.de
arjan.kuijper@igd.fraunhofer.de

Abstract: For applications like segmentation, feature extraction and classification of point sets it is essential to know the principal curvatures and the corresponding principal directions. For the purpose of curvature estimation conformal geometric algebra promises to be a natural mathematical language: Local curvatures can be described with the help of osculating circles or spheres. On one hand, conformal geometric algebra is able to directly compute with these geometric objects, as well as with lines and planes needed for the description of vanishing curvature. On the other hand, distance measures for fitting these objects into point sets can be handled in a linear way, leading to efficient algorithms.

In this paper we use conformal geometric algebra advantageously in order to locally compute continuous curvatures as well as principal curvatures of point sets without the need of costly pre-processing of raw data. We show results on artificial and real data. Numerical verification on artificial data shows the accuracy of our approach.

1 INTRODUCTION

The *curvature* of a surface is a feature which describes local surface properties in a well defined manner. Each certain location has a minimal and a maximal principal curvature with orthogonal principal directions. In (Alliez et al., 2003) this information is used to obtain lines of curvature and umbilical points as basis for a remeshing of surfaces. Other domains are feature extraction for local comparison of shapes (Gatzke et al., 2005), classification, registration or for instance the estimation of surface properties such as surface roughness (Lavoue, 2007) which uses this measure for object segmentation. Therefore, curvature estimation has become an important topic of research (Kalogerakis et al., 2007; Yang et al., 2006). Point set surfaces are very popular in computer graphics because they are powerful surface representations based on raw scanner data (Gross and Pfister, 2007), although it requires some effort to reconstruct the surface from these (noisy) point clouds (Hornung and Kobbelt, 2006; Mederos et al., 2005; Kolluri et al., 2004; Mitra et al., 2004; Adamson and Alexa, 2003). As a consequence, computing curvatures from
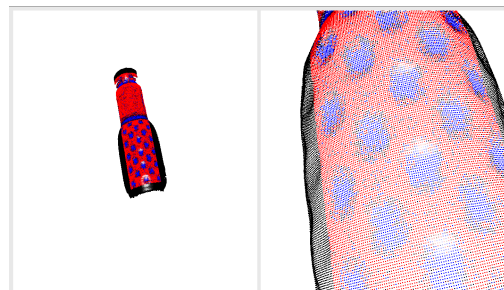

Figure 1: Point rendering of shape index estimated from a point, the shown point-set is the raw output of a structured light reconstruction system without preprocessing like smoothing or reduction. A pattern of concave areas and some noise caused by the reconstruction process can be observed at the surface of the bottle. The proposed algorithm extracts the curvature on a local neighborhood yielding a basis for object segmentation or feature detection.

point clouds without explicitly computing the surface (-mesh) is extremely difficult. Approaches to tackle this problem use statistics (Kalogerakis et al., 2007), or fitting of quadratic functions as local polynomial (Gois et al., 2006). In order to determine local cur-

vature of point sets, we use the fitting of spheres. This relates to the algebraic point set surfaces (Guennebaud and Gross, 2007). We use *conformal geometric algebra*, see e.g. (Vince, 2008) to describe the local curvature of point sets because it allows to treat geometric objects like spheres, circles, planes and lines as easy as *vectors* in vector algebra. Another reason is that it provides a linear distance measure between points and these geometric objects. Osculating circles describe the curvature of curves. The local curvature of surfaces can be described with the help of osculating spheres in all the tangent directions. Spheres and circles, as well as lines and planes needed for vanishing curvature, are handled very easily in conformal geometric algebra. Its inner product gives a linear measure between points and these geometric objects. Therefore conformal geometric algebra, as a 5D extension of the 4D projective geometric algebra, is a *natural* mathematical language to describe local curvature of point sets.

The goal of this paper is to describe an algorithm for the analysis of surface properties of scanned objects without the need of costly preprocessing of the raw point set data. The main contributions of this paper is to extend conformal geometric algebra methods in order to fit osculating circles (or lines) in discrete tangent directions (see Section 3) and to compute principal curvatures based on the estimation of continuous curvatures in all directions (see Section 4). A first evaluation of the proposed algorithms and estimation results are given in Section 5.

2 RELATED WORK

The estimation of surface curvatures based on discrete samples is sensitive to noise, as principal curvatures are second-order derivatives of the surface. Previous work in the area of surface curvature estimation covered tensor averaging using polygon edges and polyhedral mesh approximation methods. In (Hameiri and Shimshoni, 2003) the methods proposed by (Chen and Schmitt, 1992) and (Taubin, 1995) have been revisited and improved in order to overcome problems better with noisy data, as provided by 3D reconstruction systems.

While Moving Least Squares (MLS) surfaces as described in (Alexa et al., 2001) are using local plane fits based on weighted least squares, our approach is inspired by algebraic point set surfaces as introduced in (Guennebaud and Gross, 2007). The authors describe a MLS surface based on algebraic sphere fitting. Using spheres instead of planes for the local fitting allows to easily compute mean curvatures. In our

approach explained in Sections 3 and 4, we

- use an extension of weighted least squares of spheres in order to locally estimate osculating circles into point sets
- compute not only the mean curvature but continuous curvatures in all tangent directions.

Our new circle estimation algorithm based on conformal geometric algebra introduced in Section 3 is inspired by (Hildenbrand, 2005). Fitting the best suitable object in a set of points, whether it is a sphere or a plane, becomes a simple task if conformal geometric algebra primitives are used. Note that this is similar to automatically adjusting the parameter indicating how much the sphere degenerates to a plane in the fitting process of algebraic point set surfaces as described in (Guennebaud et al., 2008). Since the inner product of conformal geometric algebra can be used as a measure for distance in a linear way, this application can be solved very efficiently with the help of the Eigenvectors of a 5×5 matrix. This is already a fast solution for the fitting of spheres or planes. Our new algorithm of Section 3 for the fitting of osculating circles further improves the efficiency of the fitting algorithm to the computation of the Eigenvectors of a 3×3 matrix. In contrast to other approaches relying on surface approximations we use features of conformal geometric algebra to overcome limitations observed at the transition between planar and spherical surfaces.

3 OSCULATING CIRCLE FITTING USING GEOMETRIC ALGEBRA

We use the conformal geometric algebra in order to handle points, spheres and planes in a very intuitive way. Table 1 gives an overview on some conformal geometric algebra entities and their representations. The inner product of the vectors representing point, sphere and plane results in a scalar and can be used as a measure for the distances between them (see (Hildenbrand, 2005)).

We use this distance measure for our new approach to fitting of circles or lines into a set of points.

Let $\{\mathbf{p_i} \in \mathbb{R}^3 | i = 1, \ldots, n\}$ denote our point cloud. The corresponding points in conformal geometric algebra are then represented by

$$P_i = \mathbf{p_i} + \frac{1}{2}\mathbf{p_i^2}\mathrm{e}_\infty + \mathrm{e}_0.$$

We assume that the points to be approximated are roughly aligned in the direction of a tangent vector

Table 1: List of some basic geometric primitives provided by the 5D conformal geometric algebra. The bold characters represent 3D entities ($\mathbf{x}$ is a 3D point, $\mathbf{n}$ is a 3D normal vector and $\mathbf{x}^2$ is the scalar product of the 3D vector $\mathbf{x}$). The two additional basis vectors e_0 and e_∞ represent the origin and infinity. The parameter r represents the radius of the sphere and the parameter d represents the distance of the plane to the origin.

entity	representation
Point	$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2\mathrm{e}_\infty + e_0$
Sphere	$S = P - \frac{1}{2}r^2\mathrm{e}_\infty$
Plane	$\pi = \mathbf{n} + d\mathrm{e}_\infty$

at $\mathbf{x}$ (see Figure 2). Therefore our fitting task can be reduced to the task of fitting a circle (or in the limit case a line) in the direction of the normal vector. This circle is also called the osculating circle in $\mathbf{x}$. Let $\mathbf{n}$ be the normal vector at the location $\mathbf{x} = (x_1, x_2, x_3)$.
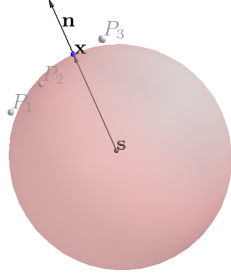

Figure 2: Estimation of curvature at $\mathbf{x}$ based on the fitting of a sphere (or plane) with center point in the opposite direction of the normal $\mathbf{n}$ into a set of points

A sphere or plane K in conformal geometric algebra is in general represented by the 5D vector

$$K = s_1e_1 + s_2e_2 + s_3e_3 + s_4\mathrm{e}_\infty + s_5\mathrm{e}_0,$$

for $s_1, ..., s_5 \in \mathbb{R}$.

If $s_5 = 0$ we get a plane, in the other case a sphere. As described above we have given that the center of the sphere S that we fit into our point set is $\mathbf{m} = \mathbf{x} + r\mathbf{n}$, where $r \in \mathbb{R}$ denotes the radius of the sphere. Therefore we can represent the sphere as

$$S' = \mathbf{m} + \frac{1}{2}(\mathbf{m}^2 - \mathbf{r}^2)e_\infty + e_0.$$

Now we scale this sphere by $t_3 \in \mathbb{R}$ and we get an object of the form Equation 3 which can be a sphere or a plane:

$$S = t_3(\mathbf{m} + \frac{1}{2}(\mathbf{m}^2 - \mathbf{r}^2)\mathrm{e}_\infty + \mathrm{e}_0)$$
$$= t_3\mathbf{m} + t_3\mathrm{e}_0 + t_2\mathrm{e}_\infty,$$

where $t_2 = \frac{1}{2}t_3(\mathbf{m}^2 - \mathbf{r}^2)$.

The inner product of S and P_i gives us a measure for their distance:

$$P_i \cdot S = (\mathbf{p_i} + \frac{1}{2}\mathbf{p_i}^2e_\infty + e_0) \cdot (t_3(\mathbf{x} + r\mathbf{n}) + t_3e_0 + t_2e_\infty)$$

$$= t_1(\mathbf{p_i} \cdot \mathbf{n}) - t_2 + (\mathbf{p_i} \cdot \mathbf{x} - \frac{1}{2}\mathbf{p_i}^2)t_3$$

$$= \sum_{j=1}^{3} w_{ij}t_j,$$

where $t_1 = t_3r$ and

$$w_{ij} = \begin{cases} \mathbf{p_i} \cdot \mathbf{n}, & j = 1 \\ -1, & j = 2 \\ \mathbf{p_i} \cdot \mathbf{x} - \frac{1}{2}\mathbf{p_i}^2, & j = 3. \end{cases} \quad (1)$$

In order to fit the best approximating sphere or plane into the point set, we consider $min \sum_{i=1}^{n}(P_i \cdot S)^2$ as residual for our least squares approach. Equivalently, we can write the problem using the bi-linear form as $min_{t \in \mathbb{R}^3}(t^T Bt)$, where $t = (t_1, t_2, t_3)$ and B is a 3×3 matrix with entries

$$b_{jk} = \sum_{i=1}^{n} w_{ij} \cdot w_{ik}, \quad \text{for } j, k = 1, 2, 3.$$

B is symmetric and we consider only normalized results t with $t^T t = 1$, the given problem can be considered as an eigenvector problem and the solution is given by the eigenvector corresponding to the smallest eigenvalue of B.

4 PRINCIPAL CURVATURE ESTIMATION

The idea of the proposed algorithm is as follows: A discrete sampling of the surface curvature can be used to obtain an estimation of the principal curvatures and the corresponding principal directions.

An outline of the algorithm is as follows:

1. Local point selection with chosen estimation radius $d_{\max}$

2. Local tangent frame definition

3. Define planes for n directions and select the points closer than a chosen threshold t to each plane

4. Fitting of circles to the points for each direction

5. Estimate parameters for the curvature model function

6. Derive curvature information from model parameters

To evaluate the curvature of a surface represented as a set $S = \{\mathbf{p}_0 \dots \mathbf{p}_n\}$ of points $\mathbf{p}_i = (x_i, y_i, z_i)$ at a certain location $\mathbf{x} \in S$, the set $N = \{\mathbf{p}_i \in S \mid |\mathbf{x} - \mathbf{p}_i| \leq d_{\max}\}$ of neighboring points within a given distance $d_{\max}$ has to be considered. To optimize the point selection appropriate hierarchical representations of the surface S such as octree or kd-tree can be used. Given additional information on the surface topology such as the triangle indices, the k-ring neighbor method may be suitable to access the neighborhood while reducing the number of distance tests in point selection.

The subsequent processing steps are applied with respect to the tangent plane at point $\mathbf{x}$. If the normal $\mathbf{n}$ is not known a normal vector can be estimated by applying a sphere/plane fit to the selected points, resulting in a surface normal or a vector to the center of an estimated sphere which can be normalized and interpreted as a normal vector. In this case the orientation of the surface is not defined from the local perspective, additional effort by means of global surface inspection is required. Given a normal vector $\mathbf{n}$ an arbitrary point on the tangent plane can be used to create one orthogonal basis vector $\mathbf{t}_u$. The second basis vector $\mathbf{t}_v$ can be calculated using the cross product $\mathbf{t}_v = \mathbf{n} \times \mathbf{t}_u$ to define a local tangent frame.

To obtain a discrete sampling of the surface properties depending on the direction a set of n planes perpendicular to the tangent plane has to be defined. The normal $\mathbf{l}_i$ for plane $L_i, i \in \{1 \dots n\}$ can be calculated by rotating the local tangent vector $\mathbf{t}_v$ counterclockwise around $\mathbf{n}$ by $\alpha_i = \frac{(i-1) \cdot \pi}{n}$ radians. As all planes need to pass the reference point $\mathbf{x}$, each of the n planes L_i can be defined in hessian normal form as $(\mathbf{l}_i \cdot \mathbf{n}) - |\mathbf{x}| = 0$. For each plane L_i the points $\mathbf{p}_j$ within the neighborhood N whose distance to the plane is less than a predefined threshold distance $d_{\max}$ are collected into point sets $D_i = \{\mathbf{p}_j \in N \mid |(\mathbf{l}_i \cdot \mathbf{n}) - \mathbf{p}_j| \leq d_{\max}\}$, Figure 3 shows the selected points within a chosen estimation radius. The point sets D_i are split into two segments each using the plane L_i rotated around the surface normal at $\mathbf{x}$ by $\frac{\pi}{2}$ radians in order to perform the following fitting algorithms on a larger set of samples covering the whole revolution. Performing this operation allows to capture local discontinuities as present at edges as well. The curvature κ at the point $\mathbf{x}$ equals the inverse radius r of the osculating circle. The osculating circle is estimated in a least squares sense by applying the curve fit explained in Section 3. For each set of cross section points D_i a circle fit f_i has to be performed. The points $\mathbf{p}_j \in D_i$ have been selected with respect to the plane L_i and the reference point $\mathbf{x}$, the points $\mathbf{p}_j$ represent the cross section of the originating surface S with the plane. The fitting result for one di-
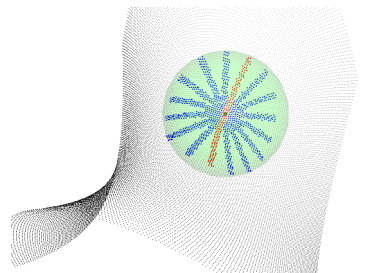

Figure 3: The selected point sets for 8 directions within estimation radius on a saddle surface are shown highlighted.
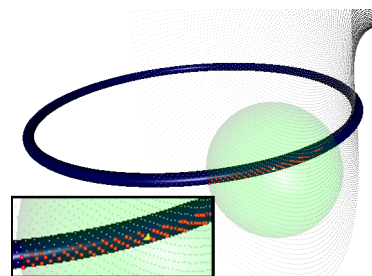

Figure 4: Circle fitting result for one direction. The highlighted points within the estimation radius have been used to approximate a circle. The resulting osculating circle is visualized as a torus.

rection is illustrated in Figure 4, the results for a 8 direction estimation can be seen in Figure 5. If the point sets D_i have been split as mentioned above the fitting has to be applied for each subset to obtain a curvature sample. According to the Euler formula the curvature values $c(\alpha)$ in direction of a tangent vector $\mathbf{t}$ rotated around the reference point $\mathbf{x}$ with angle α are defined as $\kappa(\alpha) = \kappa_1 \cdot \cos^2(\alpha) + \kappa_2 \cdot \sin^2(\alpha)$ with the principal curvatures κ_1 and κ_2, which can be rearranged to the form $\kappa(\alpha) = \frac{-(\kappa_1 - \kappa_2) \cdot \sin(2\alpha - \frac{\pi}{2}) + \kappa_1 + \kappa_2}{2}$. With κ_1 and κ_2 fixed it provides the simple model
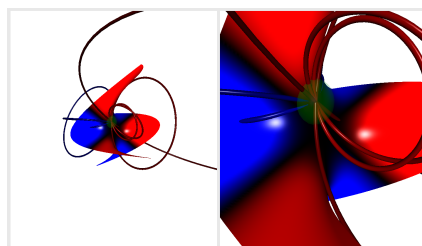

Figure 5: The circle fitting results in a set of circles, each shown as a torus for a certain direction. The base Surface is a monkey saddle surface, the colors of the surface encode the shape index. The osculating circles visualized as tori are sized and oriented according to the directional estimation results.
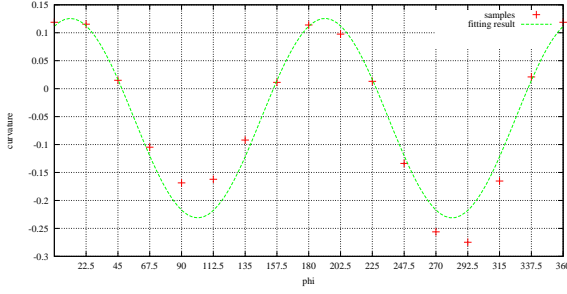
Figure 6: Sine fitting results. The curvature has been sampled in 8 directions resulting in 16 measurements. The sine function approximates the real distribution of curvatures yielding the location of curvature maxima and minima and the corresponding direction angles with respect to the local reference frame.
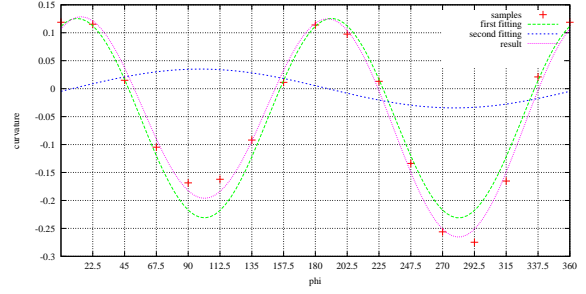

Figure 7: Refined fitting using a second model function with half frequency. The sum of both model functions shows a better approximation of the measured curvature samples.

for this function $\kappa(\alpha) = a \cdot \sin(\alpha + \varphi) + o$. This model function is fitted to the sampled curvatures by optimizing the amplitude a, phase φ and offset o using the three parameter sine fit method (IEEE, 2001), which allows to obtain a least squares solution $\mathbf{r} = (r_1, r_2, r_3)^T = (\mathbf{M}^T\mathbf{M})^{-1} \cdot (\mathbf{M}^T\mathbf{f})$ with matrix $\mathbf{M}$ defined appropriately using the angles α_i and vector $\mathbf{f} = (f_1, f_2, \cdots, f_n)^T$, which consists of the measured curvatures. The result $\mathbf{r}$ can be converted to amplitude $a = \sqrt{r_1^2 + r_2^2}$, phase

$$\varphi = \begin{cases} \arctan\left(-\frac{r_2}{r_1} + \frac{\pi}{2}\right) & \text{if } r_1 > 0 \\ \arctan\left(-\frac{r_2}{r_1} + \frac{3\pi}{2}\right) & \text{if } r_1 < 0 \end{cases}$$

and offset $o = r_3$.

The maximum and minimum values of the model function directly yield the maximum and minimum curvature. The corresponding alignment of the principal directions can be calculated using the angle of minimum and maximum relative to the tangent frame considering a phase correction φ to obtain angles of the extrema $\kappa_{1/2} = a \pm o$:

$$\mathbf{v}_{\max} = \left(\sin\left(-\varphi + \frac{\pi}{2}\right), \cos\left(-\varphi + \frac{\pi}{2}\right), 0\right)$$

$$\mathbf{v}_{\min} = \left(\sin\left(-\varphi + \frac{3\pi}{2}\right), \cos\left(-\varphi + \frac{3\pi}{2}\right), 0\right)$$

The result of such an optimization for 16 samples is shown in Figure 6.

As the curvature samples are present for each direction, the samples cover two phases of the model function. Curvature measurements for opposite directions differ, if curvature gradients are present. This allows a second optimization, which uses the difference between the sample value and the parametrized model function to perform a second fitting of a sine function

with half frequency yielding amplitude a_2, phase φ_2 and offset o_2. The evaluation of this model function at the location of maxima and minima can be used to correct the curvature measurements and to derive the direction of the gradient in the principal direction, see Figure 7. The complete model function is then $c(\alpha) = a \cdot \sin(\alpha + \varphi) + o + a_2 \cdot \sin(0.5 \cdot \alpha + \varphi_2) + o_2$. The phase shift which may be introduced by this summation is usually small and has been ignored. The maxima and minima derived by the first fitting have been corrected choosing the lowest minimum value and the highest maximum value. The vectors $\mathbf{v}_{\max}$ and $\mathbf{v}_{\max}$ are the principal curvature directions with respect to the tangent frame. The local tangent frame is defined by the three basis vectors $\mathbf{t}_u$, $\mathbf{t}_v$ and $\mathbf{n}$. The rotation of the local tangent frame with respect to the global object coordinate system is denoted $R_{\text{global}}^{\text{tangent}}$. The principal directions in object space can be calculated by applying inverse tangent rotation $R_{\text{tangent}}^{\text{global}} = (R_{\text{global}}^{\text{tangent}})^{-1}$. The result is then $\mathbf{p}_{\max} = R_{\text{tangent}}^{\text{global}} \cdot \mathbf{v}_{\max}$ and $\mathbf{p}_{\min} = R_{\text{tangent}}^{\text{global}} \cdot \mathbf{v}_{\min}$. The robustness and quality of the model function fit can be increased using an outlier detection and removal. This applies mostly to irregular sampled and distorted surfaces. Choice and evaluation of an appropriate method is subject to further improvement of this approach.

4.1 Parameters and Suitable Ranges

The proposed algorithmic approach is controlled by a set of parameters. The **estimation radius** defines the size of the surface portion for the evaluation. The estimation radius must be chosen considering two aspects. On the one hand it is necessary to select enough points for the circle fitting and on the other hand it must correspond to the maximum and minimum feature size to be detected. The **number of estimation directions** respective of normal sections used for the curvature estimation should be chosen small in order

to limit the computational effort and it should be big enough to allow a reliable fitting of the model function. A choice of 8 estimation directions resulting in 16 samples provided a reasonable ratio of robustness and accuracy for the performed tests.

The **maximum plane distance for point selection** defines the focus of the point selection with respect to the plane, can be very small for high point densities. Bigger values may decrease the estimation quality as the optimal point set for circle estimation would be coplanar. Concerning the parameters controlling the point selection a choice must assure that there are enough points for the circle estimation algorithm. A theoretical limit of three points must be exceeded, while large amounts of points can be handled without problems. In practice a point count between 10 and 30 for each circle estimation yielded best results. Given special surface constellations as closely located opposite surface areas within concave surface regions this can only be identified if topological information is present e.g. in triangle mesh structures. The estimation radius and the evaluation distance d_{max} must be chosen carefully in order to detect small features and to avoid topological misinterpretations. As the approach is suitable for dense sampled surfaces or non optimized triangulated surfaces the recommendation is to choose the radius as big as possible while being small with respect to the depicted object structures.

5 RESULTS

The proposed algorithm has been evaluated considering the two basic optimization steps and the overall performance. As the circle fit algorithm is intended to be used on data acquired from the real world, several properties of point sampled surface data have been simulated in order to evaluate the algorithm. The most important influences are point density, point distribution, distortion and a range of radii are included in the data generation process. The tests have been done using points on a circle segment simulating a fixed estimation window of width 4. According to each performed test the radius, the number of points and amount of noise have been varied. The noise has been generated using a gaussian distribution and a fixed level, as this is in practice introduced by the acquisition process and should depend only on the distance and viewing angles but not on the size and shape of the surface.

As the estimation window size remains the same for all radii it can be expected that estimation errors grow with growing radii and growing amount of

noise. An optimal estimation algorithm would yield the radius used for the point generation as estimation result. The resulting radii are used within our approach to determine the curvature, which is the reciprocal value, e.g. the interpretation of the results must take into account that errors occurring at the estimation of big radii result in smaller errors for low curvature values, e.g. flat portions of the investigated surface. An extension of the fitting algorithm to a weighted least squares approach has been very investigated, too, but the results did not turn out to be promising so far.

As the sampling density of 3D reconstruction data is varying with respect to the measurement angle and the measurement distance, one cannot always assume regular distribution of the point samples. In these tests 12 points have been generated for some radii within 1 and 10000 using a regular angle dependent distribution of points along the circle segment or a random distribution. A normal distributed distortion with $\sigma = 0.5$ has been added to investigate the influence of noisy data. The test result is shown in Figure 8, without noise the type of point distribution did not show a significant influence. With an increasing amount of noise the estimation quality decreased. In the observed cases the distribution of points did not show a significant influence, while the circle fit algorithm tends to overestimate the radii.

The next test shows the influence of the point count on the estimation quality. The points are randomly distributed, and the tested point counts were 4, 12 and 24. To point out the limits of the algorithm a random point distribution and a gaussian noise ($\sigma = 0.5$) have been chosen for this test. The estimation errors observed are bigger at very small radii, e.g. for bigger radii a quiet good estimation can be achieved. Using a small point count of 4, the estimation yields smaller radii than the real ones, while increasing point counts of 24 yield slightly bigger radii. In this test the point count of 12 showed the best estimation result, see Figure 9. The estimation results for all point counts without noise were optimal independent of random or regular point distribution.

The sine fitting algorithm explained in Section 4 has been evaluated choosing a certain amplitude, phase and offset. For the chosen parameters several samples with different levels of noise have been generated. The RMS error for the given samples with respect to the estimated function has been observed for an amount of four up to a hundred points. The estimation results remain within the expected range for all tested numbers of points. We assume that the number of estimation directions does not have a significant influence on the fitting quality, but the qual-
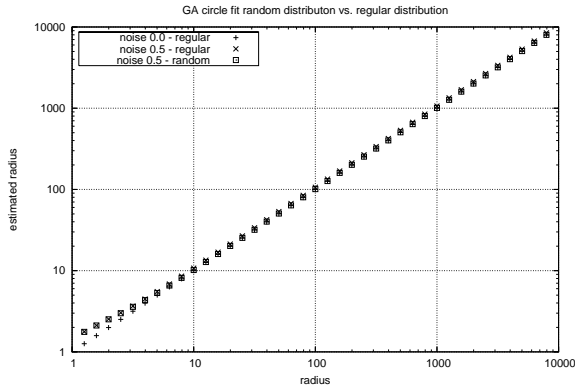
Figure 8: Circle fitting results for a regular and random respectively distribution of 12 points on small segments of circles with a fixed angle and increasing radii considering distortion. The real radii are shown on the abscissa and the estimated radii are on the ordinate. Without distortion an optimal estimation can be observed resulting in a straight line. In case of distortion the algorithm tends to overestimate radii. In case of the regular point distribution a slightly increased overestimation can be observed.
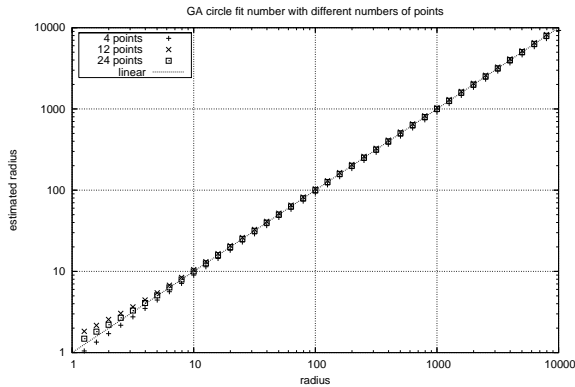

Figure 9: Circle fitting results with respect to the number of points. The increasing radii have been estimated using different numbers of points considering a gaussian noise ($\sigma = 0.5$).

ity of the samples heavily influences the results. The proposed approach relies on two subsequent steps of least-squares optimizations, each of which introduce small errors. The achievable overall performance is not yet comparable to the quiet exact evaluation considering over the years optimized state-of the art algorithms originating from the tensor method introduced by Taubin (Taubin, 1995). Nevertheless, the approach is efficient and robust to many aspects and may be very useful to obtain an initial rough estimation in large datasets.

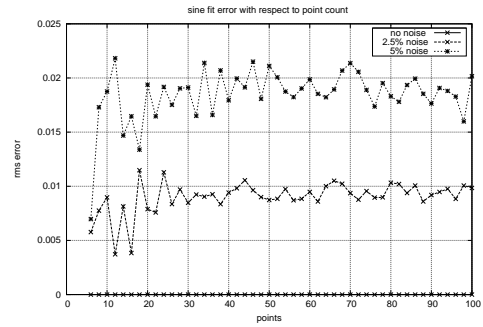The results of the proposed algorithm have been

Figure 10: RMS Errors of the sine fitting algorithm used for the fitting of the model function. The results are shown for different sample counts, considering distortions of the samples as well.

evaluated on basis of surfaces generated by simple bivariate functions. The real principal curvatures and directions of these generator functions can be analysed exactly using the fundamental forms based on the first and second derivatives. For each test a certain range in the parameter space of the generator function has been chosen and theoretical curvature values and surface points have been compared to estimations based on the proposed approach. The results are shown in table 2. An example for the result of the proposed approach applied to a real 3D reconstruction of a small bottle is shown in Figure 1. The reconstructed surface has some distortions caused by reflections of the original bottle during acquisition. Nevertheless, the estimated shape index allows an interpretation of the surface structure. The proposed methods provide several aspects we would like to investigate in future work. The sampling of curvatures in opposite directions from the point may be used to exploit the curvature gradient or to detect local discontinuities such as edges.

A first approach by setting up a model function summing two sine functions with the first function having the double frequency has been done. The model function has been fitted using a linear sine fitting approach. We are planning to explore different optimization techniques like Levenberg-Marquardt using a more complex model function. We assume that dynamic sampling and rendering of point set surfaces can be improved by this method.

6 CONCLUSION

In this paper we used geometric algebra techniques in order to compute principal curvatures of point sets in an elegant and efficient way. We could benefit espe-

Table 2: Surface curvature evaluation results for regions of the saddle surface $z = 0.2x^2 + 0.1y^2$ have been compared to the known analytical values of the generating function. $\Delta\kappa_1$ and $\Delta\kappa_2$ denote the absolute value of the difference.

Point distrib.	regular	random
est. radius	2	2
est. directions	8	8
points	13926	14049
x min	-10	-10
x max	10	10
Δ x	0.11	random
y min	-10	-10
y max	10	10
Δ y	0.11	random
$\Delta\kappa_1$ min	0.000	0.000
$\Delta\kappa_1$ max	0.028	0.030
$\Delta\kappa_1$ mean	0.008	0.008
$\Delta\kappa_1$ stdd.	0.006	0.006
$\Delta\kappa_2$ min	0.000	0.000
$\Delta\kappa_2$ max	0.085	0.090
$\Delta\kappa_2$ mean	0.019	0.018
$\Delta\kappa_2$ stdd.	0.018	0.018

cially from the property of geometric algebra of naturally describing curvatures. On the one hand, geometric algebra can easily describe geometric objects like spheres and planes, circles and lines as well as the limit between them. On the other hand its inner product provides possibilities for a fast measurement of distances. In a nutshell, we presented a novel simple and efficient approach, which allows to access curvature information within dense point sets without costly preprocessing.

REFERENCES

Adamson, A. and Alexa, M. (2003). Approximating and Intersecting Surfaces from Points . In *Eurographics Symposium on Geometry Processing (SGP)*, pages 230–239.

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2001). Point set surfaces. In *Conference on Visualization (VIS)*, pages 21–28.

Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., and Desbrun, M. (2003). Anisotropic polygonal remeshing. *ACM Transactions on Graphics*, 22(3):485–493.

Chen, X. and Schmitt, F. (1992). Intrinsic surface properties from surface triangulation. In *European Conference on Computer Vision (ECCV)*, pages 739–743, London, UK. Springer-Verlag.

Gatzke, T., Grimm, C., Garland, M., and Zelinka, S. (2005). Curvature Maps For Local Shape Comparison. In *Shape Modeling and Applications*, pages 244–253.

Gois, J., Tejada, E., Etiene, T., Nonato, L., Castelo, A., and Ertl, T. (2006). Curvature-driven modeling and rendering of point-based surfaces. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 27–36.

Gross, M. and Pfister, H. (2007). *Point Based Graphics*. Morgan Kaufmann.

Guennebaud, G., Germann, M., and Gross, M. (2008). Dynamic sampling and rendering of algebraic point set surfaces. In *Eurographics*, pages 653–662.

Guennebaud, G. and Gross, M. H. (2007). Algebraic point set surfaces. *ACM Transactions on Graphics*, 26(3):23.

Hameiri, E. and Shimshoni, I. (2003). Estimating the principal curvatures and the darboux frame from real 3-d range data. *IEEE Systems, Man, and Cybernetics B (SMC-B)*, 33(4):626–637.

Hildenbrand, D. (2005). Geometric computing in computer graphics using conformal geometric algebra. *Computers & Graphics*, 29(5):802–810.

Hornung, A. and Kobbelt, L. (2006). Robust Reconstruction of Watertight 3D Models from Non-uniformly Sampled Point Clouds Without Normal Information . In *Eurographics Symposium on Geometry Processing (SGP)*, pages 41–50.

IEEE (2001). *Std. 1241-2000 IEEE standard for terminology and test methods foranalog-to-digital converters*, chapter 3, pages 26–27.

Kalogerakis, E., Simari, P., Nowrouzezahrai, D., and Singh, K. (2007). Robust Statistical Estimation of Curvature on Discretized Surfaces. In *Eurographics Symposium on Geometry Processing (SGP)*, pages 13–22.

Kolluri, R., Shewchuk, J. R., and O'Brien, J. F. (2004). Spectral Surface Reconstruction From Noisy Point Clouds . In *Eurographics Symposium on Geometry Processing (SGP)*, pages 11–22.

Lavoue, G. (2007). A Roughness Measure for 3D Mesh Visual Masking. In *ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization (APGV)*, pages 57 – 60.

Mederos, B., Amenta, N., Velho, L., and de Figueiredo, L. H. (2005). Surface Reconstruction for Noisy Point Clouds. In *Eurographics Symposium on Geometry Processing (SGP)*, pages 53–62.

Mitra, N. J., Gelfand, N., Pottmann, H., and Guibas, L. (2004). Registration of Point Cloud Data from a Geometric Optimization Perspective . In *Symposium on Geometry Processing*, pages 23–32.

Taubin, G. (1995). Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 902–907.

Vince, J. (2008). *Geometric Algebra for Computer Graphics*. Springer.

Yang, Y.-L., Lai, Y.-K., Hu, S.-M., and Pottmann, H. (2006). Robust Principal Curvatures on Multiple Scales. In *Eurographics Symposium on Geometry Processing (SGP)*, pages 223–226.