Consistent Propagation of Normal Orientations in Point Clouds

Sören König, Stefan Gumhold

TU Dresden
Email: {Soeren.Koenig,Stefan.Gumhold}@tu-dresden.de

Abstract

Many algorithms for point cloud processing especially surface reconstruction rely on normal information available at each point. Normal directions are typically taken from a local tangent plane approximation which is obtained by fitting a surface model to the neighboring point samples. While the direction can be estimated locally, finding a consistent normal orientation over the whole surface is only possible in a global context. Existing methods for this problem can be classified into volumetric and propagation based approaches. Volumetric methods are trying to divide the space into inside and outside regions which is often complicated to implement and have problems with open surfaces and large holes. Propagation based methods can deal with open surfaces but often fail on sharp features. This paper analyses the behavior of surficial orientation methods, gives a better understanding of the underlying model assumptions of existing techniques and proposes a novel and improved propagation heuristic.

1 Introduction

Surface reconstruction from unstructured three-dimensional point sets sampled from the surface of a real world object has become a major research topic in computer graphics and vision. Beside the pure spatial positions a very important surface property is the normal of the underlying surface associated with each sample point. This information is essential for a variety of processing algorithms like denoising, filtering, matching and registration, reconstruction [10, 5, 9] or rendering. If normal information is not provided directly by the acquisition process a standard approach is to reconstruct the normal at a given position directly from the point samples. Hoppe [4] proposed to estimate the local tangent plane by applying a PCA to the k-neighborhood around a given point $\vec{p}$. Pauly et al.[11] use the normal of a fitted plane or alternatively the gradient of an additionally fitted bivariate polynomial. The main difference to Hoppe's approach is the use of a distance decaying exponential weighting function inspired by Levin's work [6]. In [10] general 3D quadrics, bivariate quadratic polynomials in local coordinates and piecewise quadric surfaces are used as local shape functions. They propose to separate points from different surface sheets or at sharp creases by clustering. Mitra et al.[8] propose to use the points within a ball of radius r instead of a k-neighborhood. They also propose an iterative method to estimate an optimal radius assuming zero mean noise with known standard deviation.

Beside such numerical techniques there also exists some combinatorial methods. The Voronoi based technique in [1] shows that the line from the pole of point $\vec{p}$ to the point $\vec{p}$ itself can be used as an approximation to the surface normal at this position. Later the big Delaunay ball method [2, 3] was suggested as an improvement to handle noisy point cloud data. One large drawback of these combinatorial methods is the computation of Voronoi cells which can be quite difficult and slow especially in the context of extremely large data sets produced by state-of-the-art scanning technologies.

Numerical methods are often preferred due to easier implementation and faster running times. A drawback of these method is that they only provide the normal direction at a given point but not a consistent normal orientation over the whole surface.

In this paper we focus on the problem of finding a consistent normal orientation which in our opinion has not been solved satisfactorily yet.

Our main contributions can be summarized as follows: We give a careful analysis of existing surficial orientation methods and provide a better understanding of the underlying model assumptions and limitations. Based on these insights we propose additional improvements to further reduce the possibility of orientation failures.

The relevance of the normal orientation problem can also be seen due to the many relations to other research areas like image segmentation, random field optimization or computational physics.

1.1 Terminology and Problem Statement

First we want to introduce some terminology, before we give a more formal problem definition. The input of a normal orientation method is an unorganized set of points $\mathbf{P} = \{\vec{p}_1, ..., \vec{p}_n\}$ sampled from an unknown orientable, two dimensional manifold surface $\mathbf{S}$ embedded in $\mathbb{R}^3$. The surface $\mathbf{S}$ can possibly be a surface with boundary e.g. due to an incomplete acquisition process. Additionally, we have the set of normal directions $\mathbf{N} = \{\vec{n}_1, ..., \vec{n}_n\}$ where $\vec{n}_i$ denotes the normal direction on $\mathbf{S}$ at point $\vec{p}_i$.

The problem we wish to solve is to find a global consistent set of normal orientations $\mathbf{O} = \{o_1, ..., o_n\}$ with $o_i \in \{+1, -1\}$ be the orientation of normal $\vec{n}_i$ so that $\vec{n}_i$ is locally consistent in a small neighborhood $\mathcal{N}_i$ around $\vec{p}_i$. Intuitively speaking a *locally consistent* normal orientation is achieved if all orientated normals $o_j \cdot \vec{n}_j$ of points in $\mathcal{N}_j$ are pointing to the same side of S. A *globally consistent* orientation is achieved if each normal $\vec{n}_i$ is oriented locally consistent by o_i. Later in section 3 we give an overview of several possible criteria to define such *consistencies*.

The main data structure on which the discussed algorithms are working is the Riemannian graph. As described in [4], a Riemannian graph G is an undirected graph, which can be constructed by enriching the Euclidean minimal spanning tree EMST over all points $\vec{p} \in \mathbf{P}$ by adding additional edges e_{ij} between $\vec{p}_i$ and $\vec{p}_j$ if $\vec{p}_i$ is in $\mathcal{N}_j$ or $\vec{p}_j$ is in $\mathcal{N}_i$. The neighborhood $\mathcal{N}_i$ is typically defined as the k-nearest neighbors of $\vec{p}_i$ in $\mathbf{P}$ but other meaningful neighborhood definitions are possible.

2 Related Work

We classify normal orientation techniques into *surficial methods* and *volumetric methods*. Surficial methods try to establish a consistent normal orientation by propagating the orientation information by one or more normal consistency rules over the sampled surface. Volumetric methods try to segment the surrounding space into inside and outside regions. Surface normals then have to point from inside to outside.

2.1 Surficial methods

The most frequently cited algorithm in this class is described by Hoppe in [4]. First a Riemannian graph is constructed. Each edge between two points is assigned a weight measuring the unreliability for propagating the normal orientation along this edge. Hoppe uses the absolute value of the dot product of the two associated normals which is a simple measure of normal variation. Normal orientations are propagated along the minimal spanning tree of the graph starting at a point with known orientation. The algorithm can deal with open surfaces but can suffer in the presence of sharp edges or corners.

In [13] Xie et al. propose to start the propagation from multiple seeds. In a first step the algorithm tries to avoid propagation along high curvatures. This results in multiple orientated patches touching each other at sharp edges or corners. In a second step neighboring patches are consistently orientated with a modified normal consistency rule which can deal with sharp features.

2.2 Volumetric methods

Mello et al. [7] suggest a method to construct a signed distance function representation from a point sampled closed surface by constructing an adaptively subdivided tetrahedral volume decomposition. All vertices of a tetrahedron which lie on the same side of a locally fitted plane should be labeled either positive or negative while the vertices on the other side should be labeled with the opposite sign. Requiring a global consistent labeling over all tetrahedrons is exactly equivalent to the problem formulation of finding a minimum energy configuration of an Ising model [7]. To solve the optimization problem a simulated annealing method is used.

Another volumetric approach is described in [12] by Xie. Starting the surrounding space of the point cloud is segmented into so called mono-orientated regions by growing active contours from multiple seeds. Mono-oriented regions are then consistently determined as inside or outside by a voting algorithm. To speed up the computation a hierarchical implementation using an octree is used. Volumetric methods are typically limited to closed surfaces and cannot easily deal with partial scans or scans with large holes but are in general more robust against noise and outliers. Surficial methods are mostly easier to implement and can handle incomplete scans with holes, but more often fail at sharp features.

In this paper we focus on the analysis and improvement of surfacial orientation methods.

3 Consistent Normal Orientation

As described in the problem statement the input to the normal orientation algorithm consists of the points **P** and the corresponding normals **N**. From the points **P** we construct a Riemannian graph G. As suggested by Hoppe the consistent normal orientation problem can be reduced to a graph optimization problem. Each edge e_{ij} is assigned a cost that measures the unreliability in the orientations of the incident normals $\vec{n}_i$ and $\vec{n}_j$. The goal is to find a global orientation of all normals that minimizes the sum of all edge costs resulting in a most consistent normal orientation. In [4] it is noted that this problem is NP-hard.

Although one could use a simulated annealing approach to find the global orientation, we follow the simpler approach proposed by Hoppe. The basic idea is to first define the orientation of one seed normal either by hand or by a simple automatic approach - for example by using the normal of the point with largest x coordinate as seed and by setting the corresponding orientation such that the x-component of the seed normal is positive. In the second step one propagates the orientation of the seed normal along edges of the Riemannian graph. To propagate orientation along edge e_{ij} from an oriented normal $\vec{n}_i$ to normal $\vec{n}_j$ a *flip criterion* is necessary that tells whether to flip the original orientation of $\vec{n}_j$. The flip criterion is typically implemented by minimizing the edge cost over both possible orientations of $\vec{n}_j$.

The edges used for propagation are computed through a minimal spanning tree (MST). A new edge cost is defined that measures the unreliability of propagating normal orientation. The unreliability cost must be independent of the initial normal orientation.

In the following we analyze existing flip criteria and unreliability costs before proposing a new approach that improves upon existing method and widens the applicability of the simple and efficient MST approach.

3.1 Zero Curvature Assumption

Hoppe proposed in the original MST approach [4] to assume that the tangent spaces of points that are adjacent in the Riemannian graph should be close to parallel, i.e. that the curvature should be small or even close to zero. The resulting flip criterion simply checks, whether the edge incident normals point in the same or opposite directions:

$$f_{\text{Hoppe}}(i,j) = \langle \vec{n}_i, \vec{n}_j \rangle < 0,$$

i.e. the normal $\vec{n}_j$ is flipped if the dot product with $\vec{n}_i$ is negative. The corresponding unreliability measure used as edge cost for MST extraction is

$$u_{\text{Hoppe}}(i,j) = 1 - \left| \langle \vec{n}_i, \vec{n}_j \rangle \right| .$$

It is zero for parallel or anti-parallel normals and equal one (maximal) for orthogonal normals. For point clouds sampling a surface with sharp features the simple assumption of small curvature does not hold independent of how densely we sample the surface. Hoppe's approach is not able to propagate normal orientation over an acute crease and therefore fails for example in case of a point cloud that samples a tetrahedron as illustrated in the results section.

3.2 Constant Curvature Assumption

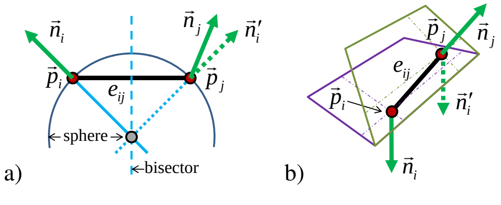

Figure 1: a) in the approach of Xie et al. the normal $\vec{n}_i$ is reflected at the bisector of e_{ij} before compared with $\vec{n}_j$. b) along acute crease edges the approach can fail.

To overcome the problems of Hoppe's approach at sharp features, Xie et al. propose a secondary flip criterion that is used exclusively in areas of high curvature, where the reference normal $\vec{n}_i$ is first reflected to $\vec{n}_i'$ at the bisector plane orthogonal to edge

e_{ij} before the comparison with $\vec{n}_j$ is performed. This approach is illustrated in Figure 1 a). The resulting flip criterion and unreliability cost compute to

$$\begin{aligned} f_{\text{Xie}}(i,j) &= \left\langle \vec{n}_i', \vec{n}_j \right\rangle < 0 \quad \text{with} \\ \vec{n}_i' &= \vec{n}_i - 2\left\langle \hat{e}_{ij}, \vec{n}_i \right\rangle \hat{e}_{ij} \quad \text{where} \\ \hat{e}_{ij} &= (\vec{p}_i - \vec{p}_j)/\left\|\vec{p}_i - \vec{p}_j\right\|. \\ u_{\text{Xie}}(i,j) &= 1 - \left|\left\langle \vec{n}_i', \vec{n}_j \right\rangle\right|. \end{aligned}$$

While in the paper of Xie [13] this criterion is only motivated empirically, we can argument as follows that it is actually motivated by assuming that the surface is locally of constant curvature: From the constraints that the surface **S** is passing through $\vec{p}_i$ and $\vec{p}_j$ and that its normal direction at $\vec{p}_i$ is $\vec{n}_i$, a unique sphere of constant curvature is defined. As shown in Figure 1 a), the sphere center is on the intersection of the bisector plane and the line through $\vec{p}_i$ with direction $\vec{n}_i$. Reflecting this line at the bisector results in the sphere normal $\vec{n}_i'$ at the point $\vec{p}_j$, which can then be compared to the actual normal $\vec{n}_j$. By symmetry does the same construction starting at $\vec{p}_j$ result in the same flip criterion and unreliability cost.

We want to mention here that the approach of Xie works as well for the planar case and can completely replace the approach by Hoppe without the need for a two stage approach with an initial segmentation of the surface in flat and curved areas. Finally, the approach of Xie also works quite well for point samples of surfaces with close sheets.

3.3 Low Curvature Variation

In this section we propose our new flip criterion and unreliability cost. We extend the approach of Xie in two ways: firstly we allow variations of the curvature and secondly we define a reference plane that allows to avoid problems with the acute crease illustrated in Figure 1 b).

Our major idea is to propagate the normal orientation from $\vec{p}_i$ to $\vec{p}_j$ along the most plausible path along the local surface. A plausible path should respect the tangent space information at $\vec{p}_i$ and $\vec{p}_j$ given by $\vec{n}_i$ and $\vec{n}_j$. Further we assume that a simple curve with low curvature is more plausible than a complicated curve. Because finding a curve with minimal curvature respecting the normal constraints at the start and end point is too costly to compute we propose to use a total of four possible Hermite curves instead as illustrated in figure 2 for the major constellations that have to be considered: planar, sharp crease, closeby sheets and a twist of the normals along the edge e_{ij}. We define the Hermite curves in a reference plane which equals the drawing plane in figure a-c) and is illustrated in light gray in figure d). In Figures 2 b)-d) one can see two red and two blue Hermite curves, a dark and a bright one for each color. The red Hermite curves interpolate the green normals $\vec{n}_i$ and $\vec{n}_j$, whereas the blue curves interpolate $\vec{n}_i$ and $-\vec{n}_j$. In the planar case in a) all four curves degenerate to lines, where one of the red curves includes no turning points and both of the blue curves one.

To decide whether we have to flip the normal or not, we compare the complexity of the Hermite curves. For each orientation of $\vec{n}_j$ we first select the Hermite curve with lower complexity, i.e. one red and one blue Hermite curve. The two selected curves are finally compared with respect to their complexities. We flip the normal $\vec{n}_j$ if the blue curve has lower complexity. Three questions remain to discuss in the next three paragraphs: how to define the reference plane, how to measure the complexity of a Hermite curve and how to define the unreliability cost.

Reference Plane For the definition of the Hermite curves we actually only need the normal direction of the reference plane, which we call the *reference normal*. To define the reference normal we have to consider three directions: $\vec{n}_i$, $\vec{n}_j$ and the direction $\hat{e}_{ij}$ along the edge e_{ij}. There are basically three cases:

1. all three directions are collinear (compare Figure 2 c). In this case we can choose any direction as reference normal that is orthogonal to common direction of normals and edge.
2. the three directions span a plane (compare Figure 2 a). Then we use the normal to this plane as reference normal.
3. the three directions span a volume (compare Figure 2 b and d).

In the last case we have to deal with sharp features and twists. The best way to deal with the twist case in Figure 2 d) is to define the reference normal as the cross product of $\vec{n}_i$ and $\vec{n}_j$. This simple approach has a problem though when the twist angle becomes very small, i.e. if normal $\vec{n}_j$ rotates around $\hat{e}_{ij}$ until it becomes nearly parallel to $\vec{n}_i$. Then the reference

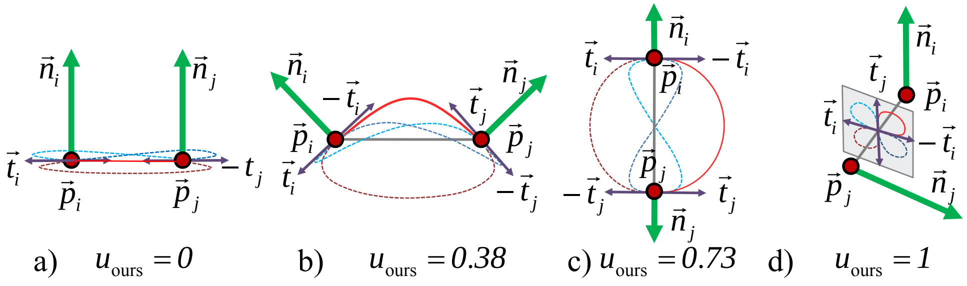

Figure 2: Illustration of our approach for the major cases a) planar, b) crease, c) closeby sheets, d) twisting surface. In case a-c) the reference plane equals the drawing plane in d) it is depicted in light gray. In each case we compare all four possible Hermite curves interpolating the projections of points $\vec{p}_i$,$\vec{p}_j$ and normals $\vec{n}_i$, $\vec{n}_j$ onto the reference plane. The light and dark red curves interpolate the constraints, whereas the light and dark blue curves interpolate the constraints with $\vec{n}_j$ flipped. In case a) the shapes of the four curves were marginally modified for better illustration. The correct curves would all lie on top of each other.

normal should rather point orthogonal to the plane spanned by $\hat{e}_{ij}$ and one or both of the to be interpolated normals. We therefore compute a second potential reference normal from the cross product between the average of the normals and $\hat{e}_{ij}$ and interpolate with the help of the dot product of $\vec{n}_i$ and $\vec{n}_j$ between the two potential reference normals:

$$\vec{n}_{\text{ref}} = \vec{n}_i \times \vec{n}_j + \left\langle \vec{n}_i, \vec{n}_j \right\rangle^2 \left(\frac{\vec{n}_i \pm \vec{n}_j}{\left|\vec{n}_i \pm \vec{n}_j\right|} \times \hat{e}_{ij} \right).$$

The sign of $\vec{n}_j$ in the normal average is chosen from the sign of the dot product between the normals, such that $\vec{n}_j$ is flipped if necessary. We took the scalar product of the normals to the power of two as this yielded the best results in the experiments. Finally, we only used the second term in the sum, if the normal average did not vanish.

Curve Complexity To construct the four Hermite curves in the reference plane we need to define their tangents. Each of the normals $\vec{n}_i$ and $\vec{n}_j$ projected to the reference plane defines two oppositely oriented tangents $\pm\vec{t}_i$ and $\pm\vec{t}_j$. We choose $\vec{t}_i$ and $\vec{t}_j$ by rotating the normals $\vec{n}_i$,$\vec{n}_j$ by 90 degree counterclockwise for $\vec{t}_i$,$\vec{t}_j$ and clockwise for $-\vec{t}_i, -tangj$. The two red Hermite curves in Figure 2 are defined with the tangent combinations $(\vec{t}_i, \vec{t}_j)$ and $(-\vec{t}_i, -\vec{t}_j)$, whereas the blue curves are defined by $(\vec{t}_i, -\vec{t}_j)$ and $(-\vec{t}_i, \vec{t}_j)$.

In order to make the construction scale independent we set the length of the tangents to twice the length of the edge e_{ij}. The factor of 2 was again determined by experimenting with the four major cases of Figure 2. Other choices for the factor like 1 or 4 failed in one of the cases completely.

For each of the four 2D Hermite curves $\vec{c}_{k=1\ldots4}(t)$ we define the complexity $C(\vec{c}_k)$ as the integral of the absolute value of the curvature along the curve, i.e.

$$C_k = C(\vec{c}_k) = \int |\kappa_k(s)| ds,$$

where κ_k denotes the curvature at parameter value t and ds the infinitesimal length element. As the curvature can also be written as the infinitesimal angle change $d\phi_k$ per length element ds, the complexity measure simplifies to

$$C_k = C(\vec{c}_k) = \int |d\phi_k|.$$

This means that we have to compute the total angle change that the tangent of the curve performs when traversing the Hermite curve. In case the Hermite curve does not have a turning point, the complexity is simply the angle between the tangents at $\vec{p}_i$ and $\vec{p}_j$. If the signed curvature at $\vec{p}_i$ or $\vec{p}_j$ is negative the angle must be computed clockwise otherwise counterclockwise. In case there are turning points, the curve can be split into monotonous parts at the turning points and the complexity is the sum of the complexity of the parts. A neccessary condition for a turning point of a Hermite curve $\vec{c}_k(t)$ is that the first derivative $\vec{c}_k'(t)$ with respect to t is parallel to the second derivative $\vec{c}_k''(t)$. Let $\vec{q}_1$, $\vec{q}_2$ be the start and end points and $\vec{m}_1$, $\vec{m}_2$ the corresponding tangents defining the 2D hermite curve $\vec{c}_k$ then $\vec{c}_k'(t)$ and $\vec{c}_k''(t)$ are given by

$$\begin{aligned}\vec{c}_k'(t) =& (6t^2-6t)\vec{q}_1 + (-6t^2+6t)\vec{m}_1 + \\ & (3t^2-4t+1)\vec{q}_2 + (3t^2-2t)\vec{m}_2 \\ \vec{c}_k''(t) =& (12t-6)\vec{q}_1 + (-12t+6)\vec{m}_1 + \\ & (6t-4)\vec{q}_2 + (6t-2)\vec{m}_2.\end{aligned}$$

Arranging the two vectors $\vec{c}_k'(t)$ and $\vec{c}_k''(t)$ into a 2x2 matrix, the determinant of this matrix must be zero for the two vectors being parallel:

$$\det\left(\vec{c}_k'(t)|\vec{c}_k''(t)\right) = 0$$

Expanding and simplifying the determinant expression results in a quadratic polynomial $p(t) = a_2t^2 + a_1t + a_0$ with coefficients

$$\begin{aligned}a_0 &= \det\left(\vec{m}_1|\vec{v}\right) \\ a_1 &= 2\cdot\det\left(\vec{m}_1|\vec{w}\right) \\ a_2 &= \det\left(\vec{v}|\vec{w}\right)\end{aligned}$$

where $\vec{v}$ and $\vec{w}$ are given by

$$\begin{aligned}\vec{v} &= 6(\vec{q}_1-\vec{q}_2) - 4\vec{m}_1 - 2\vec{m}_2 \\ \vec{w} &= 3(\vec{m}_1+\vec{m}_2) - 6(\vec{q}_1-\vec{q}_2).\end{aligned}$$

All cubic terms vanish during the simplifications. The roots of polynomial p lying in the interval $[0,1]$ are the split positions of interest. We don't have to care about saddle points because splitting the curve at a saddle point would not cause any problems.

Unreliability Measure Let C_{keep} be the smaller complexity of the two complexities of the red Hermite curves and C_{flip} the smaller complexity of the blue Hermite curves. Then we simply define the flip criterion and the unreliability measure as

$$\begin{aligned}f_{\text{ours}}(i,j) &= C_{\text{flip}} < C_{\text{keep}} \\ u_{\text{ours}}(i,j) &= \frac{\min\{C_{\text{flip}},C_{\text{keep}}\}}{\max\{C_{\text{flip}},C_{\text{keep}}\}}.\end{aligned}$$

By definition our unreliability criterion is in the range of $[0,1]$. In Figure 2 the unreliabilities of our method are shown. Especially important is that the twist case in d) results in the largest unreliability value, delaying this edge in the MST computation to the very end.

name	num. of vertices	num. of edges
tetraeder	9967	138012
triceratops	9540	131174
gearplate	12244	165968

Table 1: Sizes of the Riemannian Graphs. The number of vertices equals the number of points and normals as well as the edges in the MST plus one.

name	Hoppe	Xie	Our
tetraeder	10198	6856	1876
triceratops	2894	1366	879
gearplate	24330	15494	14309

Table 2: Numbers of edges in the Riemannian Graph for which the flip criterion failed.

4 Results

For the evaluation we use the three ground truth data sets depicted in Figure 3. The tetrahedron is used to validate the behavior at acute surface edges and corners. The triceratops model contains some sharp edges, low and high curvatures and a moderate number of nearby sheets e.g. at the three horns and the frill. The gear plate is the most complicated one. It contains a lot of sharp edges and many different configurations of nearby sheets. The data sets are created by sampling the surface of the corresponding polygonal meshes. Normals are taken from the triangle faces of these meshes to provide ground truth orientations. The number of vertices and edges of the corresponding Riemannian graphs G are given in Table 1. In all cases we used the 13 nearest neighbors to construct the graphs.

To evaluate the different criteria we performed several measurements. First we counted the number of edges within the Riemannian graph G for which the flip criterion failed with respect to the ground truth normals, see Table 2.

Next we counted the number of bad edges remaining within the MST in Table 3.

name	Hoppe	Xie	Our
tetrahedron	6	6	0
triceratops	24	4	0
gearplate	1598	46	46

Table 3: Number of edges in the MST for which the flip criterion failed.

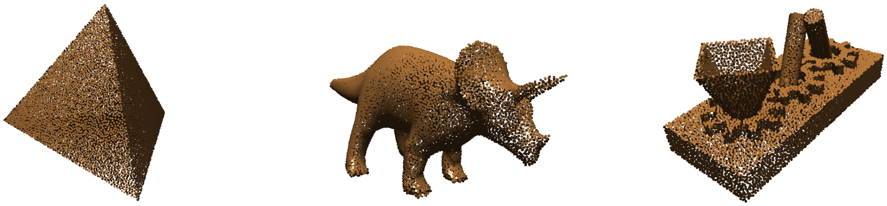

Figure 3: Tetraeder, triceratops and gearplate point cloud data used for evaluation purposes.

To analyze the unreliability costs proposed in the different approaches we have computed histograms comparing the relation between the number of good and bad edges for the different unreliabilities. The good edges are the ones where the flip criterion succeeds. A method suitable for the MST approach assigns low costs to good edges and high costs for bad edges. We only show the histograms of the bad edges in Figure 4 as the histograms for the good edges look very similar.

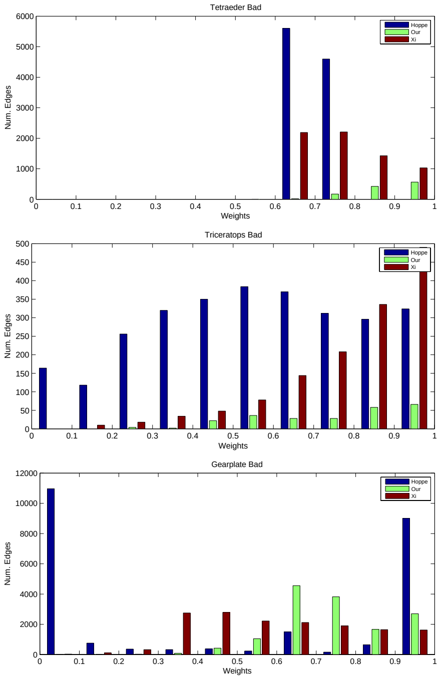

Figure 4: Histograms of the number of bad edges in the Riemannian Graph where the flip criterion fails plotted over the unreliability cost for the three models.

name	Hoppe	Xie	Our
tetraeder	2822	2822	0
triceratops	63	5	0
gearplate	2064	2139	2025

Table 4: Number of incorrectly flipped normals during the orientation propagation along the MST

	Hoppe	Xie	Our
time per edge	0.023 ms	0.026 ms	0.21 ms
tetraeder	3.17 s	3.59 s	30.0 s
triceratops	3.02 s	3.40 s	27.9 s
gearplate	3.82 s	4.32 s	34.7 s

Table 5: Timings for the computation of the unreliability measures for each edge in the Riemannian graph.

Finally, we counted the number of incorrectly flipped normals during the orientation propagation along the minimal spanning tree (see Table 4).

The computation time (see Table 5) for the unreliability measures of Hoppe and Xie are nearly equal between 0.02-0.03 ms per edge. Due to the more complex arithmetics involved in our criterion it is a linear factor of about 10 times slower. The time for constructing the Riemannian graph, computing the minimal spanning tree and for propagating the normal orientations is the same for all three methods and is less than a second for all models. All measurements are done with an Intel Core 2 CPU 2.33 GHz with 2 GB RAM. Our implementation is done in software and is not parallelized.

5 Discussion

All the models used in the results section were without noise. We believe that the analysis is still relevant also for 3D scans because all denoising procedures that we are aware of can be formulated in a way such that they do not rely on a consistent normal orientation and can therefore be performed before the normal orientation process.

We only selected models with sharp features or closeby sheets because smooth models work fine with all three methods. The gearplate model is especially difficult as its closeby sheets are undersampled. We chose this model to find the limitations of our method. The two other models worked perfectly with our new method. This is reflected in the fact that for all edges in the MST the flip criterion succeeded (compare Table 3) and therefore all normals could be oriented correctly. Both other approaches did not succeed on any of the models.

Table 4 shows that the failures of the other approaches on the triceratops model were not so severe as only very few normals received the wrong normal in the end. This good behavior always happens when one subtree of a wrongly flipped edge is very small. In case of the gearplate all methods failed and oriented a significant amount of normals wrongly.

The histograms in Figure 4 reveal again that our method is superior for the tetrahedron and triceratops models where the flip criterion fails for much less edges and these bad edges arise for higher unreliability costs only. The histogram for the bad edges of the gearplate model is quite different. The approach of Hoppe fails a lot for zero unreliability. This is due to the large number of edges connecting in between the close surface sheets of the funnel. Furthermore, does Hoppe's approach fail much less for unreliabilities of medium size compared to Xie's method and also to our method. In our method the unreliability criterion is clearly superior to the other reliability criteria as the edges where the criterion fails arise for larger unreliability costs. The total number of bad edges is only slightly less than in the approach of Xie.

6 Conclusion and Future Work

In this paper we gave a careful analysis of existing surficial orientation methods. We provided a better understanding of the underlying model assumptions and limitations of existing techniques especially in case of the criterion suggest by Xie. Finally we proposed an improved criterion to further reduce the possibility of orientation failures and demonstrate its effectiveness. In future work it would be useful to have a closer look at more sophisticated optimization strategies and to evaluate how robust the different methods are to noise.

References

[1] Nina Amenta and Marshall Bern. Surface reconstruction by Voronoi filtering. In *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry*, pages 39–48, New York, NY, USA, 1999. ACM.

[2] Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 330–339, 2004.

[3] Tamal K. Dey and Jian Sun. An adaptive mls surface for reconstruction with guarantees. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 43, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[4] H. Hoppe, T. DeRose, T. Duchamp, J. MCDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of ACM SIGGRAPH 1992*, volume 26, pages 71–78, 1992.

[5] Michael Kazhdan. Reconstruction of solid models from oriented point sets. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 73, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[6] D. Levin. Mesh-independent surface interpolation. *Advances in Computational Mathematics*, 2003.

[7] Viní cius Mello, Luiz Velho, and Gabriel Taubin. Estimating the in/out function of a surface represented by points. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 108–114, New York, NY, USA, 2003. ACM.

[8] N J Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. In *Internat. J. Comput. Geom. & Applications*, 2004. p. to appear.

[9] Yutaka Ohtake, Alexander Belyaev, and Marc Alexa. Sparse low-degree implicit surfaces with applications to high quality rendering,

feature extraction, and smoothing. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 149, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[10] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.

[11] M. Pauly, R. Kreiser, L. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. In *Proceedings of ACM SIGGRAPH 2003*, pages 641–650. ACM Press, 2003.

[12] Hui Xie, Kevin T. McDonnell, and Hong Qin. Surface reconstruction of noisy and defective data sets. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 259–266, Washington, DC, USA, 2004. IEEE Computer Society.

[13] Hui Xie, Jianning Wang, Jing Hua, Hong Qin, and Arie Kaufman. Piecewise C1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. *Visualization Conference, IEEE*, 0:13, 2003.