Point-based graphics 2006 [SPBG '06] : Eurographics/IEEE VGTC Symposium proceedings, Boston, Massachusetts, USA, July 29 - 30, 2006 / papers chaire Mario Botsch ... . - Aire-la-Ville : Eurographics Association, 2006. - S. 85-94. - ISBN 3-905673-32-0

Eurographics Symposium on Point-Based Graphics (2006)
M. Botsch, B. Chen (Editors)

Approximating Geodesics on Point Set Surfaces

M. R. Ruggeri, T. Darom[2], D. Saupe & N. Kiryati[2] †

Dept. of Computer and Information Science, University of Konstanz, Konstanz, Germany
[2] School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel

Abstract

We present a technique for computing piecewise linear approximations of geodesics on point set surfaces by minimizing an energy function defined for piecewise linear path. The function considers path length, closeness to the surface for the nodes of the piecewise linear path and for the intermediate line segments. Our method is robust with respect to noise and outliers. In order to avoid local minima, a good initial piecewise linear approximation of a geodesic is provided by Dijkstra's algorithm that is applied to a proximity graph constructed over the point set. As the proximity graph we use a sphere-of-influence weighted graph extended for surfel sets. The convergence of our method has been studied and compared to results of other methods by running experiments on surfaces whose geodesics can be computed analytically. Our method is presented and optimized for surfel-based representations but it has been implemented also for MLS surfaces. Moreover, it can also be applied to other surface representations, e.g., triangle meshes, radial-basis functions, etc.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation; I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations; I.3.5 [Computer Graphics]: Geometric algorithms, languages, and systems.

1. Introduction

In many fields, including cultural heritage, reverse engineering, architecture, and medical applications, 3D surfaces must be acquired from reality. The current 3D acquisition technologies produce large sets of 3D point clouds from which a proper and convenient representation of the real surface must be extracted for rendering, analysis and processing. In computer graphics surfaces are commonly represented by triangular meshes. However, in recent years a significant trend in computer graphics has been the shift towards point-sampled surfaces, especially for rendering. Pfister et al [PZvBG00] presented surfels as a powerful point-based surface representation scheme for rendering. Surfels are oriented circular or elliptical disks containing color information that describe surface characteristics within the neighborhood of each sample point. Neither connectivity nor other topology information is given. The importance of point-based representations and point-based primitives such as surfels, is rapidly increasing. During the past years several paradigms known from triangle meshes have been transferred to point-based geometry and many new techniques have been developed and optimized for point-based representations [KB04].

Point sets geometry represents the typical output of the current 3D acquisition techniques, which produce sets of noisy 3D point clouds that subsequently have to be processed for different purposes. The problem of computing geodesics, i.e., shortest paths on surfaces, arises in different applications such as robot motion planning [HPJS01, HP04], shape analysis [EK01] and geographical information systems. Furthermore, geodesics have been widely used in 3D surface analysis and processing, including parameterization [ZKK02, SWG*03], partitioning [KT03], flattening [GKK02], modelling [KCVS98, ZG04], segmentation [PC04], remeshing [OSG03], texture mapping [ZKK02], and also compression [MD04]. We propose a new technique

† This research was supported by the Kurt Lion Foundation. At Tel-Aviv University, it was supported also by MUSCLE: Multimedia Understanding through Semantics, Computation and Learning, a European Network of Excellence funded by the EC 6th Framework IST Programme. At Konstanz University, it was supported also by the DFG Graduiertenkolleg "Explorative Analysis and Visualization of Large Information Spaces". We also thank Anna Dowden-Williams for proofreading the paper.

for computing approximations of geodesics directly on noisy point set surfaces with outliers.

Given a surface Ψ represented by a set of N three dimensional sample points $\Pi = \{p_i \in \mathbb{R}^3, i = 1..N\}$ we compute an approximation of the geodesic path and the corresponding geodesic distance between two arbitrary points of Π on the surface Ψ.

Geodesics are defined on a surface. In our case the surface is given only indirectly by the set of 3D points Π. Since point based surfaces are typically acquired with 3D scanning techniques, we assume that our point set Π includes noise and outliers. In this scenario extracting a precise approximation of the real surface may be difficult and computationally expensive. However, several methods can be used to extract geometric surfaces from points clouds. The most used approaches extract implicit surface representations that locally approximates the real surface via moving least-squares (MLS) surfaces [ABCO*03, AK04, DS05, BH05], or radial basis functions (RBF) [CBC*01, OBS04, RTSD03]. Working directly with local approximations through implicit surfaces allows a rigorous formulation of the problem of computing geodesics on the real surface. Nevertheless, this approach can be too costly for several applications, such as rendering and other processing techniques, which need to compute such information on the fly. Surfels have been invented for rendering in order to provide 3D points with information about colors, orientations (normals), and their influence in covering the underlying surface. Such information can be extracted by analyzing the surface samples in the neighborhood of each point $p_i \in \Pi$ [PZvBG00, KB04, WK04, AA04]. Thus, the surfel set $S = \{s_i, i = 1..N\}$ extends the original point set Π and can be considered as a discrete approximation of the real surface Ψ.

We address the problem of computing piecewise linear approximations of geodesics on point set surfaces by minimizing an energy function defined for piecewise linear paths. The function considers path length, closeness to the surface for the nodes of the piecewise linear path and for the intermediate line segments. In order to avoid local minima, a good initial approximation is provided by Dijkstra's algorithm that is applied to a proximity graph constructed over the point set. Our method robustly handles noisy surfaces with outliers, i. e., typical output of 3D scanning frameworks. We implemented our method for two point based surface representations: surfels and MLS surfaces. Since surfels as a general point based primitive, are computationally easy to manage, suitable for interactive/multi-resolution systems [RL00], and are the main point based primitives for standard and high-quality rendering [PZvBG00, RL00, ZPKG02], our method is formulated and optimized for surfels. Nevertheless, our technique is general, since it can be directly extended to other surface representations, e.g., triangle meshes, RBF surfaces, etc.

We experimentally analyze the convergence and the precision of our method for noisy point samples of smooth surfaces whose geodesics can be computed analytically. We also compare the precision of our approach with the techniques presented by Hofer and Pottmann in [HP04].

2. Related work

Different algorithms are available for computing geodesics and measuring distances on surfaces. They are typically developed for high-dimensional manifolds and differ with respect to the particular surface representation. One approach to approximate geodesics is to apply a Dijkstra-like algorithm on a graph generated by a refined representation of surfaces (e.g. polyhedron surfaces [KS01, KS04]). Kiryati and Székely in [KS93] presented an algorithm for geodesic distance estimation on voxel-based surfaces. Kimmel and Kiryati in [KK96] proposed an algorithm for finding shortest paths in two stages. In the first stage the algorithm proposed in [KS93] is used to obtain an approximation of the globally shortest path. In the second stage the approximation is refined to a locally optimal path. Kimmel and Sethian in [KS98] proposed an efficient algorithm for computing geodesic distances and paths on triangulated manifolds based on a fast marching method. This approach has been widely used due to its speed and precision. Furthermore, different efficient variations of this method were implemented [NK02, Kir04]. Surazhsky et al. in [SSK*05] proposed an alternative technique to compute exact and approximate geodesics on triangle meshes. Their technique is based on the idea to unfold a set of adjacent triangles into a common 2D plane and extract geodesics by reconstructing the distance field associated to each mesh edge.

Geodesics for point clouds have been considered before by other authors. Memoli and Sapiro in [MS03] constructed an offset band of the point set surface consisting of the union of Euclidean balls centered at the given points. Geodesics were computed by applying the fast marching algorithm on a 3D Cartesian grid built inside the constructed band. The accuracy depends on the resolution of the 3D grid, and the execution time depends on the number of grid points. Their method is robust with respect to noise provided that it is bounded by the radii of the balls defining the offset band. Klein and Zachmann [KZ04] approximated geodesics as shortest paths on a geometric proximity graph over point clouds. They experimented with Delaunay and spheres-of-influence graphs (SIG) as proximity graphs and proposed an extended version of SIG. Hofer and Pottmann [HP04] proposed to compute geodesic curves as energy minimizing discrete curves constrained on a MLS surface. They proposed an efficient method for high dimensional constrained minimization of certain energy functional leading to geodesic curves. Even though the method is efficient, the precision of the final geodesic is sensitive to noise, since it relies on the MLS surface construction.

Our approach is related to the method of Hofer and

Pottmann [HP04] in the sense that it is characterized by an energy function minimization. However, we do not constrain our path on a specific surface like in, which is basically unknown in a point cloud, but rather we embed this constraint in the function itself, by using surfel disks (or MLS surfaces) as a tool to estimate the proximity of the path to the unknown surface.

3. Approximating geodesics by energy minimization

Let $S = \{s_i, i = 1..N\}$ be the surfel set approximating a surface Ψ. Let p_a and p_b be two distinct points in S, our goal is to find a piecewise linear path $P_g = v_0 v_1 .. v_{n+1}$ with $v_0 = p_a$ and $v_{n+1} = p_b$ approximating a geodesic path from p_a to p_b, i.e., the shortest path between p_a and p_b on Ψ. The geodesic distance $g(p_a, p_b)$ between p_a and p_b on Ψ is then approximated by the length of the path P_g.

As the exact surface Ψ is not known and only approximated by the surfel set S, we propose to approximate geodesic curves by curves close to the surfels. We do not require the control points $v_i, i = 1, .., n$ to be on the surfel disks. Thus, we seek a short path that is close to the surfel set and that accommodates the geometry of the underlying surface. For this purpose we propose to minimize an energy function taking these factors into account. We define our energy function $E_S : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ as a sum of three terms:

$$E_S(P) = L(P) + \alpha \cdot D_S(P) + \beta \cdot U_S(P), \qquad (1)$$

where $P = v_1 .. v_n$ is the sequence of control points of a path between $v_0 = p_a$ and $v_{n+1} = p_b$. The term $L(P)$ is the sum of the squared distances between consecutive points and is related to the path length:

$$L(P) = \sum_{i=0}^{n} \|v_{i+1} - v_i\|_2^2. \qquad (2)$$

This term serves to shorten the path. Moreover, considering equidistant control points v_i constrained to the ideal underlying surface, the term $L(P)$ can be assumed as the discretization of a functional whose minimization has been demonstrated to lead to a geodesic curve [PH05]. $D_S(P)$ represents a fidelity term and serves to keep P close to the surfel set S,

$$D_S(P) = \sum_{i=1}^{n} d_S^2(v_i) = \sum_{i=1}^{n} \min_j (d^2(v_i, s_j)), \qquad (3)$$

where n is the number of control points v_i, and $d^2(v_i, s_j)$ indicates the squared distance between the 3D point v_i and the disk of the surfel s_j. The term $D_S(P)$ brings control points v_i close to the surfels. In order to also ensure that line segments $v_i v_{i+1}$ are close to surfels (Figure 1) so that the path is geometrically correct with respect to the underlying surface the we introduce the term $U_S(P)$.

$$U_S(P) = \sum_{i=0}^{n} d_S^2(\hat{v}_i) \cdot \frac{d_S^2(\hat{v}_i)}{\|v_{i+1} - v_i\|_2^2}, \qquad (4)$$

where $\hat{v}_i = (v_{i+1} + v_i)/2$ is the midpoint between two consecutive sample points. The squared distance $d_S^2(\hat{v}_i)$ from $\hat{v}_i$ to S serves to keep the line segment $v_i v_{i+1}$ close to S. The ratio between $d_S^2(\hat{v}_i)$ and the squared length of the line segment $\|v_{i+1} - v_i\|_2^2$ puts more weight on line segments where curvature is high. $U_S(P)$ also regulates the distribution of control points v_i over the path P with respect to the path curvature (Figure 1). $d_S(\hat{v}_i)$ roughly approximates the mean of the distance to the ideal surface Ψ over the i-th path segment (highlighted area in Figure 1). Better estimates can be calculated. However, experiments on noisy surfaces showed that the gain in terms of precision of the whole method only was small, while increasing the computational complexity significantly.

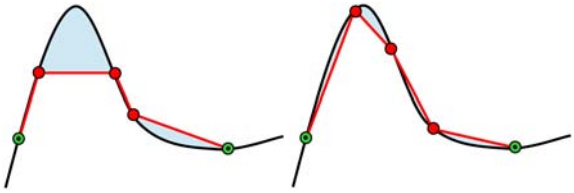


Figure 1: *2D view of two piecewise approximations of a geodesic between two points. The highlighted area represents a measure of the path proximity to the underlying surface (black curve).*

The parameters α and β influence the convergence speed of a numerical method for minimization of $E_S(P)$, as well as the quality of the resulting approximation of a geodesic. For paths passing across sharp edges the term $D_S(P)$ and especially the term $U_S(P)$ are crucial in order to achieve geodesic curves with good accuracy. As shown in Figure 2 the path obtained with β set to 0 (in orange) is smoother and farther from the surface than the path obtained by setting β to 2.0 (in dark-red). For surfaces with low curvature the term $U_S(P)$ can be almost neglected, since it does not strongly bias the convergence of the algorithm to a good solution. However, experiments suggested to set α and β to 0.7. Also on surfaces with sharp edges, these values have produced paths close to the true geodesic.

3.1. Computing an initial path

For our algorithm it is important to select a good initial path in order to keep our minimization procedure from falling into a local suboptimal minimum, especially in case of surfaces with high curvature (see Figure 2).

We aim at a fast initial approximation of the geodesic path. One possibility is to compute an approximation by applying the fast marching method on a coarse 3D grid built in an offset band of the point set surface [MS03]. Another possibility is to construct a weighted graph to describe surfel connectivity and use Dijkstra's algorithm to obtain a shortest path on the graph. We chose the second approach. From

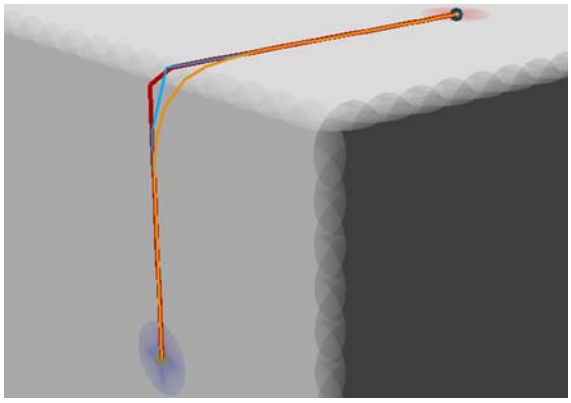
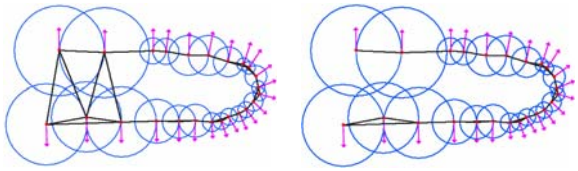


Figure 2: *Approximations of geodesic paths computed on a surfel-based model with sharp edges: the initial path computed as shortest path on the graph using Dijkstra's algorithm (in light-blue); the path computed with our method using the $U_S(P)$ term with $\beta = 2.0$ (in dark-red) and with $\beta = 0.0$ (in orange).*

the various graphs proposed in the literature [JT92] for point samples, we chose to adapt the spheres-of-influence graph (SIG) [KZ04] to surfels since it accommodates variations in the point sample density and produces a good description of non-smooth surfaces [JT92, KZ04]. Moreover, a SIG can easily be extended and optimized for surfels by using surfel properties to handle cases such as depicted in Figure 3.

The sphere centered at point sample (surfel position) p_i with radius given by the distance to the nearest neighbor is called the sphere of influence of p_i. The SIG is the graph with vertices p_i in which two vertices p_i and p_j, $i \neq j$, are connected by an edge e_{ij} if the corresponding spheres of influence intersect. Each edge e_{ij} is weighted by the Euclidean distance between the connected vertices.



(a) (b)

Figure 3: *(a) Canonical SIG, and (b) SIG extended with surfel orientations. 2D view.*

As shown in [KZ04], in case of noisy or irregular point clouds, a SIG may produce small clusters of connected points that are inter-cluster disconnected even though they are part of the same surface. To overcome this problem, different approaches that focus on modifying the radii of the spheres of influence can be applied. In [KZ04] the radius is set to the distance to the k-th nearest neighbor with $k > 1$. This may produce extraneous long edges in the graph requiring an additional edge pruning step. In our case, we benefit of the surfel-based representation by using the surfel attributes to extend our SIG. We define the radius of the spheres of influence to be equal to the corresponding surfel radii, since the radius of a surfel disk identifies the contribution of its sample point in representing the underlying surface. In addition, we require that two surfels connected by an edge have approximately the same orientation. Thus, we connect two vertices by an edge only if the dihedral angle between their surfel normal vectors is less than a certain threshold angle θ_t. Using this approach undesirable edges are discarded, see Figure 3.

The approximation of the initial geodesic path between p_a and p_b is computed using Dijkstra's algorithm. Searching for the shortest path between a fixed source and a fixed destination can be implemented more efficiently [LaV06]. Bi-directional search and A* algorithm provide better performance especially in the average case. However, in our case the time needed for computing the shortest path on the graph is very small with respect to the time needed by the energy minimization algorithm. The computation of the graph takes $O(|V|\log|V|)$ in time [JT92], where V is the set of vertices. Table 1 shows the execution time for computing our extended SIG with different sizes. The algorithm implemented with non-optimized C++ code has been executed on an Intel Pentium 4 2.80 GHz with 2GB of RAM running under a Windows XP Professional system.

Model	# surfels	Time (sec.)
Bunny [Sta]	35,948	3.50
Max Planck [PSh]	52,809	3.90
Chameleon [PSh]	101,685	4.73
Igea [PSh]	134,345	6.08
Lion [QSp]	183,408	7.16
Imperia(528KT) [KNR]	263,908	17,25
Imperia [KNR]	546,137	33.07
Buddha [QSp]	1,060,220	46.75
Dragon [PSh]	1,279,519	56.39

Table 1: *Execution time for computing weighted graphs*

3.2. Energy function minimization

Our goal is to minimize the energy function $E_S : \mathbb{R}^{3n} \rightarrow \mathbb{R}$, where n is the number of control points of a path P, and $3n$ is the number their coordinates. We considered three numerical optimization methods: the downhill simplex method, the Powell method, and the conjugate gradient method [PFTV92]. All methods start from an initial path P_0 of length g_0 obtained by the method of Section 3.1 and result in a path P^* of length g^*. In our experiments all the final paths were very close to the surfel set surface and their lengths

were shorter than the initial path P_0. Among these three optimization methods the Polak-Ribiere variant of the conjugate gradient method [PFTV92] proved to be the fastest and the most robust with respect to the number of control points. The additional cost to compute derivatives in the minimization process is compensated by the rapid convergence to a minimum especially when the initial solution is close to it, as in our case. Another advantage of this method is the low memory usage. In fact, the Powell method and the downhill simplex method require respectively a matrix of size $(3n)^2$ and $3n(3n+1)$, while the conjugate gradient only requires a vector of size $3n$. Considering its performance we chose the Polak-Ribiere variant of the conjugate gradient method to minimize our energy function.

The execution time of our energy minimizing method depends mainly on two factors: (1) the number of evaluations of the energy function and its gradient required to find a local minimum, and (2) the number of surfels of the model. The first factor is related to the characteristics of the minimization method, and depends on the proximity of the initial path to a local minimum of the energy function. In our case the conjugate gradient method has been optimized by choosing an adaptive bracketing criterion for Brent's line minimization [Mac, PFTV92]. The second factor is related to the estimation of the distance $d_S^2(v)$ between the control points and the surfel set surface S (Section 3). This estimation requires for each control point a search of the closest surfels in the model. This search is implemented through adaptive range-queries on a kd-tree constructed from the point set.

The number of control points involved in the minimization algorithm affects the convergence speed of the minimization method to a local minimum of the energy function as well as the precision of the final path with respect the true geodesic. We apply our method to the shortest path computed on the SIG with Dijkstra's algorithm, by adding one control point in the middle of those path segments that have endpoints in which the path has a curvature higher than a threshold. However, our method can also be applied iteratively starting from a low resolution initial path. In this case, the resolution of the subsequent resulting paths can be increased at each iteration to enhance the path quality and precision. With such a multi-resolution approach, we can balance the convergence speed of our minimization method with the precision of the energy minimizing path with respect to the real geodesic.

Figure 4 shows the result of our technique on the surfel-based model of Igea [PSh]. The path P^* minimizing our energy function (in dark-red) properly fits the underling surface and is shorter than the initial P_0 (in light-blue).

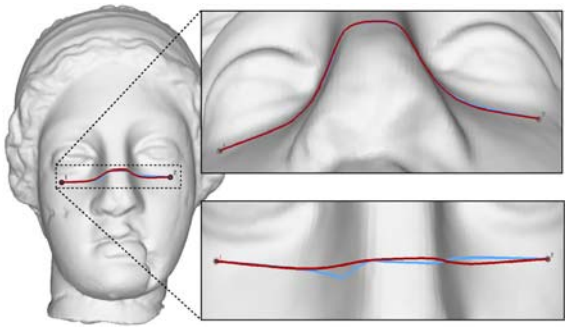


Figure 4: *Approximations of geodesic path computed with our method (dark-red) with respect to the initial path (light-blue) computed by applying Dijkstra's algorithm on the SIG (Section 3.1). Our path properly fits the underling surface (top-right) and is shorter than the initial path (bottom-right).*

4. Results

We present experimental results of our method on surfel set surfaces. Figure 9 shows several approximations of geodesics on surfel-based models of different size and geometry computed with our method. All surfel based models shown in this paper have been rendered with PointShop3D [ZPKG02].

We also consider noisy surfaces, whose noise is created by perturbing both surfel positions and surfel normals. Surfel positions are perturbed by shifting them along their surfel normal vectors. The offset of a surfel center is uniformly random in the interval $[-r_i/2, r_i/2]$ or $[-r_i, r_i]$, where r_i is the surfel radius. We identify these two cases as noise level 50% and 100%, respectively. Surfel normal vectors are perturbed by rotating them by a uniformly distributed azimuthal and polar angle. We use a maximal tolerance of 10 degrees for both angles.

Our method has been tailored for surfel based representation, but can be applied to other surface representations, e.g., triangle meshes, MLS/RBF surfaces, etc. In fact, our approach minimizes an energy function relying on estimations of the squared distances between the path sample points and the point set surface $d_S^2(v)$ (Section 3). We have implemented our technique also for MLS surfaces. The surface S is then given by the MLS surface fitted to the points cloud [ABCO*03]. The squared distance $d_S^2(v)$ is estimated as the squared Euclidean distance between the control point v and its projection on the MLS surface. The paths computed with this technique on the MLS surface have approximately the same characteristics of the paths computed on surfels (Figure 5, 6). However, our technique applied on MLS surfaces was slower than our method applied on surfels, due to the large number of MLS surface estimations and projections required by the method. Optimizations of this technique will be considered in future work.

We compare these two techniques with the method presented in [HP04] to compute piecewise geodesic curves; we call this the *HP method*. For this HP method we take as

initial path the same initial path of our methods, i.e., the shortest path computed on the SIG with Dijkstra's algorithm. We then project the points of this path onto the MLS surface, as is required by the HP method. Figure 5 shows a visual comparison of four approximations of geodesic paths obtained with: (1) our method applied on surfels; (2) our method applied on MLS surfaces; (3) the HP method; and (4) the initial path from which all methods started. Both our method on MLS surfaces and the HP method have been executed with the same MLS surface fitting parameters. The same paths are computed on the same models on which we have added noise of level 100% and also outliers (Figure 6). Figure 6 shows that our methods are robust with respect to noise and outliers. The three zooms on the right of the show that the paths computed with our technique have not been affected by noise and outliers, while the HP method produced a jagged path when noise/outliers were present. This effect that we also observed in other experiments, is due to the sensitivity of the MLS projection operator [ZPKG02] to noise and outliers. This effect can partially be reduced with a proper selection of the MLS surface fitting parameters. Selecting proper and robust parameters is a delicate task. In fact, we noticed that they depend on the surface geometry and especially on the noise. There is no guaranty that a proper setting of these parameters can be found, especially in case of extremely noisy surfaces with outliers, as depicted in Figure 6. In order to overcome this problem more sophisticated techniques of MLS surface fitting may be applied [AA04, KZ04, AK04, DS05].

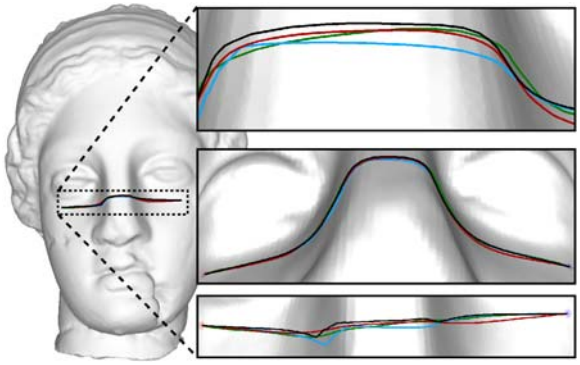


Figure 5: *Comparison of different approximations of geodesic paths computed on the surfel-based model of Igea [PSh] with different methods: initial path computed as shortest path on the graph with Dijkstra's algorithm (light-blue), path computed with our method on surfels (dark-red) and on MLS surface (dark-green), and path obtained with the HP method (black).*

A comparison of time performances of our method with respect to our implementation of HP method is reported in Table 2. Both algorithms have been executed on an Intel Pentium 4 2.80 GHz with 2GB of RAM running under a Windows XP Professional system. The execution of both methods has been stopped when the fractional difference between two consecutive paths was below to a certain threshold.

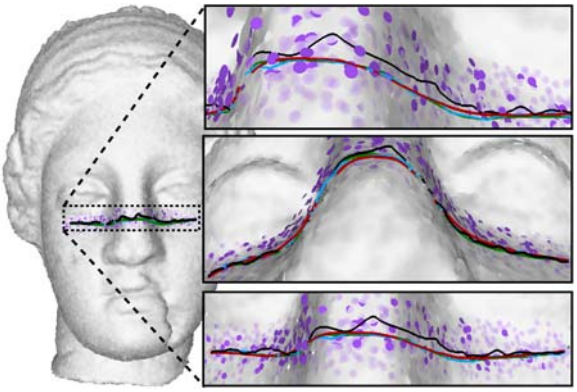


Figure 6: *Comparison of different approximations of geodesic paths computed on a noisy surfel-based model with outliers. Noise of level 100% and outliers (surfels in violet) have been added to the surfel-based model of Igea [PSh]. Outliers are generated by adding new noisy surfels with noise level 200% to the noisy model. Paths: initial path computed as shortest path on the graph with Dijkstra's algorithm (light-blue), path computed with our method on surfels (dark-red) and on MLS surface (dark-green), and path obtained with the HP method (black).*

Model	# surfels	Avg. # Pts	Our M.	HP
Igea	134345	74.55	19.37	5.68
Noisy Sphere	65544	68.55	3.44	1.83
Noisy Plane	10000	20.58	0.68	1.21
Noisy Cylinder	12060	37.04	7.11	2.05
ScanPlane	6580	30.81	0.69	0.57

Table 2: *Comparison of time performances of our method with respect to the HP method. The table reports the average execution time taken to compute 100 approximations of geodesic paths between 100 random point pairs. Times are reported in seconds. Column 3 reports the average number of sample points of the computed geodesic paths. The noisy surfaces have a noise level of 50%.*

4.1. Experimental convergence and error analysis

For an arbitrary surfel set surface it is difficult to establish whether the path computed with our method converges to the exact geodesic, since we need to know the original surface in parametric or implicit form and compare our results with the real geodesic curve. Computing geodesic curves on a general surface is also difficult and may be imprecise, since it requires solving differential equations with numerical methods [Car76]. However, for some surfaces geodesics can be

computed analytically. On a plane a geodesic between two points is a straight line segment. On a cylinder a geodesic is also a straight line segment when the cylinder is unrolled into a plane. On a sphere a geodesic is the shortest arc on the great circle passing through two points.

We tested the convergence and the precision of our method on three surfaces which have been appropriately sampled with surfels at different densities: a plane, a sphere, and a cylinder. The plane has been obtained by regularly sampling a square planar surface on a grid. The sphere has been obtained by regularly sampling a unit sphere with 65544 surfels. We regularly subdivided an octahedron seven times using Loop's scheme [Loo87], and we projected the vertices onto the sphere. The cylinder has been obtained by sampling a cylinder of 30 m diameter and 30 m height using an equiangular sampling on its circumference and an equidistant sampling on its axis.

In order to estimate the precision of our approach we consider two different errors. The first error indicates the precision of the approximated geodesic distance and is defined by $e_{geod}(g^*) = |g^* - g_a|/g_a$, i.e., the absolute relative error with respect to the exact geodesic distance g_a. We also consider the proximity of the computed path with respect to the exact geodesic path,

$$d_{path}(P,\hat{P}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_v^2(v_i,\hat{P})}, \qquad (5)$$

where $d_v^2(v_i,\hat{P})$ is the squared distance between the control point v_i of the path P and the piecewise linear path $\hat{P}$. The second error is then defined by $e_{path}(P^*) = max(d_{path}(P^*,P_a), d_{path}(P_a,P^*))/g_a$, i.e., the symmetric distance between the approximated geodesic path P^* and the exact geodesic path P_a divided by the length of P_a. In our case P_a is a continuous path, which is properly sampled at high density. This allows to compute $d_{path}(P_a,P^*)$ and to render P_a properly.

We statistically compare these two errors by considering the root mean square of the errors $e_{geod}(g^*)$ (EG) and $e_{path}(P^*)$ (EP). Table 3 reports the results of our methods and the HP method on the three test surfel set surfaces described above. On all these surfaces, our method converged to piecewise linear paths having small EG and EP errors, and that were very close to the exact geodesics. Furthermore, we observed that the MLS projection operator used in the HP method keeps the paths closer to the real surface than our method.

The convergence of our method to a good approximation of the exact geodesics is biased by the sampling of the test surfaces. Surfels linearly approximate a surface. A correct sampling is important to keep the approximation error under a certain threshold, especially for surfaces with high curvature. On the plane our method converges to paths that are closer to the exact geodesics than the paths computed on the sphere and the cylinder (e.g., see Table 3). On the sphere and the cylinder the errors $e_{geod}(g^*)$ and $e_{path}(P^*)$ decrease by increasing the sampling density, in case both the sphere and the cylinder are regularly or uniformly sampled. Thus, for numerical comparison, we experiment with five different surfel-based approximations of the unit sphere, and compute the errors EG and EP, which are reported in Table 4. The five approximation levels of the unit sphere are obtained by regularly subdividing an octahedron using Loop's scheme [Loo87] with vertices projected onto the sphere. We then associate a surfel to each of these vertices. The coarsest model obtained through 3 subdivision levels has 258 surfels. The finest model obtained through 7 subdivision levels has 65538 surfels. Figure 7 shows the surfel-based models of three unit spheres obtained through different subdivision levels. The path computed with our approach is shown with respect to the exact geodesic and the initial path.

		Our Met.		HP	
Model	# Surf.	EG%	EP%	EG%	EP%
Sphere	65544	8.0e-2	3.5e-2	4.0e-2	1.0e-2
Plane	10000	2.1e-5	1.1e-4	2.7e-5	2.7e-7
Cylinder	12060	4.5e-2	5.1e-3	4.6e-1	9.0e-2

Table 3: *Comparing the approximation error of our method with the HP method on different surfel set surfaces. Root mean square of the errors $e_{geod}(g^*)$ and $e_{path}(P^*)$ are reported in percentage as EG% and EP%, respectively. The approximation error depends on the sampling of the surface (see Table 4).*

# Subdiv. lev.	# surfels	EG%	EP%
3	258	6.20	2.162
4	1026	2.95	0.885
5	4098	1.02	0.275
6	16386	0.34	0.074
7	65538	0.08	0.035

Table 4: *Approximation error of our method on surfel-based models of unit spheres with different samplings. Spheres are sampled regularly using the Loop's scheme at different subdivision levels. Root mean square of the errors $e_{geod}(g^*)$ and $e_{path}(P^*)$ are reported in percentage as EG% and EP%, respectively.*

4.2. Robustness with respect to noise

We present experimental results of our method on noisy planes, noisy spheres, and noisy cylinders. The noisy surfaces are created by adding random noise of level 50% or 100% (see Section 4) to the test surfaces used in the experiments reported in Table 3. We also consider a planar surface scanned with a 3D Scanner Minolta VI-900, which

Model	NL	# Surf.	Our Met. EG%	Our Met. EP%	HP EG%	HP EP%
Sphere	50	65544	0.49	0.25	8.50	0.10
Sphere	100	65544	0.88	0.31	22.20	0.15
Plane	50	10000	0.019	0.097	0.12	0.24
Plane	100	10000	0.023	0.11	0.49	0.48
Cylind.	50	12060	0.69	0.46	2.50	0.94
Cylind.	100	12060	1.04	0.71	16.38	3.70
ScanPlane		6580	0.15	0.25	0.40	0.33

Table 5: *Comparing the approximation error of our method with the HP method on noisy surfaces. Root mean square of the errors $e_{geod}(g^*)$(EG) and $e_{path}(P^*)$(EP) are reported in percentage as EG% and EP%, respectively. The second column reports the noise level in percentage. Results on the original surfel-based models without noise are reported in Table 3. ScanPlane is a planar surface scanned with a 3D Scanner Minolta VI-900.*

presents the typical noise given by laser light-stripe triangulation range-finders.

We compare the precision of our method with respect to the HP method, by computing the errors EG and EP for both methods. Table 5 shows the result of our comparison. Figure 8 shows the approximation of a geodesic path computed with our method on three noisy surfaces, together with the exact geodesic curve, and the initial path computed on the SIG. Our method demonstrates a better robustness with respect to noise than the HP method. With our method we have obtained non-jagged paths very close to the exact geodesics. However, on both noisy spheres our method has a EP error greater than the HP method even though its EG error is smaller (Table 5). The reason is that the HP method generated jagged paths closer to the exact geodesic curves than the paths computed with our method, even though their lengths were greater than the exact geodesics, and those of our paths.

The EP error of our method depends on the proximity of the initial path to the exact geodesic. The quality of the initial paths in this sense can be slightly improved by increasing the number of edges of the SIG through the parameter k (see Section 3.1). Other approaches can be used like the fast marching method applied on a coarse 3D grid built in an offset band of the point set surface, presented by Memoli and Sapiro in [MS03].

5. Conclusions

We have presented a technique for computing piecewise linear approximations of geodesics on point set surfaces by minimizing an energy function defined for piecewise linear paths. The function considers path length, closeness to the surface for the nodes of the piecewise linear path and for the intermediate line segments. In order to avoid local minima, our approach needs to start from a good initial approximation of a geodesic. We propose to compute this approximation as the shortest path between the geodesic endpoints on an extended sphere-of-influence built over the point set.

A study of experimental convergence of our method has been provided both on noisy surfaces and on surfaces without noise. Our experiments show that our method is robust with respect to noise and outliers. A comparison of our method with the HP method [HP04] has been provided. In general, on noisy surfaces our method proved to be more precise and reliable than the HP method.

Although our technique has been designed for surfels, it has been implemented also for MLS surfaces. The results with MLS surfaces are similar to those with surfel-based surfaces. Optimizations of our method for MLS surfaces will be considered in future work. However, our technique is general in the sense that it can be applied to other surface representations, where the distance between a 3D point and the surface can be estimated, e.g., triangle meshes, RBF surfaces, etc.

Our method is suitable to compute good approximations of geodesics directly on noisy surfaces with outliers, i.e., the typical output of 3D scanning systems. It also presents multi-resolution features. In fact, it can be applied iteratively on initial paths at different resolutions, in order to improve the precision of the method and its execution time, especially for long geodesics. However, the time complexity of our method is still too high and hence not feasible for the computing of large numbers of geodesics, especially on large data sets. Future work will be oriented towards developing the multi-resolution characteristics of our method by considering multi-resolution curves representations, e.g., curves based on wavelets or hierarchical B-splines.

References

[AA04] ADAMSON A., ALEXA M.: On normals and projection operators for surfaces defined by point sets. In *Proc. EG Symp. on Point-Based Graphics* (2004), pp. 149–156.

[ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Trans. on Visualization and Computer Graphics 9*, 1 (2003), 3–15.

[AK04] AMENTA N., KIL Y. J.: Defining point-set surfaces. *ACM Trans. on Graphics 23*, 3 (2004), 264–270.

[BH05] BREMER P.-T., HART J. C.: A sampling theorem for mls surfaces. In *Proc. EG Symp. on Geometry Processing* (2005).

[Car76] CARMO M. P. D.: *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01* (2001), pp. 67–76.

[DS05] DEY T. K., SUN J.: An adaptive mls surface for reconstruction with guarantees. In *Proc. EG Symp. on Geometry Processing* (2005), pp. 43–352.

[EK01] ELAD A., KIMMEL R.: Bending invariant representations for surfaces. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (2001), vol. 1, pp. 168–174.

[GKK02] GROSSMANN R., KIRYATI N., KIMMEL R.: Computational surface flattening: A voxel based approach. *IEEE Trans. Pattern Anal. Mach. Intell. 24* (2002), 433–441.

[HP04] HOFER M., POTTMANN H.: Energy-minimizing splines in manifolds. *ACM Trans. on Graphics 23*, 3 (2004), 284–293.

[HPJS01] HU J., PRANDINI M., JOHANSSON K. H., SASTRY S.: Hybrid geodesics as optimal solutions to the collision-free motion planning problem. In *Hybrid Systems: Computation and Control : 4th International Workshop* (2001), pp. 305–318.

[JT92] JAROMCZYK J., TOUSSAINT G.: Relative neighborhood graphs and their relatives. *P-IEEE 80* (1992), 1502–1517.

[KB04] KOBBELT L., BOTSCH M.: A survey of point-based techniques in computer graphics. *Computers & Graphics 28*, 6 (2004), 801–814.

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH '98* (1998), pp. 105–114.

[Kir04] KIRSANOV D.: *Minimal Discrete Curves and Surfaces*. PhD thesis, Applied Math., Harvard University, 2004.

[KK96] KIMMEL R., KIRYATI N.: Finding shortest paths on surfaces by fast global approximation and precise local refinement. *Int. Journal of Pattern Recogn. and Artif. Intellig.. 10* (1996), 643–656.

[KNR] Konstanz 3d model repositor.y. http://www.inf.uni-konstanz.de/cgip/projects/surfac/.

[KS93] KIRYATI N., SZEKELY G.: Estimating shortest paths and minimal distances on digitized threedimensional surfaces. *Pattern Recognition 26* (1993), 1623–1637.

[KS98] KIMMEL R., SETHIAN J. A.: Computing geodesics on manifolds. In *Proc. of National Academy of Sciences* (1998), vol. 95, pp. 8431–8435.

[KS01] KANAI T., SUZUKI H.: Approximate shortest path on a polyhedral surface and its applications. *Computer-Aided Design 33*, 11 (2001), 801–811.

[KS04] KAGEURA M., SHIMADA K.: Finding the shortest path on a polyhedral surface and its application to quality assurance of electric components. *Journal of Mechanical Design 126*, 6 (Nov. 2004), 1017–1026.

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graphics* (2003), 954–961.

[KZ04] KLEIN J., ZACHMANN G.: Point cloud surfaces using geometric proximity graphs. *Computers and Graphics 28*, 6 (Dec. 2004), 839–850.

[LaV06] LAVALLE S. M.: *Planning Algorithms*. Cambridge University Press, 2006.

[Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. PhD thesis, CM.S. Mathematics, University of Utah, 1987.

[Mac] Macopt: Conjugate gradient optimizer. http://wol.ra.phy.cam.ac.uk/mackay/c/macopt.html.

[MD04] MOENNING C., DODGSON N. A.: Intrinsic point cloud simplification. In *GraphiCon* (2004).

[MS03] MÉMOLI F., SAPIRO G.: Distance functions and geodesics on points clouds. In *IEEE Workshop on Variational Geometric and Level Set Methods in Computer Vision* (2003).

[NK02] NOVOTNI M., KLEIN R.: Computing geodesic distances on triangular meshes. In *WSCG'2002* (2002), pp. 8431–8435.

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: 3d scattered data approximation with adaptive compactly supported radial basis functions. In *Shape Modeling International 2004* (2004), pp. 31–39.

[OSG03] OREN S., SHEFFER A., GOTSMAN C.: Geodesic-based surface remeshing. In *Proc. of 12th International Meshing Roundtable* (2003), pp. 189–199.

[PC04] PEYRÉ G., COHEN L.: Surface segmentation using geodesic centroidal tesselation. In *3DPVT 2004* (2004).

[PFTV92] PRESS W. H., FLANNERY B. P., TEUKOLSKY S. A., VETTERLING W. T.: *Numerical Recipes in C: The Art of Scientific Computing, 2nd edition*. Cambridge University Press, 1992.

[PH05] POTTMANN H., HOFER M.: A variational approach to spline curves on surfaces. *Comput. Aided Geom. Design 22*, 7 (2005), 693–709.

[PSh] Pointshop3d repository. http://graphics.ethz.ch/pointshop3d/download.html.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: Surface elements as rendering primitives. In *ACM SIGGRAPH 2000* (2000), pp. 335–342.

[QSp] The qsplat multiresolution point rendering system repository. http://graphics.stanford.edu/software/qsplat/models/.

[RL00] RUSINKIEWICZ S., LEVOY M.: Qsplat: a multiresolution point rendering system for large meshes. In *ACM SIGGRAPH 2000* (2000), pp. 343–352.

[RTSD03] REUTER P., TOBOR I., SCHLICK C., DEDIEU S.: Point-based modelling and rendering using radial basis functions. In *ACM Graphite 2003* (2003), pp. 111–118.

[SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Trans. on Graphics 24*, 3 (2005), 553–560.

[Sta] The stanford 3d scanning repository. http://www.graphics.stanford.edu/data/3Dscanrep/.

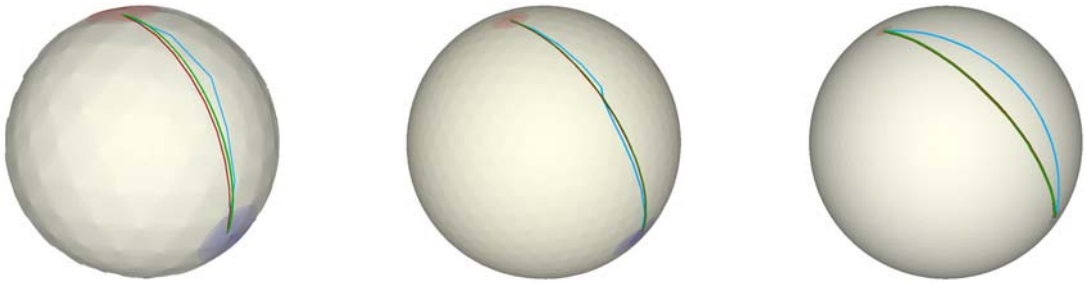
[SWG*03] SANDER P., WOOD Z., GORTLER S., SNYDER J., HOPPE H.: Multi-chart geometry images. In *Proc. EG Symp. on Geometry Processing* (2003), pp. 146–155.

[WK04] WU J., KOBBELT L.: Optimized sub-sampling of point sets for surface splatting. *Computer Graphics Forum 23*, 3 (2004), 643–652.

[ZG04] ZELINKA S., GARLAND M.: Similarity-based surface modelling using geodesic fans. In *Proc. EG Symp. on Geometry Processing* (2004), pp. 209–218.

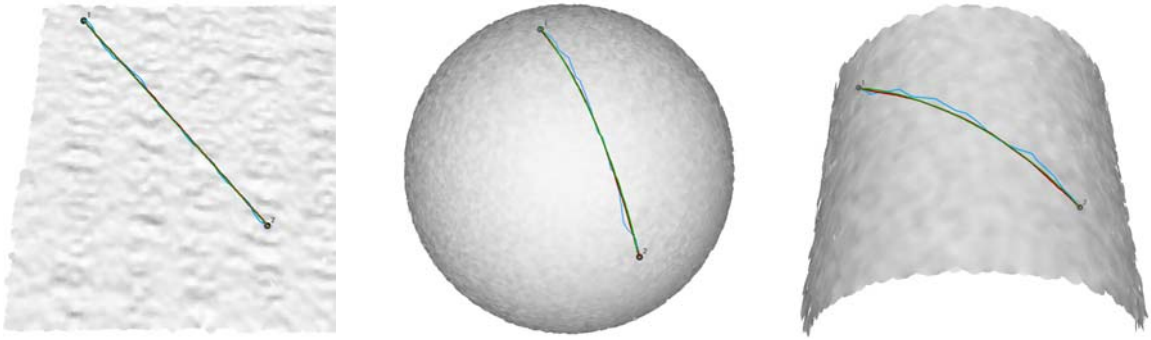
[ZKK02] ZIGELMAN G., KIMMEL R., KIRYATI N.: Texture mapping using surface flattening via mds. *IEEE Trans. on Visualization and Computer Graphics 8* (2002), 198–207.

[ZPKG02] ZWICKER M., PAULY M., KNOLL O., GROSS M.: Pointshop 3d: an interactive system for point-based surface editing. *ACM Trans. on Graphics 21*, 3 (2002), 322–329.



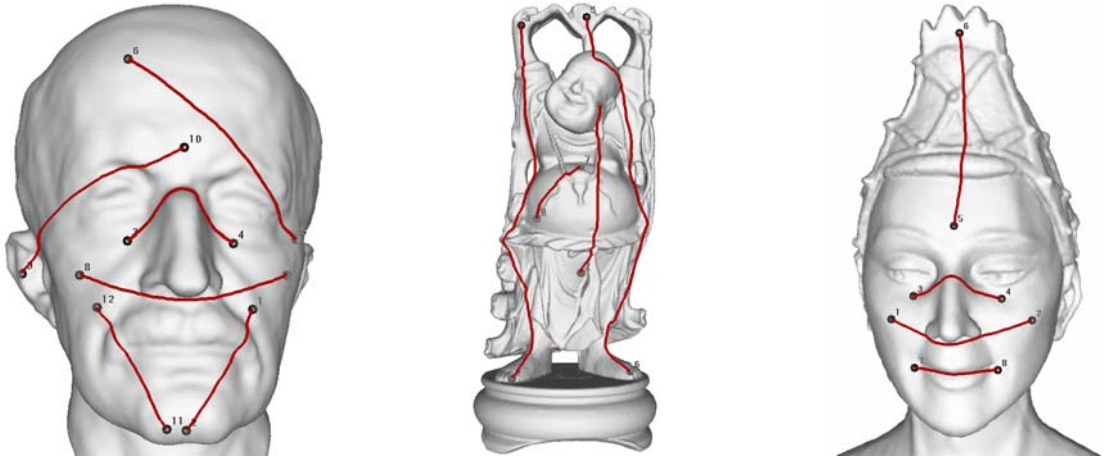
(a) (b) (c)

Figure 7: *(a) Result of our method on surfel-based models of unit spheres obtained from different subdivision levels using the Loop's scheme (3-(a), 4-(b), 5-(c)). Exact geodesic path computed analytically (green), Dijkstra's shortest path computed on the graph (light-blue), and geodesic path approximation computed with our method (dark-red).*



(a) (b) (c)

Figure 8: *Comparing the exact geodesic path (green) with the geodesic path approximation computed with our method (dark-red), and the initial solution computed on the graph with Dijkstra's algorithm (light-blue) on different noisy surfaces: a plane (a) a sphere (b), and a cylinder (c).*



(a) (b) (c)

Figure 9: *(a) Approximations of geodesics computed with our method on surfel-based models: Max Planck [PSh] 52,809 surfels (a); Buddha [QSp] 1,060,220 surfels (b); Zoom of Imperia [KNR] 546,137 surfels (c).*