

MESHLESS GEOMETRIC SUBDIVISION

By

Carsten Moenning

Facundo Mémoli

Guillermo Sapiro

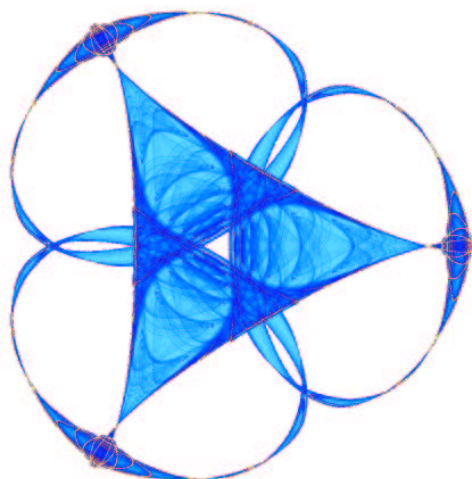
Nira Dyn

and

Neil A. Dodgson

IMA Preprint Series # 1977

(April 2004)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
514 Vincent Hall
206 Church Street S.E.
Minneapolis, Minnesota 55455-0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: <http://www.ima.umn.edu>

Meshless geometric subdivision

Carsten Moenning
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
cm230@cl.cam.ac.uk *

Facundo Mémoli
Electrical and Computer Engineering Department
University of Minnesota
Minneapolis, MN 55455, USA
memoli@ece.umn.edu

Guillermo Sapiro
Electrical and Computer Engineering Department
University of Minnesota
Minneapolis, MN 55455, USA
guille@ece.umn.edu

Nira Dyn
School of Mathematical Sciences
Tel Aviv University
Tel Aviv 69978, Israel
niradyn@post.tau.ac.il

Neil A. Dodgson
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
Neil.Dodgson@cl.cam.ac.uk

Abstract

Point-based surface processing has developed into an attractive alternative to mesh-based processing techniques for a number of geometric modeling applications. By working with point cloud data directly, any processing is based on the given raw data and its underlying geometry rather than any arbitrary intermediate representations and generally artificial connectivity relations. In this paper, we introduce the notion of meshless, or point cloud, subdivision by extending concepts from recursive mesh subdivision to the point cloud case. We are primarily concerned with showing the conceptual viability of this idea and propose a first geometric meshless subdivision framework. By replacing the role of mesh connectivity by intrinsic point proximity information and devising a meshless geodesic subdivision operator, we avoid the costly surface reconstruction, simplification and potential remeshing preprocessing steps typically required for supporting mesh-based subdivision, steps which are in general not directly related to the underlying object geometry. Furthermore, the maintenance of any global combinatorial data structure such as a mesh connectivity graph is not required. This property also makes our approach relatively easily extensible to the processing of point-based representations of higher-dimensional sur-

faces. Apart from introducing the idea of meshless subdivision, our main contributions are, firstly, a first meshless geodesic subdivision operator. Secondly, we present a new method for the computation of geodesic weighted averages on manifold surfaces, which are at the heart of our point cloud subdivision framework.

1 Introduction

Point-based editing, free-form and multiresolution modelling and visualisation have developed into viable alternatives to mesh-based processing methods [ABCO*03, PGK02, PKKG03, ZPvBG01, ZPKG02]. With numerous applications in medical imaging, reverse engineering, cultural heritage, geoscience, etc. acquiring point sets of significant density from three or higher-dimensional manifold surfaces, it is particularly attractive to be able to work with point-sampled geometry directly.

In the context of mesh-based surface processing, subdivision has developed into a powerful and widely-used tool for the free-form design and representation of smooth surfaces. These schemes use a subdivision operator which is recursively applied to a coarse base mesh. As a result, a sequence of refined meshes is obtained which quickly converges to a smooth limit surface. For many applications, the initial base mesh is

*The author performed this work whilst visiting the University of Minnesota.

obtained by costly polygonisation of point-sampled geometry. Highly dense point sets thereby translate into excessively dense polygonal models with arbitrary connectivity. This is generally followed by another costly mesh simplification step and often the need for remeshing ([ACSD*03] and the references therein) to enforce subdivision connectivity of the base mesh. Ease of manipulation tends to be further affected by the inherent overhead of mesh data structure maintenance and traversal.

This paper presents a first framework for meshless, or point cloud, subdivision. That is, we propose to avoid the consideration of mesh connectivity graphs and instead to work with the given point-sampled geometry directly and intrinsically throughout. We undertake a first step towards such an extension of recursive mesh subdivision to point clouds by introducing a meshless geodesic subdivision operator reminiscent of widely-used mesh subdivision operators. More specifically, we replace mesh connectivity by intrinsic proximity information and introduce meshless subdivision rules using geodesic centroids of local intrinsic point neighbourhoods. Although some of these geometric operations may be approximated in an Euclidean context when working with large sampling densities, regular meshes and very local subdivision rules, working with the point-sampled geometry directly and intrinsically has the benefit of not requiring any non-geometric preprocessing steps. Furthermore, an intrinsic point cloud subdivision approach is more generally applicable in that geodesic averaging rules can be non-local and do not vary with the particular type of data (mesh) representation used.

The focus of this paper is on highlighting the conceptual viability of meshless subdivision by putting forward a first suitable framework. Our ideas on the theoretical analysis of meshless geodesic subdivision schemes will be reported elsewhere.

Both our intrinsic point cloud simplification method for the determination of a base point set and our geodesic centroid technique are based upon the intrinsic distance mapping algorithm for point clouds presented in [MS03]. Following a brief overview of related work in Section 2, we therefore summarise the main aspects of this technique in Section 3. Section 4 presents the various elements of our intrinsic meshless subdivision framework. Section 5 gives experimental results. In the concluding Section 6, we briefly remark on our idea for the theoretical analysis of intrinsic meshless subdivision schemes.

2 Related work

We start with providing a brief summary of the ideas underpinning mesh-based subdivision of surfaces in \mathbb{R}^3 . The overview is motivational in nature, for a thorough

formal treatment of mesh subdivision, see [DL02].

Following the notation of [DL02], surface subdivision schemes consist of a subdivision operator S recursively applied to control nets $N_l = N(V^l, E^l, F^l)$ of arbitrary topology, with $l \in \mathbb{Z}_0$ denoting the subdivision level, V representing a set of control vertices in \mathbb{R}^3 and E and F describing the topological relations in the form of edges and faces respectively. That is, the iterative application of a subdivision scheme generates a sequence

$$N_{l+1} = SN_l.$$

Define the surface $X \subseteq \mathbb{R}^3$ as the limit surface of a scheme given that

$$\lim_{l \rightarrow \infty} d_H(V^l, X) = 0,$$

with

$$d_H(X, Y) = \max\{\sup_{y \in Y} \inf_{x \in X} \|x - y\|_2, \sup_{x \in X} \inf_{y \in Y} \|x - y\|_2\}$$

representing the Euclidean Hausdorff distance between the closed subsets $X, Y \subset \mathbb{R}^3$. Starting with the coarse base net N_0 , at each iteration, new control vertices are inserted and connected according to the scheme's refinement rule. Control vertices are then re-positioned following the operator's geometric averaging rule. Both the refinement and the geometric rule give the position of control vertices in N_{l+1} in the form of weighted averages of topologically neighbouring vertices in N_l . The careful choice of these rules in relation to the valency of the control vertex under consideration guarantees the convergence of the scheme to a limit surface of provable continuity. Note that for the subdivision schemes in the literature the convergence is in each component and in the uniform norm. Convergence in the Hausdorff distance is not yet achieved.

Not every existing mesh subdivision operator allows for such a simple distinction between a topological refinement and a geometric averaging rule applicable at all points. However, those that do allow for this kind of distinction, include the most widely-used schemes. For example, Loop [Loo87] subdivision for triangular control nets may be cast in this form. In the case of Loop subdivision, the refinement rule consists of the insertion of a new point at the midpoint of every edge, while the geometric rule, applicable at all points in the new mesh, smoothes the locations of the points by weighted averaging of their topological neighbours in the refined mesh.

In this paper, we propose to replace this use of mesh connectivity by intrinsic proximity information and formulate meshless refinement and geometric averaging rules in the form of weighted geodesic centroids of local neighbourhoods. As a result, we obtain a meshless subdivision scheme reminiscent of mesh-based subdivision operators.

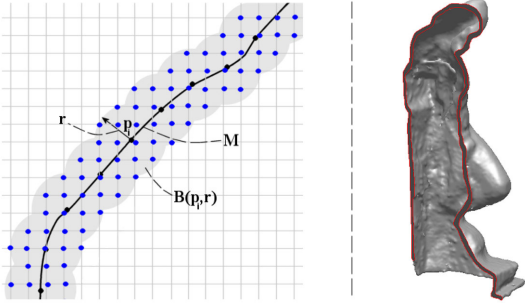


Figure 1: Intrinsic distance mapping using Fast Marching for point clouds operates in an offset band consisting of the union of balls $B(p_i, r)$ centred at (black) points p_i of the surface M (left). Only those (blue) grid points falling inside the offset band are considered during processing. Cross-sectional view of a constant radius offset band for the Michelangelo Youthful data set (right).

To the best of our knowledge, there exists only one previous piece of work which touches upon this notion of meshless subdivision. In Fleishman et al. [FCOAS03], the authors generate progressive levels-of-detail of point clouds by transferring the mesh-based idea of subdivision displacement maps to the point cloud case. They devise a point cloud simplification method for the generation of a base point set and present both a projection and a local, uniform upsampling operator with the help of local surface reconstruction using Moving Least Squares [ABCO*03]. By contrast, we are not aiming to mimic the principle of mesh-based subdivision displacement mapping for point-based surface representations but rather are interested in transferring the idea of mesh subdivision to the point cloud case.

In the following, we propose a first intrinsic framework for the meshless subdivision of a point cloud P_l to a refined point cloud P_{l+1} , with l defined as above. We start our discussion with a summary of the intrinsic distance mapping algorithm [MS03] at the heart of this intrinsic framework. This summary is followed by the presentation of the intrinsic meshless subdivision framework itself.

3 Intrinsic distance mapping across point clouds

We summarise the extension of the well-known original Fast Marching level set method [HPCD96, Set99, Tsi95] to the case of surfaces in point cloud form as introduced in [MS03]. Our review is necessarily terse, presenting just the key results. For full details, see [MS03].

Let $P = \{p_1, p_2, \dots, p_n\}$ denote a set of points, or point cloud, acquired from a smooth, compact manifold

surface M in $m \geq 3$ dimensions. Define the r -offset Ω_p^r as the union of Euclidean balls with radius r centred at points $p_i \in P$

$$\Omega_p^r := \bigcup_{i=1}^n B(p_i, r) = \{x \in \mathbb{R}^m : d(p_i, x) \leq r\},$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance function (Figure 1). To approximate the weighted intrinsic distance map originating from a source point $q \in M$ on M , Mémoli and Sapiro [MS03] suggest computing the Euclidean distance map in Ω_p^r . That is

$$|\nabla_M T_M(p)| = F(p), \quad (1)$$

for $p \in M$ and with boundary condition $T_M(q) = 0$ is approximated by

$$|\nabla T_{\Omega_p^r}(p)| = \tilde{F}(p), \quad (2)$$

for $p \in \Omega_p^r$ and boundary condition $T_{\Omega_p^r}(q) = 0$. \tilde{F} represents the (smooth) extension of the propagation speed F on M into Ω_p^r . $T(p)$ denotes the arrival time at p of the front originating from q and ∇_M and ∇ represent the intrinsic and the Euclidean gradient operators respectively. The problem of computing an intrinsic distance map is therefore transformed into the problem of computing an extrinsic (Euclidean) distance map in the tubular neighbourhood Ω_p^r around the surface, i.e. in an Euclidean manifold with boundary. In [MS03], the authors prove uniform probabilistic convergence between these two distance maps and thus show that the approximation error between the intrinsic and extrinsic distance maps is of the same theoretical order as that of the Fast Marching algorithm for both noise-free and noisy point clouds (assuming noise to be bounded from above by r). The Fast Marching method can therefore be used to approximate the solution to (2) in a computationally optimal manner and without the need for any prior surface reconstruction by only slightly modifying the original Cartesian Fast Marching technique to deal with bounded spaces. The relatively straightforward implementation of this technique consists of, firstly, computing the offset band Ω_p^r , followed by the application of standard Cartesian Fast Marching restricted to Ω_p^r . For more implementation details, see Mémoli and Sapiro [MS03].

This approach underpins both the point cloud simplification technique for the generation of a base point set and our geodesic computation algorithm presented as parts of our meshless subdivision framework in the following section.

4 Intrinsic meshless subdivision

We start with a brief summary of the intrinsic point set simplification method utilised for the computation of a

base point set P_0 . This is followed by the consideration of a suitable intrinsic neighbourhood concept. The section concludes with the presentation of the meshless subdivision operator and our method for the computation of geodesic centroids.

4.1 Intrinsic point cloud simplification

Depending on the method used for the acquisition of a point cloud representation of a surface, the resulting point-sampled geometry might be extremely dense. We simplify any such input point cloud P to a coarser base point set P_0 in a preprocessing step. This operation is to be performed subject to a minimum density condition to support the meaningful computation of geodesic centroids across P_0 . Note in this context that in contrast to mesh subdivision preprocessing, this simplification step is fully geometric in nature.

As regards the particular simplification technique used, Pauly et al. [PGK02] introduce a number of point cloud simplification methods by adapting various widely used mesh simplification techniques to the point cloud setting. Although their particle simulation-based simplification method represents an interesting alternative, we opt for the farthest point simplification scheme presented in [MD03] instead. This choice is due to the method’s superior efficiency, its purely intrinsic nature, its simple density control and its close relationship with a useful (intrinsic) local proximity concept (Section 4.2).

The point cloud simplification method [MD03] exploits the observation that the progressive farthest point sampling of a point cloud P may be implemented by incremental intrinsic (bounded) Voronoi diagram, $BVD(P)$, computation [MD03, OBS00]. This incremental computation is performed following an expanding waves view: In analogy to the dropping of pebbles in still water, circular fronts move across the surface from the point of impact. The locations where wave fronts meet or leave the domain define the intrinsic bounded Voronoi diagram of the points of impact. See Figure 2 for a triangular mesh-based example produced using public domain software [PC03]. This wave propagation is discretised and simulated accurately by solving the Eikonal equation (1) using the extended Fast Marching method for intrinsic distance mapping across point clouds [MS03] summarised in the previous section. By letting the user control the radius of the largest empty circle in the domain of the simplified point set, this method provides guaranteed bounds on the minimum and maximum distance between neighbouring sample points (Figure 3). It thus allows for the relatively simple control of a guaranteed density of the simplified point set [MD03]. We exploit this property to simplify any highly dense input P to a base point set P_0 still sufficiently dense to support meaningful geodesic centroid



Figure 2: Wave propagation for the incremental computation of discrete intrinsic bounded Voronoi diagrams and thus progressive intrinsic (red) farthest sample point localisation of 21, 23 and 25 sample sites on a triangulated surface (from left to right).

computations.

The use of this simplification technique also yields a discrete intrinsic Voronoi diagram of the simplified point set. As discussed next, the availability of this diagram suggests the collecting of local proximity information using a Voronoi diagram-based neighbourhood concept.

4.2 Intrinsic proximity information

Subdivision schemes incorporate refinement and geometric averaging rules in the form of weighted averages of local neighbourhoods. Whilst in a surface mesh context local neighbourhoods follow from mesh connectivity, in the meshless case this connectivity information is replaced by local proximity information. It is interesting to note in this context how information which is generally not related to the geometry of the problem such as mesh connectivity plays such an important role for mesh subdivision. For example, the same object when represented with different types of connectivity requires the application of different mesh subdivision schemes although it is geometrically unchanged. This importance has contributed to the substantial research efforts in the area of remeshing. The role of mesh connectivity in mesh subdivision also explains the existence of numerous subdivision operators restricted in applicability to a particular type of mesh only. By contrast, as described in the following, in the case of our meshless subdivision operator, point cloud proximity is determined intrinsically with the subdivision operator purely formulated in terms of local proximity.

To allow for any irregularity in the base point set P_0 , we favour the use of a neighbourhood concept ensuring a spherical distribution of neighbours q_i around the point p under consideration. Simple ball or k nearest neighbourhoods fail to guarantee such a neighbour distribution [FR01]. Neighbourhood concepts meeting this requirement include Linsen’s [Lin01] enhanced k nearest neighbourhood, which enforces a maximum angle between successive neighbours around p , and Floater and Reimer’s [FR01] local Delaunay neighbourhood. Since our simplification algorithm provides a (discrete) intrinsic

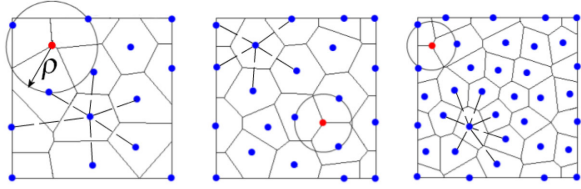


Figure 3: The intrinsic point cloud simplification algorithm [MD03] incrementally computes a geodesic Voronoi diagram across the point-sampled geometry. As illustrated here for the planar case, it guarantees a user-controlled density by letting the user set the radius ρ of the largest empty circle in the domain of the simplified point set. The availability of the geodesic Voronoi diagram supports the immediate determination of intrinsic proximity information in the form of natural neighbourhoods (dashed lines). Note the spherical distribution of these neighbours all around the input point under consideration for the kind of irregular uniformity resulting from the enforcement of ρ .

sic Voronoi diagram of P_0 , we collect local proximity information by considering the set of intrinsic Voronoi, or natural, neighbours N_p instead,

$$N_p = \{q : p \text{ and } q \text{ are neighbours in } BVD(P_l)\},$$

for $p, q \in P_l$, $p \neq q$. $BVD(P_l)$, and thus the neighbourhood information, may be maintained by updating the diagram with any points inserted into P_l thereby obtaining $BVD(P_{l+1})$ (Figure 4). Once a refined point set has reached a certain density, the natural neighbourhood information may be replaced by the k neighbours intrinsically nearest to p . Note that although the natural neighbourhood concept does not guarantee a spherical neighbour distribution when dealing with extreme irregularity, as indicated in Figure 3, it performs well with the uniform irregular density guaranteed by the intrinsic point cloud simplification algorithm discussed in the previous section.

This neighbourhood information is exploited for the computation of local weighted centroids as part of our meshless subdivision scheme presented next.

4.3 An intrinsic subdivision operator for point clouds

Mesh-based subdivision uses rules based on weighted averages of local neighbourhoods following from mesh connectivity. Within our intrinsic meshless subdivision framework, we replace these extrinsic weighted averages of topological neighbours by intrinsic weighted averages of intrinsically neighbouring points. More specifically, we suggest the following set of rules:

Geometric averaging rule: At each iteration, replace

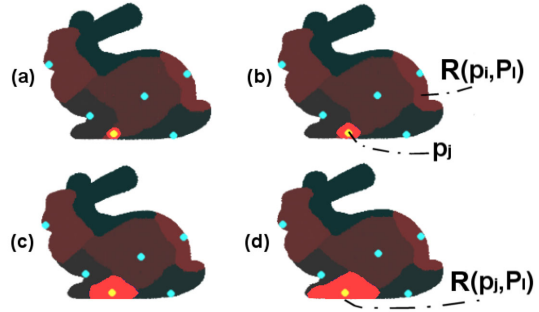


Figure 4: Updating of an intrinsic Voronoi diagram following insertion of a point p_j . The new Voronoi region $R(p_j, P_l)$ is grown across the refined point cloud P_l rendered using an enlarged point size. This is achieved by propagating a front from the newly inserted point p_j outwards, (a)-(c), until it encounters its neighbouring regions and reaches its final size, (d).

$p \in P_l$ by the weighted geodesic centroid, $c(N_p) \in P_{l+1}$, of its intrinsic neighbourhood N_p .

Refinement rule: For each neighbour $q_i \in N_p$, consider the joint intrinsic neighbourhood N_{pq_i} of $p, q_i \in P_l$. Upsample P_l by inserting the weighted geodesic centroid, $c(N_{pq_i}) \in P_{l+1}$, of N_{pq_i} .

This use of weighted centroids in the refinement and geometric averaging rules is reminiscent of both classical subdivision schemes [ZS00] (Section 2) and the “repeated averaging” approach towards the generation of subdivision surfaces ([OS03] and references therein).

By performing the averaging intrinsically on the underlying surface represented by point cloud P , the above set of rules raises the questions of how to compute geodesic averages on manifold surfaces and how to determine a suitable weighting scheme. In this paper, we are interested in showing the conceptual viability of meshless subdivision. We therefore guide the choice of weights by experimental results rather than any theoretical evidence for the scheme’s convergence towards a smooth limit surface. Future work will consider formal proofs of the scheme’s convergence to a limit surface and, consequently, any light such proofs may throw on the optimal choice of weights.

As regards the computation of geodesic centroids, Buss and Fillmore [BF01] present an algorithm for the computation of geodesic averages on spheres. We generalise the underlying, earlier idea ([Kar77] and references therein) of minimising a least squares expression in weighted geodesic distances to non-spherical geometry in the following section.

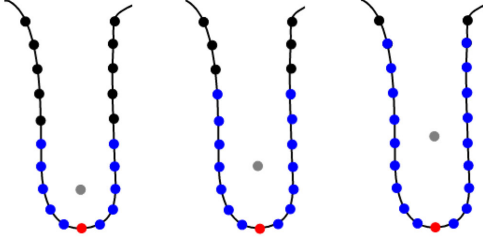


Figure 5: The unweighted centroid of a (blue) subset of this symmetric set of points on a surface is expected to be located on or near the underlying surface. Since it is based on the consideration of intrinsic distances, this is the case when computing the geodesic centroid (red). By contrast, in the case of the Euclidean averaging of the (blue) points, the resulting centroid (grey) is located away from the underlying surface. This effect gets more pronounced when increasing the size of the subset whilst the position of the geodesic centroid remains unaffected due to the symmetry of the point constellation (from left to right).

4.4 Geodesic centroid computation

When considering the concept of meshless geometric subdivision by recursive weighted averaging of local point neighbourhoods, the benefit of performing these centroid computations intrinsically rather than in the Euclidean domain may not be immediately clear. To see the need for operating intrinsically throughout, note that subdivision should ideally generate increasingly smoother representations which are still geometrically close to the surface represented by the input data. This is not guaranteed to be the case when considering Euclidean instead of geodesic centroids. For the simple example illustrated in Figure 5, Euclidean averaging moves the centroid away from the surface. By contrast, since, by definition, geodesic centroid computation is based on intrinsic distances, the geodesic centroid falls onto the surface. We therefore consider geodesic centroids throughout the subdivision process and present our method for computing these centroids on manifold surfaces next.

The weighted (geodesic) centroid of a set of points $\{p_1, \dots, p_n\} \subset M \subset \mathbb{R}^m$ is defined as the point $g \in M$ which minimizes

$$J(g) := \frac{1}{2} \sum_{k=1}^n w_k d_M^2(g, p_k),$$

where w_1, \dots, w_n are the weights: $0 \leq w_i \leq 1$, $\sum_{i=1}^n w_i = 1$ and $d_M(\cdot, \cdot)$ is the geodesic distance. In general, the set of minimizers of $J(\cdot)$ need not be a single point. However, when p_1, \dots, p_n are all contained in a sufficiently small open geodesic ball B_M on M , there exists exactly one minimizer, g_{B_M} of $J(\cdot)$, which happens to lie in B_M [Kar77]. The property we are alluding to here is

convexity: For any $p, q \in B_M$, we require that the shortest geodesic from p to q is unique in M and contained in B_M .

In the Euclidean case, direct differentiation of $J(\cdot)$ yields the minimizer $g = \sum_{k=1}^n w_k p_k$. This simple result does not extend to the general case considered here but we can prove that any minimizer must satisfy:

$$V(g) := \sum_{k=1}^n w_k \nabla_M \frac{1}{2} d_M^2(g, p_k) = 0.$$

Then, starting from a *good* initial guess g_0 , we can track the minimizer g using back propagation with velocity field $V(\cdot)$. This is a sensible procedure since if $g_0 \in B_M$ and B_M as above, then $-V(x)$ points towards g_{B_M} for $x \in B_M$ [Kar77].

In practise, we set $g_0 = \Pi_M(\sum_{k=1}^n w_k p_k)$, where $\Pi_M : \Omega_M \rightarrow M$ is the orthogonal projection operator. We now show that in the light of the considerations presented above, this represents either a reasonable initial condition for the back propagation or a first approximation to the intrinsic centroid. By a simple application of Lemma 17 of [WD03], we have that

$$\left\| \sum_{k=1}^n w_k p_k - \Pi_M \left(\sum_{k=1}^n w_k p_k \right) \right\| \leq C(\text{diam}(B))^2,$$

where C is a global constant which depends on the curvatures of M . Then, let $B = B_M(x, \varepsilon)$, for some $x \in M$ and $\varepsilon > 0$. Since $\|p_i - x\| \leq d_M(p_i, x) \leq \varepsilon$, one also has $\|\sum_{k=1}^n w_k p_k - x\| \leq \varepsilon$. Therefore, since $\|g_0 - x\| \leq \|g_0 - \sum_{k=1}^n w_k p_k\| + \|\sum_{k=1}^n w_k p_k - x\|$, we obtain

$$\|g_0 - x\| \leq C\varepsilon^2 + \varepsilon,$$

which implies $d_M(g_0, x) \leq \varepsilon(1 + D\varepsilon)(1 + C\varepsilon)$, for another constant D depending on global metric properties of M [MS03]. We only care for a simplified bound

$$d_M(g_0, x) \leq E\varepsilon.$$

Finally, let $\bar{\delta} > 0$ be the maximal $\delta > 0$ such that $B_M(x, \delta)$ is convex. Note that it is a fact that if $\bar{\delta} \leq \frac{1}{2} \min \left(\text{inj}(M), \frac{\pi}{\sqrt{K}} \right)$, where $\text{inj}(M)$ is the injectivity radius of M and K bounds all sectional curvatures in M from above, then $B_M(x, \delta)$ is convex for any $x \in M$. See §7.6 and §7.7 in [Cha97]. For such a $\bar{\delta} > 0$ and provided $\varepsilon \leq \bar{\delta}/E$, and $\{p_1, \dots, p_n\} \subset B_M(x, \varepsilon)$ for some $x \in M$, $g_0 \in B_M(x, \bar{\delta})$ and $-V(g_0)$ will be pointing towards g_{B_M} . Also, in case we want to use g_0 as an approximation to g_{B_M} , we have the (weak) bound $d_M(g_{B_M}, g_0) \leq (E + 1)\varepsilon$. Therefore, g_0 , as defined above, represents a sensible choice as the initial condition of an eventual back propagation step, or, in any case, a rough approximation to g_{B_M} with known error bound. Note in particular that it is also a useful choice from the point of view of computational ease.

To demonstrate the applicability of this approach in the context of meshless subdivision, we consider the case of M representing the unit sphere in the following section.

5 Experimental results

We begin by considering the intrinsic meshless subdivision of a set of points sampled relatively regularly uniformly from the surface of the unit sphere. This initial restriction to spherical geometry allows us to discuss qualitative and quantitative aspects of our meshless subdivision operator without the need for taking into account the effects of any particular normal estimation, projection and gradient descent operations performed when processing more complex geometry.

To implement the intrinsic centroid computation method and thus our intrinsic meshless subdivision operator presented in the previous sections, techniques for the computation of intrinsic distances between points on the surface, the projection of the starting point for the back propagation onto the surface and the computation of the back propagation itself are required. In the case of the unit sphere, these techniques are readily available. Intrinsic distances between points follow trigonometrically and orthogonal projection is trivial. Similarly, the exponential map and its inverse are directly available and may be used to implement the back propagation procedure. As a result, for the case of spherical geometry, our approach for geodesic centroid computation narrows down to the technique of [BF01].

We apply our intrinsic meshless subdivision operator to a base point set P_0 of 2144 points sampled relatively regularly uniformly from the unit sphere and shown in Figure 7. To obtain initial natural neighbour proximity information for points in P_0 , we use the intrinsic point cloud simplification algorithm [MD03] without requesting any simplification. The application of our intrinsic subdivision operator to P_0 using this initial proximity information then yields the subdivided point set P_1 presented in Figure 7. The result, P_2 , obtained from the application of the operator to P_1 using natural neighbour information updated as described in Section 4.2 is also shown. Given the relatively strong regularity of the input data, uniform weighting was used for both the refinement and the geometric averaging rule in both iterations. The results produced by our meshless subdivision operator are presented alongside the point sets produced by the application of the Loop subdivision scheme to a triangular mesh representation of the base point set. As indicated by the detail views of Figure 7, the point distributions obtained from the two operators after two iterations are qualitatively similar with the slight irregularities in the distribution of P_0 being slightly more pronounced in the case of meshless subdivision due to the

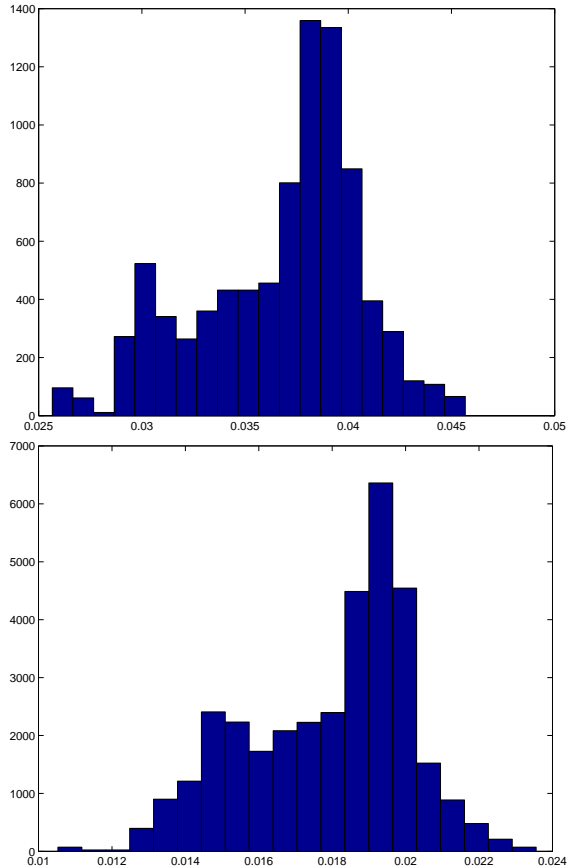


Figure 6: Histograms of the (spherical) distance from each point in P_1 (top) and P_2 (bottom) to its closest neighbour in the respective set.

use of uniform weights. The size of 8570 (P_1) and 34275 (P_2) points respectively of the subdivided point sets coincide for both subdivision schemes. There are no noticeable differences in the smoothing effect of these two operators.

In order to add some quantitative detail to the analysis of the point sets generated by our subdivision operator, we compute the mean and the standard deviation of the distance from each point in the set to its closest neighbor(s) for the subdivided point sets P_1 and P_2 . For each x in the point set $P \subset S$, let $sd_k(x)$ denote the distance from x to its k th closest neighbour. As an indicator for the uniformity of the density of a point set P (at level k), we use, for a positive integer k , $\rho(k) = \frac{\min_P sd_k(x)}{\max_P sd_k(x)}$. Since $\rho(k)$ represents an absolute measure it may be too sensitive, therefore we compute instead $\widehat{\rho}(k) = \frac{\text{mean}(sd_k) - \text{std}(sd_k)}{\text{mean}(sd_k) + \text{std}(sd_k)}$, where mean and std stand for the mean and standard deviation over the point set, respectively.

The histograms of $sd_1(x)$ corresponding to the two sets of points are given in Figure 6. Note in Table 1 that the values of $\widehat{\rho}(k)$ (for $1 \leq k \leq 10$) are quite close

Model \ k	1	2	3	4	5
P_1	0.807	0.832	0.832	0.821	0.788
P_2	0.781	0.815	0.823	0.804	0.785
	6	7	8	9	10
P_1	0.809	0.8132	0.786	0.818	0.828
P_2	0.799	0.798	0.784	0.81	0.822

Table 1: Values of $\widehat{\rho}(k)$ for P_1 and P_2 and with $k \in \{1, 2, \dots, 10\}$.

to 1 therefore indicating small dispersion up to the 10th closest neighbor.

Finally, in Figure 8 we present examples dealing with more complicated geometry. These results were produced by using the projected Euclidean centroid g_0 (Section 4.4) and k nearest neighbourhoods. It is easy to prove that the use of k nearest neighbourhoods results in non-uniform point distributions even if starting from a uniform sampling such as the one produced by the technique discussed in Section 4.1. We are currently introducing the back propagation flow to produce the true geodesic centroid g_{B_M} from g_0 , and are changing the neighbourhood rule. This yields a more uniform point distribution and corresponding examples will be reported elsewhere. Note that for the processing of the spherical geometry, we use g_{B_M} and the neighbourhood is completely intrinsic.

6 Conclusion

In this paper, we introduced the concept of meshless, or point cloud, subdivision. Apart from the introduction of this paradigm, we presented a particular meshless subdivision scheme based on the computation of intrinsic centroids for point clouds. We implemented and showed the applicability of this technique with the help of a new method for the computation of geodesic centroids on manifold surfaces.

The consideration of local proximity instead of mesh connectivity information makes meshless subdivision attractive for the processing of point clouds representing higher-dimensional surfaces. By operating directly across the point cloud, problems associated with the complexity of the topological subdivision of higher-dimensional meshes are avoided. Meshless subdivision operators may be devised for this type of input data which upsample the point-sampled geometry more conservatively than the operator suggested in this paper. In this direction, we are considering introducing adaptive neighbourhoods based on curvature estimators such as those reported in [CP03, MN03]. We leave this to future work.

As regards future work on the theoretical analysis of our meshless intrinsic subdivision scheme, Wallner

and Dyn [WD03, WD04] show the convergence and smoothness of non-linear geodesic curve subdivision by proximity to a corresponding linear extrinsic subdivision scheme. Since our meshless subdivision scheme operates inside a small Euclidean offset band around the point cloud throughout, following [MS01, MS03], we intend to exploit this idea in a surface context to analyse the convergence and continuity of intrinsic meshless subdivision schemes.

Acknowledgements

The Loop subdivision implementation was obtained from Yutaka Ohtake's web site at <http://www.mpi-sb.mpg.de/~ohtake/software/>. The Venus data set was taken from the Cyberware web site at <http://www.cyberware.com/samples/index.html>. This work was partially supported by the Office of Naval Research and the National Science Foundation.

References

- [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 Visual. and Comp. Graph.* 9, 1 (2003), 3–15.
- [ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. In *Proc. SIGGRAPH '03* (2003), pp. 485–493.
- [BF01] BUSS S. R., FILLMORE J. P.: Spherical averages and applications to spherical splines and interpolation. *ACM Trans. on Graphics* 20, 2 (2001), 95–126.
- [Cha97] CHAVEL I.: *Riemannian Geometry: a modern introduction*. Cambridge University Press, Cambridge, UK, 1997.
- [CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. Eurographics Symp. on Comput. Geom.* (2003), pp. 177–187.
- [DL02] DYN N., LEVIN D.: Subdivision schemes in geometric modelling. *Acta Numerica* 12 (2002), 73–144.
- [FCOAS03] FLEISHMAN S., COHEN-OR D., ALEXA M., SILVA C. T.: Progressive point set surfaces. *ACM Trans. on Comp. Graph.* 22, 4 (2003), 997–1011.
- [FR01] FLOATER M. S., REIMERS M.: Meshless parameterization and surface reconstruction. *Computer Aided Geometric Design* 18, 2 (2001), 77–92.
- [HPCD96] HELMSEN J., PUCKETT E. G., COLLELA P., DORR M.: Two new methods for simulating photolithography development in 3d. In *Proc. SPIE Microlithography IX* (1996), pp. 253–261.
- [Kar77] KARCHER H.: Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.* 30, 5 (1977), 509–541.
- [Lin01] LINSSEN L.: Point cloud representation. In *CS Tech. Rep. 2001-3, Universität Karlsruhe, Germany* (2001). www.ubka.uni-karlsruhe.de/vvv/ira/2001/3/3.pdf.
- [Loo87] LOOP C.: Smooth surface subdivision based on triangles. In *Master's th., Dep. of Math., University of Utah, USA* (1987).

- [MD03] MOENNING C., DODGSON N. A.: Fast marching farthest point sampling for implicit surfaces and point clouds. *Tech. Rep. No. 565, Computer Laboratory, University of Cambridge, UK* (2003). www.cl.cam.ac.uk/~cm230/docs/pdfs/FastFPSISPC.pdf.
- [MN03] MITRA N. J., NGUYEN A.: Estimating surface normals in noisy point cloud data. In *Proc. of 19th Conf. on Comput. Geom.* (2003), pp. 322–328.
- [MS01] MÉMOLI F., SAPIRO G.: Fast computation of weighted distance functions and geodesics on implicit hypersurfaces. *Journal of Comput. Phys.* 173, 1 (2001), 764–795.
- [MS03] MÉMOLI F., SAPIRO G.: Distance functions and geodesics on point clouds. *TR 1902, IMA, University of Minnesota, USA* (2003). www.ima.umn.edu/preprints/dec2002/1902.pdf.
- [OBS00] OKABE A., BOOTS B., SUGIHARA K.: *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams, 2nd ed.* John Wiley & Sons, Chichester, UK, 2000.
- [OS03] OSWALD P., SCHROEDER P.: Composite primal/dual sqrt(3)-subdivision schemes. *Computer Aided Geometric Design* 20, 3 (2003), 135–164.
- [Par] Pointstopolys software, paraform, inc. <http://www.paraform.com/ppdl>.
- [PC03] PEYRÉ G., COHEN L.: Geodesic re-meshing and parameterization using front propagation. In *Proc. 2nd IEEE VLSM Workshop* (2003).
- [PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *Proc. 13th IEEE Visualization* (2002), pp. 163–170.
- [PKKG03] PAULY M., KEISER R., KOBBELT L. P., GROSS M.: Shape modeling with point-sampled geometry. *Proc. SIGGRAPH '03* (2003), 641–650.
- [Set99] SETHIAN J. A.: *Level Set Methods and Fast Marching Methods, 2nd ed.* Cambridge University Press, Cambridge, UK, 1999.
- [Tsi95] TSITSIKLIS J. N.: Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control* 40 (1995), 1528–1538.
- [WD03] WALLNER J., DYN N.: Smoothness of subdivision schemes by proximity. *Tech. Rep. No. 112, Tech. Univ. Wien, Austria* (2003). www.geometrie.tuwien.ac.at/wallner/nsubdiv.pdf.
- [WD04] WALLNER J., DYN N.: Convergence and c1 analysis of subdivision schemes on manifolds by proximity. *preprint* (2004). www.geometrie.tuwien.ac.at/wallner/sbd1.pdf.
- [ZPKG02] ZWICKER M., PAULY M., KNOLL M., GROSS M.: Pointshop3d: An interactive system for point-based surface editing. *Proc. SIGGRAPH '02* (2002), 322–329.
- [ZPvBG01] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Surface splatting. *Proc. SIGGRAPH '01* (2001), 371–378.
- [ZS00] ZORIN D., SCHROEDER P.: Subdivision for modeling and animation. *SIGGRAPH '00 course notes* (2000).

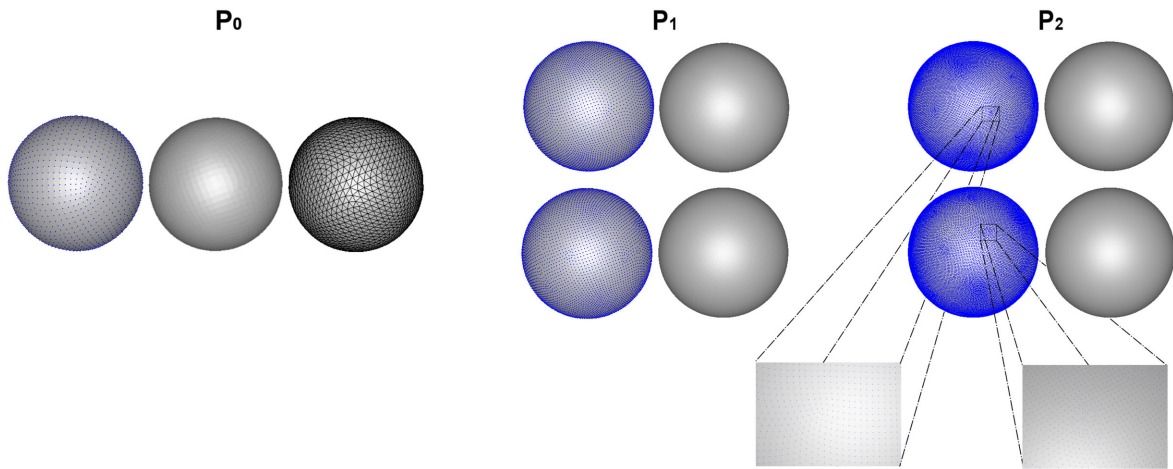


Figure 7: The base point set, P_0 , of 2144 points acquired from the unit sphere (left), is subdivided recursively using both our meshless subdivision operator (bottom) and Loop subdivision (top). The triangular base mesh generated from P_0 for the support of Loop subdivision is shown on the left. The subdivided point sets P_1 (8570 points) and P_2 (34275 points) are shown in the center and on the right respectively. The corresponding reconstructed surfaces are given next to each point set. Both these surfaces and the base mesh were computed with the help of public domain software [Par]. (This figure is best seen on-screen.)

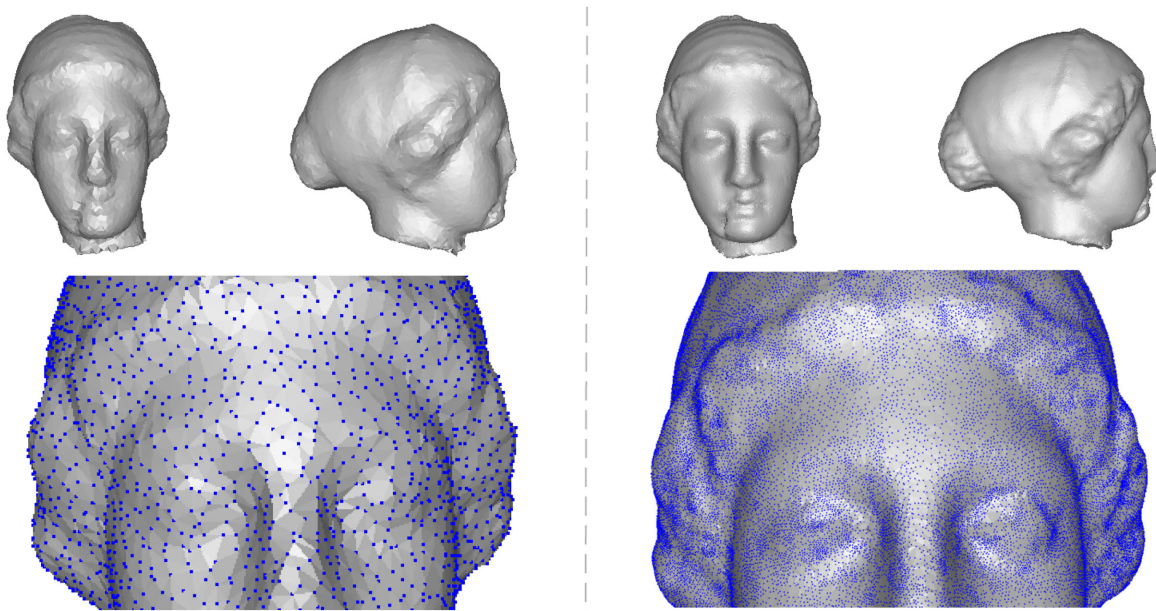


Figure 8: Example for the meshless geometric subdivision of more elaborate geometry in form of the Venus data set. The base point set is displayed on the left, the result obtained after two iterations of meshless geometric subdivision is shown on the right. See text for details on the steps for the production of these results and comments on the slightly non-uniform distribution of the subdivided point sets. All surfaces were reconstructed with the help of public domain software [Par].