SGP 2006

Symposium on Geometry Processing 2006

Fourth Eurographics Symposium on Geometry Processing

Cagliari, Sardinia, Italy
June 26–28, 2006

Papers Co-Chairs
Alla Sheffer, University of British Columbia
Konrad Polthier, Freie Universität Berlin

Co-sponsored by ACM SIGGRAPH and EUROGRAPHICS Association

These proceedings are available only in electronic version

Table of Contents

Eurographics Symposium on Geometry Processing, Poster Proceedings (2006)
Konrad Polthier, Alla Sheffer (Editors)

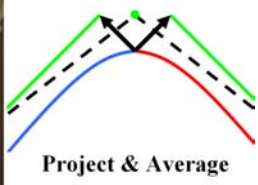
Constrained Planar Remeshing for Architecture

Barbara Cutler
Rensselaer Polytechnic Institute

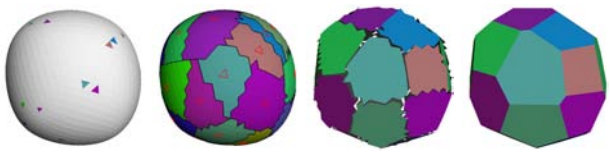
Emily Whiting
Massachusetts Institute of Technology

We are exploring an important class of architectural fabrication constraints: those associated with planar construction materials such as glass or plywood. Although glass can be bent into curved panels, flat planar sheets are more cost-effective to manufacture. With standard meshing techniques, designers are limited to triangular or quadrilateral primitives.

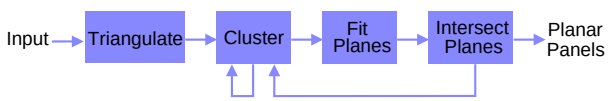
In the design study below, a Voronoi diagram is texture-mapped onto a 3D surface. A flexible rubber sheet was necessary to skin the non-planar cells. The designer wanted a planar remeshing tool that would generate *planar* meshings with similar patterns.



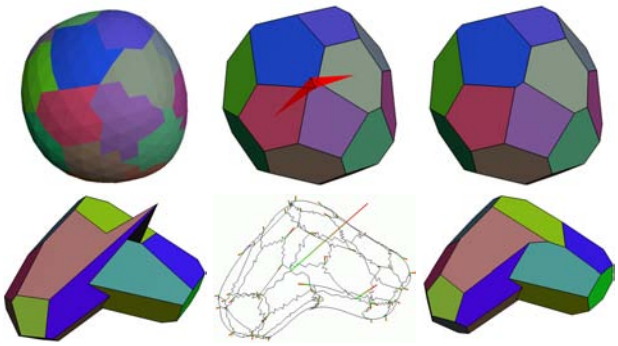
Cohen-Steiner et al. fit a pre-specified number of face clusters to a surface using Lloyd's relaxation. Planar proxies are assigned to these clusters and connected to create a polygonal mesh:



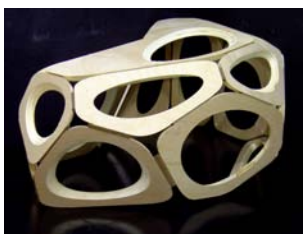
However, in their work the final polygon vertices are computed by *averaging* the projection of an original mesh vertex onto each proxy. Thus, facets that have more than three vertices will likely be non-planar. Unfortunately, the vertices from this remeshing cannot simply be moved such that all facets are planar. Instead we place vertices where neighboring planes intersect:

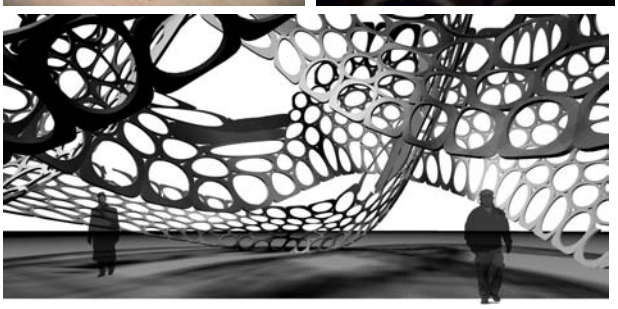


In this process the mesh may become inconsistent or degenerate. We detect and correct situations where an edge has "flipped" orientation or the intersection vertex "spikes" to infinity as shown below. The discontinuous nature of the solution space presents challenges for optimization because the neighborhood of a particular facet is not stable from iteration to iteration.



Using automated milling equipment, the panels are fabricated from planar plywood sheets and assembled to create a unique and inspiring outdoor sculpture. Our prototype tool has potential to impact not only architectural design, but also the engineering for general fabrication problems.





Acknowledgments

Much thanks and credit go to workshop organizer Professor Mark Goulthorpe and the workshop participants, including Michael Powell, John Rothenberg, Dennis Michaud, Matt Trimble, and Jeff Anderson, all from MIT's Department of Architecture.

References

Cohen-Steiner D., Alliez P., Desbrun M.: Variational Shape Approximation. ACM Transactions on Graphics 23, 3 (Aug. 2004), 905—914.

Eurographics Symposium on Geometry Processing, Poster Proceedings (2006)
Konrad Polthier, Alla Sheffer (Editors)

A Lagrangian Approach to Dynamic Interfaces through Kinetic Triangulation of the Ambient Space

Jean-Philippe Pons[1,2] Jean-Daniel Boissonnat[1]

[1] INRIA, Sophia-Antipolis, France
[2] École Nationale des Ponts et Chaussées, Marne-la-Vallée, France

Abstract

We propose a robust and efficient Lagrangian approach for modeling dynamic interfaces between different materials undergoing large deformations, in two dimensions. The principle of our approach is to maintain a two-dimensional triangulation which embeds the one-dimensional polygonal description of the interfaces. Topology changes can then be detected as inversions of the faces of this triangulation. Each triangular face is labeled with the type of material it contains. The connectivity of the triangulation and the labels of the faces are updated consistently during deformation, within the kinetic data structures framework. Thanks to the exact computation paradigm, the reliability of our algorithm, even in difficult situations such as shocks and topology changes, can be certified. Our work brings an interesting alternative to the heuristic remeshing procedures of traditional Lagrangian methods, for two-dimensional and axisymmetric simulations.

1. Introduction

Usually, in Lagrangian methods, the crossing of points and the formation of *swallow-tails* cause the failure of the simulation. In our approach, the embedding triangulation enforces the absence of loops and the proper handling of crossing points. This idea is closely related to the use of time-dependent pseudo-triangulations for the efficient detection of collisions between several moving polygonal objects in the plane [BEG*99, KS02, ABG*02]. However, the problem that we address in this work is less restrictive: our method allows the simulation to continue after a collision, by properly handling the fusion or the splitting of the different materials.

The maintenance of our embedding triangulation relies on the *kinetic data structures* framework, introduced by Basch and Guibas [BG99]. Rather than trying to repair the possibly entangled triangulation after each iteration of the simulation, we smoothly interpolate the coordinates of interface points between two consecutive time steps, and we modify the connectivity of the triangulation and the labels of the faces exactly as and when it is required. As a result, our *lazy kinetic triangulation* has the desirable property not to introduce any unnecessary perturbation of the interfaces in the absence of topology changes, and to handle them exactly and efficiently when they occur.

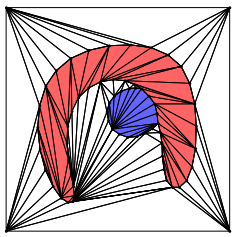
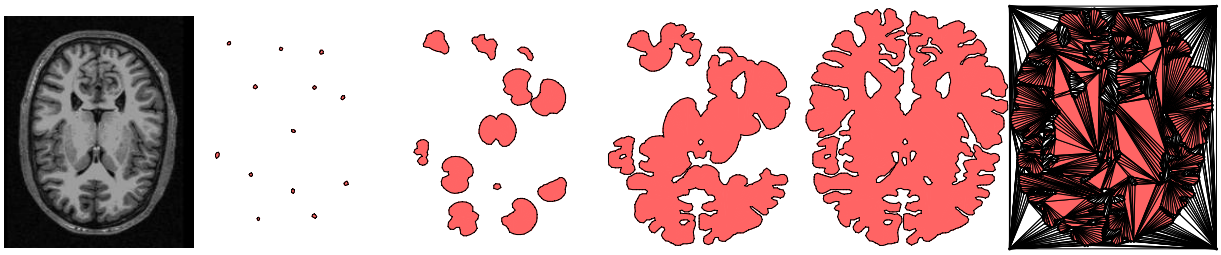


Figure 1: *Embedding labeled triangulation.*

2. Methods

The principle of our approach is to couple the numerical solver which outputs the position of the vertices at some (typically evenly-spaced) time instants, with a kinetic triangulation. A physical interpretation is attached not only to its vertices, but also to its faces, namely the type of material that it contains, that we call for short the *label* of a triangle. The interfaces of interest are embedded in this triangulation. They are composed of the edges adjacent to two triangles having different labels. This representation enforces watertight interfaces and seamlessly accomodates any number of materials. An example of this embedding triangulation, including three materials with some triple junctions, is displayed in Figure 1.



(a) (b) (c) (d) (e) (f)

Figure 2: *Application to brain segmentation: (a) MR image, (b-e) different stages of the evolution, (f) labeled triangulation of the final shape.*

Our kinetic triangulation is termed "lazy" because it undergoes no connectivity change as long as it remains a valid geometric triangulation, that is to say as long as the orientations of the triangles do not change and the triangulation covers the convex hull of the vertices. The lazy behaviour guarantees that the triangulation remains a faithful embedding of the polygonal interfaces.

Contrarily to a Delaunay triangulation, the state of our triangulation is not canonically defined: it does not depend only on the positions of the points, but also on the history of the motion. As a result, it does not suffer from an instability and a multiplication of meaningless events when the points are very close to a configuration change, like it happens with Delaunay when four or more points are nearly co-circular. Another advantage of an arbitrary triangulation is that the predicates are cheaper to compute than for Delaunay. For polynomial trajectories of degree d, the failure times of the *Orientation* predicates are roots of polynomials of degree $2d$, against degree $4d$ for the *InCircle* predicates.

An objection to the lazy kinetic triangulation could be the absence of control on the quality of the triangles. In some other contexts where the numerical robustness directly depends on the quality of the worst-shaped element of the triangulation, this can impair the simulation. This limitation does not apply to our approach, because we are only interested in the interfaces between the different materials. The tiling of each material is not used in the simulation, it is only available internally to the kinetic data structure.

Here, for the sake of simplicity, we assume that the convex hull remains constant during motion. This can be easily achieved by adding to the triangulation fixed points that delimit a bounding box of the simulation. As a result, the certification of our lazy kinetic triangulation only requires one orientation predicate for each triangle. The events are processed with elementary local connectivity modifications, namely edge flips and edge collapses. For example, the generic event of a flat triangle whose vertices have three distinct positions is handled by flipping the longest edge of the flat triangle, so as to restore a valid geometric triangulation. An edge flip requires the two adjacent triangles to form a convex quad. Note that this condition is automatically fulfilled in our case. Also, consistent labels must be assigned to the two new triangles generated by the edge flip: these two triangles are given the label of the original non-flat triangle involved in the flip so that the location of the interfaces is not altered.

Degenerate cases, like flat triangles with two or three vertices having the same position, or the handling of simultaneous events, require special care. We have designed a strategy that is proven to terminate and to restore a valid geometric triangulation in all cases. It is not described here due to space limitations.

3. Results

We give a glimpse of the possible applications of our approach in medical imaging. Deformable curves and surfaces are a widely used technique for shape reconstruction in image processing. Here, we address the automatic delineation of the brain from magnetic resonance imaging (MRI) of the head. Figure 2 shows the 2D magnetic resonance image whose intensities drive the motion, the different stages of the evolution and the embedding triangulation of the final interface. Note how the initial seeds grow and progressively merge to fit the complex shape of the brain.

References

[ABG*02] AGARWAL P., BASCH J., GUIBAS L., HERSHBERGER J., ZHANG L.: Deformable free-space tilings for kinetic collision detection. *The International Journal of Robotics Research 21*, 3 (2002), 179–197. 1

[BEG*99] BASCH J., ERICKSON J., GUIBAS L., HERSHBERGER J., ZHANG L.: Kinetic collision detection between two simple polygons. In *Symposium on Discrete Algorithms* (1999), pp. 102–111. 1

[BG99] BASCH J., GUIBAS L.: Data structures for mobile data. *Journal of Algorithms 31*, 1 (1999), 1–28. 1

[KS02] KIRKPATRICK D., SPECKMANN B.: Kinetic maintenance of context-sensitive hierarchical representations for disjoint simple polygons. In *Annual Symposium on Computational Geometry* (2002), pp. 179–188. 1

Eurographics Symposium on Geometry Processing, Poster Proceedings (2006)
Konrad Polthier, Alla Sheffer (Editors)

Outlier Robust ICP for Minimizing Fractional RMSD

Jeff M. Phillips Ran Liu Carlo Tomasi

Duke University, Durham, NC, USA

Introduction

Aligning an input data set to a model data set is fundamental to many important problems such as scanned model reconstruction, structural biochemistry, and medical imaging. The input data and the model data are typically given as a set of points with the relative positions not known, making the task of registering them nontrivial.

A popular approach to solving this problem is known as the iterative closest point (ICP) algorithm [BM92, CM92] which alternates between finding the optimal correspondence between points, and finding the optimal transformation of one point set onto the other. As both steps reduce the distance between the point sets, this process converges, but only to a local minimum.

However, ICP, and its many variations, are vulnerable to point sets with outliers. Because ICP will find correspondences for all points, and then find the optimal transformation for the entire point set, the outliers will skew the alignment. Many heuristics have been suggested [DF02, CSK05] including only aligning points within a set threshold, but most of these techniques are not guaranteed to converge, and thus can possibly go into an infinite loop, or require an expensive check to prevent this. If the fraction f of points which are outliers is known, then Trimmed ICP (TrICP) [CSK05] can be used to find the optimal alignment of the most relevant fraction f of points. However, this fraction is rarely known a priori. If an alignment is given then RANSAC-type methods can be used to determine a good threshold for determining these outliers. None of these existing approaches both find a local minimum and converge.

Our contributions. Our solution to these problems is to incorporate the fraction of points which are outliers into the function being optimized. To this end, this abstract (see full version [PLT06] for details) makes the following contributions:

- We formalize a new distance measure between point sets which accounts for outliers: FRMSD.
- We provide an algorithm, Fractional ICP, to optimize FRMSD which we prove to converge to a local optimum in the correspondence, transformation, and fraction of outliers.
- We show that Fractional ICP finds an alignment with only the points which are more likely to be inliers than outliers.

Fractional RMS Distance

Consider two point sets $D, M \in \mathbb{R}^d$. The goal of this paper is to align an input data set D to a model data set M under some class of transformations, $\mathcal{T}$, such as rigid motions. We assume M and D are quite similar and there exists a strong correspondence between most points in the data; however, there may be outliers, points in either set which are not close to any point in the other set. Our goal is to define and minimize over a set of transformations a relevant distance between these two point sets.

We define the root mean squared distance

$$\text{RMSD}(D, M, T, \mu) = \sqrt{\frac{1}{|D|} \sum_{p \in D} ||T(p) - \mu(p)||^2},$$

and we seek to minimize this quantity over a set of transformations $T \in \mathcal{T}$ and matchings $\mu : D \to M$. RMSD is quite susceptible to outliers because the squared distance gives a large weight to outliers. To counteract this, for a fraction $f \in [0, 1]$ choose the $f|D|$ points with the smallest residual distance $r = ||p - \mu(p)||$. Let D_f be this set.

We define fractional root mean squared distance

$$\text{FRMSD}(D, M, f, T, \mu) = \frac{1}{f^\lambda} \sqrt{\frac{1}{|D_f|} \sum_{p \in D_f} ||T(p) - \mu(p)||^2},$$

and we seek to minimize this quantity over a set of transformation $T \in \mathcal{T}$, matchings $\mu : D \to M$, and fractions $f \in [0, 1]$.

Value of λ. Under reasonable assumptions on the distribution of outliers and through some straightforward but tedious probability theory we can show that for $\lambda = 1.3$ (resp. $\lambda = 0.95$) for point sets in $\mathbb{R}^2$ (resp. $\mathbb{R}^3$), that FRMSD considers only the points which are more likely to be inliers than to be outliers. This is somewhat dependent on the noise among the inliers, but only weakly dependent on the fraction of inliers. We show empirically that as λ is increased up to 4 or 5, the minimizing fraction f and the RMSD value does not change much. Thus, the algorithm is not sensitive to this regularization parameter.

Fractional ICP

A simple modification of ICP, shown in Algorithm 0.1, will find a local minimum of FRMSD. We refer to this algorithm as Fractional ICP or FICP.

Algorithm 0.1 FICP(D, M)

1: Compute $\mu_0 = \arg \min_{\mu_0 : D \to M} \text{RMSD}(D, M, T_0, \mu_0)$.
2: Compute $\arg \min_{f_0 \in [0,1]} \text{FRMSD}(D, M, f_0, T_0, \mu_0)$.
3: $i \leftarrow 0$.
4: **repeat**
5: $\quad i \leftarrow i + 1$.
6: $\quad$ Compute $\arg \min_{D_f} \text{RMSD}(D_f, M, T_{i-1}, \mu_{i-1})$.
7: $\quad$ Compute $\arg \min_{T_i \in \mathcal{T}} \text{RMSD}(D_f, M, T_i, \mu_{i-1})$.
8: $\quad$ Compute $\arg \min_{\mu_i : D \to M} \text{RMSD}(D, M, T_i, \mu_i)$.
9: $\quad$ Compute $\arg \min_{f_i \in [0,1]} \text{FRMSD}(D, M, f_i, T_i, \mu_i)$.
10: **until** $(u_i = u_{i-1}$ and $f_i = f_{i-1})$

Implementation. Given a standard implementation of ICP, we need the additional steps of computing the subset D_f (step 6) and computing the fraction f (step 9). Since the rest of the algorithm is unchanged, most variations of ICP can incorporate this adaptation. The subset D_f can be computed by sorting the residuals $r = ||p - \mu(p)||$ and letting D_f be the $f|D|$ points with smallest corresponding residuals. Once the residuals are sorted, the fraction f can be computed by considering all $|D|$ possible subsets and choosing the one with smallest value in FRMSD.

Convergence of algorithm. Algorithm 0.1 converges to a local minimum of FRMSD(D, M, f) over the space of all transformations, matchings, and fractions of points used in the matching. This is a local minimum in a sense that if all but one of transformations, matchings, or fractions is fixed, then the value of the remaining variable cannot be changed to decrease the value of FRMSD(D, M, f).

Theorem 0.1 For any two points sets $D, M \in \mathbb{R}^d$, Algorithm 0.1 converges to a local minimum of FRMSD(D, M, f, T, μ) over $(f, T, \mu) \in [0, 1] \times \mathcal{T} \times \{D \to M\}$.

Experiments

In the full version [PLT06] we demonstrate that FICP has a larger radius of convergence than TrICP, and is faster and more accurate than TrICP and ICP. Also, we empirically demonstrate that FRMSD is not sensitive to λ, but FICP is more robust with λ set larger than its optimal value. Thus we run all experiments, unless otherwise specified, with $\lambda = 3$. After converging, λ can be reset to its optimal value and the process will reconverge quickly.

Figure 1 shows the alignment of the scan at 0° (blue) aligned with the scan at 48° (red) of the Stanford dragon using ICP and FICP. Notice how when the scans are aligned with ICP (right), the points in the dragon's tail are slightly misaligned, whereas with FICP (left), the alignment is much better.

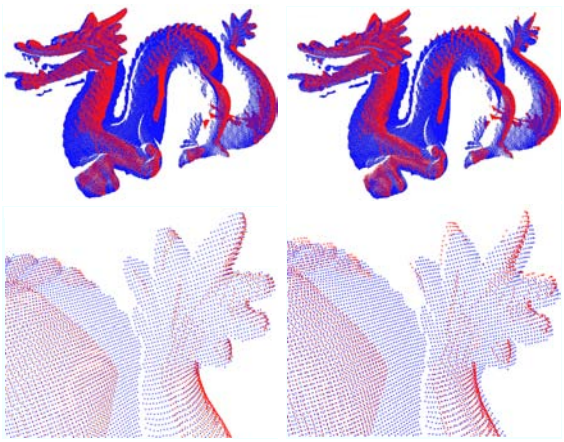


Figure 1: *Alignment of scans for dragon model with ICP (right) and FICP (left). Bottom shows zoomed in view of dragon's tail.*

References

[BM92] Besl P. J., McKay N. D.: A Method for Registration of 3-D Shapes. *IEEE Trans. on Pat. Anal. and Mach. Intel. 14*, 2 (February 1992).

[CM92] Chen Y., Medioni G.: Object modelling by registration of multiple range images. *Im. and Vis. Comp. 10* (1992), 145–155.

[CSK05] Chetverikov D., Stepanov D., Krsek P.: Robust Euclidean alignment of 3D point sets: the Trimmed Iterative Closest Point algorithm. *Im. and Vis. Comp. 23*, 3 (2005), 299–309.

[DF02] Dalley G., Flynn P.: Pair-Wise Range Image Registration: A Study of Outlier Classification. *Comp. Vis. and Im. Und. 87* (2002), 104–115.

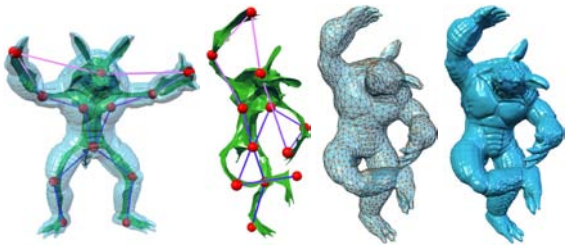
[PLT06] Phillips J. M., Liu R., Tomasi C.: Outlier Robust ICP for Minimizing Fractional RMSD. *Duke TR*, CS-2006-05 (2006).

Eurographics Symposium on Geometry Processing, Poster Proceedings (2006)
Konrad Polthier, Alla Sheffer (Editors)

Skeleton-driven Laplacian Mesh Deformations

Shin Yoshizawa, Alexander Belyaev and Hans-Peter Seidel

MPI Informatik, Saarbrücken, Germany



(a) (b) (c) (d)

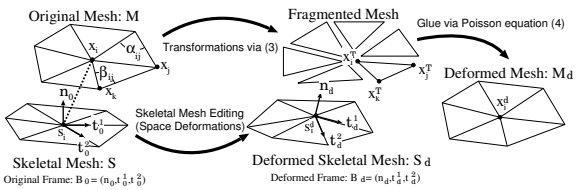
An example of skeleton-driven Laplacian mesh deformations. (a): Armadillo (332K triangles), its skeletal mesh (5K triangles), and the stick-figure skeleton. (b): A space deformation of the skeletal mesh. (c): A coarse deformed mesh obtained by using discrete differential coordinates from the deformed skeletal mesh (0.13 sec). (d): Using a multiresolutional mesh representation [LMH00] quickly delivers a desired deformation of that highly-detailed model (3.8 sec).

We present a novel semi-rigid mesh deformation technique based on combining skeleton-driven deformations [Blo02, YBS03] with discrete differential coordinates [Sor05]. Our method allows to generate natural-looking complex mesh deformations by preserving the original shape thickness and fine geometric details with a certain smoothness.

Skeletal Mesh Extraction and Editing

A skeletal mesh S is extracted from a given original mesh M by using a two-sided Voronoi-based approximation technique [HBK02]. The Voronoi poles are calculated for all vertices of M, and they are triangulated by copying the mesh connectivity of M in order to generate S. A user defames S to the deformed skeletal mesh S_d via space deformations.

Skeleton-driven Laplacian Deformation



According to changes of local frames assigned to triangles of S and S_d, M is transformed into a fragmented mesh by

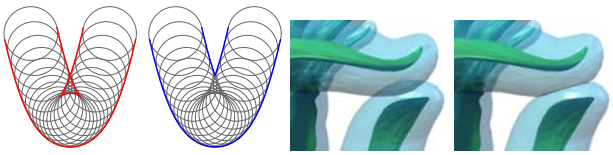
$$\mathbf{x}_l^T = \mathbf{s}_l^d + |\mathbf{x}_l - \mathbf{s}_l| \frac{A(\mathbf{x}_l - \mathbf{s}_l)}{|A(\mathbf{x}_l - \mathbf{s}_l)|}, \quad A = B_d B_0^{-1},$$

where $|\mathbf{x}_l - \mathbf{s}_l|$ represents the original shape thickness given by the medial ball radius function. The final deformed mesh M_d is obtained by minimizing a difference between the deformed and fragmented meshes in discrete differential coordinates. This minimization is equivalent to solving a sparse system of linear equations

$$\triangle_{\mathbf{R}}(\{\mathbf{x}^d, \mathbf{s}^d\}) = \mathrm{div}_{\mathbf{R}} \nabla_{\mathbf{R}}(\{\mathbf{x}^T, \mathbf{s}^d\}) \Rightarrow \mathbf{u} = L_0^{-1}\mathbf{b},$$

where L_0 is a graph-Laplacian matrix constructed by M and S.

Self-Intersection Fairing

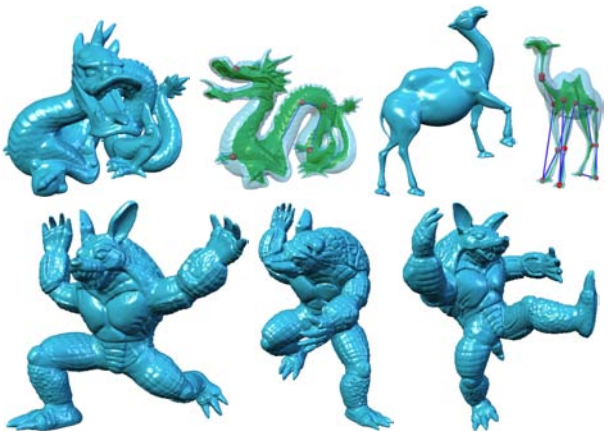


Envelope vs. Union. Global Self-Intersection Fairing.

The self-intersections are eliminated via the evolution

$$\frac{\partial W(t)}{\partial t} = F(U), \quad \triangle_{\mathbf{R}} U(t) = W(t),$$

where $F(U) = f(\mathbf{z})\nabla f(\mathbf{z})$ if $\mathbf{z} = \mathbf{x}^d$, otherwise $F(U) = 0$. Here $f()$ is a union of medial balls, $U(t)$ is an evolving volume, and $W(t)$ is the evolving volume represented in the differential coordinates.



References

[Blo02] BLOOMENTHAL J.: Skinning: Medial-based vertex deformation. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation* (2002), pp. 147–151.

[HBK02] HISADA M., BELYAEV A. G., KUNII T. L.: A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Computer Graphics Forum 21*, 4 (2002), 1–12.

[LMH00] LEE A., MORETON H., HOPPE H.: Displaced subdivision surfaces. In *Proceedings of ACM SIGGRAPH* (2000), pp. 85–94.

[Sor05] SORKINE O.: Laplacian mesh processing. In *Eurographics:State-of-the-art report* (2005).

[YBS03] YOSHIZAWA S., BELYAEV A., SEIDEL H.-P.: Free-form skeleton-driven mesh deformations. In *Proceedings of 8th ACM Symposium on Solid Modeling and Applications* (2003), pp. 247–253.

Eurographics Symposium on Geometry Processing, Poster Proceedings (2006)
Konrad Polthier, Alla Sheffer (Editors)

Shape Detection in Point Clouds

Ruwen Schnabel Roland Wahl Reinhard Klein†

Universität Bonn, Computer Graphics Group

Abstract

In this work we present an automatic algorithm to detect basic shapes in unorganized point clouds. The algorithm decomposes the point cloud into a concise, hybrid structure of inherent shapes and a set of remaining points. Each detected shape serves as a proxy for a set of corresponding points. Our method is based on random sampling and detects planes, spheres, cylinders and cones. For models with surfaces composed of these basic shapes only, e.g. CAD models, we automatically obtain a representation solely consisting of shape proxies. We demonstrate that the algorithm is robust even in the presence of many outliers and a high degree of noise. The proposed method scales well with respect to the size of the input point cloud and the number and size of the shapes within the data. Even point sets with several millions of samples are robustly decomposed within less than a minute. Moreover the algorithm is conceptually simple and easy to implement. Application areas include measurement of physical parameters, scan registration, surface compression, hybrid rendering, shape classification, meshing, simplification, approximation and reverse engineering.

Contribution We introduce novel extensions and optimizations that allow an efficient application of the RANSAC paradigm. The method not only inherits the desirable properties of simplicity, generality and robustness, but, additionally achieves high performance and scalability. In detail, our novel extensions are: A sampling strategy that adapts to the size of the shapes present in the data. A criterion to determine when enough candidates have been drawn based on an analysis of our sampling strategy. A lazy score evaluation on a sequence of random subsets that considers only as many points as necessary to determine the best shape.

The impact of our novel sampling strategy is best illustrated with an example. If the ordinary RANSAC sampling strategy were to be used to detect a shape that comprises only three thousandth of the point-cloud, 151,522,829 candidates would have to be drawn to achieve a detection probability of 99%. With our strategy only 64,929 candidates have to be generated for the same probability. That is an improvement by three orders of magnitude, i.e. in this case that is the difference between hours and seconds.

model	$\lvert\mathcal{P}\rvert$	ε	α	τ	$\lvert\Psi\rvert$	$\lvert\mathcal{R}\rvert$	sec
fandisk	12k	0.01	10	50	23	38	0.57
rocker arm	40k	0.01	20	100	67	833	2.9
oil pump	542k	0.01	20	500	106	61k	12.3
choir screen	1,922k	0.002	20	4,000	81	543k	20.8

Table 1: *Statistics on processed models. The approximation tolerance ε is given as ratio of the bounding box width. α is the allowed normal deviation and τ the minimum size of a shape. $\lvert\Psi\rvert$ gives the number of detected shapes and $\lvert R\rvert$ of unassigned points.*

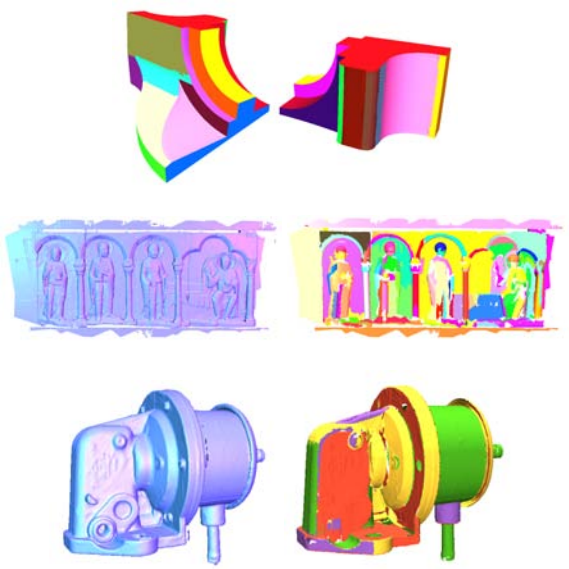


Figure 1: *Top: On CAD objects like the fandisk, that consist of the basic shapes detected by our algorithm, shape proxies for every part of the surface are obtained. Middle: Detected shapes in the choir screen point-cloud with 2 million points. Shapes were detected in 20 seconds and are depicted in random colors on the right. Bottom: The scanned oil pump model contains ca. 500k points. Points in the model on the right are colored according to the type of the shape they have been assigned to: planes are red, cylinders green, spheres yellow and cones purple. The model is provided courtesy of INRIA and ISTI by the AIM@SHAPE Shape Repository.*

† e-mail:{schnabel,wahl,rk}@cs.uni-bonn.de