

Robust and Efficient Surface Reconstruction From Range Data

P. Labatut, J.-P. Pons and R. Keriven

IMAGINE, ENPC/CSTB, LabIGM, Université Paris-Est, France

Abstract

We describe a robust but simple algorithm to reconstruct a surface from a set of merged range scans. Our key contribution is the formulation of the surface reconstruction problem as an energy minimisation problem that explicitly models the scanning process. The adaptivity of the Delaunay triangulation is exploited by restricting the energy to inside/outside labelings of Delaunay tetrahedra. Our energy measures both the **output surface quality** and how well the surface agrees with **soft visibility constraints**. Such energy is shown to perfectly fit into the minimum $s - t$ cuts optimisation framework, allowing fast computation of a globally optimal tetrahedra labeling, while avoiding the “shrinking bias” that usually plagues graph cuts methods.

The behaviour of our method confronted to noise, undersampling and outliers is evaluated on several data sets and compared with other methods through different experiments: its strong robustness would make our method practical not only for reconstruction from range data but also from typically more difficult dense point clouds, resulting for instance from stereo image matching. Our effective modeling of the surface acquisition inverse problem, along with the unique combination of Delaunay triangulation and minimum $s - t$ cuts, makes the computational requirements of the algorithm scale well with respect to the size of the input point cloud.

Keywords: surface reconstruction, minimum $s - t$ cut, Delaunay triangulation

ACM CCS: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

The problem of reconstructing a surface from a set of sample points in 3D is motivated by numerous applications in reverse-engineering, prototyping, visualisation or computer vision and has consequently always been an active field of research. In a laboratory setting, point clouds are typically acquired as sets of range images by light-stripe laser scanners with an optical triangulation system. With a registration algorithm, these range images are then merged together to form a dense point set. While recent reconstruction methods have exclusively considered unoriented point sets or, on the opposite, pre-required good normal estimates for these points, the only requirement to apply our method is the availability of approximate lines of sight: despite being either available or easily recoverable, such datum is often simply thrown away.

Our method put this additional information to use to formulate the reconstruction problem as an energy minimisation

on a Delaunay triangulation. Our energy basically measures how well an inside/outside labeling of Delaunay tetrahedra agrees with soft visibility constraints derived from lines of sight and the likeliness of the output surface resulting from this labeling. Fortunately, our energy can be interpreted as an $s - t$ cut in a special graph allowing a globally optimal labeling of tetrahedra with respect to these constraints and the surface quality measure to be efficiently found as a minimum $s - t$ cut. This simple combination of a labeling of Delaunay tetrahedra with the global optimisation of a visibility based energy exhibits strong resilience to various kinds of alterations of the input data.

1.1. Related work

Most surface reconstruction methods roughly fall into two major categories: implicit surface methods and Delaunay-based methods. Other less common approaches include deformable models or template-based methods.



Figure 1: *Berkeley angel*. 361K vertices, 716K triangles, genus 3 (the original model has genus 1).

1.2. Implicit methods

By constructing a function of space from the samples, a surface can be implicitly defined as a level-set of the function allowing smooth and approximating surface reconstruction. A first example is [HDD*92], where tangent planes are estimated from the k nearest neighbors of each sample. A consistent orientation is found and the considered function is the signed distance to the tangent plane of the closest point in space. In [CL96], the importance of the scanning process is acknowledged and lines of sight are exploited to blend weighted distance functions derived from range images but restricted to a thin shell around the samples for efficiency reasons. Later methods represent the function as a weighted sum of basis functions, typically radial basis functions (RBF): RBFs are placed at constraint points where the value of the function is known and the weights are globally solved to satisfy the constraints exactly or approximately and to minimize a smoothness measure. To avoid the trivial solution, [CBC*01] has to impose on- and off-surface constraints. In [OBS04], compactly supported RBFs with adaptive support are relied upon to handle noise. Other recent approaches construct local functions near the sample points and blend them together to obtain the implicit function using locally supported weight functions. In [OBA*03], the multi-level partition of unity implicit surface representation is introduced: low-degree polynomials approximate the shape of the surface in each cell of an adaptive octree and an efficient implementation is demonstrated to handle large sets of points. Moving least squares (MLS) [LS81] can handle moderate amount of noise and be used to define similar implicit functions with signed distance to local planes as local approximants, yielding the implicit MLS method of [SOS04]: reconstruction guarantees are provided for sufficiently dense and uniform point clouds [Kol08]. A related but different method is the projection-based MLS of [Lev03], where the surface is sought as the fixed point of a parametric fit procedure. [ABCO*03] introduced the technique to the field of computer graphics with a polynomial fitting step. Numerous variants exist among which [FCOS05] which cope with noise with a least-median-of-squares estimator from robust statistics while still preserving sharp features.

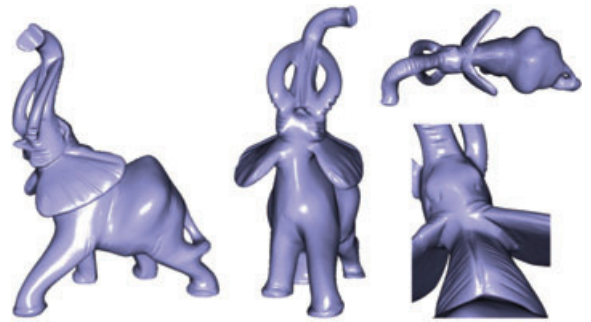


Figure 2: *INRIA/ISTI Livingstone elephant*. 821K vertices, 1,586K triangles.

The method however requires very dense sampling. Another choice of function is the indicator function. [KBH06] align, in the least-squares sense, the gradient of the indicator function with a vector field computed from the oriented input samples. This leads to a Poisson problem: locally supported RBFs are used over an adaptive octree for efficiency and produce excellent results making the method very competitive. Methods based on the minimal surfaces framework with graph cuts of [BK03], such as [HK06], [LB07], also belong to this category: values of the indicator function are assigned to whole elementary volumes over a regular grid so as to globally minimize an energy with a minimum $s - t$ cut optimization. Post-processing is required to remove artifacts arising from the regular grid discretisation.

This family of implicit approaches is sometimes limited by their sensitivity to noise, outliers or non-uniform sampling or even simply by the lack of reliable and consistent normal estimates and orientation.

1.3. Delaunay methods

The other most common approach to surface reconstruction follows the initial intuition of [Boi84] of using a Delaunay triangulation for surface reconstruction: the underlying idea is that when the sampling is noise-free and dense enough, points close on the surface should also be close in space. Eliminating facets of Delaunay tetrahedra according to some criteria should then allow the reconstruction of a triangulated mesh. Among Delaunay-based methods, perhaps the most well-known algorithms are the Crust [ABK98], [ACK01] and the Cocone [ACDL02], [DG03] families of algorithms. Crust algorithms exploit the fact that **Voronoi cells** of points on the surface are elongated in a direction perpendicular to the inferred surface. The extremal vertices of these cells, called *poles*, can be used to estimate the medial axis and filter out facets not belonging to the surface. The Power Crust [ACK01] is an extension, more robust for realistic inputs, that instead relies on the power diagram, a weighted

Voronoi diagram of the poles. A simple modification, suggested in [MAVdF05], improves the robustness of the method to noise. Cocone algorithms use poles in a simpler way to compare facets normal with the vectors to poles. The Robust Cocone [DG06] generalizes the definition of poles to cope with a specific noise model. While [AB99] was the first to provide theoretical guarantees for smooth surfaces with the notion of local feature size and ε -sampling, several of the mentioned algorithms are also provably correct in the absence of noise and outliers or under specific noise model related to the local feature size. In contrast with these computational geometry approaches, [Cha03] proposes to translate the surface convection scheme of [ZOF01] over the Delaunay triangulation of the input points. Degradations of the input data may make these local techniques fail. A notable exception to this rule is the spectral surface reconstruction of [KSO04], which applies a global partitioning strategy to label Delaunay tetrahedra as inside or outside the surface and robustly handles quantities of outliers. A more detailed review of Delaunay-based surface reconstruction can be found in the recent survey of [CG06].

The recent work of [ACSTD07] mixes together a Delaunay-based approach and an implicit one with a spectral method. The method is thus not interpolatory and seems robust to noise. However, its computational requirements seem high and may prevent its application to large amounts of data.

1.4. Deformable models

Active contours [WT88] or deformable models have also been applied to surface reconstruction. The work of [Whi98] relies on the level set framework of [OS88] and the evolution guides the model towards a maximum a posteriori by considering the squared error along the lines of sights. [ZOF01] instead proposes to minimize a functional measuring the distance of the surface to the samples. These evolution methods require a good initialization and are prone to local minima. More recently, [SLS*06] evolve an explicit mesh in a scalar field guided by the local feature size in a coarse to fine manner to avoid local minima and capture details. The method also requires a volumetric grid to evaluate the distance transform and topological changes have to be tracked.

1.5. Other methods

In [OBS05], a reconstruction pipeline designed to cope with measurement noise and varying density is described and consists of different steps: estimation of unoriented normals and weights, computation of an adaptive spherical covering and mesh extraction and cleanup. The pipeline is efficient, has low memory usage, but is not suited to sparse, non-uniform sampling or very noisy data. Bayesian modeling is used in [JWB*06] to reconstruct an augmented point cloud from

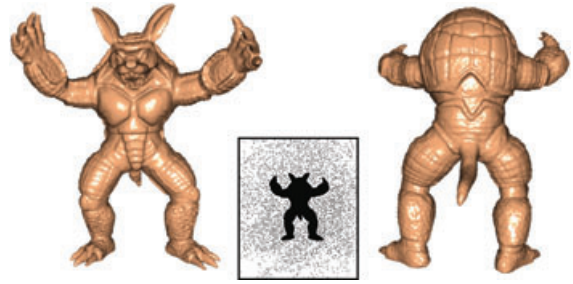


Figure 3: *Stanford armadillo*. 428K vertices, 841K triangles. The original range scans contain many outliers but our method has automatically eliminated them.

the input samples as a maximum a posteriori *w.r.t.* density, smoothness and sharpness priors. A standard MLS surface reconstruction is then applied to reconstruct a surface. The presented results on noisy data sets are impressive, but the method is computationally very expensive even on small data sets. In [GSH*07], the input point cloud is also augmented but with patches of points coming from a training set of prior models and matching according to a multi-scale descriptor of local neighborhoods. Again, a standard MLS algorithm is applied to reconstruct a surface. Finally, to cope with heavily incomplete data, approaches based on templates may be more relevant than traditional surface reconstruction algorithms to complete missing data either with a single template mesh [KS05] or with an entire shape database [PMG*05]. Unfortunately, user interaction is often required with correctly annotated models and the result clearly depends on the used templates.

In the following section, we briefly give some background about applying minimum $s - t$ cuts for optimal surface reconstruction in perspective of the more general cut minimisation problem. We also discuss the shortcomings of previous cut-based methods bearing some similarity to our approach to surface reconstruction from 3D points.

1.6. Minimum cuts for optimal surface reconstruction

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a *directed* graph with vertices $\mathcal{V} = \{v_1, \dots, v_n\}$ and oriented edges \mathcal{E} with weights w_{ij} . Graph partitioning consists in removing the edges connecting two sets of vertices, that is finding two disjoint sets \mathcal{S} and \mathcal{T} such that $\mathcal{S} \cup \mathcal{T} = \mathcal{V}$, $\mathcal{S} \cap \mathcal{T} = \emptyset$. This partition $(\mathcal{S}, \mathcal{T})$ is called a cut and is assigned a cost: the sum of the capacities of the edges going from \mathcal{S} to \mathcal{T} (the oriented edges “crossed” by the cut):

$$c(\mathcal{S}, \mathcal{T}) = \sum_{\substack{v_i \in \mathcal{S} \\ v_j \in \mathcal{T}}} w_{ij}. \quad (1)$$

This cost can be seen as a measure of similarity between the two sets \mathcal{S} and \mathcal{T} .

Surface reconstruction has already been expressed as a problem equivalent to computing a cut with minimum cost. In particular, spectral partitioning methods consider a normalised version [SM00] of the cut cost (to avoid partitioning out small sets of nodes). Such a cost requires non-negative and symmetric weights. Unfortunately, finding a normalised cut with minimum cost is an NP-complete problem, which has to be relaxed into an eigenvalue problem followed by a thresholding to get an approximate solution. The spectral surface reconstruction of [KSO04] applies this criterion to partition subsets of Delaunay tetrahedra but varies slightly on this scheme by modifying the eigenvalue problem: the Laplacian matrix involved in the objective function is altered by allowing negative weights and modifying its diagonal to make it positive definite. While these adjustments void the interpretation of the solution as an optimal normalised cut, in practice, they greatly improve robustness and increase speed. The method is nevertheless still quite slow and besides requiring two successive partitioning steps, several additional ad hoc treatments seem to be needed for it to be applied to real data (filtering out spurious tetrahedra by thresholding and restricting the labeling to tetrahedra near the sample points along their line of sights).

Another approach to graph partitioning adds two special “terminal” vertices to \mathcal{V} , the source s and the sink t . The weights are restricted to non-negative values but asymmetry is allowed. In addition to edges to its incident vertices, each vertex v_i now has links to s and t , respectively, weighted s_i and t_i . An $s - t$ -cut $\mathcal{C} = (\mathcal{S}, \mathcal{T})$ is a cut such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The cost of such a cut may be split as follows:

$$c(\mathcal{S}, \mathcal{T}) = \sum_{\substack{v_i \in \mathcal{S} \setminus \{s\} \\ v_j \in \mathcal{T} \setminus \{t\}}} w_{ij} + \sum_{v_i \in \mathcal{S} \setminus \{s\}} t_i + \sum_{v_i \in \mathcal{T} \setminus \{t\}} s_i. \quad (2)$$

This cost can be interpreted as an energy $E(\mathcal{C})$ attached to the corresponding partition with a “regularising” term between the \mathcal{S} and \mathcal{T} sets (the sum of edge weights w_{ij} , which is actually the cost of the cut without considering the terminals) and a “data” term for \mathcal{S} and \mathcal{T} (the sums of the link weights s_i and t_i). The minimum $s - t$ cut problem consists in finding an $s - t$ cut \mathcal{C} with the smallest cost. According to the Ford-Fulkerson theorem [FF62], this problem is the same as computing the maximum flow from the source s to the sink t : several efficient algorithms with low-polynomial complexity have been developed to solve this problem, making it possible to globally minimise the energy $E(\mathcal{C})$. Most often, methods using minimum $s - t$ cuts for optimal binary segmentation or partitioning straightforwardly apply the *graph cuts* framework of [BK03]. The whole domain of interest is sliced with a regular grid, and the previous equation is interpreted as the sum of the discretisations of an integral over the

interface (the surface S) between \mathcal{S} and \mathcal{T} and two integrals over \mathcal{S} (the outside volume V_{out}) and \mathcal{T} (the inside volume V_{in}):

$$E(S) = \int_S f \, dS + \int_{V_{\text{out}}} g_{\text{out}} \, dV + \int_{V_{\text{in}}} g_{\text{in}} \, dV. \quad (3)$$

The few methods that rely on minimum $s - t$ cuts for optimal surface reconstruction from point clouds adopt this point of view, which has several weaknesses. First, a regular subdivision of space seriously impedes the scalability of minimum $s - t$ cuts as empty space has to be modeled explicitly. Then, the area-based regularisation term is the cause of the “shrinking bias”: the optimal surface for such an energy is the trivial empty one. Workarounds include restricting the domain of interest or adding a uniform balloon force, which requires data-specific adjustment. The minimisation of such an area-based term with other compensating terms often results in over smoothing and is unable to recover thin protusions and concavities. The approach of [HK06] unfortunately suffers from both problems: the domain is regularly subdivided with a grid, which introduces metrication errors and requires a post-processing step to smooth out artifacts. The computational burden of this grid is limited and the empty solution avoided thanks to the use of “banded” graph cuts of [LSGX05]. This actually reduces the minimum $s - t$ cuts optimisation to a local optimisation in a neighborhood of the initially computed proxy surface. [LB07] maximise the flux of a coarsely oriented vector field, which is equivalent to an intelligent balloon force. However, to make this approach robust against undersampling and outliers, the authors have to resort to an area term and a regional term based on lines of sight (a simple non-uniform balloon force). The authors of [LB07] also use a dedicated algorithm that requires proper initialisation to speed up the maximum flow computation on the voxelised volume.

Alternatives to regular grids exist: [KG04] first proposed to use graph cuts on complexes to globally optimise surface functionals and developed the idea of using random sparse complexes for their flexibility over regular subdivisions.

Our method strongly follows [LPK07] and circumvents the two common drawbacks of graph cuts. First, instead of imposing a regular grid, we exploit the adaptivity of the Delaunay triangulation to the input samples. A visibility term taking into account the acquisition procedure is proposed: this term explicitly avoids the empty surface solution. It is a refinement of the term proposed in [LPK07] and is designed in the context of densely (but noisily) sampled surfaces with few lines of sight per sample. In addition to these soft visibility constraints, a simple surface quality term derived from a generalisation of the β -skeleton of [ABE98] in 3D is introduced. This new term advantageously replaces the area-weighted photo-consistency used in [LPK07]: that photo-consistency term is obviously not applicable in our context. It also awkwardly mixed area-based smoothing with

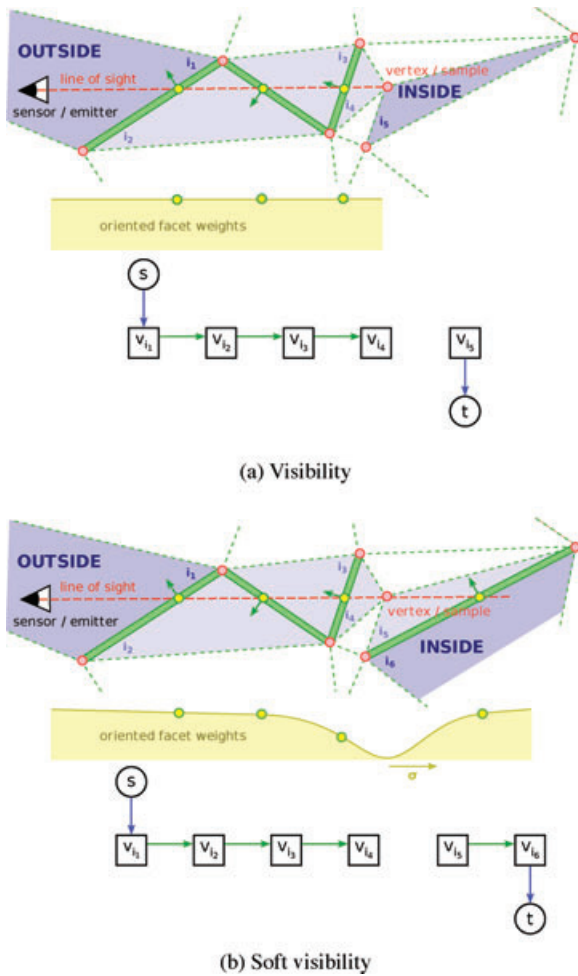


Figure 4: Visibility and soft visibility. How a single line of sight (pink) of a vertex of the triangulation (red) from a sample point to a laser (or to a sensor) contributes to the weights assigned to the origin tetrahedron, to the facets it crosses and to the final tetrahedron (blue).

a purely discrete visibility term. Our new formulation is free from any area-based smoothing that graph cuts methods systematically apply. This term lies at the root of the “shrinking bias” problem and counterbalancing its influence usually requires fine-tuning on each datum.

2. Reconstruction Algorithm

The first step of our method computes the Delaunay triangulation of the 3D point cloud composed of all merged range images. Each finite vertex of this triangulation comes from one range image, and the relative location of the laser and/or the sensor(s) is (at least, approximately) known. As a consequence, the corresponding line(s) of sight emanating from a

vertex to the laser and/or the sensor(s) should not cross the reconstructed surface.

2.1. Optimal labeling of the Delaunay tetrahedra with minimum $s - t$ cut

As previously mentioned, we consider the surface reconstruction problem as a Delaunay tetrahedra labeling problem: tetrahedra are each assigned an inside or outside label. The reconstructed surface, denoted by S in the following, is therefore a union of oriented Delaunay triangles: it is guaranteed to be watertight and intersection free as it bounds a volume. We define an energy $E(S)$, attached to a reconstructed surface S , and gathering two distinct terms:

$$E(S) = E_{\text{vis}}(S) + \lambda_{\text{qual}} E_{\text{qual}}(S). \quad (4)$$

The term $E_{\text{vis}}(S)$ is a sum of penalties for misalignments and wrong orientations of the surface S with respect to the constraints imposed by all the lines of sight from the sample points. The term $E_{\text{qual}}(S)$ penalises the triangles of S unlikely to appear on the true surface. λ_{qual} is a positive constant weighting $E_{\text{qual}}(S)$.

In the next two sections, we present these two energy terms and show how they can be interpreted as costs of $s - t$ cuts on a special graph, allowing our energy to be globally and efficiently minimised with a standard maximum flow algorithm as reminded in section 1.2. The considered graph is obviously related to the Delaunay triangulation: it has vertices representing the Delaunay tetrahedra and directed edges representing the oriented triangles between adjacent tetrahedra. This graph is augmented with the (abstract) source and sink vertices and with links from each tetrahedron to the source and the sink. The vertices linked to the source correspond to tetrahedra labeled as outside and symmetrically, vertices linked to the sink are inside tetrahedra. The directed edges of a cut are triangles on the oriented surface.

Note that the infinite tetrahedra (the tetrahedra lying outside the convex hull of the input points) are also included as vertices in our graph: this allows the labeling to recover open surfaces. Such property is especially useful for outdoor scenes as shown in section 4.

2.2. Surface visibility

In this section, the original visibility term of [LPK07] is first described and then improved to better cope with scarce visibility information and sample noise.

Let us consider one vertex of the triangulation and **one line of sight from this vertex to the laser (or sensor).**

Provided the sample position is noise free, the tetrahedra intersected by this line of sight from this vertex to the sensor or to the laser should be labeled as outside and the tetrahedron

behind the vertex should be labeled as inside. By minimising the number of intersections of this line of sight with the oriented surface and penalising a wrong orientation, we can try imposing this visibility constraint: the triangles crossed by a line of sight from the vertex to the laser (or sensor) are to be penalised. In addition, the surface should go through the vertex originating the line of sight and the last tetrahedron traversed by the line of sight should be labeled as outside. Let us translate this into weights in the corresponding $s - t$ graph (see Figure 4a):

1. the left-most darker blue tetrahedron gets an α_{vis} -weighted link to the source (α_{vis} is a positive constant for the line of sight),
2. the darker green-oriented facets on the left of the vertex, crossed by the line of sight and pointing towards the vertex get an α_{vis} -weighted edge,
3. the darker blue tetrahedron right behind the vertex gets an α_{vis} -weighted link to the sink.

If a confidence measure is available for the line of sight, it should be incorporated into α_{vis} : for instance, [CL96] assigned a confidence value that depends on the angle between the sample normal (evaluated from the range image) and the direction of the line of sight. A surface that goes through the vertex and does not cross the line of sight will not cut any of the weighted edges and links just constructed and will therefore not increase the cost of the $s - t$ cut. This construction is repeated for all available lines of sight of all the vertices of the triangulation by summing their weight contributions: newly generated weights are added to the previously assigned. This can be seen as a kind of “vote” from each line of sight for tetrahedra to be labeled as inside or outside and for oriented triangles to belong to the surface or not. Note that the only tetrahedra to get a non zero-weighted link to the source are those (possibly infinite ones) containing the laser sources or sensors optical centers. This integration over hundreds or thousands of line of sights combined with a global optimisation allows our method to exhibit a strong resilience to different kinds of errors in the input data.

While this construction effectively avoids the empty surface solution and thus the “shrinking bias”, it suffers from several flaws in the context of reconstruction from range scans which sample a surface very densely (but noisily): it tends to generate overly complex surfaces (see Figures 5 and 6) that are bumpy and have many handles inside the model. The measurement noise found in range image is responsible for the bumpiness of the surface, and the large tetrahedra being mislabeled inside the model appear because each sample point only has one or two line(s) of sight: the tetrahedra that should be labeled as inside because they lie behind a vertex are at a much greater risk of being mislabeled because no ray from their vertices will ever intersect them. This is precisely what happens in Figure 5: some inside tetrahedra of the bunny

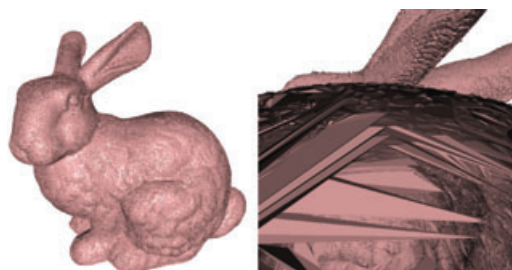


Figure 5: Stanford bunny (with visibility only). A constant visibility term per line of sight is not suited to the reconstruction of densely (but noisily) sampled surfaces with few lines of sight per sample: it tends to generate bumpy surfaces and mislabels many interior tetrahedra.

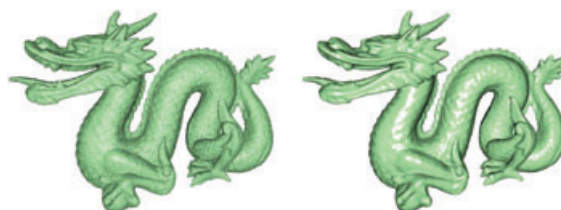


Figure 6: Stanford dragon (visibility vs. soft visibility). On the left, no tolerance is used and the reconstruction is bumpy and overly complex (1,176K vertices, 2,322K triangles). On the right, a reconstruction with tolerance generates a smoother and much coarser mesh (304K vertices, 580K triangles).

get mislabeled as outside. In multi-view stereo, [LPK07] circumvented this problem first by aggregating nearby pairwise reconstructed 3D points together (merging their line of sight information) and also by relying on an area-weighted smoothing. Here, these two problems are solved differently and more elegantly by relaxing the visibility constraints: a tolerance parameter σ is introduced and we modify the edges and links weight constructions. As shown in Figure 4b, the previous construction is extended so that the final tetrahedron on the line of sight does not lie strictly behind the considered vertex but a bit further: it is actually shifted to a distance of 3σ along the line of sight. We also make the oriented facets weights decay with the distance of the intersection of the line of sight with the vertex: each oriented facet intersected by the line of sight of a vertex at a distance d of this vertex gets a weight of $\alpha_{\text{vis}}(1 - e^{-d^2/2\sigma^2})$ from this line of sight. As shown in Figure 7, the value of σ should be set conservatively. However, changing it reasonably allows to generate more or less complex output meshes (see Figure 8). Finally, note that $\sigma = 0$ is equivalent to the first (flawed) visibility weight construction.



Figure 7: *Stanford happy buddha*. 380K vertices, 738K triangles. Setting the tolerance parameter σ too high might create unwanted holes (inside the square).

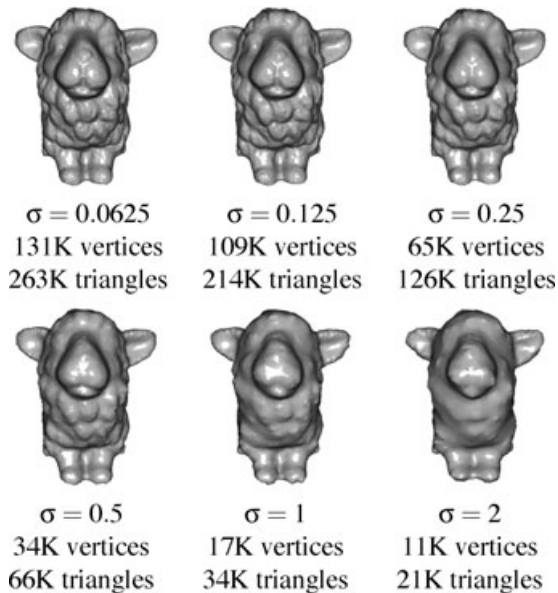


Figure 8: *Influence of the tolerance parameter σ on the reconstruction details and complexity of the UU sheep.*

2.3. Surface quality

The soft visibility constraints of the previous visibility term are sometimes insufficient to get a good reconstruction: handles might still appear due to tiny elongated tetrahedra being wrongly labeled as outside because few (or no) line of sight intersected them. We contrast with [LPK07] by eliminating the cumbersome combination of a purely discrete visibility term with an area-weighted smoothing which requires datum-specific parameter tuning.

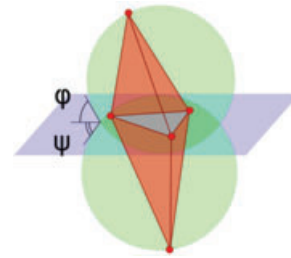


Figure 9: *Soft 3D β -skeleton*. A facet of the triangulation, its two adjacent tetrahedra (red) and their circumscribing spheres (green). Their angles ϕ and ψ with the plane (blue) of the facet influence the weight of this facet.

At first, a simple heuristic can be used to filter out these tetrahedra and ensure the “quality” of the triangles in the output surface: the quality of surface triangle is evaluated as the ratio of the length of their longest edge over the length of their shortest edge (minus one). In our graph, each oriented triangle is weighted with this value. This new quality term tries preventing “badly shaped” triangles from appearing on the surface. In practice, this quality measure gives satisfying results and also happens to be used in the second labeling step of [KSO04]; it is however slightly too discriminating towards skinny triangles that may be required on the surface itself especially in areas where holes in the range images are to be patched (thanks to the Delaunay triangulation).

Instead, we propose to apply a “soft” generalisation to 3D of the β -skeleton described in [ABE98] for curve reconstruction. In 2D, the β -skeleton algorithm computes the Delaunay triangulation of the sample points and chooses the edges of the triangulation whose adjacent triangles have circumspheres centered on opposite sides of the edge and whose radius are both greater than $\beta/2$ times the length of the edge. For dense enough samples, this selection of edges with large empty circumspheres is guaranteed to output a correct reconstruction. Unfortunately, in 3D, almost flat tetrahedra can lie on the surface despite having small empty circumspheres, so the β -skeleton does not generalise well to 3D and may introduce holes. Rather than crudely relabel some tetrahedra selected with a threshold (or even in a greedy way), we integrate this quality criterion into our global optimisation framework. For a given facet of the triangulation, we consider its two adjacent tetrahedra as pictured in Figure 9: the circumscribing spheres of these tetrahedra intersect the plane of the facet at an angle ϕ and ψ . To favor facets with large empty circumspheres, a weight $1 - \min\{\cos(\phi), \cos(\psi)\}$ is added to the two oriented weights of each facet. This way, facets with large empty circumspheres get small penalties for being cut as they are more likely to belong to the surface and conversely, facets with smaller empty spheres are more penalised.

Applying the two constructions described earlier for the surface visibility term and for the surface quality term assigns oriented edge and link weights to nodes in a directed $s - t$ graph. By computing a minimum $s - t$ cut on this graph, an optimal labeling of tetrahedra with respect to these two combined criteria is obtained and a resulting watertight and intersection-free surface mesh can then be extracted.

For very noisy range scans, any interpolating method may output bumpy surfaces when applied directly to the point cloud. As seen in section 4.2, our method can still be used, at least to help bootstrapping local refinements based on partial differential equations [Whi98], [ZOF01], whose initialisation is often problematic. For rendering purpose, in Figures 1–3 and 6–8, at most two steps of a Laplacian-based smoothing were applied.

3. Implementation

The presented algorithm was implemented in C++ and relies on the latest CGAL library [BDTY00] for the computation and traversal of the Delaunay triangulation. It also uses Kolmogorov's max-flow algorithm [BK04] and implementation for the partitioning. We believe our current prototype still allows for improvement in both running time and memory use. The max-flow library was designed for efficiency on grid graphs and energies typically used in computer vision. While our network graphs also have fixed connectivity (each node has four neighbors), the visibility term in our energy design does not lead to short paths from the source to the sink. Switching to a more adapted max-flow algorithm may significantly improve running times. Moreover, due to lim-

itations of the max-flow library, edge weights are required to be computed, and only after the whole network graph can be constructed at once. This means that storage for the weights is duplicated. The graph itself, which can be trivially derived from the Delaunay triangulation, is actually stored twice in memory. Finally, an important increase in memory use can be observed in Table 1 between the weights and the minimum $s - t$ cut computations. The algorithm of Kolmogorov's library caches entire search trees, which again impacts seriously on the memory footprint.

4. Experimental Results

We tested our method on several (variously sized) publicly available sets of range scans from either the Stanford 3D Scanning Repository (bunny, dragon, armadillo and buddha), the AIM@SHAPE Shape Repository (sheep and elephant) or the U.C. Berkeley Computer Animation and Modeling Group (angel) and also on a new outdoor large-scale data set (rue Soufflot). Only for the Stanford and rue Soufflot data sets a reliable estimation of the laser position and/or direction to the sensor(s) was available. This should however not be seen as a strong limitation since for the other data sets, we used less precise, approximate lines of sight and this did not result in significant artefacts in the reconstructions (the more strongly penalised facets and tetrahedra lies inside or far outside the object). Moreover, even if lines of sight can not be reliably guessed, the hidden point removal operator of [KTB07] could potentially be applied from virtual laser positions to recover such visibility information, at least for properly sampled data without outliers.

Table 1: Running time and peak memory use (rounded) of the different steps of our algorithm for the presented reconstructions on an Intel Xeon 3 GHz computer. The elephant and rue Soufflot data sets required a 64-bits environment to complete the computation.

Model	#Points	#Tets	Delaunay	Visibility	Quality	Min. $s - t$ cut	Overall	
Sheep (Fig. 8)	153K	966K	2s	3s	1s	3s	10s	\ll 1m
Bunny (Fig. 12)	362K	2,252K	10s	11s	2s	7s	31s	< 1m
Dragon (Fig. 6)	1,770K	11,383K	38s	68s	15s	59s	180s	3m
Angel (Fig. 1)	2,008K	12,637K	41s	86s	16s	48s	190s	3m 10s
Armadillo (Fig. 3)	2,247K	14,519K	47s	58s	13s	177s	295s	4m 55s
Buddha (Fig. 7)	2,644K	17,167K	62s	120s	14s	74s	271s	4m 31s
Elephant (Fig. 2)	4,413K	27,487K	98s	274s	35s	–	–	–
Elephant (Fig. 2) / 64-bits	4,413K	27,487K	93s	189s	32s	102s	417s	6m 57s
Rue Soufflot (Fig. 14) / 64-bits	6,592K	42,062K	176s	416s	40s	521s	1154s	19m 14s
Sheep (Fig. 8)	153K	966K	48M		67M		133M	
Bunny (Fig. 12)	362K	2,252K	109M		154M		306M	
Dragon (Fig. 6)	1,770K	11,383K	543M		771M		1.6G	
Angel (Fig. 1)	2,008K	12,637K	605M		858M		1.7G	
Armadillo (Fig. 3)	2,247K	14,519K	690M		981M		2.0G	
Buddha (Fig. 7)	2,644K	17,167K	815M		1.1G		2.4G	
Elephant (Fig. 2)	4,413K	27,487K	1.3G		1.8G		–	
Elephant (Fig. 2) / 64-bits	4,413K	27,487K	2.2G		2.7G		6.5G	
Rue Soufflot (Fig. 14) / 64-bits	6,592K	42,062K	3.6G		5.2G		9.9G	

Note: 1G = 1024M.

Table 2: Comparison of running time and peak memory use of our algorithm with Poisson surface reconstruction [KBH06] for the presented reconstructions (maximum octree depth in brackets). The reconstructions of the elephant and rue Soufflot data sets were run in a 64-bits environment.

Model	#Points	Our method	Poisson		Our method	Poisson	
Sheep (Fig. 8)	153K	10s	19s [8]	20s [9]	133M	237M [8]	239M [9]
Bunny (Fig. 12)	362K	31s	65s [9]	66s [10]	306M	300M [9]	311M [10]
Dragon (Fig. 6)	1,770K	180s	180s [10]	461s [11]	1.6G	655M [10]	975M [11]
Angel (Fig. 1)	2,008K	190s	363s [10]	376s [11]	1.7G	785M [10]	856M [11]
Armadillo (Fig. 3)	2,247K	295s	208s [11]	751s [12]	2.0G	728M [11]	1.4G [12]
Buddha (Fig. 7)	2,644K	271s	602s [11]	546s [12]	2.4G	1.2G [11]	1.2G [12]
Elephant (Fig. 2) / 64-bits	4,413K	417s	566s [11]	934s [12]	6.5G	1.1G [11]	1.7G [12]
Rue Soufflot (Fig. 14) / 64-bits	6,592K	1154s	1058s [13]	1608s [14]	9.9G	3.3G [13]	9.5G [14]
			Running time		Peak memory use		

Note: 1G = 1024M.

In all the experiments, the same value of 5 for λ_{qual} was used to balance visibility and quality. Instead of weighting lines of sight with a confidence estimation as in [CL96] and suggested in section 2.2, α_{vis} was purposefully fixed to a constant 32. These two constants were heuristically found on one data set and kept for all the presented results. Finally, the tolerance σ which is supposed to reflect the expected noise level in the data was uniformly set on a per model basis but estimated the same way for every model (1/2 of the median range grid diagonal).

As indicated in Tables 1 and 2 and despite the shortcomings of our implementation, our method proves to be fast and scales well (almost linearly) with the size of the input point cloud both in running time and peak memory use (much better than spectral methods): this might be a hint that modeling the surface reconstruction problem by taking into account as much information as possible about the scanning process actually leads to a better posed minimisation problem. Due to the implementation issues discussed earlier and its interpolatory nature, our method is unfortunately not as competitive as Poisson surface reconstruction [KBH06] in terms of memory resources but compares favorably with it in reconstruction time.

In addition to the previously shown reconstructions, we highlight in the next three sections the robustness of our approach compared to several other methods: the Delaunay-based local algorithms Robust Cocone [DG06] and Power Crust [ACK01], and the implicit methods Adaptive Compactly Supported Radial Basis Functions [OBA*03] and Poisson surface reconstruction [KBH06]. The two implicit methods require oriented normal estimates. While more elaborate methods exist (see [DLS05] for a recent study), these normals were computed in each scan by fitting a plane to the neighbours of each sample with distance-weighted least squares and were correctly oriented using the lines of sight. In the experiments, the normals were obviously estimated from the

modified data to make the comparison fair. More involved methods should not be expected to provide much more precise normal estimations for the different altered data used in the experiments.

Finally, in the last section, Poisson surface reconstruction and our approach are challenged on a difficult large-scale outdoor scene.

4.1. Robustness to non-uniform sampling

Figure 10 illustrates the adaptivity of our method to a non-uniform sampling of the surface (plus a decent amount of measurement noise): a plane partitions the input point set into halves and one of these halves is heavily downsampled. While Robust Cocone seriously degrades when a $128\times$ undersampling is reached, from the beginning, Power Crust splits the two front paws of the sheep. It better handles the undersampling until a $1024\times$ downsampling when its local approach make some important details disappear on the front part of the sheep: the right ear fades away, and the bottom right part around the paw almost vanishes. Poisson progressively shrinks the right part of the sheep, losing all features and Adaptive CS RBF is quickly in trouble. By relying on the visibility information available from the scarce samples, our method is still able to reconstruct a surface that resembles the original model.

4.2. Robustness to noise

While all the data sets used to present our result already contain various amounts of measurement errors, we provide further evidence of the ability of our method to cope with severe amounts of noise. In Figure 11, we adopt a protocol analogous to [KSO04] which add isotropic Gaussian noise to the original point coordinates in the Stanford bunny scans. We instead add anisotropic Gaussian noise along the laser

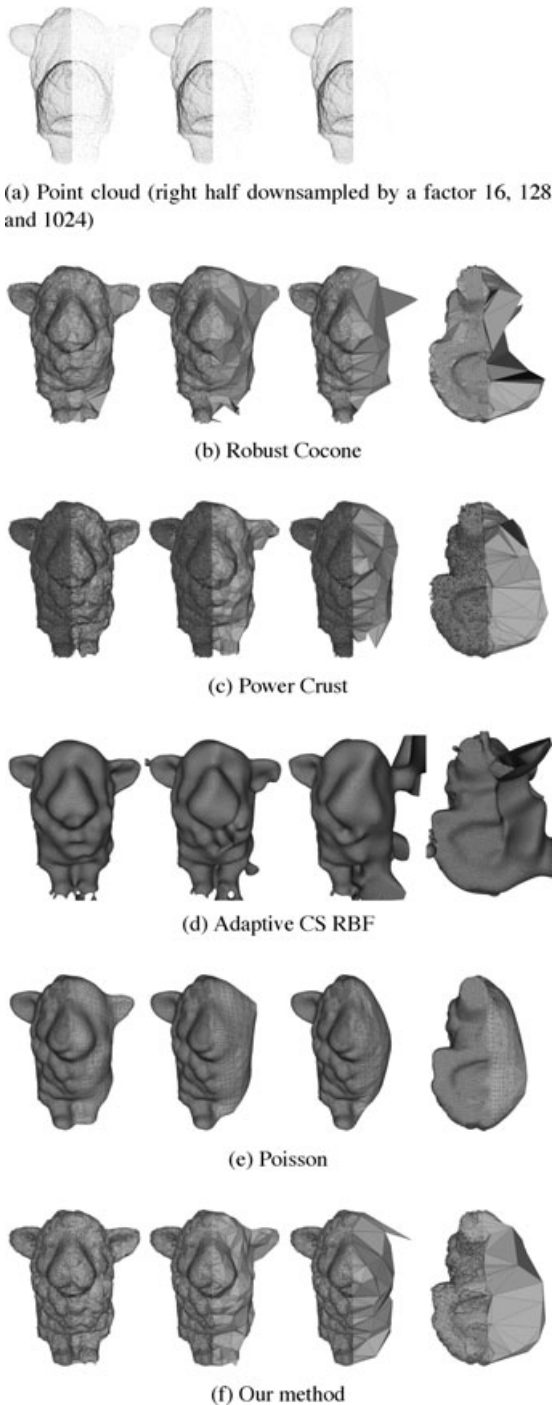


Figure 10: Robustness to undersampling. The right-most bottom view corresponds to a $1024\times$ downsampling.

line of sight only. Results are exceptionally presented not only with the usual fixed tolerance parameter σ but also results with a varying value of σ matching the amount of added noise. The standard deviation of the added noise is

measured in terms of the median length $d = 0.001$ of the range grids diagonals. Point clouds altered with a noise of deviation several times this length are extremely fuzzy, and a correct reconstruction is hardly expected. Holes quickly appear at $+0.8d$ with Power Crust. Robust Cocone which is designed to cope with reasonable amount of noise quickly loses the features of the model (ears, neck and front paws) between $+0.4d$ to $+0.9d$. Adaptive CS RBF is unable to handle the additional noise. Poisson, however, is extremely resistant and still reconstructs a smooth surface even at very high noise levels. It nevertheless begins to seriously degrade after $+2d$. Our method still outputs a genus 0 (albeit bumpy) surface with deviation $+2d$, after this point, the ears of the model begin to fade away and after a $+3d$ deviation the reconstruction irreversibly but slowly degrades. Our method with an adapted σ outputs a much smoother reconstruction and degrades more gracefully.

4.3. Robustness to outliers

As illustrated in Figure 3, range scans usually contain some outliers. On a synthetic example of 26K noise-free points, the spectral surface reconstruction method of [KSO04] was shown to handle 1,200 outliers (or 4.5% of outliers) without any degradation, it then slightly degrades with 1,800 outliers (6.5%) and completely disintegrates with about 10,000 outliers (28%). Here, outliers (along with estimated oriented normals for implicit methods) are added to the original data in much larger amount or ratio. In Figures 12 and 13, we show how the results of other algorithms and ours degrade as randomly generated outliers are gradually added to the 362K points of the Stanford bunny (in fact, we are showing robustness to measurement noise *and* synthetic outliers): the outliers are added scan per scan, their position projects to the range grid and their location is randomly chosen within the bounding box of the range image. This protocol effectively simulates outliers generated during the acquisition. All other tested methods are defeated earlier than ours and unable to recover any useful reconstruction. Poisson surface reconstruction is the strongest contender, but the estimated oriented normal field tends to be inconsistent at outliers and this may actually help this method to filter out outliers.

It is nevertheless pleasant to observe that taking into account the visibility information from the scanning process allows our method to deal with an impressive number of outliers (up to 850,000 or $\sim 70\%$ of outliers) with only very slight degradation of the recovered surface (a handful of outside tetrahedra might be mislabeled now and then). We have consistently observed that our reconstructions irreversibly degrade only when the number of outliers begins to exceed twice the number of inliers: this definitely confirms the suitability of a global optimisation based on lines of sight for outlier removal. While such massive amounts of outliers are not realistic for laboratory acquisitions, our method can be

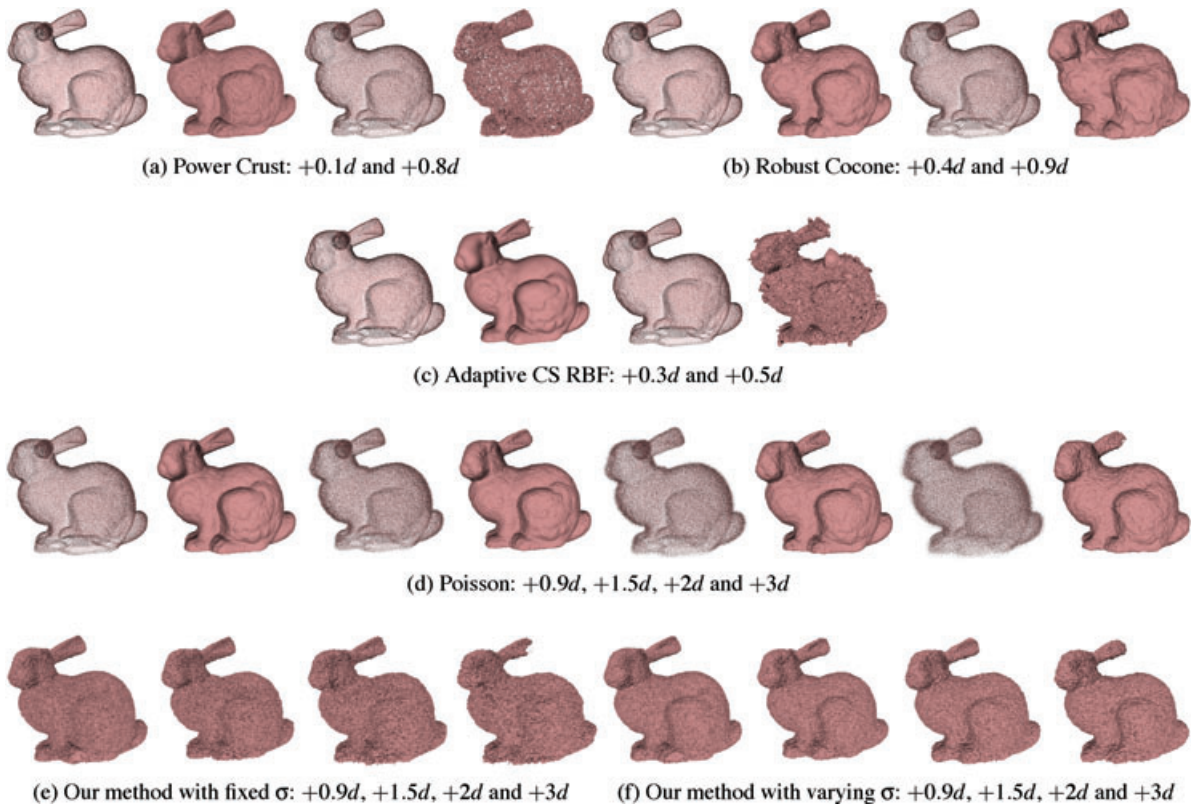


Figure 11: Robustness to noise. The input point cloud is on the left of the first corresponding reconstruction result.

applied with success to outdoor range data or to quasi-dense point clouds from image matching or video tracking that would contain some amount of outliers.

4.4. Large-scale outdoor range data

In this subsection, we show the result of our reconstruction algorithm compared to Poisson surface reconstruction [KBH06] on a challenging data set. The range data was acquired in the rue Soufflot in Paris while driving a mobile vehicle equipped with a time-of-flight range finder paired with a GPS/IMU unit which automatically registers the acquired data (the vehicle was also equipped with several cameras). This datum is particularly difficult: it hardly meets satisfying sampling conditions, it includes moving objects (the pedestrians and the other vehicles), features are present at a wide range of different scales (the street is about 250 m long) and many parts are occluded. In addition the whole point cloud counts 6.7 M samples.

While the presented results would probably require some post-processing before any use in applications, they still demonstrate the potential of our approach even on data acquired without controlled scanning conditions. The running time and peak memory use of our method are about 19 m

and 9.9G. Poisson surface reconstruction was executed at the largest possible depth (14) on the same machine and takes 26 m and 9.5G of memory. As shown in Figures 14–17, our method reconstructs the whole open scene with very thin details (however, for illustration purposes, large triangles close to the convex hull had to be filtered out from the reconstruction by thresholding). Poisson reconstructs a closed scene (which thus required editing), which is less complete than ours (the side streets are much less extended and the roofs are missing, see Figures 14 and 16). It also tends to smooth out the fine structures our method is able to recover (Figures 15 and 17).

5. Conclusion and Future Work

We have presented an algorithm to reconstruct a watertight and intersection-free triangular mesh from range images: this algorithm is based on simple acquisition information. The surface reconstruction is cast as an energy minimisation problem that can be globally optimised by computing a minimum $s - t$ cut problem in a graph. Our approach has two key differences with previous cut-based ones: it does not use a regular subdivision of the domain to approximate integrals but rather label Delaunay tetrahedra and, thanks to a special visibility term and corresponding weight construction, it

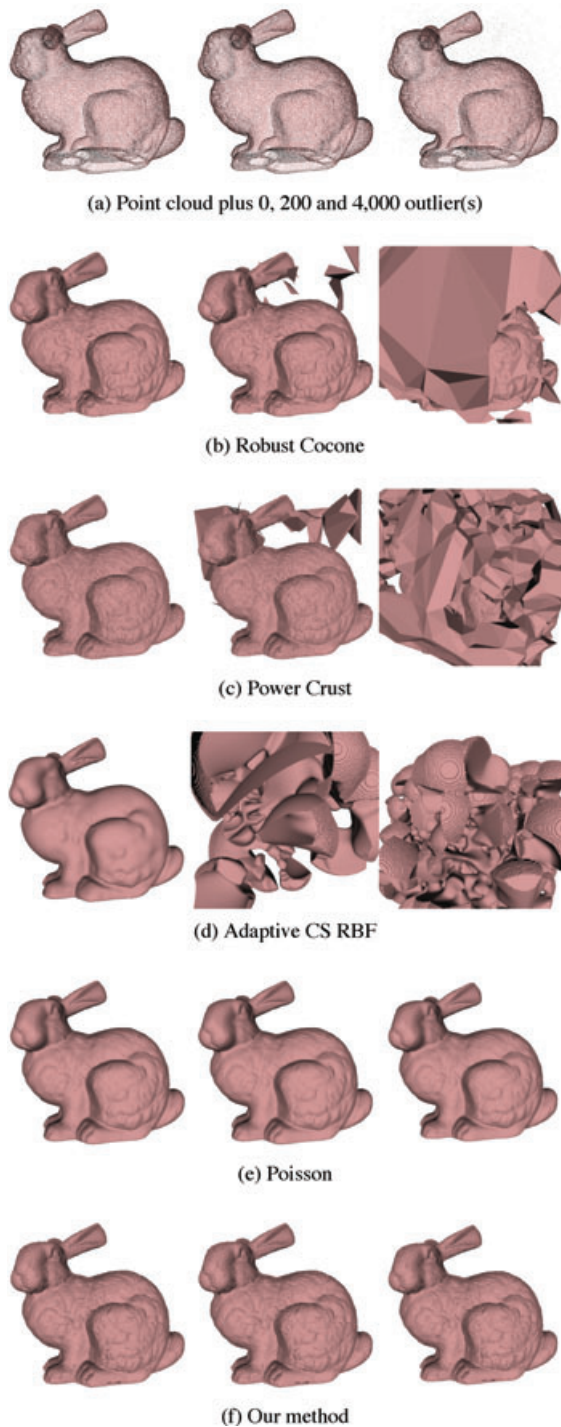


Figure 12: Robustness to relatively few outliers.

explicitly avoid the “shrinking bias” that often plagues graph cuts approaches for segmentation. Moreover, the robustness of our method is demonstrated on several examples: it is able to cope with severe undersampling, noisy data and out-

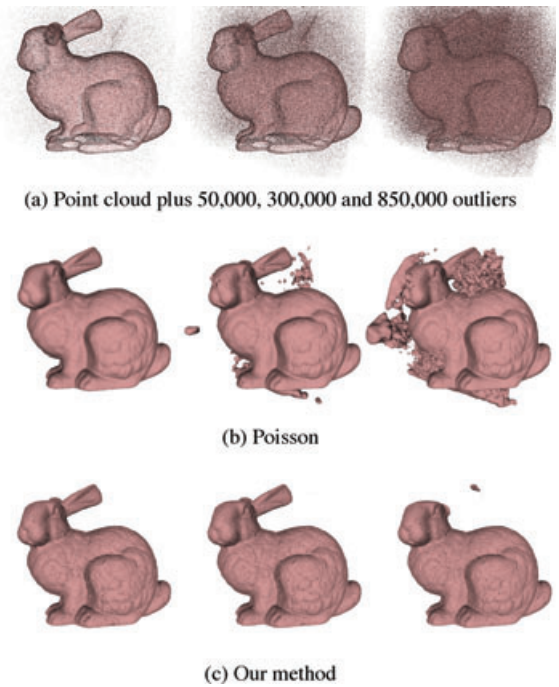


Figure 13: Robustness to large amounts of outliers.

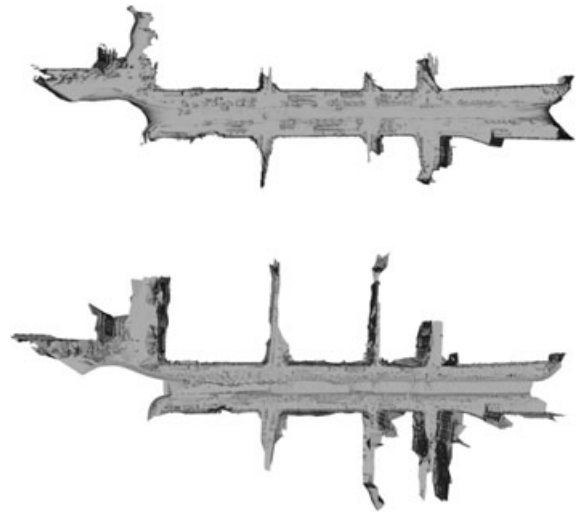


Figure 14: Top view of rue Soufflot. Reconstruction results for Poisson (top) and our method (bottom).

standing amounts of outliers. While such conjunction of data alterations are rare on range scanner laboratory data, they are much more commonplace on outdoor range data acquisitions or on dense point clouds from image matching, which justifies our approach and makes our method a versatile tool for surface reconstruction. In spite of a currently limited



Figure 15: *Reconstruction details of rue Soufflot. Acquired images (top), corresponding reconstruction results of Poisson (middle) and our method (bottom).*

implementation, great scalability is achieved with respect to the size of the input point cloud both in running times and memory use.

Our surface reconstruction algorithm is however interpolatory, which is probably one of its main limitations. A possible remedy would be to try to account for the noise model in the spatial subdivision, or more realistically, to split the visibility-based outlier filtering and the final surface reconstruction into two distinct successive steps. Another limitation is the scalability to very large data sets that are becoming more widespread in range scanning [LPC*00], [BMOI08]. Some reconstruction algorithms have been extended with out-of-core or streaming versions to handle massive data sets with limited memory [FCS07], [ACA07], [BKBH07]. In particular, Poisson surface reconstruction [KBH06] is particularly well suited to such extension [BKBH07] with its simple octree subdivision structure, locally supported RBFs, and a Poisson equation that results in a sparse symmetric system suitable to multigrid techniques. In contrast, our method is a special linear programming problem over an unstructured domain where the visibility of a sample may have a long range influence. While streaming Delaunay computation exists [ILSS06] and the work of [ACA07] has shown success in extending a local Delaunay-based method to a streaming one, our case seems more difficult and a more plausible first step in this direction would be a simpler adaptive filtering of the input point cloud as in [ACA06] but based on visibility to eliminate redundant or inconsistent sample points before applying our final

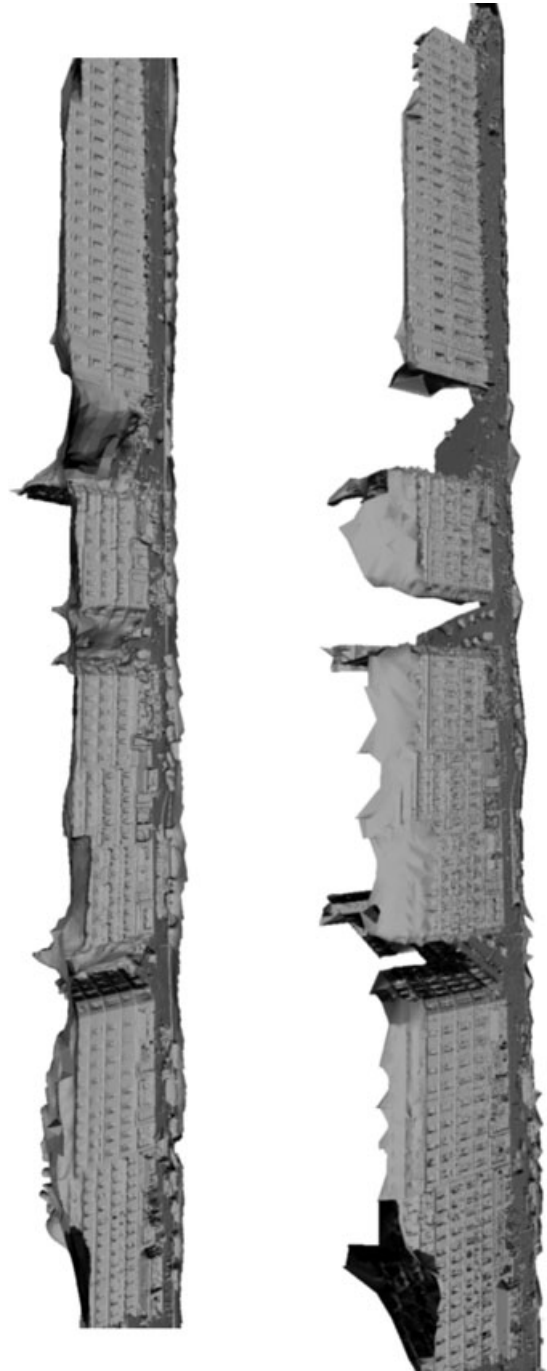


Figure 16: *Panorama view of rue Soufflot. Reconstruction results of Poisson (left) and our method (right).*

reconstruction. Finally, applications and generalisations of our approach to possibly other problems are expected, mesh repair or shape reconstruction from cross-sections, for instance.

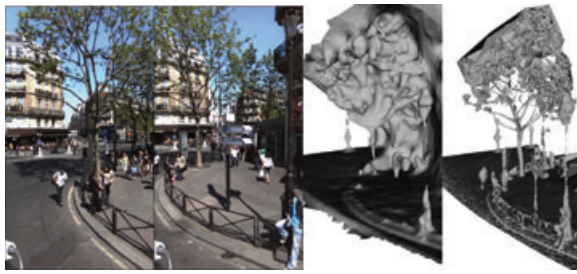


Figure 17: Reconstruction details of rue Soufflot. Acquired images (left), corresponding reconstruction results of Poisson (center) and our method (right).

Acknowledgements

The authors thank the various repositories and institutes sharing laser data: the U.C. Berkely Computer Animation and Modeling Group for the angel model, the Utrecht University, INRIA and ISTI through the AIM@SHAPE Shape repository for the sheep and elephant model, and of course, the Stanford 3D scanning repository for the armadillo, bunny, buddha and dragon models. The images and range data of the rue Soufflot are provided courtesy of MATIS, French Mapping Agency (IGN).

The authors also express gratitude to Brian Curless for explaining and providing documentation on the exact acquisition process behind the various Stanford data.

Finally the authors of [ACK01, OBA*03, DG06, KBH06] are thanked for sharing the implementations of their respective reconstruction algorithms.

References

- [AB99] AMENTA N., BERN M.: Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry* 22, 4 (1999), 481–504.
- [ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics* 9, 1 (2003), 3–15.
- [ABE98] AMENTA N., BERN M., EPPSTEIN D.: The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing* 60, 2 (1998), 125–135.
- [ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new Voronoi-based surface reconstruction algorithm. In *ACM SIGGRAPH* (1998), pp. 415–421.
- [ACA06] ALLÈGRE R., CHAINE R., AKKOUCHE S.: A dynamic surface reconstruction framework for large unstructured point sets. In *Symposium on Point-Based Graphics* (2006), pp. 17–26.
- [ACA07] ALLÈGRE R., CHAINE R., AKKOUCHE S.: A streaming algorithm for surface reconstruction. In *Symposium on Geometry Processing* (2007), pp. 79–88.
- [ACDL02] AMENTA N., CHOI S., DEY T. K., LEEKHA N.: A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and Applications* 12, 1–2 (2002), 125–141.
- [ACK01] AMENTA N., CHOI S., KOLLURI R.: The power crust. In *ACM Symposium on Solid Modeling and Applications* (2001), pp. 249–260.
- [ACSTD07] ALLIEZ P., COHEN-STEINER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry Processing* (2007), pp. 39–48.
- [BDTY00] BOISSONNAT J.-D., DEVILLERS O., TEILLAUD M., YVINEC M.: Triangulations in CGAL. In *ACM Symposium on Computational Geometry* (2000), pp. 11–18.
- [BK03] BOYKOV Y., KOLMOGOROV V.: Computing geodesics and minimal surfaces via graph cuts. In *IEEE International Conference on Computer Vision* (2003), pp. 26–33.
- [BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 9 (2004), 1124–1137.
- [BKBH07] BOLITHO M., KAZHDAN M., BURNS R., HOPPE H.: Multilevel streaming for out-of-core surface reconstruction. In *Symposium on Geometry Processing* (2007), pp. 69–78.
- [BMOI08] BANNO A., MASUDA T., OISHI T., IKEUCHI K.: Flying laser range sensor for large-scale site-modeling and its applications in Bayon digital archival project. *International Journal of Computer Vision* 78, 2–3 (2008), 207–222.
- [Boi84] BOISSONNAT J.-D.: Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics* 3, 4 (1984), 266–286.
- [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 for 3D objects with radial basis functions. In *ACM SIGGRAPH* (2001), pp. 67–76.
- [CG06] CAZALS F., GIESEN J.: *Effective Computational Geometry for Curves and Surfaces*. Mathematics and Visualization. Springer, ch. Delaunay Triangulation Based Surface Reconstruction, (2006), pp. 231–276.

- [Cha03] CHAINE R.: A geometric convection approach of 3-D reconstruction. In *Symposium on Geometry Processing* (2003), pp. 218–229.
- [CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *ACM SIGGRAPH* (1996), pp. 303–312.
- [DG03] DEY T. K., GOSWAMI S.: Tight cocone: A watertight surface reconstructor. In *ACM Symposium on Solid Modeling and Applications* (2003), pp. 127–134.
- [DG06] DEY T. K., GOSWAMI S.: Provable surface reconstruction from noisy samples. *Computational Geometry: Theory and Applications* 35, 1 (2006), 124–141.
- [DLS05] DEY T. K., LI G., SUN J.: Normal estimation for point clouds: A comparison study for a Voronoi based method. In *Symposium on Point-Based Graphics* (2005), pp. 39–46.
- [FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. In *ACM SIGGRAPH* (2005), vol. 24, pp. 544–552.
- [FCS07] FIORIN V., CIGNONI P., SCOPIGNO R.: Out-of-core MLS reconstruction. In *International Conference on Computer Graphics and Imaging* (2007), pp. 27–34.
- [FF62] FORD L. R., FULKERSON D. R.: *Flows in Networks*. Princeton University Press, 1962.
- [GSH*07] GAL R., SHAMIR A., HASSNER T., PAULY M., COHEN-OR D.: Surface reconstruction using local shape priors. In *Symposium on Geometry Processing* (2007), pp. 253–262.
- [HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *ACM SIGGRAPH* (1992), pp. 71–78.
- [HK06] HORNUNG A., KOBELT L.: Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Symposium on Geometry Processing* (2006), pp. 41–50.
- [ILSS06] ISENBURG M., LIU Y., SHEWCHUK J., SNOEYINK J.: Streaming computation of Delaunay triangulations. In *ACM SIGGRAPH* (2006), vol. 25, pp. 1049–1056.
- [JWB*06] JENKE P., WAND M., BOKELOH M., SCHILLING A., STRASSER W.: Bayesian point cloud reconstruction. In *EUROGRAPHICS* (2006), vol. 25, pp. 379–388.
- [KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Symposium on Geometry Processing* (2006), pp. 61–70.
- [KG04] KIRSANOV D., GORTLER S. J.: *A Discrete Global Minimization Algorithm for Continuous Variational Problems*. Technical Report TR-14-04, Harvard Computer Science, 2004.
- [Kol08] KOLLURI R. K.: Provably good moving least squares. *ACM Transactions on Algorithms* 4, 2 (2008).
- [KS05] KRAEVOY V., SHEFFER A.: Template-based mesh completion. In *Symposium on Geometry Processing* (2005), pp. 13–22.
- [KSO04] KOLLURI R., SHEWCHUK J. R., O'BRIEN J. F.: Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing* (2004), pp. 11–21.
- [KTB07] KATZ S., TAL A., BASRI R.: Direct visibility of point sets. In *ACM SIGGRAPH* (2007), vol. 26.
- [LB07] LEMPITSKY V., BOYKOV Y.: Global optimization for shape fitting. In *IEEE Conference on Computer Vision and Pattern Recognition* (2007), pp. 1–8.
- [Lev03] LEVIN D.: Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, Berlin, Springer, Brunnett G., Hamann B., Müller H., Linsen L. (Eds.), pp. 37–49, 2003.
- [LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINTON M., ANDERSON S. E., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital Michelangelo project: 3D scanning of large statues. In *ACM SIGGRAPH* (2000), pp. 131–144.
- [LPK07] LABATUT P., PONS J.-P., KERIVEN R.: Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts. In *IEEE International Conference on Computer Vision* (2007), pp. 1–8.
- [LS81] LANCASTER P., SALKAUSKAS K.: Surfaces generated by moving least-squares methods. *Mathematics of Computation* 37 (1981), 141–158.
- [LSGX05] LOMBAERT H., SUN Y., GRADY L., XU C.: A multilevel banded graph cuts method for fast image segmentation. In *IEEE International Conference on Computer Vision* (2005), pp. 259–265.
- [MAVdF05] MEDEROS B., AMENTA N., VELHO L., DE FIGUEIREDO L. H.: Surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing* (2005), pp. 53–62.
- [OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. In *ACM SIGGRAPH* (2003), vol. 22, pp. 463–470.
- [OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: 3D scattered data approximation with adaptive compactly supported

- radial basis functions. In *Shape Modeling International* (2004), pp. 153–164.
- [OBS05] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: An integrating approach to meshing scattered point data. In *Symposium on Solid and Physical Modeling* (2005), pp. 61–69.
- [OS88] OSHER S., SETHIAN J. A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 79, 1 (1988), 12–49.
- [PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L.: Example-based 3D scan completion. In *Symposium on Geometry Processing* (2005), pp. 23–32.
- [SLS*06] SHARF A., LEWINER T., SHAMIR A., KOBELT L., COHEN-OR D.: Competing fronts for coarse-to-fine surface reconstruction. In *EUROGRAPHICS* (2006), vol. 25, pp. 389–398.
- [SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.
- [SOS04] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH* (2004), vol. 23, pp. 896–904.
- [Whi98] WHITAKER R. T.: A level set approach to 3D reconstruction from range data. *International Journal of Computer Vision* 29, 3 (1998), 203–231.
- [WT88] WITKIN M. K. A., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision* 1, 4 (1988), 321–331.
- [ZOF01] ZHAO H. K., OSHER S., FEDKIW R.: Fast surface reconstruction using the level set method. In *IEEE Workshop on Variational and Level Set Methods in Computer Vision* (2001), pp. 194–201.