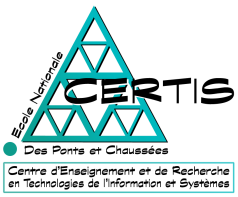
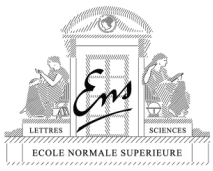
Voronoi Features Cut for Surface Reconstruction from Multiple Views

Patrick Labatut
Jean-Philippe Pons
Renaud Keriven

Research Report - CERTIS 07-34
April 2007





CERTIS - École des ponts
6-8, avenue Blaise Pascal
77420 Champs-sur-Marne
France

DI - École normale supérieure
45, rue d'Ulm
75005 Paris
France

Voronoi Features Cut for Surface Reconstruction from Multiple Views

Coupe minimale dans un diagramme de Voronoi pour la reconstruction de surface à partir de plusieurs images

Patrick Labatut[1]
Jean-Philippe Pons[1,2]
Renaud Keriven[1,2]

[1]DI, École normale supérieure, 75005 Paris, France, http://www.di.ens.fr/
[2]CERTIS, ENPC, 77455 Marne la Vallee, France, http://certis.enpc.fr/

Abstract

We present a novel method to reconstruct the 3D shape of a scene from several calibrated images. Our motivation is that most existing multi-view stereovision approaches require some knowledge of the scene extent and often even of its approximate geometry (*e.g.*visual hull). This makes these approaches mainly suited to compact objects admitting a tight enclosing box, imaged on a simple or a known background. In contrast, our approach focuses on large-scale cluttered scenes under uncontrolled imaging conditions. It first generates a quasi-dense 3D point cloud of the scene by matching keypoints across images in a lenient manner, thus possibly retaining many false matches. Then it builds an adaptive tetrahedral decomposition of space by computing the 3D Delaunay triangulation of the 3D point set. Finally, it reconstructs the scene by labeling Delaunay tetrahedra as empty or occupied, thus generating a triangular mesh of the scene. A globally optimal label assignment, as regards photo-consistency of the output mesh and compatibility with the visibility of keypoints in input images, is efficiently found as a minimum cut solution in a graph.

Résumé

Nous présentons une nouvelle méthode de reconstruction de la forme 3D d'une scène à partir de plusieurs images calibrées. Notre approche est motivée par le fait que la plupart des autres approches existantes pour la reconstruction requièrent des informations supplémentaires sur l'étendue de la scène voire même sa forme approximative (c'est-à-dire son enveloppe visuelle). Ce pré-requis empêche l'utilisation de ces méthodes pour des scènes autres que celles représentant des objets compacts assortis d'une boîte englobante précise, filmés sur un fond connu. Au contraire, notre approche se focalise sur de larges scènes encombrées sans aucun contrôle des conditions de prise de vues. Tout d'abord, un nuage presque dense de points 3D est généré en appariant le plus possible de points clés entre les différentes images (beaucoup de faux appariements pouvant être générés). Une décomposition adaptative de l'espace en tétrahèdres est ensuite construite à partir de ce nuage de points en calculant sa triangulation de Delaunay. Enfin, la scène est reconstruite en étiquetant chaque tétrahèdre comme intérieur ou extérieur, et l'interface entre l'intérieur et l'extérieur permet d'extraire un maillage de la scène. Un étiquetage optimal des tétrahèdres vis-à-vis de la photo-consistence du maillage créé et de la compatibilité de ce maillage avec les contraintes de visibilité (imposées par les points clés issus des images) est obtenu efficacement comme coupe minimale dans un graphe.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Novelty of our approach 2

2 Background **3**
2.1 Keypoint extraction and description 3
2.2 Delaunay triangulation 3
2.3 Energy minimization by graph cuts 4

3 Reconstruction Method **6**
3.1 Quasi-dense 3D point cloud generation 6
3.2 Match aggregation and Delaunay triangulation 6
3.3 Surface extraction 7
3.3.1 Surface visibility 8
3.3.2 Surface photo-consistency 9
3.3.3 Surface smoothness 10

4 Experimental Results **10**
4.1 Implementation aspects 10
4.2 Temple 11
4.3 Toys 11

5 Conclusion and Future Work **12**

Bibliography **12**

1 Introduction

1.1 Motivation

As pointed out in the review by Seitz *et al.* [32], most top-performing algorithms for dense multi-view stereo reconstruction require significant knowledge of the geometry of the scene. This ranges from a tight bounding box to a closer approximation by the visual hull.

The visual hull is defined as the intersection of cones generated by the silhouettes of the objects in the input views [25]. This technique requires an accurate segmentation of input images. In real-life examples, however, such segmentation is not available or even feasible. In practice, visual hull computation only applies to datasets obtained under controlled imaging conditions, namely on a simple or a known background.

Despite this serious limitation, in the last few years, a number of multi-view stereovision algorithms exploiting visual hull have been proposed. They rely on visual hull either as an initial guess for further optimization [16, 12, 20, 35, 39, 41, 43], as a soft constraint [12] or even as a hard constraint [16, 34] to be fulfilled by the reconstructed shape.

While the unavailability of silhouette information discards many of the top-performing multi-view stereovision algorithms, the requirement for the ability to handle large-scale scenes discards most of the others, and in particular volumetric methods, *i.e.*methods based on a regular decomposition of the domain into elementary cells, typically voxels. Obviously, this approach is mainly suited to compact objects admitting a tight enclosing box, as its computational and memory cost quickly becomes prohibitive when the size of the domain increases.

Volumetric multi-view stereovision methods include space carving [9, 24, 33, 40, 42], level sets [13, 21, 31], and volumetric graph cuts [8, 20, 26, 35, 39, 41]. Actually, what distinguishes these three categories is the type of optimization they rely on: a greedy occupancy assignment in space carving, a surface deformation driven by a gradient descent in level sets, and a global combinatorial optimization in graph cuts.

Large-scale cluttered scenes for which no reliable initial guess of geometry is available also disqualify the deformable model framework [11, 13, 12, 21, 27, 31]. Indeed, it is based on a local optimization by gradient descent. As a result, it is highly sensitive to initial conditions.

The multi-view stereovision methods which have proven more adapted to reconstruct large-scale scenes (*e.g.*outdor architectural scenes) are those representing geometry by several depth maps [18, 17, 22, 36, 37, 38]. However, their performance for complete reconstruction seems to be lower than previously discussed approaches, either as regards accuracy or completeness of the obtained

model. This may be due to the difficulty to handle visibility globally and consistently in this approach. Moreover, in the complete reconstruction case, the several partial models of the scene have to be fused at post-processing using a volumetric technique [10].

From the above discussion, we draw the conclusion that, although very impressive progress has been made in the last few years in the multi-view stereovision problem as regards reconstruction accuracy, novel algorithms that can handle more general scenes are still needed.

1.2 Novelty of our approach

In this paper, we propose a novel multi-view reconstruction approach adapted to large-scale cluttered scenes under uncontrolled imaging conditions. Our method first generates a quasi-dense 3D point cloud of the scene by matching keypoints across images in a lenient manner, thus possibly retaining many false matches. Then it builds an adaptive tetrahedral decomposition of space by computing the 3D Delaunay triangulation of the 3D point set. Finally, it reconstructs the scene by labeling Delaunay tetrahedra as empty or occupied, thus generating a triangular mesh of the scene. A globally optimal label assignment, as regards photo-consistency of the output mesh and compatibility with the visibility of keypoints in input images, is efficiently found as a minimum cut solution in a graph.

Our method shares with existing multi-view graph cuts approaches [8, 16, 20, 22, 26, 34, 35, 39, 41] the desirable property of yielding an exact global optimum of an energy functional. Compared to these methods, however, our approach enjoys a unique combination of desirable features:

1. It uses a fully adaptive unstructured tetrahedral decomposition of space, namely the Delaunay triangulation of a quasi-dense point sample of the surface of the scene, in constrast with a regular subdivision used in volumetric graph cuts [8, 20, 26, 35, 39, 41]. This yields several significant benefits:
    - it removes the need for a predefined bounding box of the scene, since the Delaunay triangulation seamlessly accounts for the convex hull of the point cloud.
    - it considerably alleviates quantization artefacts, namely the stair-casing effect.
    - it keeps the computation and memory cost sustainable on large-scale scenes, since empty space regions are represented by few large tetrahedra.
    - it allows to directly output a high-quality triangular mesh of the scene, free of self-intersections.

2. It exploits visibility information coming from keypoints to guide the position of the surface. As a result, it avoids the mininum cut solution from being an empty surface. Hence it exonerates from the different techniques proposed in the literature so far to solve this problem: a heuristic ballooning term [26, 41], a restriction of the feasible set using silhouette information [16, 20, 34, 35, 39, 43], or a maximization of photo-flux [8]. Moreover, this visibility information is not enforced as a hard constraint but integrated in the very optimization framework, thus yielding robustness to outliers.

3. It can handle closed as well as open scenes. For example, it can simultaneously recover the walls of an indoor scene and a complete reconstruction of objects seen from all sides in the input images.

The remainder of this paper is organized as follows. Section 2 gives some background on the different techniques needed in our approach: interest point detectors, Delaunay triangulation and graph cuts. In Section 3, we describe in detail the different steps of our multi-view stereo reconstruction algorithm. Section 4 discusses implementation aspects and presents some numerical experiments that demonstrate the potential of our approach for reconstructing large-scale cluttered scenes from real-world data.

2 Background

2.1 Keypoint extraction and description

Our method relies on the extraction of robust keypoints that can be matched across different viewpoints: we use the keypoint extraction and description method of Lowe [29]. The first stage of the Scale-invariant feature transform (SIFT) searches for scale-space extrema in the difference-of-Gaussian function convolved with the image in order to find interest points [28]. The second stage associates a descriptor (a high dimension vector) to each keypoint localization: this descriptor represents the distributions of smaller scale features in the neighbourhood of the detected point, it is invariant to scale and rotation and is robust to small affine or projective deformations and illumination changes. It has also been shown to perform among the very best descriptors [30] and has become one of the most widely used descriptor in practice nowadays, justifying our choice.

2.2 Delaunay triangulation

The following definitions are taken from a computational geometry textbook [6]. Let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be a set of points in $\mathbb{R}^d$. The Voronoi cell associated to a

point p_i, denoted by $V(p_i)$, is the region of space that is closer from p_i than from all other points in $\mathcal{P}$:

$$V(p_i) = \{\, p \in \mathbb{R}^d \;:\; \forall\, j \neq i,\ \|p - p_i\| \leq \|p - p_j\| \,\}$$

$V(p_i)$ is the intersection of $n-1$ half-spaces bounded by the bisector planes of segments $[p_i p_j],\ j \neq i$. $V(p_i)$ is therefore a convex polytope, possibly unbounded. The Voronoi diagram of $\mathcal{P}$, denoted by $\mathrm{Vor}(\mathcal{P})$, is the partition of space induced by the Voronoi cells $V(p_i)$.

The Delaunay triangulation $\mathrm{Del}(\mathcal{P})$ of $\mathcal{P}$ is defined as the geometric dual of the Voronoi diagram: there is an edge between two points p_i and p_j in the Delaunay triangulation if and only if their Voronoi cells $V(p_i)$ and $V(p_j)$ have a non-empty intersection. It yields a triangulation of $\mathcal{P}$, that is to say a partition of the convex hull of $\mathcal{P}$ into d-dimensional simplices (*i.e.*into triangles in 2D, into tetrahedra in 3D, and so on). Figure 1 displays an example of a Voronoi diagram and its associated Delaunay triangulation in the plane.

The algorithmic complexity of the Delaunay triangulation of n points is $\mathcal{O}(n \log n)$ in 2D, and $\mathcal{O}(n^2)$ in 3D. However, as was recently proven in [2], the complexity in 3D drops to $\mathcal{O}(n \log n)$ when the points are distributed on a smooth surface, which is the case of interest here.

Our choice of Delaunay triangulation as a space subdivision for multi-view stereo reconstruction is motivated by the following remarkable property: under some assumptions, and especially if $\mathcal{P}$ is a "sufficiently dense" sample of a surface, in some sense defined in [1], then a good approximation of the surface is "contained" in $\mathrm{Del}(\mathcal{P})$, in the sense that the surface can be accurately reconstructed by selecting an adequate subset of the triangular facets of the Delaunay triangulation.

2.3 Energy minimization by graph cuts

Given a finite directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$ with non-negative weights (capacities), and two special vertices, the source s and the sink t, an s-t-cut $\mathcal{C} = (\mathcal{S}, \mathcal{T})$ is a partition of $\mathcal{V}$ into two disjoints sets $\mathcal{S}$ and $\mathcal{T}$ such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The cost of the cut is the sum of the capacity of all the edges going from $\mathcal{S}$ to $\mathcal{T}$: $c(\mathcal{S}, \mathcal{T}) = \sum_{(p,q) \in \mathcal{S} \times \mathcal{T} | p \to q \in \mathcal{E}} w_{pq}$. The minimum s-t-cut problem consists in finding a cut $\mathcal{C}$ with the smallest cost: the Ford-Fulkerson theorem [14] states that this problem is equivalent to computing the maximum flow from the source s to the sink t and many classical algorithms exist to efficiently solve this problem. Such a cut can be viewed as a binary labeling of the nodes: by building an appropriate graph, many segmentation problems in computer vision can be solved very efficiently [19]. More generally, global minimization of a whole class of energy is achieved by graph cuts [23].

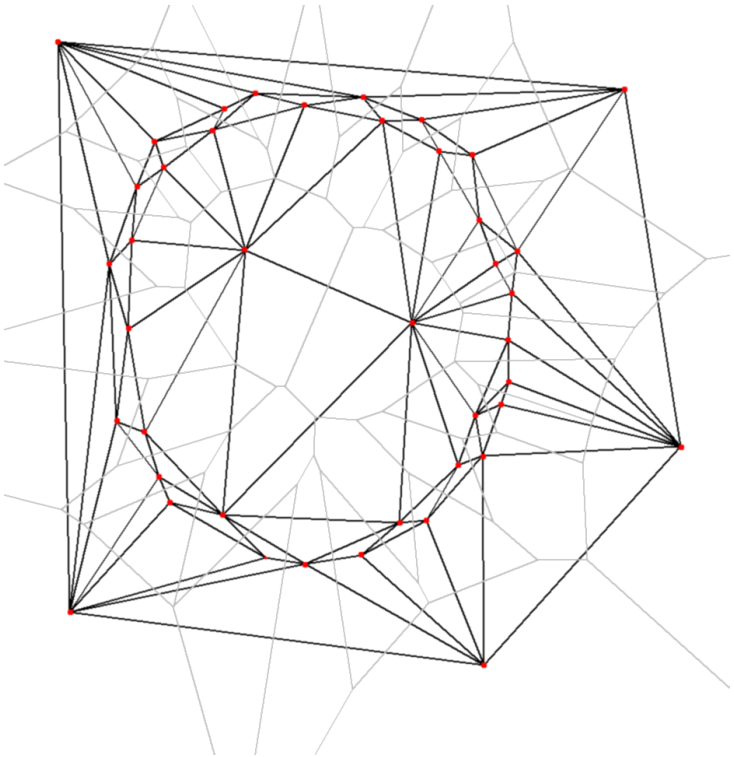


Figure 1: The Voronoi diagram (gray edges) of a set of 2D points (red dots) and its associated Delaunay triangulation (black edges).

The graph considered in our approach derives from an adaptive space decomposition provided by the Delaunay triangulation. This differs from the graphs commonly used in computer vision, which are often regular grids in the input images or in the bounding volume of the scene.

3 Reconstruction Method

Our algorithm can be decomposed in four main steps: the first step is quite straightforward as it reduces to extracting features from the input views. The keypoints are then matched pair-wise between different views by taking epipolar geometry into account: these matches enable the generation a quasi-dense 3D point cloud, which is later refined and structured by incrementally building a Delaunay triangulation and merging 3D points that are close enough. Finally a graph-cut optimization is used to extract the surface of the scene from this triangulation.

3.1 Quasi-dense 3D point cloud generation

The first step in our method is the generation of a quasi-dense 3D point cloud. To this end, pairs of keypoints are matched across different views. The usual way of obtaining robust matches is, given one keypoint, to find the best match in the other image, and to keep it provided its matching score is significantly better than the second best matching score. Here, however, as the global optimization in the final step is able to cope with false matches, we favor density over robustness and we admit a lot of false positives. To achieve this, given one keypoint, we always keep the best match along the epipolar line, plus we keep all other matches along the epipolar line whose matching scores are not significantly lower than the best match. This step outputs a 3D point cloud by computing the 3D position associated to each match.

3.2 Match aggregation and Delaunay triangulation

The next step in our method consists in adding some structure to the previous 3D point cloud, while efficiently aggregating matches in tuples. This is accomplished by incrementally building a Delaunay triangulation of the 3D point set. Each vertex of the triangulation does not only store its position, it also maintains the list of keypoints it originates from. Each time a candidate point from the original 3D point cloud is to be added, its nearest neighbour in the triangulation is found (this query is very efficient in a Delaunay triangulation [5]) and the maximum reprojection error between the two 3D points is computed.

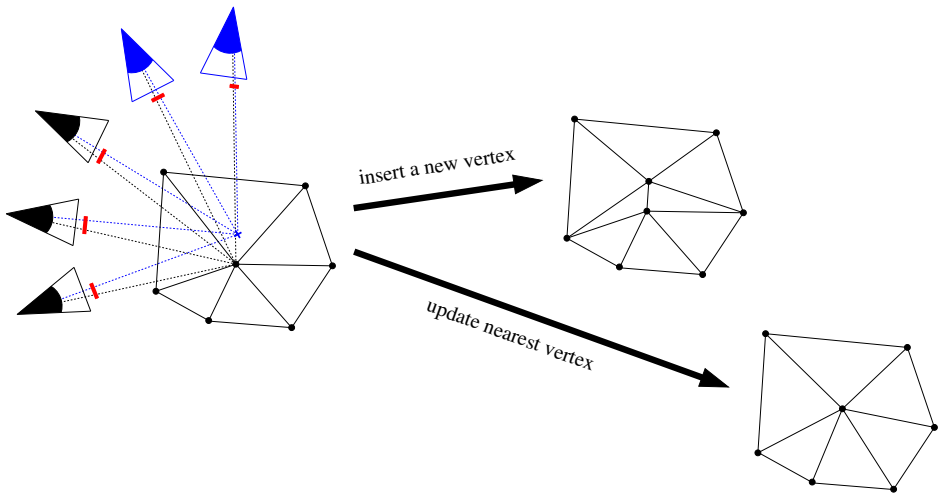


Figure 2: A candidate point (blue cross) updates the Delaunay triangulation depending on the maximum reprojection error between it and the nearest vertex: either it is inserted as a new vertex or it updates the position of the nearest vertex.

As illustrated in Figure 2, two different cases can occur. If the maximum reprojection error is above some threshold, the candidate point is regarded as a distinct point and is inserted in the Delaunay triangulation. If the error is below the threshold, the candidate point is not inserted in the Delaunay triangulation. Instead, the nearest point is updated: first, the list of keypoints it originates from is complemented with the two keypoints from which the candidate point was generated, then its position is recomputed using its updated keypoint list, and the Delaunay triangulation is modified accordingly, if needed.

This step outputs a Delaunay triangulation, whose vertices store a keypoint tuple and the best-fit corresponding 3D position. Note that the size of keypoint tuples is related to the confidence of 3D points, since false matches are unlikely to aggregate into large tuples.

3.3 Surface extraction

The final step in our method consists in labeling each tetrahedron of the Delaunay triangulation as inside or outside of the scene. The output triangular mesh is then obtained by taking the triangular facets between adajacent tetrahedra having

different labels. This constrains the reconstructed surface to be included in the Delaunay triangulation. This is not a limitation, however, as soon as the point cloud is sufficiently dense, as discussed in Section 2.2.

A globally optimal label assignment is efficiently found using graph cuts. To this end, we consider the dual graph to the Delaunay triangulation, in other words, the graph whose vertices correspond to Delaunay tetrahedra, and whose edges correspond to the triangular facets between adajacent tetrahedra. Actually, this coincides with the vertices and edges of the Voronoi diagram of the point set. In addition, there are links between each vertex of the graph (*i.e.*each Delaunay tetrahedron) and the sink and the source.

In the sequel, we note $\mathcal{S}$ the surface to be reconstructed. As discussed above, $\mathcal{S}$ is a union of Delaunay triangles. We wish to minimize an energy functional composed of three terms, one dealing with visibility, one dealing with photo-consistency and one dealing with surface smoothness:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\text{vis}}(\mathcal{S}) + \lambda_{\text{photo}}\,\mathrm{E}_{\text{photo}}(\mathcal{S}) + \lambda_{\text{area}}\,\mathrm{E}_{\text{area}}(\mathcal{S})$$

where λ_{photo} and λ_{area} are positive weights. In the rest of this section, we give the exact definition of each energy term and we describe how it can be implemented in the graph cuts framework.

3.3.1 Surface visibility

Each vertex in the triangulation has some visibility information: the keypoint tuple from which it was reconstructed (this tuple can contain as little as two keypoints or as many tuples as the total number of input views if the point was the result of multiple merges).

This information is decisive to design the $\mathrm{E}_{\text{vis}}(\mathcal{S})$ term: if some vertex belongs to the final surface then it should be visible in the views it comes from. Consequently, all the tetrahedra intersected by the ray emanating from the vertex to the camera center of one of these views should be labelled as outside (and the tetrahedron behind the vertex should be labelled as inside).

The following term: $\mathrm{E}_{\text{vis}}(\mathcal{S}) = \lambda_{\text{vis}}\,\#\{\text{ray conflicts}\}$, where a ray from a vertex to a camera center is in conflict if it intersects a tetrahedron labelled as inside, naturally comes to mind. Unfortunately, such energy term is not suitable for graph-cut optimization, as it would require complex interactions between more than two nodes in the graph [15, 23].

Instead, the number of intersections of the ray with the oriented surface is used (only ray crossings of a triangle from the inside to the outside are penalized). Moreover the surface should go through the vertex originating the ray and the last tetrahedron traversed by the ray should be labelled as outside. The construction of the corresponding visibility terms for one ray is summarized in Figure 3. Note that

the term $V_{p_2 q_2}$ cannot be translated to weights in the graph because the tetrahedra whose nodes are p and q do not share a triangle and the nodes p and q are thus not linked in the graph. Fortunately this term only amounts to a link to the sink with weight $w_{q_2 t} = \lambda_{\text{in}}$. The positive weights λ_{in}, λ_{out} and λ_∞ possibly take into account the confidence in the reconstructed vertex yielding the ray.

The global visibility term sums all the contributions of the rays cast by the vertices of the triangulations (the corresponding weights of the edges of the graph are accumulated the same way).

Note that the edges for the origin and for the end of the ray (with weights λ_{in} and λ_∞ respectively) can straightforwardly be adjusted (thanks to the Delaunay triangulation) to allow the reconstruction of, for instance, the walls of an indoor scene: not only finite tetrahedra can be used as nodes in the graph but also infinite tetrahedra (which have three vertices on the convex hull of the 3D point cloud and share an infinite vertex)...

3.3.2 Surface photo-consistency

The photo-consistency term $\mathrm{E}_{\text{photo}}(\mathcal{S})$ of our energy measures how well the given surface $\mathcal{S}$ matches the different input images in which it is seen. It is defined as the sum over the whole surface of some photo-consistency measure $\rho \geq 0$ (in our case, every triangle of the surface has a uniform photo-consistency):

$$\mathrm{E}_{\text{photo}}(\mathcal{S}) = \int_{\mathcal{S}} \rho \, \mathrm{d}S = \sum_{T \in \mathcal{S}} \rho(T) \, \mathcal{A}(T)$$

The photo-consistency of each triangle is computed in all the views from which its three vertices were reconstructed. Furthermore, as a triangle of the surface $\mathcal{S}$ lies by definition on the interface between the inside and the outside of the reconstructed object(s), its orientation needs to be taken into account: an "oriented photo-consistency" is used, which means that the two possible orientations of a given triangle get different photo-consistencies, each computed only in the subset of the considered views compatible with the given orientation of the triangle.

This maps quite easily onto the graph cuts framework: for each directed pair of tetrahedra (represented by nodes p and q in the graph) which shares a triangle T with normal $\vec{n}$ (pointing from tetrahedron p to tetrahedron q), an edge $p \rightarrow q$ is added with a weight $w_{pq} = \rho_{\{\Pi_i | \vec{d_i}.\vec{n} > 0\}}(T)$, where $\vec{d_i}$ is the direction from the center of the triangle to the center of the i-th camera Π_i.

3.3.3 Surface smoothness

Surface smoothness is encouraged by minimizing the area of the surface. Hence it is the simplest term of our energy:

$$\mathrm{E}_{\mathrm{area}}(\mathcal{S}) = \mathcal{A}(\mathcal{S}) = \int_{\mathcal{S}} \mathrm{d}S = \sum_{T \in \mathcal{S}} \mathcal{A}(T)$$

This is also trivially minimized in the graph cuts framework: for each pair of tetrahedra (sharing a triangle T) represented by nodes p and q in our graph, an edge $p \rightarrow q$ is added with a weight $w_{pq} = \mathcal{A}(T)$ and, similarly, an opposite edge $q \rightarrow p$ with the same weight $w_{qp} = w_{pq}$ is also added.

4 Experimental Results

4.1 Implementation aspects

In order to boostrap our algorithm, keypoints are extracted from the datasets images with the help of the freely available demo software of the SIFT keypoint detector[3].

The Delaunay triangulation is computed using the Computational Geometry Algorithms Library (CGAL)[4] [5]. CGAL defines all the needed geometric primitives and provides an excellent algorithm to compute the Delaunay triangulation in 3D: it is robust to degenerate configurations and floating-point error, thanks to the use of exact geometric predicates, while being able to process millions of points per minute on a standard workstation. It provides all the elementary operations needed in our algorithm: vertex insertion, vertex move, nearest vertex query and various traversals of the triangulation.

The photo-consistency is evaluated with a software rasterizer with sweeps the projection of each triangle of the Delaunay triangulation in the chosen views and computes the mean color variance of the pixels in this triangle.

Finally we compute the minimum s-t-cut of the graph we designed using the software[5] described in [7] which is in fact better suited for regular grid-based graphs more commonly found in computer vision.

Currently our implementation leaves room for improvement in term of computational speed: leaving aside the time required to extract the keypoints, it can take (on an Intel®Core™2 Duo 2.13 GHz PC) as little as a minute and a half to reconstruct a scene from a ~ 50 images dataset to a few dozens minutes from a

[3] `http://www.cs.ubc.ca/~lowe/keypoints/`

[4] `http://www.cgal.org/`

[5] `http://www.adastral.ucl.ac.uk/~vladkolm/software.html`

~ 300 images dataset depending on the number of input keypoints to match. Fortunately our method is quite versatile: we can use any type of features as input and we could resort to much faster features detector such as SURF [3]. The matching of keypoints is presently done by brute force so most of the above computational time is actually spent on feature matching alone: this could be improved by using a more adapted nearest neighbor search in high-dimension spaces [4]. Finally the software rasterizer used for the photo-consistency computation could obviously take great advantage of modern graphics hardware.

4.2 Temple

The first experiment (shown in Figure 4) uses the 312 views temple dataset from the review of Seitz *et al.* [32]. It shows that our approach is quite flexible and while able to reconstruct large-scale scenes, it can still cope with more traditional multi-view stereo without using any of the usual clues that most high-precision algorithms would require. Also recall that in our case the final shape of the object(s) depends on the 3D point cloud reconstructed from matched features, so regions without many matched keypoints are reconstructed as large triangles whereas densily sampled regions are more detailed.

4.3 Toys

The data for the second experiment (shown in Figures 5 and 6) was acquired with a consumer-grade handheld DV camcorder shooting soft toys laid on a table; one frame out of ten was extracted from the video sequence resulting in a 237 views dataset and calibration was done with a tracking software. The imaging conditions were absolutely not controlled, most of the images show large specular highlights on the tablecloth. No additional stabilizer was used and besides motion blur, many important color sampling and aliasing artefacts due to video compression requirements are clearly noticeable. Despite this impressively hard dataset, our algorithm was able to reconstruct the table and the soft toys showing its robustness and its ability to cope with a large-scale cluttered scene without any additional information about its extent. Note that some small details compared the global scale of the scene are still recovered (the antennas, the ears or the tail of some of the soft toys, for instance) but areas that lack matchable features are less accurately reconstructed.

5 Conclusion and Future Work

We have presented a new multi-view reconstruction method adapted to large-scale cluttered scenes under uncontrolled imaging conditions. First a quasi-dense 3D point cloud of the scene is generated by matching keypoints across different views. An adaptive tetrahedral decomposition of the space is then built by means of a Delaunay triangulation of the 3D point set. Finally the scene is reconstructed by labeling the tetrahedra as empty or occupied using an assignement globally optimal as to photo-consistency of the output mesh and compatibility with the visibility of the matched keypoints. This new approach is free from numerous restrictions of previous reconstruction algorithms: it does not require any knowledge of the extent of the scene, it can deal with large-scale scenes at a reasonable computational cost, it exploits visibility information from keypoints to guide the position of the surface in a robust way, lastly, it can handle closed and open scenes.

We have demonstrated our method on real data: a classical dataset acquired in a controlled setup and a new real-world data set showing the efficiency of our method in handling difficult imaging conditions. The experimental results shown are quite promising and only give an insight into the potential of our approach. We are eager to evaluate it on other challenging data sets. We also expect to greatly improve the computation time of our implementation by switching to faster kinds of features, by computing the photo-consistency on graphics hardware and by matching features using an adapted high-dimensional nearest-neigbor search. Our method could eventually be incorporated into a full reconstruction system in which the feature extraction and matching step would be shared between calibration and reconstruction.

References

[1] Nina Amenta and Marshall Bern. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.

[2] Dominique Attali, Jean-Daniel Boissonnat, and André Lieutier. Complexity of the Delaunay triangulation of points on surfaces: the smooth case. In *Annual Symposium on Computational Geometry*, pages 201–210, 2003.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*, 2006.

[4] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

[5] Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, and Mariette Yvinec. Triangulations in CGAL. In *Annual Symposium on Computational Geometry*, pages 11–18, 2000.

[6] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*, chapter Voronoi diagrams: Euclidian metric, Delaunay complexes, pages 435–443. Cambridge University Press, 1998.

[7] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[8] Yuri Boykov and Victor Lempitsky. From photohulls to photoflux optimization. In *British Machine Vision Conference*, volume 3, pages 1149–1158, 2006.

[9] Adrian Broadhurst, Tom W. Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *IEEE International Conference on Computer Vision*, volume 1, pages 388–393, 2001.

[10] Brian Curless and Marc Levoy. A volumetric approach for building complex models from range images. In *ACM SIGGRAPH*, pages 303–312, 1996.

[11] Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In *European Conference on Computer Vision*, volume 3, pages 238–251, 2004.

[12] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[13] Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.

[14] Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in Networks*. 1962.

[15] Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[16] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. In *European Conference on Computer Vision*, volume 1, pages 564–577, 2006.

[17] Pau Gargallo and Peter Sturm. Bayesian 3D modeling from images using multiple depth maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 885–891, 2005.

[18] Michael Goesele, Briand Curless, and Steven M. Seitz. Multi-view stereo revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2402–2409, 2006.

[19] D. M. Greig and B. T. Porteous A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B, Methodological*, 51(2):271–279, 1989.

[20] Alexander Hornung and Leif Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 503–510, 2006.

[21] Hailin Jin, Stefano Soatto, and Anthony J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *The International Journal of Computer Vision*, 63(3):175–189, 2005.

[22] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, volume 3, pages 82–96, 2002.

[23] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[24] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *The International Journal of Computer Vision*, 38(3):199–218, 2000.

[25] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[26] Victor Lempitsky, Yuri Boykov, and Denis Ivanov. Oriented visibility for multiview reconstruction. In *European Conference on Computer Vision*, volume 3, pages 225–237, 2006.

[27] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005.

[28] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

[29] David G. Lowe. Distinctive image features from scale-invariant keypoints. *The International Journal of Computer Vision*, 60(2):91–110, 2004.

[30] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[31] Jean-Philippe Pons, Renaud Keriven, and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *The International Journal of Computer Vision*, 72(2):179–193, 2007.

[32] Steven Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Rick Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–526, 2006.

[33] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *The International Journal of Computer Vision*, 35(2):151–173, 1999.

[34] Sudipta Sinha and Marc Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *IEEE International Conference on Computer Vision*, volume 1, pages 349–356, 2005.

[35] Jonathan Starck, Gregor Miller, and Adrian Hilton. Volumetric stereo with silhouette and feature constraints. *British Machine Vision Conference*, 3:1189–1198, 2006.

[36] Christoph Strecha, Rik Fransens, and Luc Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 552–559, 2004.

[37] Christoph Strecha, Rik Fransens, and Luc Van Gool. Combined depth and outlier estimation in multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2394–2401, 2006.

[38] Christoph Strecha, Tinne Tuytelaars, and Luc Van Gool. Dense matching of multiple wide-baseline views. In *IEEE International Conference on Computer Vision*, volume 2, pages 1194–1201, 2003.

[39] Son Tran and Larry Davis. 3D surface reconstruction using graph cuts with surface constraints. In *European Conference on Computer Vision*, volume 2, pages 219–231, 2006.

[40] Adrien Treuille, Aaron Hertzmann, and Steven M. Seitz. Example-based stereo with general BRDFs. In *European Conference on Computer Vision*, volume 2, pages 457–469, 2004.

[41] George Vogiatzis, Philip H. S. Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, 2005.

[42] Ruigang Yang, Marc Pollefeys, and Greg Welch. Dealing with textureless regions and specular highlights: A progressive space carving scheme using a novel photo-consistency measure. In *IEEE International Conference on Computer Vision*, volume 1, pages 576–584, 2003.

[43] Tianli Yu, Narendra Ahuja, and Wei-Chao Chen. SDG cut: 3D reconstruction of non-lambertian objects using graph cuts on surface distance grid. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2269–2276, 2006.

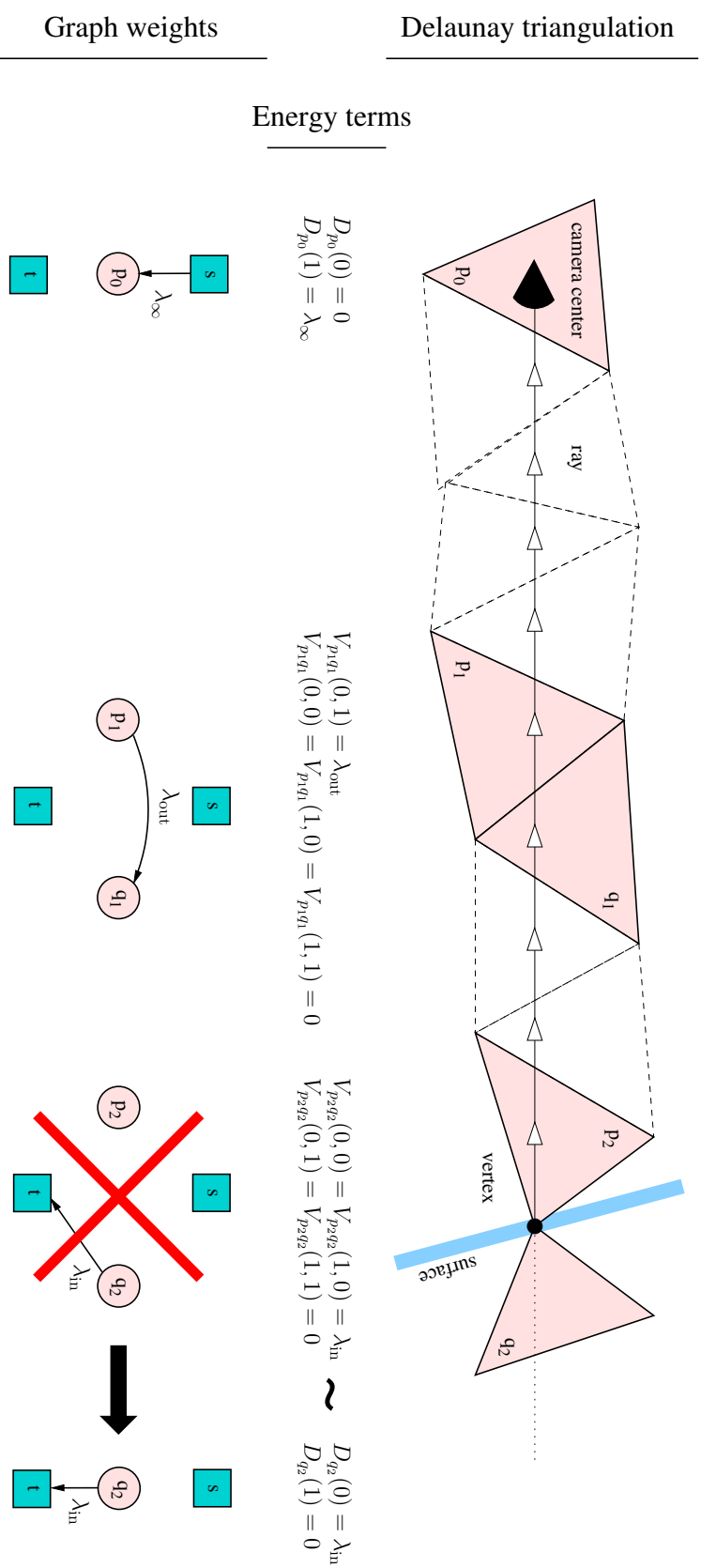


Figure 3: A ray emanating from a vertex to a camera center and the corresponding visibility-related energy terms and edge weights of the crossed tetrahedra (the label 0 means s / "outside" and the label 1 means t / "inside")

Figure 4: Temple dataset sample images and results

Figure 5: Toys dataset sample images

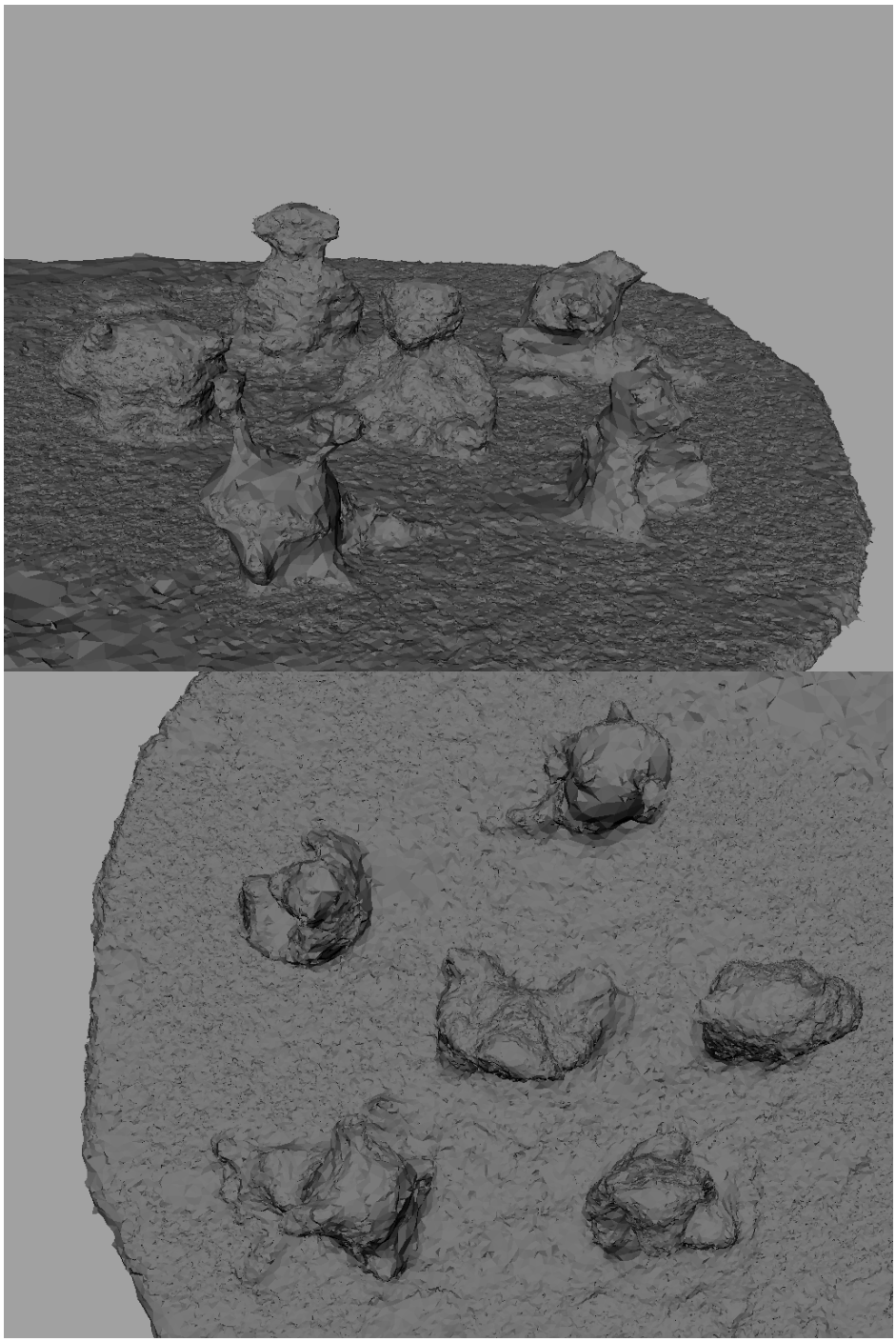


Figure 6: Toys dataset results