Applying Two-dimensional Delaunay Triangulation to Stereo Data Interpolation

E. Bruzzone[1] *M. Cazzanti*[2] *L. De Floriani*[2] *F. Mangili*[1]

[1] Elsag Bailey spa, Research & Development, I-16154 Genova (Italy)
[2] Dipartimento di Matematica - Università di Genova, I-16132 Genova (Italy)

Abstract. Interpolation of $3D$ segments obtained through a trinocular stereo process is achieved by using a $2D$ Delaunay triangulation on the image plane of one of the vision system cameras. The resulting two-dimensional triangulation is backprojected into the $3D$ space, generating a surface description in terms of triangular faces. The use of a constrained Delaunay triangulation in the image plane guarantees the presence of the $3D$ segments as edges of the surface representation.

1 Introduction

Delaunay triangulation has turned out to be a very powerful tool in many application fields, including finite element analysis, motion planning, digital terrain modeling and surface reconstruction in computer tomography [LR1, Ch1, Bo1, DP1]. Such representation has several important properties: it is invariant through rigid transformations, it adapts to the data distribution, it is easy to update because of the local effect of inserting new points or segments.

In classical computer vision problems, like scene reconstruction and autonomous navigation, Delaunay triangulation has been often adopted for both $2D$ and $3D$ data. In particular, its discontinuity-preserving nature makes it especially suitable to interpolate passive stereo data, which usually correspond to scene discontinuities. The use of $3D$ Delaunay triangulation for interpolation of data obtained by a stereo process was first proposed in [Bo1]. A coherent and comprehensive presentation of this approach can be found in [FL1], where the authors suggest a modification to standard Delaunay triangulation to include stereo segments as part of the triangulation, based on the addition of extra points.

A new approach to $3D$ surface reconstruction which starts from stereo data and makes use of a two-dimensional Delaunay triangulation including the projections of the segments as part of the triangulation, has been proposed in [BG1]. The basic idea is to interpolate the image segments which form the input for the stereo reconstruction process. The computed $2D$ mesh is then backprojected into the $3D$ space using the corresponding reconstructed stereo data. The result of the whole process is a triangular-faced piecewise linear surface, in which the stereo segments are somehow preserved. Interesting features of this approach are its fairly low computational cost, due to the fact that most of the processing is done in $2D$, and its robustness toward calibration and stereo reconstruction errors. A drawback of this approach is in the splitting of the segments in the image plane, which requires the computation of the $3D$ coordinates of the introduced points and produces many small triangles in special segment configurations.

In this paper, we present a further development of that work by proposing an approach to $3D$ surface reconstruction from stereo data based on the computation of constrained Delaunay triangulation in the image plane which avoids the segment splitting and therefore the computation of the $3D$ position of the added points [Ch1, LL1, DP1].

2 Three-Dimensional Surface Reconstruction Strategy

The surface reconstruction process consists of three phases: stereo segment reconstruction, constrained Delaunay triangulation in the image plane, and backprojection of the two-dimensional tessellation.

The edge segment-based stereo process developed under the Esprit Project P940 [AL1, Mu1] has been adopted. Three images are acquired from slightly different points of view. On each image a low-level processing made of edge detection, edge linking and polygonal approximation is performed, resulting in a set of $2D$ segments corresponding to relevant scene features. One of the three images is selected as *reference* image. For each segment of the reference image, possible matches, i.e., segments corresponding to the same feature in the other two images, are selected, making use of the epipolar constraint. Then, for each triple of matched segments the $3D$ segment is reconstructed, on the basis of perspective projection.

The triangulation is computed on the $2D$ segments of the reference image plane selected by the stereo process. Note that such segments are perspective projection of real observed features, and therefore they reflect the visibility properties of the world features from which they have been originated. As the low-level phases of edge linking and polygonal approximation guarantee that the segments are disjoint, each triangle is bounded by only one stereo segment. Moreover, as the image segments are directly the output of the stereo matching, the triangulation can be computed independently of the stereo reconstruction, avoiding the errors which may occur in the reconstruction phase.

The $2D$ mesh is then backprojected into the $3D$ space using the corresponding $3D$ segments endpoints evaluated during the stereo phase. The result of the whole process is a triangular-faced piecewise linear surface, in which the stereo segments are somehow preserved. For each triangular face of the surface, the normal unit vector is computed, achieving a space-variant *needle map* representation of the observed scene. The geometric structure resulting from the backprojection can be defined by a function $\rho = \rho(\vartheta, \phi)$ in a system of spherical coordinates centered in the pin-hole of the camera. Therefore, possible intersections among the triangular faces of the $3D$ surface can be caused only by errors occurred in the stereo process.

3 Computing Constrained Delaunay Triangulation

The two-dimensional Delaunay triangulation of a set $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ of points in the plane is the straight-line dual of the Voronoi diagram [PS1]. The *Voronoi diagram* of $\mathcal{P}$ is a collection $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$ of convex regions, called *Voronoi regions*, such that V_i is the locus of the points of E^2 closer to P_i than to any other point in $\mathcal{P}$. Given a set $\mathcal{P}$ of points in the plane and a set $\mathcal{S}$ of non-intersecting straight-line segments whose endpoints are contained in $\mathcal{P}$, the pair $G = (\mathcal{P}, \mathcal{S})$ defines a planar straight-line graph, called the *constraint graph.* A triangulation $\mathcal{T}$ of $\mathcal{P}$ whose edge set contains $\mathcal{S}$ is called a *constrained triangulation* of $\mathcal{P}$ with respect to $\mathcal{S}$. A *Constrained Delaunay Triangulation* (CDT) $\mathcal{T}$ of a set of points $\mathcal{P}$ with respect to a set $\mathcal{S}$ of line segments is a constrained triangulation of $\mathcal{P}$ in which the circumcircle of each triangle t of $\mathcal{T}$ does not contain (in its interior) any other vertex P_i of $\mathcal{P}$ which can be joined to each vertex of t by a line segment not intersecting any constraint segment (see Figure 1).

Static algorithms for computing a CDT appeared recently in the computational geometry literature [LL1, Ch1]. The algorithm we use, proposed in [DP1], is instead based on incremental refinements of a Delaunay triangulation. It starts from an initial Delaunay triangulation of a specified subset of the input data, and then modifies the triangulation

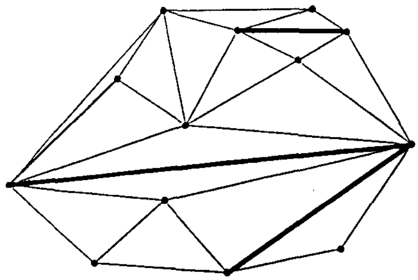


Figure 1: An example of constrained Delaunay triangulation. Thick lines represent constraint segments.

by inserting the points of $\mathcal{P}$ and the segments in $\mathcal{S}$ one at a time. Thus, the two major computational steps of the algorithm are (i) CDT modification when inserting a point P, (ii) CDT modification when inserting a segment l.

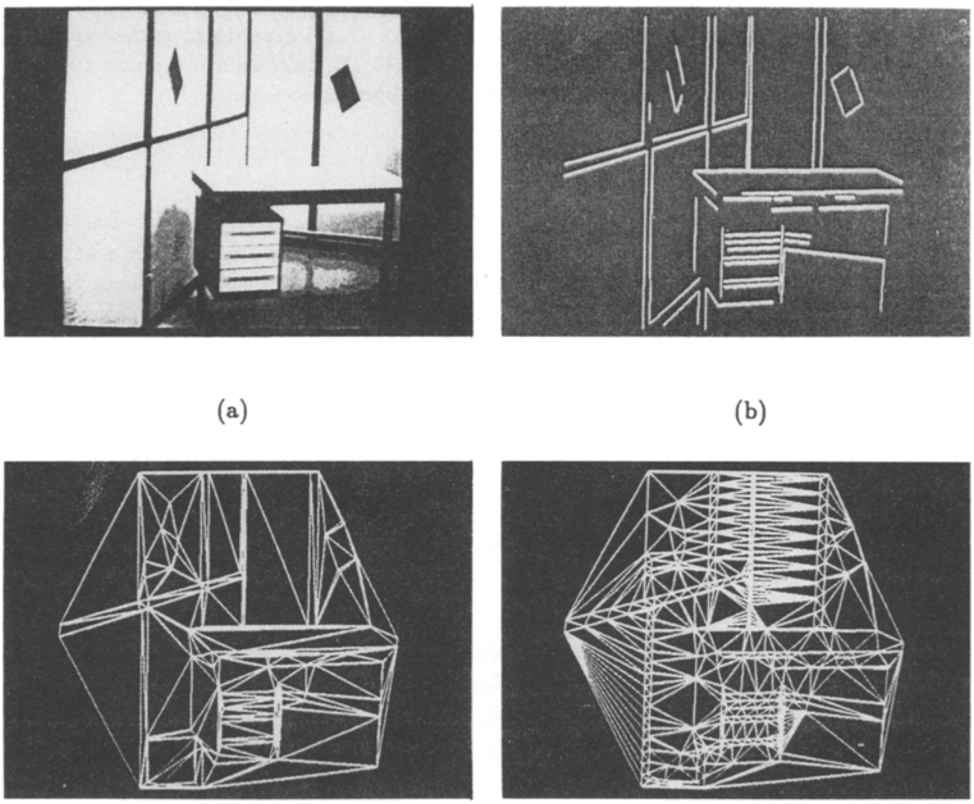
Step 1 is performed by extending a standard method for adding a point to a Delaunay triangulation to the constrained case [Wa1]. When a new point P is inserted, the triangles whose circumcircle contains P are deleted and the resulting star-shaped polygon is triangulated by connecting the vertices of such a polygon to P. The worst-case complexity of this step is $O(n)$. Thus, inserting all n data points leads to an $O(n^2)$ worst-case complexity, which reduces to $O(n \log n)$ if randomized algorithms are used [GK1].

Step 2 is performed by intersecting the new segment l with the existing triangulation and retriangulating the region of the plane defined by the union of the triangles intersected by l. The edges bounding the region of $\mathcal{T}$ intersected by l, called *influence region*, form a simple polygon Q_l, called *influence polygon*, of which l is a diagonal. l splits Q_l in two simple polygons π_1 and π_2, which are triangulated by recursively splitting them into three subpolygons. The resulting triangulation of π_1 and π_2 is then locally optimized by an iterative application of the empty circle criterion for a CDT [DP1].

The time complexity of the influence region computation of a constraint segment l is linear in the number of triangles intersected by l. Both rebuilding the constrained Delaunay triangulation of a polygon and its optimization have a quadratic worst-case complexity in the number of vertices of the influence polygon. The worst-case complexity of the segment insertion algorithm is $O(mn^2)$, where m is the number of constraint segments ($m = 2n$ if the points of $\mathcal{P}$ are the endpoints of the segments of $\mathcal{S}$). By using an asymptotically optimal Delaunay triangulation algorithm for simple polygons [LL1], the worst-case complexity of the algorithm could be reduced to $O(mn \log n)$, by losing the implementation simplicity.

An alternative approach to include sets of segments to a Delaunay triangulation, consists of splitting the segments into subsegments (by adding additional vertices), so that the constrained Delaunay triangulation of all subsegments is the same as the Delaunay triangulation of the augmented vertex set. In [FL1] a preprocessing step is used to split the segments according to their minimum distance.

A comparison between the CDT algorithm and the segment-splitting algorithm described in [FL1] has been done. Experimental results show that the average number of inserted points triples the number of original points (segments endpoints). The number of points (and triangles) increases dramatically when close parallel segments occur in the input data. Figure 2 shows the results of CDT and segment-splitting algorithms on a real indoor scene.



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

Figure 2: Reference image of a trinocular stereo system (a) and matched segments (b). *CDT* (c) and segment-splitting triangulations (d).

4 Experimental Results on Scene Reconstruction

The complete process of scene reconstruction has been tested on a set of real scenes. Assuming as reference applications both scene surface characterization and free space detection for autonomous navigation tasks, indoors images (i.e., office and laboratory images) have been acquired.

The DMA machine, developed under the Esprit Project P940, has been used to get both the $3D$ reconstructed segments and the corresponding $2D$ segments of the reference image. First an unconstrained Delaunay triangulation is built on the segments endpoints. Then, the resulting triangulation is updated, by adding the input segments as Delaunay edges.

The surface obtained backprojecting the CDT into $3D$ is made of a minimum number of triangles (for instance, two parallel segments define only two triangles). Besides, very elongated triangles, which may occur in the image-plane CDT, often correspond to more equiangular triangles in $3D$, due to the perspective projection under which the scene has been seen.

Experimental results have shown that the running time of the whole surface reconstruction process reduces of about the 50% using the *CDT* algorithm, rather than the segment-splitting one. Such a reduction is due to both the triangulation phase (without the splitting of the constraint segments) and the backprojection phase.

5 Concluding Remarks

The proposed scene reconstruction process starting from stereo segments is based on a two-dimensional Constrained Delaunay Triangulation done in the image plane and results in a triangular-faced piecewise linear description of scene surfaces. With respect to what presented in [BG1], the main novelty is in the use of a powerful algorithm which constrains the triangulation to the input segments, avoiding the insertion of extra points. Some experimental tests on real data have confirmed the foreseen advantages of this new approach in terms of both computational efficiency and improvement of the resulting surface description.

As the bottleneck of the whole strategy is in the computation of the *CDT*, a parallel implementation of this phase on the Elsag Bailey multiprocessor machine EMMA2 has been completed.

References

[AL1] Ayache N., Lustman F.: Fast an Reliable Passive Trinocular Stereovision, Proceedings 1^{st} International Conference on Computer Vision, London (1987).

[Bo1] Boissonnat J.D.: Geometric Structures for Three-Dimensional Shape Representation, ACM Transaction on Graphics, **3**, 4 (1984).

[BG1] Bruzzone E., Garibotto G., Mangili F.: Three-Dimensional Surface Reconstruction using Delaunay Triangulation in the Image Plane, Proceedings International Workshop on Visual Form, Capri (1991).

[Ch1] Chew, L.P.: Constrained Delaunay Triangulation, Proceedings 3^{rd} Symposium on Computational Geometry, Waterloo (1987).

[DP1] De Floriani L., Puppo E.: Constrained Delaunay Triangulation for Multiresolution Surface Description, Proceedings 9^{th} International Conference on Pattern Recognition, Roma (1988).

[FL1] Faugeras O.D., Le Bras-Mehlman E., Boissonnat J.D.: Representing Stereo Data with the Delaunay Triangulation, Artificial Intelligence, **44** (1990).

[GK1] Guibas L.Y., Knuth D.K., Sharir M.: Randomized Incremental Construction of Delaunay Triangulations and Voronoi Diagrams, Proceedings ICALP (1990).

[LL1] Lee D.T., Lin, A.K.: Generalized Delaunay Triangulation for Planar Graphs, Discrete Computational Geometry, **1** (1986).

[LR1] Lewis B.A., Robinson, J.S.: Triangulation of Planar Regions with Applications, The Computer Journal, **21**, 4 (1979).

[Mu1] Musso G.: Depth and Motion Analysis: the Esprit Project P940, Proceedings 6^{th} Annual ESPRIT Conference, Brussels (1989).

[PS1] Preparata F., Shamos M.I.: Computation Geometry: an Introduction, Springer-Verlag, New York (1985).

[Wa1] Watson D.F.: Computing the n-dimensional Delaunay Triangulation with Application to Voronoi Polytopes, The Computer Journal, **24** (1981).