MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo No. 1185 **September 1989**

An Alternative to Using the 3-D Delaunay Tessellation for Representing Freespace

David J. Braunegg

Abstract: Representing the world in terms of visible surfaces and the freespace existing between these surfaces and the viewer is an important problem in robotics. For example, such a representation can be used to plan mobile robot navigation routes or manipulator paths for pick-and-place operations. Recently, researchers have proposed using the 3-D Delaunay Tessellation for representing 3-D stereo vision data and the freespace determined therefrom. We discuss problems with using the 3-D Delaunay Tessellation as the basis of the representation and propose an alternative representation that we are currently investigating. This new representation is appropriate for planning mobile robot navigation and promises to be robust when using stereo data that has errors and uncertainty.

Acknowledgments. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014–85–K–0124 and under Army contract number DACA76–85–C–0010.

1 Introduction

Representing the world in terms of visible surfaces and the freespace existing between these surfaces and the viewer is an important problem in robotics. For example, such a representation can be used to plan mobile robot navigation routes or manipulator paths for pick-and-place operations. We are mainly concerned with representing freespace to aid mobile robot navigation and therefore discuss in this paper some ongoing research on the problem.

Recently, researchers have proposed using the 3-D Delaunay Tessellation for representing 3-D stereo vision data and the freespace determined therefrom [Boissonnat, et al. 1988a] [Boissonnat, et al. 1988b] [Le Bras-Mehlman, et al.]. We discuss problems with using the 3-D Delaunay Tessellation as the basis of the representation if the data is supplied sequentially and the representation built up incrementally. We propose an alternative representation based on the 2-D Delaunay Tessellation and 2-D uncertainty grids [Matthies and Elfes] that we are currently investigating. This new scheme promises to avoid the problems identified in the 3-D Delaunay-based scheme.

2 The 3-D Delaunay Method

[Boissonnat] has used the 3-D Delaunay tessellation as a volume-based representation for three-dimensional objects. This work has been extended by using the tetrahedra resulting from the 3-D Delaunay tessellation to represent freespace in a scene processed by a stereo vision algorithm [Boissonnat, et al. 1988a] [Boissonnat, et al. 1988b] [Le Bras-Mehlman, et al.]. In their approach, Boissonnat *et al.* first run a stereo algorithm on a pair of images to obtain a set of 3-D line segments. The space viewed by the cameras is then tessellated into tetrahedra with a 3-D Delaunay algorithm, ensuring that the stereo segments are included as edges of the tetrahedra. Freespace tetrahedra are identified by considering the line (actually planar patch) of sight between the viewer and the stereo segments. Any tetrahedron intersected by one of these lines of sight is marked as being freespace. (Such a tetrahedron cannot represent occupied space since a stereo segment is visible through it.) All remaining tetrahedra are marked as being non-freespace and are either occupied or are part of freespace but have no stereo segments visible through them to allow the tetrahedra to be marked as freespace.

3 Proposed Method

We propose a different method for determining and representing freespace from stereo data. First, we triangulate the matched stereo segments *in the image plane* without regard to their depth, ensuring that each stereo segment appears in the triangulation. (The 2-D Delaunay triangulation is used to minimize the occurrence of thin, elongated triangles.) Triangular planar patches in three-dimensional space are then derived from the triangles in the image plane by interpolating the depth between the edges and vertices of the triangles. These edges and vertices correspond to pieces of matched stereo segments. All space lying between the 3-D triangular patches and the viewer is then declared to be freespace. This freespace-labeling step implicitly assumes a straight-line surface approximation between matched stereo segments. Following the work of [Matthies and Shafer] [Matthies and Elfes] [Moravec and Elfes] in two dimensions, we divide space into cubes (say, 6 inches on a side) by imposing a 3-D grid on the space to be represented. The cubes that correspond to freespace can then be marked in this representation, with the unmarked cubes representing non-freespace.

4 Comparison with Perfect Data

In comparing the above methods for determining freespace, let us first consider the case where the output of the stereo algorithm is error-free, i.e., that stereo edges are localized with zero error, that no false matches are returned, and that every possible match was found. Assume that after a stereo pair from a view is processed and this information incorporated into the freespace representation, the cameras are moved (translation and rotation) and a second stereo pair is taken. Since the stereo output is error-free, the results of the second stereo run can be compared with the first and the translation and rotation can be determined exactly.

4.1 3-D Delaunay

Considering just the stereo results from a single image pair, the 3-D Delaunay has a problem. Assume that tetrahedron A lies behind edge a. Furthermore, let b be an edge that lies behind A and does not touch A. Figure 1 shows the projection of this situation into 2-D. Now, tetrahedron A must be labeled as freespace since edge b is visible through it. However, if it is labeled as freespace, then edge a is "floating" since it is not connected to any non-freespace tetrahedron. If we modify this labeling scheme by labeling each tetrahedron that lies directly behind an edge as non-freespace, then tetrahedron A will be labeled as non-freespace because of edge a, but will occlude edge b. Since the 3-D Delaunay tessellation does not use any information about the configuration of the stereo edges in the images when finding the 3-D tetrahedra, this freespace/non-freespace contradiction can potentially occur for any pair of edges. Also, since the 3-D Delaunay tessellation for a given set of edges is unique, we cannot retessellate to obtain a different set of tetrahedra that avoid this contradiction. Thus, we are left with an edge which has no non-freespace to justify its existence. To avoid this problem, [Boissonnat, et al. 1988a] suggest only marking as freespace those tetrahedra that have a number of lines of sight passing through them.

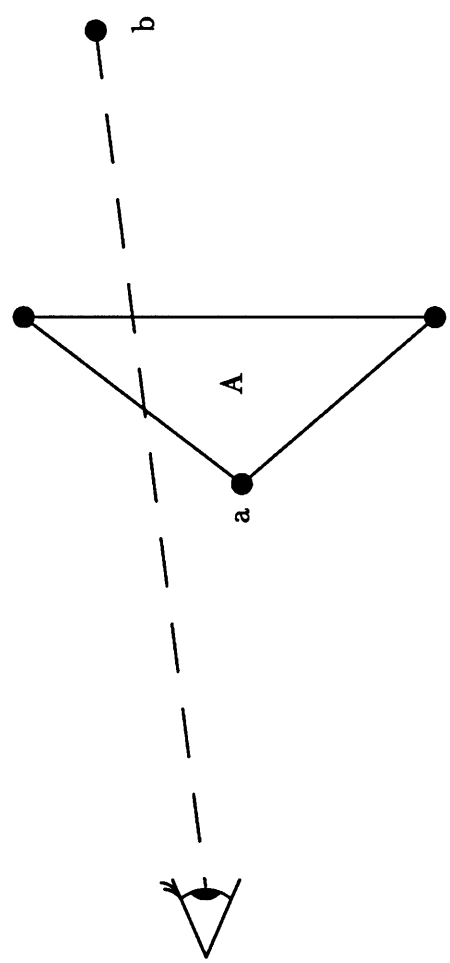

Figure 1. Tetrahedron A behind edge a and in front of edge b. (Projected into 2-D.)

Now consider the stereo edges obtained from the second pair of images. This stereo information is to be added to the representation already built from the first pair. After determining the correspondences between edges that are visible from both views, the new edges in the second view are added to the original set of edges and the tessellation is updated. Some of the original tetrahedra will be eliminated and others will appear in their places. To determine the freespace status of these new tetrahedra, the line of sight method is again employed. However, this time we must perform the line of sight test from *both* of the viewpoints used to obtain the edges. Furthermore, we must perform this test from each viewpoint only considering the edges that were visible from that viewpoint. Thus, we must label each edge with the viewpoint(s) from which it is visible to allow this selective processing. Furthermore, these labels must be kept until we no longer intend to update the freespace representation.

4.2 Proposed Method

Now consider the method we have proposed for determining and representing freespace when presented with perfect data. For the first viewpoint, we mark as freespace each cube that lies between the viewer and the 3-D triangles. The occlusion contradiction of the 3-D Delaunay method cannot occur since the relationships between the edges in the image were considered when the 2-D triangulation was performed. After obtaining the set of matched stereo edges from the second viewpoint, we again triangulate in the image plane. Triangles that are formed from edges common to both viewpoints will be the same. However, there will be some edges in the second pair of images that were not visible in the first pair. If an edge from the second viewpoint falls outside the field of view from the first viewpoint, then the freespace resulting from the triangles generated by that edge is marked as before. If a new edge from the second viewpoint falls within the field of view of the first viewpoint (Figure 2.a), then it will either fall within an area previously marked as freespace (edge a in Figure 2.a) or in an area that is non-freespace (edge b in Figure 2.a). If the edge falls in a freespace area, then the triangles associated with it bound a region of freespace. We simply adjust the cubes within this region to be marked as non-freespace (Figure 2.b). Similarly, if the edge falls in a non-freespace area, we mark the cubes within the region bounded by the edge's triangles as freespace (Figure 2.b).

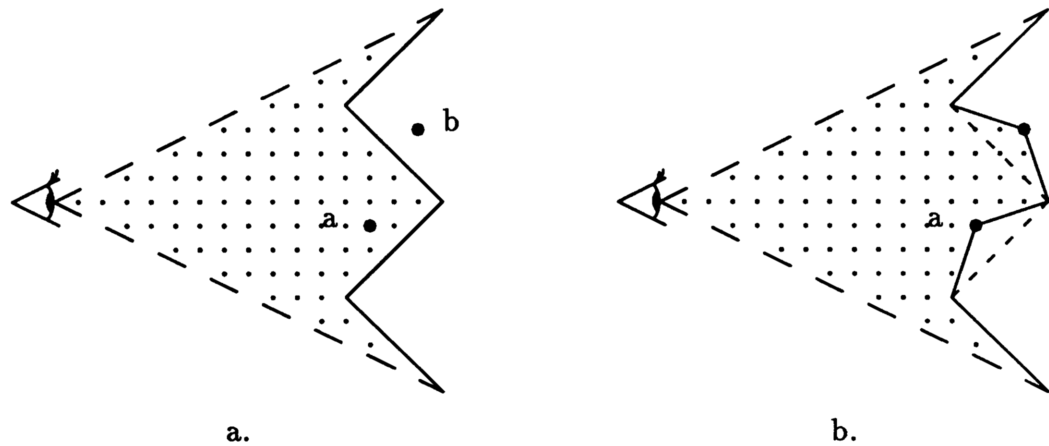

Figure 2. a. Edge *a* falls within an area previously marked as freespace (shaded area) while edge *b* falls within a non-freespace region. (Projected into 2-D.) b. The freespace/non-freespace regions due to edges *a* and *b* have been adjusted. (Projected into 2-D.)

5 Comparison with Missing and Extraneous Edges

Because no stereo vision algorithm (or any other sensing scheme) is perfect, we now consider the case where the stereo output may return extraneous edges or may not return some edges that were visible and could have been matched correctly. For edges that were matched correctly, we assume that the edges were localized without error.

5.1 3-D Delaunay

Since edges are never retracted from the 3-D Delaunay tessellation, an edge that is missing from a stereo pair in a sequence of views presents no problem since its occurrence in one of the views is enough to ensure its inclusion in the tessellation. However, since there is no mechanism for retracting an edge from the tessellation, the case of an extraneous edge is much more serious. An extraneous edge (and its associated tetrahedra) will be added to the tessellation when it is seen. Tetrahedra in the tessellation will be marked as freespace based on this edge, and since this edge will remain in the tessellation, it will continue to cause tetrahedra to be erroneously marked as freespace. A secondary problem is that space will be tessellated about such an extraneous edge and the tessellation in this area will never be retracted.

5.2 Proposed Method

A slight modification to our proposed scheme for representing freespace will allow us to handle the cases of missing and extraneous edges. Rather than simply marking a cube as freespace, we mark it with a value representing our *confidence* that it represents freespace. Each time we have a view that declares a cube to be freespace, we update its confidence with a running average that takes into account its previous marking. A cube that had previously been marked as freespace has its confidence level reinforced (raised) each time a different view marks it as freespace. Thus, the effect of any one view declaring a cube to be a part of freespace will be tempered with the decisions regarding that cube from previous views.

We still need a way to decrease the confidence that a cube is a part of freespace or, equivalently, to increase the confidence that it is a part of non-freespace. We cannot simply mark all cubes behind a triangle as being part of non-freespace since this area has infinite extent. All we can presume is that locally the space at a triangle is non-freespace. So, when we find a surface triangle, we label the cubes through which it passes as non-freespace, again with a confidence factor. If a freespace cube is labeled as freespace by a new stereo pair, its freespace confidence is reinforced. If it is instead labeled as non-freespace, then its freespace confidence is reduced. The obvious dual applies to non-freespace cubes. Also, based on the confidence levels, a cube can change from being labeled as a freespace cube to being a non-freespace cube, or vice-versa. [Moravec and Elfes] describe such an grid-update algorithm for two-dimensional maps derived from sonar data. [Stewart] describes a three-dimensional grid-based system to combine multisensor data for an underwater environment.

One problem remains. Suppose that a new edge is seen that was missed in previous stereo pairs and that the space around this edge has already been labeled as freespace. The cubes around the edge's triangles will now get marked as non-freespace and eventually (after enough views of this edge) their confidences will indicate that the cubes are non-freespace instead of freespace. What happens to the cubes that are behind the triangle and that have now been "fenced in" by the triangle and other surrounding surfaces? The answer is nothing. As long as the surfaces that are bounding these cubes are perceived, the confidences of these cubes cannot be updated one way or the other. This does not matter since, as they are surrounded by surfaces, they are implicitly non-freespace, i.e., there is no way for a mobile robot to get to them. If one of the real-world surfaces that bound this area is removed or one of the surfaces is no longer perceived (because it was previously matched in error), then the non-freespace confidence of the surface in the representation will be reduced and will eventually become a freespace confidence. Also, if one of the bounding surfaces is not perceived, then the freespace confidences associated with this bounded area will be averaged in with the current labeling of the bounded area as freespace.

[Matthies and Elfes] discuss a Bayesian estimation method for labeling the confidences of the occupancy grid squares that models the sensor uncertainty. We are experimenting with a simpler method which does not explicitly model the sensor uncertainty, but instead depends on a large number of data samples to develop the position estimates. For the implementation of this confidence idea, we suggest freespace confidences in the range $[-1, 0)$ and non-freespace confidences in the range $(0, 1]$. Zero represents areas where we have no information (or where the opposing confidences have exactly canceled each other). Each cube is associated with a confidence and also with a count of how many times this confidence has been updated. This count is used to weight the confidence resulting from the current view when combining it with the stored confidence. The current update rule for keeping a running average of the confidences is

$$\begin{aligned} \hat{x}_{k+1} &= \frac{1}{k+1}\sum_{i=1}^{k+1} z_i \\ &= \frac{1}{k+1}\sum_{i=1}^{k} z_i + \frac{1}{k+1}v_{k+1} \\ &= \frac{k}{k+1}\hat{x}_k + \frac{1}{k+1}v_{k+1} \\ &= \hat{x}_k + \frac{1}{k+1}(z_{k+1} - \hat{x}_k) \end{aligned} \tag{1}$$

where

$\hat{x}_{k+1}$ is the new cumulative confidence for the cube

$\hat{x}_k$ is the previous cumulative confidence for the cube

z_i is the confidence for the cube determined from view i

and

k is the update count.

(It should be obvious that if the current view makes no assignment of confidence to a cube, then neither its cumulative confidence not its update count is updated.) Other factors can be substituted for the $1/(k+1)$ in equation (1) to adjust the relative importance of the current freespace marking and the cumulative marking. Other update rules are under consideration.

Note that with the confidence updating scheme, we do not change the freespace/non-freespace designation of cubes based on where a new edge falls, as we did above in the section about the use of this method with perfect data. Since we are not guaranteed that a new edge is indeed a true edge, we do not want to change the freespace designation of a cube absolutely based on a single observation of an edge, but rather, we want to take into account previous observations when we detect a new edge.

6 Comparison with Uncertainty of Position

Finally, let us consider the case that we have when dealing with input from a stereo algorithm applied to real images. In addition to extraneous and missing edges, the locations of the edges are not known with absolute certainty, but instead the locations are known within some uncertainty bound. Thus, although a set of edges may be visible in two different views of a scene, no registration of the two views will cause these common edges to exactly coincide. We can derive an estimate of the rotation and translation between camera positions, but this will be a best-fit transformation under which most of the corresponding edges will not coincide.

6.1 3-D Delaunay

Since two different views of an edge will not coincide, each edge will be added separately to the tessellation. We will begin to add long, thin tetrahedra that will occur between these different instantiations of the same edge. Since edges are never removed in this (3-D Delaunay) scheme, the location of the freespace/non-freespace boundary will shift farther away from its true location with each additional view. Instead of refining the shape of freespace with additional stereo information, we will distort it. Also, with many perceived edges appearing near the location of a true edge, the freespace/non-freespace contradiction discussed in Section 4.1 will occur frequently.

6.2 Proposed Method

There are at least three solutions to the uncertainty problem in our proposed method. Testing will show which works well enough for building representations for navigational purposes.

The first solution is to proceed as described above for dealing with missing and extraneous edges. Over many views of a scene, the location of an edge will tend to average out to its true location. This implicitly depends on the law of large numbers, but we will probably be dealing with a small set of images of any particular scene. Still, we probably do not need to know the location of edges and surfaces very exactly. The transformation of viewpoint will be determined by a best-fit criterion for matching the edges between views (*à la* RAF [Grimson and Lozano-Pérez]).

The second solution is to attach a "radius of certainty" to each edge based on its distance from the viewer [Durrant-Whyte]. The farther away an edge is, the larger the radius due to the lower certainty of its true location. When averaging the non-freespace confidence values for a triangle with the cubes that it intersects, we spread the triangle out to intersect a section of cubes, with the width of the spread interpolated from the radii of uncertainty for the triangle edges. The non-freespace certainties inside the area defined by a radius of uncertainty can be decreased with distance from the locus of the non-freespace area, but this is probably more complex than the problem warrants. Edge correspondence and determination of the translation and rotation between views would take this radius and the certainty associated with it into account.

The third solution is like the second, but instead of using a radius of uncertainty which models the uncertainty in the location of a point as a sphere around that point, we use an ellipsoid of uncertainty. The ellipsoid reflects the fact that the uncertainty in depth is different from the angular uncertainty when considering the vector from the viewer to a point. [Matthies and Shafer] have modeled the uncertainty of the location of an edge with a 3-D Gaussian when updating the position of a mobile robot from successive views of a scene. They report improved robot localization when using this ellipsoidal model instead of a spherical model. For the scheme described above, however, using a constant-density ellipsoid instead of a 3-D Gaussian may be sufficient since we are not trying to build a highly accurate map of the world. Instead, we want a map of the area immediately around the robot to allow us to do simple obstacle avoidance and path planning in that area. If we will be in an area long enough to need a better freespace representation, then we will depend on the locations of the perceived world features to average over time to a good estimate of the actual locations.

7 Conclusions

The method of Boissonnat *et al.* for representing freespace for navigational planning using the 3-D Delaunay tessellation has been examined. Deficiencies have been identified regarding the marking of tetrahedra in the tessellation as freespace or non-freespace and regarding the difficulty in removing edges from the tessellation. No method readily presents itself for handling extraneous stereo features or features with positional uncertainty. These problems occur when the data is presented incrementally and the representation built up over time and when there are errors in the data. An alternative method for representing freespace has been proposed which overcomes the deficiencies in the 3-D Delaunay approach. This new method avoids the freespace labeling contradiction that the 3-D Delaunay approach encounters. It also deals effectively with stereo data that may be in error. Finally, it handles the practical problem of the uncertainty of position of matched stereo features.

The foregoing discussion assumes that one wants a three-dimensional model of the world for navigational planning. However, a two dimensional model may be sufficient if the only goal is to plan collision-free paths. (The ideas presented above extend to 2-D by projecting the stereo features to the ground plane and tessellating in that plane.) A three-dimensional model will be more important if this representation is to be used for tasks other than path planning, such as search or world location recognition.

8 References

Boissonnat, J. D., "Geometric Structures for Three-Dimensional Shape Representation," *ACM Transactions on Graphics*, **3** (4), October 1984, 266–286.

Boissonnat, J. D., O. D. Faugeras, and E. Le Bras-Mehlman, "Representing Stereo Data with the Delaunay Triangulation," INRIA, Technical Report 788, February 1988a.

Boissonnat, J. D., O. D. Faugeras, and E. Le Bras-Mehlman, "Representing Stereo Data with the Delaunay Triangulation," *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 1988b, 1798–1803.

Le Bras-Mehlman, E., M. Schmitt, O. D. Faugeras, and J. D. Boissonnat, "How the Delaunay Triangulation Can Be Used for Representing Stereo Data," *Second International Conference on Computer Vision*, Tampa, FL, December 1988, 54–63.

Durrant-Whyte, H. F., "Sensor Models and Multisensor Integration," *The International Journal of Robotics Research*, **7** (6), December 1988, 97–113.

Grimson, W. E. L. and T. Lozano-Pérez, "Model-Based Recognition and Localization From Sparse Range or Tactile Data," *The International Journal of Robotics Research*, **3** (3), Fall 1984, 3–35.

Matthies, L. and S. A. Shafer, "Error Modelling in Stereo Navigation," Computer Science Department, Carnegie-Mellon University, CMU-CS-86-140, 1986.

Matthies, L. and A. Elfes, "Integration of Sonar and Stereo Range Data Using a Grid-Based Representation," *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 1988, 727–733.

Moravec, H. P. and A. Elfes, "High Resolution Maps from Wide Angle Sonar," *IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985, 116–121.

Stewart, Jr., W. K., "Multisensor Modeling Underwater with Uncertain Information," MIT AI Lab, Technical Report 1143, 1988.