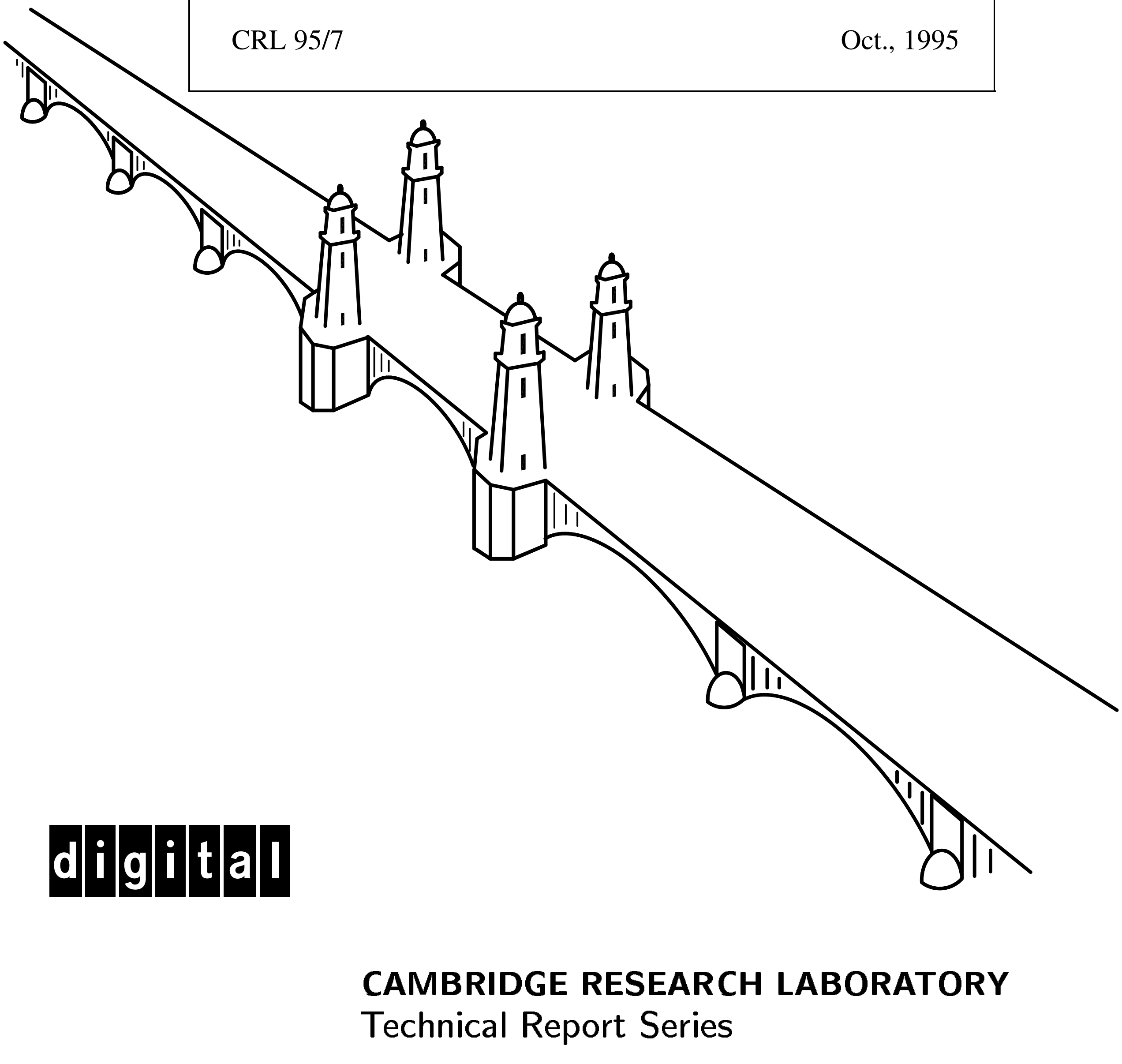
Extraction of Concise and Realistic 3-D Models from Real Data

Sing Bing Kang, Andrew Johnson, and Richard Szeliski

Digital Equipment Corporation
Cambridge Research Lab

CRL 95/7 Oct., 1995



digital

CAMBRIDGE RESEARCH LABORATORY
Technical Report Series

Cambridge Research Laboratory
One Kendall Square
Cambridge, Massachusetts 02139

Extraction of Concise and Realistic 3-D Models from Real Data

Sing Bing Kang, Andrew Johnson[1], and Richard Szeliski[2]

Digital Equipment Corporation

Cambridge Research Lab

CRL 95/7 Oct., 1995

Abstract

We have developed an algorithm for extracting concise surface models of a scene from real 3-D sensor data that can be used for virtual reality modeling of the world. Our algorithm produces an accurate description of a scene even in the presence of sensor noise, sparse data and incomplete scene descriptions. We have demonstrated order of magnitude reduction in the number of faces used to describe a scene using data from multibaseline omnidirectional stereo, structure from motion, and a light-stripe range finder. The algorithm takes as input a surface mesh describing the scene and a co-registered image. It outputs a surface mesh with fewer vertices and faces to which the appearance of the scene is mapped. Our simplification and noise reduction algorithm is based on a segmentation of the input surface mesh into surface patches using a least squares fitting of planes. Simplification is achieved by extracting, approximating, and triangulating the boundaries between surface patches.

Keywords: 3-D scene modeling, 3-D mesh simplication, planar fitting.

---

[1]The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213

[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399

Contents

List of Figures

List of Tables

1 Introduction

Modeling the world is necessary for many virtual reality applications, including tele-robotics [Johnson *et al.*, 1995], tele-presence [Earnshaw *et al.*, 1993], flight simulation, and modeling of famous buildings and scenes for archiving, education, and entertainment [Chen, 1995]. These applications immerse a viewer in a model of the world where the viewer is able to move around and at times interact with his surroundings. For the immersion to be complete and convincing, the models of the world must be realistic and rendered in real time.

To accurately and realistically model the world, data describing the 3-D shape and appearance (color and surface properties) of objects in the world must be gathered. This data is generally collected using vision sensors and algorithms that generate surface shape and appearance because the data can be collected non-intrusively and at any scale and resolution. Furthermore, vision sensors and algorithms generate physically correct descriptions of the world automatically, so that the need for traditional labor-intensive model-building using CAD packages is very much reduced. Unfortunately, there exist two fundamental problems with modeling the shape of the world using real data: the shape data is generally redundant because it is collected from multiple viewpoints using fixed resolution sensors, and the data is noisy because of imprecise sensors and imperfect shape recovery algorithms.

In realistic real world modeling, appearance is mapped onto the faces of a surface representation using texture mapping. Texture mapping, or appearance mapping, is a time-consuming rendering operation that takes time linear in the number of faces used to represent the scene. For believable viewing of 3-D models generated from real data, methods for reducing the number of faces while maintaining the shape must be developed so that rendering takes the least amount of time possible. These methods should also handle noise in the data, incomplete scene descriptions, and scene data of varying density.

To these ends we have developed an algorithm for simplification of scene shape that lends itself to appearance mapping. Our algorithm produces a concise and accurate description of a scene even in the presence of sensor noise, sparse data and incomplete scene descriptions. We demonstrate its feasibility using data from multibaseline omnidirectional stereo, structure from motion, and a light-stripe range finder. The algorithm takes as input a surface mesh describing the scene and a co-registered image. It outputs a surface mesh with fewer vertices and faces to which the appearance of the scene is mapped.

Our algorithm has four stages. First, the surface mesh is segmented into planar surface patches. Then the boundaries of the planar patches in the form of connected chains of points in the surface mesh are extracted and simplified. Next, the discretization of the interior of each planar patch is determined by triangulating the points on the boundary of the patch. Finally, the remaining vertices in the simplified surface mesh are projected onto the planes determined by the original segmentation of the surface mesh. In a final post processing stage on the surface mesh, the shape of the scene can be corrected to more accurately model the world through the application of world knowledge.

Most contributions to simplification of surface meshes have come from the areas of computer graphics and medical visualization. Schroeder *et al*. [Schroeder *et al.*, 1992] present an iterative technique based on local distance and angle criterion for reduction of triangular meshes. Turk [Turk, 1992] uses a curvature measure to reduce the number of vertices in surface meshes of curved surfaces. Hoppe *et al*. [Hoppe *et al.*, 1993] cast mesh simplification as an optimization problem where they seek to minimize an energy function that weighs conciseness of the model versus trueness to surface shape. Gourdon [Gourdon, 1995] presents a mesh simplification algorithm that reduces the number of vertices in a surface mesh based on a curvature measure and then regularizes the positions of the vertices to create a mesh of uniform vertex density. Eck *et al*. [Eck *et al.*, 1995] and Lounsbery [Lounsbery, 1994] use multiresolution analysis of meshes of arbitrary topology to create meshes at varying levels of detail which can be used for level of detail viewing and low resolution editing of models.

The main difference between our work and previous work is our explicit handling of noise, scene borders due to the limited field of view of the sensor, and varying vertex densities in the original surface mesh that are common when dealing with vision sensors and algorithms. Furthermore, our method produces comparable results to the above methods and is also amenable to appearance (i.e., image texture) mapping.

Our algorithm takes as input a 3-D surface mesh describing the shape of the scene and a co-registered image of the scene. Section 2 details the algorithm for segmenting the faces of the mesh into planar surface patches using a global region growing algorithm. Section 3 explains how the boundaries of the resulting planar patches are simplified to produce a 3-D polygon for each planar patch; Section 4 describes how the interior of each polygon is triangulated to create new mesh faces onto which the appearance of the scene can be mapped. In Section 5, noise in the final surface mesh is reduced by projecting the vertices of the simplified mesh onto the best fit planes. Section 6 details the mapping of appearance onto the surface mesh, and Section 7 shows the performance of

Figure 1: Frontal view of a surface mesh generated from a projective depth map, its corresponding image and the surface mesh with its appearance mapped onto its faces.

the algorithm on an extremely noisy data set. Section 8 explains how world knowledge can be used to improve the shape of the scene, and Section 9 discusses future research directions.

2 Mesh Segmentation

The fundamental structure used by our scene simplification algorithm is a surface mesh. We have chosen this representation because a surface mesh is fully three-dimensional, allowing arbitrary 3-D scenes to be represented (unlike images which can be only represent 2-$\frac{1}{2}$ D scenes). Generalizing the scene description from images to a 3-D structure allows us to use depth maps from different sensing modalities (stereo, depth from focus, and light-stripe and laser range finding) and multiple viewpoints.

We further restrict the surface meshes used for appearance mapping to be composed only of triangular faces. This ensures that the appearance of the world can be efficiently and unambiguously mapped onto the surface mesh using linear texture mapping techniques from computer graphics. The appearance of the scene can be mapped onto the surface mesh if each triangular face can be located in a single image that is being mapped onto the scene. Note that the segmentation and subsequent planar patch boundary simplification techniques that we have developed do not depend on the faces of the mesh having a fixed number of edges or topology. It is only when the appearance of the scene is to be mapped onto the scene that triangular faces are needed.

Generating a surface representation from 3-D data is an area of active research, as indicated by papers such as [Eck *et al.*, 1995; Hoppe *et al.*, 1993; Lounsbery, 1994; Schroeder *et al.*, 1992].

However, a triangular surface mesh can be generated easily from a set of 3-D points that have associated image coordinates in a single image because the points can be considered to lie on a 2-D manifold parameterized by the image coordinates. A planar triangulation scheme applied to the image coordinates will establish the connectivity of the points. Each planar edge and face in the triangulation of the image coordinates has a corresponding 3-D edge and face resulting from connecting 3-D points whose image coordinates are connected by the triangulation. We choose the Delaunay triangulation of the points to construct our surface meshes because this triangulation scheme connects points that are nearest neighbors in the image. Generally this will result in a triangular mesh faces that are as close to equilateral as possible in the plane [Preparata and Shamos, 1985]. Connecting points that are near to each other in the image will connect points that are near to each other in 3-D, unless there exists a large depth discontinuity between the points. The segmentation relies on the surface mesh for adjacency information, so it is important that the mesh connect points that are near to each other. Figure 1 shows a 3-D frontal view of a surface mesh created from a dense projective depth map using a structure from motion algorithm [Szeliski and Kang, 1995], the corresponding color image and the same surface mesh with appearance (i.e., image texture) mapped onto it.

The first processing stage the surface mesh must undergo is a partitioning into planar surface patches. We use a modified version of the segmentation algorithm for surface meshes presented in [Hebert *et al.*, 1995]. The segmentation proceeds by a global region growing process in which neighboring regions in the mesh are merged until a threshold on the number of regions or total mesh error is passed. This segmentation procedure is ideal for partitioning 3-D surface meshes generated from real data because the resulting segmentation is pose and scale invariant. This segmentation procedure also handles meshes of arbitrary connectivity and topology.

A planar surface patch is defined as a group of adjacent vertices in the mesh that lie close to the same plane. A plane is defined with the equation $\hat{\mathbf{n}}^{\mathrm{T}}\mathbf{p} + d$, where $\hat{\mathbf{n}}$ is the unit surface normal of the plane, and d is the perpendicular distance of the plane to the origin. The RMS planar fit error of the planar patch is defined as

$$\sum_i \left(\hat{\mathbf{n}}^{\mathrm{T}}\mathbf{x}_i + d\right)^2, \tag{1}$$

where $\mathbf{x}_i$ are the points grouped by the patch. The plane parameters ($\hat{\mathbf{n}}$ and d) are found by minimizing (1) over all of the data in the patch. The well-known solution for the surface normal is the eigenvector of the smallest eigenvalue of the inertia matrix of the points [Duda and Hart, 1973].

The best fit plane passes though the center of mass $\mathbf{c}$ of the points so $d = -\hat{\mathbf{n}}^{\mathrm{T}}\mathbf{c}$.

The partitioning of the mesh is initialized by creating a planar surface patch from the points defining each face in the mesh. From these initial planar patches, an adjacency graph is created, where each face is considered a node and each edge connects faces that share an edge in the mesh. Each edge between patches in the adjacency graph is weighted by the RMS planar fit error given by (1) of the points within the two patches that it connects. The edges of the adjacency graph are inserted into a priority queue sorted on planar fit error.

Larger regions are grown from the initial partitioning by merging adjacent regions. At each merge, the edge with the minimum planar fit error is taken from the top of the edge priority queue; the two regions that it connects are then merged by combining their points into one planar patch. The new patch is subsequently connected to all of the patches that were adjacent to the previous two regions. The planar fit error of the new connections are calculated and the new edges are inserted into the priority queue.

There exist two thresholds for stopping the merging of the planar patches which are appropriate in different situations. If the user is able to view the scene before simplification occurs, an estimate of the number of planes in the scene can be determined and used as an input threshold parameter T_N. Segmentation stops when the number of planes reaches T_N. The other method relies on limiting the sum of planar fit error for all the planar patches in the mesh. This method applies when a more automatic stopping criterion is desired. First an estimate of noise and curvature in the scene $\bar{E}$ is calculated by finding the average of the distance of each vertex in the mesh from the best fit plane of its neighboring vertices. $\bar{E}$ increases as the noise and curvature in the data increases; it is also a measurement of scale for the expected total planar fit error of the mesh. Setting the threshold on the total planar fit error as some fraction of $\bar{E}$ removes much of the consideration of scale and shape from the setting of the threshold. Since the goal of this work is to reduce polygon count in the surface mesh and not feature extraction for object recognition, the threshold can be set at some fixed fraction of $\bar{E}$ with good results for all surface meshes. In practice, we have found a good threshold on the total planar fit error to be $0.2 * \bar{E}$.

A necessary result of initializing the adjacency graph with faces (as opposed to points) is that every two adjacent patches share a chain of points along the boundary between the patches. Thus a natural way of defining the boundary between two patches using mesh vertices exists, and ad-hoc procedures for determining the boundary are not needed. The boundaries between regions are used to define the extent of the planar patches and are used in subsequent processing stages.

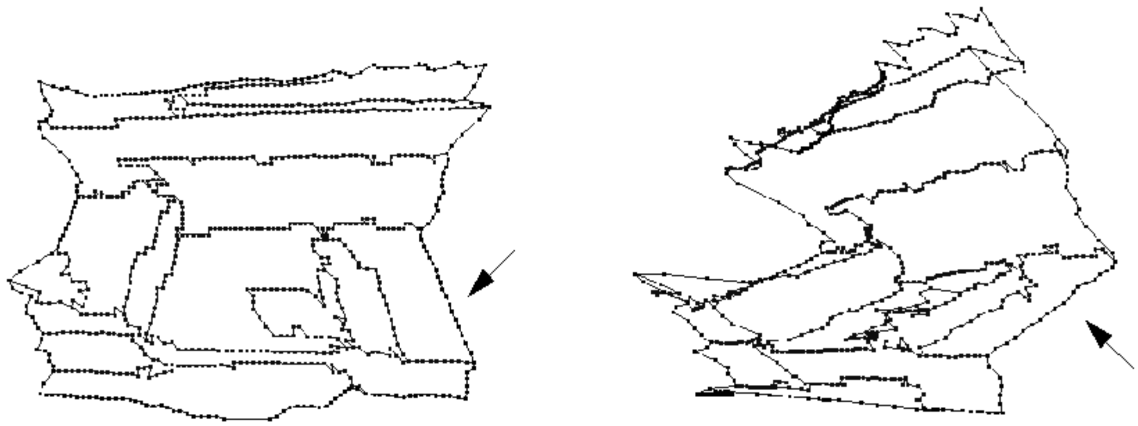


Figure 2: Frontal and side views of the boundary edges and vertices of the planar surface patches after segmentation of the mesh in Figure 1 into 14 patches. A redundant chain of point is marked with an arrow in both views.

The purpose of segmenting the mesh is to determine roughly planar groupings of points in the surface mesh. Given that all of the points within each patch are close to the plane to which they are fit, the faces in each patch can be replaced by faces that lie in the fit plane without large changes in surface shape. Using points in the interior of the patches to determine the faces within each patch is undesirable when simplifying the mesh. This is because extra and unnecessary faces will be created. Hence, all points in the interior of the patch are eliminated from the surface mesh, and the faces within each patch are created only from the points on the boundary of the patch. Figure 2 shows the boundaries between planar patches (after segmenting the surface mesh shown in Figure 1 into 14 planar surface patches). The boundary vertices are easily extracted from the mesh by finding the vertices that are grouped by more than one planar patch or lie on the border of the mesh. The boundary edges are between faces belonging to two different patches or are part of only one face and hence on the border of the mesh.

3 Simplification of Planar Patch Boundaries

The result of the segmentation stage of the algorithm is a wireframe mesh representation that delineates the boundaries of planar patches in the original mesh. A *chain* is an ordered list of points and edges in the mesh that either lie between two planar patches or is part of the mesh border. The chain ordering is based on adjacency in the mesh. The wireframe representation of the scene is composed solely of chains. Some of the chains in the wireframe contain redundant information about the shape of the surface mesh. In particular, chains that lie roughly along straight lines in space are redundant because the chain is adequately represented by the endpoints of the chain. Linear chains are common in the wireframe, because many chains lie along the line created by the intersection of two planar patches. An obvious redundant chain in the example wireframe has been flagged with an arrow in Figure 2. Scene simplification entails the elimination of all redundant points and faces used to represent the scene. This means that the redundant points in the chains must be eliminated.

The redundancy in the wireframe can be removed by using an iterative end point fitting algorithm in 3-D [Duda and Hart, 1973]. We have chosen the simplest form of the algorithm with a fixed threshold and no post fitting. This simple approach is taken because our notion of scene simplification is primarily motivated by appearance mapping and faster rendering, which does not necessarily call for optimal simplification. Each chain is processed as follows:

The line $\overline{AB}$ between the endpoints A and B of the chain is determined.

The distance to $\overline{AB}$ of all of the interior points is calculated.

IF none of the distances are greater than a threshold, the process terminates.

ELSE the farthest point C is added to the fit line and the process is repeated on $\overline{AC}$ and $\overline{AB}$.

The result of the algorithm is a set of simplified chains where the distance to the original chain is less than the specified threshold. The threshold is normalized by the average edge length in the original mesh to eliminate scaling effects on the setting of the threshold. The resulting wireframe mesh from Figure 2 after simplification is shown in Figure 3. The simplification of the planar patch boundaries is significant; this is evident by comparing the flagged chains in Figure 2 and Figure 3.

4 Triangulation of Patch Boundary Polygons

The wireframe resulting from the simplification of the planar patch boundaries contains all of the vertices that will be in the final simplified mesh. The boundary of each planar patch is a 3-D non-

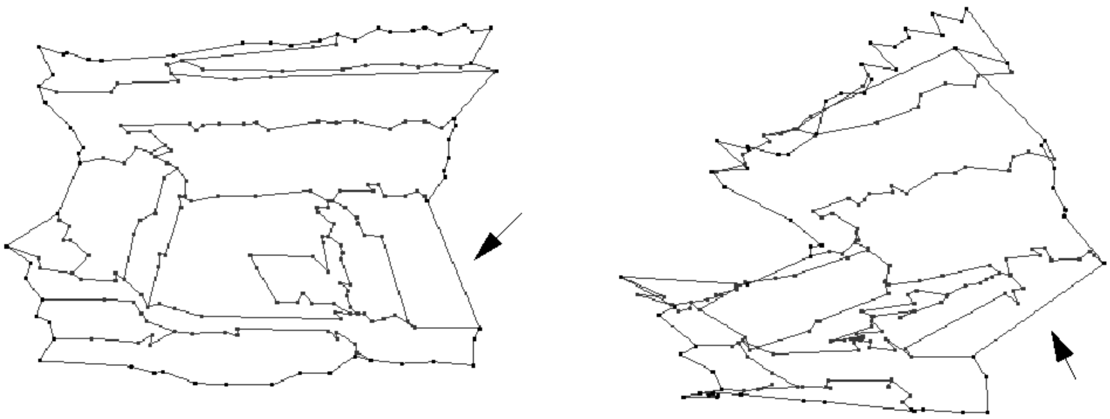


Figure 3: Frontal and side views of the simplified boundaries of the planar patches shown in Figure 2. Arrows mark the result of the simplification of an especially redundant chain of points.

planar polygon whose interior defines a roughly planar surface. However, if a texture mapped or shaded model of the scene is to be created, the interior of each patch boundary must be broken into convex planar facets to create a surface for rendering.

The straight-forward method that we use to creating convex facets in the interior of a patch boundary polygon is to project the boundary polygon onto a 2-D surface before triangulating the interior of the boundary. The triangulation will break the interior of the boundary polygon into triangular facets which are both planar and convex.

At this point, there is no grouping of the vertices in the wireframe into planar patch boundary polygons; hence, the first step in the triangulation is to determine the ordering of the vertices in each boundary polygon. For each planar patch, the ordering proceeds as follows: First the set of vertices in the simplified mesh grouped by the planar patch is determined. A starting vertex is chosen from this set and is initialized to the first element of the ordered list of vertices describing the boundary polygon. The next vertex in the polygon is chosen from the two vertices that are adjacent to the starting vertex in the simplified mesh, and is appended to the ordered list of boundary vertices. The rest of the vertices in the boundary are then ordered by iteratively appending the next adjacent vertex in the simplified mesh that is not yet a member of the ordered list until the starting vertex is returned to.

If not every vertex in the set of boundary vertices has been appended to the ordered list, then there exists a hole(s) in the planar patch caused by a one planar patch completely surrounding another. To completely describe the boundary of the polygon, the above process is repeated on the remaining boundary vertices. The two ordered lists of vertices then describe the interior and exterior boundary polygons of the planar patch.

Once the boundary polygon of the planar patch is determined, the vertices of the polygon are projected into 2-D so that the interior of the boundary polygon can be triangulated. In the case of appearance mapping or image-based range maps, image coordinates exist for every point in the mesh. By mapping the vertices of a boundary polygon to their image coordinates, a simple 2-D polygon is generated.

When the vertices in the mesh do not have corresponding image coordinates, the 2-D boundary polygon is generated by projecting the polygon vertices onto the best fit plane of the planar patch that corresponds to the boundary polygon. Occasionally this mapping will generate non-simple 2-D polygons. Future work will investigate ways to handle this case without changing the topology of the surface mesh.

The simple 2-D polygons corresponding to each planar patch boundary can now be triangulated. We use a greedy algorithm to triangulate the interiors of these polygons that is tailored to produce regular triangular faces. A pseudo-code description of the triangulation procedure follows:

```
Initialize the set T of all segments in the triangulation with
  the segments making up the boundary polygon.

Create the set S of all possible segments in the triangulation
  by connecting every point in the polygon to every other
  point.

Eliminate from S all segments that intersect the polygon.

Eliminate from S all segments whose midpoint is outside the
  polygon

Sort S based on segment length

REPEAT

Pop the shortest segment off of S and add it to T if it does
  not intersect any segments in T.

UNTIL S is empty
```

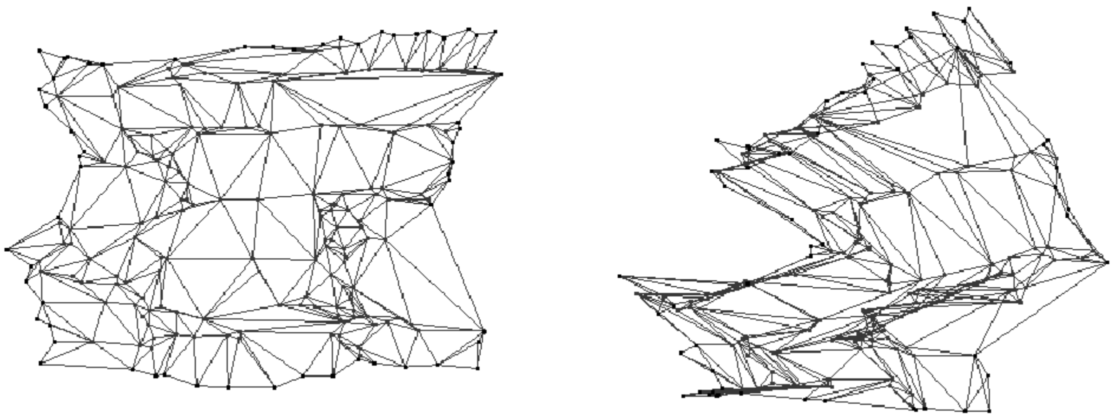


Figure 4: Frontal and side view of the triangulated surface mesh generated from the wireframe shown in Figure 3.

The connectivity of vertices in the 2-D polygons are used to connect the vertices in the 3-D boundary polygons. The result of the triangulation of the simplified wireframe in Figure 3 is given in Figure 4. Triangulating the planar patch boundaries has created a triangular surface mesh from the simplified wireframe. The final stage of the simplification algorithm will reduce the noise in the vertices of the mesh by projecting them onto the best fit planes of the planar patches.

5 Vertex Projection

The noise present in the original surface mesh will affect the shape of the simplified surface mesh because every vertex in the simplified mesh comes from the original mesh. However, in the case of a polyhedral world, the segmentation of the mesh into planar patches can be used to reduce the effects of this noise on the final simplified surface mesh. Because the segmentation algorithm groups vertices based on a least squares fitting criterion, the parameters of each fit plane will average out the random errors present in the mesh vertices. We exploit the stability of the plane parameters to reduce the effects of noise on the final shape of the simplified mesh.

Every vertex in the simplified mesh is either on the boundary between two or more planar patches or on the border of the mesh. The noise in the position of the mesh border vertices is reduced by projecting each vertex perpendicularly onto the plane associated with the planar patch to which the vertex is a member. Vertices that lie on the boundary between two planar patches are projected per-

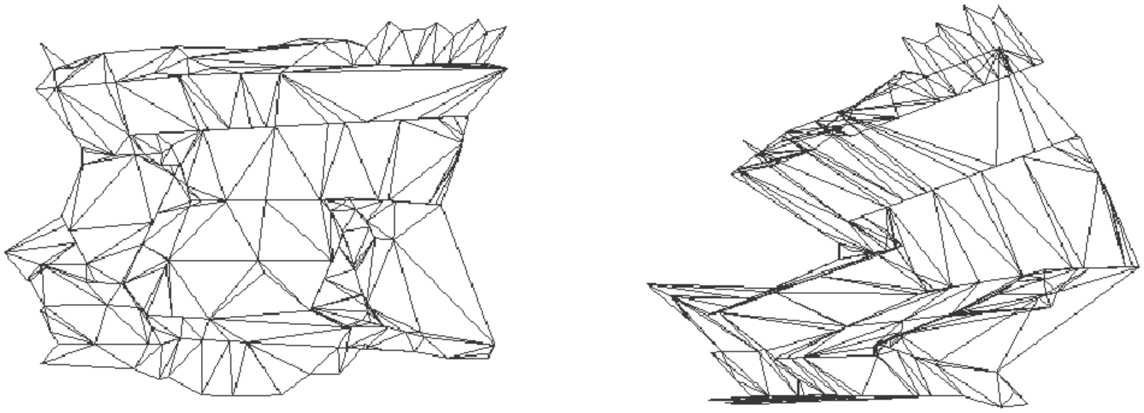


Figure 5: Frontal and side view of the surface mesh shown in Figure 4 after the mesh vertices have been projected onto the best fit planes generated from the mesh segmentation.

pendicularly onto the line caused by the intersection the two planar patch planes. Vertices that are on the boundary between $k \geq 3$ planes are moved to the intersection of the planes found by solving the following set of equations

$$\begin{aligned} \mathbf{n}_1^{\mathrm{T}} + d_1 &= 0 \\ \mathbf{n}_2^{\mathrm{T}} + d_2 &= 0 \\ &\cdots \\ \mathbf{n}_k^{\mathrm{T}} + d_k &= 0 \end{aligned} \tag{2}$$

for x. In the overconstrained case, the least squares solution for x can be found using the pseudoinverse. The simplified mesh shown in Figure 4 after the mesh vertices have been adjusted by projection onto proximal planes is given in Figure 5.

The projection of mesh vertices onto planes in the segmentation is appropriate when the scene is polyhedral or when there is a large amount of noise in the data. The projection step is less favorable when the scene has many curved surfaces and the data is accurate because the projection will have the effect of creating corners in the scene that do not exist. Occasionally a vertex will exist at the intersection of planes that are close to parallel. In this case, the intersection of the planes could be far from the original position of the vertex; hence, to prevent drastic shape changes, the projection of the vertex is compared with its original position. If the distance between the two locations is

data set	number points	number polygons	polygon reduction	$E_{\rm avg}$	$E_{\rm stdev}$	$E_{\max}$	number planes	T_S
original	3072	5922						
simplified 1	204	366	16.2	1.01	1.19	11.28	14	5.0
simplified 2	74	112	52.9	1.64	1.72	17.36	8	10.0

Table 1: Simplification reduction and error statistics for the meshes shown in Figure 6.

large, the vertex is not projected.

At the top row of Figure 6, the original surface mesh from Figure 1 is shown next to the simplified surface mesh of Figure 5. Another surface mesh of increased simplification that was generated from the same original surface mesh, but with fewer final surface patches and a higher boundary simplification threshold, is shown in Figure 6 for comparison. The original mesh has 5922 polygons and the simplified meshes have 366 and 112 polygons, which amounts to a 16 and 59 times reduction, respectively. This large degree of reduction is possible without drastically changing the shape of the scene. A summary of the simplification results and errors introduced by the simplification is listed in Table 1. The error statistics come from the distribution of distances of the points in the original mesh to the simplified meshes. The distance between a point and a mesh is defined as the distance from the point to the face closest to the point in the mesh. The error statistics are in units of pixels in the depth map. The vertices in the original mesh are on average five pixels from adjacent vertices because the depth map was subsampled by one in five to create the original mesh. Because the average and standard deviation of the error distributions of both simplified meshes are close to one (smaller than the average distance between vertices), it can be quantitatively concluded that the shape of the simplified meshes do not change drastically from the original mesh. The maximum errors are attributed to simplification on the border of the mesh. This leaves some original mesh vertices hanging as a result of the removal of faces on the border of the mesh.

6 Image Mapping

Projecting the mesh vertices changes their 3-D positions as well as their location in the image that is used for appearance mapping. As a result, the new image coordinates of each mesh vertex are

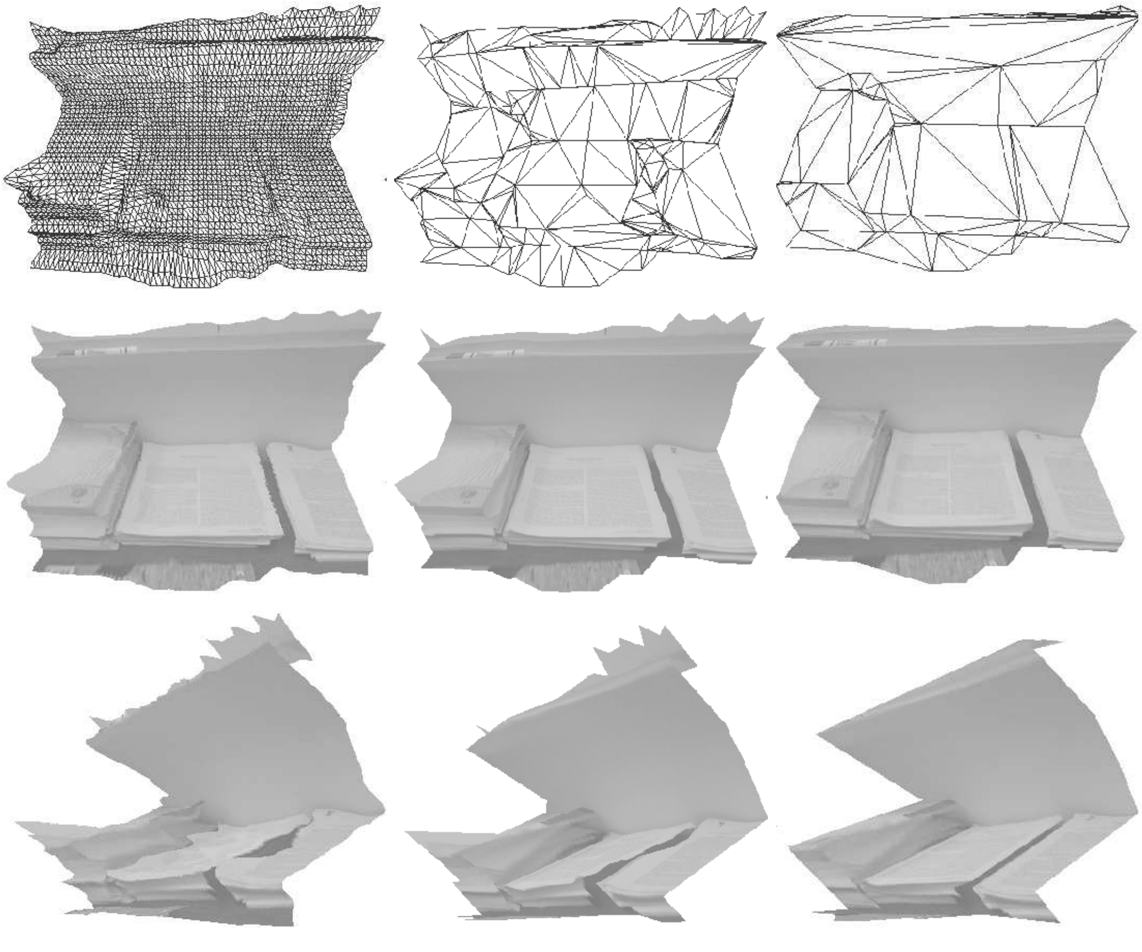


Figure 6: A comparison of the original surface mesh to two simplifications. Column (b) was created from a segmentation of the surface mesh into 14 planes, column (c) from 8 planes. The original mesh is composed of 5922 faces while the simplified surface meshes have 366 and 112 faces, respectively.

calculated using the new 3-D position of the vertices and the sensor image projection. New image coordinates must be calculated for correct mapping of scene appearance onto scene shape. Once the image coordinates are known for all of the vertices in the mesh, graphical texture mapping is used to map the image onto the triangular faces of the surface mesh, assuming that the image projection is linear. In Section 7, we will show how to handle non-linear image projections.

Figure 6 shows a comparison of the appearance of the simplified surface meshes to the appearance of the original surface mesh. As the degree of simplification increases, more and more details of the surface shape are eliminated while the overall shape of the scene is retained. The noise reduction feature associated with our method is apparent from the oblique views of the scene. In the original mesh, noise in the data causes the appearance of the top of the stacks of paper to appear distorted. However, after mesh simplification, the distortion in the appearance of the appearance of the papers decreases. This is a direct effect of projecting the vertices in the simplified meshes onto fit planes to reduce noise in the simplified mesh.

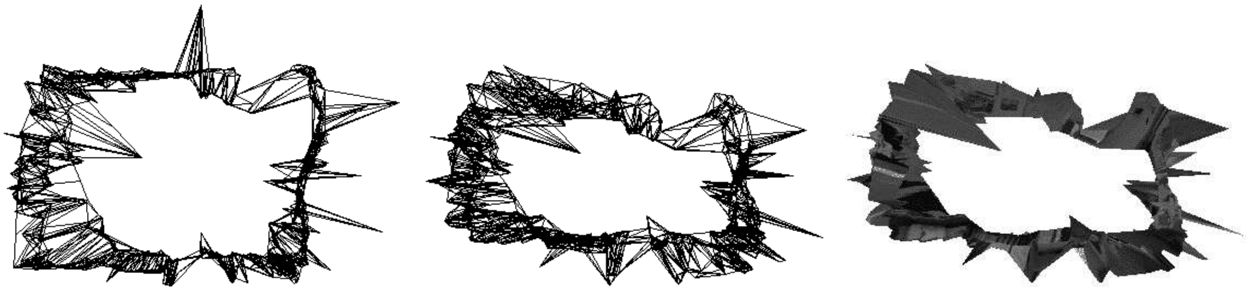
7 A Difficult Data Set

So far the results presented in this paper have been for a relatively smooth projective depth map with a fixed point density. This section shows the effectiveness of our algorithm in simplifying a surface mesh with both large amounts of noise and varying point density.

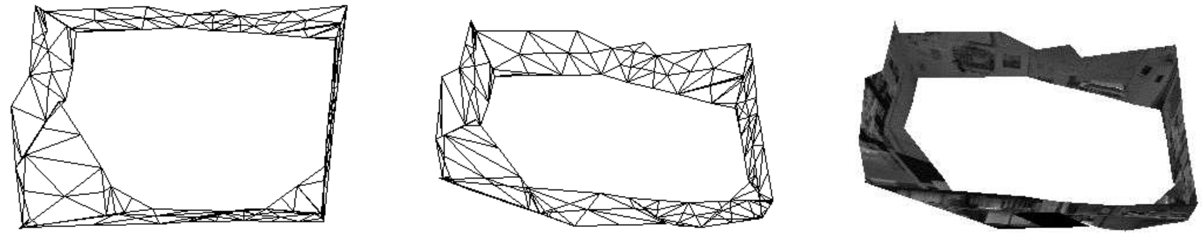
The raw data set shown at the top of Figure 7 was generated using a multibaseline omnidirectional stereo system that produces cylindrical depth maps though stereo matching of 360° panoramas [Kang and Szeliski, 1995]. A panorama of the scene is generated by acquiring a stream of images taken as the camera is rotated 360° about the vertical axis passing through its optical center. The image stream is composited in a serial fashion to produce a full 360° cylindrical panoramic image of the scene. Panoramas are created at multiple camera positions in the scene. Point features are then tracked across these panoramas and the 8-point structure from motion algorithm [Longuet-Higgins, 1981; Hartley, 1995] is applied to these feature positions to recover the epipolar geometry and the camera relative poses. The extracted camera information subsequently enables 3-D positions of the scene to be determined. A surface mesh is generated using the Delaunay triangulation of the image coordinates of the feature points. The input to our scene simplification routine is a surface mesh co-registered with a cylindrical image.

The raw data shown in Figure 7 was generated from panoramas of an office which have views

(a) Cylindrical panorama of an office



(b) Top view of mesh (c) Oblique view of mesh (d) Texture-mapped version of (c)



(b) Top view of mesh (c) Oblique view of mesh (d) Texture-mapped version of (c)

Figure 7: A cylindrical panorama of an office. An oblique and top view of a raw surface mesh generated from an omnidirectional multibaseline stereo system and an oblique appearance mapped view of the raw data. A simplified surface mesh generated from the raw data showing the 6 prominent planes in the scene.

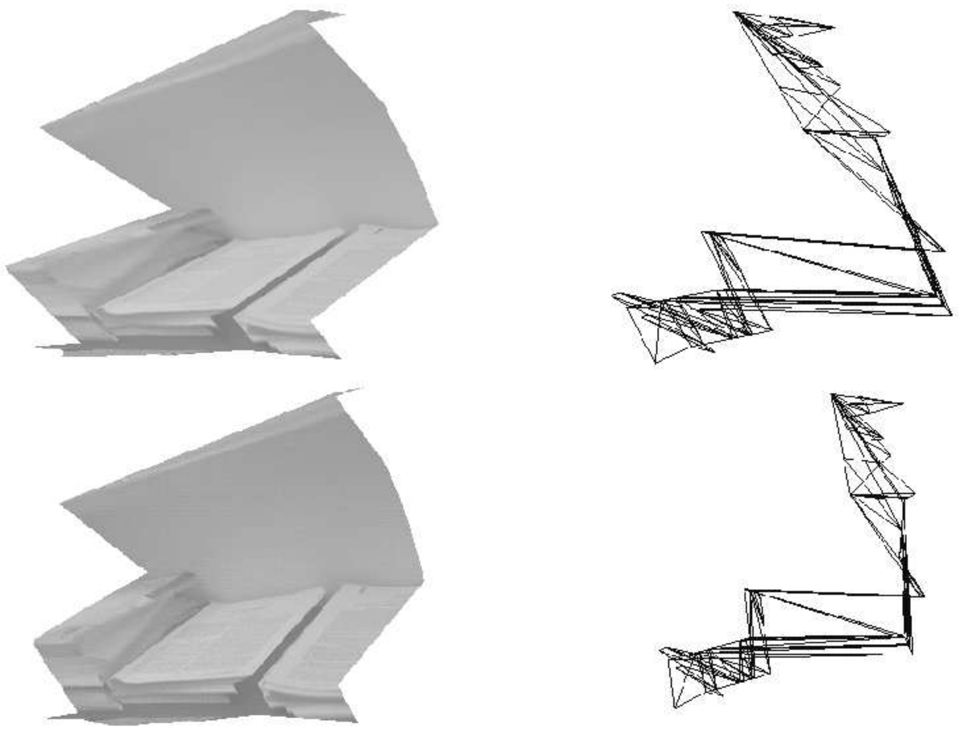
of windows, bookshelves, and computer monitors. The data is noisy and has varying densities of vertices in the mesh. In spite of this, our algorithm is capable of extracting the prominent surfaces in the scene because the segmentation of the mesh into planar patches is not affected by the connectivity or distribution of points in the mesh. The noise in the scene is reduced because the points in the simplified surface mesh are projected onto the best fit planes that the segmentation determines. Figure 7 shows the extracted surface model of the room generated from the six most prominent planes in the scene. The original surface mesh has 2701 points and 5309 polygons and the extracted model has 88 points and 131 polygons. Of course, the segmentation of the scene is affected by the original data, so the final best fit planes are not exactly orthogonal as is expected of walls in a room[1]. The appearance of the scene is extremely distorted when mapped onto the original surface mesh. However, when mapped onto the simplified mesh the appearance of the room is much clearer: the book shelves, whiteboard and other objects can be made out clearly. Therefore, the simplified mesh is a more useful model of the scene for viewing purposes.

Another difficulty with viewing appearance mapped scenes is dealing with non-linear image projections. For example, the cylindrical panoramas used in omnidirectional stereo map straight horizontal lines in the world into curved lines in the image. If the appearance of the scene were taken directly from the image, curved lines would be mapped onto the scene in places where straight lines actually exist. To eliminate this problem, we break non-linear images up into smaller linear images that correct for the distortions. For example, the distortions in a cylindrical image are corrected by covering the image with overlapping planar images at regular angular intervals with a fixed field of view. The appearance of a pixel in the planar images is created by finding the intersection of the ray from the optical center to the planar image pixel with the non-linear image. Bilinear interpolation of the four nearest pixel values in the non-linear image ensures that the planar image will have a smoother appearance. Special attention is given to the arrangement and overlap of the planar images so that every face in the simplified scene can be appearance mapped from a single image.

8 Application of World Knowledge

Another benefit of our algorithm is the creation of high-level planar descriptions of the scene which can be manipulated to improve the shape of the scene based on world knowledge. The best fit planes

[1] Actually, the points are not expected to generate exactly perpendicular planes because a significant number of the extracted points are those of the bookshelves and computer monitors, which protrude out of the walls.

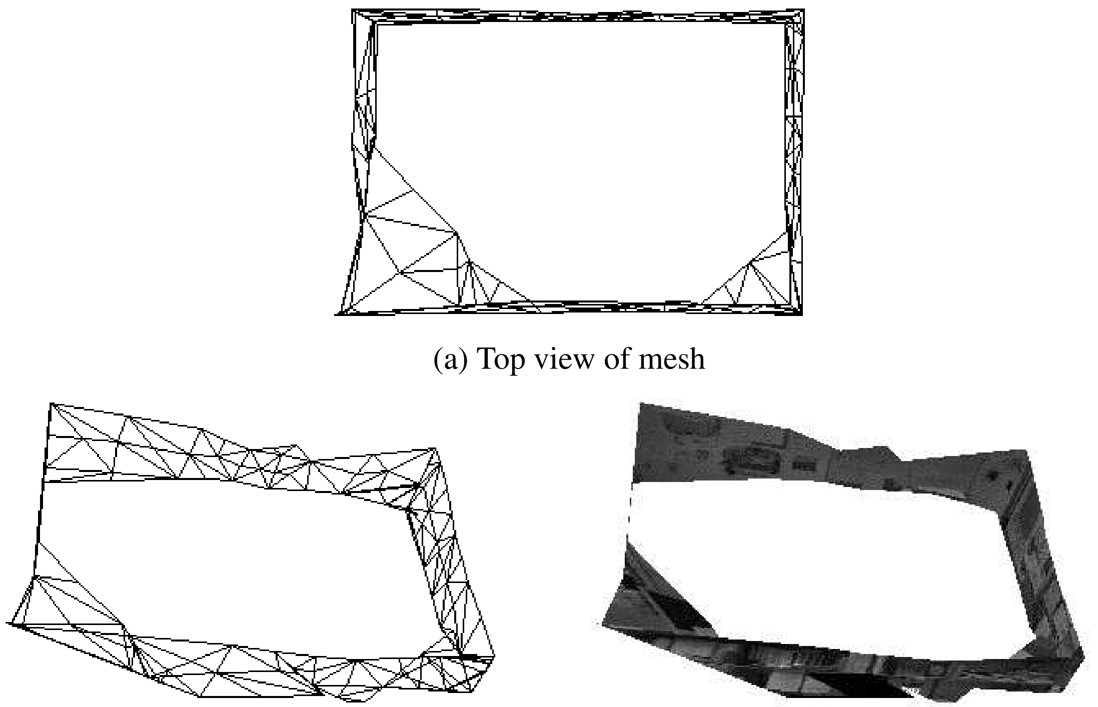


(a) Oblique views (b) Side views

Figure 8: Views of the simplified mesh shown on Figure 6(c) before (1st row) and after (2nd row) correction of depth scale using world knowledge.

determined by the segmentation of the surface mesh may not exactly correspond to planar descriptions of the scene due to systematic sensor noise or incorrectly scaled depth estimates in the shape recovery algorithm. However, given constraints on the parameters of the best fit planes, vertices in the mesh can be adjusted to correctly model the scene.

The results given in Figure 6 were generated using a projective depth map, where the scale on the depth estimates was chosen arbitrarily to create the original surface mesh. With the segmentation of the surface mesh into planar patches, knowledge about the orientation of the normals of the adjacent planar patches can subsequently be used to solve for the correct depth scale. It is known that many of the planes in the scene should be orthogonal, which places constraints on the normals of the planes. These constraints are used to solve for the correct depth scale, which can then be applied to the vertices in the mesh to correct the shape of the model. Figure 8 shows a comparison between an oblique view of the appearance and a side view of the surface mesh of the simplified scene with



(a) Top view of mesh

(b) Oblique view of mesh (c) Oblique view of texture-mapped model

Figure 9: Views of simplified surface mesh from Figure 7 after planes have been made orthogonal by applying world knowledge.

the original and corrected scales. With the corrected scale, the scene more accurately models the real world; the tops of the stacks of papers are perpendicular to the wall and parallel to each other.

The final planes in the simplified surface mesh in Figure 7 are not orthogonal, as they should be for a correct model of the room, because noise in the surface mesh has effected the parameters of the best fit planes. Forcing the surface normals of the best fit planes corresponding to walls meeting at a corner to be orthogonal and perpendicular to the z-axis (up) results in the corrected scene model shown in Figure 9.

9 Conclusions

We have presented an algorithm to extract concise models of scene shape from dense surface meshes created from real 3-D data sets. This algorithm also lends itself well to appearance mapping. We have demonstrated its use on projective depth maps and noisy omnidirectional stereo data with an order of magnitude reduction in model faces without significant degradation of scene appearance. The algorithm can handle meshes of arbitrary topology and vertex densities as well as large amounts of noise in the data. The resulting simplified models are less noisy than the data and can be further corrected to accurately model the world based on high-level world knowledge.

The algorithm is based on the segmentation of a surface mesh into planar surface patches, so it is ideally suited to man-made environments in which planar surfaces abound. However, it is also applicable to free-form surfaces, although the simplification may not be as great. In future, we will quantitatively characterize the performance of our algorithm on free-form surfaces and compare its results to other algorithms.

The results we have shown used appearance from a single image. However, complex scenes may require the mapping of appearance taken from many views onto the scene surface mesh. In future, we will investigate the mapping of multiple images onto a single surface. This will require techniques for combining data from overlapping images and elimination of view-dependent effects (e.g., photometric variation and specularities) from the images.

References

[Chen, 1995] S.E. Chen. QuickTime VR – An image-based approach to virtual environment navigation. *Computer Graphics (SIGGRAPH'95)*, :29–38, Aug. 1995.

[Duda and Hart, 1973] R Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[Earnshaw *et al.*, 1993] R. Earnshaw, M. Gigante, and H. Jones. *Virtual Reality System*. Academic Press, London, 1993.

[Eck *et al.*, 1995] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics (SIGGRAPH'95)*, :173–182, Aug. 1995.

[Gourdon, 1995] A. Gourdon. Simplification of irregular surfaces meshes in 3d medical images. *Virtual Reality and Robotics in Medicine (CVRMed '95)*, :413–419, Apr. 1995.

[Hartley, 1995] R. Hartley. In defence of the 8-point algorithm. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 1064–1070, IEEE Computer Society Press, Cambridge, Massachusetts, June 1995.

[Hebert *et al.*, 1995] M. Hebert, R. Hoffman, A. Johnson, and J. Osborn. Sensor-based interior modeling. In *Proceedings of the American Nuclear Society 6th Topical Meeting on Robotics and Remote Systems*, pages 731–737, Feb. 1995.

[Hoppe *et al.*, 1993] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. *Computer Graphics (SIGGRAPH'93)*, :19–26, Aug. 1993.

[Johnson *et al.*, 1995] A. Johnson, P. Leger, R. Hoffman, M. Hebert, and J. Osborn. 3-d object modelling and recognition for telerobotic manipulation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 103–110, Aug. 1995.

[Kang and Szeliski, 1995] S. B. Kang and R. Szeliski. *3-D Scene Data Recovery using Omnidirectional Multibaseline Stereo*. Technical Report 95/6, Digital Equipment Corporation, Cambridge Research Lab, October 1995.

[Longuet-Higgins, 1981] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[Lounsbery, 1994] J. Lounsbery. *Multiresolution Analysis of Meshes of Arbitrary Topological Type*. PhD thesis, University of Washington, 1994.

[Preparata and Shamos, 1985] F. Preparata and M. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.

[Schroeder *et al.*, 1992] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of triangular meshes. *Computer Graphics (SIGGRAPH'92)*, :65–70, July 1992.

[Szeliski and Kang, 1995] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, Massachusetts, June 1995.

[Turk, 1992] G. Turk. Re-tiling polygonal surfaces. *Computer Graphics (SIGGRAPH'92)*, :55–64, July 1992.