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

Andrew E. Johnson[1]

Sing Bing Kang[2]

Richard Szeliski[3]

[1]The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
aej@ri.cmu.edu

[2]Digital Equipment Corporation
Cambridge Research Lab
One Kendall Square
Cambridge, MA 02139
sbk@crl.dec.com

[3]Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
szeliski@microsoft.com

Abstract

We have developed a novel algorithm for extracting concise 3-D surface models of a scene from real sensor data that can be used for virtual reality modeling of the world. Our algorithm produces an accurate and realistic 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.

CONFERENCE CATEGORY: Shape and Object Representation

* This work was done at Cambridge Research Laboratory and was supported by Digital Equipment Corporation.

1 Introduction

Modeling the world is necessary for many virtual reality applications including tele-robotics [9], tele-presence [3], flight simulation, and modeling of famous buildings and scenes for archiving, education and entertainment [1]. These applications immerse a viewer in a model of the world where the viewer is able to move around and sometimes 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 traditional labor-intensive model building using CAD packages is not necessary. Unfortunately there exist two fundamental problems with modeling 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 (i.e., appearance mapping). Texture 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. [14] present an iterative technique based on local distance and angle criterion for reduction of triangular meshes, while Turk [16] uses a curvature measure to reduce the number of vertices in surface meshes of curved surfaces. Hoppe et. al. [8] 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. 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 is

presented by Gourdon [5]. Eck et. al. [4] and Lounsbery [12] 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 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 producing a 3-D polygon for each planar patch, and Section 4 explains how the interior of each polygon is triangulated creating the new faces of the mesh onto which the appearance of the scene can be mapped. In Section 5 the 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 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 meshes can represent fully three dimensional scenes, un-

like images which can be only represent 2-1/2 D scenes. Generalizing the scene description from images to a 3-D structure allows us to use depth maps from different sensing modalities (stereo, structured light, depth from focus, 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 will ensure 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. It should be noted that our scene simplification algorithm does 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 active area of research. However, a triangular surface mesh can be generated easily from a set of 3-D points if they have associated image coordinates in a single image. The points are embedded in a 2-D manifold parameterized by the image coordinates, so a planar triangulation scheme applied to the image coordinates will establish the connectivity of the points. We choose the Delaunay triangulation of the image coordinates to construct the surface meshes shown in this paper because this triangulation scheme connects points that are nearest neighbors[13]. 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. A 3-D frontal view of a surface mesh created from a projective depth map using a dense structure from motion algo-

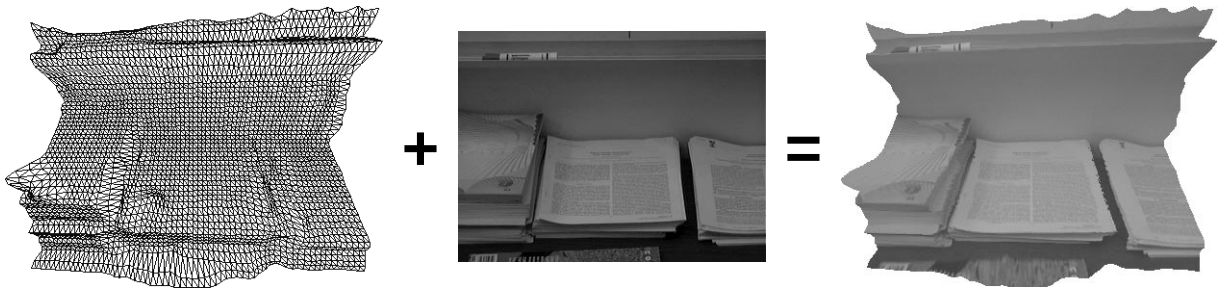
rithm [15], the corresponding color image and the same surface mesh with appearance mapped onto it are given in Figure 1.

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 [7]. The segmentation proceeds by a global region growing process in which neighboring regions in the mesh are merged until a threshold on 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 $n^t x + d$ where n is the surface normal of the plane and d is the perpendicular distance of the plane to the origin. The squared RMS planar fit error of the planar patch is defined as

$$\sum_i \left(n^t x_i + d \right)^2 . \tag{1}$$

where x_i are the points grouped by the patch. The plane parameters (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



(a) **(b)** **(c)**

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

eigenvector of the smallest eigenvalue of the inertia matrix of the points[2]. The best fit plane passes though the center of mass c of the points so $d = -n^t c$.

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 planar fit error given by (1) of the points of the two patches it connects. The edges of the adjacency graph are inserted into a priority queue sorted on planar fit error.

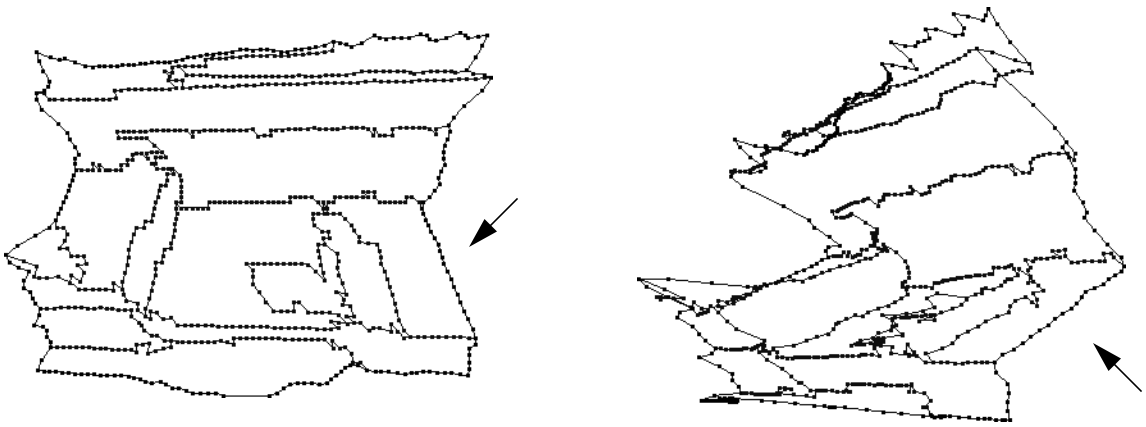
Larger regions are grown from the initial partitioning by merging adjacent regions. At each merge the edge with the minimum planar fit error is popped off of the edge priority queue and the two regions it connects are merged by combining the points they group into one planar patch. The new patch is connected 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.

We have investigated two thresholds for stopping the merging of the planar patches. 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 input as a threshold T_N such that the segmentation stops when the estimated number of planes is reached. The other method relies on limiting the sum of planar fit error for all the planar patches in the mesh and applies when a more automatic stopping criterion is desired. First an estimate of noise and curvature in the scene $\overline{E}$ is calculated by finding the average of the distance of each vertex in the mesh from the best plane fit to its neighboring vertices in the mesh. $\overline{E}$ increases as the noise and curvature in the data increases, and 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 $\overline{E}$ 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 $\overline{E}$ with good results for all surface meshes. In practice we have found a good threshold on the total planar fit error to be $0.2*\overline{E}$.

A 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.

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 made from vertices on the boundaries of the planar patches without large changes in surface shape. Using points in the interior of the patch-



(a) **(b)**

Figure 2: (a) Frontal and (b) 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.

es to determine the faces within each patch is undesirable when simplifying the mesh 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 segmentation of 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 belong to 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 delineating the boundaries of planar patches in the original mesh. A chain is an ordered list of points and edges in the mesh that lie between two planar patches or a planar patch and the border of the mesh where the 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 its endpoints. 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 requires elimination of such redundancy in the points and faces used to represent the scene, so the redundant points in the chains must be eliminated.

The redundancy in the wireframe can be eliminated by simplifying each chain using an iterative end point fitting algorithm in 3-D [2]. We have chosen the simplest form of the algorithm

with a fixed threshold and no post fitting because an optimal simplification is not needed for scene simplification motivated by appearance mapping. 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 is a set of simplified chains where the distance to the original chain is less than the specified threshold T_C. The threshold is normalized by the average edge length in the original mesh to eliminate the effects of scale on the setting of the threshold. The resulting wireframe mesh from Figure 2 after simplification is given in Figure 3. The simplification of the planar patch boundaries is significant as can be seen 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-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 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 and then triangulate the interior of the boundary in 2-D. The triangulation will break the interior of the boundary polygon into triangular facets which are both planar and convex.

At this point in the algorithm there is no grouping of the vertices in the wireframe into planar patch boundary polygons, so 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, the first element of the ordered list of vertices describing the boundary polygon, is chosen from this set. 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, 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 remain-

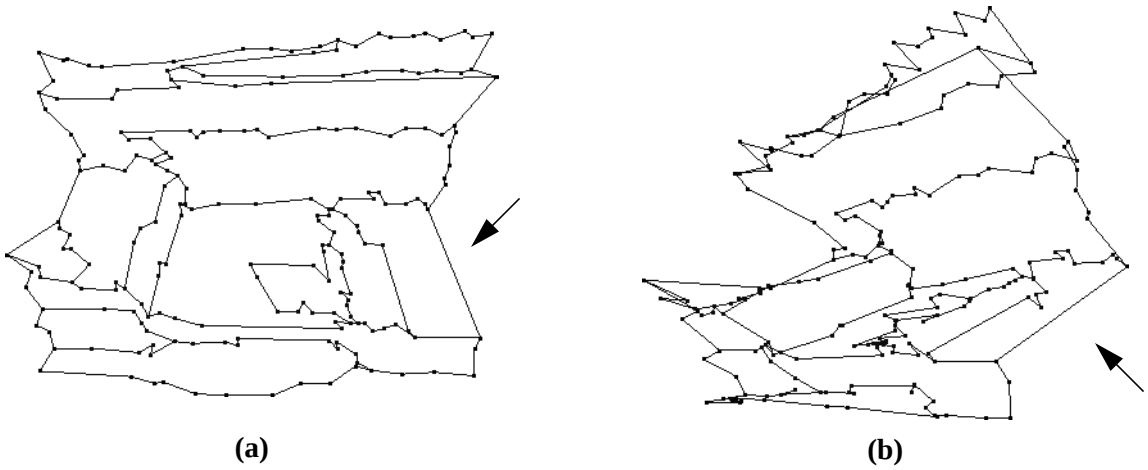


Figure 3: (a) Frontal and (b) side view of the simplified boundaries of the planar patches shown in Figure 2. Arrows mark the result of the simplification of an especially redundant chain of points.

ing boundary vertices. The 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 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.

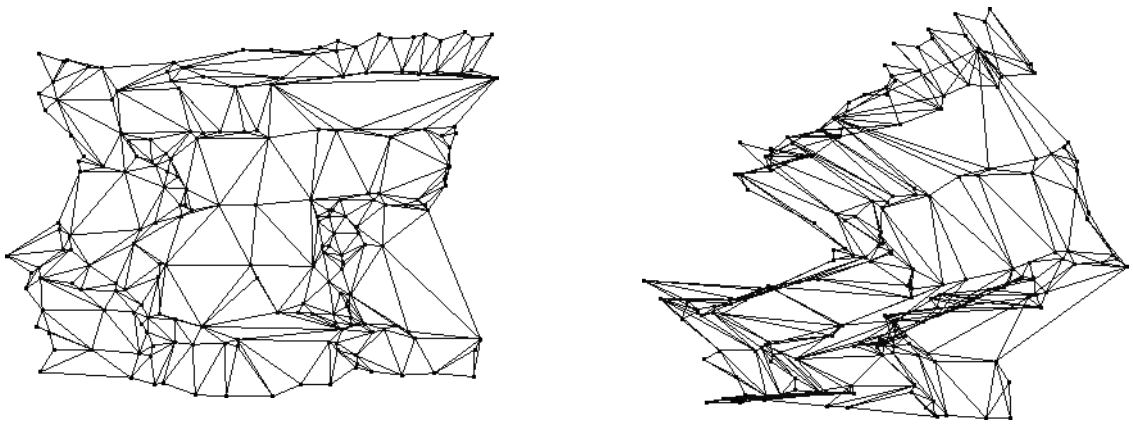
The simple 2-D polygons corresponding to each planar patch boundary can now be triangulated. We use a greedy algorithm to triangulate the interiors of these polygons that is tailored to produce regular triangular faces[13]. A pseudo-code description follows:

```
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 con-
necting every point in the polygon to every other point.

Eliminate from S all segments that intersect the polygon and 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 in-
   tersect any segments in T.

UNTIL S is empty
```

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 up to this point in the algorithm, the position of every vertex in the simplified mesh is the same as its position in 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 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.



(a) **(b)**

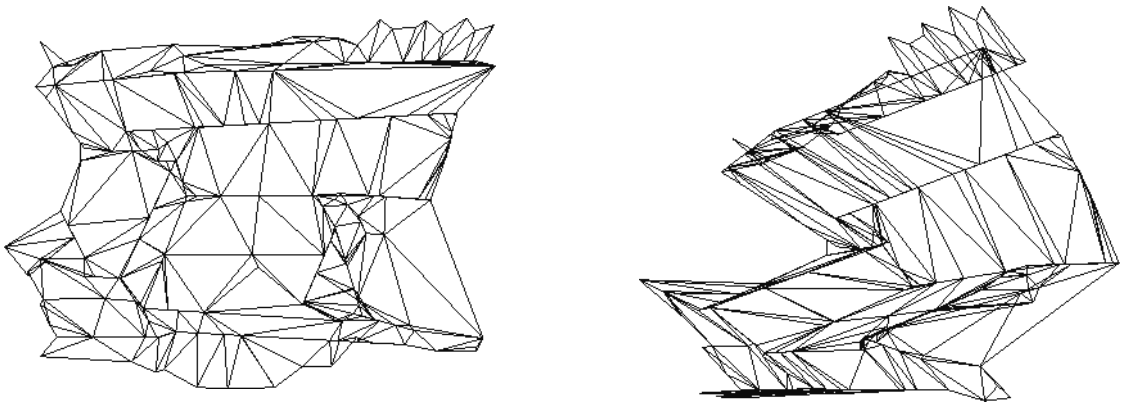
Figure 4: (a) Frontal and (b) side view of the triangulated surface mesh generated from the wireframe shown in Figure 3.

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 of its associated surface patch. Vertices that lie on the boundary between two planar patches are projected perpendicularly onto the line created by the intersection the two planes from the adjacent surface patches. Vertices that are on the boundary between $k \geq 3$ planes are moved to the intersection of the k planes from the adjacent surface patches found by solving

$$n_1^t x + d_1 = 0 \qquad n_2^t x + d_2 = 0 \qquad \ldots \qquad n_k^t x + d_k = 0 \tag{2}$$

for x. In the overconstrained case the least squares solution for x can be found using the pseudo inverse. The triangulated mesh shown in Figure 4 after the position of 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 there is a large amount of noise in the data. Projection 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 in-



(a) **(b)**

Figure 5: (a) Frontal and (b) 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.

tersection 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, so, to prevent drastic shape changes, the projection of the vertex is compared with its original position. If the distance between the two locations is large, the vertex is not projected.

At the top of Figure 6, the original surface mesh from Figure 1 is show next to the simplified surface mesh of Figure 5. Another surface mesh of increased simplification generated from the same original surface mesh, but with fewer final surface patches and a higher boundary simplification threshold is also 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 amount of reduction is possible with out drastically changing the shape of the scene. A summary of the simplification results and errors introduced by the simplification is given 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 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 which leaves some original mesh vertices hanging far from the simplified mesh because of the removal of faces on the border of the mesh.

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

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

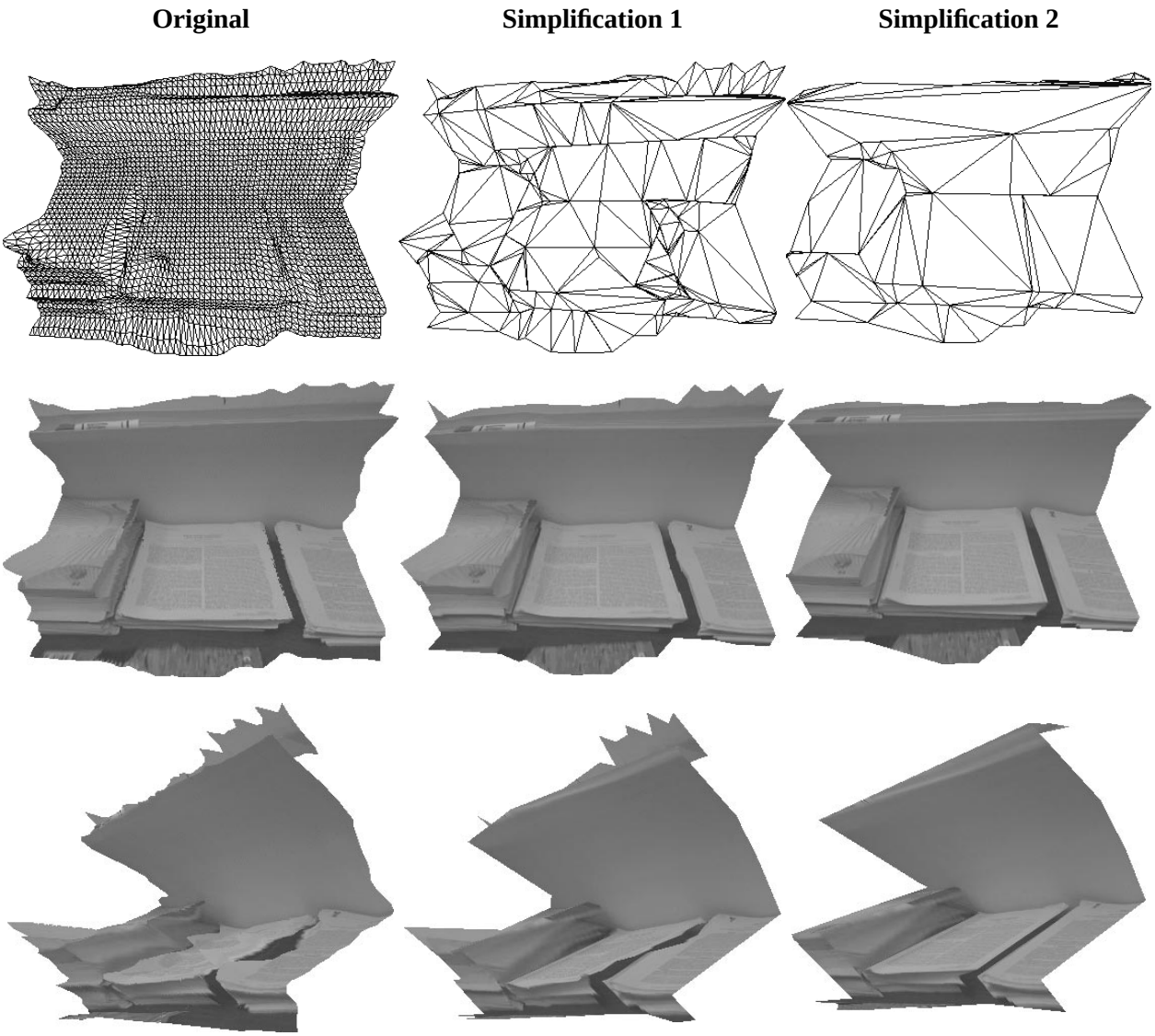


Figure 6: A comparison of the original surface mesh to two simplifications. The middle simplification was created from a segmentation of the surface mesh into 14 planes, the right from 8 planes. The original mesh is composed of 5922 faces while the simplified surface meshes have 366 and 112 faces.

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, so the new image coordinates of each mesh vertex are 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 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 simplification increases, details of the surface shape are eliminated, but the overall shape of the scene is maintained. The noise reduction possible with our method is apparent when considering the sequence of 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, but with simplification of the mesh, 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. In this section will explore the effectiveness of our algorithm in simplifying a surface mesh with 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 degree

panoramas [10]. A panorama of the scene is generated by acquiring a stream of images taken as the camera is rotated 360 degrees about its optical center. The image stream is composited in a serial fashion to produce a full 360 degree cylindrical panoramic image of the scene. Panoramas are created at multiple camera positions in the scene. 3-D positions of the scene are subsequently recovered by tracking features across these panoramas and applying the 8-point structure from motion algorithm on the tracked features.* 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 of windows, bookshelves and computer monitors. The data is very 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 density of vertices 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 effected by the original data, so the final best fit planes are not exactly orthogonal as is expected of walls in a room. 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

* Actually, the 8-point algorithm only recovers the epipolar geometry and the camera relative poses. A second step is required to recover 3-D data.

is much more clear: 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.

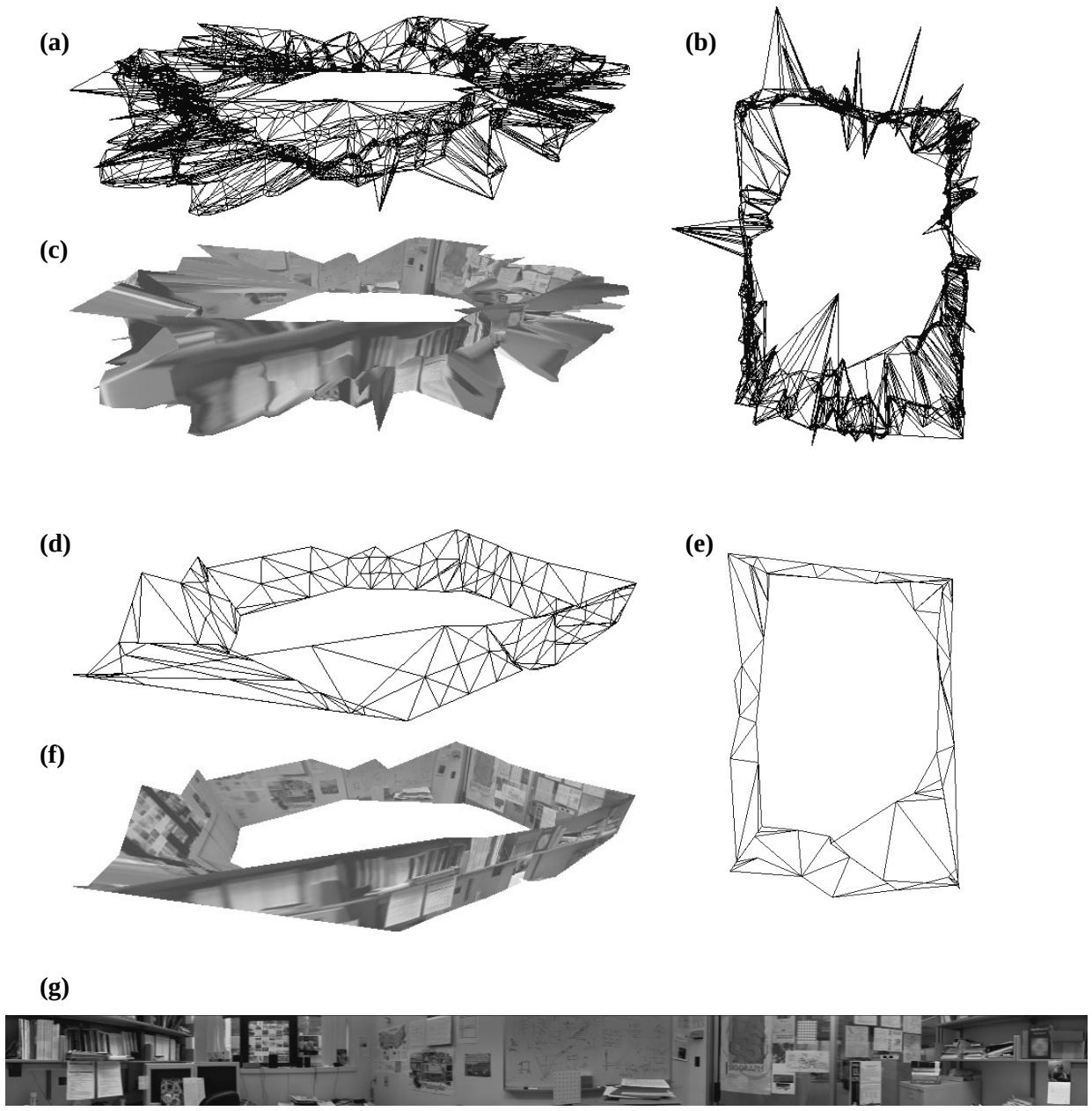


Figure 7: (a) An oblique and (b) top view of a raw surface mesh generated from an omnidirectional multibaseline stereo system and (c) an oblique appearance mapped view of the raw data. (d) An oblique and (e) top view of the simplified surface mesh generated from the raw data showing the 6 prominent planes in the scene. (f) An oblique appearance mapped view of the simplified surface mesh clearly showing objects in the office. (g) One of the cylindrical panoramas used in the omnidirectional stereo.

Another difficulty with viewing appearance mapped scenes is dealing with non-linear image projections. For example, the cylindrical panoramas used in omni-directional 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 smooth 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 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 segmenta-

tion of the surface mesh into planar patches, knowledge about the orientation of the normals of the adjacent planar patches can be used to solve for the correct depth scale. It is known that many of the planes in the scene should be orthogonal or parallel 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 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 the original scale and the correct scale. With the corrected scale, the scene more accurately models the 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 contaminated the parameters of the best fit planes. Constraining the surface normals of the best fit planes corresponding to

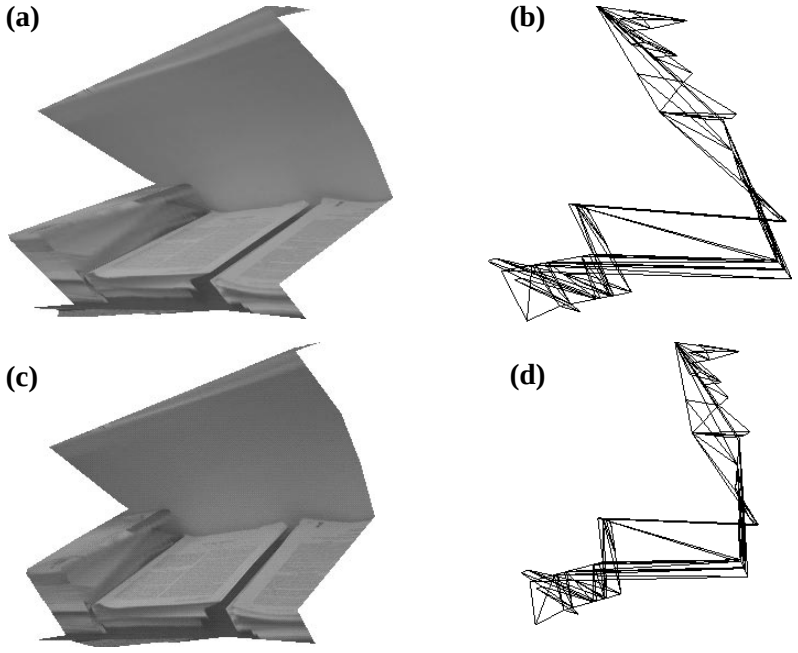


Figure 8: (a) Oblique appearance and (b) side surface mesh views of the simplified mesh shown on the right in Figure 6 with incorrect depth scale. (c) Oblique appearance and (d) side surface mesh views of the simplified mesh after correction for projective depth scale using world knowledge.

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 Conclusion

We have presented an algorithm for extracting concise models of scene shape from dense surface meshes created from real 3-D data sets that lends itself well to appearance mapping. We have demonstrated its use on projective depth maps and noisy omni-directional 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.

Our 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 the future we

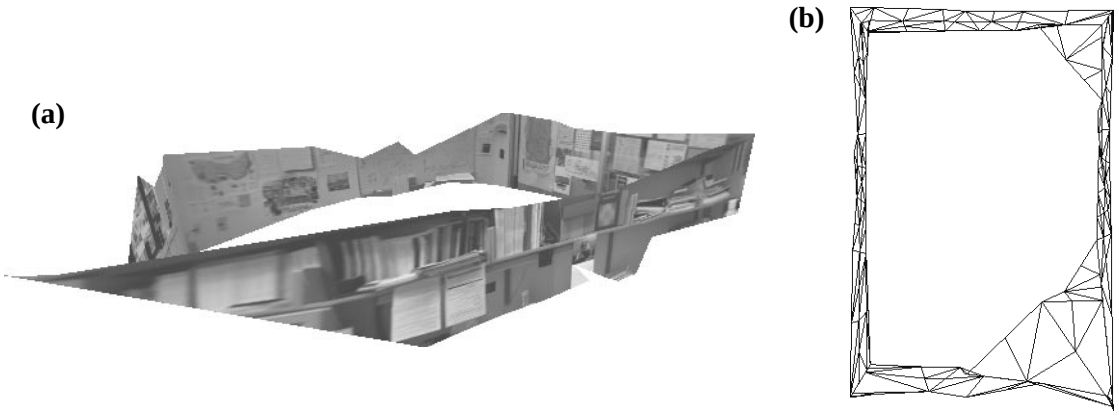


Figure 9: (a) A view of oblique appearance and (b) a top view of the simplified surface mesh from Figure 7 after planes have been made orthogonal by applying world knowledge.

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 the future, we will investigate the mapping of multiple images onto a single surface. This will require techniques for combining data from of overlapping images and eliminating view dependent effects (e.g., specularities) from the images.

References

[1] Shengchang Eric Chen. QuickTime VR - An Image Based Approach to Virtual Environment Navigation. *Computer Graphics, (SIGGRAPH '95 Proceedings)*. pp. 29-38, August 1995.

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

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

[4] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. *Computer Graphics, (SIGGRAPH '95 Proceedings)*. pp. 173-182, August 1995.

[5] Alexis Gourdon. Simplification of Irregular Surfaces Meshes in 3D Medical Images. *Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed '95)*. pp. 413-419, April 1995.

[6] Richard Hartley. In Defence of the 8-point Algorithm. *Fifth International Conference on Computer Vision (ICCV'95)*. pp. 1064-1070, 1995.

[7] M. Hebert, R. Hoffman, A. Johnson and J. Osborn. Sensor-Based Interior Modeling. *Proceedings of the American Nuclear Society 6th Topical Meeting on Robotics and Remote Systems*. pp. 731-737, February 1995.

[8] H. Hoppe, T. Derose, T. Duchamp, J. McDonald and W. Stuetzle. Mesh Optimization. *Computer Graphics, (SIGGRAPH '93 Proceedings)*. pp. 19-26, August 1993.

[9] A. Johnson, P. Leger, R. Hoffman, M. Hebert and J. Osborn. 3-D Object Modelling and Recognition for Telerobotic Manipulation. *Intelligent Robots and Systems, (IROS '95)*. pp. 103-110, 1995.

[10] S. B. Kang and R. Szeliski. *3-D Scene Data Recovery using Omnidirectional Multibaseline Stereo*. CRL-TR-95/6, Cambridge Research Lab., Digital Equipment Corp., Oct. 1995

[11] H. C. Longuet-Higgins. A Computer Algorithm for reconstructing a Scene from Two Projections. *Nature*, 293:133-135, 1981.

[12] John Lounsbery. *Multiresolution Analysis of Meshes of Arbitrary Topological Type*. Ph.D. Thesis, Department of Computer Science and Engineering, University of Washington, September 1994.

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

[14] W. Schroeder, J. Zarge and W. Lorensen. Decimation of Triangular Meshes. *Computer Graphics, (SIGGRAPH '92 Proceedings)*. pp. 65-70, July 1992.

[15] R. Szeliski and S. Kang. Direct Methods for Visual Scene Reconstruction. *Proceedings of IEEE Workshop on Representations of Visual Scenes*. Cambridge, MA. June 1995.

[16] Greg Turk. Re-Tiling Polygonal Surfaces. *Computer Graphics, (SIGGRAPH '92 Proceedings)*. pp. 55-64, July 1992.