Reconstruction of Scene Models from Sparse 3D Structure

Anastasios Manessis, Adrian Hilton, Phil Palmer, Phil McLauchlan and Xinquan Shen
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford GU25XH, UK

Abstract

In this paper we present a geometric theory for reconstruction of surface models from sparse 3D data captured from N camera views which are consistent with the data visibility. Sparse 3D measurements of real scenes are readily estimated from image sequences using structure-from-motion techniques. Currently there is no general method for reconstruction of 3D models of arbitrary scenes from sparse data. We introduce an algorithm for recursive integration of sparse 3D structure to obtain a consistent model. This algorithm is shown to converge to the real scene structure as the number of views increases and to have a computational cost which is linear in the number of views. Results are presented for real and synthetic image sequences which demonstrate correct reconstruction for scenes containing significant occlusions.

1 Introduction

An important problem in computer vision is the reconstruction of 3D models of complex rigid scenes from monocular image sequences. Recent research has focused on the development of structure-from-motion for automatic recovery of 3D shape by matching image structure between multiple frames. Feature based approaches have been widely developed for automatic recovery of sparse 3D structure based on automatic tracking of corners, lines and curves between consecutive images [13, 9]. A common goal of developing methods to estimate structure from image sequences is to reconstruct 3D scene models for visualisation. Current techniques for reconstruction of 3D models from sparse data are limited to simple planar scenes [6] or modelling of isolated objects [5].

Previous research aimed at constructing 3D models has addressed the problem of reconstruction from dense 3D surface measurements captured using active range sensors [2, 12, 3] or multi-baseline stereo [7]. Volumetric techniques have been widely used to achieve reliable reconstruction of complex objects [2] and environments [3, 11]. Methods for reconstruction from dense data assume that the distance between adjacent surface measurements can be used to estimate the local topology of the 3D surface. This assumption is not valid for interpolation of sparse 3D data.

Model reconstruction from sparse 3D data of arbitrary geometry scenes is an open problem. Faugeras et al. [4] addressed this problem using 3D Delaunay triangulation (tetrahedralisation) of a set of image features together with their visibility for each camera view to construct a volumetric model. The principal limitation of this approach is the assumption that the entire feature is visible which prohibits partial occlusion. Furthermore, this is a batch method which requires all the 3D structure prior to reconstruction.

Recently Kutulakos and Seitz [8] presented a general theory of N-view shape recovery. The principal assumption of their approach is that a locally computable consistency criteria is available to test point correspondence in multiple views. In image sequences of real-scenes such as indoor environments lack of surface texture will result in a reconstruction which deviates considerably from the real surface

In this paper we address the problem of reconstructing surface models from sparse 3D scene structure captured from N camera views. Our approach offers two principal contributions. Firstly, a theoretical framework is developed for a consistent surface representation to interpolate sparse 3D data. Secondly, we formulate an efficient recursive algorithm for reconstruction of a consistent surface model from sparse 3D data of an arbitrarily-shaped scene. The algorithm presented in this paper provides a unified approach to scene reconstruction from any available sparse or even dense 3D scene structure.

2 Reconstruction from Sparse Data

In this section we present a geometric theory for reconstruction of a surface model from sparse 3D data captured from N camera views. We then define a provably correct algorithm for reconstruction of a 3D scene model which is consistent with the data visibility constraints from N views. This algorithm is shown to converge to a correct reconstruction of the real surfaces

in the 3D scene as the number of views increases.

No prior assumptions are made concerning our knowledge of the structure of the 3D scene, scene illumination, surface properties, camera positions or the ability to separate the foreground and background.

2.1 Problem Statement

Given a set of sparse 3D features captured from N-views of an arbitrary unknown 3D scene, together with the feature visibility from each view, reconstruct a consistent 3D surface model. Here a scene feature is any image structure which can be matched across multiple views including points, lines and curves. A consistent representation is defined as:

> **Definition 1 (consistent representation):** A consistent 3D model is a set of surfaces which interpolate the space between the sparse 3D features and do not violate any of the constraints on feature visibility.

This defines a family of possible solutions which are all consistent reconstructions. This family of solutions is the most we can say about the true scene structure, given the sparse set of 3D features, without making prior assumptions on surface type. Any model, which is consistent is an equally valid approximation of the scene geometry and topology.

2.2 Consistent 3D scene models

An arbitrary real 3D scene can be represented by a set of surfaces $S = \{s_i\}_{i=1}^{N_s}$. This set of surfaces can be approximated to arbitrary precision by a set of planar triangular surface primitives $S \approx M_s = \{t_i\}_{i=1}^{N_{ts}}$. The use of a triangulated surface model to approximate the 3D scene does not require any prior assumptions on surface type although it does impact on representation efficiency for curved surfaces.

Given a set of sparse 3D features $F = \{f_i\}_{i=0}^{N_f}$ we can triangulate them to obtain a consistent model $M = \{t_i\}_{i=1}^{N_t}$ which satisfies Definition 1. Each feature f_i which is visible in the j^{th} view taken at position $\vec{v}_j$ defines a visibility constraint c_{ij} as follows:

> **Definition 2 (visibility constraint):** The space between the view position $\vec{v}_j$ and the scene feature f_i is not occupied by an (opaque) object.

This constraint for a point feature is a line, for a line feature is a triangle and for a curve feature is a 2-manifold surface (which can be approximated to arbitrary precision by a set of triangular surface primitives). Therefore, given a set of features F and their visibility from N views results in a set of visibility constraints $C = \{c_i\}_{i=0}^{N_c}$ (Figure 1).

A consistent representation M must satisfy the set of N-view visibility constraints, C, therefore we obtain

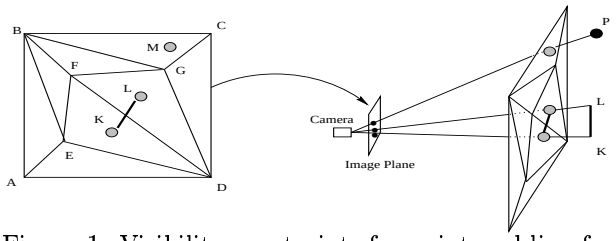

Figure 1: Visibility constraints for point and line features.

the following definitions:

> **Definition 3 (consistent triangle):** A triangle $t_i \in M$ is consistent if it does not intersect any of the visibility constraints in C.
>
> **Definition 4 (consistent model):** A model M is consistent if all triangles $t_i \in M$ are consistent.

2.3 Single-view Reconstruction

A consistent model for the set of visible features F_j in the j^{th} view can be constructed by a constrained triangulation in a plane orthogonal to the view direction. For a single view this projection is injective. The order of feature projections in the plane is preserved with respect to their relative ordering in 3D space from the view point $\vec{v}_j$.

> **Proposition 1:** Constrained triangulation of the features, F_j, in the 2D projection plane results in a model, M_j, which is consistent with the visibility constraints C_j.

Proof: Constrained triangulation of a set of features F_j in the 2D projection plane guarantees that no edge in the triangulation intersects a feature. The features form the vertices and edges of the triangulation. As the projection between 3-space and the 2D plane is injective the resulting 3D triangulated mesh M_j will not reorder the features. Therefore M_j does not contain any triangles which intersect the visibility constraints C_j and the resulting model is consistent. QED

2.4 N-view Reconstruction

In general for multiple views of a 3D scene there is no single 2D plane to which the scene features can be injectively projected without reordering of the features F. Reconstruction of a consistent model for two views i and j can be achieved by integrating consistent models from the two views M_i and M_j such that all triangles in the resulting model M_{ij} satisfy the union of the visibility constraints $C_{ij} = \{C_i \bigcup C_j\}$.

A general algorithm for reconstructing a consistent model of a 3D scene from N views can be achieved by the incremental integration of the model and constraints for each view. The general N-view algorithm can be stated as follows:

1. Initialise the global model and constraints as the empty set: $M = \{\emptyset\}$, $F = \{\emptyset\}$ and $C = \{\emptyset\}$.
2. Build a consistent model for the i^{th} view, M_i, by constrained triangulation of the visible features, F_i, in a plane orthogonal to the view direction.
3. Eliminate triangles in M which violate the viewpoints constraints for the i^{th} view C_i to give M' which is consistent with the combined constraints $C' = \{C \bigcup C_i\}$.
4. Eliminate triangles in M_i which violate the viewpoint constraints for the global model C to give a M_i' which is consistent with C'.
5. Integrate non-redundant triangles from M_i' into M' to give a consistent model $M'' = \{M' \bigcup M_i'\}$.
6. Update the global model $M = M''$ and constraints $C = C'$.
7. Repeat steps 2-6 for all N views to compute a global model M consistent with the union of feature visibility constraints from all views C.

It should be noted that the general N-view algorithm for constructing a consistent 3D scene representation is order independent. Although the algorithm is based on the incremental addition of new views the visibility constraints for all views are applied equally to each triangle in the model.

2.5 Proof of convergence

Given a model M which is consistent for the set of visible features F over N views we want to prove that as additional views are incorporated the model converges towards an approximation of the true scene surface S.

Proposition 2: If a model M is consistent over N views then as $N \rightarrow \infty$ the reconstructed model M approximates a subset of the real scene surfaces $M_\infty \subseteq S$.

Proof A consistent model M consists of two categories of triangles: (a) Real triangles which are planar approximations of real surfaces $M^r = \{t_i^r\}_{i=0}^{N_{tr}}$ where $M^r \subseteq S$; and (b) Virtual triangles which occur at occlusion boundaries $M^v = \{t_i^v\}_{i=0}^{N_{tv}}$. The union of real and virtual triangles is the model $M = \{M^r \bigcup M^v\}$. By definition the set of real triangles M^r correspond to (opaque) surfaces in the real scene S and therefore can not be intersected by a visibility constraint from any viewpoint. Therefore, this model is consistent for all possible viewpoints. Given a consistent model M for N views, applying the visibility constraints from a new view C_{N+1} will only result in elimination of virtual triangles from the set M^v. Virtual triangles will be eliminated provided one or more scene features is visible on the other side of the triangle. A virtual triangle for which no scene feature is visible on the other side from any view is a valid planar approximation of the real surface and therefore belongs to the set of real surfaces M^r. As the number of views increases all virtual triangles will be eliminated $M_\infty^v \rightarrow \{\emptyset\}$. The resulting model M_∞ will converge to the set of triangles which correspond to real scene surfaces $M_\infty \rightarrow M^r \subseteq S$. QED

3 Real Scene Reconstruction

The goal of our work is to develop an automatic system for scene reconstruction from image sequences. In this section we present the algorithm developed for scene reconstruction from a sequence of images.

Images are captured using a camera mounted on an autonomous mobile robot platform. This system captures a sequence of images of an indoor scene with approximately known camera positions. A recursive structure-from-motion (SFM) algorithm [9] is applied to estimate the 3D location of the sparse scene features together with the camera position and orientation. A sparse feature based SFM algorithm has been used for computational efficiency of reconstruction for long image sequences. Point and line features are used in this work although the approach also extends to higher order features. The SFM algorithm incorporates constraints between features such as co-planarity and surface perpendicularity to increase reconstruction accuracy if information on feature groupings is available. Further details of this system are provide in [10].

3.1 Algorithm Overview

The algorithm developed for surface reconstruction from sparse 3D data is based on recursive integration of the set of feature data, F_i, and feature visibility constraints, C_i, for each new camera view. The algorithm recursively integrates visibility constraints from new views into the current global model M to reconstruct a model which converges to the real scene structure as the number of views increases.

In practice the computational cost of implementing the algorithm presented in section 2.4 is prohibitively expensive. The worst-case computational complexity of testing visibility constraints for all features over all N-views is $O(N_f^2 N^2)$ where N_f is the number of features. The cost increases with the square of the number of views. Therefore, we adopt the approach of only testing the visibility constraints for the new view C_i against the current global model M.

In section 3.3 we show that for a closed-system this approach converges to an approximation of the real scene surfaces. The worst-case computational complexity becomes $O(N_f^2 N)$. However, practically N_f is

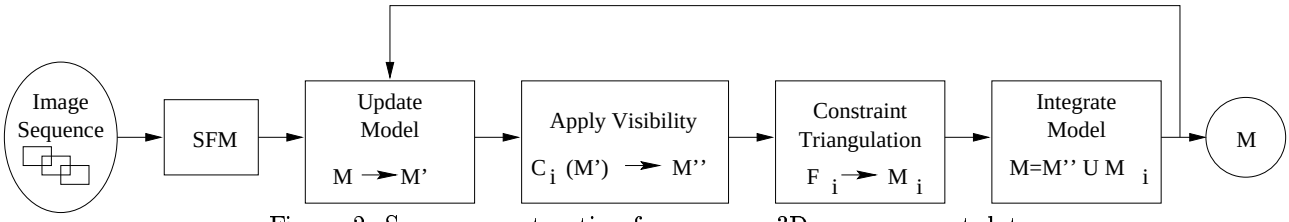

Figure 2: Scene reconstruction from sparse 3D measurement data

the number of visible features which is considerably smaller than the total number of features. Recursive reconstruction of a new global model at the i^{th} view is performed in three steps:

1. **Update feature position:** All 3D positions of features F in the global model M are updated based on the new SFM estimates resulting in M'.
2. **Apply visibility constraint:** Visibility constraints, C_i , for features F_i visible in the i^{th} view are applied to the global model M' producing M''.
3. **Integrate new features:** Non-redundant triangles for the i^{th} view model, M_i, for features not in the global model, $F_i \notin F$, are integrated into the global $M = M_i \bigcup M''$.

3.2 Algorithm Convergence

In this section we show that for a closed-scene the algorithm proposed in the previous section converges to a reconstruction of the real scene surfaces as the number of views increases. A closed-scene has a finite set of scene features F whose 3D location can be reconstructed using the SFM algorithm.

Proposition 3: For a closed-scene with features F the model M reconstructed by only applying visibility constraints for each new views C_i converges to a subset of the real scene surfaces as the number of views increases $M_\infty \subseteq S$

Proof: If the system is closed then as the number of views increases the set of new features F_{ni} visible in each frame converges to zero, $F_{ni} \notin F = \{\emptyset\}$. Consequently the set of non-redundant triangles in each new frame converges to zero and no new triangles are added to the model. Each new view results in a new set of visibility constraints, C_i, for features previously incorporated to the model $F_{pi} \in F$ $(F_{ni} \bigcup F_{pi} = F_i)$ Following the proof of Proposition 2, the i^{th} view visibility constraints, C_i, will eliminate only virtual triangles $M^v \in M$. As the number of views increase the global model will converge to the set of triangles approximating real surfaces $M_\infty = M^r \subseteq S$. QED

3.3 Constrained Triangulation

A constrained triangulation [1] of the visible point and line features F_i in the image plane is performed for each camera view. The algorithm initially performs an unconstrained Delaunay triangulation of the point features and line end-points. A recursive edge-swapping algorithm between adjacent pairs of triangles is then applied to generate a triangulation where the line features are incorporated as triangle edges. Implementation of this algorithm is approximately $O(N_f log N_f)$ computational cost in the number of features N_f. The 2D triangulation can then be projected into 3D space using the estimated 3D feature positions to generate a model M_i for the i^{th} view. This results in a reconstruction which is consistent with the i^{th} view feature visibility constraints, C_i, as shown in 2.3.

This is the only set of possible real surface patches visible in the i^{th} view which will be integrated to the global model. All triangles which are between new image features F_{ni} and image features previously incorporated in the global model, F_{pi}, are candidate non-redundant surfaces. All other triangles, which only connect features in F_{pi}, are not integrated as they may have been eliminated previously by visibility constraints for previous frames. This prohibits triangulation between two features which have both previously been incorporated in the global model but have not previously been visible in the same frame.

3.4 Visibility Constraints

Implementation of visibility constraints requires testing the intersection of the triangles in the global model, M, with all of the visibility constraints, C_i, for each camera viewpoint. In practice the computational cost can be dramatically reduced by partitioning the space. This is achieved by projecting the visible part of the global model to the current image plane for the i^{th} view using a bucketing structure [4].

For a point feature the projection of the point in the 2D-image plane must be inside the projection of the model triangle. This reduces the cost of performing this operation to a simple lookup operation. For a line feature the projection of the line in the 2D image plane must lie inside the triangle (Figure 1). Each overlapping triangle in the global model is then tested for 3D intersection with the triangle which is formed by the camera position and the 3D line feature.

4 Results

In this section we present results of applying the incremental reconstruction algorithm to sparse 3D structure from real and synthetic image sequences. The sequences are from simple 3D scenes which contain multiple objects such that the entire scene is not visible from a single viewpoint. Automatic feature tracking is used to match corner-points and edge-lines between consecutive images. Manual labelling is applied to group scene features on the same surface and identify surface perpendicularity. Recursive structure-from-motion with surface constraints is applied to estimate the 3D location of scene features and camera positions [10].

Results for a synthetic image sequence of 25 images are shown in Figure 3. The sequence is of two cubes one in front of the other and three perpendicular planes behind. In the initial frame the smaller cube is completely occluded and as the camera moves from left to right becomes visible. The scene contains 9 visible surfaces which are approximated by 96 triangles in the resulting model. The reconstructed models clearly approximates the real scene where multiple occluded surfaces exist.

Results for a real-image sequence of 10 frames are shown in Figure 4. The scene is of the corner of a room with several occluding objects in the scene. Recursive reconstruction using the feature visibility constraints removes surfaces which violate the scene visibility. The reconstructed model contains 15 real surfaces which are represented by 244 triangles. Some artifacts are visible in the texture mapped model when viewed from novel directions for regions which are occluded in the input image sequence. Results illustrates the recursive reconstruction of models that approximate real scene surfaces with significant occlusions.

5 Conclusions

In this paper we have presented a general geometric theory for scene reconstruction from sparse 3D data captured from N-views. A consistent model is defined as a set of surfaces which interpolate the sparse 3D data and do not violate the feature visibility. We have introduced an efficient recursive algorithm for reconstruction of a consistent model by incremental integration of new 3D data and visibility into a global model. This algorithm is shown to converge to the set of real scene surfaces as the number of views increases. Except for surface opacity no prior assumptions are made about our knowledge of the scene geometry, surface properties or camera positions.

Sparse 3D data is commonly reconstructed from monocular image sequences using structure-from-motion techniques. Results are presented for reconstruction of 3D models from real and synthetic image sequences of simple scenes containing multiple occluding objects that demonstrate the feasibility of our approach. This is the first algorithm addressing scene reconstruction from sparse 3D data with arbitrary geometry and multiple occluding objects.

References

[1] E. Bruzzone, M. Cazzanti, L. De Floriani, and F. Mangili. Applying two-dimensional delaunay triangulation to stereo data interpolation. In *ECCV*, pages 368–372, 1992.

[2] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[3] S. El-Hakim, C. Brenner, and G. Roth. An approach to create virtual environments using range and texture. In *ISPRS International Symposium of Real Time Imaging*, pages 331–338, June 1998.

[4] O.D. Faugeras, E. Le Bras-Mehlman, and J.D. Boissonnat. Representing stereo data with the delaunay triangulation. *Artificial Intelligence*, 44:41–87, 1990.

[5] A. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3d model construction for turn-table sequences. In R. Koch and L. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments, LN CS 1506*, pages 155–170. Springer-Verlag, 1998.

[6] A. Fitzgibbon and A. Zisserman. Automatic 3d model acquisition and generation of new images from video sequences. In *European Signal Processing Conference*, pages 1261–1269, 1998.

[7] S.B. Kang and R. Szeliski. 3-d scene data recovery using omnidirectional multibaline stereo. In *CVPR*, pages 364–370, 1996.

[8] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *ICCV*, pages 307–314, 1999.

[9] P. McLauchlan and D. Murray. A unifing framework for structure and motion recovery from image sequences. In *ECCV*, pages 314–320, 1995.

[10] P. McLauchlan, X. Shen, A. Manessis, P. Palmer, and A. Hilton. Surface-based structure-from-motion using feature groupings. In *ACCV*, Jan 2000.

[11] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *IEEE Computer*, 22(6):85–90, 1989.

[12] V. Sequeira, K. Ng, E. Wolfart, J.G.M. Gonalves, and D. Hogg. Automated reconstruction of 3d models from real environments. *Photogrammetry and Remote Sensing*, 1999.

[13] P. Torr, A.W. Fitzgibbon, and A. Zisserman. Mainting multiple motion model hypotheses over many views to recover matching and structure. In *ICCV*, pages 485–491, January 1998.

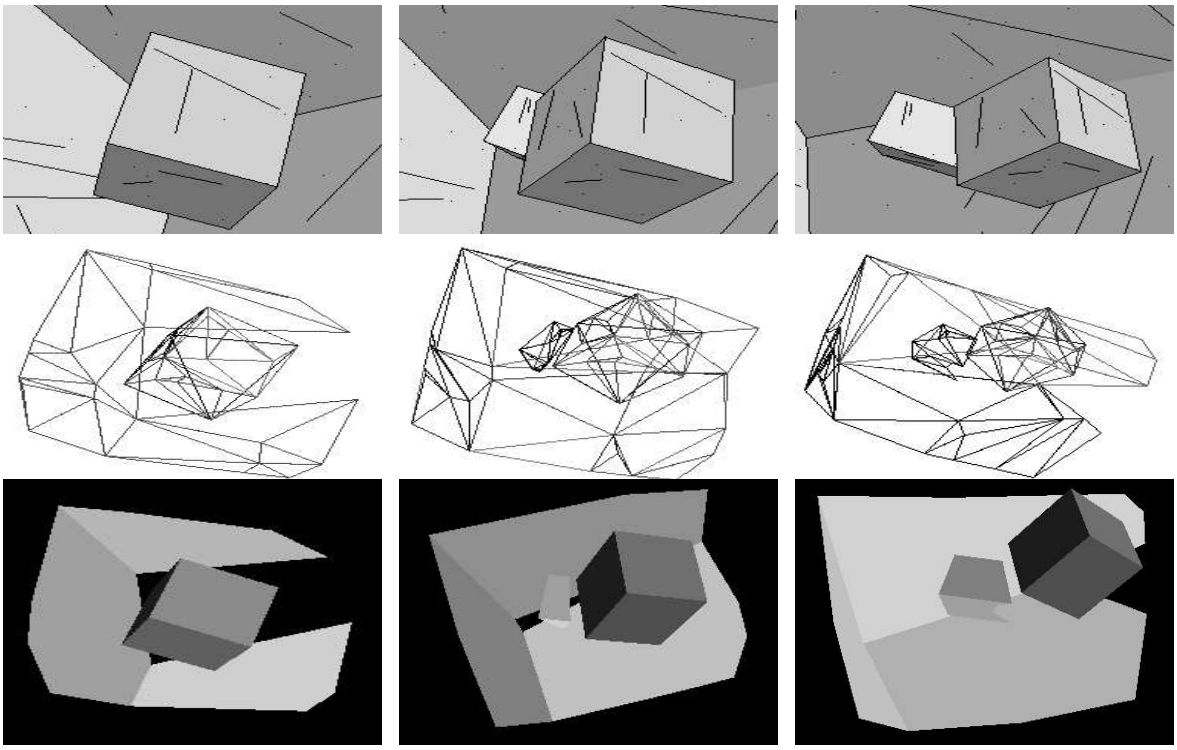

Figure 3: Reconstruction of synthetic sequence. Frames 0,10 and 25 of the original images, together with the corresponding wireframe and VRML models.

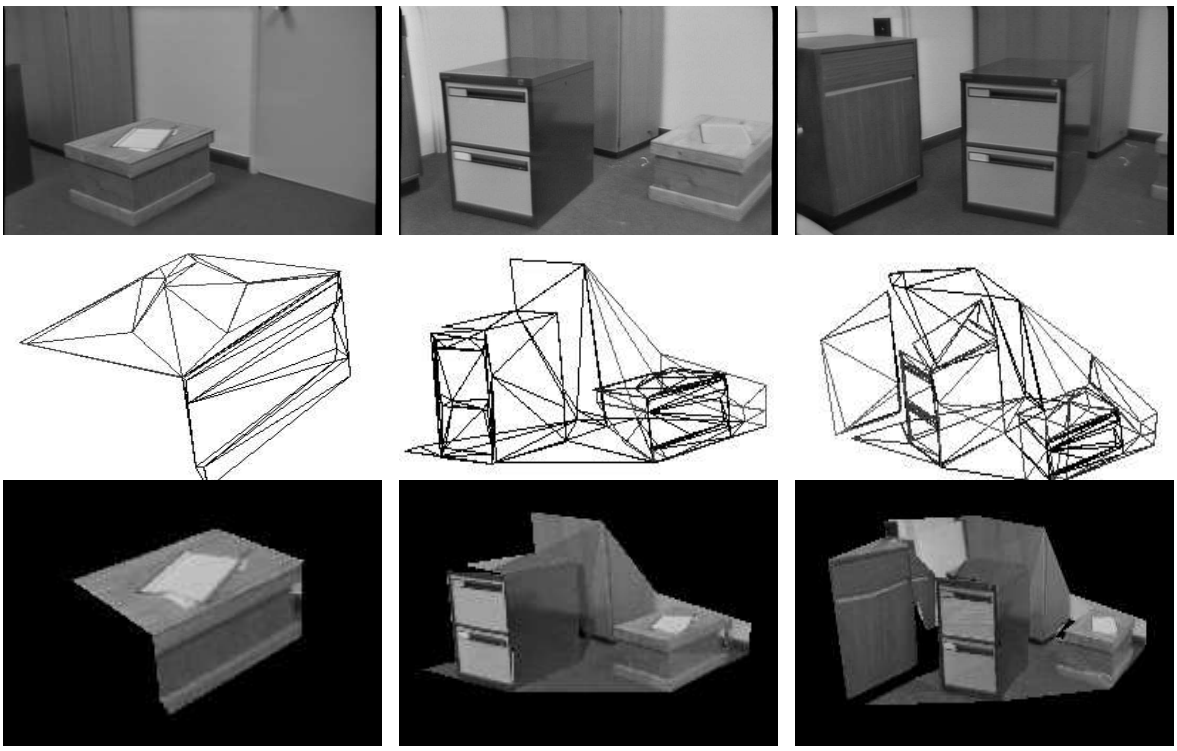

Figure 4: Frames 0,6 and 10 of room sequence along with reconstructed models in wireframe and VRML.