The 11th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2010)
A. Artusi, M. Joly-Parvex, G. Lucet, A. Ribes, and D. Pitzalis (Editors)

*Short and Project Papers*

# Using Abstract Model Representations to Complete Three Dimensional Scans of Architectural Space

Chen Xu and Holly Rushmeier, Yale University

**Abstract**

*Three dimensional scanners can be used to create models of architectural spaces. The resulting models often have missing elements because it is impossible to place the scanner in positions to view all surfaces. Missing elements can be modeled by simplified shapes derived from drawings or sparse linear measurements. Furthermore, in architecture spaces, there are often multiple instances of objects that can be combined to improve the object model. We build on previous work to represent models abstractly as graphs of relationships between primitive shapes such as planes, cylinders, and spheres. We present an automatic approach to search for incomplete instances of objects using abstract shape representations of both simple models and of the large detailed point cloud. The simple models can be used to fill in the missing objects, and the partially scanned portions of multiple object instances can be combined to refine the model. We demonstrate this approach on a 3D scan of a historic synagogue.*

## 1. Introduction

Models of indoor architectural spaces created using laser scanners have been used in many cultural heritage applications (e.g. [WHL*]). Laser scans of indoor spaces have two disadvantages – they contain too much data, and too little. Too much data is collected in that relatively large simple shapes, such as walls, are substantially over-sampled. Too little data is collected in that there are generally many objects such as furniture and support columns that make it impossible for the laser to see parts of the environment. In this work we build on methods created to reduce oversampling to enable filling in missing data with the aid of simplified models obtained with alternative measurement techniques.

Our motivating project is the modeling of the interior of a historic synagogue. Photographs of the space are shown in Fig. 1, and the scanned geometry is shown on Fig. 2. A number of factors prevented capture of the full geometry – primarily occlusions in the environment and equipment malfunctions. Many elements in the scene that act as occluders or are occluded are repeated elements – such as the benches and windows. Simple models of these elements can be obtained using manual linear measurements, as described in [RXW*07]. Our goal is to locate where instances of such a simple model should be in the full scan, and merge the simple model with the scanned details that are available.

The two major areas of previous work we build on are the identification of objects and primitives in point clouds (e.g. [GKF09]) and scene completion by exploiting detected symmetry [PMG*05, PMW*08]. In particular, we follow the work presented in a series of papers by Schnabel et al. [SWK07, SWWK07, SDK09, SWWK08]. Schnabel et al. use the RANSAC algorithm [BF81] to locate 3D primitives in point clouds. Then, they represent the scene as a graph of primitives – with each graph node being a primitive and the edges between nodes representing the geometric relationship between the primitives. Individual objects can also be represented as graphs of primitives. In [SWWK08] the ability to locate an object in a scene is illustrated as seeking the object subgraph within the full scene graph.

In our work-in-progress we seek extend the work of Schnabel et al. in two ways. First, we are using the graph representation of a simplified object model to find *incomplete* instances of the object in the full scene graph. Second, we seek to use the incomplete object instances in the full scene to enrich our original simplified object model.

## 2. Proposed Method

Figure 3 is the overview of our proposed method. First we represent the unstructured point cloud as a topology graph of primitive shapes and relations among them as in

DOI: 10.2312/PE/VAST/VAST10S/001-004

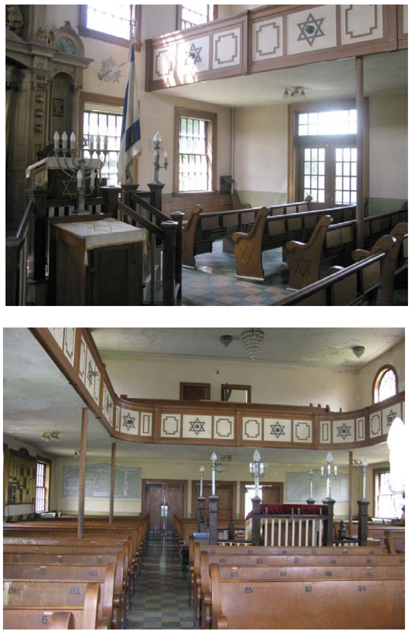

**Figure 1:** *The interior of a historic synagogue: the top image shows a view looking forward, the lower image shows the view to the back. Many surfaces are occluded by furniture and support elements.*

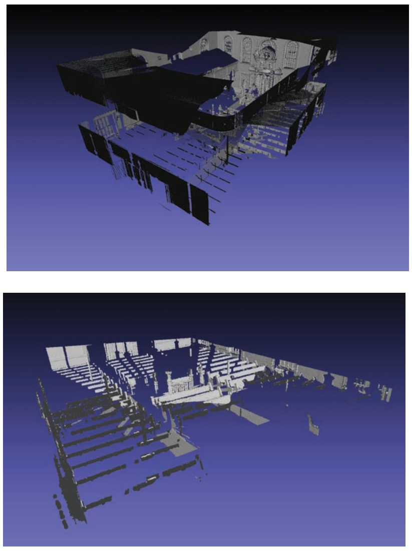

**Figure 2:** *Data obtained from 3D scanning of the synagogue interior: the top image shows the full data set looking forward, the lower image is the view to the back that shows the portions of benches that were captured by scanning.*

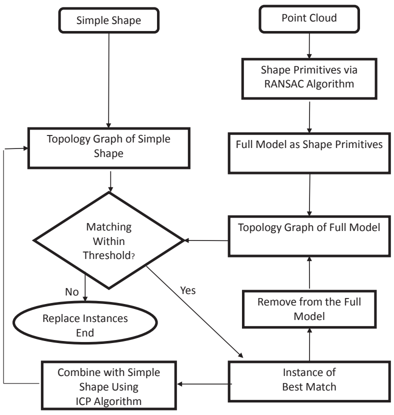
Simple Shape
Point Cloud
Shape Primitives via RANSAC Algorithm
Topology Graph of Simple Shape
Full Model as Shape Primitives
Matching Within Threshold?
Topology Graph of Full Model
No
Yes
Remove from the Full Model
Replace Instances End
Combine with Simple Shape Using ICP Algorithm
Instance of Best Match

**Figure 3:** *Overview of our proposed method for improving the scanned model.*

[SWWK07]. The simple shape of a repeated object (such as one of the benches) is also represented in this way, so that we can search for possibly incomplete instances using subgraph matching. In each loop we locate the best match or several matches within some predefined threshold, and those instances are aligned with the current abstract shape using the ICP (Iterative Closest Point) algorithm [RL01]. When there are no more instances found, we end the process by replacing all instances found with the integrated and detailed shape.

**Topology Graphs** In this paper, we call shapes such as planes, spheres, cylinders, and so on primitive shapes. They are atomic elements of the topology graph representation and can be determined by a few parameters. Figure 4 is an example of a simple model of one of the benches represented by the abstract topology graph. Each vertex in the topology graph is a primitive shape. The parameters of each primitive shape are fixed, and the variables are defined to be within a range. For example, a vertex in the graph can be a plane

$$ax + by + cz + d = 0,$$

where parameters a, b, c, and d are fixed, and $x \in [x_1, x_2]$, $y \in [y_1, y_2]$, $z \in [z_1, z_2]$. Alternativel, a cylinder is:

$$r \cdot (\mathbf{v_1} \cdot \sin\theta + \mathbf{v_2} \cdot \cos\theta) + h \cdot \mathbf{v_0} + \mathbf{v},$$

where r is the fixed radius, $\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v}$ are fixed vectors of the axes, along with two other elements of an orthogonal basis, and $\theta \in [\theta_1, \theta_2], h \in [h_1, h_2]$. Two vertices within a user-defined distance are connected by an edge, which is associated with a value. For two planes, the value will be the relative angle between their surface normals; and for two parallel planes, the value will be the distance between them; and for one plane and one cylinder, the value is the relative angle between the normal of the plane and the axes of the cylinder. The shapes can be generated in various ways. They may be derived manually from blueprints or hand drawings, or derived from the mesh using some abstraction method such as [MZL*09]. We use the approach in [RXW*07]. Feature

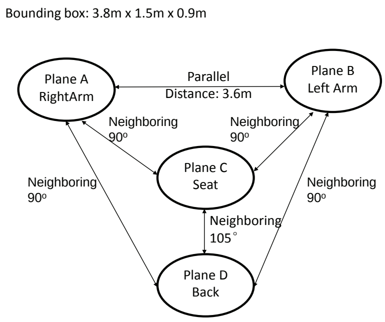

**Figure 4:** *Example of a simple model of a bench represented as an abstract topology graph.*

points are marked on the object and pairwise distances are measured so that a simple polygonal shape can be created using multidimensional scaling on the pairwise distances. The abstract topology graph is naturally derived from the simple polygonal model.

**Primitive shapes from the point cloud** We use the algorithm presented in [SWK07] to decompose the unstructured point cloud to planes, cylinders, spheres, cones and tori. These primitive shapes are organized into a topology graph similar that described above, but with a bitmap associated with each primitive shape. The bitmap facilitates defining holes in the primitives, such as window openings in planes representing walls.

**Subgraph matching** The subgraph isomorphism problem is a task in which two graphs G and H are given as input, and one must determine whether G contains a subgraph that is isomorphic to H. Subgraph isomorphism is a generalization of both the maximum clique problem and the problem of testing whether a graph contains a Hamiltonian cycle, and is therefore NP-complete. Although the subgraph isomorphism problem is NP-complete, the practical problem we are working on can be solved within polynomial, or even linear, time because we can exploit constraints in our scene. Constraints we exploit include only considering nodes within a certain bounding box each time, and noting that many instances of interest are oriented with the same up direction.

The algorithm we use is similar to the subgraph matching algorithm in [SWWK07], but adapted to search for incomplete instances. For every node in the abstract shape, our algorithm tries to find its match in the full scene graph. If no such matched node is found, it will be marked as “no_match" and continue to next node. We assume the node with largest area always has a match in the full scene graph, and the algorithm starts with this node. We use a stack data structure to store current matched nodes. Edges are only rejected when there is an inconsistency. Whenever a transformation matrix can be determined, the consistency between the abstract shape and the point cloud will be evaluated and the score is recorded if it is the current best match.

**Instance integration** Once instances are located, we seek to merge them to form an improved model of the original simple shape. We make use of the graph matching to bring the instances into the same coordinate system. For example, suppose there are three planes $q_1$, $q_2$ and $q_3$ in the graph for the simple shape, whose non-coplanar normals are $\mathbf{n_1}$, $\mathbf{n_2}$ and $\mathbf{n_3}$ respectively. The counterparts of these planes in the point cloud model are $p_1$, $p_2$ and $p_3$, whose normals are $\mathbf{m_1}$, $\mathbf{m_2}$ and $\mathbf{m_3}$. The rotation matrix is determined by these 6 normals, and the translation is determined by the insection of $q_1$, $q_2$, $q_3$, and the insection of $p_1$, $p_2$, $p_3$. Because we know the transformation matrix from each instance to the simple shape, all of the instances can be aligned together based. The alignment is not perfect, so we plan to use the ICP algorithm to refine alignment.

## 3. Results

We first show a small test case to illustrate the process. Figure 5a shows the scan of a small birdhouse. As shown Fig. 5b, in the full model was decomposed into 116 primitive shapes (shown with bounding boxes here). We created a simple search shape for a chimney as 8 planes, as shown in Fig. 5c. Two instances of the chimney are found in the full model, with the primitives for one found instance shown in Fig. 5d. Both instances found are marked green in the original model in Fig. 5. As shown in Fig. 5d, although both instances found are nearly complete in the original model, their abstract representations are incomplete, because some parts of the chimneys are included in larger primitive shapes during the RANSAC processing.

Figure 6 shows results for the synagogue model. The dimensions are 30m x 12m x 8m, and the point cloud has 4 million points.The primitives found in the cloud are shown in Fig. 6a. The shape we seek to refine is the bench. We built a model for the bench, shown in Fig. 6b, using the approach in [RXW*07]. The primitives found for the bench are shown in Fig. 6c (resulting in a slightly more complex topology graph than in Fig. 4). Using the matching algorithm, seven instances of the bench were found. The instances are marked in different colors in the original model Fig. 6d, and are shown by themselves in Fig. 6e. In this case, the alignment provided by the matching algorithm was not adequate to immediately start the ICP in a configuration that converged to a single model. Our work is continuing to improve this final alignment.

**Conclusion** We have described an automated method to identify incomplete instances in an unstructured point cloud using abstract shapes. Both the point cloud and the abstract shape are represented by primitive shapes such as planes, cylinders, spheres, and so on, along with relationships among these primitives. Instances are identified using

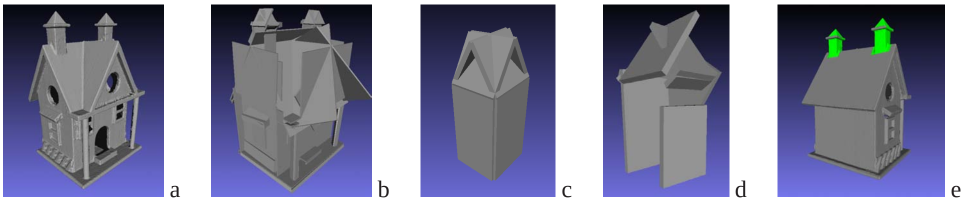

**Figure 5:** *Result: the wooden house example. Top-left: original model. Top-right: primitive shapes, 116 in total. Bottom-left: abstract shape and one instance found of chimney. Bottom-right: instances marked in the original model.*

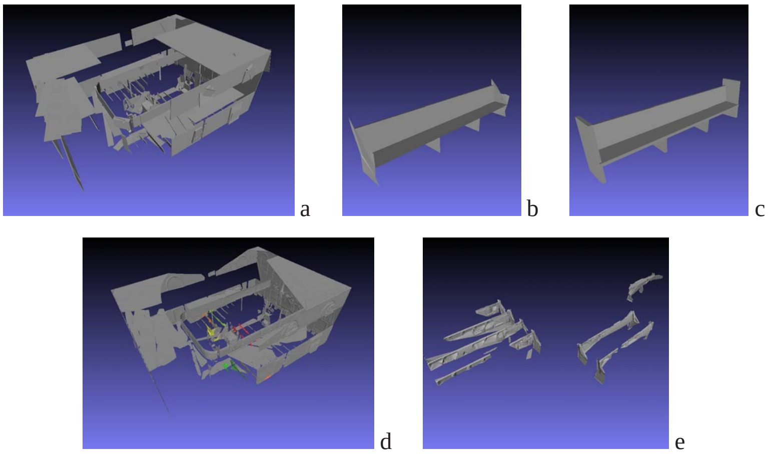

**Figure 6:** *Result: the Synagogue example. Top-left: original model. Top-right: primitive shapes, 405 in total. Bottom-left: shape from simple tools method, more abstract shape, instances found. Bottom-right: instances marked in the original model.*

subgraph matching algorithm. We are working on using instances found to refine initial abstract shape both to search for more incomplete instances, and to eventually replace all instances with a full detailed shape.

## References

[BF81] Bolles R., Fischler M.: A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *International Joint Conference on Artificial Intelligence* (1981), pp. 637–643.

[GKF09] Golovinskiy A., Kim V. G., Funkhouser T.: Shape-based recognition of 3d point clouds in urban environments. *ICCV* (2009).

[MZL*09] Mehra R., Zhou Q., Long J., Sheffer A., Gooch A. A., Mitra N. J.: Abstraction of man-made shapes. *ACM Trans. Graph 28*, 5 (2009).

[PMG*05] Pauly M., Mitra N. J., Giesen J., Guibas L., Gross M.: Example-based 3d scan completion. In *EUROGRAPHICS SYMPOSIUM ON GEOMETRY PROCESSING* (2005), Eurographics Association, pp. 23–32.

[PMW*08] Pauly M., Mitra N. J., Wallner J., Pottmann H., Guibas L. J.: Discovering structural regularity in 3D geometry. *ACM Trans. Graph 27*, 3 (2008).

[RL01] Rusinkiewicz S., Levoy M.: Efficient variants of the icp algorithm. In *INTERNATIONAL CONFERENCE ON 3-D DIGITAL IMAGING AND MODELING* (2001).

[RXW*07] Rushmeier H., Xu C., Wang B., Rushmeier R., Dorsey J.: Shape capture assisted by traditional tools. In *VAST07: The 8th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage* (Brighton, UK, 2007), Arnold D., Niccolucci F., Chalmers A., (Eds.), Eurographics Association, pp. 1–8.

[SDK09] Schnabel R., Degener P., Klein R.: Completion and reconstruction with primitive shapes. *Comput. Graph. Forum 28*, 2 (2009), 503–512.

[SWK07] Schnabel R., Wahl R., Klein R.: Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum 26*, 2 (2007), 214–226.

[SWWK07] Schnabel R., Wahl R., Wessel R., Klein R.: *Shape Recognition in 3D Point Clouds*. Tech. Rep. CG-2007-1, Universität Bonn, May 2007.

[SWWK08] Schnabel R., Wessel R., Wahl R., Klein R.: Shape recognition in 3d point-clouds. In *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008* (Feb. 2008), Skala V., (Ed.), UNION Agency-Science Press.

[WHL*] Williams N., Hantak C., Low K., Thomas J., Keller K., Nyland L., Luebke D., Lastra A.: Monticello through the window. In *Proceedings of the 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST 2003), Brighton, UK (November 2003)*.