

Real-time Spatial Relationship Based 3D Scene Composition of Unknown Objects

Lee Seversky*

State University of New York at Binghamton

Lijun Yin†

State University of New York at Binghamton

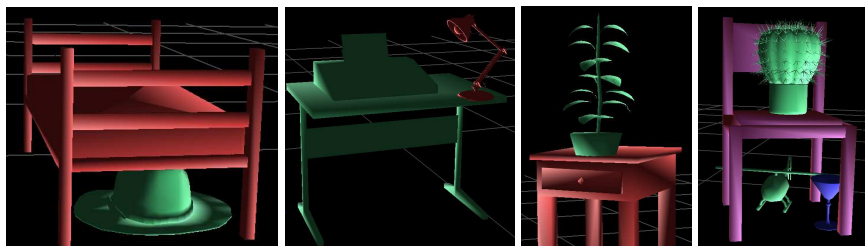


Figure 1: Automatic placement of objects using voice and text input. From left to right: The green hat is under the red bed. The red lamp is on the green desk. The green plant is on the red table. The green plant is on the purple chair and the blue cup is under the purple chair and the green helicopter is under the purple chair.

1 Introduction

Manual scene composition in 3D is a difficult task and existing approaches attempt to construct scenes automatically [Coyne2001][Xu2002]. These methods depend heavily on explicit per object knowledge that is used to determine placement. We present a method for automatically generating 3D scenes composed of unknown objects in real-time. Our method does not require any a priori knowledge of the objects and therefore the objects are considered to be unknown to our system. All necessary information is computed from the object’s geometric representation and is designed to support varying qualities of polygon models. The use of spatial relationships and relative positioning of objects is a natural and effective way for scene composition. Our method composes scenes by computing object placements that satisfy a desired spatial relationship such as on, under, next to, above, below, in front of, behind and to the left or right of. To illustrate our placement algorithm and its ability to be used interactively, a real-time scene composition framework using text and voice natural language input is developed.

2 Algorithm

Given a relationship type γ and two objects, O1 and O2, represented by a standard mesh model structure, the placement of the two objects is calculated as follows:

- (1) Bounding box localization: compute the object’s bounding box from the center of the object.
- (2) Object mesh voxelization: convert each mesh into a normalized voxel representation, with a resolution of 128x128x128 voxels.
- (3) Based on the pre-defined relationship γ of the two objects, extract cross-sectional planes of each object from the voxel representation.
- (4) Object surface extraction: for each object, extract voxels positioned on the surface from the cross-sectional planes. The relationship type γ defines which areas of the object’s surface to extract.
- (5) Determination of object placement: placement is found for object O2 with respect to O1 by comparing the surface layers in each set according to γ . Two conditions must be met for a position to be valid: a) The maximum number of voxel collisions of any adjoining surface must be less than or equal to a threshold (i.e. 0). b) Layers of O2 that reside in the bounding volume of O1 or any other object already in the scene must not collide.
- (6) Upon satisfying both conditions, convert the calculated position of O2 relative to O1 from a voxel to a world coordinate space. The

*e-mail: lsevers0@binghamton.edu

†e-mail: lijun@cs.binghamton.edu

composed objects (e.g. polygon models) are then added to the scene.

3 Framework

Our current application framework composes scenes using spatial relationships that define the relative placement of an object in terms of other objects. We use a speech recognition engine and a natural language parser to parse the voice and text input. We currently extract the object names, relationships, and a set of optional attributes (i.e. colors). Our primary focus of this initial implementation is to show the contribution of our algorithm within an automatic scene placement system and to illustrate its ability to be used in a real-time environment. Figure 2 shows an overview of our framework.

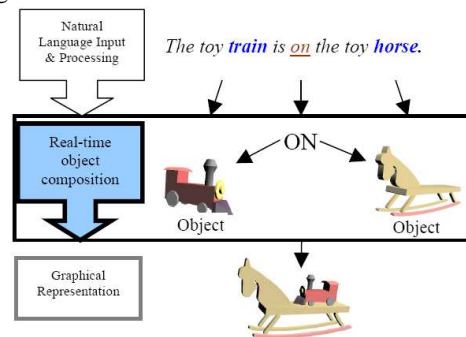


Figure 2: Real-time automatic scene composition framework using our placement algorithm for composition (outlined)

4 Conclusion

Our method for scene composition uses a unique approach that determines an object’s placement according to a desired spatial relationship. It is robust to different model qualities and we show that it can be used in a real-time application for constructing scenes using natural language voice or text input. Future work includes increasing realism and accuracy by adding support for different object orientations and improving collision detection between objects.

Acknowledgments: Thanks to Jason Moore at AFRL for his suggestions and continued guidance.

References

- B COYNE, R. S. 2001. Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 487–496.
- KEN XU, JAMES STEWART, W. F. 2002. Constraint-based automatic placement for scene composition. In *Graphics Interface*, 25–34.