

# Constraint Based 3D Scene Construction

Tim Salzman (M.Sc. Student), Graham Smith (M.Sc. Student),  
and Wolfgang Stuerzlinger (Supervisor)  
York University Department of Computer Science

## Abstract

Realistic 3D scenes often contain thousands of objects, and are in general difficult to model using today's CAD programs. We address this problem by exploiting predefined semantic relationships between objects to dynamically group and constrain them [GS99]. Virtual constraints, which de-couple the object's geometry from its constraints, are presented. Also discussed are techniques for dynamically grouping and re-grouping objects based on their semantic and virtual constraints. Preliminary testing shows that our system provides a fast, intuitive user interface for 3D scene construction. Finally, we will present ideas for future work.

Positioning objects in a virtual environment is not an intuitive task. One approach is direct manipulation using a six degree of freedom (DOF) input device. However, it is hard to position objects precisely using these devices, users often become fatigued, and many object interactions can be more effectively accomplished using a simple 2D interface such as a mouse [HPGK+94]. For these reasons, much research has been focused on software techniques for manipulating 3D objects using standard 2D input devices.

Bier [Bi90] presented snap-dragging for interactive solid modeling systems. This system used a general purpose gravity function and alignment manifolds to position objects in the scene. Snap-dragging provided easy selection of object features but the system required a complex user interface, had an unchangeable view and was computationally intensive.

In Object Associations [BS95], Bukowski and Sequin use a combination of physical properties (pseudo gravity) and goal oriented behaviour (alignment) to position and manipulate objects in a scene. This system was used to model the Berkeley Soda Hall WALKTRHU environment, which contained thousands of objects. However, adding new objects to the library is difficult, as object association must be coded. Constrained objects do not always move together because there is no dynamic grouping, instead the scene is searched for associated objects each time an object is moved. Further, all associations in this system are limited by object geometry.

Gosele exploited *natural* object behaviour to define and maintain object constraints [Go99]. Polygons were used to define *offer* and *constraint* areas, and a hierarchical labelling system was used to define which polygons could be constrained together. Typical constraint labels were on-Floor, on-Wall, and on-Workspace. Collision detection was added to the system to add realism by preventing inter-object penetrations. However, once constrained, objects in this system could not be unconstrained. Also, multiple constraints between two-objects were not possible.

Our system improves upon previous work in the following ways. Firstly, we allow constraints to be broken by *pulling* away from the constraining surface. For example, an object constrained to the floor and the wall can be unconstrained from the wall by pulling the object away from the wall. Also, an object can be re-constrained by translating it to another acceptable offer area. To show the user which offer areas will accept an object, all acceptable offer areas are highlighted when an object is selected for translation. Unconstraining and re-constraining an object invokes the dynamic grouping mechanism, which maintains the constraint relationships between objects.

We present the notion of virtual constraints. A virtual-constraint can be any polygon in 3D object-space, not necessarily associated with the object's geometry. An example is a polygon somewhere beneath a table, to which the front of a chair may be constrained. We present negative constraints, which are a specialization of virtual constraints. Negative constraints are useful for defining volumes of space in which certain objects should not be placed. For example a desk would not be placed against a doorway because it would make the door inaccessible.

Our system has optimized the constraint satisfaction algorithm by pruning the search for valid offer areas and using a minimal distance constraint, typically the bounding radius of an object.

Finally, we present *dual-constraints*, bi-directional constraints between objects. When two or more objects are constrained by dual constraints, they form a dual-group. Using a drag-add technique, a row of constrained, neatly aligned books can be created on a bookshelf with one interaction step. We introduce the push-pull metaphor for interacting with a dual-group. When a dual-group is selected for translation, a connected-component search finds all group members who are attached to the selected object by dual-constraints in the approximate direction of translation. Those objects that are not attached in the direction of translation are un-grouped. With this technique, a row of cabinets that are connected by dual groups can be split into two groups by selecting a cabinet in the center and pulling in the direction that the new group is to translate.

Preliminary testing of the system is very promising. After a very short demo of the system, first time users were able to recreated a reasonably complicated scene in a matter minutes. Future plans are to test the system rigorously and compare results to a system with conventional interaction techniques.



This scene was created in under two minutes and illustrates virtual, dual, and surface constraints.

## References

- [Bi90] E. Bier, Snap-Dragging in three dimensions, ACM Computer Graphics, vol. 24, no. 2, pp. 193-204, Mar. 1990.
- [BS95] R. Bukowski, C. Séquin, Objects Associations: A simple and practical approach to virtual 3D manipulation, 1995 Symposium on Interactive 3D Graphics, pp. 131-138, 1995.
- [GS99] M. Goesele, W. Stuerzlinger, Semantic Constraints for Scene Manipulation, in Proceedings Spring Conference in Computer Graphics '99 (Budmerice, Slovak Republic), pp. 140-146, Apr. 1999.
- [HPGK+94] K. Hinkley, R. Pausch, J. Goble, N. Kassell, Design hints for spatial input, Course Notes - Developing Advanced VR Applications, SIGGRAPH 94, Aug. 1994.