

Imposing Geometric Constraints on 3-D Virtual Objects within an Immersive Modeling System

Norikazu Hiraki, Kiyoshi Kiyokawa, Haruo Takemura and Naokazu Yokoya

Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST)
8916-5 Takayama, Ikoma, Nara, 630-0101, Japan
{norika-h, kiyosi-k, takemura, yokoya}@is.aist-nara.ac.jp

ABSTRACT

This paper describes a user interface technique that allows a user to design not only shapes of 3-D virtual objects but also geometric constraints among these objects in an immersive modeling system. The proposed technique provides an intuitive and easy manipulation for imposing geometric constraints on 3-D virtual objects within a virtual environment.

Keywords

Virtual reality, immersive modeling, geometric constraints

INTRODUCTION

It is desirable for a modeling tool to have the ability to design not only the shapes of 3-D objects, but also their constraints that most real objects have. With such a modeling system [1], a user is able to create partially movable 3-D objects; for example, a cabinet containing drawer and a puppet having a number of joints, as shown in Figure 1.

On the other hand, immersive modeling is a promising approach for designing 3-D objects because of its two key advantages: real-time head-tracked stereoscopic view and spatial direct manipulation to 3-D virtual objects. The former improves the understanding of shapes of 3-D virtual objects and spatial relationships among these objects. The latter improves the accessibility to 3-D virtual objects. For the last decade, a number of immersive modeling systems have been developed [2].

We focus on geometric constraints, and present a novel approach to impose geometric constraints on 3-D virtual objects within an immersive modeling system. The key concept of our approach is that the system provides geometric constraints as primitives that have visible shapes, so that a user can intuitively and easily recognize the effect of geometric constraints and impose geometric constraints

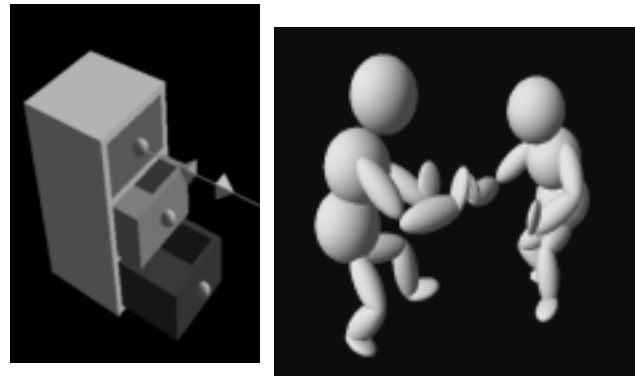


Figure 1: Examples of 3-D virtual objects constructed of geometrically constrained primitives.

on 3-D virtual objects.

SYSTEM OVERVIEW

The system is implemented on a graphics workstation, SGI Indigo2 Maximum Impact. A user observe a virtual workspace through a HMD, Olympus MediaMask, with a real-time head-tracking facility using a 3-D tracker, Polhemus 3SPACE Fastrak. He or she holds a pair of 6DOF input devices using 3-D trackers to manipulate two 3-D cursors. Each input device has two push button switches, one is used to pick 3-D objects, and the other is used to assemble 3-D objects. We will refer to the latter as an *assembling button*.

In the system, the shapes of 3-D objects are constructed by assembling simple geometric shapes (*shape primitives*) such as spheres, boxes, etc. Geometric constraints also have visible shapes (*constraining primitives*) that represent their attributes (constraint type, principal axis and movable range), so that a user can intuitively understand the behavior of the 3-D virtual objects that are bound by the geometric constraints. Geometric constraints are imposed on shape primitives by simply assembling shape and constraining primitives. Therefore, a user need not to learn complex manipulation to impose geometric constraints.

The system supports two-handed manipulations for natural and efficient spatial activity. The system also employs

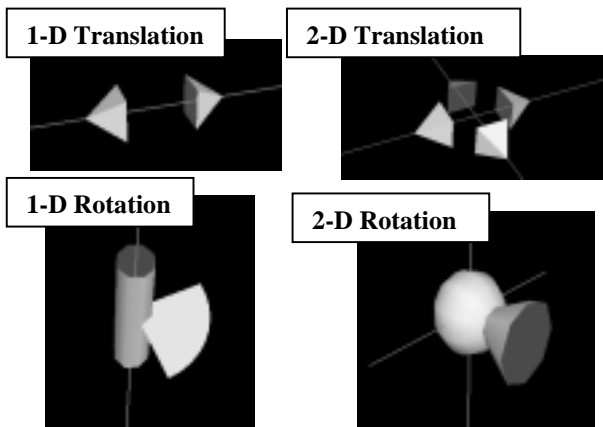


Figure 2: Constraining Primitives.

discrete placement constraints to help a user to arrange 3-D virtual objects precisely. Figure 1 shows examples of 3-D virtual objects constructed in the system.

ASSEMBLING PRIMITIVES

This section explains the assembling manipulation implemented in the system. First, a user sticks a tip of a 3-D cursor into a primitive (a *source primitive*), and pushes the assembling button. After these actions, a line appears from the tip of the 3-D cursor to the center of the source primitive. Next, he or she drags the 3-D cursor into another primitive (a *target primitive*), and releases the assembling button. Then, the source primitive is stuck onto the target primitive, and the line disappears.

GEOMETRIC CONSTRAINTS

We have begun with simple translation and rotation constraints. Figure 2 illustrates four types of constraining primitives employed in our system.

- **1-D and 2-D Translation Constraining Primitives:** These primitives restrict an object to move parallel to an axis and a plane defined by two axes respectively. The principal axes are represented by red lines, and ends are represented by gray pyramids.
- **1-D and 2-D Rotation Constraining Primitives:** These primitives restrict an object to rotate around an axis and two axes respectively. The principal axes are represented by red lines, and the rotation range is represented by a fan and a cone respectively.

The principal axes of a constraining primitive can readily be modified by changing its location and orientation. The movable range of a constraining primitive can be modified by moving its corresponding components (pyramids, a fan and a cone).

Imposing geometric constraints is also achieved by the assembling manipulation. In this case, the order of assembling (which primitive is the target?) is important, since the source primitive is constrained by the target

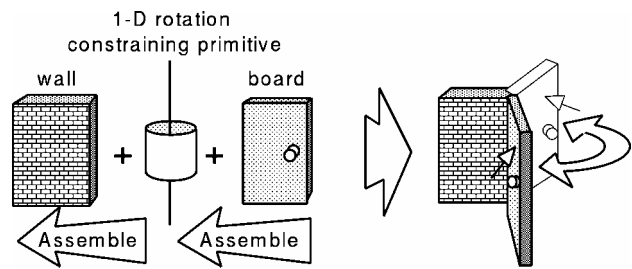


Figure 3: An example of imposing geometric constraints.

primitive when the target primitive is a constraining primitive.

Figure 3 illustrates a process of constructing a door by imposing geometric constraints. The door consists of two shape primitives (a board and a wall) and a 1-D rotation constraining primitive. First, when the board is stuck onto the constraining primitive, the board is bound by the constraining primitive. That is, the board turns around the axis of the constraining primitive. Then, when the constraining primitive is stuck onto the wall, the constraining primitive and the wall are grouped and move together. As a result of these two actions, the wall substantially constrains the board through the constraining primitive.

CONCLUSIONS

In this paper, we have proposed a new user interface technique to impose geometric constraints on 3-D virtual objects. We have also introduced an immersive modeling system which uses the proposed technique. Our system allows a user to construct interactive and dynamic virtual world that includes geometric constraints entirely immersed in the virtual world. Through the use of the system, it is found that the proposed technique is useful for imposing geometric constraints, in spite of limited types of constraints in the current system.

For further study, implementing propagation of constraints and providing other types of constraints (e.g. time constraints which make animation, kinetic constraints like a spring model, etc.) are planned to create more complicated 3-D virtual objects.

REFERENCES

1. C. Elliott, G. Schechter, R. Yeung and S. Abi-Ezzi: TBAG: A High-Level, Fast-Turnaround 3D Animation System, *Proc. ACM SIGGRAPH'94*, pp.421-434, 1994.
2. K. Kiyokawa, H. Takemura, Y. Katayama, H. Iwasa and N. Yokoya: VLEGO: A Simple Two-handed Modeling Environment Based on Toy Blocks, *Proc. of ACM Simpo. on Virtual Reality Software and Technology (VRST'96)*, pp.27-34, 1996.