

BLOCK MANIPULATION WITH HAND IN VIRTUAL SPACE

Kenji Funahashi*, Takami Yasuda**, Shigeki Yokoi**, Jun-ichiro Toriwaki*

*Faculty of Engineering, Nagoya University, Furocho, Chikusa, Nagoya 464-01, Japan

**School of Informatics and Sciences, Nagoya University, Furocho, Chikusa, Nagoya 464-01, Japan

Tel: +81-52-789-3310, Fax: +81-52-789-3807

E-mail: kenji@toriwaki.nuie.nagoya-u.ac.jp

Abstract

This paper describes an interactive manipulation system with a computer generated hand in a virtual work space. We can grasp, move and throw virtual objects in the similar way as real ones. In order to represent the behavior of virtual objects with reality, it is necessary to introduce a precise physical model. However, the more precisely a physical model is made, the more computation time is required to generate CG images according to motion of the hand. Therefore, we developed a new geometric model for block manipulation with a hand in virtual space. In this model, three basic hand actions are concerned such as grasping, moving and releasing. Since simplified gravity and inertia are introduced in the model, the blocks are naturally moved with interactive rate. This model has also a simplified interference check function among objects.

1. Introduction

In recent years, high performance graphic work stations (GWS) have been developed which can generate high quality images of computer defined objects in real time. In addition, several input devices are easily implemented on such GWSs. Using these new technologies, we can not only see the motion of the objects in a sequence of CG images, but also manipulate objects defined in computer.

If we can manipulate virtual objects with a hand in the similar way as real ones, we are able to have many experiences in various virtual worlds. Over the past few years, researchers have expended much effort to develop fundamental technologies[1]. One of the most important elements is simulating dynamic force of grasping object[2]. Another is considering multi-sensors for hand joints[3]. However, they cannot manipulate many objects intuitively at the same time, since it is difficult to calculate the objects interaction exactly[4].

Therefore, we developed a new geometric model for block manipulation with a hand in virtual space[5]. In this model, three basic hand actions are concerned such as grasping, moving and releasing. Since simplified gravity, inertia, and the motion on a floor are introduced in this model, blocks are naturally moved with interactive rate. This model has also a simplified interference check function among objects. Using this model, we realize a virtual block manipulation system with a hand based on a simplified physical rule, which responds in interactive rate.

2. Manipulating blocks with hand

2.1 Hand and objects in virtual space

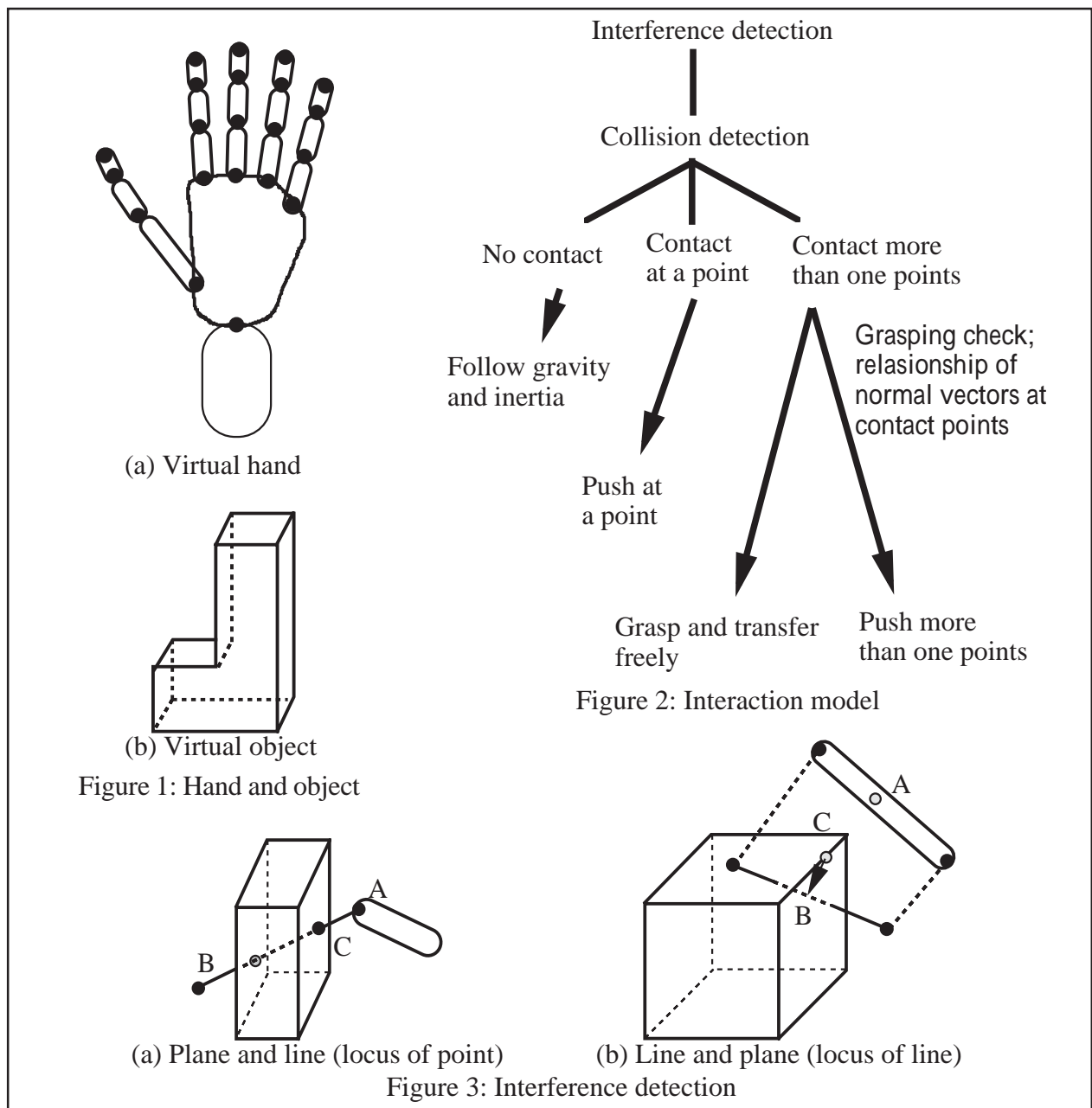
In order to establish a well defined virtual environment, the shapes of a hand and objects to be manipulated should be precisely modeled in computer. However, for the purpose of virtual manipulation, it is necessary to manage virtual objects with interactive rate. Therefore, we have to use a simplified model for a hand and objects due to the restriction of the computational power. A

virtual hand is modeled as cylinders and a polyhedron, indicating fingers and palm, respectively. Several specific points are put on tips and joints of each finger for interference check of other objects. Our simplified hand model is illustrated in Fig. 1(a). It is assumed that virtual objects to be manipulated are rigid and constructed by polyhedron, as shown in Fig. 1(b).

2.2 Interaction model

In this section, we describe an interaction model between a hand and a block in our system (Fig. 2). For natural manipulation, interference detection is indispensable[6]. First, we should find an interference point between each plane of the block and each approximate line which is the locus of a check point on a finger (Fig. 3(a)). Then, we have to examine the interferences between each edge of the block and each approximate plane which is the locus of a line decided by two neighboring check points (Fig. 3(b)). We assume the perpendicular at the center of gravity to the edge as the pseudo-normal vector of it (Fig. 4). Using the locus, we can calculate collision points. Then the block behaves as shown in Fig. 2 according to the number and the location of collision points. When no interference is detected, the block falls according to gravity and inertia described in section 3.1.

If the hand touches the block at one point, it is pushed and moved according to the rule described



in section 2.3. If the hand touches at more than one points, the pseudo-normal vectors of touched planes or edges decide whether the hand grasps the block or pushes it.

For the interactive manipulation, the touched block is not allowed to slide in our model. Furthermore, by ignoring the friction, grasping process can be simulated only by relationship among the pseudo-normal vectors.

2.3 Motion of a block

We consider the pseudo-normal vectors to decide the motion of the block, if the hand touches on more than one points of the block (Fig. 4). In the case that the angle between any two vectors is more than a pre-defined threshold value, the block is considered to be grasped by the hand and moves together with the hand. The fingers are relocated at proper position, if they penetrate into the block. In other case, one should say that the hand pushes it at several touched points. However, since the motion of the block is so complicated according to the relative positions of each touched point, we simplify that the block only translates due to the hand motion.

If the hand touches at one point, the hand push the block without grasping. In Fig. 5, a check point, located on the tip of a finger, for example, moves from point A to point B. We now assume the collision point and the center of gravity as point C and G, respectively, and the vector \vec{CB} is divided into two components, namely \vec{T} and \vec{R} , where

$$\vec{CG} \parallel \vec{T}, \vec{T} \perp \vec{R}, \vec{CB} = \vec{T} + \vec{R}. \quad (1)$$

The block is translated with \vec{T} and rotated with θ around \vec{X} at the center point G, where

$$\theta = \arctan \frac{|\vec{R}|}{CG}, \vec{X} = \vec{R} \times \vec{T}. \quad (2)$$

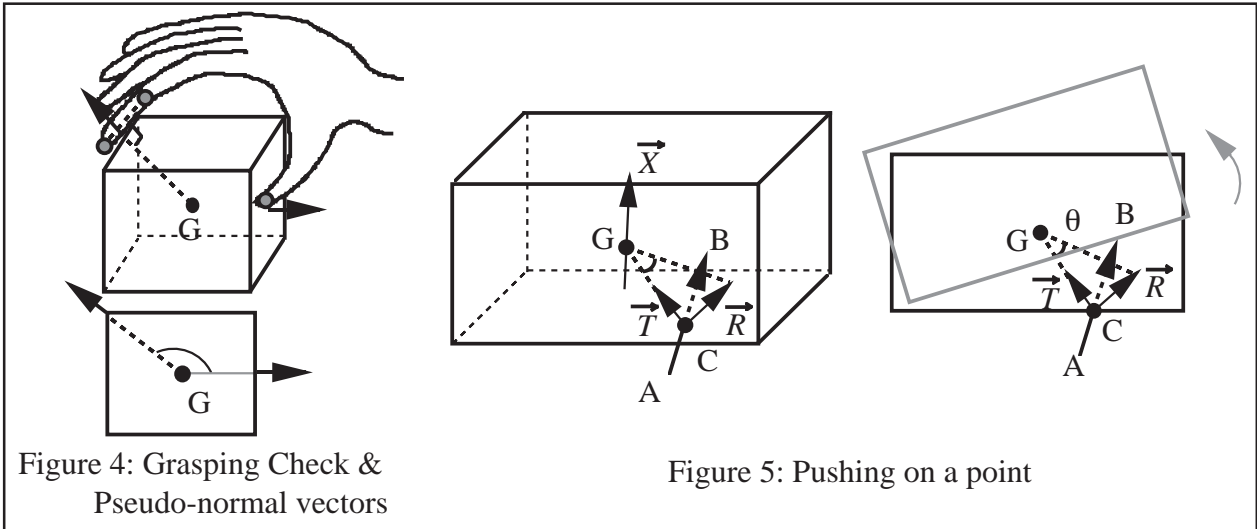


Figure 4: Grasping Check & Pseudo-normal vectors

Figure 5: Pushing on a point

3. Virtual environment

3.1 Simplified gravity, inertia and friction

It is important to know when a hand touches or releases virtual objects during manipulation. In addition, the manipulated object may be interfered by other objects. Therefore, we have to establish the study has been reported about simulating virtual objects with a geometric and physical rule, the simulation system was not modeled hand manipulation[7].

In our interactive manipulation model, gravity and inertia are modeled in a simplified way, that is, when no interference is detected, the block falls with pre-defined distance and is translated with the moving vector for the center of gravity from the previous frame to this one. The momentum for rotation is not implemented yet in the model. The inertia of the block is not preserved after collision with others and floor. They do not rebound. Perfect friction is assumed here, therefore blocks do not slide.

3.2 The interference among fallen blocks and floor

In this section, we describe the motion of a fallen block and its collision with floor. The block falls freely by the effect of the gravity and inertia until interference is detected between floor and itself. If the block penetrates into the floor, it is translated up with the maximum depth of penetrating vertices of the block. Letting contact vertices with floor be V_i ($i = 1, \dots$, the number of them), the convex-hull of V_i is calculated. Once any part of the fallen block comes into contact with floor, all of the interferenced block vertices V_i ($i = 1, 2, \dots$) are found. We now assume the projected point of the point G for the block in the direction of \overrightarrow{Gm} (which is the moving vector for point G from the previous frame to this one) as point G'. The position of the block is stable and it does not move on the floor, if the point G' is projected inside the convex-hull area defined by V_i s (Fig. 6(a)). When the projected point G' is located outside of the convex-hull, we should say the block is instable on the floor (Fig. 6(b)). In this case, the block must be moved until stability is achieved according to the condition between the block and the floor. Three basic conditions are now considered based on the number of contact vertices as shown in Figs. 7 to 9.

(1) Contacting at a vertex V_1 : In the similar way as pushing a block with a finger, \overrightarrow{Gm} is divided into two components such as \overrightarrow{T} and \overrightarrow{R} , where

$$\overrightarrow{GC} \parallel \overrightarrow{T}, \overrightarrow{T} \perp \overrightarrow{R}, \overrightarrow{Gm} = \overrightarrow{T} + \overrightarrow{R}. \quad (3)$$

Then it is rotated by θ around the axis \overrightarrow{X} at the center point C ($=V_1$) as shown in Fig. 7, where

$$\theta = \arctan \frac{|\overrightarrow{R}|}{CG}, \overrightarrow{X} = \overrightarrow{R} \times \overrightarrow{T}. \quad (4)$$

(2) Contacting at two vertices V_1 and V_2 : At first, two lines are drawn on a floor, that pass through V_1 or V_2 and are perpendicular to the line V_1V_2 . If point G' is not located between them (Fig. 8(a)), let one of the two vertices closer to G be point C. The motion of the block is decided by the rule described in (1). Otherwise, $\overrightarrow{Gm'}$ is calculated as follows.

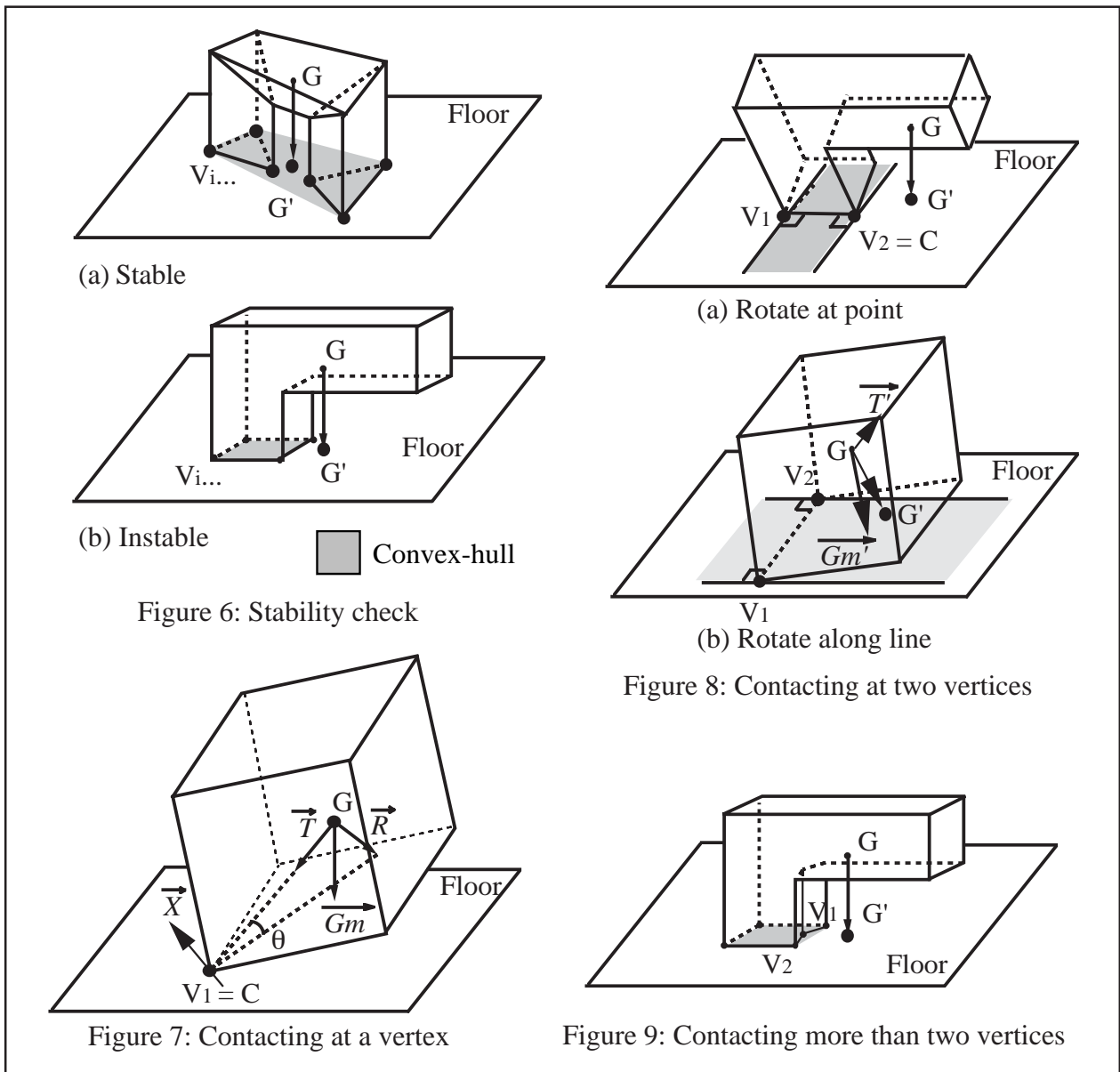
$$\overrightarrow{V_1V_2} \parallel \overrightarrow{T'}, \overrightarrow{T'} \perp \overrightarrow{Gm'}, \overrightarrow{Gm} = \overrightarrow{T'} + \overrightarrow{Gm'}. \quad (5)$$

In the same way of (1), $\overrightarrow{Gm'}$ is divided into \overrightarrow{T} and \overrightarrow{R} . Then it is rotated by θ made by \overrightarrow{R} along the line V_1V_2 (Fig. 8(b)).

(3) Contacting at more than two vertices V_i : In this case, G' is no more inside the convex-hull. In other words, the block is not stable. Renaming two of the closest vertices to G' on the convex-hull as V_1 and V_2 (Fig. 9), the motion of the block follows the case (2) above.

3.3 Interference among blocks

When several blocks collide each other, we could solve the interference problem in the same way as the case between the block and hand or floor. However, it is time consuming to decide each motion of multiple rigid-bodies dynamically. Thus, we simplified this in the following way. If one block is touched with a hand and moved, all of blocks interfered with it are also translated following it. If a block is interfered with other ones, it is not affected by gravity and inertia.



4. Manipulation system

We have implemented the block manipulation system on a graphic work station(GWS); SGI IRIS Crimson+RealityEngine by C language. Our system configuration is shown in Fig. 10. The position and direction of user's hand can be specified with Polhemus 3SPACE FASTRAK, and the angles of wrist and each fingers be with Virtual Technologies CyberGlove. For stereo view, StereoGraphics CrystalEyes is used.

Fig. 11 shows examples of operation in this system. The upper picture shows how one can manipulate a virtual object with CyberGlove. The lower one is a CG image on computer.

5. Conclusion

In this paper we have proposed a simplified computer model for block manipulation with a hand, and developed an interactive manipulation system with a computer generated hand in a virtual work space. Using this system, we can grasp, move and throw virtual objects in interactive rate. While our experimental system suggests a new framework for developing interactive manipulation of 3D objects by a virtual hand, problems remain to be studied in the future. Blocks should be more

naturally moved on a floor with interactive rate. Putting a block on others, it stays there unnaturally although it is not stable in real world. When pushing it by more than one fingers, the block moves with quite different way from real one.

Acknowledgments

We wish to thank the colleagues in Toriwaki laboratory of Nagoya University for useful discussions. This work was supported in part by Artificial Intelligence Research Promotion Foundation and Hayao Nakayama Foundation for Science & Technology and Culture.

References

- [1] M. Bergamasco, et al.: Grasping and moving objects in Virtual Environments: a preliminary approach towards a realistic behaviour, Proc. IEEE International Workshop on Robot and Human Communication, pp.44-49 (1994)
- [2] Y. Kunii, et al.: Object Grasping in Virtual Environment Using Dynamic Force Simulator, Proc. IEEE International Workshop on Robot and Human Communication, pp.261-264 (1994)
- [3] Z. Huang, et al.: A Multi-sensor Approach for Grasping and 3D Interaction, Proc. Computer Graphics: Developments in Virtual Environments, pp.235-253 (1995)
- [4] D. Baraff: Interactive Simulation of Solid Rigid Bodies, IEEE Computer Graphics and Applications Vol. 15, No. 3, pp. 63-75 (May 1995)
- [5] K. Funahashi, et al.: Direct Manipulation System with Virtual Hand for Computer Generated Virtual Space, Proc. Visual Computing '95, pp.114-115 (1995)
- [6] Y. Kitamura, et al.: Interference Detection Among Objects For Operator Assistance In Virtual Cooperative Workspace, Proc. IEEE International Workshop on Robot and Human Communication, pp.442-447 (1993)
- [7] M. Shinya, et al.: Laying out objects with geometric and physical constraints, The Visual Computer Vol. 11, No. 14, pp.188-201 (1995)

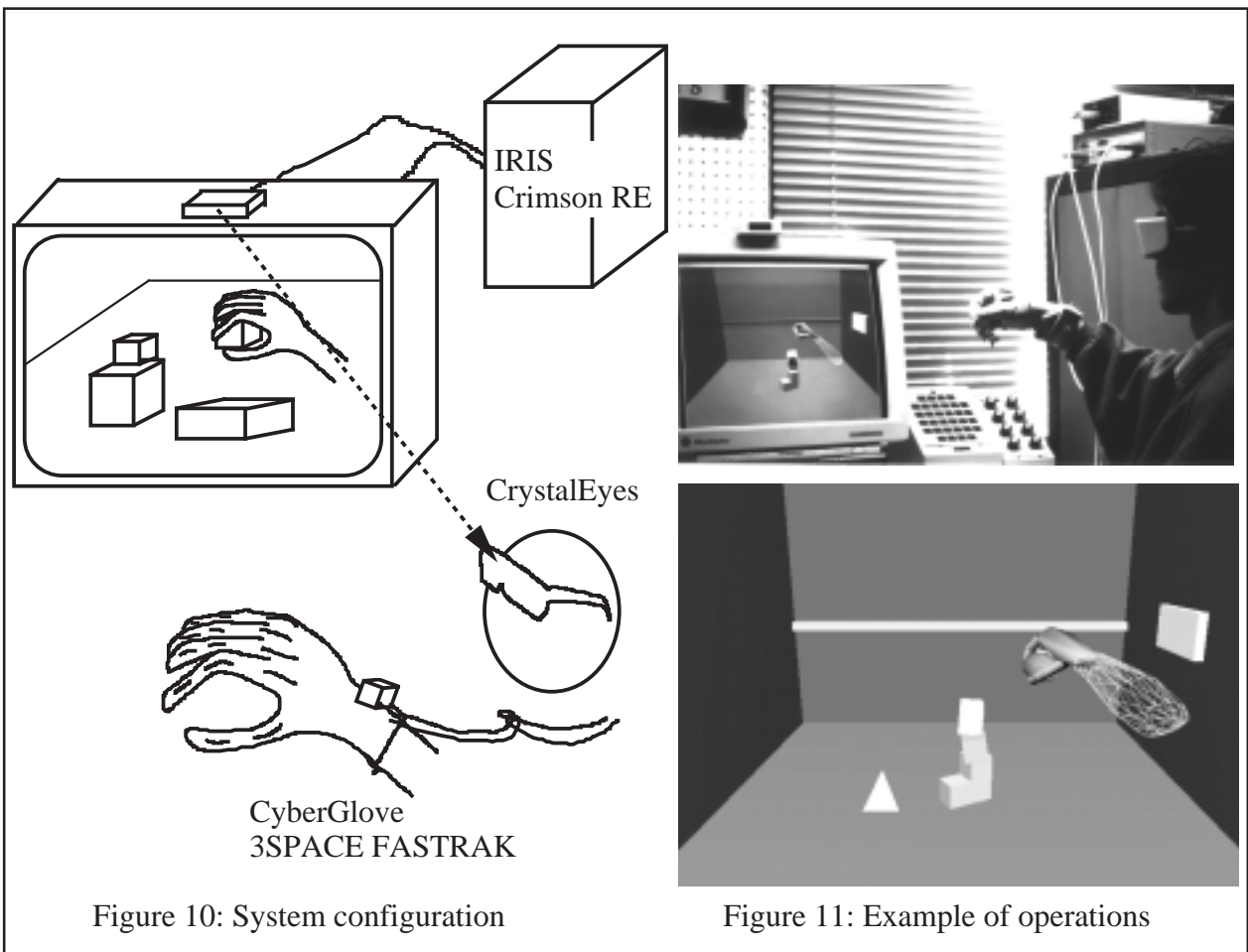


Figure 10: System configuration

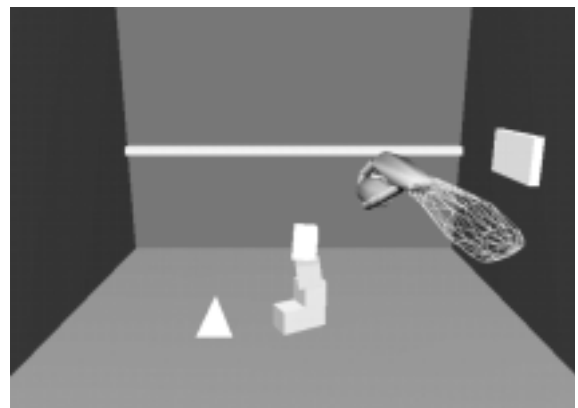


Figure 11: Example of operations