Yoshifumi Kitamura*
kitamura@eie.eng.osaka-u.ac.jp

Amy Yee†
Fumio Kishino*
ATR Communication Systems
Research Laboratories
2-2 Hikaridai, Seika-cho
Soraku-gun
Kyoto 619-0288 Japan

A Sophisticated Manipulation Aid in a Virtual Environment using Dynamic Constraints among Object Faces

Abstract

A natural and intuitive method is proposed to help a user manipulate an object in a virtual environment. The method does not need to assign special properties to the object faces in advance and does not require special hardware. Instead, it uses only the visual constraints of motion among object faces that are dynamically selected by a real-time collision detection method while the user manipulates the object. By constraining more than two faces during the user's manipulation, the proposed method provides an efficient tool for complicated manipulation tasks. First, the method of manipulation aid is described. Then several experiments demonstrate the effectiveness of this method, particularly when the user is requested to precisely place a virtual object in a certain location. Finally, as an application of the proposed manipulation aid, an experiment is conducted to compare the performances of a task (constructing a simple toy) in a real versus a virtual environment. Results show that the distance accuracy and completion time of the virtual task with the manipulation aid is close to that of the real task.

1 Introduction

Virtual reality techniques can be used to provide an intuitive and sophisticated user interface that exploits human spatial perception. Limitations in computational power, however, make it difficult to develop a perfect virtual environment, which transforms a simple task in a real environment into an operation requiring skill in a virtual environment. For example, the easy task of putting a block on a table is difficult in the simplest virtual environment that has no constraints on the position of the object. To overcome such difficulty and perform this easy task in a virtual environment as it would be done in the real world, it is necessary to calculate and simulate such factors as the avoidance of intersection by the test of interference among virtual objects, the fall of virtual blocks caused by gravity, and friction between blocks and table.

One useful way of providing a natural user interface in a virtual environment is to restrict the degrees of freedom (DOF) concerning the motion of objects or the user's hand. Two principal approaches exist to restrict the DOF. The first is to restrict the DOF of the user's hand motion with devices such as force-

Presence, *Vol. 7, No. 5, October 1998,* 460–477

*Currently with the Graduate School of Engineering, Osaka University, Japan
†Currently with Newbridge Networks Corporation, Canada

feedback tools that generate a reaction between two faces touching each other. Examples of this include contact sensations for fingers (Ishii & Sato, 1993), a master station with force functions for telerobotics (Kotoku, Takamune, & Tanie, 1994; Sayers & Paul, 1994), and force sensations for hands (Iwata, 1990). However, these all require special force feedback hardware that operates under special configurations; therefore, free and natural user motion is often impeded.

The second approach is to restrict the DOF of the object motions at all visual positions without restricting the motion of the user's hand. A simple configuration is sufficient for this method; however, careful verification is necessary because the user may feel a sense of incompatibility caused by the differences between visual feedback and motor control. Considerable study has been done on visual techniques using physical simulations, such as contact analysis (Bouma & Vanecek, 1993) and dynamical simulation (Baraff, 1989). These visual techniques have included CAD-based applications (Venolia, 1993; Bier, 1990) and simulations involving operator-assisted magnetic attraction (Chanezon, Takemura, Kitamura, & Kishino, 1993). These are limited to a single level of constraint complexity; otherwise, special functions are attached to the objects in advance. For example, the operator assistance method described in (Chanezon, Takemura, Kitamura, & Kishino, 1993) requires attracting faces to be predefined in order to achieve real-time performance. Moreover, it considers only single face-to-face interactions, and the number of attracting faces is limited to only one for each object. Therefore, it is not flexible and cannot be applied to a variety of tasks employing multiple objects with complicated shapes. The synthetic fixtures described by Sayers and Paul (1994) can handle different object feature interactions at multiple points, but the interaction properties must be attached to the appropriate locations in advance. Another negative point of this method is that it is specific to robotic tasks whose motions are typically predetermined, and thus it is not sufficiently adaptable for the manipulation of arbitrary objects. Others (Fa, Fernando, & Dew, 1993; Snyder, 1995; Kijima & Hirose, 1995) use similar constraints among objects. To date, object manipulation involving six DOF by nontraditional 3-D / 6-D interaction devices has not been studied. Moreover, none of the above studies has discussed the ''naturalness'' of the employed method in comparison with an object manipulation task that is usually done in the real world.

In this paper, we describe an intuitive method to help a user manipulate an object in a virtual environment without requiring that special properties be assigned to the object faces in advance. The method does not use special hardware but instead uses only visual constraints among object faces that are dynamically selected while the user manipulates the object. By constraining more than two faces during assistance, the proposed method efficiently helps the user perform complicated manipulation tasks. We designed two types of experiments. Results from the first experiment show the effectiveness of this method, particularly when the user is requested to place a virtual object in a precise location. The second experiment compares the performances of a task (constructing a simple toy) in a real versus a virtual environment. Results show that performance of the virtual task with the manipulation aid is closer to that of the real task in distance accuracy and completion time than performance of the task without the aid.

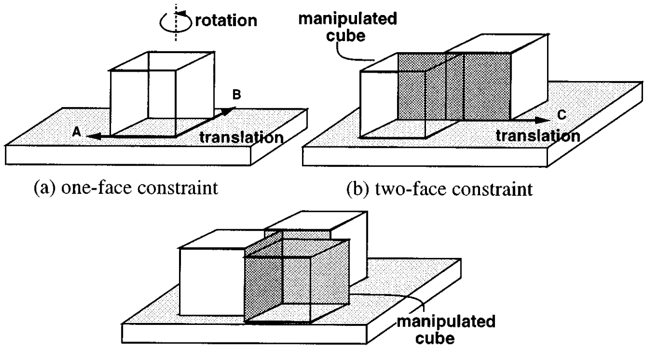
2 Object Manipulation using Constraints among Faces

2.1 DOF in Object Manipulation Task

Consider a general sequence of object manipulations to be given by the following four steps.

(1) Grasp the target object.
(2) Move the object to its destination space.
(3) Make adjustments to the precise position and orientation.
(4) Release the object.

In order to understand the fundamental problems, consider the simple task of aligning three cubes on a table. Each cube has six planar sides that are connected perpendicularly. First, when one cube is placed on the table as shown in Figure 1(a), the motion of the manipulated cube is constrained by the upper surface of the table; therefore, the DOF for object manipulation that the user must adjust during step (2) and step (3) are six (three



(a) one-face constraint

(b) two-face constraint

(c) three-face constraint

Figure 1. *Constraints among faces for object manipulation.*

Table 1. *DOF of Object Motion in Object Alignment Tasks*

Constraints among faces	Real environ.	Simple virtual environ.	Insufficient DOF
(a) 1	3	6	3
(b) 2	1	6	5
(c) 3	0	6	6

translations and three rotations) and three (two translations and one rotation), respectively. Subsequently, if the second cube is aligned adjacent to the first one on the table as shown in Figure 1(b), the motion of the manipulated cube is constrained by two faces (i.e., the upper surface of the table and the contacting face of the first cube). In this case, the DOF of the cube motion is one (one translation). Finally, when the third cube is placed as shown in Figure 1(c), the manipulated cube has no DOF. It is constrained by three faces (i.e., the upper surface of the table and the two contacting faces of the other blocks).

However, in the simplest virtual environment, in which no constraint exists on the position of objects, even the manipulation of step (3) in all of these situations requires six DOF. Therefore, the precise alignment of objects in a virtual environment becomes an operation requiring skill. To avoid this and to provide a simple and natural interface for virtual object manipulation, it is necessary to restrict the DOF of object motion. The relations among DOF of object motion for Figures 1(a)–(c) are listed in Table 1. In this paper, our manipulation aid method considers the object alignments shown in Figures 1(a)–(c); these alignments are called *one-face constraint, two-face constraint,* and *three-face constraint.*

2.2 Real-Time Colliding Face Detection

In order to flexibly apply a manipulation aid method using the constraints among faces to an assembly task using multiple objects with complicated shapes, it is necessary to detect constraining faces dynamically according to object motion rather than being limited to predefined faces. For this purpose, the method has to detect collisions or interference among objects; however, expensive computation has made it difficult to detect them in real time. Much literature has been devoted to solving these problems, but most of the proposed algorithms have limitations on either object shape or the environment. A few methods have been proposed to accurately detect collisions in a general environment in real time.

In this paper, we use a method of real-time colliding face detection for polyhedral objects with complicated shapes (Smith, Kitamura, Takemura, & Kishino, 1995; Kitamura, Smith, Takemura, & Kishino, 1998) and video (Kitamura & Kishino, 1996). This method can detect colliding pairs of faces within 70 ms when the objects have less than 4,000 faces by using an efficient spatial subdivision technique implemented on a workstation (Section 4). Figure 2 (page 463) shows an example of this method: 40–60 ms were required to detect the colliding faces (provided by changes of color to red) between the statue of Venus (1,816 faces) and a space shuttle (528 faces). By using this method, therefore, we can design a virtual object manipulation aid using dynamic constraints among faces. Furthermore, the method provides the user with a natural impression of motion by finishing all procedures within the cycle time of the human perceptual processor (100 ms) (Card, Moran, & Newell, 1983). Details of the method's computation time is discussed by Kitamura, Yee, and Kishino (1996).

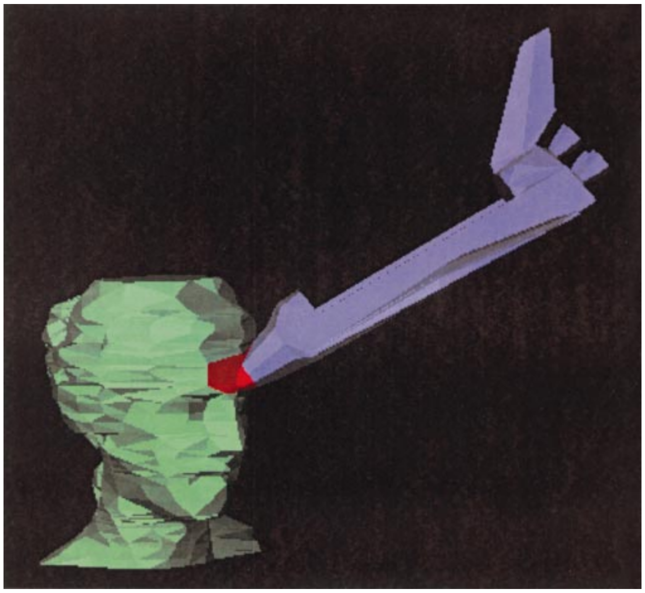


Figure 2. *Result of detecting colliding face pairs (provided by changes of color to red) between a statue of Venus (1,816 faces) and a space-shuttle (528 faces).*

3 Method of Manipulation Aid

The method of our manipulation aid in a virtual environment is described in this section. All objects in the world are modeled as polyhedra (boundary representations), are rigid (nondeformable), and can be concave or convex. Objects may be grasped and moved in a nonpredetermined way, and assistance can be given only for the object currently grasped by the user.

3.1 Overview

Figure 3 shows the role the manipulation aid algorithm plays in the execution flow of a virtual reality system. The system consists of modules to read tracker data, detect collisions, assist manipulation, and render graphics. The main input is a six-DOF tracker device that the user employs to manipulate the virtual objects.

Accurate collision data is necessary to determine how an object should be constrained; therefore, the efficient collision detection algorithm (Smith, Kitamura, Takemura, & Kishino, 1995) and video (Kitamura & Kishino, 1996) is used. This algorithm detects colliding

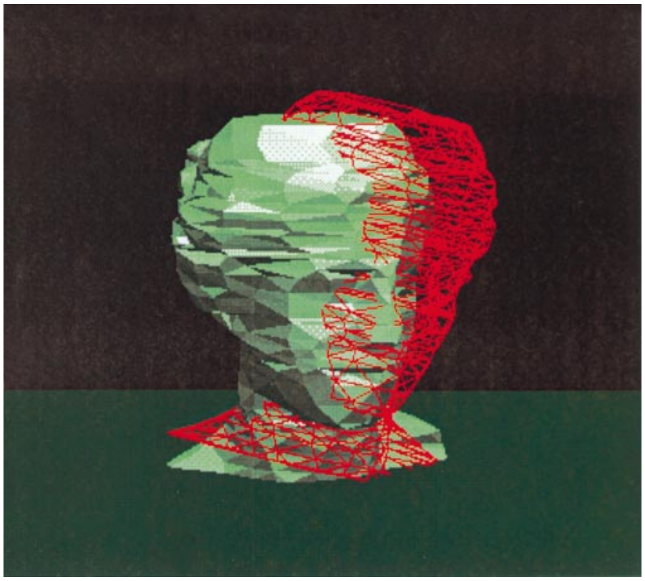


Figure 5. *A snapshot of sensor and displayed object positions in the virtual object manipulation aid. Wireframe shows a sensor position, and solid object shows a displayed position.*

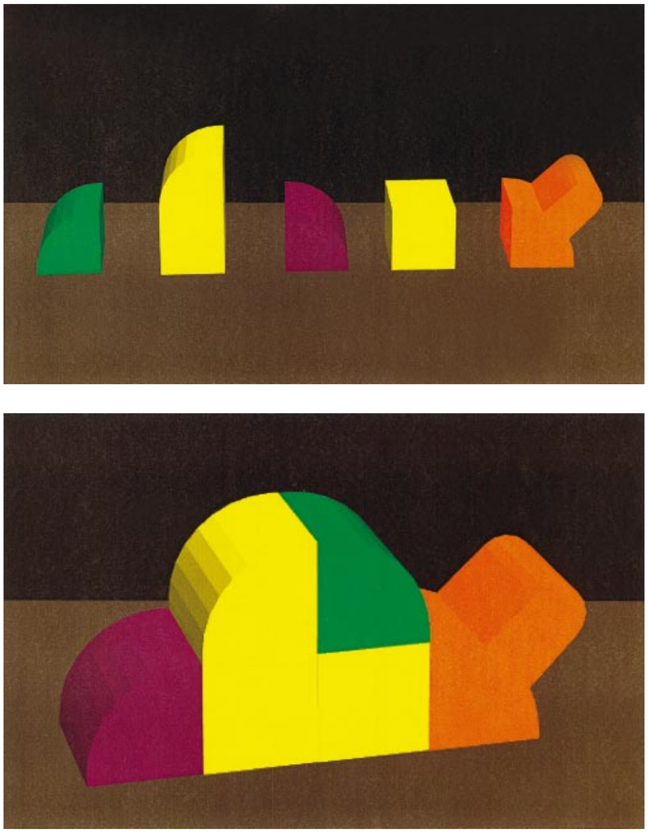


Figure 8. *Initial positions (top) and finished construction (bottom) of blocks for toy snail.*

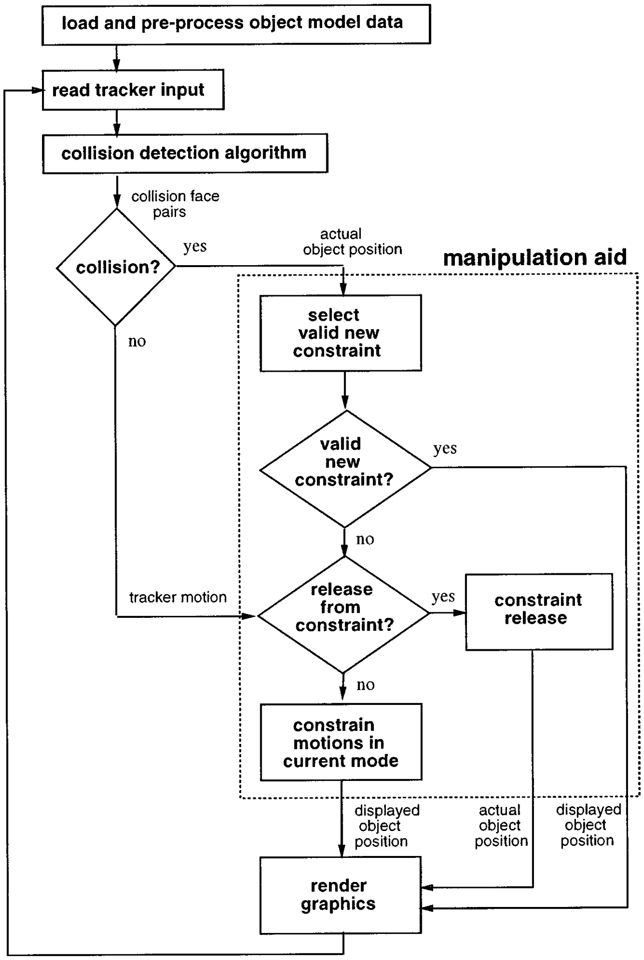


Figure 3. *Role of virtual object manipulation aid in execution flow.*

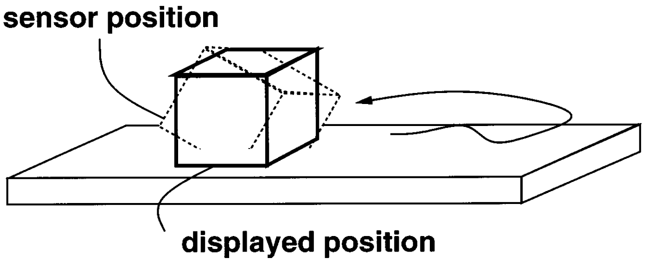


Figure 4. *Sensor and displayed object positions in virtual object manipulation aid.*

pairs of faces in real time for 3-D graphical environments where objects undergo arbitrary motion. The algorithm can be used directly for both convex and concave objects.

The manipulation aid uses the collision data and user inputs to dynamically constrain colliding objects in the environment. The method of manipulation aid is a visual technique that restricts the motion of virtual objects but not the motion of the user's hand. Therefore, there is a distinction between the actual (sensor) and displayed object position (Figure 4). The actual sensor position of the object is controlled by the user's hand inputs. (This is the usual object position on the screen without any constraint.) The displayed object's position is the modified position of the grasped object on the screen after constraints are applied using the manipulation aid method. An example of manipulation aid with a complicated object is shown in Figure 5 (page 463). Wireframe shows a sensor position of the object, and a solid object shows its displayed position in this figure. The key to the method is how the manipulation aid modifies the object position to provide intuitive and compatible assistance to the user during object interaction. In this paper, we use a ''magnetic'' metaphor to express the state transition of objects: the user feels as if the objects have an attracting power generated from a pseudomagnet on the faces when initiating the aided condition. On the other hand, the user feels as if the pseudomagnet has been detached when the aided condition is discontinued.

The manipulation aid algorithm consists of three steps: select constraint, constrain object motion, and release from constraint. The following subsections describe how these steps work.

3.2 Select Face Constraint

An object with a complicated shape may have a number of colliding face pairs that are detected in the above collision-detection stage. By examining the geometry between the collision pairs and the speed of interaction, the intention of the user can be predicted and the face to be constrained can be dynamically selected. Since face-to-face interaction is considered, several conditions are imposed to reduce the number of possible pair candidates. The conditions for a valid candidate pair are as follows.

- The face pair is not already constrained.
- One face in the pair belongs to the object grasped by the user.
- The angle between the two face normals is more than 120 degrees.
- The ratio of the overlapping area of the two faces to the smaller face (subsection 3.4.1) is more than a chosen threshold.

If no valid candidate face pair is found, there is no new constraint, and the algorithm continues to handle previous constraints. If valid candidate face pairs satisfying all of the above conditions are found, the best pair is selected by calculating attraction values with the equation

$$attraction = rC_r + vC_v, \tag{1}$$

where r is the rotation angle between the two face normals, v is the angle between the manipulated object's velocity vector and the normal vector of the target object's colliding face, and C_r and C_v are parameter coefficients. The face pair having the highest attraction value is selected as a valid new constraint.

3.3 Constrain Object Motions

If a new constraint face pair is found by the procedure described in the previous subsection, the displayed object position must be modified to reflect the new assisted position. Translations and rotations are applied to the object to move the selected face in a parallel manner onto the constraining face. The translation vector is the projection of the moving face's centroid to the target face. The rotation matrix is found by

$$M_{rot} = T(O, vtx)A(\theta, \vec{\mathrm{v}})T(vtx, O), \tag{2}$$

where M_{rot} is the rotation to apply to the current object position, vtx is the center of gravity of the moving face, $A(\theta, \vec{\mathrm{v}})$ is the matrix for aligning the moving face parallel to the target face, $\vec{\mathrm{v}}$ is the normal vector orthogonal to the two face normals, θ is the angle of rotation found by the dot product of the two face normals, and $T(vtx, O)$ is the translation from vtx to the origin of the coordinate.

After the grasped object is moved onto the target face, the current constraining faces restrict further motion of the object until the user deliberately releases the constraint. The method for release is described in the next subsection. The rest of this subsection describes how object motion is constrained for simple one-, two- and three-face constraints.

In the *one-face constraint* mode, the object has one constrained face and the motion is constrained to three DOF: two translations on the plane of the target face and one rotation around the normal on that face (Figure 1(a)). The constrained translation is determined by projecting the change in a translational hand-motion vector (sensor data) onto the plane of the target face,

$$\vec{X} = \vec{T} = \overrightarrow{\mathrm{d}t} - (\overrightarrow{\mathrm{d}t} \cdot \vec{n})\vec{n}, \tag{3}$$

where $\vec{X}$ is the constrained translation, $\vec{T}$ is the change in hand translation projected onto the target face, $\overrightarrow{\mathrm{d}t}$ is a vector representing the change in hand translation (sensor data), and $\vec{n}$ is the unit normal of the constraining face. The constrained rotation angle ang is determined by the change in angles of hand motion (sensor data) and the direction of rotation,

$$ang = n_z\omega_a + n_y\omega_e + n_x\omega_r, \tag{4}$$

where, ω_a, ω_e, and ω_r are azimuth, elevation, and roll, respectively, from the hand motion (sensor data), and $\vec{n} = (n_x, n_y, n_z)$ is the unit normal in the direction of rotation. The constrained object motion is determined by X and ang calculated by Equations (3) and (4).

In the *two-face constraint mode*, the object has two constrained faces, and the motion is constrained to one DOF, translation along the two faces (Figure 1(b)). The constrained translation is along a vector orthogonal to both constrained face normals. The equation is

$$\vec{X} = (\vec{n}_1 \times \vec{n}_2) \cdot \vec{T}(\vec{n}_1 \times \vec{n}_2), \tag{5}$$

where $\vec{X}$ is the constrained translation, $\vec{n}_1$ and $\vec{n}_2$ are the unit normals of the two constraining faces, and $\vec{T}$ is the projected hand motion calculated in Equation (3).

In the *three-face constraint* mode, the object has three assisted faces; thus, there is no DOF while the object remains constrained to all three faces (Figure 1(c)).

3.4 Release from Constraint

Since the manipulation method uses an intuitive ‘‘magnetic’’ attraction to constrain an object, the method for releasing the object from a constraint is also intuitive. The release action is like pulling something from a magnetic surface. There are two conditions when a constrained object may ‘‘unsnap’’ from a face: *overlap ratio* and *distance from face.* Satisfying either of these conditions is sufficient for unsnap.

3.4.1 Overlap Ratio. The overlap ratio is calculated from the displayed object position. For a face-to-face constraint, checking the overlap area ensures that an object is constrained only when it is still touching another object. The overlap ratio is used to take into account different object sizes and scaling. The overlap ratio is the ratio of the overlapping area of the two faces to the smaller face. If the *overlap_ratio* is less than the $overlap\_threshold - H$, the manipulated object unsnaps from the face. Here, the *overlap_threshold* should be a value near 0 with hysteresis (width H) to prevent object snapping and unsnapping in borderline cases.

3.4.2 Distance from Face. The distance from face condition uses the distance of the grasped position to the constrained face of the target object. Checking this distance allows the user to deliberately unsnap an object from a face by pulling far enough away. When this distance becomes larger than *dist_threshold*, the manipulated object unsnaps from the constrained face. Here, the *dist_threshold* is a dynamic threshold that varies with the overlap area. To better simulate the magnetic property of our constraint method, we modeled a simple magnet that has a uniform magnetic field. Accordingly, we assume that the greater the contact area, the more force is required to pull the object away from the surface, and thus a greater distance is required to unsnap the object from the surface.

$$dist\_threshold = k\sqrt{A} \tag{6}$$

where A is the overlap area and k is a positive parameter. The parameter k is adjusted to provide the most-realistic feeling of magnetic behavior in a simulation. For the experiments described in the next section, this parameter was determined subjectively since the experimental tasks are more focused on alignment than detaching conditions. This parameter may be determined objectively in a future experiment.

4 Experimental Method

Several experiments are conducted to determine the accuracy and efficiency of the manipulation aid method for the face-to-face constraints described in Section 3. As an application of the method, experiments are conducted for comparison between a virtual task with the proposed manipulation aid and a real task of building toy blocks.

4.1 Experimental Setup

Figure 6 shows the hardware configuration of the experimental system. All input and output devices and sensors are controlled by an SGI ONYX workstation. A 70″ CRT projector displays position-tracked stereoscopic images. User eye position is derived from Fastrack™, a six-DOF magnetic tracker attached to LCD shutter glasses used for stereo viewing. Accordingly, the system can present nondistorted images with depth sensations and motion parallax. The user can grasp and manipulate objects using the ADL-1™, a six-DOF mechanical tracker connected to a serial port of the workstation. The specifications of the ADL-1 device are given in its manual.

4.2 Experiments for Accuracy and Efficiency Evaluation

This subsection describes the methods used for subjective experiments conducted to evaluate the accu-

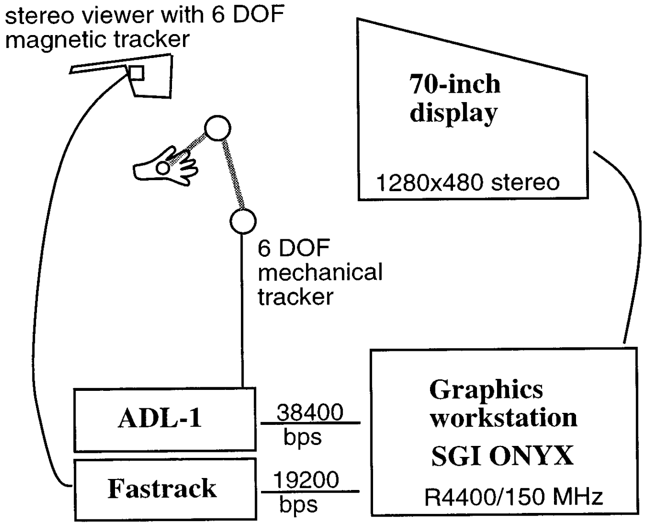
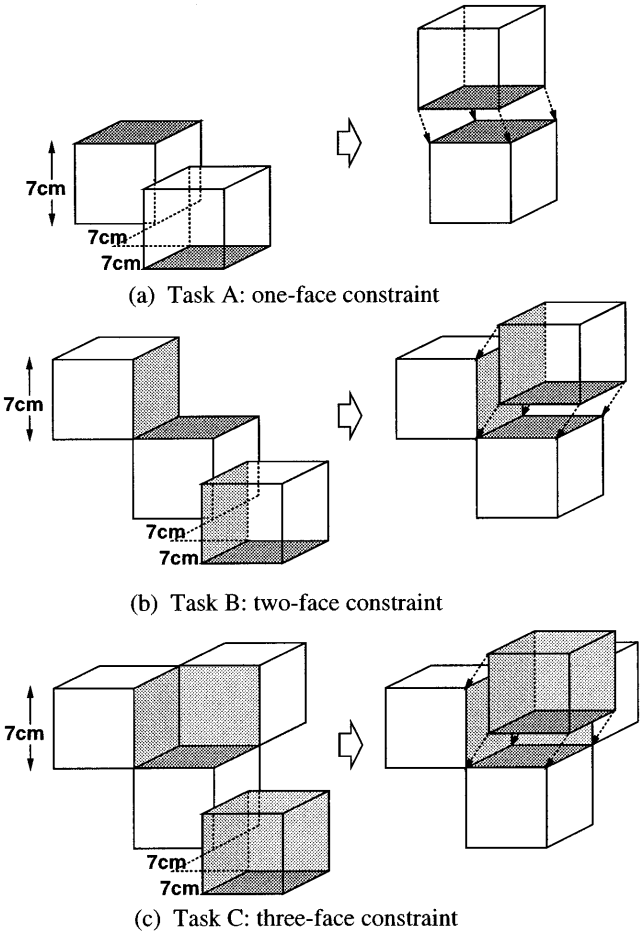


Figure 6. *Hardware configuration for experiment.*



(a) Task A: one-face constraint

(b) Task B: two-face constraint

(c) Task C: three-face constraint

Figure 7. *Experimental tasks for one-, two- and three-face constraints.*

racy and efficiency of the manipulation aid method. First, we define the experimental task for object manipulation, then the feedback to the user for each task is explained. Finally, the sequences of subjective experiments are described. Here we used simple blocks for the experiments because it is difficult to know the constraint status, and it is also difficult to analyze the usability of the proposed system if complicated objects (e.g., Figure 5) are used.

4.2.1 Task Definition. Three tasks—Tasks A, B, and C—are designed for testing one-, two-, and three-face constraints, respectively. Figure 7 shows the task configurations.

Task A: One-face Constraint—consists of two 7 cm cubes, initially separated by 14 cm (between centers) in the horizontal and forward directions. The task is to place the front cube on top of the other, aligning all four corners of the faces as shown in Figure 7(a).

Task B: Two-face Constraint—consists of three 7 cm cubes; two cubes share an edge and form surfaces at a right angle, and the third cube is initially 14 cm away from the lower cube as in the above task. The task is to place the third cube into the right angle surfaces formed by the two other cubes, aligning all six vertices as shown in Figure 7(b).

Task C: Three-face Constraint—consists of four 7 cm cubes, the same as for Task B but with a fourth cube adjacent to the three cubes forming a right angle corner. The task is to place the fourth cube into this corner, aligning all seven vertices as shown in Figure 7(c).

4.2.2 Feedback to the User. To determine the effectiveness of the constraint method, three modes for comparison are designed.

Mode 1: No aid and no cues—no user interface aids and no visual feedback are given.

Table 2. *Accuracy Experiment for Virtual Block Assembly*

Task	Task A, Task B
Mode	Mode 1, Mode 2, Mode 3
Measured	Completion time, distance accuracy, angular accuracy
Trials	10 each, total = 60

Mode 2: No aid with collision cue—no manipulation aid, but the colliding face patches for both objects change color.

Mode 3: With aid and constraint cue—with the proposed manipulation aid, and the constrained faces for both objects change color.

The two color cues of constraint and collision cue modes are different. In Mode 2, colliding face patches detected by the collision-detection part change to a light-green color. In Mode 3, constrained faces change to either red, green, or blue for the one-, two-, or three-face constraint, respectively. The original object colors are soft pastel, while the cue colors are slightly darker.

4.2.3 Method for Accuracy Evaluation.

The purpose of this experiment is to compare the accuracy of object placement in each of the three modes described above. Subjects are asked to complete Tasks A and B as accurately and quickly as possible in each trial. (Task C is skipped in this experiment because with the manipulation aid, the accuracy is close to 100% due to 0 DOF with three-face constraint.) Ten trials are done for each mode and task. Three measurements are taken: *completion time*, *distance accuracy*, and *angular accuracy*. Completion time is the time between object grasp and release, as measured by the computer's real-time clock. Accordingly, each task consists of only one object grasp and one release, that is, the user cannot readjust the object after it has been released. Distance accuracy is the sum of the distances between the four, six or seven vertices of the cubes in Tasks A, B, and C, respectively. Angular accuracy is the sum of the three angle errors (azimuth, elevation, and roll) from the target position. A summary of the accuracy experiment's method is shown in Table 2.

Table 3. *Efficiency Experiment for Virtual Block Assembly*

Task	Task A, Task B, Task C
Mode	Mode 1, Mode 3
Level	4mm, 3mm, 2mm average distance accuracy per vertex
Measured	Completion time
Trials	10 each, total = 180

4.2.4 Method for Efficiency Evaluation.

The purpose of this experiment is to compare the time efficiency for task completion given a certain accuracy level. We use the distance accuracy as our accuracy criterion since the requirement levels are chosen relatively low (accuracy) and the angles are small. In order to compare the results of all three tasks, we use distance error per vertex average instead of the sum.

We can this average vertex error the *accuracy requirement level.* The smaller the accuracy requirement, the more difficult the task becomes. When the object is placed in such a way that the distance accuracy falls below the required level, the object changes color to indicate task completion. The subjects do Tasks A, B, and C for each of three accuracy levels (4 mm, 3 mm, and 2 mm distance error per vertex average) in Modes 1 and 3. (We skip Mode 2 because it is shown to be inefficient by the results given in Section 5.) The task completion times are measured for ten trials for each combination of level, mode, and task as summarized in Table 3.

4.3 Comparison with Real Task

As an application of the method, several experiments are conducted to compare a virtual task with the proposed manipulation aid and a real task of building toy blocks.

4.3.1 Task Definition.

The task is to construct a ‘‘snail’’ using five predefined blocks. Figure 8 (page 463) shows the initial positions of the blocks (top) and the completed snail (bottom). The virtual blocks are modeled to resemble the real blocks in geometry, size, and color.

The assembly of the toy snail is carried out in three different modes: *Virtual without aid*—assemble virtual blocks without manipulation aid or collision color cues (same as constraint Mode 1 in 4.2.2.) *Virtual with aid*—assemble virtual blocks with manipulation aid and constraint color cues (same as constraint Mode 3 in 4.2.2.) *Real*—assemble real blocks.

In order to optimize the comparison of the virtual and real manipulations, the following rules are established for the manipulation of real blocks.

- Only the thumb and one other finger of one hand are allowed to grasp an object.
- Turning the object with fingers is not allowed. The user must turn the wrist or arm.
- Only the grasped object may be touched.
- During the time between object grasp and release, the subject's elbow or palm cannot rest on any structure (e.g., table, other blocks).
- Objects already placed cannot be moved while placing other blocks.

As with the experiments in subsection 4.2, there are two experiments in each set: one for comparing accuracy and the other for efficiency. The methods for both experiments are described in the next subsections.

4.3.2 Method for Accuracy Evaluation. The purpose of this experiment is to compare the distance accuracy and completion times for the three modes. Subjects were asked to build the toy snail as quickly and accurately as possible. Two measurements were taken: *completion time* and *distance accuracy.* Completion time is the real clock time from grasp of the first block to release of the last block in the assembly sequence. Distance accuracy is the sum of the distance errors between all adjacent vertices. In a real environment, the distance errors are difficult to precisely measure because of the imprecise shapes of the real blocks. In several test trials, the distance errors measured by a vernier micrometer are roughly 2 mm or less per vertex. Therefore, a maximum distance error of 2 mm per vertex is assumed for the real task. Eight trials are completed in each of the three modes, as summarized in Table 4.

Table 4. *Accuracy Experiment for Toy Snail Assembly*

Task	Toy snail
Mode	Virtual without aid, virtual with aid, real
Measured	Completion time, distance accuracy (virtual task)
Trials	8 each, total = 24

Table 5. *Efficiency Experiment for Toy Snail Assembly*

Task	Toy snail
Mode	Virtual without aid, virtual with aid, real
Level	3mm average distance accuracy per vertex
Measured	Completion time
Trials	8 each, total = 24

4.3.3 Method for Efficiency Evaluation. The purpose of this experiment is to compare the time required to construct the toy snail within a certain distance accuracy. The accuracy requirement was selected to be 3 mm per vertex. Ideally, 2 mm per vertex would best correspond to the real task, but the task becomes very difficult for the virtual task without any manipulation aid. The task is to place each object until the distance error falls below 3 mm per vertex, which is indicated by a change of color in the object. In this stage, eight trials are also carried out in each of the three modes, and completion times are measured (Table 5).

5 Experimental Results from Accuracy and Efficiency Evaluation

Results obtained for the two experiments are discussed in this section. Five subjects participated in the experiments, their ages ranging from the mid-twenties to early thirties. There were three males and two females; four had experience working with 3-D virtual environments, while one had no such experience. The experiments consisted of two stages: accuracy evaluation and efficiency evaluation. Prior to the first experiment, the subjects practiced using the system in the various modes to become familiar with the behaviors of the virtual environment and system hardware. During the trials for both experiments, subjects were allowed to take short rest breaks as needed. Parameters employed in the experiments were $C_r = 3.0$, $C_v = 1.0$, $overlap\_threshold = 0.05$, $H = 0.05$, and $k = 0.5$. Images were updated 15–30 frames per second during the experiments. Details of the variance of update rate are discussed by Kitamura, Yee, and Kishino (1996).

5.1 Accuracy Evaluation Results

Results obtained for the accuracy evaluation of the experiments are discussed in this section.

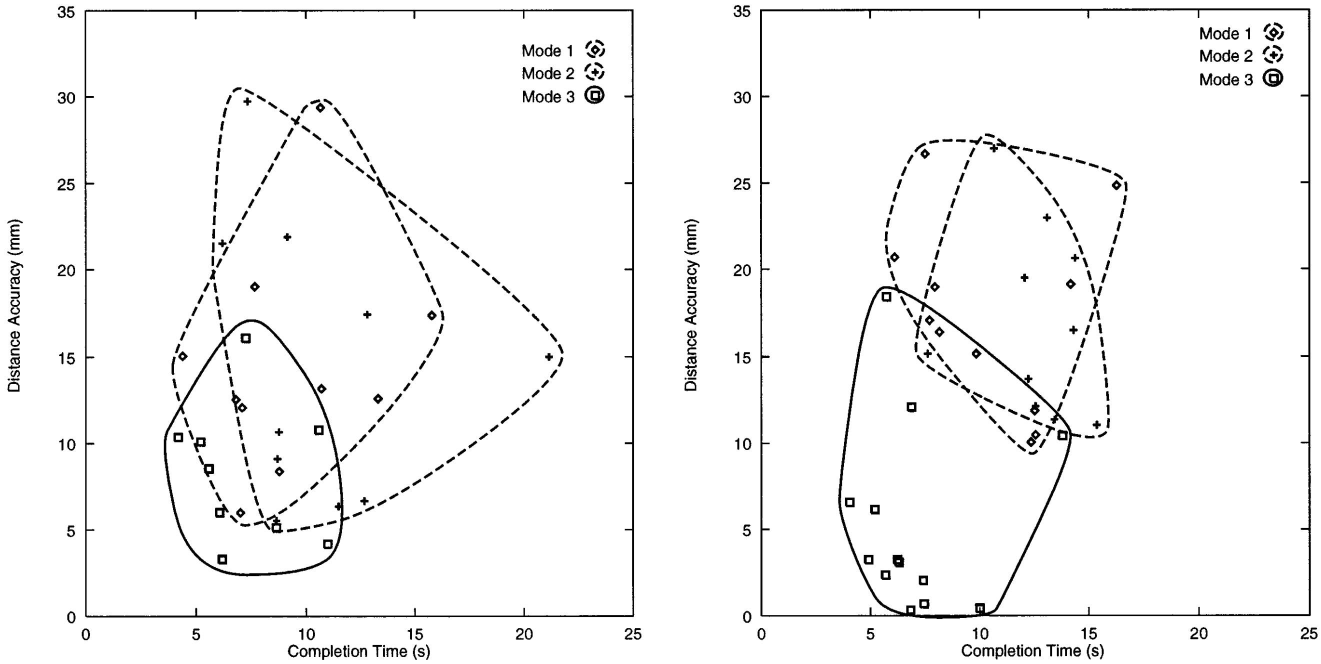
5.1.1 Results of a Typical Subject. Distance accuracy versus completion time for Task A performed by one typical subject is shown in Figure 9(a). The distance errors and completion times with the manipulation aid (Mode 3) are, on average, less than without the aid (Mode 1 and 2). A comparison of Modes 1 and 2 reveals that Mode 2 took slightly longer without gaining much improvement in accuracy. This result corresponds to the comment made by several subjects that the collision color cue did not aid but rather distracted them while performing object alignment. Figure 9(b) shows the same conditions as above but for Task B. Again, the distance errors and completion times are less for Mode 3 than for the other two modes. Compared to Task A, there is a bigger separation between the Mode 3 data group and the other groups, indicating that the constraint method provided more gain for Task B than for A. This result is expected since Task B is more difficult than A, but in the constraint mode Task B had fewer DOF than Task A.

The next two graphs, Figures 10(a) and (b), show the angular accuracy versus completion time from the same trial data of the previous two graphs. The difference between trials with aid data and without aid data is clearer for angular accuracy than distance for both Tasks A and B. This result is reasonable because an angular offset will cause distance errors while there can be a distance offset without angular errors. Therefore, Task A, which has more angular errors than Task B, also has higher distance errors. For Task B, the angular errors are virtually zero because the only DOF remaining is one translation. Again, we see that Modes 1 and 2 have almost the same amount of errors despite use of the collision color cue.

5.1.2 General Trend of all Subjects. The scatter plots for the other four subjects showed results similar to the four previous graphs. The average distance accuracy of each subject can be seen in Figures 11(a) and (b) for Tasks A and B, respectively. All subjects had smaller distance errors with the manipulation aid than without it, and the majority of subjects had worse results using the collision cue than not using it. The errors were considerably higher for the fifth subject compared to the others because this (inexperienced) subject had difficulty with depth perception; the problem may be due to the subject not having much exposure to or experience working with 3-D virtual environments.

Figures 12(a) and (b) show each subject's average angular accuracy for both tasks. Comparing the distance and angular averages for the three modes, the gain in angular accuracy is more substantial than that in distance with Mode 3 (manipulation aid). The angular errors for Task B with Mode 3 have no rotational DOF in a two-face constraint.

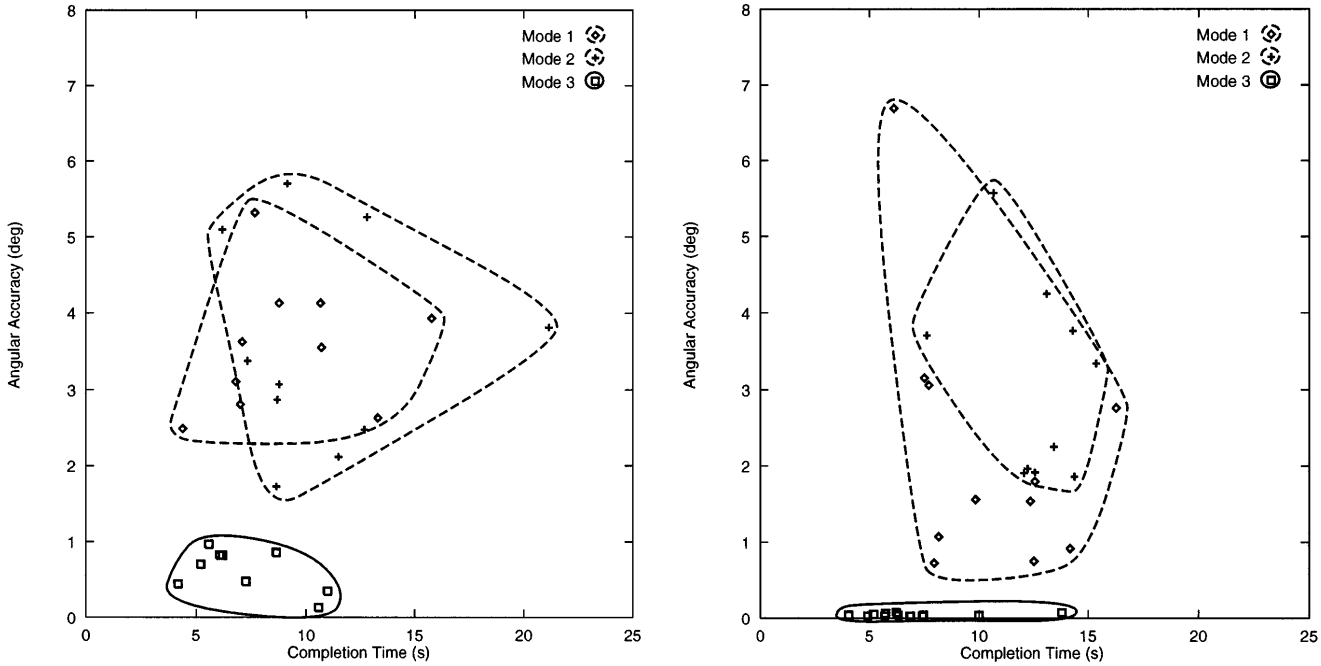
5.1.3 Effect of the Manipulation Aid. Table 6 shows the average gains of the measured data in Tasks A and B using the manipulation aid. Here, the gains are calculated by $gain = (1 - ratio) \times 100\%$, where $ratio$ is the performance ratio with aid (Mode 3) to without aid (Mode 1). The gains are higher for Task B because this task is more difficult without any constraint than Task A, while at the same time it is easier with a constraint. The angular gain of Task B should theoretically be 100%, since in the constraint mode only one DOF remains and it is a translation. With the manipulation aid, these experiments show that, on average, the distance accuracy can be as much as 60% better with a time saving of up to 40%.



(a) Task A

(b) Task B

Figure 9. *Distance accuracy for Task A and B of one subject.*



(a) Task A

(b) Task B

Figure 10. *Angular accuracy for Task A and B of one subject.*

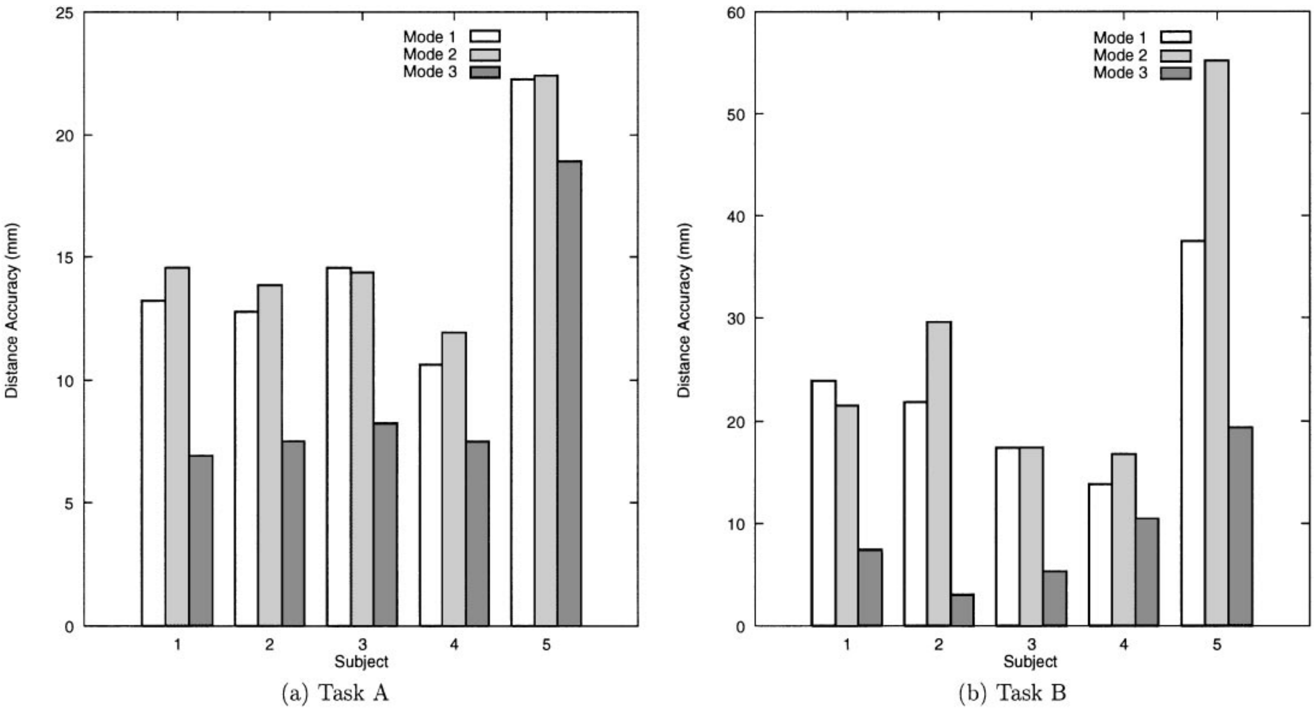


Figure 11. *Average distance accuracy for Task A and B.*

Figure 13 combines Table 6 with some theoretical values to obtain the gain versus the number of constrained DOF. (Task A has three constrained DOF, B has five and C has six.) Theoretically, Task C should have 100% gain because there are no DOF remaining, and zero-assisted DOF corresponds to zero gain. The gains in Table 6 are plotted for distance and angular accuracy with standard deviations. The large standard deviations may be due to the small number of subjects and the difference in capabilities. Since there are only two experimental points with two theoretical points for each curve, this graph presumably shows just rough tendencies for estimating the amount of gain for a certain number of constrained DOF with the proposed method.

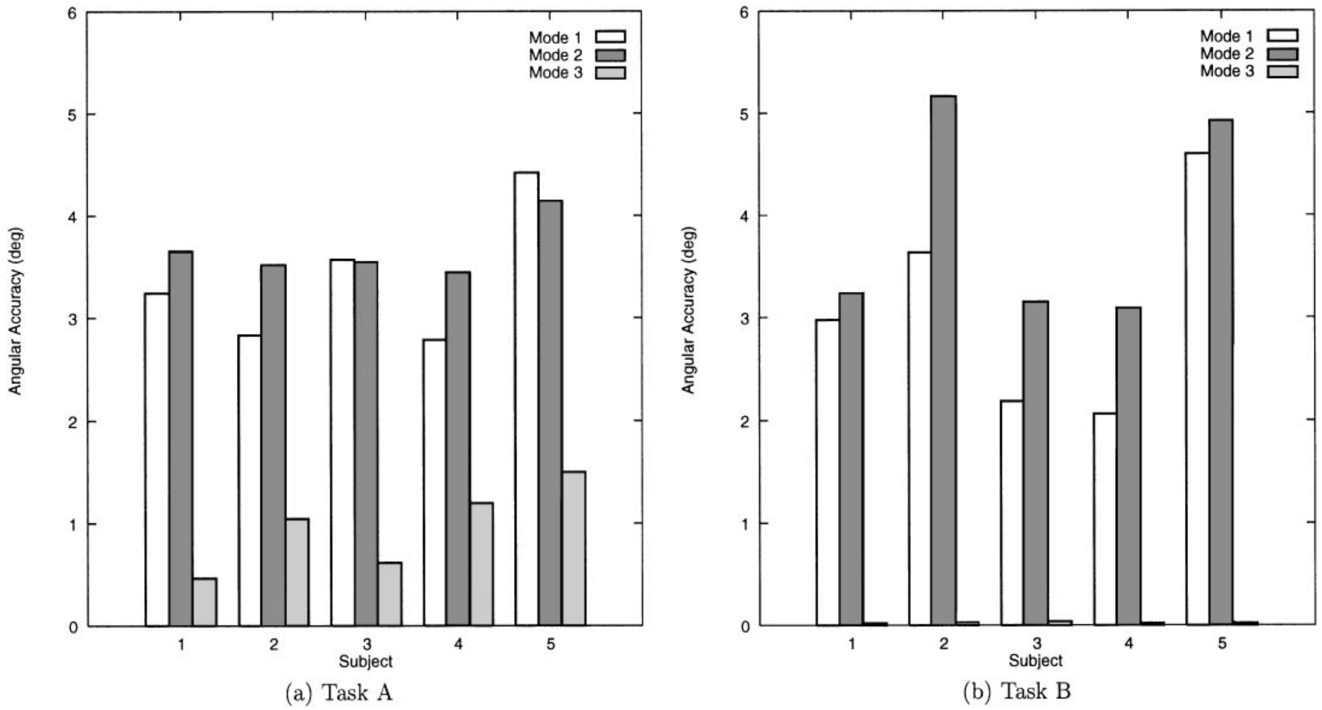
5.2 Efficiency Evaluation Results

Results obtained for the efficiency evaluation of the experiments are discussed in this section. Since each subject had a different skill level in using the system, the completion times had to be normalized before determining the average completion time (over all subjects) for each combination of level, mode and task. For each subject, the times for each combination were first averaged. These averages were normalized using Equation (7), resulting in normalized averages in the range of 0 to 100 for each subject:

$$t_{navg} = \frac{t - t_{\min avg}}{t_{\max avg} - t_{\min avg}} \times 100(\%), \qquad (7)$$

where t_{navg} is the time average of one subject in the chosen combination normalized against the averages of all combinations for the same subject, t is the average time for the chosen combination, and t_{minavg} and t_{maxavg} are the minimum and maximum averages over all combinations for the same subject, respectively.

The normalized times, t_{navg}, are then averaged over the five subjects to obtain the average normalized time. Figure 14 shows the normalized times for the three tasks in Modes 1 and 3 for three accuracy levels. The graph shows that without constraints the normalized times are



(a) Task A

(b) Task B

Figure 12. *Average angular accuracy for Task A and B.*

Table 6. *Average Gains for Tasks A and B (%)*

	Task A	Task B
Distance accuracy	35	59
Angular accuracy	71	99
Completion time	21	41

always higher in all three accuracy levels. As the accuracy requirement increases, the difference between the Mode 1 and Mode 3 curves becomes greater.

6 Experimental Results from Comparison with Real Task

This section presents the results from the experiment conducted to compare performance of the task of constructing a simple toy in a real versus virtual environment. Five subjects (different from those in the previous

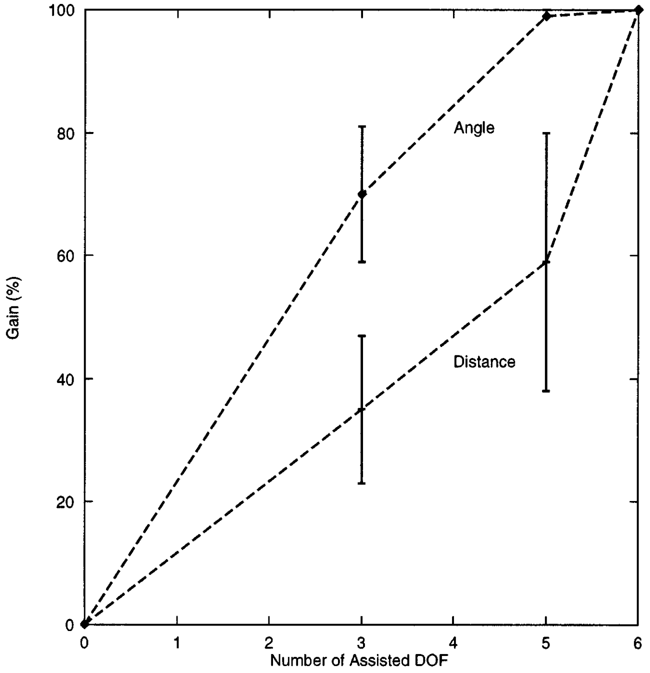


Figure 13. *Gain from using dynamic constraints as a manipulation aid.*

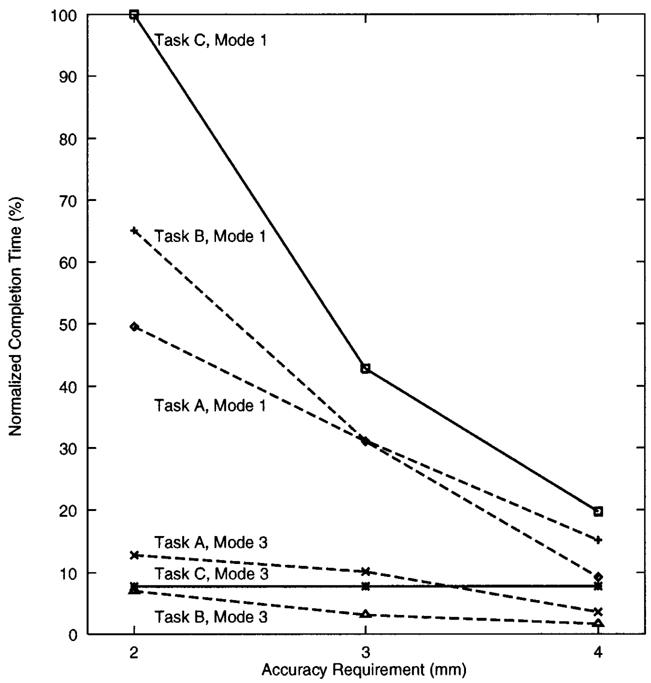


Figure 14. *Normalized completion times for three tasks.*

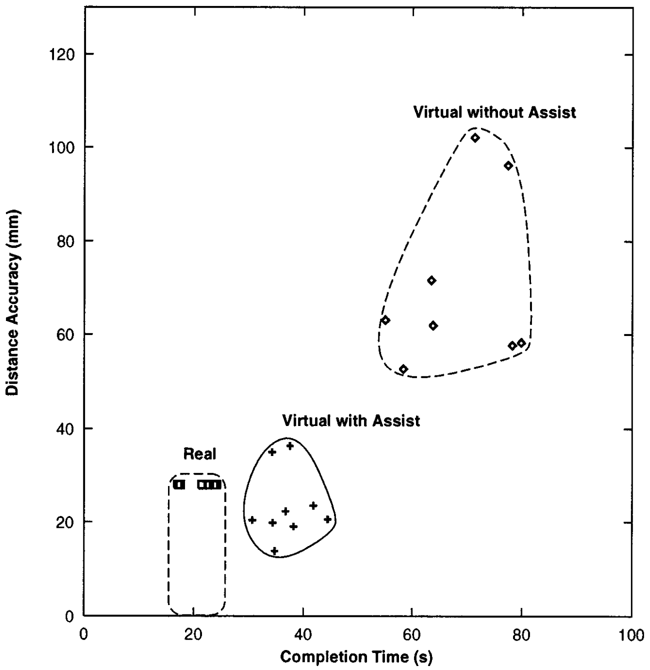


Figure 15. *Distance accuracy for virtual/real tasks of one subject.*

section) participated in the experiments, their ages ranging from the mid-twenties to early thirties. There were three males and two females, and all had experience working in 3-D virtual environments. Prior to the experiment, the subjects practiced using the system in the various modes to become familiar with the behaviors of the virtual environment and system hardware in the experiment. During the trials for both stages, subjects took as many rest breaks as they required.

6.1 Accuracy Evaluation Results

An example plot of completion time versus distance accuracy for one subject is shown in Figure 15. (Similar results were obtained from the other subjects.) There are three distinct groups of data, corresponding to the three modes. For the real task, the distance accuracy of all points were set to a maximum estimated value, with the actual accuracy somewhere between 0 and the maximum. The maximum error was estimated by assuming a 2 mm error per vertex and multiplying by 14 vertex pairs to yield 28 mm for the maximum distance accuracy for the real task.

The scatter plot shows that the virtual task group with the manipulation aid is closer to the real task than in the virtual task without the aid. In virtual-with-aid mode, the distance accuracy was close to those of the real task, while the average completion times were slightly longer. The time delay may be due to the nature of virtual environments and limitations in using a mechanical device instead of hand and fingers. As expected, the tasks without the manipulation aid had higher errors and completion times.

6.2 Efficiency Evaluation Results

In this experiment, objects were placed within a distance accuracy of 3 mm per vertex. The completion times were measured, and the average times for each subject are plotted in Figure 16. The average times with the manipulation aid are generally closer to those of the real task than are the times without the manipulation aid. To compare the time difference across all subjects,

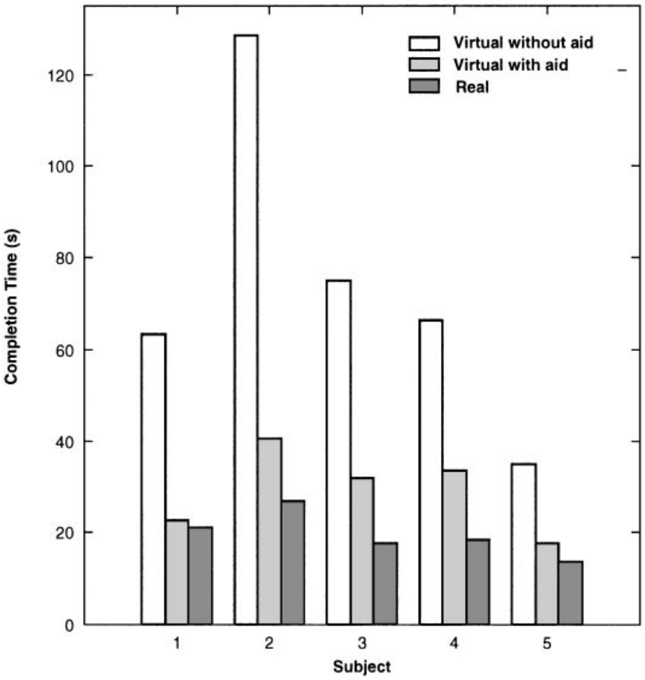


Figure 16. *Average task completion time for virtual/real tasks.*

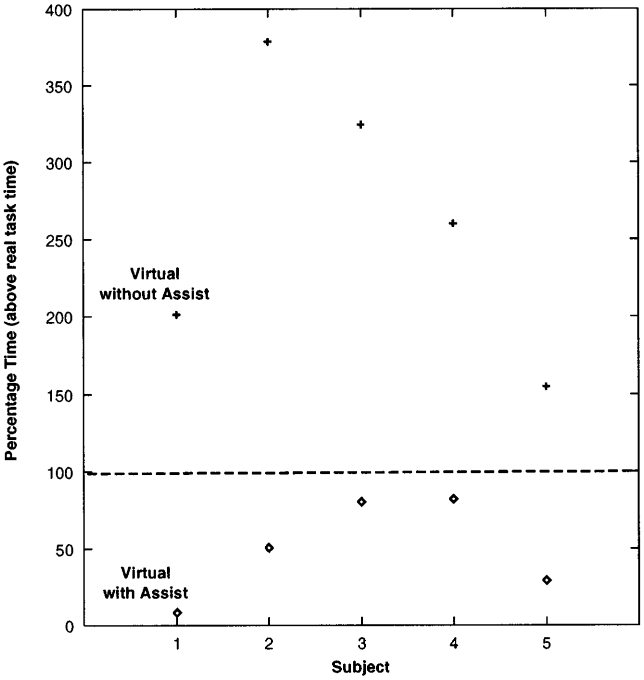


Figure 17. *Percentage of task completion time above real task time.*

time percentages were calculated for each subject as follows:

$$percentage = \frac{t_M - t_{realavg}}{t_{realavg}} \times 100(\%),$$

where t_M is the average time for either with aid or without aid, and $t_{realavg}$ is the average time for the real task. Figure 17 shows the percentage results. The dashed line separates the two modes. With the manipulation aid, all subjects took less than twice the real task time, whereas without the aid they took up to five times longer. The most efficient virtual task took only 8% longer than the real task. These percentages indicate that while the virtual task with manipulation aid requires more time than the real task, the additional time is typically less than twice.

7 Conclusions

In this paper, we proposed a natural and intuitive method to assist a user in manipulating an object in a virtual environment without the need to assign special properties to the object faces in advance. The method does not require special hardware but uses only visual constraints among object faces that are dynamically selected while the user manipulates the object. Results showed that this manipulation aid method provides significant gains in distance accuracy and angular accuracy as well as in task completion time, making this method a useful manipulation aid for virtual environments. The experimental results also showed higher gains for more-constrained faces. In a complicated virtual environment with many objects, the precise placement of objects becomes increasingly difficult without the use of a manipulation aid, but easier with the proposed manipulation aid method. The method also shows higher gains when a high level of precision is required. The results show that the virtual task with the proposed manipulation aid is natural and intuitive, i.e., closer to the real task in distance accuracy and computation time.

It was thought that a collision color cue might assist the user when no other interface aid is available. However, the results showed that this is not the case. Instead, the use of a collision cue was found to distract the user's attention (based on users' comments during alignment), and the results were poorer than without a collision cue. Thus, a collision color cue without a manipulation aid is inadequate for precise manipulation of virtual objects.

Observations of the qualitative effectiveness of the proposed manipulation aid method indicated that users immediately felt confident in using the intuitive manipulation aid but felt frustrated when there was no aid for precise alignment tasks. Accordingly, the constraint method can not only provide better accuracy and times but can also reduce the work strain on the user. Even in simple tasks, the manipulation aid method enabled the user to manipulate and place objects more precisely and in less time. It is reasonable to predict that this method will provide even greater assistance in a more complicated environment.

One of the extensions of our work is to design and evaluate the manipulation aid not only with visual feedback-based constraints but also with other feedback (or integration), such as force and sound (Noma, Kitamura, Miyasato, & Kishino, 1996). Another direction is to apply our manipulation aid technique to the mixed-reality environment that includes interactions among virtual and real objects (Kitamura & Kishino, 1997).

Acknowledgements

The authors would like to thank Dr. Paul Milgram of the University of Toronto and Dr. Haruo Takemura of the Nara Institute of Science and Technology for their useful discussions. Special thanks also to Dr. John Dill of Simon Fraser University for his useful comments.

References

ADL-1 Manual. Shooting Star Technology Inc., Brunaby, BC, Canada. ADL-1 manual, version 2.1 firmware edition.

Baraff, D. (1989). Analytical methods for dynamical simulation of nonpenetrating rigid bodies. *Computer Graphics, 23*(3), 223–232.

Bier, E. A. (1990). Snap-dragging in three dimensions., *In Proceedings of 1990 Symposium on Interactive 3-D Graphics,* 193–204. ACM.

Bouma, W. J., & Vanecek, G., Jr. (1993). Modeling contacts in a physically based simulation. *In Proceedings of Symposium on Solid Modeling and Applications,* 409–418. ACM.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Chanezon, A., Takemura, H., Kitamura, Y., & Kishino, F. (1993). A study of an operator assistant for virtual space. In *Proceedings of Virtual Reality Annual International Symposium,* 492–498. IEEE.

Fa, M., Fernando, T., & Dew, P. M. (1993). Interactive constraint-based solid modeling using allowable motion. In *Proceedings of Symposium on Solid Modelling and Applications,* 243–252. ACM.

Ishii, M., & Sato, M. (1993). A 3-D interface device with force feedback: a virtual workspace for pick-and-place tasks. In *Proceedings of Virtual Reality Annual International Symposium,* 331–335. IEEE.

Iwata, H. (1990). Artificial reality with force-feedback: development of desktop virtual space with compact master manipulator. *Computer Graphics,* 24(4), 165–170. ACM.

Kijima, R., & Hirose, M. (1995). The impetus method for the object manipulation in virtual environment without force feedback. *Symbiosis of Human and Artifact* (*Proceedings of HCI'95*), 479–484.

Kitamura, Y., & Kishino, F. (1996). Real-time colliding face determination in a general 3-D environment. In *Video Proceedings of Virtual Reality Annual International Symposium.* IEEE.

Kitamura, Y., & Kishino, F. (1997). Consolidated manipulation of virtual and real objects. In *Proceedings of Virtual Reality Software and Technology,* 133–138, ACM.

Kitamura, Y., Smith, A., Takemura, H., & Kishino, F. (1998). A Real-Time Algorithm for Accurate Collision Detection for Deformable Polyhedral Objects. *Presence: Teleoperators and Virtual Environments, 7*(1), pp. 36–52.

Kitamura, Y., Yee, A., & Kishino, F. (1996). Virtual object manipulation using dynamically selected constraints with real-time collision detection. In *Proceedings of Symposium on Virtual Reality Software and Technology.* 173–181. ACM.

Kotoku, T., Takamune, K., & Tanie, K. (1994). A virtual environment display with constraint feeling based on position / force control switching. In *Proceedings of International Workshop on Robot and Human Communication,* 255–260. IEEE.

Noma, H., Kitamura, Y., Miyasato, T., & Kishino, F. (1996).

Haptic and visual feedback for manipulation aid in virtual space. In *Proceedings of Fifth Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems,* 469–476, ASME.

Smith, A., Kitamura, Y., Takemura, H., & Kishino, F. (1995). A simple and efficient method for accurate collision detection among deformable polyhedral objects in arbitrary motion. In *Proceedings of Virtual Reality Annual International Symposium,* 136–145. IEEE.

Snyder, J. M. (1995). An interactive tool for placing curved surfaces without interpenetration. *Computer Graphics, Annual Conference Series,* 209–218. ACM.

Sayers, C. P., & Paul, R. P. (1994). An operator interface for teleprogramming employing synthetic fixtures. *Presence: Teleoperators and Virtual Environments, 3*(4), 309–320. MIT Press.

Venolia, D. (1993). Facile 3-D direct manipulation. In *Proceedings of INTERCHI,* 31–36. ACM.