A Virtual Reality Based System for Exterior Design

Shoujie He[1], Yousuke Kawamura[2], Kazuaki Tanaka[2], Norihiro Abe[2], Jiangyu Zheng[2], and Hirokazu Taki[3]

[1]Dept of Computer Science and Electronics,
[2]Dept of Mechanical Systems Engineering,
Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology
Kawazu 680-4, Iizuka, Fukuoka 820-8502, JAPAN
[3]Mitsubishi Electric Inc. JAPAN
hesj@cse.kyutech.ac.jp

Abstract

3D CAD systems are widely used for the exterior design of industrial products in recent years. The user interface of the 3D CAD systems, however, is not yet friendly enough for the professional exterior designers. Learning to use the CAD systems is a tough task. Sometimes, even having mastered the usage, the designers still feel difficult to concentrate on the artistic aspects because the interaction with the target objects is not as intuitive as expected. This paper presents a virtual reality based system for exterior design, which enables the designers to manipulate and work with the virtual objects extremely intuitively. The input devices of our system include a data glove and a 3D-location sensor. The system design makes it possible for the designers to change the 3D shape of a virtual object in real-time.

Key words: Virtual Reality, Exterior Design, Head-Mounted Display, Data Glove, Magnetic Location Sensor

1. Introduction

By exterior design, we mean the appearance design of artificial objects such as cars, electrical appliances, and so on. The focus of the exterior design is on the external features but not the functional issues and the internal mechanisms. Exterior design can actually be considered a kind of artistic design. In the past, the professional designers accomplished their design work by using illustrations or models. The design process was very inefficient and most of the design efforts are not reusable.

CAD/CAM technologies have been widely used for the improvement of the efficiency with the product design and manufacturing. Furthermore, 3D CAD systems have been developed in recent years for dealing with the 3D shape data and thus an integrated design and manufacturing system is no longer a dream [1]. The 3D CAD systems, however, are designed for those who have not only design knowledge but also computer expertise. So far, the professional designers have worked with illustrations or models. Most of them have never had chance of using a computer. In order to change the shape of the computer-generated objects as they like, the professional designers need a lot of training for mastering the usage of a large set of shape changing tools. Many shape-modeling tools have been developed, with which 3D shape changing can be done either by specifying the numerical values of 3D vectors or through a 2D mouse input. Regrettably, neither of the shape changing methods is as intuitive as expected.

With the advancement of 3D computer graphics technologies and I/O equipments, virtual reality as a new enabling technology is drawing more and more attention from various fields. The virtual environment generated with the most advanced computers has not much difference from the real world in term of the visual effects. With the virtual reality technologies, professional designers could directly manipulate and change the shape of the computer-generated objects.

This paper presents a virtual reality based system which provides the professional designers with an intuitive means for the manipulation of the 3D shape of virtual objects. The rest of this paper is organized as follows. Section 2 describes the organization of our system. In Section 3, we describe in detail the modules for hand gesture recognition, interference detection, and shape changing operations. Section 4 shows the experimental results. Section 5 concludes the paper and discusses on the future work.

2. System Organization

As shown in Figure 1 and Table 1, our system consists of data input devices, a PC for generating a virtual environment and incorporating the input data into the virtual space, and a stereo device for visualizing the virtual space. In addition, World ToolKit [3] library is used for the generation of the virtual space. Designbase [1], a library for manipulating solid models, is also integrated.

Table 1. Specification of the Equipment

Data glove	DataGlove Model JV1 (Nissho Electronics Co.)
3D location sensor	ISOTRAK2 (POLHEMUS Co.)
VR PC	TD-300 (INTERGRAPH Co.)
Stereo device	CrystalEyes (StereoGraphics Co.)

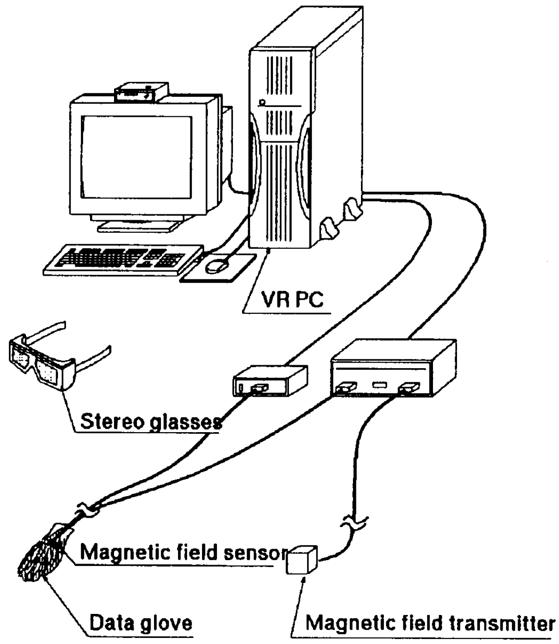


Figure 1: System organization.

3. System Design and Implementation

The fundamental requirement to the system design is to let users have an intuitive interaction with the virtual objects through a data glove and a 3D magnetic location sensor. To satisfy the requirement, we must keep the system informed which surface of the virtual object has been selected and what kind of shape changing operations will be applied to the surface. All these are done through the recognition of hand gestures and pre-defined types of the shape changing operations and the detection of the interference between the hand model and the object model. Figure 2 shows the overall interaction process for changing the 3D shape of a virtual object.

3.1 Hand gesture recognition

Hand gesture recognition module must be capable of recognizing hand gestures of all the potential users. Since the size of hands and the position of finger joints are different from people to people, it is very difficult to set up a set of general criteria for recognizing all the hand gestures. This problem can be solved by using a neural network [4]. With a pre-trained neural network, the recognition of the hand gestures of all the users becomes possible.

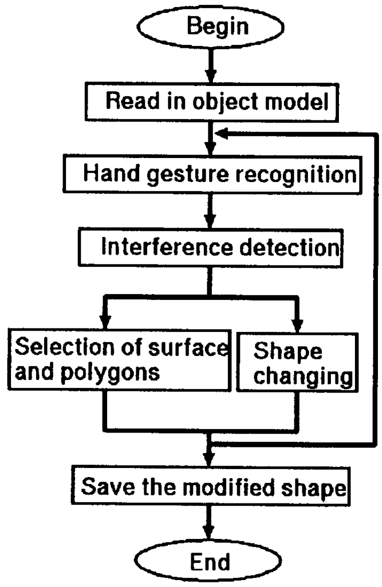


Figure 2: Overall interaction process for changing the shape of the virtual object.

Before addressing the recognition methods for the hand gestures, we need to define a set of meaningful hand gestures. Our hands could have a large set of different gestures but if all of the gestures are adopted and each of them is interpreted in a different way, system users will feel difficult to remember all of the gestures. Besides, the measurement technology with data gloves still has limitation on distinguishing two gestures with a subtle difference. Therefore, five basic hand gestures, which are shown in Figure 3 (a)-(e), are defined for the use in our system. The meaning assignment of the five hand gestures is as shown in Table 2.

To use a neural network for the hand gesture recognition, we need decide the input data for the input layer of the neural network first. The most straightforward way is to directly use the data measured from a data glove. However, our experiments have shown that if we directly use the measurement data, we need a neural network with more than one hidden layers and with a large number of nodes and thus the training process is very time-consuming. For this reason, we generate seven input data from the measurement data by following the computation procedures shown in Table 3. As a result, a single hidden layer will be enough for the recognition of the five basic gestures and of course the training costs are reduced remarkably.

As shown in Figure 5, the neural network for the hand gesture recognition is composed of seven nodes in the input layer, ten nodes in the hidden layer, and six nodes in the output layer. The number of the nodes in the hidden layer is experimentally determined through a trial and error process. Note that the number of the nodes in the output layer is six. Among the six nodes, five of them correspond to the five basic hand gestures shown in Figure 3. The remaining node corresponds to the gestures other than the five defined ones. This extra node is important because its existence enables the network not to react to the illegal gestures. Of course, we need train the network by using the possible illegal gestures as well.

Table 2. Hand Gestures and the Meaning Assignment

Hand gestures	Meaning assignment
Gesture 1	(1) Confirm the selection of surfaces (2) Cancel the selection
Gesture 2	(1) Select multi-surfaces at a time (2) Change the shape of the surfaces downwards (effect: a large region) (3) Push a set of surfaces down
Gesture 3	(1) Select a single surface at a time (2) Divide the surface (3) Change the shape of the surface downwards (effect: a small region) (4) Push a set of surfaces down
Gesture 4	(1) Change the shape of a surface upwards (effects: a small region is changed) (2) Draw up a set of surfaces
Gesture 5	(1) Change the shape of surfaces upwards (effect: a large region) (2) Draw up a set of surfaces

3.2 Interference detection

The detection of interference between two surface models in a virtual space is generally done with a coarse-to-fine approach [2]. The principle is to check roughly the interference between the bounding boxes of the models. Only if the bounding box interference has been detected, will the interference at the polygon level be checked in detail. Undoubtedly, such a coarse-to-fine approach is effective and efficient for the interference detection. Regrettably, the coarse-to-fine approach will still be computationally expensive as the 3D shape of the models becomes complicated. Although the hand model used in this research is not that complicated, a fast interaction with the design object is required and thus the interference detection with much more reduced computational costs is desired. In order to reduce the computational costs more efficiently, we take a unique strategy for the detection of interference between the hand model and the design object. The strategy is to apply the interference detection algorithms to'a limited set of polygons on the hand model. The limited set of polygons is determined by hand gestures. Only the polygons relevant to a given gesture are used for the detection of the interference with the design object. Table 4 shows the relevant polygon sets selected for the interference detection for each of the hand gestures.

Table 3. Neural Network Input Data

Node ID	Input Data Items
1	Sum of angles between neighboring joints on thumb
2	Sum of angles between neighboring joints on the index finger
3	Sum of angles between neighboring joints on the middle finger
4	Sum of angles between neighboring joints on the third finger
5	Sum of angles between neighboring joints on the little finger
6	Area of the triangle at finger tips (see Fig. 4)
7	Sum of the length of the sides of the Triangle

Table 4. Hand Gestures and the Relevant Polygon Sets

Hand Gestures	Relevant Polygon Sets for Interference Detection
Gesture 1	None
Gesture 2	Polygons on the index, middle, and third fingers
Gesture 3	Polygons on the index finger
Gesture 4	Polygons on the thumb and the index and middle fingers
Gesture 5	Polygons on the thumb and the index and middle fingers

3.3 Surface selection

To change the shape of a given object, we need to select a surface from the object. The selection is done through the detection of interference between the hand model and the design object. The selection could be a single surface or a set of surfaces. The basic idea is to use the defined hand gestures in Table 2 and their relevant polygon sets in Table 4. Let us explain the surface selection procedure with an example: the selection of a single surface. The selection procedure is as follows:

1. Make sure the hand gesture is Gesture 3.

2. Recognize the hand gesture with the pre-trained neural network.

3. Check the interference between the relevant polygons

with the polygons on the design object. If the interference is detected, go to step 5.

4. Move the hand model towards the design object and go to step 3.

5. Change the hand gesture to be Gesture 1 to confirm the selection. If the surface selected is a wrong one, take Gesture 1 once again to cancel the selection.

4. Shape Manipulation

4.1 Surface division

The curved surfaces are represented with parametric patches in this system. Therefore, the changes on the surfaces are represented by first generating curves and then interpolating the surface enclosed by the curves [1]. Figure 6 shows a procedure for the division of a surface. The division is done simply by pointing to a selected surface with Gesture 3. Note that a fine division can be achieved by repeating the procedure shown in Figure 6. The shape manipulation in our system is applicable not only to the border of surfaces but also the apexes inside a surface. In principle, we can divide a selected surface as fine as we like but the computational costs increase overwhelmingly as the division is repeated.

4.2 Shape changing operations

Shape changing operations provided in our system are as follows.

Shape revision operations: With the shape revision operations, the surface shape changes smoothly. This type of operations is realized by Gestures 2, 3, 4, and 5. As shown in Table 2, Gestures 2 and 3 are used to change the shape downwards and Gestures 4 and 5 are used to change the shape upwards. Note that only geometrical constraints are considered for the shape changing effects and no material factors have been integrated yet.

Currently, the following two types of shape revision operations are available in the system.

(1) **Shape revision to a small region** The operation is realized by applying either Gesture 3 or Gesture 4 to a set of selected polygons. As shown in Figure 7, with this operation a big shape change at a small region becomes possible. As depicted in Figure 8, this operation only changes the location of a particular set of the apexes and never affects their neighbors. The apexes are those that are in the selected surface and sufficiently close to the index finger. The closeness is measured with a preset threshold.

(2) **Shape revision to a large region** The operation is realized by applying either Gesture 2 or Gesture 5 to a set of selected polygons. As shown in Figure 9, with this operation a big shape change at a large region becomes possible. As depicted in Figure 10, this operation changes the location of not only the particular set of the apexes but also their neighbors.

Surface translation: With the surface translation, a set of polygons are drawn up or pushed down with the same movement. The intra-relationships among the polygons keep no change. This type of operations is realized by Gestures 2, 3, 4, and 5. As shown in Table 2, Gestures 2 and 3 are used to change the shape downwards and Gestures 4 and 5 are used to change the shape upwards.

Currently, the following two types of surface translation operations are available in the system.

(1) **Translation with direction unconstrained** As shown in Figure 11, this translation operation moves a set of selected polygons with the translation direction unconstrained.

(2) **Translation along a constrained direction** Figure 12 shows an example of translation along the direction perpendicular to the surface.

5. Experimental Results

Figure 13 shows a rectangular solid, which is the initial object for our shape changing experiments. Figure 14 shows the surface selection result. Figure 15 shows the surface division result. Figure 16 shows the polygon selection result. The remaining figures show the shape changing effects in accordance with each of shape changing operations.

6. Concluding Remarks

In this research, we developed a virtual reality based system for supporting professional exterior designers. Experiments show that using our system, an intuitive interaction with the virtual object is possible. Our current system is still at an infant stage. Only a limited set of primitive shape changing operations is equipped. For the future enhancement, we are considering (1) to add more primitive shape changing operations; (2) to develop some methods for the integration of the primitive operations; (3) to incorporate a speech recognition module for a more intuitive and more friendly user interface with the system.

References

1. H.Toriya and H.Chiyokura, 3D CAD Fundamentals and Applications, Kyoritsu Publishers, 1991.

2. E. Nakamae and T. Nishida, 3D Computer Graphics, Syokoudo Publishers, 1986.

3. World ToolKit Reference Manual, SENSE8, 1996.

4. J. Dayhoff and H. Katsurai, Neural Network Architecture, Morikita Publishers, 1992.

5. J. Yamashita and Y. Fukui, Direct Operations on Curves and Curved Surfaces, CG Symposium, 1992.

6. M. Hashimoto and K. Shinohara, Applying a Hierarchical B-Spline to Modeling Curved Surfaces in a Virtual Space, CG Symposium, 1992.

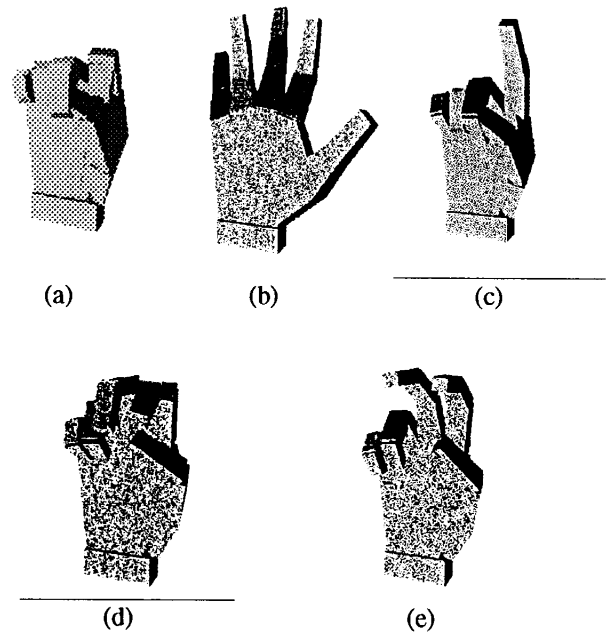


Figure 3: Hand gestures defined for the interaction with our system.

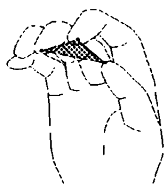


Figure 4: A triangle with its vertices located at the finger tips.

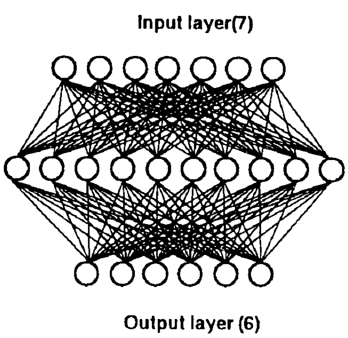
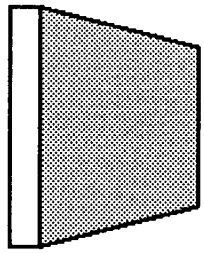
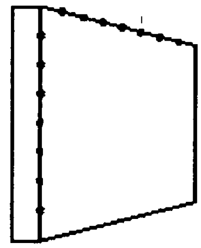


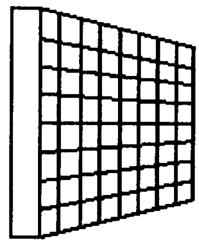
Figure 5: A neural network for recognizing the hand gestures.



(a) Surface selection.

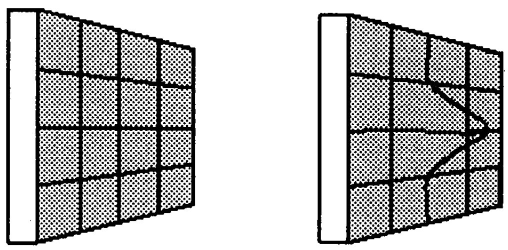


(b) Border division.



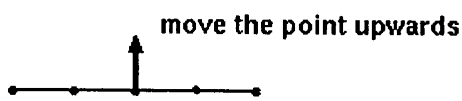
(c) Surface division along the apexes on the border.

Figure 6: Surface division.

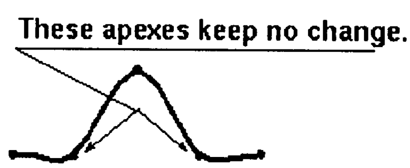


(a) (b)

Figure 7: Shape changing effect: a sharp change at a small area. (a) Prior to the shape changing operation; (b) Posterior to the shape changing operation.

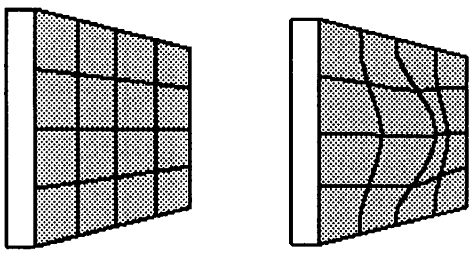


(a) Prior to the shape changing operation.



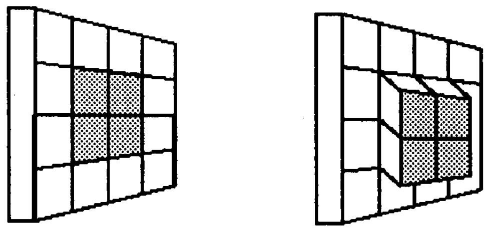
(b) Posterior to the shape changing operation.

Figure 8: Cross section views of Figure 7.



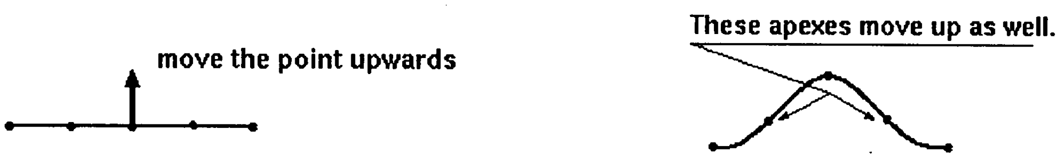
(a) (b)

Figure 9: Shape changing effect: a sharp change at a large area. (a) Prior to the shape changing operation, (b) Posterior to the shape changing operation.



(a) (b)

Figure 11: Surface translation with an unconstrained (a) Prior to the shape changing operation, (b) Posterior to the shape changing operation.



(a) Prior to the shape changing operation.

(b) Posterior to the shape changing operation.

Figure 10: Cross section views of Figure 9.

(a) (b)

Figure 12: Translation along a constrained direction. (a) Prior to the shape changing operation, (b) Posterior to the shape changing operation.

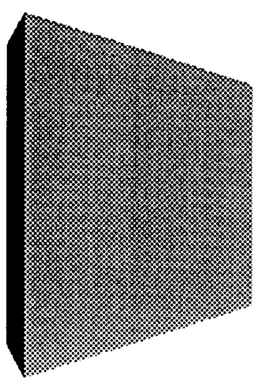


Figure 13: A rectangular solid for the shape changing experiments.

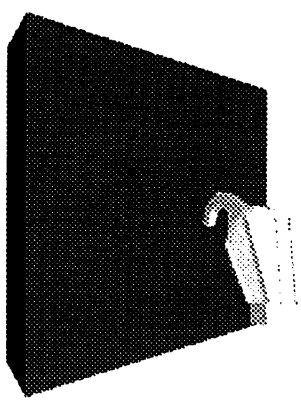


Figure 14: Surface selection.

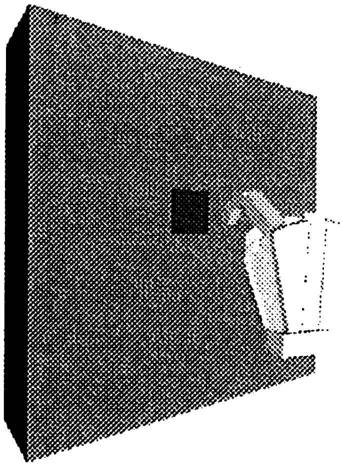


Figure 15: Surface division.

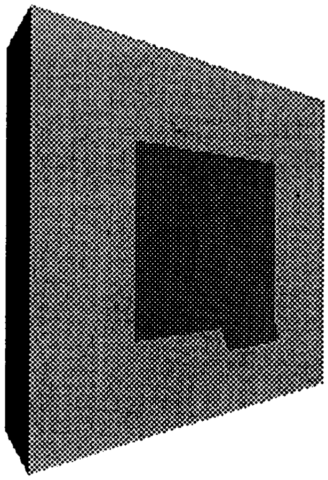


Figure 16: Polygon selection for shape changing experiments.

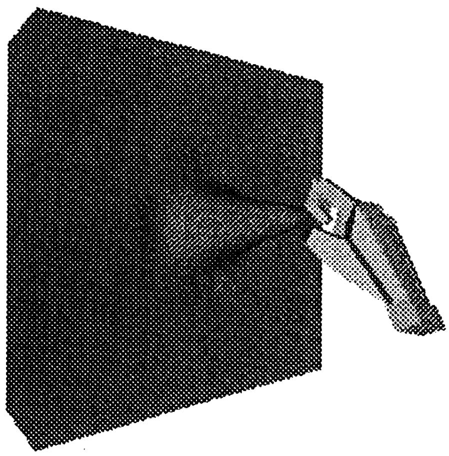


Figure 17: Shape modification to a small area.

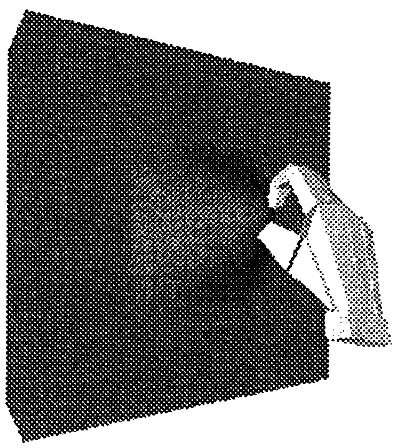


Figure 18: Shape modification to a large area.

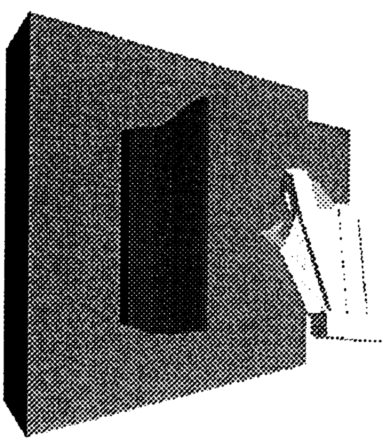


Figure 19: Translation towards an unconstrained direction.

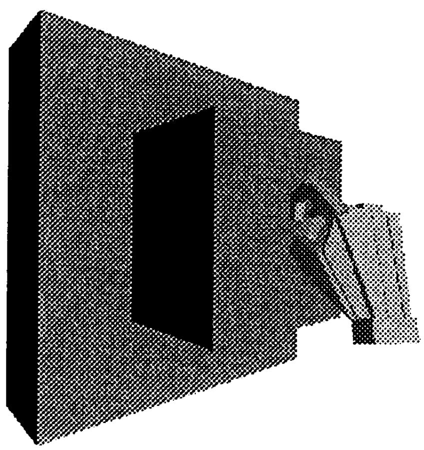


Figure 20: Translation towards a constrained direction.

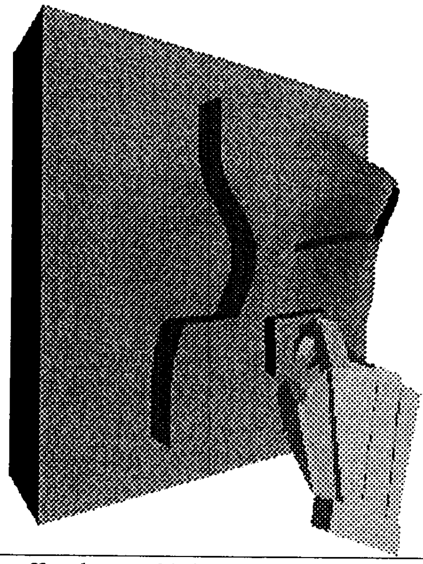


Figure 21: Shape changing effect by combining the primitive shape changing operations.