ICAT 2002
December 4-6, Tokyo, Japan

Extracting Relational Constraints among Objects With Haptic Vision Toward Automatic Construction of Virtual Space Simulator

Hiromi T.TANAKA^{*1}, **Kayoko YAMASAKI**^{*1}, **and Masaru YAMAOKA**^{*1*2}

^{*1}Computer Vision Laboratory, Department of Computer Science, Ritsumeikan University
^{*2} Matsushita Electric Industrial Co., Ltd.
hiromi, yacco@cv.cs.ritsumei.ac.jp

Abstract

Virtual Reality technology provides a new methodology for human interface with realistic sensation. Recently, haptic interface has been intensively studied for providing a sense of touch. Moreover, high-performance force-feedback displays also have been developed for realiziong haptic interface with the virtual environment. In this paper, we propose a novel approach to extraction of 2D relational constraints among objects under gravity based on haptic vision apploach, which was proposed for vision-based automatic construction of virtual space simulator. Haptic vision is based on active sensing and realtime image understanding methodology, which enables objects in the virtual space to behave, and change realistically with virtual force, and to be operated with a sence of real touch through haptic interface devices. We apply this technique to automatic construction of virtual indoor space simulator. Experimental results showed the feasibility of the proposed approach towards the observation-based automatic construction of virtual space simulator.

Key words: automatic construction of virtual space simulator, haptic interface, virtual object operation, haptic vision, relational constraints extraction among objects

1. Introduction

A virtue of Virtual Reality(VR) is in the fact that it can provide a user a realistic "experience " with realistic sensation. Recently, haptic interface has been intensively studied ([2]-[5]) for providing a sense of touch in virtual environment, which is an essential modality to explore, recognize and understand the real world. Real-world objects exhibit rich physical interaction behaviours on contact.

Such behaviours depend on how heavy and hard it is when hold, how its surface feels when touched, how it deforms on contact, and how it moves when pushed, etc. These aspects of visual and haptic behaviour provide important interaction cues for manipulating and recognizing objects in virtual environments. Reseach activities have been energetically advanced on haptic exploration to estimate and extract physical object properties such as mass, friction, elasticity, relational constraints etc. Thus, we have proposed a novel paradigm, we call haptic vision [6], which is a vision-based haptic exploration approach toward an automatic construction of reality-based virtual space simulator.

The work on physical object properties from interaction with robots had first introduced in the early 1980 [1]. While recent progress in haptic exploration with dextrous robots([2]-[4]), which requires complex robot control and grasping technique, we believe Haptic Vision is the first non-contact vision-based automatic approach for haptic exploration to model both geometrical and physical properties of real-world objects.

In this paper, we propose a novel approach to automatic extraction of 2D relational constraints among objects based on Haptic Vision, so that a reality-based virtual object manipulation can be simulated with haptic interface. Furthermore, we propose a method to describe 3D relational constraints among objects using hierarchical structure of 2D relational constraints extracted in such way.

As figure 2 shows, first the Haptic Vision system attached on a robot hand with range sensor observes the scene and aquires images from multiple viewpoints of an object in the scene based on the assumption that the objects are put on a level surface, i.e., a horizontal supporting plane. The system projects these images onto a 3D voxel space to generate a volume representation. Then, it decomposes the scene into 3D objects or parts using concave roof edges of the volume space to estimate free spaces of each object or part. Next, estimate the most suitable contact force and a contact point to detect the relational constraints between objects, and then we make such contact to the object using a robot hand. Finally, it extract a relation between objects and their free spaces to generate a hierarchically structured graph of 2D relational constraints. Note that we can limit the action of the robot hand into Push, since what we have to detect is the 2D relational constraints between solid objects on a horizontal supporting plane.

We then use these results to construct virtual object manipulation simulator. Experimental results show that feasibility and validity of the proposed approach.

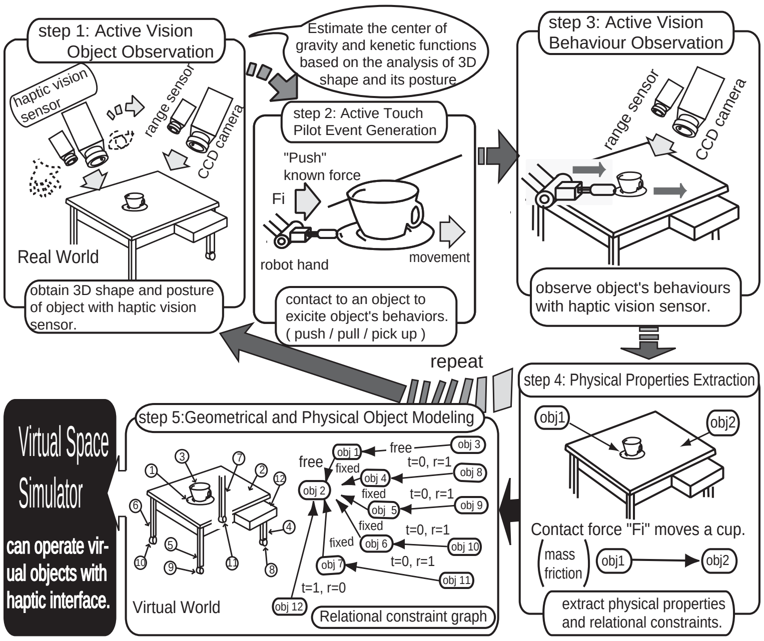


Fig. 1: Haptic Vision Approach

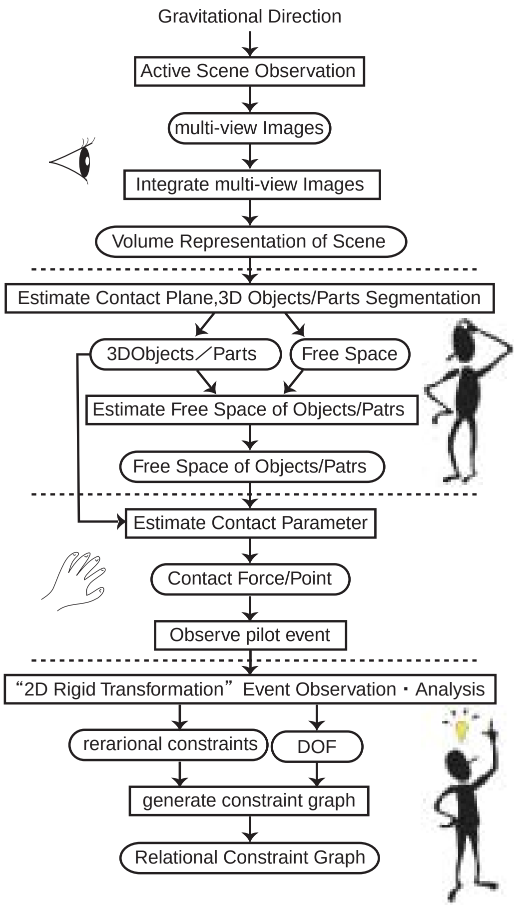


Fig. 2: Process Flow

2. Haptic Vision

In order to realize a virtual space simulation faithful to the real world, we have to enter huge volume of high quality information into the system. For example, even in a simple situation where a cup is placed on a desk shown in figure 1, we have to describe relation between objects contacting each other and how (and how far) the relation is maintained when we move the cup. This is true even when we want to simulate a simple operation such as picking up a cup or pulling out a drawer. Thus, we have proposed the idea of see-and-touch system, that is Haptic Vision which acquire haptic information to simulate a real object in a virtual space [6]. Haptic Vision is the evolutional form of Active Vision system that actively collects visual information [9], [10]. In our Haptic Vision, the concept of viewpoint control in Active Vision is expanded to contact and external force control to obtain information for the haptic interface. It means that our Haptic Vision is built by integrating a new concept, so to speak "active touch", onto the concept of Active Vision while enhancing such concepts.

As shown in figure 1, Haptic Vision system estimates geometric properties of an object such as 3D shape, surface texture, posture and center of gravity by observing it with a range sensor and a color camera (step 1). Here, the camera acts as eyes to take visual information in wide view angle and the range sensor acts as hands to know 3D structure of a neighboring object by touching it. Next, the system tries to cause a pilot event, that is the most effective event to know the physical properties and relational constraints, by applying a force to the contact point selected from the geometric properties estimated in the step 1 (step 2). Then, it analyzes the motion of the object (step 3) and extracts physical properties and relational constraints of the object and the support plane (step 4). Finally, it constructs up a model by generating a relational constraint graph in which objects are nodes and the contacting relation is expressed by an arc.

Haptic Vision is a system to acquire haptic information automatically by applying a force (pushing, drawing or pinching) to an objects after selecting the contact point and strength and direction of the known force, that are most adequate to obtain haptic information, by itself from the shape, posture and gravity center of the object.

3. Active Scene Observation

It is possible to obtain highly accurate range images in the realtime due to the recent improvement of range sensors [8]. Today, range sensors become far faster, smaller and lighter. This technology is also used in Haptic Vision system as an active vision with range sensor to acquire range images taken from multiple viewpoints to know stable and accurate 3D structure of an object.

3.1 Acquisition of Range Images Based on "Shape from Function" Approach

The method shown below is based on "Shape from Function" approach [11] that is applicable to an artificial object in indoor scenes which is homogeneous or uniform in material. This is a method to estimate stability of an object under the gravity against external force, that is most important for an object to perform a function, by observing it form multiple viewpoints based on the fact that the object is has a plane of symmetry.

First, the method estimates a plane of symmetry from a range image taken from the top view point to evaluate stability of the object against the gravity. Then, it takes four contour images of the object taken from left, right, front and rear. Thus we have five orthogonal images corresponding to the three base plane of the 3D world.

Figure 3 shows five orthogonal viewpoints to take images. First, we have to define a orthogonal coordinate system in which the origin is the intersecting point of the line of gravity including mass center of the Vtop image and laser calibrated plane, the z-axis is the direction of gravity and the y-axis is the normal line of the plane of symmetry (**S**). Thus, the observation point $\mathbf{V}_{top}$ can be defines as $(0, 0, H)$. Then, we define left, right, front and rear observation points from the maximum height (z_{height}), width in the x-axis (x_{width}) and width in the y-axis (y_{width}). Each observation point can be defined as $\mathbf{V}_{front}\,(H - \frac{x_{width}}{2}, 0, \frac{z_{height}}{2})$, $\mathbf{V}_{rear}\,(\frac{-x_{width}}{2}, 0, \frac{z_{height}}{2})$, $\mathbf{V}_{left}(0, \frac{-y_{width}}{2}, \frac{z_{height}}{2})$, $\mathbf{V}_{right}(0, H - \frac{y_{width}}{2}, \frac{z_{height}}{2})$, respectively.

Multi-view range images that are effective in shape recognition are acquired automatically by the recursive process shown above.

3.2 Generating Volume Representation of Scene

Next, we generate a 3D volume representation by projecting these multi-view range images onto the 3D voxel space assuming that 1) multi-view range images are transformed into orthogonal projection images, 2) all the concave parts of an object are observed from the multi-view range images and 3) there is no hole in the object.

See figure 4. First, define a ray parallel to the viewing angle for each pixel on a range image in the voxel space. Then define the value of a voxel into 0 (=free), that represents free space, if the ray goes through the voxel, and 1 (=body), that is inside of the object, for voxels from the object surface to the calibration plane (**S**). For jump edge points that lie in between the object and its background, set the value of a voxel to plus (+) if it is inside the object, that is convex edge, and minus (-) if it is in the side of background, that is concave or L-joint edge.

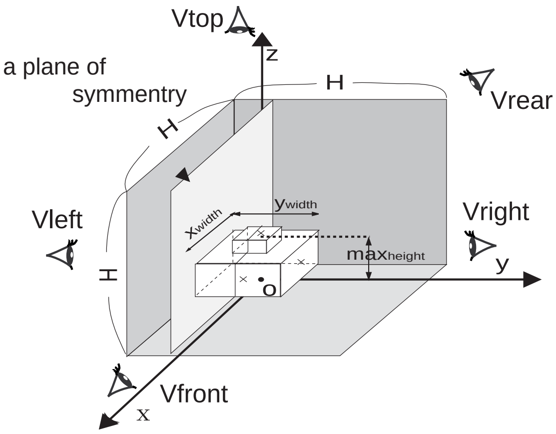


Fig. 3: Viewpoints from a Plane of Symmetry

4. Decomposition into 3D Objects or Parts

4.1 Estimating Contact Planes

In order to decompose the observed scene into each objects or parts, we have to estimate planes where each object can contact with other objects. We have adopted limb-based decomposition method [12], that is well coincide with the way in which a man guesses if they are contacting or not, and extract a plane surrounded by concave roof edge, that shows the L-joint, as a candidate contact plane. Since we have limited a support plane to level or horizontal plane, all the planes thus extracted are level planes. Note that the object bodies, jump edges and convex and concave edges are extracted by the parallel image

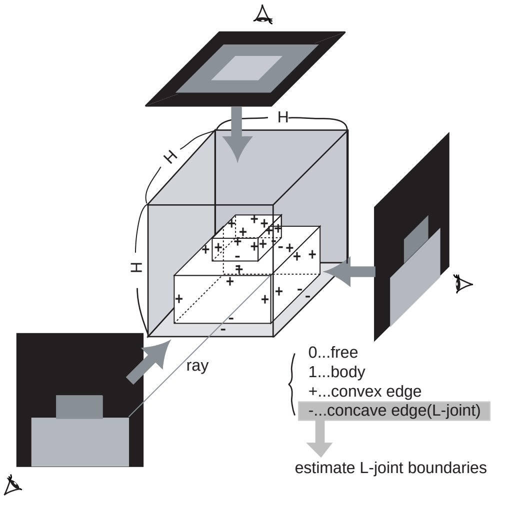


Fig. 4: Volume Data Reconstruction from Multi-view Range Image

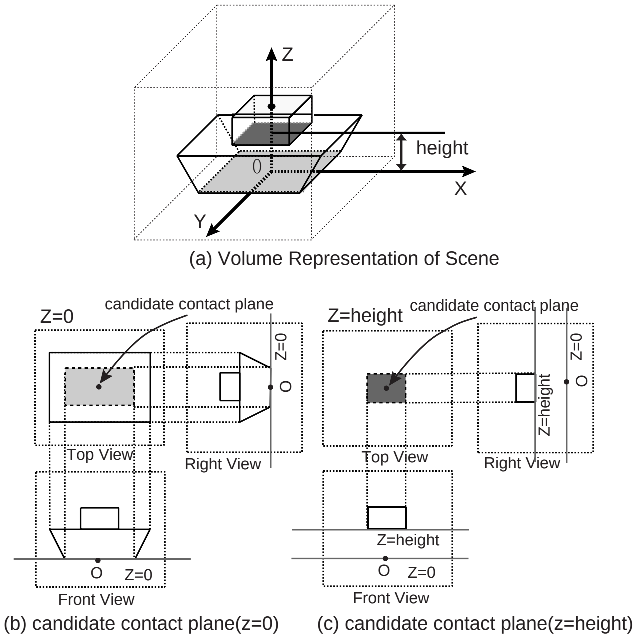
segmentation method using an adaptive mesh [13].

As shown in figure 5, first level (bottom) contact plane is extracted by concatenating voxels in the concave edges lying on the floor (collection of voxels of which x components are zeros) of the volume represented scene. In the same way, next level contact plane candidate is detected by slicing the volume representation of the scene horizontally and concatenating voxels in the concave edges. If the concatenation gives us a closed figure, it is the next candidate.

4.2 Initializing 3D Scene Graph

We have to decompose the set of body voxels (voxel value = 1) into objects using the volume representation of the scene and contact plane candidates and initialize the relational constraints graph that shows the relation of objects adjacent each other.

See figures 6, 7 and 8. The graph shows adjacency condition of two objects in a horizontal plane with a bidirectional arc and the hierarchical relation of supporting in the direction of garavity with a unidirectional arc. Note that it is possible to represent mutual constraints in the 3D space with a set to graphs showing 2D relation (mutual constraints) of objects that are hierarchically arranged. The root node (FL) in the graph shows a floor, and other nodes are objects. Each node has its properties such as 3D shape, mass center, weight, volume, density, friction coefficient and elastic coefficient. The unidirectional arc between the parent and child nodes shows the supporting relation in the direction of gravity and it also has its properties such as kind of constraint, flexibility, shape of contacting plane and mass center. If a parent node has multiple child nodes, it means that the multiple objects are on top of parent object. The bidirectional arc between brother nodes shows the adjacency relation on a horizontal plane. All the leaf nodes are connected to the "Free" node, which shows a free space, meaning that they can be picked up. The graph of the scene shown in figure 6(a) is initialized as figure 6(b) leaving the degree of freedom of constraint and flexibility of linear and rotary motion in abeyance.

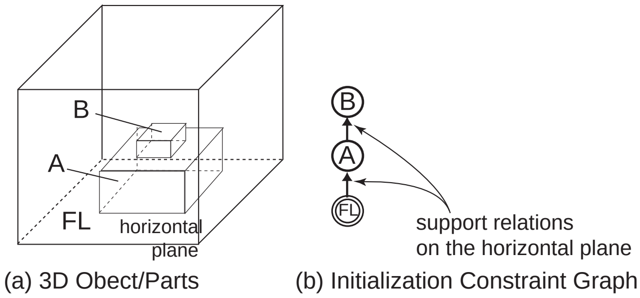


(a) Volume Representation of Scene

(b) candidate contact plane(z=0)

(c) candidate contact plane(z=height)

Fig. 5: Contact Plane Extraction



(a) 3D Obect/Parts

(b) Initialization Constraint Graph

Fig. 6: Initialization of Constraint Graph

4.3 Estimating Free Space for 3D Objects

In order to manipulate an object, we have to estimate a free space for each object. This estimation is done by selecting a pair of objects in parent-child relation and checking up a supporting plane of the parent object. The following two cases may be assumed for each supporting hierarchy.

- Only one object exists in a supporting plane
  If only one object exists in a supporting plane as shown in figure 7, horizontal supporting plane of the parent node one level lower in the direction of gravity will be the free space for this object provided that the object may not fall down. In this case the free space for object B is the top plane of object A, and all the image area is the free space for object A since there is no object below it.

- Multiple objects exist in a supporting plane
  If multiple objects exist in a supporting plane of a parent as shown in figure 8, we have to set a free space so that the motion of one object may not affect the other object(s). In this case, objects are contacting each other at the jump edge. Therefore, if we divide the whole free space into two parts with a plane including the jump edge, one half of such space including the object is completely free for that object, that is the motion in this half dose not affect other object provided that there are only two objects on the supporting plane.

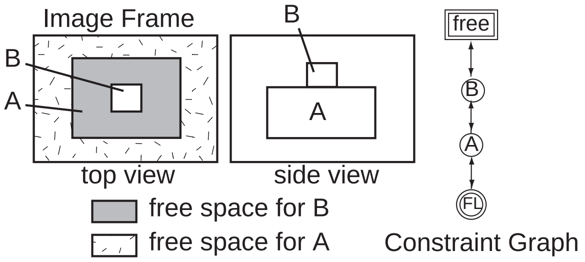


Fig. 7: Free Space Estimation for Nested Object

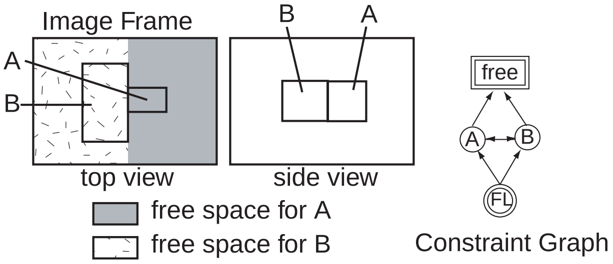


Fig. 8: Free Space Estimation for Adjacent Objects

5. Extracting Relational Constraints Based on Haptic Vision

5.1 Kind of Relational Constraints and Its Priority

Degree of freedom (DOF) that shows the constraints between rigid objects on a horizontal support plane may be 0, 1 or 2 for translation, and 0 or 1 for rotation. Figure 9 shows various constraints for the set of DOFs. In the case of free support type, the DOF of an object will be 3 for the application of external force. If the contact is the type of rail such as drawer of a desk or window on a rail, the DOF of such object is 1. The contact of axis support type is a contact with rotational DOF of 1 only, that is the object can rotate but not allowed other movement. Cup on a saucer is an example of this type of contact. Note that we can hardly imagine a contact with translational DOF of 2 and rotational DOF of 0, or translational DOF of 1 and rotational DOF of 1. The priority shown in the table relates to how many Push operations are required to detect the relation or constraints between objects. In the cases of non-contacting support or axis support contact, we can set the contact point so that only one Push operation can detect the type of contact. This is the reason why the priority is the highest. However if you want to detect rail type contact, you have to try multiple Push operation. If there are multiple objects on a plane, you have to carefully select directions of multiple Push operation. Thus, the priority is lower for these cases.

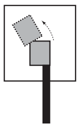
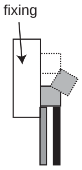
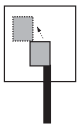
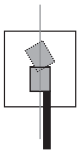
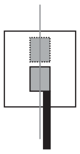
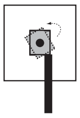
type	free	bounded	—	—	rail	axis
movement	free	partially free	—	—	single direction	rotation along centroid
degree of freedom of translation	2	2	2	1	1	0
degree of freedom of rotation	1	1	0	1	0	1
priority	1	3	—	—	2	1
translation and rotation of "push"						

Fig. 9: Classification of Support-Constraints

5.2 Causing "Slide Event" with Push Operation

Before pushing an object with a robot hand, we have to detect a contact point and strength of a force to be applied from the constraints graph and free space defined in the way stated above. Before performing first push operation, all the relations are presumed to be a type of non-contacting support according to the priority explained in section 5.1. If the relation between an object and a support plane is non-contacting, translational DOF of the object is 2 and rotational DOF is 1, that is the object can be moved freely in the free space.

First the system selects the contact point at which the moment becomes maximum to cause maximum translational and rotational motion. If the object is supported without any contact, we can detects total 3 DOF of the object, translated into 2D solid object motion, by one Push operation. If the object is supported by an axis, we can detect it from the fact that translational DOF is 0. If the system cannot detect the DOF of 3, it selects another contact point and a direction of a force according to the priority explained in section 5.1 based on the 3D shape and free space of the object.

5.3 Observing "Slide Event"

The system observes the sliding motion of an object caused by the force application robot in the realtime using th observation robot. The observation is done from the point perpendicular to the object where the event can be observed most stably.

We can derive 2D rotation matrix $\mathbf{R}$ and translational matrix $\mathbf{t}$ from the motion of two or more characteristic points on the object. However, motion may push the characteristic points out of the observable area, since detection range of a range sensor is smaller than that of a camera. Therefore, we have adopted a CCD camera and tracked four points on the object in the way shown in figure 10. If the positions of points before Push operation are $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3), P_4(x_4, y_4)$, and points after Push operation are $P_1'(x_1', y_1')$, $P_2'(x_2', y_2')$, $P_3'(x_3', y_3')$, $P_4'(x_4', y_4')$, the rotation matrix $\mathbf{R}$ and the vector of translational motion $\mathbf{t}$ will be;

$$[\mathbf{Rt}] = \mathbf{A}'\mathbf{A}^{\mathbf{T}}(\mathbf{A}\mathbf{A}^{\mathbf{T}})^{-1} \tag{1}$$

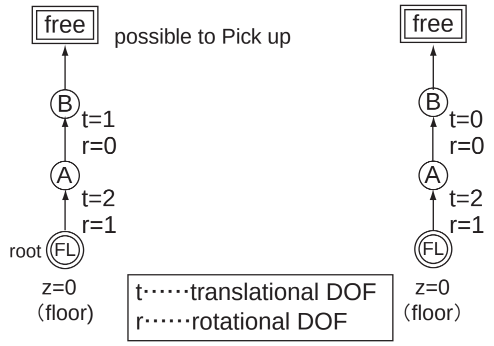
where

$$\mathbf{A} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{A}' = \begin{bmatrix} x_1' & x_2' & x_3' & x_4' \\ y_1' & y_2' & y_3' & y_4' \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The system updates the properties of relation and DOF of each arc in the graph using data thus obtained. Figure 10 shows graphs of rail type contact and the case if object B sticks to object A, that is one solid object.

6. Experiment

We have experimented our algorithm using Haptic Vision system. The force application robot is Mitubishi Electric's robot manipulator RV-E2 equipped with CKD's range finder called Cubicscope. The force applied to an object by the push operation is 19.6 kg. The resolution of Cubicscope is 512 by 242 pixels (Δx = about 0.2 mm, Δy = about 0.5 mm). The observation robot is Mitubishi Electric's robot manipulator RV-E4N equipped with Toshiba's CCD camera and controlled by DELL's PC (CPU: Pentium II 400 MHz, OS: Windows 98) through RS-232C port. Cubicscope is controlled by its dedicated control software called NetCubic on Gateway's PC (CPU: PentiumPro 120 MHz, OS: MS-DOS 6.2) through TCP/IP protocol.



(a) separate object(rail type) (b) single object

Fig. 10: Relational Constraint Graph

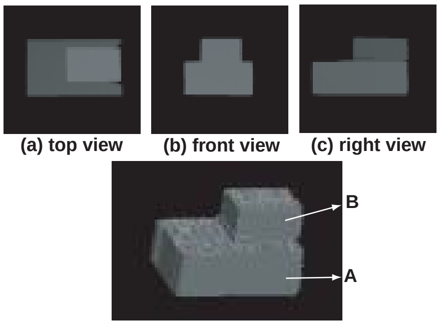
Diameter of marks on an object is 5 mm and the system observes the motion of its center. We have adopted 5 mm as the threshold for linear and rotational motion of objects from the size of marks. The DOF in relative constraints is detected by the difference of moved distance of marks between the objects that is applied external force and the supporting object one level below in the relation graph. The results are as follows.

6.1 Single Block or Adjacent Two Blocks

Object(s) in a scene may be a single block or two blocks contacting each other even if they have the same appearance, that is they have the same volume representation. This difference can be detected by applying an external force. Figure 11 (a), (b) and (c) are range images from $\mathbf{V}_{top}$, $\mathbf{V}_{front}$ and $\mathbf{V}_{right}$, respectively. These are range images that can be projected by conforming their scales using the maximum length of an object in x and y axes direction. Figure 11 (d) is the volume data of the object estimated from these range images. The robot pushes an object at the point where the moment becomes maximum after estimating the contact planes from concave edge in the volume data. Events caused by applying external force that are observed by the CCD camera are shown in figure 12 (in case of single block) and 13 (in case of contacting two blocks). Note that figures marked (a) are the states before applying force and figures marked (b) are the states after that.

6.2 Relation Between a Cup and a Saucer

Figure 14 shows the result of experiment on a model of a cup and a saucer. The model of the cup does not have a cavity (completely filled) in order to make it easy to acquire range image. Figure 14 (a), (b) and (c) are range images of this model and 14 (d) is its volume data. The system has detected the relation as shown in figure 14 (e). Rotation by an external force observed by the CCD



(a) top view **(b) front view** **(c) right view**

(d) Volume Representation

Fig. 11: 3D Volume from Multiview range images

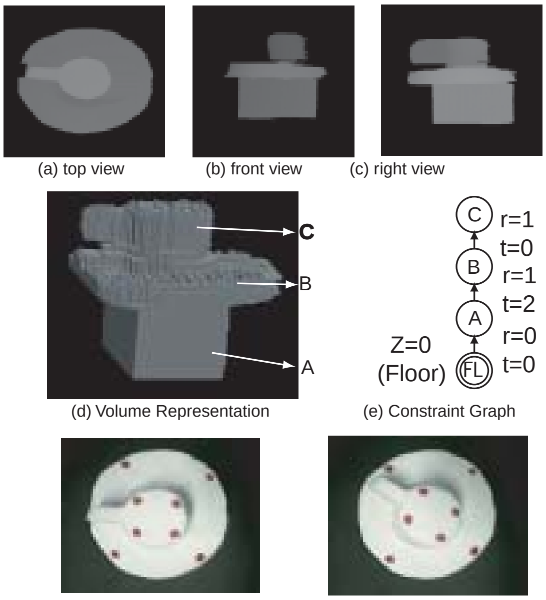
(a) Initial State **(b) After Push** **(c) Constraint Graph**

Fig. 12: Push Effect on One Object

(a) Initial State **(b) After Push** **(c) Constraint Graph**

Fig. 13: Push Effect on Separate Objects

camera is shown in figures 14 (f) and (g).



(a) top view (b) front view (c) right view

(d) Volume Representation (e) Constraint Graph

(f) Initial State (g)Change After "Push"

Fig. 14: Experimental Result (Coffee Cup)

6.3 Virtual Object Manipulation with Haptic Interface

We have built up a simulation system in which a user can operate a virtual object using the relational constraints data as input. Figure 15 shows the system configuration of this simulator. The 3D position sensor is Phemus's 3SPACE FASTRAK and the data capture glove is Virtual Technologies' CyberGlove. The system also uses Virtual Technologies' CyberGrasp as a force feedback device. They are controlled by Virtual Technologies' CyberGrasp Force Control Unit (FCU) which in tern is controlled by WindowsNT workstation (CPU : Intel Pentium III 800MHz (133MHzx6) x2 Dual Processor). 3SPACE FASTRAK and CyberGlove are connected to RS-232C ports and CyberGrasp is connected by a dedicated cable. In the system, the data capture glove is equipped with CyberGrasp with a receiver of 3D position sensor. Two kinds of information are entered into CyberGrasp; 1) properties of an object such as number, shape, mass, stiffness and 2) interaction information such as friction, relational constraints, gravity and acceleration. Note that the whole virtual space can be expressed by an tree diagram in which objects in the space are nodes. We have make the system acquire number of objects, 3D shapes (normal line of a plane) and relation between objects by itself from the graph of relation and property values of nodes. Stiffness is set to the maximun since the objects are solid. Mass of an object is calculated from its friction coefficient already known according to the related research carried out by Abe [8]. Virtual Technologies' libraries bundled with the FCU are used to obtain 3D position, contact determination and force feedback control. A software using OpenGL programming interface is also used for visual representation. Figure 16 (a) shows the state before grasping a block on a floor and 16 (b) after that according to the graph of relational constraints.

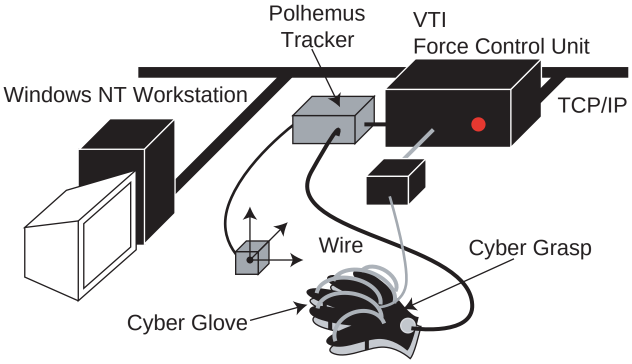


Fig. 15: Virtual space Simulation System

Fig. 16: Haptic-Interface based Virtual Object Manipulation System

7. Conclusion

we have proposed a novel approach to automatic extraction of 2D relational constraints among objects based on Haptic Vision, so that a reality-based virtual object manipulation could be simulated with haptic interface. Furthermore, we have proposed a method to describe 3D relational constraints among objects using hierarchical structure of 2D relational constraints

The experimental results demonstrated that we could exrtact the relational constraints between solid objects sliding on a supporting plane. Moreover, we have confirmed that the relational constraints thus extract can be effectively used as input information to a virtual space simulator to realize the feel of manipulating real object through the haptic interface.

References

1. E. Krotkov, "Perception of Material Properties by Robotic Probing: Preliminary Investigations ," Proc. Intl. Joint Conf. Artifical Intelligence, Montreal, August, pp. 88-94, 1995.
2. A.M. Okamura, M.L. Turner and M.R. Cutkosky, "Haptic Exploration of Objects with Rolling and Sliding," Proc. the 1997 IEEE ICRA, Vol. 3, pp. 2485-2490, 1997.
3. P.K. Allen and P. Michelman, "Acquisition and interpretation of 3-d sensor data from touch," IEEE Trans. on Robotics and Automation, 6(4), pp. 397-404, 1990.
4. A.M. Okamura, M.A. Costa, M.L. Turner, C. Richard, and M.R. Cutkosky, "Haptic Surface Exploration," Experimental Robotics VI, Lecture Notes in Control and Information Sciences, Vol. 250, Springer-Verlag, pp. 423-432, 2000.
5. D.K. Pai, J. Lang, J.E. Lloyd, and R.J. Woodham, "Acme, a telerobotic active measurement facility," Experimental Robotics VI, Lecture Notes in Control and Information Sciences, Vol. 250, Springer-Verlag, 2000
6. Hiromi T.TANAKA and Kiyotaka KUSHIHAMA, "Haptic Vision - Vision-based Haptic Exploration - ," Proc.the 2002 IEEE ICPR.
7. Masaru YAMAOKA, Kayoko YAMASAKI and Hiromi T.TANAKA, "Extracting Relational Constraints among Objects with Haptic Vision," IEICE Transactions, D-II, vol.J84-D-II,No.7,pp.1439–1447,2001. (in Japanese)
8. Kazuyuki HATTORI and Yukio SATO, "Handy Rangefinder Using Laser Scanning Space-Encoding Method," IEICE Transactions, D-II, vol.J-76-D-II,No.8,pp.1528–1535,1993.
9. D. H. Ballard, "Reference frames for animate vision," Proc. Int. Joint Conf. Artificial Intelligence, pp.1635–1641, 1989.
10. Y. Aloimonos, "Active vision revisited," Active Perception, pp.1-18, Lawrence Erlbaum Associates, Pub, 1993.
11. Kengo Nishimura and Hiromi T.TANAKA, "Active shape inferring based on the symmetry in stable poses — shape from function approach —," Proc. of the 13th ICPR, vol.I-A, pp.136–140, Aug., 1996.
12. Kaleem Siddiqi and Benjamin B.Kimia, "Parts of Visual Form:Computational Aspects," IEEE PAMI,Vol.17, No.3, pp.239–251, March, 1995.
13. Sang-Sun LEE, Hiromi T.TANAKA, "Parallel Image Segmentation with Adaptive Mash," IEICE Transactions, D-II, vol.J82-D-II,No.7,pp.1171–1179,1999.