ICAT 2003
December 3-5, Tokyo, Japan

Analysis by Simulation Approach to Inferring Functions of Articulated Objects

Hiroyoshi Ota, Asako Kimura, Hiromi T.Tanaka

Department of Computer Science, Ritsumeikan University
{*hiroyoshi, asa, hiromi*}*@cv.cs.ritsumei.ac.jp*

Abstract

Recently, there are growing needs for haptic exploration to estimate and extract physical object properties such as mass, friction, elasticity, and function, etc.. In this paper, we propose a novel "analysis by simulation" approach to inferring function of articulated objects, through reality-based interactions for function tests in virtual environments. Function inferred as a result of virtual function tests should be consistent with that inferred in the real world, because such interactions are simulated and generated according to: 1)the *real* intension and *real* action taken by the agent, and also 2)the *real* geometrical and *real* physical object properties obtained from observation of their behaviors in the real world. We applied the proposed method to inferring multiple funciotns of a pair of nutcrackers and a pair of tongs, such as "pinch", "hold", "carry" and "crash" etc.. Experimental results demonstrate the vilidity, effectiveness and performance of the proposed approach.

Key words: Inferring Fuction, Function Extraction, Haptic Vision, Virtual Space Simulation, Analysis by Simulation

1. Introduction

Real-world objects exhibit rich physical interaction behaviors on contact. Such behaviors 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 behaviors provide important interaction cues for manipulating and recognizing objects in virtual environments. Thus, there are growing needs for haptic exploration to estimate and extract physical object properties such as mass, friction, elasticity, function etc..

In the field of psychology, haptic has long been investigated by analyzing function and motion of a human hand such as active touch, grasping, etc.. EPs (Exploratory Procedures) by Lenderman and Klatzky is the landmark in this field [1]. EPs are highly stereotype motion patterns of a human hand for obtaining information about an object, as shown in Fig1. EPs consists of six patterns for extraction of texture, rigidity, temperature, weight, volume and shape, and two patterns for test and extraction of function and parts motion respectively.

Toward the goal of *automatic* haptic exploration, i.e., to automatically acquire physical object properties obtained by EPs, we have proposed a vision-based haptic exploration approach, we call Haptic Vision [2] [3], and have applied it to extraction of shape and volume [4], weight [5], relational constrains [6], viscoelasticity [7] and also applied to active modeling of the dynamic structure of articulated objects [8].

In this paper, we propose a novel "analysis by simulation" approach to inferring functions of articulated objects, through reality-based interactions for function tests in virtual environments. Function inferred as a result of virtual function tests should be consistent with that inferred in the real world, because such interactions tests are simulated and generated according to: 1)the *real* intension and *real* action taken by the agent, and also 2)the *real* geometrical and physical object properties obtained from observation of their behaviors in the real world.

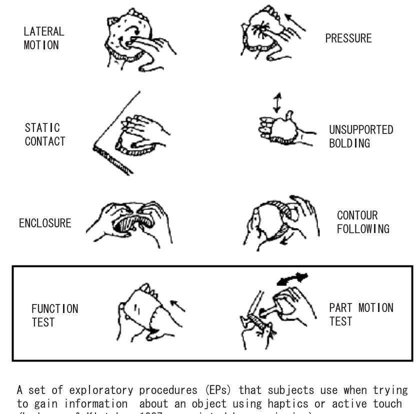

Fig. 1: EPs(Exploratory Procedures)

We have developed three kinds of reality-based interactions among an agent(a human hand), an acting object and an actee (an intended recipient), based on the physical object model acquired from observation of its behaviors in the real world with Haptic vision. Function inferring processes are generated and proceed as state transition of mutual relations among the agent, the acting object and the actee, and their conditions. Function is then inferred from the final state reached through a series of mutual interactions.

We applied the proposed method to inferring multiple functions of a pair of nutcrackers and a pair of pliers, such as "pinch", "hold", "carry" and "crash" etc.. Experimental results demonstrate the vilidity, effectiveness and performance of the proposed approach.

2. Why Analysis by Simulation Approach?

2.1 Related Work on Function Recognition

Researchers in the fields of artificial intelligence and computer vision have long made effort toward automatic function recognition and reported that 3D shape of the object is vital cue to estimate various object function [9]– [12]. Researches were first focused on static function recognition of such objects as a desk or a chair [9], then were extended to dynamic function recognition of rigid objects such as a cap opener or a screw wrench, where attention was paid on the importance of the actee with respect to the acting object [10]. Dynamic functions of articulated objects such as a pair of scissors were discussed from 3D shape analysis, however, interactions between objects were not considered [11]. Recently, motion-based approach was proposed to constrain the relation between the agent and the acting object by analyzing motion on the acting object by the agent. However, recognition of functions were not complete because changes on the actee were not observed from aquired images [12].

These research results suggest that the importance of simultaneous observation of mutual relations among the agent, the acting object and the actee, and their conditions toward robust and reliable function recognition. However they also suggest that simultaneous and non-contact observation of their behaviors are very hard to implement in the real world.

2.2 Our Motivation to "Analysis by Simulation" Approach

Contrarily to the difficulties in the conventional approaches in the real world, "analysis by simulation" approach could provide us reality-based function tests of the object through interactions in the virtual environment. Function inferred as a result of virtual function tests should be consistent with that inferred in the real world, as long as such interactions for function tests are simulated and generated according to: 1)the *real* intension and *real* action taken by the agent, and also 2) the *real* geometrical and physical object properties obtained from observation of their behaviors in the real world.

Thus, advantages of the analysis by simulation approach are,

(1) An agent can arbitrarily choose an actee according to the agents intension, moreover the attributes of the actee are arbitrarily defined. Such variety of interactions among arbitrarily defined objects provide effective and wide range of function tests.

(2) The conditions and motions, i.e., action and reactions, of the agent, the acting object and the actee are all recorded as *complete* information of 3D geometry and force for reliable analysis and inference.

(3) Given a reality-based physical object model, all possibilities such as possible contacts points, contact force, and actees can be computed and displayed to the agent as guides to provide efficient function inferring.

(4) The simulation system is capable to find "functional improvisation", since all possible simulations can be implemented and suggested to the agent.

3. Active Modeling of Articulated Objects with Haptic Vision

3.1 Levers as Functional Primitives of Articulated Objects

A lever is the most basic element to generate dynamic function of an object, and most researches on dynamic function recognition have considered objects having function of a lever [10] [11]. Based on these researches, we assume that articulated objects having dynamic functions such as "pinch", "hold", "cut", "crash", etc. have a lever as a functional primitive.

Generally, levers are categorized into the following three types Type 1, Type 2 and Type 3, as shown in Fig.2 , according to which of the points of *emphasis*, *application* or *fulcrum*, becomes the center of the lever [13]. Lever of type 1 can be further classified into three types; 1) magnifies power and attenuates distance, 2) attenuates power and magnifies distance, and 3) balances power and distance, as shown in Fig.2, as lever patterns of (a), (b) and (c).

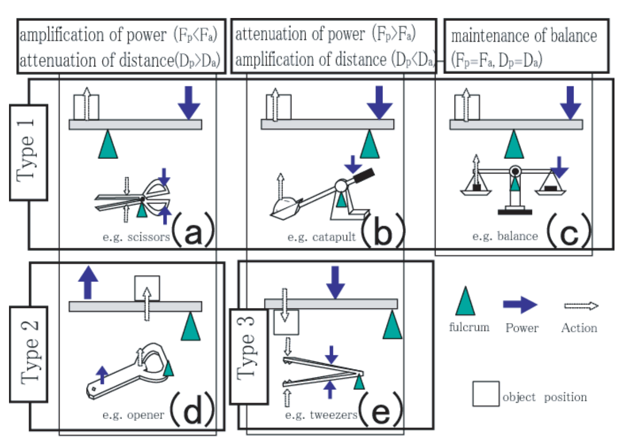

Fig. 2: Lever Patterns

3.2 Steps for Reality-based Object Modeling

We briefly describe steps in active modeling of an articulated objects with Haptic Vision [8]. Figures 3 and 4 show the structure and stable pose for functioning in the real world, we call Functional Pose, the representation of an articulated object to be generated through the following steps.

(1) Acquire range images of multiple viewpoints by observing an object placed on the horizontal plane using a range sensor with active vision.

(2) Project the acquired range images onto the 3D voxel space to generate volume representation *Ov* of the object, and estimate the center of gravity *Og*. The 3D surface shape *Os* is also reconstructed from the range images.

(3) Extract planes of symmetry *Osp* and obtain the cross section shape *Ocross* on the plane of symmetry *Osp* and extract a line of symmetry of *Oaxis* of *Ocross*, to estimate the stable pose for functioning in the real world, as shown in Fig.3.

(4) Analyze symmetry of the cross section shape to estimate contact points *Pemphasis* on the contact plane *Lcp*, and a support plane *Lsp*. Contact force *Fcontact* is estimated in order to cause rotational behaviors of a lever of the object most effectively and stable. Interaction space *Lis* is also extracted from the volume representation *Ov*.

(5) Actively contact to the object and exert the estimated contact force *Fcontact* at the estimated contact point *Pemphasis* with a robot hand.

(6) Observe the rotational motion of the lever with possible pattern of (a),(b),(c),(d) and (e) in Fig.2. Then extract the dynamic function information: the minimum and maximum rotation angles, $L\theta_{min}$ and $L\theta_{max}$ estimate the position of a fulcrum using a range sensor and CCD camera.

(7) Observe the transition of the contact force *Fcontact* using a force-feedback sensor mounted on the robot hand. Then estimate a spring constant *Sk* and a spring position *Sp* at the fulcrum position *Lf*.

(8) Finally, describe the shape, the staple pose for functioning, the force, the lever, and the spring information in the 3D model of an articulated object, as shown in Fig.4.

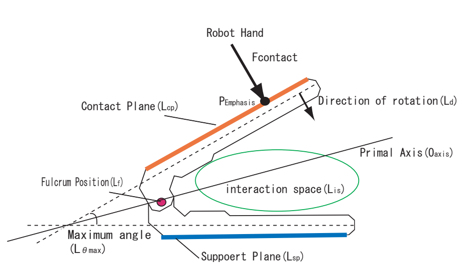

Fig. 3: Functional Pose of Articulated Object

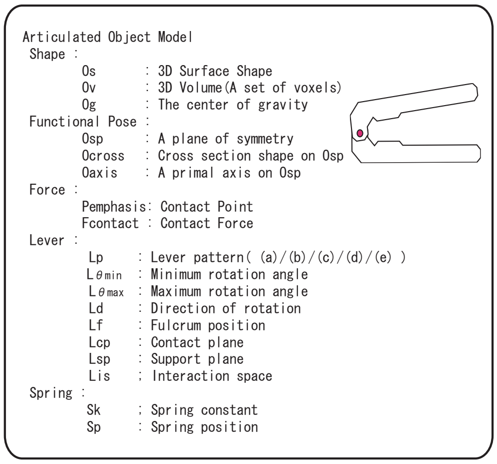

Fig. 4: Model of Articulated Object

4. Analysis by Simulation

4.1 System Architecture

The analysis by simulation system can be divided into three components: a motion capture subsystem, a graphics subsystem and a haptic subsystem. Fig.5 shows the system architecture of our simulation system. In the motion capture subsystem, the 3D position of every finger is measured in real time by a 3D position tracker (3SPACE FASTRK by Phemus Co.) and a glove-type a 3D position tracker (CyberGlove by Virtual technologies a Co.). The force feedback control is programmed using a library provided by Vti Co. In the haptic subsystem, haptic feedback to each finger is displayed through a glove-type haptic device (CyberGrasp by Virtual technologies Co.) fixed on CyberGlove. In the graphics subsystem, all movement of CG models are programmed using OpenGL library. These tree subsystems are controlled on WindowsNT machine(CPU: Intel Pentium 800MHz(133MHz × 6) × 2 Dual Processor).

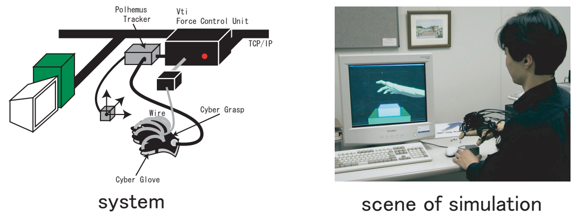

Fig. 5: Sytem Configuration

4.2 Modeling

4.2.1 Acting Object

An object model of the acting object has already defined at chapter three (Fig.4).

4.2.2 Actee

Condition of the actee is calculated by the force from the acting object to it. Since the purpose of this paper is to evaluate the feasibility of inferring the acting objects' function, we defined a very simple model for the actee. Fig.6 shows the actee model, where *As* is 3D shape of the actee, *Ad* is the force direction which effect on the actee condition, and *Apv* is predefined value which is the limitation of force magnitude for the actee, as shown in Fig.6.

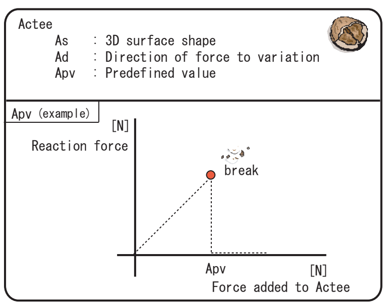

Fig. 6: Actee Model

4.2.3 Agent

The agent is a subject that operates acting objects. In this paper, we condider human hand as an operating agent. There is variety of hand shapes, however, since a human ordinarily "grasp" operating objects, the action by the agent could be limited to "grasp" action. Cutkosky and Howe investigated variety of grasping hands (Fig.7) [14]. As you see in Fig.7, in most cases, a thumb is used for direct object manipulation and other fingers are used only for holding the object. Therefore, we construct a agent model which features the thumb movement. Fig.8 shows the model of an agent. *Hs* includes the 3D shape of a hand, *Hp* includes the positions of every fingertips, *Hg* define grasping orientations. Grasping orientations could be classified by each fingertips' distal, proximal, metacarpal angles from the back of one's hand(Fig.8). By updating *Hs* and *Hp* in real time processes, the latest grasp type of each agent's hand could be specified.

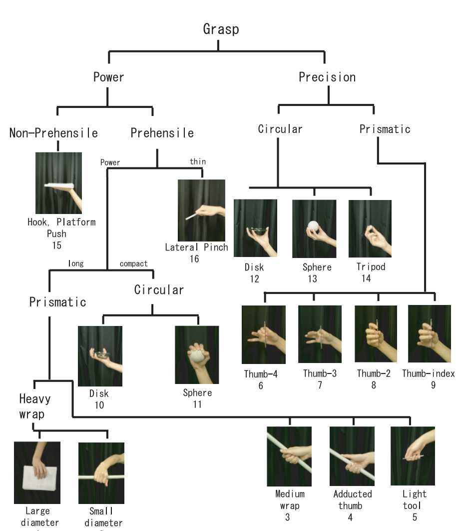

Fig. 7: Variety of Grasp

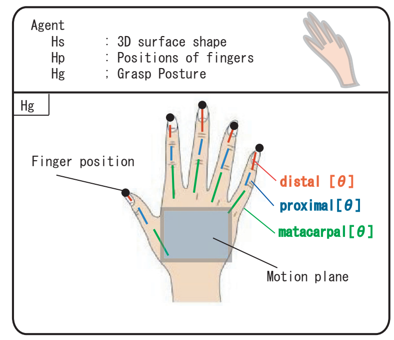

Fig. 8: Agent model

4.3 Interaction Among Three Models

4.3.1 Agent-Object Interaction

- Contact
  To detect contact points between an agent and an acting object, we use V-Clip [15] algorithm which is one of the minimum distance calculation methods. This algorithm calculates a pair of points which minimize distance between two convex polyhedrons, and as a result, contact points between convex polyhedrons could be detected. By using this algorithm, a number of contact points could also be calculated. Since, when the agent contacts with the acting object, the grasp posture (*Ug*) and the positions of fingers (*Up*) are measured by CyberGlove, the grasp type could be decided. If the present grasp type is included in the grasp types shown in Fig.7 , the agent is considered grasping the acting object.
- Action
  Direction of an acting object's motion is calculated from the movement of the agent. Fig.9 shows the position of contact points and the rotation direction of a lever, when the acting object consists with a lever. As shown in Fig.9, the arm of the acting object is rotated around the fulcrum to the direction of Ld and the movement of the arm is same with the absolute of orthographic projection of the motion vector of thumb position($P(t)$) on to Ld (direction of lever rotation). During this procedure, the other fingers do not move, and work on a stable plane to support the operation. Orthographic projection of $P(t)$ on to Ld (${}_{proj_{Ld}}P(t)$) becomes:

$$
{}_{proj_{Ld}}\mathbf{P(t)} = \frac{\mathbf{P(t)} \cdot \mathbf{Ld}}{\parallel \mathbf{Ld} \parallel^2}\mathbf{Ld} \tag{1}
$$

If positions of fingers (*Hp*) change but the grasping

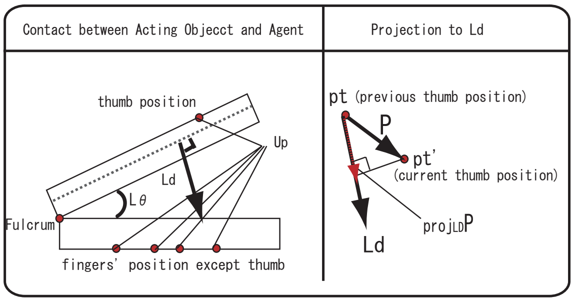

Fig. 9: Agent-Object Interaction 1

posture (*Hg*) doesn't, then the acting object is moved with the agent fingers.

- Force

The system architecture shown in Fig.5 can measure the agent finger's movement (*P(t)*), however cannot measure force magnitude of the agent to move the acting object. Therefore, in this paper, we define the method to calculate the force from the agent to the acting object (*Fag(t)*) based on the force balance between the acting object and the agent. When *Fagr(t)* is the force from the acting object to the agent,

Fag(t)<Fagr(t): an agent is departing from an acting object and reduces force magnitude to it

Fag(t)=Fagr(t): an agent keeps its position from an acting object and keeps balance with the force from the acting object

Fag(t)>Fagr(t): an agent makes toward to an acting object and increases force magnitude to it

In the case of *Fag(t)<Fagr(t)* and *Fag(t)>Fagr(t)*, the value of *Fag(t)* could not be calculated. Therefore, we assume that the faster the agent moves, the larger the value of *Fag(t)* becomes. In the case that a contact point at time t *p(t)* (*p(t)*) is moved to the point *p(t')* at time *t'*, the velocity (v) becomes:

$$v = \frac{\Delta x}{\Delta t} = \frac{p(t) - p(t')}{t - t'} \tag{2}$$

Fag(t) becomes:

$$F_{ag}(t) = Fagr(t) + fa \times v \tag{3}$$

Where, fa[N] is the per unit time constant of force magnitude from the agent. Next, Fig.10 shows dynamics between springs, *Fag(t)* and *Fagr(t)* in the case that the acting object are consist with a lever and a spring. When *Fagr'(t)* is the force feedback from spring at *Spos*, *Fagr'(t)* becomes:

$$F_{agr'}(t) = S_k \times d(t) \tag{4}$$

Where d is shrinking distance of the spring, *Sk* is the spring constant. For example, when a lever pattern *Lp* of the acting object model is Fig.2(e), force from the acting object to agent (*Fagr(t)*) becomes:

$$F_{agr}(t) = F_{agr'}(t)D_{ag}/D_s + F_{acr}(t)D_{ag}/D_{ac} \tag{5}$$

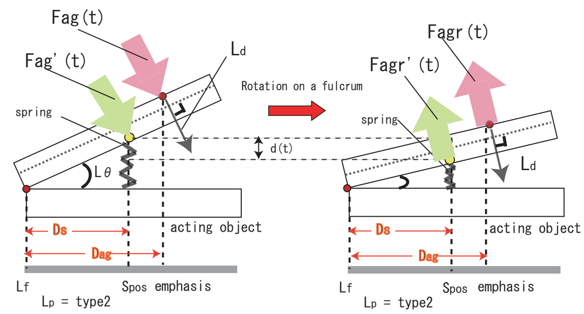

Fig. 10: Agent-Object Interaction 2

Where *Fagr'(t)* is the force from the acting object to the agent, *Dag* is the distance between emphasis and fulcrum, *Ds* is the distance between the spring and the fulcrum, *Facr(t)* is the force from the actee to the acting object, and *Dac* is the distance between locations of an application and a fulcrum, as in Fig.10. This force (*Fagr(t)*) is displayed to the agent's fingers as force feedback during the simulation. *Fag* is amplified or declined by the lever, and the force (*Fag'(t)*) is added to the spring. *Fag'(t)* is calculated by the following equation.

$$F_{ag'}(t) = F_{ag}(t)D_{ag}/D_s \tag{6}$$

4.3.2 Object-Actee Interaction

- Contact

The method to detect contact points between the acting object and the actee is same with the method in agent-object interaction. A number of contact points between the object and the actee is also calculated. Fig.11 shows examples of contact points.

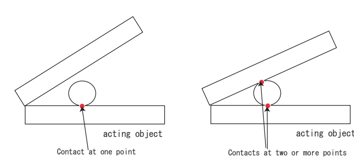

Fig. 11: Contact

- Action

Since we have an assumption that there are enough friction between the acting object and the actee, the actee does not slip from the acting object.

- Force

Fig.12 shows dynamics between acting the object and the actee. When the distance between a spring and a fulcrum (*Ds*), and the distance between locations of an application and a fulcrum (*Dac*) are measured, the force from the acting object to the actee (*Facr(t)*) becomes:

$$F_{ac}(t) = (F'_{ag}(t) - S_k d(t))D_{ac}/D_s \tag{7}$$

Where d is shrinking distance of the spring and *Sk* is the spring constant. Then the reaction force(*Facr(t)*) is calculated with *Fac(t)* and *Apv* predefined value in the actee model, which is the limitation of force magnitude for the actee. The reaction force from the acting object (*Fagr'(t)*) becomes:

$$F'_{agr}(t) = A_{pv} D_{ac} / D_s \qquad (8)$$

Where *Apv* is the predefined reaction force from the actee.

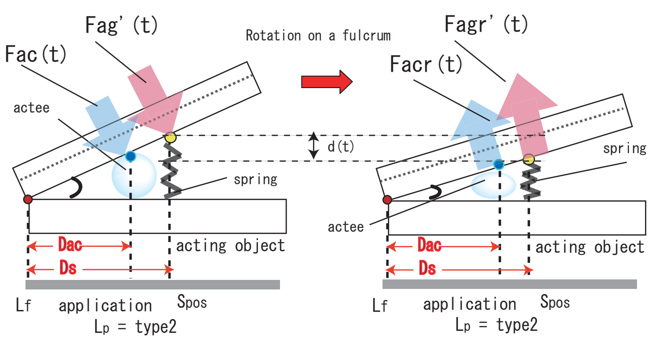

Fig. 12: Object-Actee Interaction

4.3.3 Agent-Actee Interaction

- Contact
  The method to detect contact points between agent and actee is same with the method in agent-object interaction.
- Action
  If there are more than three contact points between the agent and the actee, and if more than three direction of normal vector at the contact points are opposite to *Ad* (the directions of force from the actee to each finger: Fig.13), then the actee is grasped by the agent(Fig.13) and is moved with the agent.
- Force
  When an agent grasps an actee, the magnitude of force feedback is calculated from *Apv* and *Ad*, and the force feedback is displayed to the agent.

Fig. 13: Agent-Actee Interaction

4.4 Guide to Agent

Since our "analysis by simulation" approach is simulation-based method, the system can collect the history of agent's action easily to support agent's next foolproof and efficient action taking. During the function inferring process, four types of guides described below support the agent.

- Guide for stable orientation of Acting Object
  Most acting objects have a plane of symmetry. Under the gravity, it is easy to manipulate an acting object when its symmetry plane is parallel to the gravity vector. Using this feature, it is possible to guide the stable orientation to operate the acting object. When the simulation starts, all acting objects are set in the stable orientation for functioning. The guide of the plane of symmetry (*Osp*: Acting Object Model(Fig.4) is also shown as Fig.16(a).
- Guide for the Actee
  There are so many virtual object with various shape size, hardness...etc., prepared for the actee conditions. However, it is not easy to select suitable one as an actee. However, an appropriate shape and size of an actee can be estimated from the interaction space (*Lis*: Acting Object Model(Fig.4) information of the acting object. The interaction space is the space where an actee can be placed, and is estimated from a lever pattern(*Lp*: Acting Object Model(Fig.4), 3D surface shape of an acting object(*Os*: Acting Object Model(Fig.4) and an actee position. The guide for interaction space is shown as Fig.16(b).
- Guide for emphasis
  During the simulation, all potential planes to be emphasis are shown as Fig.16(a). This guide is available from contact plane (*Lcp*: Acting Object Model (Fig.4) information.
- Guide to amplify or decline Agent force
  Agents can amplify or decline their hand force on the location of a emphasis and an application. From lever pattern (*Lp*) information, it is possible to guide the location of an emphasis and an application to amplify or to decline the agent's hand force in real time (Fig.16(c)).

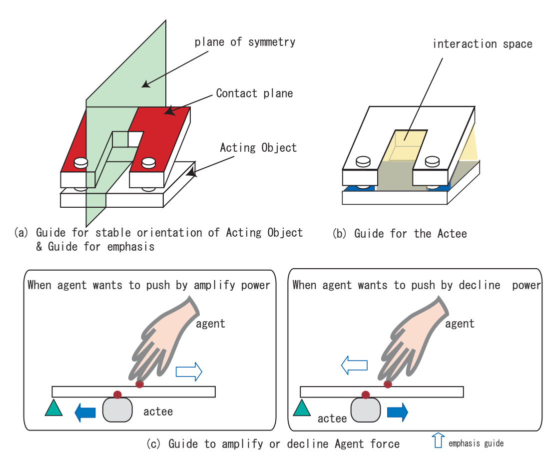

Fig. 14: GUIDEs

4.5 Algorithm to Inferring Function

Fig.15 shows the model of function. Interaction condition between the agent and the acting object, and between the acting object and the actee are described in each node of this model. Function can be inferred by tracing each condition. For instance, if a simulation meets all of the conditions described below, the function "Crash" is inferred.

(1) The agent grasps the acting object.

(2) The agent push the acting object.

(3) There are more than two contact points between the acting object and the actee.

(4) The acting object pushes with stronger force (*Fact(t)*) than predefined force value in the actee model (Apv), which is the limitation of the force magnitude for the actee.

There is variety of functions, such as "crash" and "hold" according to the force magnitude of an agent and the location of an application. If the acting object holds the actee and is moved by the agent, a function "carry" is inferred. Thus, to infer a function, the system compares the interaction condition of the agent, the acting object and the actee to the function model.

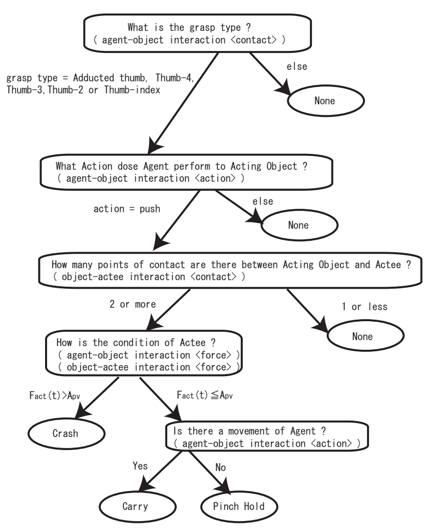

Fig. 15: Function Model

4.6 Simulation Flow of Function Inffering

Fig.17 shows the simulation flow of function inferring. First, the models of an acting object, an actee, and an agent are loaded in the system. Then the guide for orientation stable pose for functioning of Acting Object, the guide for the Actee, and the guide for emphasis are shown. Based on these guides, the agent chooses a simulation pattern. The next step is selection of a simulation pattern. step1-1)the agent chooses an actee arbitrary, step1-2) moves the actee to any place where the acting object can be operated, step1-3) grasps the acting object, step1-4) select emphasis position, and then the simulation starts. Step2-1)the lever pattern is decided according to the application and emphasis positions, step2-2) simulation of force amplify or decline is proceeded and reaction force is displayed to the agent, step2-3)the agent iterates "trail and errors", and step2-4)the system compares the present status of the agent, the acting object and the actee to the function model in real-time. Finally, if they consist with same conditions, it is possible to infer a function.

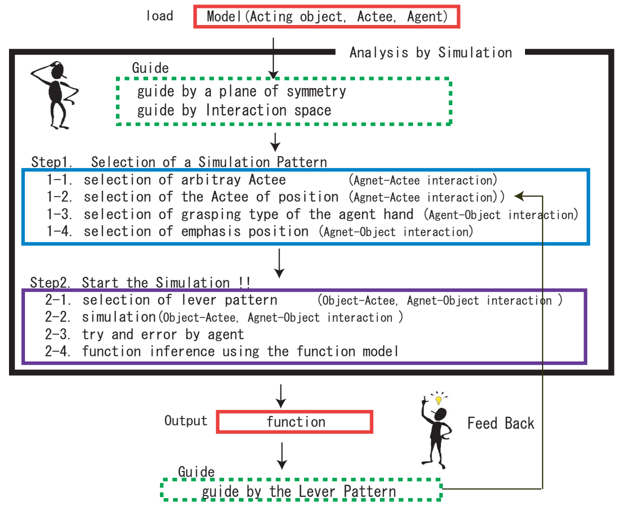

Fig. 16: Simulation Flow of Function Inffering

5. Experiment and Result

5.1 Experiment Purpose

We have developed a prototype of the proposed inferring function system and made experiments to estimate

- whether the system could infer the functions of articulated acting objects by the proposed method or not?
- whether our analysis by simulation approach including the guide and haptic feedback helps agents to understand acting object's function or not?

5.2 Experiment Method

During the experiment, subjects wear CyberGrobe and CyberGrasp on their hands and watch the display, where they can have haptic and visual feedback of virtual simulation. Models of an acting object, actees and an agent hand are loaded before the experiments start. We apply a pair of nut crackers and a pair of tongs as an articulated acting objects, which include both a lever and a spring. In the experiment, subjects were asked to find out as many functions as they can do. The subjects are instructed that they can touch, grasp and push the acting object and also can select and change an actee anytime from the actee guide area. After the experiment, subjects were asked whether the guides were helpful or not to find out acting object's functions, and whether haptic feedback helped to understand how to operate the acting object.

5.3 Result of Experiment

Function is inferred from the final state reached from a series of mutual interactions.

Fig.17 shows an example of function inferring process in a case of applying a tong and the actee with a hard surface. The first line of Fig.17 ((1)-(4))shows that a subject agent try to find how the tong moves and how much force is enough to close it. Second line ((5)-(8)) shows interaction between the agent and the actee. The subject agent moves the actee position and stars operations on the actee. Third line ((9)-(12)) shows that the object stops its movement when it hit the actee. Forth line ((13)-(16)) shows when the agent keeps the tong closed and moves it, the actee also moves with the tong and the simulation system inferred a function of this acting object "carry". Fifth line shows that when the subject agent increased his force to close the tong, the actee was broken. Then the simulation system inferred another function "crash."

All subjects can find these functions. During the experiments, it seemed that they learned from guide, which the actee to select, where to push, where to put an actee and how to increase or decrease their power to an acting object. We have got some comments from subjects. They told that they could easily understand how to operate the acting object from its shape, haptic feedback and guides. They also felt no difficulty to guess that the acting objects have "carry" and "crash" function, and to try these functions.

Fig. 17: Inferring Functions of "pinch", "carry", "crash"

6. Conclusion and Future Work

We have proposed "analysis by simulation" approach to inferring functions of articulated objects, through reality-based interactions for function tests in virtual environments.

We have developed three kinds of reality-based interactions among an agent(a human hand), an acting object and an actee , based on the physical object model acquired from observation of its behaviors in the real world with Haptic vision. Function inferring processes were generated and proceeded as state transition of mutual relations among the agent, the acting object and the actee, and their conditions. Function was then inferred from the final state reached through a series of mutual interactions. Mutipule Functions were inferred effectively from the acting object's shape, the haptic feedback and the guide.

In the future, we will apply more acting objects which have the other function primitives, such as a blade, a linkage and a screw. After that, we would like to assemble a database of agents' actions, during they infer functions of the acting objects, to investigate prototypical human action pattern to test and infer object functions.

References

1. Lederman, S.J., and Klatzky, R.L.: *Hand movements: A window into haptic object recognition.* Congnitive Psychology, vol.19, pp.342–368, 1987.
2. H.T. Tanaka, K.Kushihama: *Haptic Vision.* Proc. IEEE 16th Int. Conf. on Pattern Recognition, Vol.2, Session1.7 Robotics, Vol.II, pp.852–855, Aug. 2002.
3. H.T. Tanaka, N. Ueda, K. Kushihama and S. Hirai: *Vision-based Haptic Exploration.* Proc. IEEE Int. Conf. on Robotics and Automation, Sep.2003.
4. K.Nisimura, H.T. Tanaka: *Active Shape Inferring Based on the Symmetry in Stable Poses –Shape from Function Approach–.* IEEE Proc. on In. Conf. on Pattern Recognition(13th ICPR), pp.136–140, Aug.1996.
5. T.Tanigawa, H.Tanaka: *Estimating Mass based on Haptic Vision.* Proc. 2003 ACCV(to appear).
6. H.T.Tanaka, K. Yamasaki, M. Yamaoka: *Extracting Relational Constraints among Objects with Haptic Vision.* Proc. ICAT2002, pp.105-112, Dec. 2002.
7. N. Ueda, S. Hirai and Hiromi T. Tanaka: *Extracting Rheological Properties of Deformable Objects with Haptic Vision.* Proc. ICAT2003, Dec. 2003 (to appear).
8. M. Uejyo, H.T. Tanaka: *Active Modeling of an Articulated Object with Haptic Vision.* , submitting to IEEE 17th Int. Conf. on Pattern Recognition.
9. T.Yamamoto, H.Kato, S.Sato and S.Inokuchi: *3-D Object Recognition Using Functional Models.* IEICE, Vol.74-D-2[5], pp.601–609, 1991.
10. H.Hattori, K.Kise, T.Kitahashi and K.Fukunaga: *Recognition of Functions of Objects Based on Models of Dynamic Functions.* IPSJ, Vol.36[10], pp.2277–2285, 1995.
11. K. Green, D. Egger , L. Stark and K. Bowyer: *Generic Recognition of Articulated Objects through Reasoning about Potential Function.* CVIU, Vol.62[2], pp.177–193, 1995.
12. Z.Duric, J.A. Fayman and E. Rivin: *Function From Motion.* IEEE PAMI, Vol.18[6], pp.576–578, 1996.
13. D. Macaulay, N. Ardley: *THE NEW WAY THINGS WORK.* Dorling Kindersley Limited, London, Revised Edition 1998.
14. C. L.Mackenzie, T.IberallL: *THE GRASPING HAND.* NORTH-HOLLAND, 1994.
15. B. Mirtich: *V-Clip.* ACM Trans. on Grahics, Vol.17[2], pp.177-208, 1998.