가상현실을 위한 객체 연결 모델

주 우 석† 최 성 운† 박 경 희†† 이 희 승††

요 약

가상현실 기법이 일반적인 3차원 컴퓨터 그래픽스 기법과 구별되는 가장 큰 차이는 상대적인 현실감의 극대화에 달려있으며, 따라서 가상현실 기법에서는 입체음향이나 데이타장갑 등 인간의 지각및 인식 기능을 향상시키는 수단이 중요시되기도 한다. 그러나 이러한 하드웨어적 요소보다 중요한 것은 가상현실 기법으로 표현되는 내부 객체들의 행동에서 유래되는 현실감이다. 본 연구에서는 가상현실의 각 장면을 구성하는 본질적 요소인 객체들을 연결하는 상호작용을 능동적으로 모델링함으로써 다양한 현실감을 부여한다. 이를 위해 본 연구에서는 개개의 객체의 특성에 따른 지각반경및 자극반경을 기초로, 가상현실 공간을 하나의 물리적인 장의 형태로 설정한 필드 모델을 제시하고 구현한다. 현실감을 극대화하기 위한 가장 본질적인 요소로서의 객체간의 인과작용및 일반적인 상호작용은 이 필드 모델 안에서 상호간의 에너지 교환의 형태로 나타내어지며, 결과적으로 지각반경, 자극반경및 이들간에 적용되는 행동논리만으로 가상현실 내부의 모든 객체가 능동적으로 반응할 수 있는 능동객체 시스템을 이룰 수 있게 된다.

FTFM: An Object Linkage Model for Virtual Reality

Wou seok Jou†, Sungwoon Choi†, Kyung-hee Park†† and Heeseung Lee††

ABSTRACT

The most fundamental difference between general three dimensional computer graphics technology and virtual reality technology lies in the degree of realism as we feel, and thus the virtual reality method heavily relies on such tools as data gloves, 3D auditory system to enhance human perception and recognition. Although these tools are valid for such purpose, more essential ingredients toward the realism is the one derived from object behavior itself inside the virtual environment. This paper provides further realism by modeling active interactions between the objects inside scenes. For this purpose, this paper proposes and implements a field model where the virtual reality space is treated as a physical field defined on the characteristic radius of stimulus and sense corresponding to the individual object. In the field model, the rule of cause and effect as an essential feature of the realism can be interpreted simply as an energy exchange between objects and consequently, variation on the radius information together with behavioral logic alone can build the virtual environment where each object can react to other objects actively and controllably.

1. INTRODUCTION

Virtual environments, or virtual reality (VR, hereafter) technology is now finding numerous industrial and commercial applications in such areas as entertainment, education, simulation, medical, and scientific visualization [1].

Originated from the concept of teleoperation [2, 3], the technique is primarily concerned with conveying a level of personal presence within synthetic environments. "Sy-

· This paper was supported (in part) by NON DIRECTED RESEARCH FUND, Korea Research Foundation
† 정 회 원 : 명지대학교 컴퓨터공학과 조교수
†† 준 회 원 : 명지대학교 공과대학 컴퓨터공학과 석사과정
논문접수 : 1995년6월28일, 심사완료 : 1995년12월13일

nthetic" means that the scene can be artificial or virtual. For instance, the VR allows one to travel medieval cities, to fly into the space of the solar system, or to work through the double helix of DNA. "Personal presence" means that the technology works so realistically that one feels as if he were present on the environments.

It is this feeling of reality that the VR is aiming at. Human instinct and reaction driven by the real feeling is the most critical advantage of the technology over the general three dimensional computer graphics technology. In this respect, major concern in the current development of sensory devices such as head-mounted display, eye-tracking electronics, 3-D audio localizer, and tactile systems [4, 5] is also on how to enhance the feeling of reality in the VR. While the hardware devices can address the problem of the realistic sensing, softwares in the virtual domain require more sophisticated mechanism to approach the problem of reality. This is because of the fact that human perception of the virtual environments occurs only through the objects inside the VR. Human can also be treated as an object, and the sensing itself is merely an interaction between objects. If the object behavior lacks in the reality, then the entire virtual environments may fall into a fictitious scene. Therefore, an integrated and structured object synthesis model is a crucial requirement in the implementation of the VR.

The most prevalent way of synthesizing a VR is the concept of actors [6-8], and Ellis [9] devised a way of implementing the concept. In this concept, three components, called the content, geometry, and dynamics compose a VR scene. The contents correspond to object attributes while the dynamics relates to the physical interaction between objects. Here, only some special objects, called the actors, can interact with other object and therefore, the relative geometry of all the objects must be specified with respect to each actor. Furthermore, the field to be affected by the action must be predefined for an actor to move, so that only the objects inside the field are allowed to interact with the actor.

Nevertheless, the field of action, as we observe in ordinary life, must change as a function of time. That is, once an actor interacts with some other object, it may be accelerated enough to affect objects far from the current field of action, and this point must be properly incorporated. In general, environment surrounding an actor, and thus the reaction of the objects driven by the environment must be varying as a function of time and event. Moreover, the synthesis system must be structured such that all the objects are treated as actors, for the VR to be more generally and realistically controlled.

Studies to overcome this limitation of the actors can be found in the context of animation. Space-time constraints method [10-12] finds its way of realistic animation by suggesting that animators specify only what the objects should do, and how they should do it is determined by the objects on a time and event basis. For instance, given the starting and ending position of a particle, its motion trajectory is determined by some optimization techniques such as motion dynamics [13], or behavioral model [7, 15] by the object itself. In this method, the object interaction called stimulus-response is computed for each instance, and the next reaction is made to be dependent on the current environment. But, the method did not imply such issues as the VR environment or how to make objects act and react independently.

Consequently, the problem in composing a

VR may be stated as "How to build a VR scene where all the participating objects may act and react independently as an individual subject?", and "How the VR environment can trace the interaction linking the objects?" In this paper, we present an integrated VR synthesis system which, will treat all the objects in a VR as actors, and will also capture the time-varying object interaction as real as possible. An active object system for the first question, and a field model for the second question are proposed and implemented.

2.1. Field to Field Model

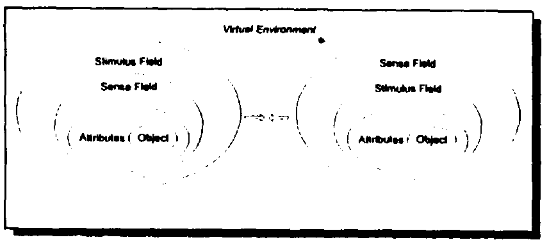
The source of feeling or sensing in a VR can be said to be the flow of energy between various types of objects including the human being. Visual perception is possible when the movement of an object is conveyed to another object through the light energy, and audio-sensory experience is possible when the shouting of an object is propagated to another object through the medium of sound wave. Also, tactile sensing corresponds to the case where the surface characteristic of an object is propagated to another object in the form of energy when the two objects get touched. The transmission medium such as the light energy, sound energy, or more general form of energy play a major role in a VR scene, as a communication mechanism between objects.

What are transferred between the objects as a result of the flow of energy can be said to be the object attributes such as the colors, orientation, velocity. For instance, the appearance and velocity of the object may be transferred for the visual sensing, while the attributes, called the message, are transferred through the medium of sound wave. The surface texture may be the transferred attributes for the tactile case. When different objects are dropped to the floor, they may produce different sounds depending on their attributes, even if the energy on the instant of contact may be of the same form and the same amount.

The most important aspect that we must consider when relating various objects through the medium of energy flow is the flow mechanism. Contrary to the conventional dynamics [13,15,16], where the energy transfer occurs only when the objects collide, the VR must accommodate more general relationship. The aggregate motion of flock of birds [14] must be able to avoid collision before they come into contact. A man walking toward a noisy radio switch may have decided to do so without contacting his ear to the radio speaker. In other word, the mechanism must allow the energy flow between any spacially displaced objects.

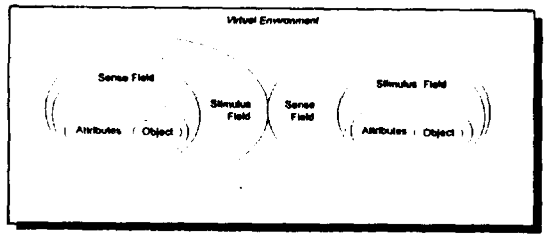
Field to Field Model (FTFM, hereafter) models the VR space after the concept of field in Physics. In Physics, two displaced objects can influence one another through the medium of a field. For instance, gravitational field means the energy space produced and radiated by the mass of the earth. Any object placed in the field is under the influence of the field energy, and falls to the earth. In our terms, the sound from an object forms a audio field toward outer space until it can no longer be heard. Any existing or newly created object within this range of space gets into the influence of the field, and the object will hear a sound of corresponding intensity.

Another aspect of the field concept that FTFM concerns is that of the sensing. A traveler in a VR must be allowed to find a new object as he proceeds into a new location. Although the new object was not in original scene, it must appear and be sensed as his perspective zooms in near the object. While the above sound example emphasizes the field in terms of the source of energy, there is another aspect in the field. It is the ability of the listener object. It may be sensitive enough to figure out a weak sound thinking that the sound is a loud noise, or it may not be hearing at all because it is deaf. This kind of duality in the object interaction is incorporated in the FTFM by defining two types of field. The first type is stimulus field representing the propagated energy intensity, and the second type is the sense field representing the sensing ability of an object.



(Fig. 1) FTFM before the Object Interaction

(Fig. 1) shows FTFM diagrammatically. Ea-ch object maintains a pair of window to the external world named stimulus field, and sense field. Both the field may be varying as a function of spatial displacement. As a result, it may be represented as a mathematical field density function or tabular representation. For instance, if the audibility drops logarithmically proportional to distance squared from the listener object, then the sense field may be represented by log (1/distance2), and the actual sensed value becomes stimulus value at that distance multiplied by this amount. Moreover, as the object moves, the field moves along the object, since the subject of the fields are object itself. Note that the fields are declared aside from the attributes, and this enables an object to have multiple stimulus/sense fields corresponding to different attributes.



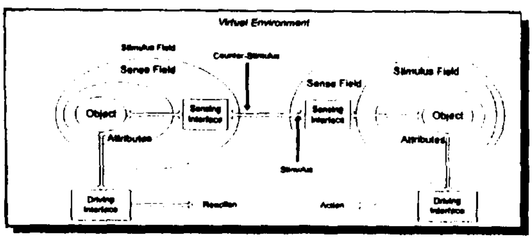
(Fig. 2) FTFM on the Instance of Object Interaction

(Fig. 2) shows the points of action when the two objects are drawn closer. The diagram shows an aspect of FTFM, that the objects react one another although they are not physically in contact. If the stimulus field of one object touches the sense field of another object, then the behavioral interaction is expected to be initiated immediately. However, depending on the behavioral logic of objects, they may not interact until they are further drawn together to trigger the initial interaction. That is, they can be made to respond only whence the intensity of stimulus/sense field is high enough to trigger the action.

Another important characteristic of FTFM is a concept of action-reaction. Right at the time of contact, energy from one object is transferred to another object and it may trigger various behavioral action inside influenced object. The FTFM enforces duality in the energy flow so that the energy must happen in pairs. In this way, FTFM can simulate more general phenomena, "You cannot touch without being touched." This rule [17], called the action-reaction rule, represents the Newton's third low of motion. Thus if one object's stimulus field influences the sense field of another object, the energy flows to another object and the same amount of energy flows back to the original object in the form of the stimulus produced by the influenced object.

This is shown in (Fig.3), where the reaction of the stimulating object is determined on the basis of the sense field of the object. If the counter-stimulus is not sensible by the current scope of the sense field, it may be neglected. The sensing and driving interfaces offer this functionality. The sensing interface provides functional sensing mechanism, and the driving interfaces offers behavioral logic between the sensed value and the the behavioral action/reaction.

As a result, by modeling and generalizing the physical communication mechanisms, FTFM can drastically increase the realism in a virtual environment. The remote physical communication including the sensing and stimulating is represented as the energy flow, and the contact communication is explained with the action reaction pair. Most importantly, the concept of field was introduced to facilitate creating a flexible virtual world.



(Fig. 3) Action-reaction Pair Induced by the Energy Flow

2.2. Active Object on Field

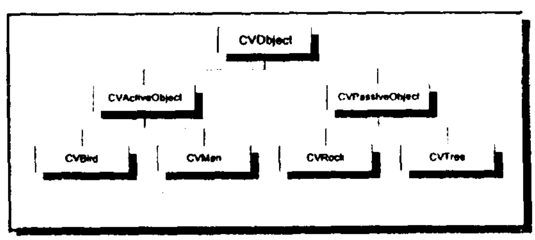
2.2.1. Object Hierarchy

Objects in our VR system are classified either as active or passive. This classification goes in parallel with the real world counterpart, animate and inanimate. If an object determines and initiate its own action by watching external environments, it is classified as active. On the other hand, if an object's action is initiated by some external mechanism, and it has no eye for looking and detecting outside world, it is classified as passive. Actually, an object with passive characteristics may be assigned active characteristics if required. For instance, an inanimate creature such as toaster may be classified as active, and then it may fly with intelligent thinking.

In order to exploit the inheritance between objects, logical object hierarchy in FTFM VR system is organized as in (Fig. 4).

The first top-level CVObject has general object attributes inherent to every object in a VR scene.

```
class CVObject
/
    protected:
        unsigned long ObjectID;
        CVector ObjectCoordinate;
        unsigned ObjectVolumeRadius;
        unsigned ObjectSound;
        CVector ObjectDirection;
        CVector ObjectForce;
    public:
        CVObject (unsigned long
        ObjectID);
          ~CVObject( );
```



(Fig. 4) Object Hierarchy in the FTFM VR System

ObjectCoordinate, and ObjectVolumeRadius provides information for visual stimulus whenever the object falls into the visual scope of a sensing object. Objectsound corresponds to the current sound volume of the object as a subject of a sound source, namely the audio stimulus. ObjectDirection and ObjectForce represent the stimulus energy of a moving object. Beside the audio visual energy, we also keep track of this type of energy flow to enable the physical interaction between the objects.

The second level, CVActiveObject has the following basic constructs.

```
class CVActiveObject: public CVObject
/
    private:
        CVector Behavior;
        virtual Look (CVObject* object);
        virtual Listen (CVObject* object);
        virtual Force (CVObject* object);
        virtual GetGlobalBehavior( );
    public:
        CVActiveObject( );
        ~CVActiveObject( );
        LookAtDistance( CVector position);
        ListenAtDistance( CVector position);
        ForceAtDistance( CVector position);
        virtual Interact( CVObject* object);
        virtual Behave( );
};
```

Along with the attributes, the functional interfaces for the sensing and driving constitute major components for this class. The functions Look, Listen, and Force represent the sensing functions corresponding to the visual, audio, and the physical stimului. If an object falls inside the stimulus fields of the other objects, these functions check the stimuli from the the other objects. Conversely, if an object is required to return a stimulus value, the functions LookAtDistance, Listen-AtDistance, ForceATDistance are executed. These functions offer the stimulus values of an individual object requested by the other object. For instance, ListenAtDistance returns the volume intensity, caused by the current object as a sound source. Similarly, Look-AtDistance may return the object countenance at a distance. The stimulus to a certain object from the other objects is calculated by calling these functions repeatedly for all the other objects.

The sensing function as a part of the sensing interface is called by the master scheduler function Interact. Subsequently, the sensing interface computes the sensed value according to the stimulus and the its own sensing logic. Based on the sensed value, the driving interface determines their behavior, records it to the variable called Behavior. Any such behavioral logic, as whether to respond to only the highest-valued stimulus, can be incorporated into this driving interface. Since this behavior is derived from the stimulus from a single external object, global behavior must be determined from the local behaviors by calling the function Behave.

The second level of class can further be divided into more refined subclasses. For instance, class CVActiveObject can be subdivided into class CVBird, and CVMan to emphasize the characteristic difference between birds and man. In this case, polymorphism can be used to facilitate different ways of sensing and driving mechanisms, depending on the characteristic of the classes. The class CVPassiveObject essentially shares the same stimulus functions as the active class, but it has no sensing functions because the class does not need to respond to a stimulus.

2.2.2. Active Objects and the VR manager

The definition of the term active object [9, 18,19] is regarded as somewhat abstract and diversified, because different authors interpret the concept "active" in different ways. But, we use the term because objects in FTFM VR system are treated as independent agent capable of active perception and interaction against the surrounding environments. In a pure active object environment, each object in a VR must be able to sense and respond to the stimuli from other object simultaneously. However, this is not possible in single-processor machines, and a different mechanism must be devised to accommodate the synchronization and consistency in the interaction among multiple objects.

The VR manager plays the role of master scheduler of all the stimulus/sense and the action/reaction part of FTFM VR system. Basically, the manager controls only the sequence of the functions, and the functions themselves are done through the message passing mechanism inside the objects. In a pseudo-code notation, the major tasks that the VR manger must perform is,

```
Phase I:
For Every Active Object
        For All the Other Active Objects
                Initiate Interaction Between the Two;
                Record the Local Behavior;
        For All the Other Passive Objects
                Initiate Interaction Between the Two;
                Record the Local Behavior;

Phase II:
For Every Active Object
        Get Global Behavior;
        Do Behave;
```

The VR manager treats the active and passive objects separately. Although a passive object may provide its own stimulus to an active object, it has no ability to initiate its own action. By separating the passive objects and eliminating the sensing process for the passive object, we can speed up the VR manager's performance.

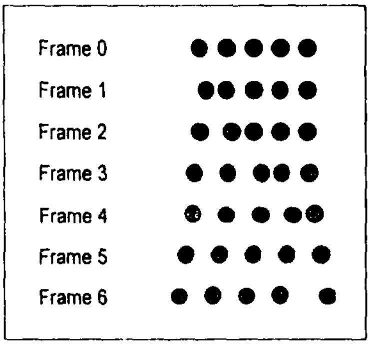
The mechanism of the above pseudo-code works on a frame by frame basis, or on a time basis. In every predetermined time interval, each object interacts with all the other objects, and records the resulting behavior in the first phase. The actual behavior of phase II occurs only when all the behavior of individual object is completely recorded. In a plain sequential machine, if object 1 is expected to interact with object 2, and at the same time, object 2 is expected to interact with object 3, the result of the interaction between the object 1 and object 2 may affect the interaction between object 2 and object 3. Through using the phase by phase synchronization mechanism, this problem is avoided in our FTFM VR system.

3. EXPERIMENTAL RESULTS

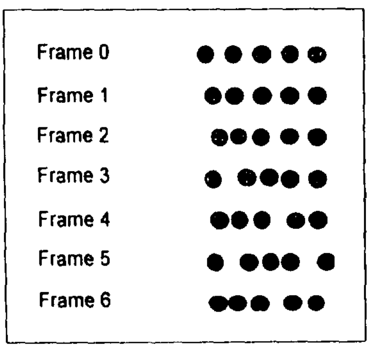
The prototype FTFM VR system is tested extensively for various situations. Five cases are presented here to emphasize how the system handles the different sensing environments and different behavioral logics.

(Fig. 5) shows the result of collision avoidance. In this case, movement of an object acts as a stimulus to another object's visual sensing organism. The sensing object's sense field is simplified as a scope of vision, and the sensing object responds as soon as the moving object gets inside this scope. With the object's volume radius of 5 units, the radius of the visual sensing is set to 7 units, so that the objects can respond even without actual collision. Frame 0 shows the initial position of objects space equidistantly. From then on, the first object moves right. In Frame 1, the second object detects the movement by its visual sensing. As a result, the second object tries to avoid the collision by moving right, which is the same direction as the movement of the first one. At the same contact time, the existence of object 2 is sensed to object 1, it changes its direction toward left thus explaining the reaction.

(Fig. 6) shows the same situation as (Fig. 5). This time, however, the motion is goal-directed in the sense that once an object moves back to its original position as a result of action-reaction, it remembers its original direction and proceed with that direction again. That is, the first object in Frame 3 remembers its goal, and moves right as shown in Frame 4. As can be verified, the collective movements of the one-dimensional objects oscillates, given the fixed outside boundaries.



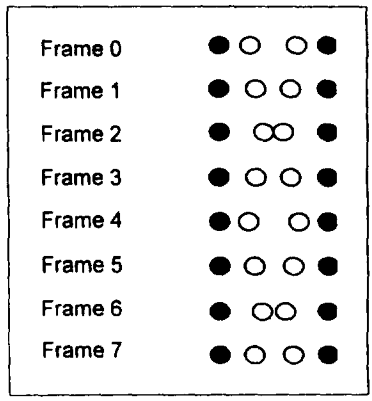
(Fig. 5) 1-D Collision Avoidance



(Fig. 6) 1-D Collision Avoidance with Goal

One way to speed up the process of examining the sense field is a bounding volume test [20, 21]. If the sense field of an object takes the form of radiation centered at that object, the sense field looks like a sphere. In this method, whether an object is inside the spherical volume or not, can easily be tested by inequality relation, and the processing speed can further be enhanced by using this method.

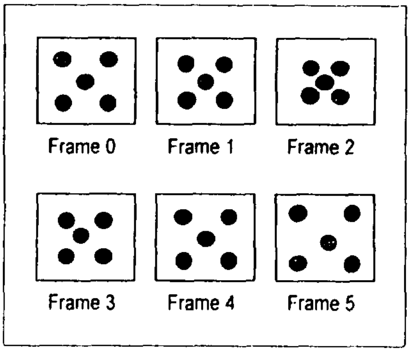
(Fig. 7) shows the interaction between the active and passive objects. Here, the black colored objects represents the passive ones. Starting from Frame 0, two inside objects, the active objects, are drawn together. As soon as their sensing boundaries are encroached upon in Frame 2, they start moving apart as in Frame 3. Although they reach the sensing boundary of the passive objects in Frame 4, the passive objects are not allowed to show any behavior in the FTFM VR system.



(Fig. 7) Active vs. Passive Objects

In (Fig. 8), two-dimensional collision avoidance is tested with the object volume radius of 5 units, and the visual sensing radius of 15 units. In the driving function, the objects are programmed such that they don't remember the goal, and they respond to the last stimulus inflicted. At the start, the four objects in the corner move toward the center object as shown in Frame 0. In Frame 3, the center object moves toward top left corner since the top left object was set to be the last object inflicting stimulus. As a result of the reaction due to this action, the center object recovers original position and moves toward bottom right as shown in Frame 4 and 5. Note that the FTFM can render the center object remain at center position if we used such behavioral logic as to average the cumulative stimulus in the generation of

Frame 3. This is possible because our system has the master scheduler that could sum out all the stimulus globally in a batchwise fashion.



(Fig. 8) 2-D Collision Avoidance

The FTFM VR system is also tested for the simulation of audio effects. In order to test the the sensing and stimulus mechanism of the field model, we made variation in the intensity of audio fields.

In the experiment, the audio stimulus field was assumed to drop proportional to the distance. For instance, an object displaced 50 units away from the source of sound emanating 100 decibel of sound will get the stimulus value of 100/50 decibel. Moreover, the sensing object is assumed to have a threshold as a variation of the sense field function. If the received stimulus value is more than the threshold, the sensing object moves in response, otherwise it remains at the same position. Moreover, we defined two types of behavioral logic. The first type of object, named pro, likes the sound and moves toward it, while the second type, named con, dislikes the sound and moves away from the source of sound. Given the threshold sensing value of 2 decibel for each of the object, the following table shows the result.

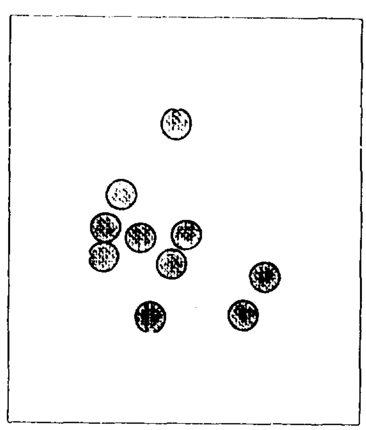
Initially, both objects were apart from each other 30 units, as was shown in Frame 0.

(Table 1) Object Interaction with Audio Visual Capability

Frame	Object 1				Object 2			
#	Type	Vol.	Position	Sensed	Type	Vol.	Position	Sensed
0	pro	40	−15	3.3	pro	100	15	1.3
1	pro	40	−11	3.8	pro	100	15	1.5
2	pro	40	− 7	4.5	pro	100	15	1.8
3	pro	40	− 3	5.6	pro	100	15	2.2
4	pro	40	1	10	pro	100	11	4
5	pro	40	− 3	5.6	pro	100	15	2.2
6	pro	40	1	10	pro	100	11	4

Since object 1 senses the volume intensity of 3.3 from the object 2 at that position, and the threshold was 2, it moves toward object 2. Even if object 2 also likes the sound, the sound perceived was so week that it remains at the same position. At Frame 4, both objects become closer, and the visual sensing made the object move apart yielding Frame 5. Nevertheless, in the position, they feel each other's sound and they are attracted toward each other as in Frame 6, and the situation is repeated. This accounts for the audio visual interaction of the FTFM VR system.

As a final experiment, multiple objects are made to be located at randomly generated positions, and the interaction was tested. Here, the object volume radius is set to 10 units, while the sensing radius is set to 20 units. Besides, the behavioral logic of the object is made such that they move opposite to

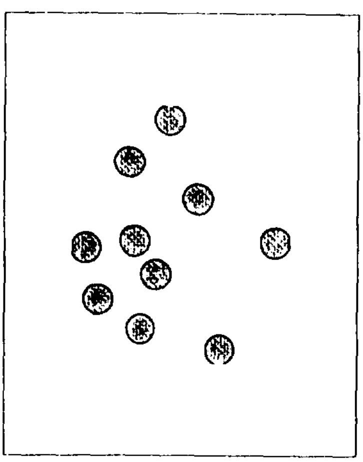


(Fig. 9) Frame 0 of Multi-object Action

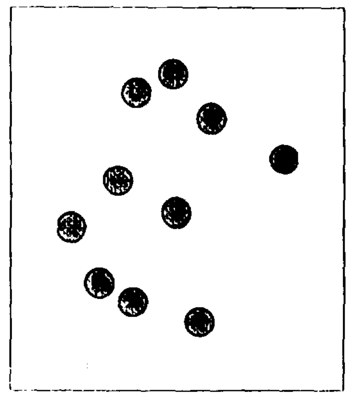
the nearest object only. (Fig.9) through (Fig. 12) show the gradual disposition of the object behavior. In general, they can be said to be dispersing from each other. FTFM VR system reveals the exact inter-object behaviors as a function of time

4. SUMMARY AND CONCLUSION

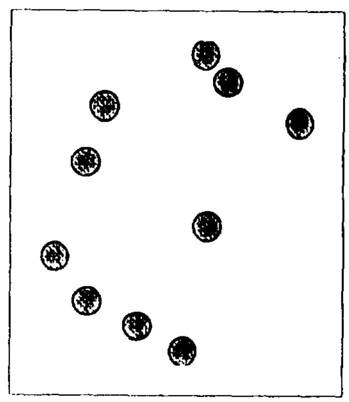
In this paper, we presented a new model for the linking of objects and the synthesis of a virtual environment. The model is derived from the concept of "field" used in Physics. Adopting this concept, we could easily model the stimulus-sense, and the action-reaction relation which is quite prevalent in the real world situation. This led us to enhance the feeling of the reality present in the physical world. Moreover, the active object system and the master scheduler, as a virtual environment manager was explained in detail as the major components of the model. The active object system is used to mimic the real world actors, while the master scheduler is used to overcome the limitation of the single processor machine. Subsequently, the experimental results with various virtual situation is presented verifying the proposed field model. The field of virtual reality technology can be said to be immature and just immersing, and the known methodology is not quite complete yet. Further studies on the software modeling and implementing of the virtual reality must be accompanied to support the enormous industry requirements and hardware developments.



(Fig. 10) Frame 6 of Multi-object Action



(Fig. 11) Frame 12 of Multi-object Action



(Fig. 12) Frame 18 of Multi-object Action

REFERENCES

[1] 윤경현, 최정단, "Virtual Reality의 개발환경", 정보과학회지, vol.11, no.6, pp.14-24, 1993

[2] 박찬모, "가상현실 개요", 정보과학회지, vol.11, no.6, pp. 5-13, 1993

[3] Kalawsky, R.S., *The Science of Virtual Reality,* Addison-wesly, Seoul, 1993
[4] Applewhite, H., "Position-tracking in virtual reality," Proceedings on 2nd Annual Virtual Reality Conference and Exhibition, pp. 1043-1048, 1991
[5] Fisher, S., McGreevy, M., Humphries, J., and Robinett, W, "Virtual Environment Display System," ACM Workshop on Interactive 3D Graphics, pp. 77-87, 1986
[6] Agha, G., *Actors: A Model of Concurrent Computation in Distributed Systems,* MIT Press, Cambridge, Messachussettes, 1986 .
[7] Hewitt, C., "Viewing Control Structures as Patterns of Passing Messages," Artificial Intelligence, vol.8, pp. 323-364, 1977
[8] Ellis, C.A., Gibbs, S.J., "Active Objects: Realities and Possibilities," pp. 561-571, Object-oriented concepts, databases, and applications, edited by Kim, W., Lochovsky, F.H., ACM Press, Addison-wesley, 1989
[9] Ellis, S.R., Representation in Pictorial Virtual Environments, Taylor and Francis, London, 1991
[10] Witkin, A., and Kass, M., "Spacetime Constraints," ACM Computer Graphics, vol.22, no.4, pp. 159-168, 1988
[11] Ngo, J.T., Marks, J., "Spacetime Constraints Revisited," ACM Computer Gr-aphics, vol.27, no.4, pp. 343-350, 1993
[12] Cohen, M.F., "Interactive Spacetime Control for Animation," ACM Computer Graphics, vol.26, no.4, pp. 309-315, 1988
[13] Brotman, L.S., and Netravali, A.N., "Motion Interpolation by Optimal Control," ACM Computer Graphics, vol.22, no.4, pp. 309-315, 1988
[14] Reynolds, C.W., "Flocks, Herds, and Schools: A Distributed Behavioral Model," ACM Computer Graphics, vol. 21, no.4, pp. 25-34, 1987.
[15] Baraff, D., "Coping with Friction for Non-penetrating Rigid Body Simulation," ACM Computer Graphics, vol.25, no.4, pp. 31-40, 1991
[16] Hahn, J.K., "Realistic Animation of Rigid Bodies," ACM Computer Graphics, vol.22, no.4, pp. 299-308, 1988
[17] Halliday, D., Resnick, R., Fundamentals of Physics, John Wiley & Sons, New York, 1988
[18] Choi, S., and Minoura, T., Active Object Systems, Ph.D Dissertation, Oregon State University, 1992
[19] Liskov, B., and Scheifler, B.,"Guardians and Actions: Linguistic Support for Robust, Distributed Systems," ACM Transactions on Programming Languages and Systems, vol.5, no.4, pp.381-404, 1982
[20] Glassner, A.S., An Introduction to Ray Tracing, Academic Press, San Diego, 1989
[21] Rogers, D.F., Procedural Elements for Computer Graphics, McGraw-Hill, New York, 1985

주 우 석
1976년~83년 서울대학교 공과대학 전자공학과(학사)
1983년~85년 한국 IBM, 데이콤 정보통신연구소
1985년~87년 University of Florida, 컴퓨터공학(석사)
1987년~91년 University of Florida, 컴퓨터공학(박사)
1992년~현재 명지대학교 컴퓨터공학과 조교수
관심분야 : 컴퓨터그래픽스, 알고리즘, 데이타베이스

최 성 운

1985년 한국외국어대 무역과(학사)
1988년 Oregon State Univ., 컴퓨터과학 석사
1992년 Oregon Stae Univ., 컴퓨터과학 박사
1993년~현재 명지대학교 컴퓨터공학과 조교수
관심분야 : Object-Oriented S/E, 멀티미디어, HCI

박 경 희

1990년~94년 명지대학교 공과대학 전자계산학과(학사)
1994년~현재 명지대학교 공과대학 컴퓨터공학과(석사과정)
관심분야 : 컴퓨터그래픽스, 알고리즘, 컴퓨터구조

이 희 승

1990년~94년 명지대학교 공과대학 전자계산학과(학사)
1994년~현재 명지대학교 공과대학 컴퓨터공학과(석사과정)
관심분야 : 컴퓨터그래픽스, 알고리즘, 컴퓨터통신