RoboCup-3D: The Construction of Intelligent Navigation System

Atsushi Shinjoh¹² kaminari@iamas.ac.jp

 International Academy of Media Arts and Sciences, 3-95, Ryoke-cho, Ogaki-city, Gifu 503, Japan.
² Softopia Japan, 4-1-7, Kagano, Ogaki-city, Gifu 503, Japan.

Abstract. RoboCup-3D is the attempt to build a virtual three-dimensional viewer for the RoboCup software league. Intelligent Navigation System(INS) is a basic concept of this RoboCup-3D, and it aims at creating a system which assists and navigates human cognition and activity in virtual three-dimensional space. Based on INS, we built SPACE, which is the first three-dimensional viewer for the RoboCup software league. In this paper, we are proposing the basic concept of this research and will show the possibility and the aims of future research of INS based on the analysis of SPACE.

1 Introduction

Recently, research and development about the virtual three-dimensional world is proceeding on the computer because of the improvement of the computer's ability. The progress of the hardware has made the real-time rendering process of three-dimensional objects on a personal computer. In addition to these factors, programming languages which are suitable for building three-dimensional worlds such as Virtual Reality Modeling Language(VRML)[11], Java3D[10] and OpenGL[8] are being made. Based on these trends, some projects on the practical use of virtual three-dimensional space are being done such as the cyber mall³[11], multi user communication systems[9], and cyber showroom⁴. In spite of these activities, most of these projects have not succeeded today. Because of these present conditions, there are many people who doubt the possibility of virtual three-dimensional space[2].

There are two problems in these projects. One is the problem of the lack of reality. A virtual space which has realty and rich content does not exist except in entertainment games and movies. Compared with the above projects, these contents are not an imitation of the real world, but the original of each virtual world. Furthermore, to create this content needs a lot of time, money and the excellent skills for creating virtual three-dimensional objects today. Part of this

³ The cyber mall is the 3D shopping mall on the net.

⁴ http://vrml.sgi.com/

problem will never solved at all, because the construction depends on human's skill and talent, and doesn't depend on its effort. However, the rest part of this problem will be solved by the development of hardware and software which will assist in the construction of three-dimensional objects. The other is the problem of human's cognition in the virtual three-dimensional world. In the real world, human-beings usually can afford information from the objects based on its experience.[3] Human-beings also will be able to use its body(such as hands, legs, eyes, ears, and other parts) and its several senses for its perceptions. Based on these perceptions, human-beings form its experiences. In the virtual threedimensional world, humans can only use their hands for operating the keyboard and some pointing devices, and cannot use its senses. Furthermore, the virtual three-dimensional world only exists in two-dimensional displays, and humanbeings never use its experiences. For these reasons, most of the virtual threedimensional worlds are not suitable for human's cognitions. Some entertainment games have coped with this problem by creating and using original devices. Some research in the study fields of virtual reality have also tried to cope with this problem in the same ways. However, cognition of the virtual three-dimensional world is still difficult, because these approaches only deal with the method of navigation, and do not deal with the whole navigation system. Because the computer display separates the virtual three-dimensional world and the real world, human-beings never use the real world's cognition. This means that the method which assists human cognition is needed for coping with this problem.

For solving and coping with this problem, we propose the basic concept called **Intelligent Navigation System(INS)** which is the basic concept and aims at creating a system which assists and navigates human cognition and activity in the virtual three-dimensional space. For developing the research of **INS**, we will describe the research of RoboCup-3D, which aims at making a three-dimensional viewer for the RoboCup software league, because **INS** is the main concept of this research.

2 Intelligent Navigation System

Intelligent Navigation System(INS) is the basic concept of RoboCup-3D which includes some research areas such as artificial intelligence, artificial life, and virtual reality. This kind of research such as adaptive autonomous agents has been done[1][7]. However, because this research was separated from the real world's objects, there are no objects which we could compare with this kind of agent. Because we developed this research based on the research of RoboCup-3D, INS can be compared with an actual soccer game. The soccer game is one of the most popular sports in the world, and we can analyze and evaluate this system based on the analysis of several different relays of soccer games⁵.

⁵ In fact, the method of the relay of soccer game is quit different in each country.

2.1 RoboCup-3D:three dimensional viewer - SPACE

Two leagues of real and software exist in RoboCup[4][5]. We have created a first three-dimensional viewer for **MARS** based on the analysis of the relay of a soccer game in the real world.[6]⁶. Because **MARS** has a two-dimensional viewer as a standard viewer, this three-dimensional viewer is an optional viewer for **MARS** now. We named this three-dimensional viewer **SPACE**.

Specifications. We used OpenInventor as a development tool of OpenGL. OpenGL is the graphic library for creating the three-dimensional virtual environment developed by SGI. Though Windows NT and SGI(IRIX) are the platforms of OpenInventor, there are some differences in these two versions. For this reason, **SPACE** runs only on SGI machines.

The following are lists of necessary environments for SPACE.

For development.

- IRIS Development Option 5.3
- C++ Translator 4.0
- ProDev/Workshop 2.5.1
- OpenInventor2.0.1

For execute only.

- OpenInventor execution only environment⁷.

Fig. 1 shows the screen shot of **SPACE**. The left window is a bird's-eye view. This left window usually show the whole field's conditions, and not moved automatically. However, the operator can operate this movement by hand. The right window is a camera view which moves automatically, and the operator cannot operate this movement.

Points of the creation. Following is the list of points which became problems.

1. The ratio of the soccer field to the soccer player:

In the real world, the ratio of the soccer field and the soccer player is very big. When we use the real world's ratio in **SPACE**, we will never be able to look at the state of the RoboCup game in detail. A two-dimensional viewer, which is attached to MARS, uses the original ratio, which is different from the real world. Though this change directly influences the total impression of the viewer, we decided the ration based on the total impression of the scene.

⁶ MARS is the server for RoboCup software league created by ETL.

⁷ This is a free software, and you can download this from SGI home page.



Fig. 1. Three-dimensional viewer - SPACE

2. The reality of the object:

Texture mapping is the typical method for describing the object in virtual three-dimensional world. However, this method needs a lot of data processing power. When we use texture mapping for describing each object, the total speed of **SPACE** decreased excessively. As a result, **SPACE** cannot describe the whole conditions of the soccer field at each time. Based on these results, we adopted the coloring of the polygon instead of texture mapping. Fig. 2 shows the soccer field. The right one is described by the texture mapping, and the left one is described by the coloring of the polygon. Compared with texture mapping, the coloring of the polygon doesn't need much power. However, this expression does not have a good appearance.

The motion of the player is also an important thing. We created two motions for 'Kick' and 'Run', at first, and we created the detailed motions, and applied these data to **SPACE**. Unfortunately, when **SPACE** used these motions, the total speed of **SPACE** went down dramatically. As a result, the motion of the soccer player in the present **SPACE** is very simple⁸

3. Camera:

There are five cameras on the soccer field. Two cameras are arranged at the back of each soccer goal. The other camera is arranged at the top of the field. The screen shot which is displayed in the right window of **SPACE** is only provided by one of these cameras. The position of each camera is fixed. As mentioned above, in general, the operation of the camera in virtual three-dimensional space is very difficult, because the operator must catch the soccer ball based on the information which shows on the two-dimensional screen.

⁸ At first, we created 30 flames per each motion such as 'Kick' and 'Run'. Finally, however, we used only 4 flames per each motion.



Fig. 2. The soccer field.

When the operator loses the soccer ball, the operator must look for the ball based on this information. For assisting this operation, we created two functions in **SPACE**. One is the function of the camera. Each camera always follows the ball automatically, and the camera which provides the scene to the right window is switched automatically. Thus, basically, **SPACE** always displayed the scene of the field in detail automatically. This is a fundamental method of **INS**. The other is the use of a bird's-eye view. The operator can get the whole condition of the field from this left window(See Fig.1). Basically this window provides the some information as the two-dimensional viewer. However, the operator can operate this camera for its purpose, and get the information which it wants. Because the position of this camera which provides the scene of this bird's-eye window is always fixed, the operator can only use some functions such as 'Zooming' and 'Targeting'. Fig. 3 is the sample of this function.

3 Discussion

3.1 Comparison the two-dimensional viewer and Space

We must compare the two-dimensional viewer and **SPACE** for answering the following question: 'What kind of advantage does **SPACE** have in RoboCup?'. MARS has a two-dimensional viewer by default. Because the present RoboCup software league only deals with the two-dimensional field, the movement of the soccer ball is two-dimensional. As a result, all data which exists on the field can be shown in this two-dimensional viewer. In fact, the movement of the soccer ball in **SPACE** is two dimensional.

A two-dimensional viewer has the following advantage in comparison with a three-dimensional viewer.

- The necessary calculation is small.

Compared with the three-dimensional world, the amount of calculation in



Fig. 3. Space(when the operator operate the left window by hand.)

the two-dimensioanl world is obviously small. Since the calculation ability of today's computer is limited, this difference is important for describing the field's information.

- The operator can see the whole of the field's information easily. Because the three-dimensioanl viewer cannot show the whole of the field's information in detail at the same time, the operator cannot catch the movements of soccer players easily. When the operator wants to analyze movements of soccer players, the two-dimensional viewer is more suitable than the three-dimensional viewer.

For the operator, the two-dimensional viewer is most suitable because the twodimensional viewer is only a data showing system, and does not have any extra functions. Compared with this, the three-dimensional viewer must fascinate the audience by its own created scene. This is exactly the main purpose of **INS**. **INS** and the three-dimensional viewer is not a system for the operator, but a system for the audience.

Furthermore, the above two disadvantages do not show the limitation of the virtual three-dimensional viewer. The first one will be solved by the development of hardware and software, such as OpenGL optimizer, or next generation's VRML. The last one will be solved by the research of **INS**. When these problems are solved, the three-dimensional viewer will be a most suitable system for the operator, and will be a standard viewer for **MARS**.

3.2 Points of constructing INS

Based on points described in the second chapter, we put together the problems as follows.

Implementation:

This point is based on No.1 and No.2 in the second chapter. I will put the problems in order as follows.



Fig. 4. The two-dimensional viewer

(a):The law.

This is the problem about relations between the real world and the virtual three-dimensional world. This problem has a relationship with No.1 in the second chapter especially. While the real world has restrictions according to the laws of nature, a conclusive law does not exist at all in the virtual three-dimensional world. When the world's constructor does not apply the laws of nature in its constructed virtual world, the operator cannot apply its common sense to this virtual world. This kind of world's construction may lead to confusion for the operator. Therefore, the virtual world needs some kind of the law of nature. In fact, some programming language which is suitable for constructing the virtual world supported this kind of function. For example, VRML2.0 supported some laws, such as collision detect and three-dimensional sound system. Though these functions are useful so that the operator grasps the virtual environment, these functions need enormous calculation quantity.

Another problem is pointed out about the law in No.1 in the second chapter. **SPACE** disfigured the ratio of the soccer field and the player. What kind of disfigurement is right? If the operator usually loses the soccer ball in **SPACE**, we had better raise the ratio of the soccer ball in **SPACE**.

(b):The description of the object.

What kind of thing is a description which has reality? In the second chapter, I pointed out that **SPACE** does not adopt texture mapping, and adopts coloring of the polygon. Fig. 2 shows the example of this method. Fig. 5 shows that a few polygons are used to build the soccer player. Data size directly influenced the total performance of **SPACE**. While the use of texture mapping or detailed description by many polygons adds reality to the object, these methods make the calculation quantity increase.

(c) The motion of the object.

Virtual Fighter⁹ shows that movements or actions of objects add reality to the objects themselves. What kind of movement gives the object reality? In SPACE, the motion of each player was built by hand. This method needs a very long time, and the result of the motion depends on the builder's skill. Compared with this method, the method of motion capture which was used by Virtual Fighter makes building of a smooth action or movement possible. However, it is generally difficult to get data for motion capture, because exclusive hardware and high costs are necessary for this method. Furthermore, detailed descriptions of motion make the calculation quantity increase.

Most of these discussion issues such as (b) and (c) seems to not be related to **INS**, but related to the development of hardware and software technology which we already discussed in the first chapter. In this version's **SPACE**, I set several conditions by hand. However, in the RoboCup software league, the various environments for running the **SPACE** are presumed, and it is actually impossible to use hand-tuning in every place.

This discussion applies to (a) as well. The most suitable ratio of the soccer field and the player depends on the audience and the total environment of the screen. Does the operator changes this ratio in every place?

Though **SPACE** does not support these autonomous settings yet, this kind of setting obviously includes **INS**, and this function must be provided by the autonomous agent system.

View:

This issue contains four different factors such as 'viewpoint', 'view position', 'the number of cameras' and 'the switch of the camera'. Each issue is related to the camera, because the scene is only provided by the camera in **SPACE**: **Viewpoint**

The viewpoint is the target of the scene held by the camera. In SPACE, the viewpoint of each camera is the soccer ball, and each camera follows the soccer ball automatically. In the actual relay of the soccer game, the camera operates by hand, and the viewpoint doesn't always hold the soccer ball. This means that the scene provided by Space supplies less information than the scene which is provided by the actual relay of the soccer game. Therefore, the operator(and the audience) of SPACE cannot catch the whole scene on the field. However, this viewpoint, which is provided by SPACE, gives priority to the operator who gets the detailed information arround the ball. This is needed to re-adjust this setting based on the analysis of the relay of soccer game in the real world.

View position

There are five cameras in the circumference of the soccer field. The 'view po-

⁹ Virtual Fighter is the entertainment arcade game created by SEGA Enterprizes,Ltd.



Fig. 5. The soccer player

sition' is the position of each camera. The 'view position' in today's **SPACE** is set based on the analysis of the actual soccer game, Because this 'view position' is fixed, Each camera does not change its 'view position' at all, and the operator also does not do this operation. Because the view position is different in each country, the ideal 'view position' is also different in each country. Thus, we cannot decide the ideal 'view position' based on the relay of the soccer game in the real world.

The number of cameras

Only five cameras exist in **SPACE**. At first, we set seven cameras in **Space**. However, this number required the switching of the camera constantly. As a result, the operator could not catch the position of the ball. Next, we set three cameras in **SPACE**. However, a small number of cameras makes the scene boring. As a result, we set five cameras. However this number depends on other three factors, and cannot be decided independently.

The switch of the camera

In this version's **SPACE**, the operator does not have to switch the camera. The camera which shows the scene on the display is switched automatically, based on the position of the ball. This kind of automatic system is one of the most fundamental systems for **INS**. Thus the important thing about this factor is the value or method for the switching camera. For example, in this version's **SPACE**, switching the camera is based on the position of the soccer ball on the field.

As mentioned above, we set each value by hand. However, each factor is a very complex task, and these factors are influenced by each other. In fact, though this navigation system is a very fundamental one, it is difficult to set each value by



Fig. 6. The soccer field

hand. Furthermore, it needs a total navigation system based on the above four factors to assist human cognition. In this version of **SPACE**, we set this system as follows:

The camera always keep track of the motion of the soccer ball from a fixed position. This means that the ball always moves from the down side to the up side on the screen. The automatic switching function for the camera also supports this system.

Though this is not always a good system for providing the scene, this method was created based on some trial and analysis about the relay of soccer games. Thus, this is one possibility, and the first standard for evaluation. To cope with these complicated problems, we will apply the autonomous agent system. Especially, the standard for evaluation is quite different in each human-being, and the feedback from the audience is an important factor for constructing an ideal **INS**. However, for effective implementation of this system, more analysis is needed about the actual soccer games which are provided in each country and their computer soccer games.

4 Conclusion

In this paper, we have tried to show the possibilities and the aim of future research of **INS** by discussing the problems of **SPACE**. Our proposed **INS** is basically based on the autonomous agent system, and is related to interactive art and virtual reality. To construct an effective virtual three-dimensional world requires the system's flexibility and the interaction between the system and the audiences or hardware, because effectiveness in this kind of world depends on

the total performance of this system and the total impression of the audiences. To provide an effective scene also requires interaction between the system and audiences, because the standard for evaluation is quite different in each humanbeing.

Because the standard of evaluation is basically different in each human-being, it is impossible to create a general **INS** based on a general standard for evaluation. The idealistic **INS** will only be a personal tool, and a different version of **SPACE** will be used in each RoboCup workshop provided by each national committee.

For developing this research issue, we welcome new research into INS.

5 Future works

This version's **Space** is an early version, and various problems were pointed out. At first, I will find a solution to these problems, and will fix them. After, we will include an autonomous agent system in one of the functions of **SPACE**, and will test this. I will release a new version of **Space** in RoboCup-Paris.

6 Acknowledgments

I would like to express special thanks to Yoko Murakami for her assistance. I also would like to express many thanks to Takemochi Ishii, Michitaka Hirose, Atsuhito Sekiguchi, Masayuki Akamatsu, Hideyuki Morita, Gayle D. Pavola and Hidekazu Hirayu for useful discussions. This research is supported by Softopia Japan.

References

- 1. 3D Planet.: 3D Assistant. http://www.3dplanet.com/
- Atsushi Shinjoh.: The construction of the interactive space. In Proceedings of the First International Conference on GeoComputation, volume-II, pages 739-749, Leeds, UK, September 1996.
- 3. Gibson, J. J.: The Preception of the Visual World. Allen and Unwin:London, 1950.
- Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa.: RoboCup:The Robot World Cup Initiative. In Proceedings of IJCAI-95 Workshop on Entertainment and AI/Alife, Montreal, 1995.
- Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa.: RoboCup:The Robot World Cup Initiative. In Proceedings of The First International Conference on Autonomous Agent(Agent-97), Marina del Ray, The ACM Press, 1997.
- Itsuki Noda.: Soccer Server Manual. http://ci.etl.go.jp/~noda/research/kyocho/ soccer/manual.newest/main.html.
- 7. Maes, P.: Modeling Adaptive Autonomous Agents. Artificial Life, 1(1994) 135-162
- OpenGL Architecture Review Board.: OpenGL Programming Guide. Addison Wesley, 1993.

- 9. Rodger Lea, Kouichi Matsuda, and Ken Miyashita.: Java for 3D and VRML worlds. New Riders publishing, 1996.
- 10. SunMicrosystems, Inc.: Java 3D API Specification. http://java.sun.com/products/ java-media/3D/forDevelopers/3Dguide/j3DTOC.doc.html, 1997.
- 11. VRML consortium. VRML97 specification. http://www.vrml.org/Specifications/ VRML97/.