Scenario-based 3D Objects Synthesizing System Design

**Ji-Seung Nam*, Hui Gao*, Mi-Young Kang*, Kyoung-Tae Kim*,
Seung-Chul Son*, Chung-Ung Pom*, and Kwon Heo***

Abstract: This paper proposes the framework of the scenario-based 3D image synthesizing system that allows common users who envision a scenario in their mind to realize it into the segments of cool animation. We focused on utilization of the existing motions to synthesize new motions for the objects. The framework is useful to build a 3D animation in game programming with a limited set of 3D objects.

We also propose a practical algorithm to reuse and expand the objects. This algorithm is based on motion path modification rules. Both linear and nonlinear curve-fitting algorithms were applied to modify an animation by key frame interpolation and to make the motion appear realistic.

Keywords: 3D Editing, 3D Synthesizing, 3D Object Reuse

1. Introduction

The proposed framework of the scenario-based 3D objects synthesizing system in this paper is made up of a database of 3D objects, 3D image file analyzer and loader, 3D object buffer, 3D image rendering unit and 3D image synthesizing unit.

Some well-known tools are capable of making 3D objects and editing a variety of 3D image file formats. However, many 3D models consist of complex hierarchical bones. Thus, it is difficult and time-consuming to change them one by one at this level [1,2,3].

The aim of the system is to make it easy to edit the motion of the 3D model without depending on any hierarchical structure [4,5]. What they need is a database including all kinds of captured 3D objects from which they can choose their desired hero and heroine of the story and an interface for viewing and easily editing these objects, for those who envision a scenario or say a story in their mind and want to convert it into the segments of cool animation.

Also, in analyzing and combing the motion data of these file formats, we put forward a method of producing new scenario motion scenes by applying the motion capture data to the motion path modification for 3D objects.

Recently, researchers have focused on creating new motions through editing the existing motions [6,7,8]. It is feasible to re-target the motion given that there are several objects with the identical bone structure and all kinds of motions in the existing database [9].

Most research on handling motion capture data has focused on techniques for modifying and varying the existing motions. Witkin and Popovic developed a method in which the motion data is warped between key frame-like constraints set by animator [10].

The space-time constraints method originally created by Witkin and Kass was developed to allow the animator to specify constraints such as the feet positions of an object [11], and then solve for these constraints by minimizing the difference from the original data [12].

This method was applied to adapt a set of motion data to objects of different sizes [13], and combined with a multi-resolution approach for interactive control of the results [14].

Physics were included in the method of Popovic and Witkin, in which the editing is performed in a reduced dimensionality space [15].

2. Overview of our proposed scheme

The structure of our scheme is illustrated in Fig. 1, the basic function of the system is to input the files of the particular style including all kinds of information about 3D objects, classify and store the different categories of the information in the specified buffers after reading and analyzing the files, and finally render the 3D models to the screen according to the synthesizing unit which provides the customized service for use. The system is developed in a Visual C++ environment using Windows API and OpenGL library.

There are many 3D image file formats such as DXF file, BVA file, ASC file, OBJ file and ASE file, which are importable or exportable for the well known 3D packages such as 3D Studio MAX and MAYA [16].

We included the ASE file format and BVA file format in our database. The ASE file format is short for ASCII Scene

Manuscript received February 21, 2006; accepted March 3, 2006.
This research was supported by the Ministry of Culture & Tourism, Korea, under the Culture Research Center support program supervised by the KOCCA(Korea Culture & Contents Agency)

Corresponding Author: Ji-Seung Nam
* Department of Computer Engineering, Chonnam National University, Buk-gu, Gwangju, 500-757, Korea.(jsnam@chonnam.ac.kr, kmy2221@yahoo.co.kr)

Exporter and is a text file format exportable in 3D Studio MAX environment. It is easy to read and analyze the ASE files as well as to edit them even in the general text editors such as Word Pad, and they are widely used in the field of 3D animations and games.

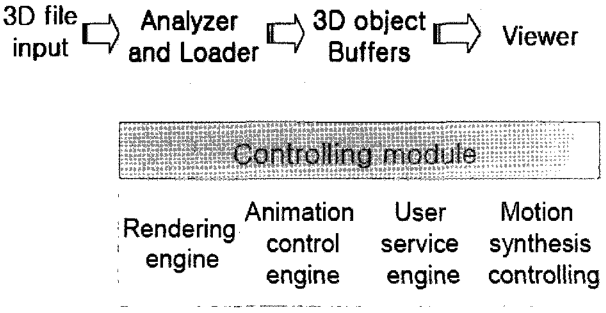


Fig. 1. Structure of the proposed system

Nowadays, motion capture data serves as a basis for creating 3D computer animations when life-like motion is desired, especially for commercial purposes [17]. The files recording motion capture data provide the details of the live motion for all degrees of freedom for the skeleton of the character. The skeleton is made up of bones. The BIOVISIONS BVA format is to record the motion of the bones.

The controlling module plays a role similar to a game engine that provides the users with an interactively friendly interface and a real time handler to edit and control the 3D scenes. We define four sub-blocks for the controlling module: the rendering engine block, the animation control engine block, the user service engine block and the motion synthesis controlling block.

After analyzing the structure of the ASE file, we will begin the procedure of reading the file, classifying and separating different information on the materials, light, and meshes of the 3D models, and then storing them in the pre-defined buffers.

Our method for managing the 3D model buffers is illustrated in Fig. 2. Each member of the linked-list *Mesh_A is aimed to point to a class which records the mesh information in *GEOMOBJECT blocks and the linked-list *Mtrl is in charge of the material information in *MATERIAL_LIST block. The Obj_counter and Mtrl_ counter are defined to calculate and record the numbers of the total objects and the material categories.

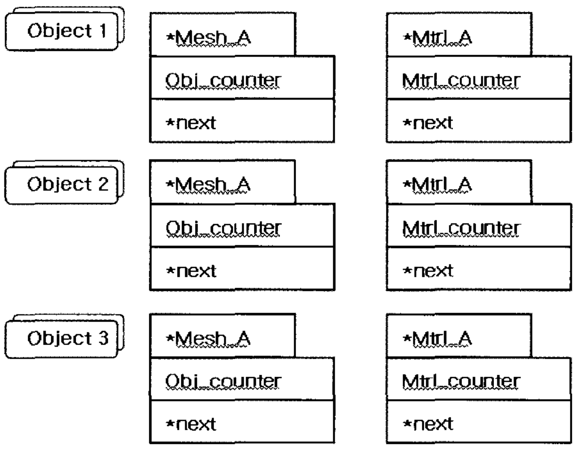


Fig. 2. Method of managing 3D model buffers

The functions used in the viewing procedures are provided with the libraries such as OpenGL and DirectX. Our design exploits the OpenGL library.

OpenGL is a library available on almost any computer for rendering computer graphics. By using it, one can create interactive applications that render high-quality color images composed of 3D geometric objects and images. Generally, OpenGL has two types that it can render: geometric primitives and image primitives.

As for rendering 3D animations, there is no mystery. As we mentioned before, in *GEOMOBJECT block, the *TM_ANIMATION sub-block contains the information on key frames of the motions. Between the key frames, we calculate the relative values of the key frames and insert the in-between frames by linear interpolation method or nonlinear method. Then, we loop the procedure so that the frames are displayed one by one with a predefined rate such as 30 fps, which makes the 3D animation work.

The synthesizing unit is in charge of how to organize the three parts above. The functionality of the synthesizing module and the style of the interface depend on the designers.

3. Motion Modifying Algorithm

It is easy to edit the motion of the single-boned model by changing its TM_ANIMATION block in ASE files because it does not have any hierarchical structure. However, nearly all 3D models are made up of complex hierarchical bones and it is difficult and not feasible to change them one by one at this level.

Basically, we applied a linear method to interpolate the frames. A timed sequence Q is a set

$$Q = \{(t_i, v_i) \mid i = 1, \ldots, n\} \tag{1}$$

where each t_i is a real number with $t_i \langle t_{i+1}$ for $i = 1 \ldots n$, and v_i can be any dimension vector but should be the same dimensional for every i. For any timed sequence Q, we can define the function interpolation (IP) as

$$IP(Q,t) = v_i + \frac{t - t_i}{t_{i+1} - t_i}(v_{i+1} - v_i) \tag{2}$$

where $i = 1 \ldots n$ and $t_i \le t \le t_{i+1}$.

In order to apply this linear interpolation function to our key frame animation algorithm, parameter time (t) is mapped to the current frame number (f)

$$IF(Q,t) = v_{j-1} + \frac{f - f_{i-1}}{f_i - f_{i-1}}(v_j - v_{j-1}) \qquad (3)$$

where $j = 1 \ldots n$ and $f_{i-1} \le f \le f_i$, because the key frame starts from zero.

It is sensible to slow down the beginning part of the animation to achieve the realistic display since the 3D object appears to move a little faster than usual because of the machine hardware and the rendering is not always perfect. The following was used to model increasing speed. We can obtain an increasing interval size with equation (4) and the time for the *jth* in-between would be calculated as equation (5).

$$1 - \cos\theta, \quad 0 < \theta < \frac{\pi}{2} \qquad (4)$$

$$t_j = t_i + (t_{i+1} - t_i)[1 - \cos\frac{j\pi}{2(n+1)}] \qquad (5)$$

The ASE format is based on identifiers in the form such as *GEOMOBJECT, which are followed by zero or more values, and then for a few of the sub-blocks of further identifiers surrounded by curly braces. Fig. 3 shows the main structure of the ASE file.

The motion parameters of each joint of the bone skeleton can be chosen to produce the new path. We define the file format '.bn' to record each new path and make an options to store them.

Commonly, there are eighteen bones defined in BVA files, so we can abstract eighteen new paths from each BVA file. After viewing the new motion path, the user can choose one suitable to the original actions of the characters of the ASE format.

```
*3DSMAX_ASCIIEXPORT 200↵
*COMMENT "AsciiExport Version 2.00- Mon Feb 7 17:49:55 2005"↵
*SCENE {···} ↵
*MATERIAL_LIST {↵
    *MATERIAL_COUNT 1↵
    *MATERIAL 0 {···}↵
}↵
*GEOMOBJECT {↵
    *NODE_NAME "  "↵
    *NODE_TM {···}↵
    *MESH {↵
        *MESH_NUMVERTEX 4↵
        *MESH_NUMFACES 2↵
        *MESH_NUMTVERTEX 12↵
        *MESH_TVERTLIST {···}↵
        *MESH_NUMTVFACES 2↵
        *MESH_NORMALS {···}↵
        }↵
    *TM_ANIMATION {···}↵
```

Fig. 3. ASE file format sample

Then, we think about motion path editing by which the motion path of one object is abstracted and applied to another object without changing its original motion behavior so that a new 3D scene can be synthesized.

A path defines a coordinate system that moves in time. The coordinate system is centered at position *T(t)* and the orientation matrix is denoted as *R(t)*. Then, the path's coordinate system is given by *T(t)R(t)*. The transformation from the global coordinate system to the local path system is derived by

$$R(t)^{-1}T(t)^{-1} \qquad (6)$$

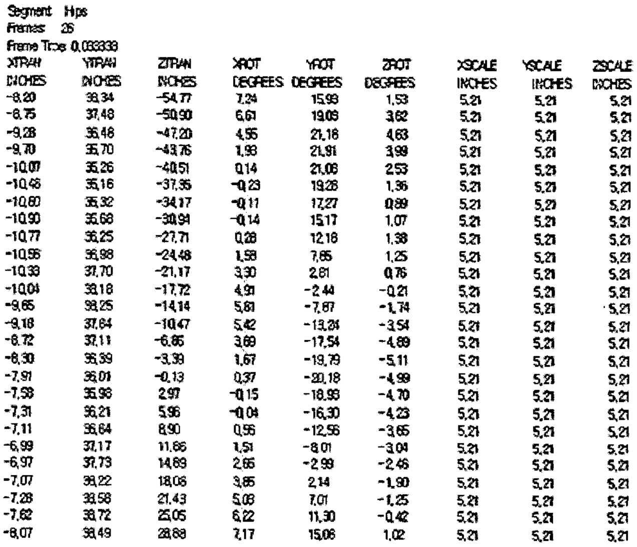
Once the path curve is changed, the new coordinate system for the object is given by,

$$T(t)R(t)R_0(t)^{-1}T_0(t)^{-1} \qquad (7)$$

where the subscript 0 denotes the initial path coordinate system and *T(t)R(t)* is the new path coordinate system. As the path *T(t)* is updated, the corresponding *R(t)* must be computed accordingly.

The proposed algorithm of new motion synthesis by editing motion path is described in the following.

Step 1: We abstract motion path from the motion capture data, which is stored in BVA files. One sample of motion capture data is listed in Fig. 4.



Segment Hips
Frames 26
Frame Time 0,033333

XTRAN INCHES	YTRAN INCHES	ZTRAN INCHES	XROT DEGREES	YROT DEGREES	ZROT DEGREES	XSCALE INCHES	YSCALE INCHES	ZSCALE INCHES
-8,20	38,34	-54,77	7,24	15,93	1,53	5,21	5,21	5,21
-8,75	37,48	-50,90	6,61	19,08	3,62	5,21	5,21	5,21
-9,28	36,48	-47,20	4,55	21,16	4,63	5,21	5,21	5,21
-9,70	35,70	-43,76	1,93	21,91	3,99	5,21	5,21	5,21
-10,07	35,26	-40,51	0,14	21,06	2,53	5,21	5,21	5,21
-10,46	35,16	-37,35	-0,23	19,26	1,36	5,21	5,21	5,21
-10,60	35,32	-34,17	-0,11	17,27	0,89	5,21	5,21	5,21
-10,90	35,68	-30,94	-0,14	15,17	1,07	5,21	5,21	5,21
-10,77	36,25	-27,71	0,28	12,16	1,38	5,21	5,21	5,21
-10,56	36,98	-24,48	1,58	7,65	1,25	5,21	5,21	5,21
-10,33	37,70	-21,17	3,30	2,81	0,76	5,21	5,21	5,21
-10,04	38,18	-17,72	4,91	-2,44	-0,21	5,21	5,21	5,21
-9,65	38,25	-14,14	5,61	-7,67	-1,74	5,21	5,21	5,21
-9,18	37,64	-10,47	5,42	-13,24	-3,54	5,21	5,21	5,21
-8,72	37,11	-6,65	3,69	-17,54	-4,69	5,21	5,21	5,21
-8,30	36,39	-3,39	1,67	-19,79	-5,11	5,21	5,21	5,21
-7,91	36,01	-0,13	0,37	-20,18	-4,99	5,21	5,21	5,21
-7,58	35,98	2,97	-0,15	-18,93	-4,70	5,21	5,21	5,21
-7,31	36,21	5,96	-0,04	-16,30	-4,23	5,21	5,21	5,21
-7,11	36,64	8,90	0,56	-12,56	-3,65	5,21	5,21	5,21
-6,99	37,17	11,66	1,51	-8,01	-3,04	5,21	5,21	5,21
-6,97	37,73	14,69	2,65	-2,99	-2,46	5,21	5,21	5,21
-7,07	38,22	18,06	3,85	2,14	-1,90	5,21	5,21	5,21
-7,28	38,58	21,43	5,08	7,01	-1,25	5,21	5,21	5,21
-7,62	38,72	25,05	6,22	11,30	-0,42	5,21	5,21	5,21
-8,07	38,49	28,88	7,17	15,06	1,02	5,21	5,21	5,21

Fig. 4. Motion capture data sample

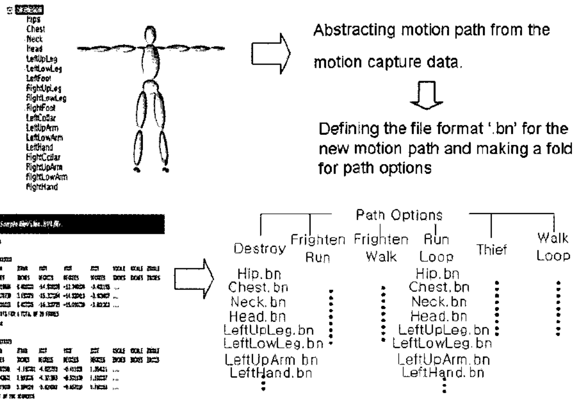
Step 2: The data of the new motion path is to be transformed into the same form that we defined for ASE files. It is necessary to match the new time sequences with the old ones. The procedure of abstracting the new motion path is shown in Fig. 5.

Attention should be given to time intervals between key frame and the total key frame number. The total key frame number of the new time sequence is not identical to the old one, which means that the timing intervals need matching.

In order to simplify the problem, we previously fixed the total key frame number for the experiments.

Step 3: The old path should be decomposed off (converting the world coordinate to the local one) before the new one is applied.

Step 4: The last step is to combine the decomposed matrix of the ASE character with the new path matrix. The illustration of applying new motion path for ASE files is shown in Fig. 6.



Abstracting motion path from the motion capture data.
Defining the file format '.bn' for the new motion path and making a fold for path options
Path Options
Destroy
Frighten Run
Frighten Walk
Run Loop
Thief
Walk Loop
Hip.bn
Chest.bn
Neck.bn
Head.bn
LeftUpLeg.bn
LeftLowLeg.bn
LeftUpArm.bn
LeftHand.bn

Fig. 5. Abstracting new motion paths procedure

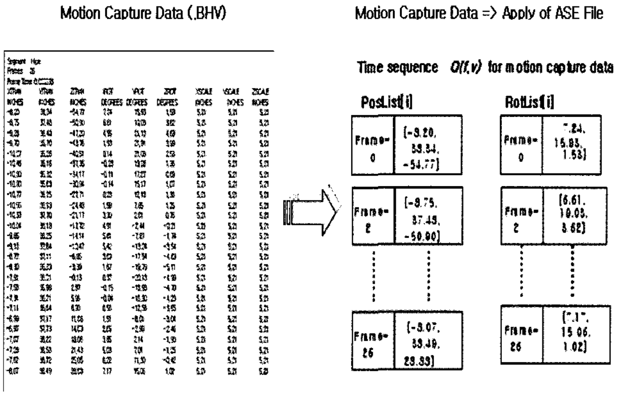
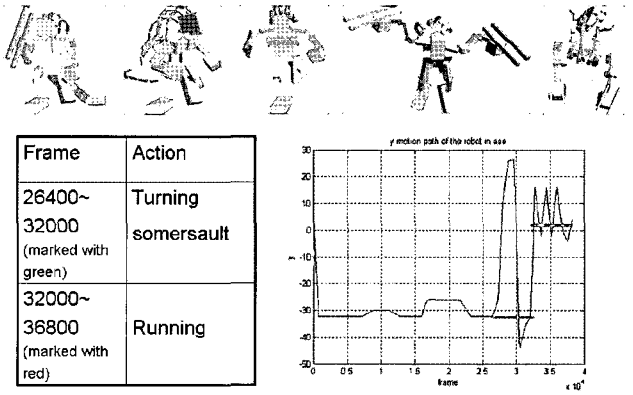


Fig. 6. Applying new motion path for ASE file

The actions and behaviors ȯf the object along the original path should be analyzed to cooperate with the new path. We analyzed the behavior of 'robot.ASE' as an example and the analysis result is shown in Fig. 7. Iterating the action 'turning somersault' or 'running' along the new path can lead to the natural synthesizing effect.

It is possible that the new path is not suitable to the original intention. For example, the size of the motion path is too small or too large. The new path should be scaled by 0.5times, 2times, and so on. X, Y and Z can be enlarged or cut scale down by the same degree or different degree according to the size of the original object. Samples of new motion path are shown in Fig. 8. If there is any reference point which the object must pass trough or avoid, it is necessary to translate or scale toward or far away from the reference point.



Frame	Action
26400~ 32000 (marked with green)	Turning somersault
32000~ 36800 (marked with red)	Running

Fig. 7. Behavior analysis of robot ASE results

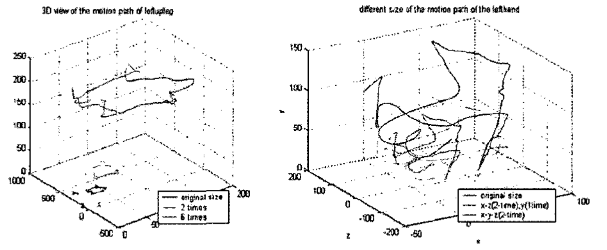


Fig. 8. New motion path example

4. Discussion

Our original motivation is to discover a general method for constructing 3D image viewing, editing and synthesizing system, and provide users with a flexible and simplified interface as well as the interactive service such as realizing their conceived scenario into vivid 3D scenes. The proposed algorithm is suitable for the common user without much professional background. It also provides an efficient way of making good use of the existing motions. In the future, we will further our study on controlling and cooperating the motion of the multiple models to improve and enhance the performance of this part.

References

[1] F. Sebastian Grassia, "Motion Editing: Mathematical Foundations," SIGGRAPH, 1999.

[2] S. W. Lee, J. G. Choi, S. D. Kim, "Scene segmentation using a combined criterion of motion and intensity," Optical Engineering, vol.36, No. 8, pp. 2346-2352, August 1997.

[3] O. Arikan, and D. A. Forsyth, "Interactive motion generation from examples," Proceeding of SIGGRAPH 2002.

[4] Michael Gleicher, "Motion Path Editing," Proceedings of the 2001 ACM, Symposium on Interactive 3D Graphics, March 2001.

[5] Michael Gleicher, Hyun Joon Shin, Lucas Kovar, and Andrew Jepsen, "Snap Together Motion: Assembling. Run-Time Animation," 2003 Symposium on Interactive

3D Graphics, April, 2003.

[6] Til Aach, Andre Kaup, and Rudolf Mester, "Statistical model-based change detection in moving video," Signal Processing, Vol. 31, No.22, pp. 165-180, Mar. 1993.

[7] J. G. Choi, M. C. Kim, M. H. Lee, and C. D. Ahn, "A User Assisted Segmentation Method for video Object Plane Generation," IEEE Trans. On Image Processing, pp. 881-898, 1996. 5.

[8] Munchurl Kim, J.G. Jeon, J.S. Kwak, M.H. Lee, C. Ahn, "Moving object segmentation in video sequence by user interaction and automatic object tracking," Image and vision Compution, vol. 19, pp. 245-260, April 2001.

[9] Michael Gleicher, "Retargeting Motion to New Characters," Proceeding of SIGGRAPH 98, pp. 33-42, July 1998.

[10] Witkin A, Popovic Z, "Motion Warping," Proceeding of SIGGRAPH 95, pp. 105-108, August 1995.

[11] Witkin A, Kass M, "Spacetime Constraints," Proceedings of SIGGRAPH 88, pp. 159-168, August 1988.

[12] Michael Gleicher, "Motion editing with spacetime constraints," Proceeding of 1997 Symposium on Interactive 3D Graphics, pp. 139-148, April 1997.

[13] Michael Gleicher, "Retargeting motion to new characters," Proceedings of SIGGRAPH 98, pp. 33-42, April 1998.

[14] Lee J, Shin S. Y, "A hierarchical approach to interactive motion editing for human-like figures," Proceedings of SIGGRAPH 99, pp. 39-48, 1999.

[15] Popovic Z, Witkin A, "Physically based motion transformation," Proceedings of SIGGRAPH 99, pp. 159-168, August 1999.

[16] Jeff Lander, "Working with Motion Capture File Formats," Game Developer, Jan 1998.

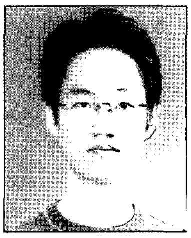
[17] Okan Arikan and david A. Forsyth and James F. O'brien. "Motion Synthesis from Annotations," ACM SIGGRAPH conference proceedings, 2003.

Ji-Seung Nam

He received his Electronics Engineering PhD degree from the University of Arizona, USA in 1992. Between 1992 and 1995, he was with the Korean Government's Electronics and Telecommunication Research Institute (ETRI) as a Senior Researcher. Since 1995, he has been a Computer Engineering Professor in the Chonnam National University's School of Electronics, Computer, and Information Communication at Gwangju, Korea. Dr. Nam had been the Director of the University's Information Communication Research Centre between 1999 and 2001. From 2001 to 2004, he had been the Director of Business Incubator at Chonnam National University. Professor Nam leads a Multimedia Data Communication Lab with his research interest being Communication Protocols, Internet Real-Time Services, Routing Techniques and Mechanisms, and Multicast Technologies. He has co-authored several technical publications in international and domestic journals and conferences.

Mi-Young Kang

She has been doing her doctorate course in the department of Computer Engineering of Chonnam National University, Korea since 2003. Before the commencement of her PhD course, Mi-Young had been involved with the Korean Government's Electronics and Telecommunication Research Institute (ETRI) Entrust Research between 1999 and 2001. In August 2001, she had completed and graduated her MS in Computer Engineering from Chonnam National University. She had been a researcher at the Information Communication Research Centre at Chonnam National University, Gwangju, Korea between 1998 and 2000 and a researcher at the Information Technology Research Centre between 2003 and 2004. Since 2004, she has been a researcher for the FTTH ETRI projects. Her research interest focuses on Computer Networks, Communication Protocols, and 3-D Graphics.



Kyoung-Tae Kim

He completed and graduated his BS in Electronics Engineering from Chonnam National University in February 2005. He has been doing his Master course in the department of Computer Engineering of Chonnam National University, Korea since 2005. He has been a research assistant for the FTTH ETRI projects since 2005. His research interest is in Computer Networks and Communication Protocols.

Choung-Ung Pom

He completed and graduated his BS in Computer Engineering from Chonnam National University in February 2005. He has been doing his Master course in the department of Computer Engineering .of Chonnam National University, Korea since 2005. He has been a research assistant for the FTTH ETRI projects since 2005. His research interest is in Computer Networks and Communication Protocols.

Kwon Heo

He completed and graduated his BS in Computer Engineering from Chonnam National University in February 2005. He has been doing his Master course in the department of Computer Engine-ering of Chonnam National University, Korea since 2005. He has been a research assistant for the FTTH ETRI projects since 2005. His research interest is in Computer Networks and Communication Protocols.