한국측량학회지, 제 28권 제 4호, 2010. 8 : pp. 403~411

# Open Standard Based 3D Urban Visualization and Video Fusion

Enkhbaatar, Lkhagva[1) · Kim, Seong-Sam[2) · *Sohn, Hong Gyoo[3)

**Abstract**

This research demonstrates a 3D virtual visualization of urban environment and video fusion for effective damage prevention and surveillance system using open standard. We present the visualization and interaction simulation method to increase the situational awareness and optimize the realization of environmental monitoring through the CCTV video and 3D virtual environment. New camera prototype was designed based on the camera frustum view model to project recorded video prospectively onto the virtual 3D environment. The demonstration was developed by the X3D, which is royalty-free open standard and run-time architecture, and it offers abilities to represent, control and share 3D spatial information via the internet browsers.

Keywords : 3D modeling, interactive demonstration, video fusion, camera prototype

## 1. Introduction

The needs for digital 3D urban models of real world and effective video monitoring system are increasing by the growth of economy in cyber world including computer game and virtual reality. Especially visualization and monitoring systems, which effectively represent the virtual environment and damage information of natural disasters in densely populated urban areas, is extremely needed.

Several related systems have been developed usually for military and commercial proposes. For example, Trivedi (Hall et al., 2002) developed the Distributed Interactive Video Array (DIVA) provides a large-scale, redundant cluster of video streams to observe a remote scene, intending for security and intelligent transportation applications. Visualization system by fusing dynamic imagery with geometry model for battlefield visualization and situational awarenes was developed (Spann et al., 2000). The VideoFlashlight system developed at Sarnoff Corporation allows immersive visualization of urban sites by offline fusion of multiple video streams and 3D site models (Kumar et al., 2000). The Video Surveillance and Monitoring (VSAM) project, conducted at CMU (Carnegie Mellon University), Sarnoff Corporation, and other institutions developed automated video understanding technologies, enabling a single human operator to monitor activities over a broad battlefield using a distributed network of active video sensors (Kanade et al., 1998). Also AVE system was developed which has the ability to fuse and display dynamic moving objects as real time 3D elements in a 3D scene model (Sebe et al., 2003).

Unlike the traditional surveillance systems providing no high-level scene comprehension, and no situational aware-

1) School of Civil and Environmental Engineering, Yonsei University · Graduate Student(E-mail:enkhbaatar@yonsei.ac.kr)
2) School of Civil and Environmental Engineering, Yonsei University · Post Doc(E-mail:samskim@yonsei.ac.kr)
3) School of Civil and Environmental Engineering, Yonsei University · Professor, *corresponding author(E-mail:sohn1@yonsei.ac.kr)

ness, this research refers to the display of the multiple sensors, which the user can view as desired and complete from arbitrary viewpoints.

Recent advances in sensing and computing technologies have started a new trend of future visualization systems for video surveillance applications. Many studies have been done on effective visualization system for video surveillance and presented the new novel methodologies based on recent advanced technologies. Some research presented the approaches for dynamic object modelling and visualization for 3D environment to track the moving objects with video textures (Sebe et al, 2004, Neumann et al., 1999). Effective visualization applications in which multiple video streams are projected onto the 3D models for better visualization have been introduced during the last few years (Wang et al., 2007, You et al., 2002 and Hu et al, 2005).

Primer purpose of the research is to demonstrate a 3D virtual urban environment and video fusion using the applicable open standard.

The research consists of 4 components: 1) Data acquisition to collect the 3D geometry sample to be used for creating 3D building models, 2) Drawing the 3D building and terrain models, 3) design the camera prototype, 4) video fusion to support improved understanding and increment of situational awareness.

## 2. Materials and Methods

### 2.1 Data acquisition

In the data acquisition phase, LiDAR (Light Detection and Ranging) sensor was employed to collect 3D geometric samples of a building. In 1990s, development of sensor technologies for LiDAR system reached a degree which satisfied the required accuracy for mapping (Lee et al., 2007).

LiDAR technology has be come very common invarious field of application such as city and building modeling, virtual reality, and mapping (Ulrich, 2003; Lee et al.,2007). The first Engineering Building at Yonsei University campus was used as sample 3D modeling. The building was scanned in 19 different positions to collect the point cloud of entire building (Fig. 1).

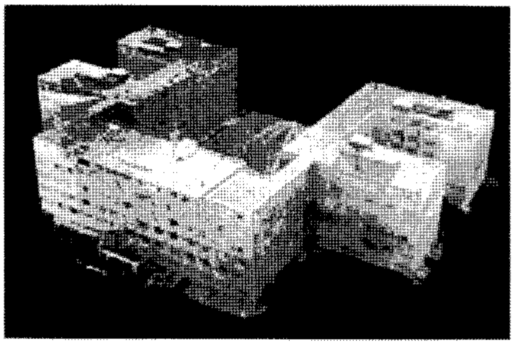

**Fig. 1. 3D view of point cloud of first Engineering Building at Yonsei University**

Density of point cloud is the important factor to impact scan time, data accuracy and utilization, and data capacity. Point cloud data was obtained with 2cm density when the range is 10m. To match the data scanned from the different positions, targets have to be installed separately in certain positions. To scan the targets accurately, we used two ways of installing targets, which are to attach targets to the building walls and objects, and use stand targets. In general, targets should be shown in scanning positions and steady. Targets have to be attached firmly to use it as reference or to register separate point clouds. For the purpose of registering point cloud or coordinate conversion, at least 3 or more have to be installed. And to increase accuracy and make the visibility certain in different position, number of targets should be increased.

Firstly, equation for 3-dimentional coordinate conversion was calculated between point clouds, using the same matching points. Then with this calculated equation, repeatedly adjust the calculation until the size of error vector in displacement among the remaining points and plane equations created with those points becomes less than regular size.

#### 2.1.1 Data vectorization

Point cloud data is very heavy itself. So, it is almost impossible to import the point cloud and process directly until it is converted to mesh (polygons such as TIN) or vector data. Major tasks in making 3D building model using LiDAR data involve classification, segmentation, and vectorization. Since 3D building model is displayed on the web, most efficient way of representation should be used. Vectorization of point cloud gives efficient and rough representation of 3D modeling. Advantage to vectorization is that data has the smallest size to visualize it on the internet. However, building is not represented precisely. Because we don't vectorize all components such as windows, doors and other small elements.

## 2.2 Drawing 3D building and terrain models

2.2.1 Drawing building model

Vectorized building model is exported as DXF (Drawing eXchange Format) and imported into the AutoCAD program to draw the 3D rough model of the building. In the final phase, we converted the 3D drawing from DXF format to X3D format in NuGraf Rendering System which is the most popular translators between graphic file formats. After the translation, first engineering building of Yonsei University looks like in Fig. 2.

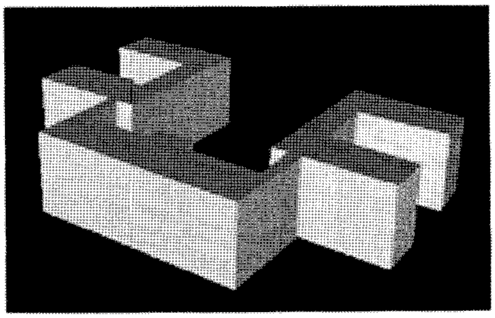

**Fig. 2. 3D building model in X3D browser**

2.2.2 Representation of ground surface

A typical 3D terrain model is represented by elevation Grid and TIN(Triangulated Irregular Network). Among the shape nodes in X3D, IndexFacedSet and Elevation Grid nodes are most appropriate to represent 3D terrain information. In the study, IndexedFaceSet node was used to create a 3D terrain model. The node has the fields defined in Table 1. This node creates a set of polygons, referred to as faces. Each polygon face is defined vertex by vertex, meaning point by point. Each point value is defined in a contained Coordinate node. IndexedFaceSet has many features. It is the most commonly used node for creation of shapes polygon by polygon.

The IndexedFaceSet nodes define geometry vertex by vertex, using index values to sequence each value from the Coordinate node point array. This approach allows for a highly compact representation of the vertices. Thus a vertex may be reused many times, but only needs to be listed once in the Coordinate node. This is an efficient way to save space when approximating a curved surface with a tightly connected mesh of polygons (Don and Leonald 2007).

**Table 1. Node syntax for IndexFaceSet**

```
<IndexedFaceSet DEF=
  "MyIndexedFaceSet"
  containerField="geometry"
  ccw="true"
  colorPerVertex="true"
  convex="true"
  creaseAngle="0"
  normalPerVertex="true"
  solid="true">
  <Coordinate DEF="DefaultCoordinate"
  containerField="coord"/>
<Color DEF="DefaultColor"
containerField="color"/>
</IndexedFaceSet>
```

In this study, elevation data of Yonsei University campus, which is from digital topography map, was used to create virtual 3D terrain model. Fig. 3 is showing the representation of 3D elevation model in X3D browser. To make the terrain model more realistic, aerial photo was projected to elevation model. Aerial photography of Yonsei University(Fig. 4) was used as a texture.

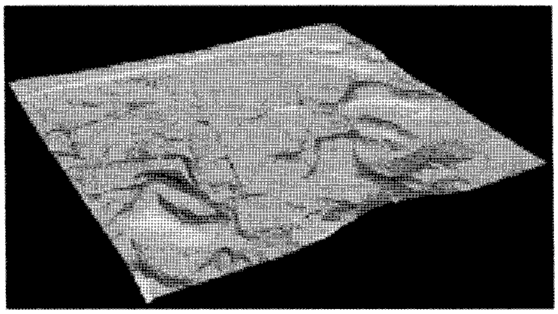

**Fig. 3. Digital elevation data in X3D browser**

In the X3D language, texturing describes the 3D graphics technique of applying 2D imagery to 3D shapes, draping a

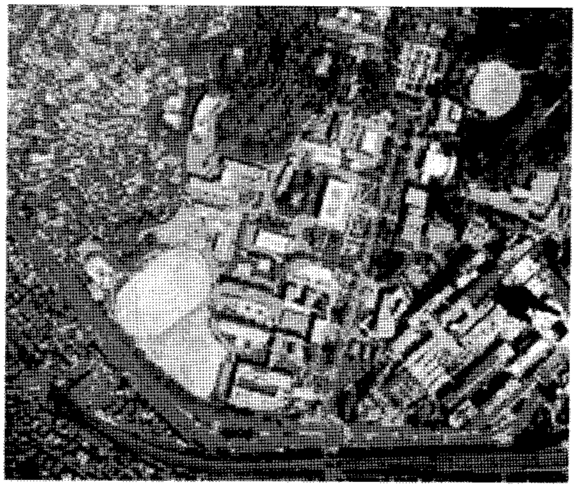

Fig. 4. Aerial photography used as a texture

picture around geometry as a way to increase visual fidelity cheaply. To cover 3D elevation model with a texture, Image Texture is used for applying the 2D aerial photography. X3D browsers support the Portable Network Graphics (.png) and Joint Photographic Experts Group (.jpg or jpeg) image file formats. These two formats offer the best quality, compression and good color fidelity for photographs. There are similar nodes in X3D library such as Pixel Texture and Movie Texture. However, Image Texture is the preferred node for texturing most objects (Don and Leonald 2007). X3D coding for texturing is shown in Table 2

**Table 2. X3D code for texturing**

```
<ImageTextureDEF="MyImageTexture"
repeatS="true"
repeatT="true"
url='"yonsei.png"
containerField="texture"/>
```

Fig. 5 is the final representation of the 3D terrain model created by IndexedFaceSet and Image Texture nodes of X3D language.

## 2.3 Design of the CCTV prototype

We used the View Frustum Model to design the CCTV model and it perspective projection in X3D language. However, there is no designated node for describing camera

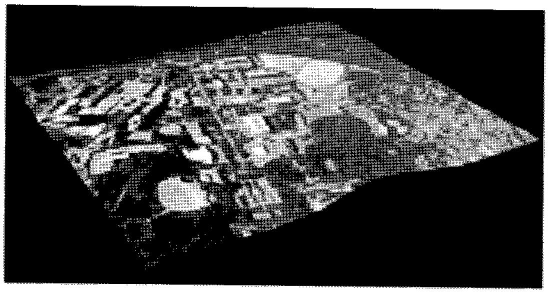

Fig. 5. 3D terrain model after draping texture

model in X3D library. Therefore, to represent the image taken by CCTV camera, we developed the new Prototype node which projects the movie on the virtual object scene considering the angle and distance from CCTV location to camera lens. There are nodes in X3D to expand the existing node or to define a new node which is not defined in X3D specification. ProtoDeclare and ProtoInstance are the designated nodes to create new features such as camera location, angle and shooting range. Appendix 1 shows the frustum model of camera which includes view point, virtual screen, and view volume. ProtoInterface node defines variables used as input fields in CCTV Prototype, whereas Protobody node presents the imagery appears on the virtual space. With the CCTV Prototype defined in Appendix 1, we used the ProtoInstance node to build the real CCTV Instance as written in Table 3. When define the Prototype, CCTV instance is created specifying input fields such as CCTV camera name, position, orientation, and location to the video file as parameters.

## 2.4 Example of CCTV video

The CCTV Prototype explained above is created using the real video recorded from Yonsei University campus. Position coordinates where video is recorded by camcorder are as follows:

- Sequence 03 : 194244.5708 , 451104.6036, 25
- Sequence 04 : 194240.4971 , 451043.5815, 25
- Sequence 06 : 194364.3976 , 451060.1345, 1.6

Three CCTV Instance are generated using camera position, orientation, and location of the video as Instance filed values. X3D code which was used to create CCTV Prototype is shown in Appendix 2. In the code, CCTV coordinates in Italic font represents the camera absolute position, whereas coordinates in bold style is the relative coordinate created by calculating distance from camera view point to the origin. This value varies with difference between local and global coordinates of virtual environment using 3D spatial information. Three CCTV Instances created by the following code are visualized in X3D browser as shown in Fig. 6.

**Table 3. Example for prototype instance**

```
<ProtoInstance name='CCTV'>
  <fieldValue name='description' value='CCTV Camera 1'/>
  <fieldValue name='position' value='0 0 0'/>
  <fieldValue name='orientation' value='0 1 0 0'/>
  <fieldValue name='url' value='Sequence3.mpg'/>
</ProtoInstance>
```

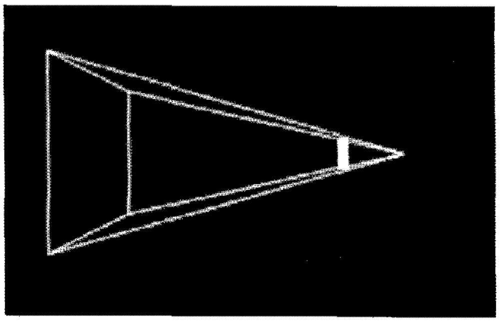

**Fig. 6. Side view of CCTV prototype**

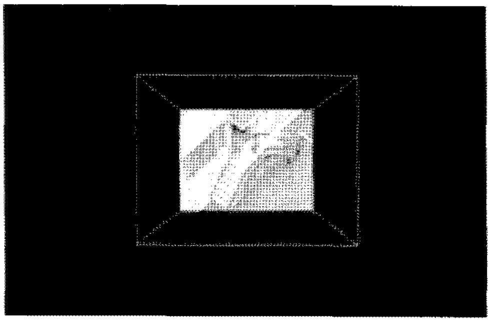

**Fig. 7. Front view of CCTV prototype**

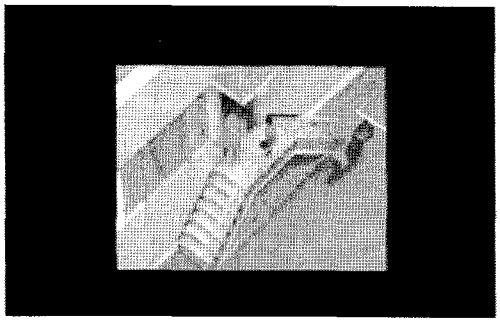

**Fig. 8. View of CCTV prototype from camera position**

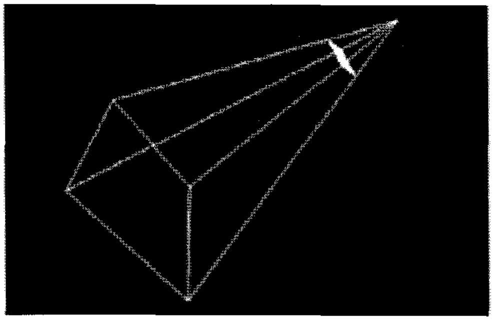

**Fig. 9. Example of view at random angle**

CCTV Instance, created by code in Table 3, is shown in Fig. 6 - 9. CCTV Instance can be created with desired position and orientation in arbitrary 3D space. If user moves to the Viewpoint where exactly CCTV camera locates, user can see the video, taken by the camera, is being projected at the same position and size as shown in Fig. 8.

## 2.5 Texture mapping

Texture mapping has been used for long time especially in computer graphics imagery. Texture mapping increases the visual effects in 3D scene. There are several research have been done on texture mapping problems especially for video surveillance and military purposes (Sebe et al., 2003).

Images are captured form the real world to create accurate and more realistic appearance rendering scene. Most graphics systems provide the high quality texture mapping operation, and thanks to the enhancement of graphics hardware and developed rich graphical libraries. They are the key factors for texture mapping applications.

### 2.5.1 Movie texture

To project the CCTV video onto the any location of 3D scene geometry, we locate the camera in virtual 3D space and project the texture image on the virtual screen located in front of the camera (Fig. 10). If we look into the virtual space from

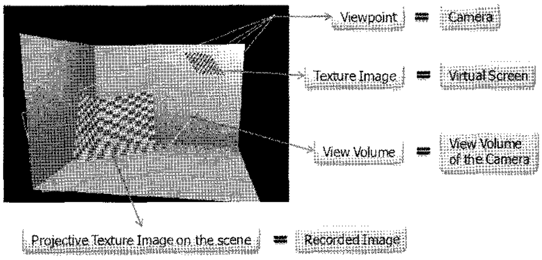


**Fig. 10. Principle of video texture mapping**

the location where video taken, image will look as actual CCTV is recording. To map the image captured from CCTV and video texture location in virtual space, correlation operation is used between camera model and viewpoint as shown in Fig. 10.

Movie texturing is implemented with MovieTexure node in X3D language. MovieTexture node has many parameters to control the texture mapping varies with time. URL field navigates to the video location used in MovieTexture node. And it supports the international standards for video file formats such as ISO/IEC 11172-1 MPEG1-Systems (audio and video) and MPEG1-Video (video-only).

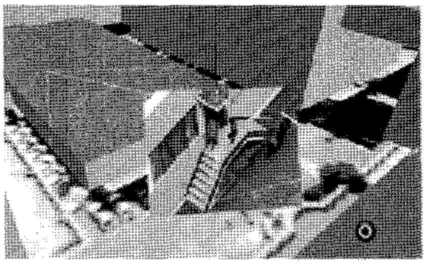

**Fig. 11. Camera model with static (recorded video projection)**

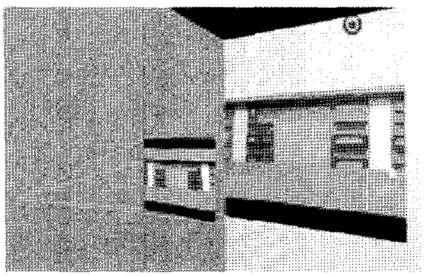

**Fig. 12. Camera model and video texture mapping on a flat object**

**Table 4. Example for movietxture node usage**

```
<MovieTexture
DEF="MyMovieTexture"
url='"MovieName.mpg"'
loop="false"
pauseTime="0"
repeatS="true"
repeatT="true"
resumeTime="0"
speed="1.0"
startTime="0"
stopTime="0"
containerField="texture"/>
```

# 3. IMPLEMENTATION RESULT

X3D Scene which combined 3D spatial information and CCTV video is completed according to the process described above. As a result, following components were developed.

- 3D terrain model of Yonsei University. 3D terrain model is important component to make 3D virtual visualization realistic. In the model drawing phase, 3D terrain model was drawn with elevation data of Yonsei University and its aerial photography used as a texture. IndexedFaceSet node of X3D was used to create the 3D terrain model.

- 3D building of 1st and 2nd engineering building of Yonsei University. In the data acquisition phase, building models were built with rough vector model which is extracted from the LiDAR point cloud. And then converted into X3D format to display it in X3D browser. Building models were not draped with textures. To make the 3D virtual environment more realistic, we should cover the buildings by suitable textures. One of the important step for making 3D virtual environment is to match building models for terrain model. Building models should be put on the terrain model matching its exact position. After creating building models, we put the models on the terrain model. However, there was a slight error in matching in terms of scaling and locating. Errors are corrected with X3D Scaling and Translation nodes.

- Six viewpoints. Six viewpoints were made to navigate and look around the 3D virtual environment. It can be controlled by clicking the PageUp and PageDown button to switch between Viewpoints in X3D browser.

- 6 CCTV camera models with video projection. CCTV camera model is one of important contributions of this research. Camera prototype was newly designed based on camera frustum model. We recorded short videos for each camera model. Videos were taken in school campus around the 1st and 2nd engineering buildings. Video files are implemented with the MovieTexture node of X3D.

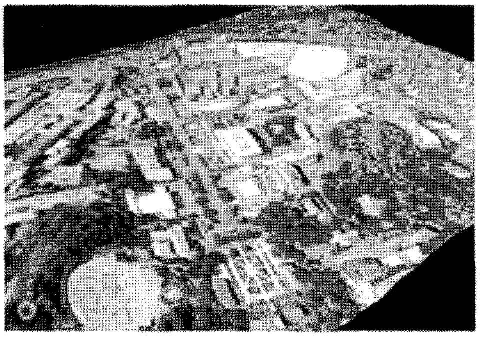

**Fig. 13. Screen shot from viewpoint 1 in X3D scene**

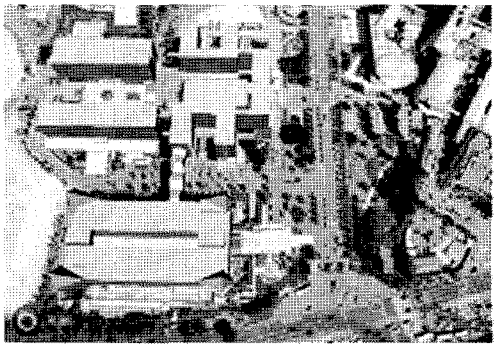

**Fig. 14. Screen shot from viewpoint 4 in X3D scene**

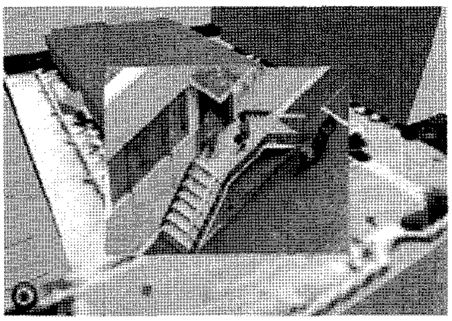

**Fig. 15. Video display at location of CCTV camera 1**

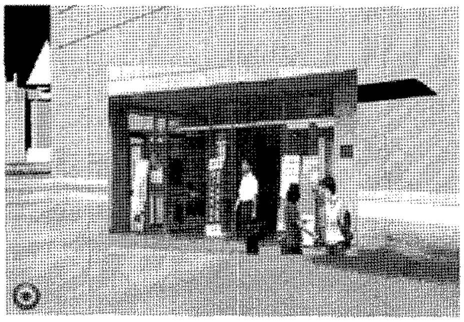

**Fig. 16. Video display at location of CCTV camera 3**

The following figures show the screen shots captured from the X3D browser.

## 4. Conclusions

In this research, we have proposed to demonstrate a 3D virtual visualization of urban area and video fusion for damage prevention and surveillance system using the X3D open standard. Also a new node for camera prototype was designed by using ProtoDeclare and Protobody nodes. We present the visualization and interaction simulation method to increase the situational awareness and realize monitoring environment through the CCTV video and 3D virtual environment. Important approach of the research was that the simulation is developed with opportunity to browse and access via the internet freely. The research can offer a solution for effectively modeling, visualizing, and presenting 3D spatial information gathered by multiple sensors (CCTV & LiDAR) in our 3D world in order to promote better understanding and comprehension. The solution is particularly suited to wide-area surveillance systems involving many cameras and sensors to cover complex sites such as military bases, government facilities, airports, commercial infrastructure, as well as disaster management, crime monitoring, car accident and traffic monitoring systems in city area.

## Acknowledgement

This research was supported by Seoul R&BD Program (10540).

## References

Don, B. and Leonard. D. (2007), *X3D Extensible 3D Graphics for Web Authors*, Elsevier Inc, ISBN-13: 978-0-12-088500-8.

Hall, B. and Trivedi, M. (2002), A novel graphical interface and context aware map for incident detection and monitoring, *9th World Congress on Intelligent Transport Systems*.

Hu, J., You, S. and Neumann, U. (2005), Texture Painting from Video, *The Journal of WSCG*, Vol. 13.

Kanade, T., Collins, R., Lipton, A., Burt P. and Wixson, L. (1998). Advances in cooperative multi-sensor video surveillance, *Proc. of DARPA Image Understanding Workshop*, Vol. 1, pp. 3-24.

Kumar R., Sawhney, H. S., Guo, Y. and Hsu, S. Samarasekera. (2000), 3D manipulation of motion imagery, *ICIP2000*.

Lee D. C., L., Jung H. S., and Yom J., H. (2007), 3D Building Reconstruction and Visualization by Clustering Airborne LiDAR Data and Roof Shape Analysis, *Korean Journal of Geomatics*, Vol. 25, No. 6-1, pp. 507-516.

Neumann, U., You, S., Cho, Y., Lee, J. and Park., J. (1999), Natural Feature Tracking for Augmented Reality, *IEEE Transactions on Multimedia*, Vol. 1, No. 1, pp. 53-64.

Sebe, O. I., Hu, J., You, S. and Neumann, U. (2003), 3D video surveillance with Augmented Virtual Environments, *First ACM SIGMM international workshop on Video surveillance*.

Sebe, O. I., You, S. and Neumann, U. (2004), Dynamic Objects Modeling and 3D Visualization, *ASPRS'04*, Colorado.

Spann, J. R. and Kaufman, K. S. (2000), Photogrammetry using 3D graphics and projective textures, *IAPRS*, Vol. 33.

Ulrich, N., Suya, Y. and Jinhui, H. (2003), Augmented virtual environments (AVE): dynamic fusion of imagery and 3D models, *Proceeding of the IEEE Virtual Reality*.

Wang, L., You, S. and Neumann, U. (2007), Single view camera calibration for augmented virtual environments, *IEEE Virtual Reality Conference*, pp. 255-258.

Yoo, B., Brutzman, D. (2008). Real-time 3D Simulation Infrastructure for Practical Application of High-resolution Satellite Imagery. *Proc. of International Symposium on Remote Sensing*, Daejeon, Korea.

You, S., Hu, J., Jiang, B., Lee, J. and Neumann, U. (2002), Creation of Augmented Virtual Environments by Dynamic Fusion of Imagery and 3D Models, *Workshop on Intelligent Human Augmentation and Virtual Environments*. DARPA/ARL/University of North Carolina at Chapel Hill.

(접수일 2009. 11. 23, 심사일 2009. 12. 03, 심사완료일 2010. 08. 14)

**Appendix 1. Declaration of prototype for CCTV camera and video**

```
<ProtoDeclare name='CCTV'>
 <ProtoInterface>
  <field accessType='inputOutput' name='position' type='SFVec3f'/>
  <field accessType='inputOutput' name='orientation' type='SFRotation'/>
  <field accessType='inputOutput' name='url' type='MFString'/>
  <field accessType='inputOutput' name='description' type='SFString'/>
 </ProtoInterface>
 <ProtoBody>
  <Viewpoint>
   <IS>
    <connect nodeField='position' protoField='position'/>
    <connect nodeField='orientation' protoField='orientation'/>
    <connect nodeField='description' protoField='description'/>
   </IS>
  </Viewpoint>
  <Transform>
   <IS>
    <connect nodeField='translation' protoField='position'/>
    <connect nodeField='rotation' protoField='orientation'/>
   </IS>
   <Transform translation='0 0 -6'>
    <Shape DEF='Projection_Plan'>
     <Appearance>
      <MovieTexture loop='true'>
       <IS>
        <connect nodeField='url' protoField='url'/>
       </IS>
      </MovieTexture>
     </Appearance>
     <Box size='4 3 0'/>
    </Shape>
   </Transform>
   <Switch whichChoice='-1'>
    <Shape>
     <Appearance>
      <Material diffuseColor='0 0 0' emissiveColor='1 0 0'/>
     </Appearance>
     <IndexedLineSet colorPerVertex='false' coordIndex='1 2 3 4 1 -1'>
      <Coordinate point='0 0 0 2 1.5 -6 -2 1.5 -6 -2 -1.5 -6 2 -1.5 -6'/>
     </IndexedLineSet>
    </Shape>
   </Switch>
   <Transform scale='5 5 5'>
    <Shape>
     <Appearance>
      <Material diffuseColor='0 0 0' emissiveColor='0 1 0'/>
     </Appearance>
     <IndexedLineSet colorPerVertex='false' coordIndex=
      '0 1 -1 0 2 -1 0 3 -1 0 4 -1 1 2 3 4 1 -1'>
      <Coordinate point='0 0 0 2 1.5 -6 -2 1.5 -6 -2 -1.5 -6 2 -1.5 -6'/>
     </IndexedLineSet>
    </Shape>
   </Transform>
  </Transform>
 </ProtoBody>
</ProtoDeclare>
```

**Appendix 2. Instance of CCTV video using declared prototype**

```
<ProtoInstance name='CCTV'>
 <fieldValue name='description' value='CCTV Camera 1'/>
 <!--fieldValue name='position' value='194244.5708 25 -451104.6036'/-->
 <fieldValue name='position' value='-12.5469 53.9803 54.75'/>
 <fieldValue name='orientation' value='-0.116351 -0.947986 -0.296286
2.43004'/>
 <fieldValue name='url' value='Sequence3.mpg'/>
</ProtoInstance>
<ProtoInstance name='CCTV'>
 <fieldValue name='description' value='CCTV Camera 2'/>
 <!--fieldValue name='position' value='194240.4971 25 -451043.5815'/-->
 <fieldValue name='position' value='-15.9688 54.8909 114.094'/>
 <fieldValue name='orientation' value='-0.190061 -0.907786 -0.3739
2.26521'/>
 <fieldValue name='url' value='Sequence4.mpg'/>
</ProtoInstance>
<ProtoInstance name='CCTV'>
 <fieldValue name='description' value='CCTV Camera 3'/>
 <!--fieldValue name='position' value='194364.3976 1.6 -451060.1345'/-->
 <fieldValue name='position' value='102.875 29.5628 95.5313'/>
 <fieldValue name='orientation' value='0 1 0 2.00044'/>
 <fieldValue name='url' value='Sequence6.mpg'/>
</ProtoInstance>
```