IFC and OpenGL-based representation and development of 3D realistic model in virtual construction

Zhang Jianping, Yu Fangqiang, Wu Dapeng
Dept. of Civil Engineering, Tsinghua University, Beijing, China

Abstract
In recent years, virtual construction (VC) has been widely applied, as it can improve efficiency, reduce cost and shorten period for construction projects. However, existing VC systems still have a number of disadvantages that limit their usage, including requirement for high-fed computing hardware, high deployment cost and limited interoperability with other applications of architecture, engineering, construction, and facility management (AEC/FM) industries. Integrating VC technology, IFC and OpenGL, a lightweight and low-cost VC platform is developed to address these limitations in this research. It has the ability of sharing and exchange data with other applications of AEC/FM industries. This paper focuses on the IFC-based 3D realistic model, which is the basis of VC model in the VC platform, and the development of this platform. Then an application of this VC platform to a real word project is introduced. The experimental results present a practical and economic way to achieve VC which has interoperability with other AEC/FM applications.

Keywords: virtual construction, realistic model, industry foundation classes, OpenGL

1 Introduction
Virtual construction (VC) is a computer aided construction and management approach that allows practitioners to perform the construction process virtually by simulation technology. With the help of VC, practitioners are able to identify problems that may occur in actual construction process so that avoiding these problems in the actual construction is possible. Accordingly, VC can improve efficiency, reduce cost and risk and shorten construction period (Zhang et al., 2003). In recent years, it has been widely used in architecture, engineering, construction, and facility management (AEC/FM) industry. However, existing VC systems still have a number of limitations, such as requirement on high-end hardware, high deployment cost, and especially the limited interoperability in AEC/FM industry (Yi et al., 2007). These limitations hinder the widespread applications of the VC technology. The objective of this research is to address these limitations by developing a lightweight and low-cost VC platform through integrating VC technology, Industry Foundation Classes (IFC) and OpenGL.

IFC has been known as a common product and process model to provide interoperability among various IT systems for AEC/FM industry (Kim and Seo, 2008). Developed by International Alliance for Interoperability (IAI), IFC is widely adopted as a specification for product and process model of the AEC/FM industry, especially building information model (BIM) (Fu et al., 2006). In IFC, building information is represented by a set of predefined classes. For example, physical components of a building (e.g., walls, doors and windows) are represented by IfcBuildingElement class and its
subclasses. Information of design, construction etc., associated with relevant building physical components is captured by attributions of the IfcBuildingElement and its corresponding property sets linked through IfcPropertyDefinition. To support rendering in VC, a 3D realistic model is developed in this research as a product model presenting material definition of surfaces (e.g., colour, diffuse reflectance and texture) through IFC classes such as IfcSurfaceStyle. Based on this model, colour, lighting effect and texture of the product model can be obtained on a VC platform. Furthermore, the data in this model can be shared with other applications through exporting as an IFC file.

OpenGL is a standard specification defining a cross language for programming applications that produce 2D and 3D computer graphics. It is widely used in CAD, virtual reality, scientific visualization, etc. (Shreiner et al. 2005). In this research, an OpenGL based lightweight 3D graphic platform is developed to visualize construction process realistically. Although the result may be a little less realistic, this platform significantly lowers the requirement for computer hardware and makes the VC process available for a wide range of practitioners.

In this paper, information necessary for VC rendering is first described. Then the IFC-based 3D realistic model to represent the information is developed, along with the OpenGL-based VC platform built on top of it. Finally, a real world test case for the VC platform is introduced and the experimental results demonstrate a practical and economic way to develop a VC platform that can share and exchange data with other IFC-compliant applications.

2 The IFC-based 3D realistic model

An IFC-based 3D realistic model is created in this research. It contains both geometric representation and material definition of physical building components. In IFC, the geometric representation is defined by IfcProductRepresentation class and relevant classes in GeometryResource schema, GeometricModelResource schema, ProfileResource schema, and TopologyResource schema. A detailed description of the geometric representations in the IFC has been presented in (Kim and Seo, 2008). Therefore, this paper focuses on the material representation in the IFC, including colour, lighting effect and texture, as well as the method to associate material with product model.

2.1 Colour model

Colour is the visual perceptual property corresponding in humans to the categories called red, green, blue and others. RGB colour model is the most widely used colour model for sensing, representation, and display of images in IT systems. It is an additive model in which red, green, and blue light are added together in various ways to reproduce a broad array of colours (Buss 2003). In IFC, colour is represented by IfcColourRgb class, which is based on the RGB colour model. It has three attributions: Red, Green and Blue, determining the red, green, and blue light in the model, respectively.

2.2 Material definition for rendering

Looking at a physical surface, human eyes’ perception of colour depends on the distribution of photon energies that arrive and trigger our cone cells. Those photons come from a light source or combination of sources, some of which are absorbed and some of which are reflected by the surface (Buss 2003). In addition, depending on materials of the surface, different surfaces may have very different properties. Computer graphics approximates light as if it can be broken into red, green, and blue components. Thus, the colour of light sources is characterized by the amount of red, green, and blue light they emit. In addition, computer graphics adopts different kinds of light: ambient, diffuse and specular light, to approximate lights coming from different light sources in real world.

The material of surfaces is characterized by the percentage of the incoming red, green, and blue components that is reflected in various directions. Like lights, materials have different ambient,
diffuse, and specular colours, which determine the ambient, diffuse, and specular reflectances of the material. Ambient and diffuse reflectances define the colour of the material. Specular reflectance is usually white or gray, so that specular highlights end up being the colour of the light source’s specular intensity. Moreover, material has a shininess, which determines the area of specular highlight (Shreiner et al. 2005).

In IFC, material of a surface is represented by IfcSurfaceStyleElementSelect class. It is a select type that contains five subtypes (e.g., IfcSurfaceStyleShading presenting a RGB colour and IfcSurfaceStyleWithTextures presenting texture). Therefore texture and colour can be related to the product model through IfcSurfaceStyleElementSelect class. The IfcSurfaceStyleRendering, a subtype of IfcSurfaceStyleShading has four attributions: SurfaceColour, DiffuseColour, SpecularColour and SpecularHighlight, determining the ambient, diffuse, specular reflectances and shininess of the material respectively.

2.3 Texture and texture mapping

Texture technology, including texture generating and texture mapping, is used to represent the details of a surface effectively and efficiently in order to construct a realistic model. Textures are simply rectangular arrays of data, such as colour data and luminance data, which are used to represent the fine structure of the surface. The data may come from an image or an explicit array of pixel values. Texture mapping is a method to map the data to surfaces of 3D model. It determines which pixel’s colour or luminance data is used to render for every pixel of surfaces (Shreiner et al. 2005).

Texture mapping is determined by object coordinates and texture coordinates for each vertex of surfaces. The object coordinates are described by 3D Cartesian coordinates of every vertex; and the texture coordinate is coordinate of a texture. For instance, texture coordinates of the four rectangular vertexes of a 2D texture are (0, 0), (0, 1), (1, 1), (1, 0), respectively. Each surface of realistic model with a specific texture has two arrays, namely texture coordinates array and object coordinates array. Then every pixel of the surface can obtain its texture data by linear interpolation based on the texture coordinates and relevant object coordinates given by the above-mentioned two arrays. In addition, another method to obtain texture coordinates of each pixel is to calculate automatically by an assigned function (Shreiner et al. 2005).

IFC uses IfcSurfaceTexture and its subtypes to represent the texture. Subtype IfcPixelTexture has an array of pixel values determined by the attribution Pixel. Another subtype IfcImageTexture refers to an existing image determined by the attribution UrlReference, which is the URL of the image.

Texture mapping is represented by IfcTextureCoordinate and its subtypes. Subtype IfcTextureCoordinateGenerator presents the automatic calculation method of texture mapping; its attribution Mode determines the function and attribution Parameter determines the parameters for the function. Subtype IfcTextureMap has a set of IfcVertexBasedTextureMap, which has an array of texture coordinates, given by attribution TextureVertices, and an array of object coordinates, given by attribution TexturePoints. Each instance of IfcVertexBasedTextureMap determines the texture mapping of a surface of the realistic model.

2.4 Methodology of correlation between material and product model

Figure 1 illustrates the 3D realistic model is built upon IFC in EXPRESS-G language, including product model, material definition and geometric representation. In IFC, IfcSurfaceStyle, a subtype of IfcPresentationStyleSelect, can be assigned to IfcRepresentationItem via the IfcPresentationStyleAssignment and IfcStyledItem. Providing a geometric representation of the IfcProduct, IfcRepresentationItem is an item of IfcRepresentation that is related to IfcProduct. As the IfcBuildingElement is a subtype of IfcProduct, material presented by IfcSurfaceStyle can be associated with corresponding product model. Accordingly, the IFC-based realistic model is a complete realistic model represented by IFC and of which the data can be shared with other applications.
3 Development of OpenGL-based VC platform

Figure 2 shows the architecture of the IFC and OpenGL-based VC platform. A VC model is the basis of VC. It contains all the information for VC and results of VC, including 3D realistic model, construction and design information, etc. The realistic model that includes geometric representation and material definition of a building’s physical components is represented in the IFC as described above. The construction information and design information will be described in IFC as well in the second phase of this research. The platform provides an interface to import IFC models generated by other IFC-compliant software, such as Architectural Desktop and Microstation (Yi et al. 2007). In addition, through this interface, the IFC-based 3D realistic model can be exported as an IFC file, so that other IFC-compliant software can share the model. This platform also provides an interface to set and modify material definition of the model and some interfaces for other requirements of interaction.

Based on this VC model, construction simulation can be carried out by construction simulation systems, such as the 4D-GCPSU (4D Graphics for Construction Planning and Site Utilization, a construction simulation and management system developed by Tsinghua University (Wang et al.
2004), after which the OpenGL-based 3D graphic platform developed in this research is able to visualize the result and process of construction simulation realistically.

![Virtual Construction Model]

This VC platform is developed on .Net platform using OpenGL and C#. The generated executable file is only 5.2Mb, which is lightweight comparing to existing VC systems.

4 An example of the VC platform’s application

The VC platform was tested in a real world construction project, Guangzhou West Tower, a 102-floor and 432-meter high building located in the CBD area of Guangzhou, China. First, the existing 3D geometric model, which had been built with the Architectural Desktop by architects, was exported to an IFC file. Then, the IFC file was imported into the VC platform and the 3D geometric model was generated. After that, material information was added to the 3D geometric model through the material definition interface. Therefore, we generated the realistic model rapidly. Figure 3 shows the realistic model rendered by the VC platform. Figure 4 shows the lighting effect and texture effect of the model. Based on this realistic model, VC was preformed with this platform on commodity computer effectively and efficiently. In addition, the realistic model was exported to an IFC file, so that other IFC-compliant software can share this realistic model. Figure 5 shows a part of the file that is a standard wall object with material definition. The application proves that the realistic model provided by this platform is good enough to support VC and the IFC-based realistic model can be shared and exchanged between this platform and other IFC-compliant applications through IFC files.

![Figure 3. Realistic model of the West Tower.](image1)

![Figure 4. Lighting effect and texture effect in the model.](image2)
5 Conclusion

This research presents an IFC-based realistic model, which can be shared with other applications through exporting as an IFC file. Then, based on this model a VC platform is developed by using VC technology and OpenGL. In general, the realistic model rendered by this platform is able to support VC tasks and the platform can provide interoperability, including data exchange and sharing, with other AEC/FM applications. In conclusion, this paper presents a practical and economic way to achieve VC which has interoperability with other AEC/FM applications in order to overcome limitations of existing VC applications.

Acknowledgements

The work reported here is supported by the National Technological Support Program for the 11th Five-Year Plan of China (No. 2006BAJ01B07).

References


