Level of Length Detail for Representing Virtual Objects' Real Length

Myeong Won Lee, Chang Hyuck Im, Yong Duck Lee*
Dept. of Internet Information Engineering, The U. of Suwon
*Dept. of Computer, The U. of Suwon
mwlee@suwon.ac.kr, singrey@gmail.com, yorer@hanmail.net

가상 객체의 실제 길이 표현을 위한 다중 레벨

이명원 임창혁 이용덕*
수원대학교 인터넷정보공학과
*수원대학교 컴퓨터학과

Abstract

Current computer graphics technology creates and displays virtual objects in a normalized environment. We cannot know or assume the real physical properties of objects related to appearance without textual information. It is also difficult to represent any two objects in relation to each other when the difference between the two objects' size is large because of the limited resolution of the computer display. In order to solve the problem, we define and implement the real length property among the physical properties in virtual environments. We define the concept of LOLD (Level of Length Detail) to represent real-world length for objects in metric units such as millimeter, meter, kilometer, etc.

요 약

현대의 컴퓨터 그래픽스 기술은 정규화된 좌표 및 장치 공간에서 가상객체를 생성하여 디스플레이 한다. 우리는 가상 객체의 그래픽스 기술 외에 텍스트 정보를 별도로 제공하지 않으면 이 객체와 관련된 실세계에서의 물리적 속성을 알아낼 수가 없다. 또한, 컴퓨터 디스플레이의 제한된 해상도 때문에 두 물체 크기의 차가 매우 클 경우에는 동시에 디스플레이할 수 없게 되어 그 두 물체 크기를 시각적으로 정확하게 비교할 수가 없게 된다. 본 논문에서는 이러한 문제를 해결하기 위하여 가상환경에서 물리적 속성 중 하나인 실제 길이를 정의하고 이를 구현하는 방법을 정의한다. 또한, 밀리미터(mm), 미터(m) 및 킬로미터(km)와 같은 실세계 물리적 길이 단위로 물체의 실제 정확한 길이를 표현하기 위한 LOLD (Level of Length Detail) 개념을 정의한다.

Keywords : virtual object, physical property, physical length, LOLD, Level of Length Detail

1. Introduction

Current VR technology aims to generate virtual objects that look like or resemble real world objects [1]. It is also intended to generate imaginary or toy objects for entertainment purposes. In this paper, we will focus on the former. Generally, VR objects are generated through modeling and rendering techniques [2][3]. These techniques attempt to represent objects to reflect their real life counterparts. The current interest in computer graphics technology is how to represent objects more realistically.

The technology has been developed so that any object with any size, including real world visible objects, invisible large spatial objects, or small microscopic objects, can be represented on a computer display [4]. However, we are not able to represent the objects in their real metric length, such as centimeter, meter, etc. This is because current computer graphics technology visualizes objects not by their physical length but by normalized coordinate values which are determined based on representing the object in the most realistic manner.

However, it is not enough to represent them visually without standardized physical properties. There are many cases that require object comparison in the virtual environment. For example, suppose one wishes to compare different 3D objects such as a fruit and its qualities in different online shops. Unless the fruit is accompanied by a text description, it would be difficult to compare attributes such as size. However, with a standardized system, one would visually be able to compare the fruit side by side. We cannot recognize and compare objects' physical lengths in the virtual environment. We cannot know the difference in real lengths between two objects, especially when they are generated by different graphics tools, or if there is a very large difference in size between the objects [4].

As a result, it is necessary to define objects' properties in real physical metric units in virtual environments. It is also a requirement that a virtual reality (VR) browser analyzes the physical properties and display objects according to those properties. Furthermore, the physical properties and metric units should be defined in a standardized way so that the VR browser can display the objects measured by defined metric units.

In this paper, we define the method of real length properties for virtual objects among various kinds of physical properties to characterize their sizes accurately in the scene. It is based on a new concept of Level of Length Detail (LOLD) that defines the levels and the ranges for representing physical length metrics.

2. Objects' Physical Properties and Real Length

In this paper, physical properties of an object imply the concept for defining an object uniquely in the real world. There are many kinds of physical properties such as length, weight, humidity, degree, softness, etc. Generally, we use metric units for representing and recognizing objects in the real world. Sometimes, the metrics are used to compare two objects' qualities or characteristics. Here we demonstrate visual representation in virtual environments according to such metrics with non-textual information.

Suppose that one wishes to purchase a product in an Internet shopping mall. Naturally, objects for purchase can be compared based on the qualities mentioned in their accompanying text descriptions if the shopping mall provides that information, as is usual. However, people also tend to compare objects' qualities by their appearance or visual representation. Therefore, if we are considering a virtual shopping mall, exhibition, or museum, people need to be aware of the objects' real sizes, colors, or materials when they look at pictures of the objects. Current computer graphics technology provides properties regarding color or material for multiple object comparisons.

However, it does not provide an acceptable method of comparison based on real size or length. If we wish to compare the lengths of two objects in the virtual world that were designed or generated with different graphics tools, it would be practically impossible to do so based solely on their appearances [5]. The metrics become even more important if we represent parts or components in CAD or interior design. And the same problems can occur when we represent objects for scientific applications visually on a computer display.

This paper provides an accurate representational metric for VR objects, focusing on the real physical length property amongst the various real world physical metrics. We propose the following method for defining a physical property of an object, using the metric of length, and implementing it.

First, we define the metric of an object's real length as nm, μm, mm, cm, m, and km. Then, the concept for a physical property of length is defined as follows:

- The metric of an object's real length is defined as nanometer (nm), micrometer (μm), millimeter (mm), centimeter (cm), meter (m), and kilometer (km).
- The basic unit is represented as millimeter (mm).
- The delimiter of length scope range is 1000.
- A level of length detail is also defined for representing the range of length.

3. Level of Length Detail (LOLD)

Generally, VR objects are represented in unrealistic sizes, depending on the graphics tools used to create them, although a common object is designed [6]. For example, we often see something like a teapot being displayed larger than a house when the two objects are rendered together in a scene. Another problem occurs when we represent objects together in a scene and the difference in their sizes is very large, for example, a difference of over 1000 pixels. In these cases, we are unable to view the objects together because computer displays have limited resolutions. As a result, we need a mechanism to represent objects selectively in all length ranges so that objects can be displayed with their length properties even when they are very large, such as with spatial objects, or when they are very small, such as with microorganisms.

We define the concept of LOLD (Level of Length Detail) in order to represent various-sized objects in relation to each other on a computer display, even if those objects are designed using different tools. Five kinds of LOLD are defined based on the following length ranges:

- 1 LOLD : nm(10^{-9}) ~ μm(10^{-6})
- 2 LOLD : μm(10^{-6}) ~ mm(10^{-3})
- 3 LOLD : mm(10^{-3}), cm(10^{-2}), m
- 4 LOLD : m ~ km(10^{3})
- 5 LOLD : km(10^{3}) ~ Mm(10^{6})

Here, we consider 3 LOLD as the default length range, and one pixel as one millimeter. The LOLD can be expanded if we need more long or short length representation metrics, besides the ranges above.

4. Implementation Algorithm Based on X3D

In order to define and implement the physical properties for VR objects, we must consider three important issues. First, we must use the metric standard for defining physical properties so that they can be recognized and understood irrespective of where virtual objects have been designed. Second, a VR viewer must recognize the physical properties and represent VR objects according to their definition. There are many kinds of properties to represent VR objects, and we need a standardized description method to visualize them, including our physical property definition. To accomplish this, we used X3D and included the concept of our physical property definition [7][8]. We refer to this as enhanced X3D (E-X3D).

Third, we need a VR viewer that can understand and represent objects with physical properties. Again, this is provided by our VR viewer based on enhanced X3D. Figure 1 shows the overall system in which the physical properties can be implemented. Generally, we can obtain object modeling and rendering information described by X3D description from various traditional tools.

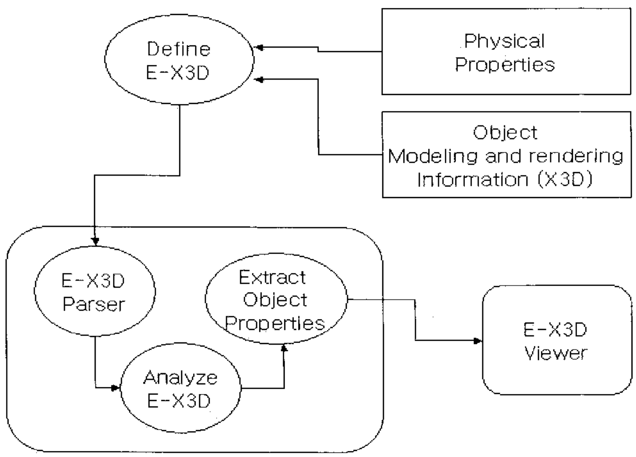


Figure 1 System organization

We have added the physical properties defined by real metric values, such as the length metric and LOLD, to the existing X3D description for an object using a Shape node and a DTD (Document Type Definition) file description. We suppose that the physical length property is applied to the total object, not to a part of an object. Using the Shape node in X3D, the physical length is defined as follows:

<Shape bboxHeight = “height”, LOLD = “LOLD level number”>

Here, bboxHeight denotes the height length property assigned by “height” real metric values, while LOLD denotes the metric unit or range defined by “LOLD level number”. The default value of LOLD is 3 and represents the range from millimeter to meter, and the unit millimeter. In addition, we describe the physical length in its corresponding DTD file for the object as follows:

bboxHeight %SFDouble; # IMPLIED
LOLD %SFInt32; “LOLD level number”

The E-X3D parser analyzes and extracts all the object properties so the viewer can render the objects according to the physical properties. The viewer performs appropriate transformations by analyzing the physical properties. The physical length properties described by real metrics and LOLD are analyzed as follows: Suppose that we are displaying an object with its height as a physical length property.

First, we calculate the scale factor using the height values of the object geometric data and of the real metric value described in the Shape node as bboxHeight. Then, the scale factor is applied to render the object in the display. The scale factor is calculated as follows:

Scale factor = Height value obtained from object's geometric data / Height value described in bboxHeight

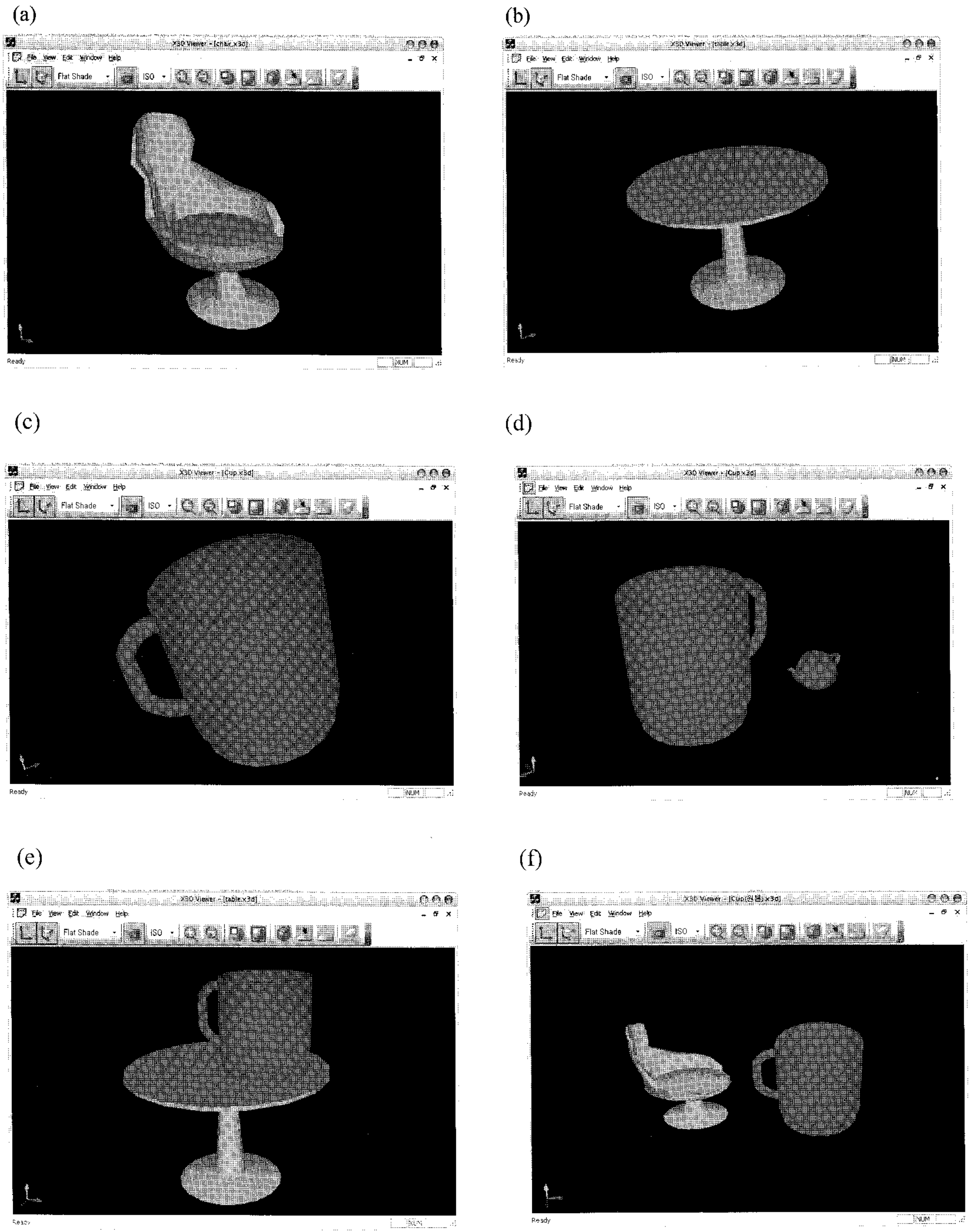
Figure 2 shows the algorithm for implementing the physical length property. Here, the height means the length of an object according to the Y-axis. We can define the length of an object in any direction.

LOLD determines the range to represent objects in the display. It is useless to represent objects together in the same scene when they have different LOLD values. However, there *are* cases where the objects can be represented together although they have different LOLD values *if* the physical length difference is below 1000 units. The numeric value of a unit depends on the resolution of the computer display. Here, we suppose that the display has a one side resolution of approximately 2^{10}.

Accordingly, it is necessary to define the real values as the numbers below 1000 within each LOLD range. In addition, two objects can be displayed together, although they are included in different LOLD levels, if the difference in their lengths is below 2^{10} millimeters.

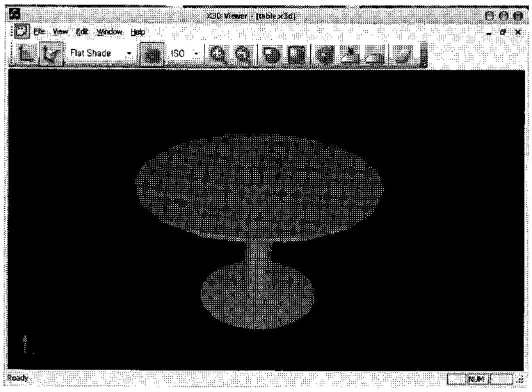
```
float bboxHeight = atof(Height Value); // User defined Height value
float LOLD = atof(LOLD Value); // User defined LOLD value
LOLD = pow(10, LOLD-3); // LOLD's default value is 3
m_pMesh->GetBounds(&pBox, TRUE); // Search the bounding box for the object
float fHeight = pBox.m_fMax[Y] - pBox.m_fMin[Y];

// Calculate the Y interval value between the maximum and minimum Y values
// of the bounding box
if (fHeight==0.0) return;
float fScale=bboxHeight/fHeight * LOLD; // Scales objects according to LOLD

// Calculate the scaling rate from the bounding box and user-defined height value
Matx4x4 XformMatrix; // Define a 4x4 scaling matrix for applying to the object
Scale3D(fScale,fScale,fScale, XformMatrix);

// Process scaling for all X,Y, and Z directions
m_pMesh->TransformVertex(XformMatrix);
// Apply the scaling matrix to all the coordinates of the object
```

Figure 2 An algorithm for physical length property with LOLD

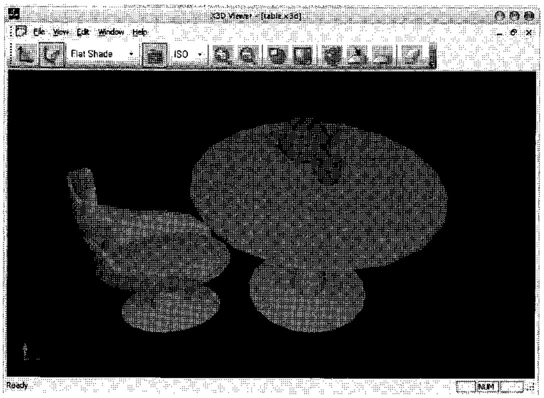


(Figure 3) Object representation without LOLD concept

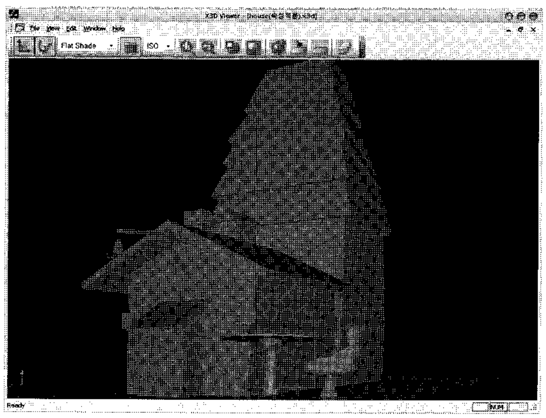
(a)



(b)



(c)



(Figure 4) Object Representation with LOLD concept

5. Experimental Results

The LOLD grammer, including physical length description, was implemented by inserting it into the general X3D specification. Figure 3 shows how virtual objects with physical length and LOLD description are represented in the display differently from general models generated by a conventional 3D modeling tool. Figures 3 (a), (b) and (c) were modeled by a conventional 3D graphics tool, 3ds MAX.

The objects have no description of LOLD and physical length. Usually, such models are generated automatically without consideration of physical length. We cannot compare the real sizes of the objects even though they were generated with the same tool. And even if they were generated by different tools, it would be difficult to discern the objects' real sizes when the objects are read and displayed together on a computer display. Figures 3 (d), (e) and (f) show that the two objects, the cup and the teapot, the table and the cup, and the chair and the cup are read into the browser together respectively without consideration of their physical lengths.

In this case, the two objects in each figure are combined using their original geometric data generated by the respective tools used to define the objects. As shown in the figure, the objects look unrealistic when displayed in relation to each other. To address this, we defined the LOLD and real length for each object in the object's X3D specification. As a result, we obtain realistic appearances and correct sizes with scaled geometries when the objects are displayed in the 3D browser, enhanced by the function of analyzing the real length properties in the LOLDs.

Figures 4 (a), (b) and (c) display several objects respectively, with the objects' real lengths defined in metric units, and scaled accordingly automatically in our browser that can analyze the physical properties and the LOLD.

6. Conclusions

In this paper, we have defined a method for inserting and implementing real length properties with metric units in a virtual environment. This enables users to compare virtual objects visually by their physical sizes according to a real-life metric system, although they were generated by different graphics systems.

We enhanced the function of X3D specification to define a new syntax for the real length properties in the Shape node and the DTD file. By implementing the physical length properties for virtual objects, we can compare and distinguish the real sizes of two objects although they were generated by different graphics tools. This capability can be used to display virtual objects in many scientific applications where it is necessary to include their precise representation in real metric units.

The physical length property is different from traditional scaling in computer graphics because the latter calculates relative size compared with an object while the former does scaling in absolute values of metric units assigned by users. We can apply the physical length properties to 3D graphics objects in many areas, including CAD, medical applications, education, interior design, and architecture - any application that requires precise metric values using standard metric units.

Future study includes the method for defining and implementing other kinds of physical properties that define an object uniquely in the world, so as to recognize objects' quantitative and qualitative properties precisely and comparatively.

References

[1] Oliver Deusen, The Elements of Nature: Interactive and Realistic Techniques, SIGGRAPH 2004 Course 31, 2004., 2003.

[2] Richard S.Wright, Jr.and Benjamin Lipchak, OpenGL SuperBible Third Edition, SAMS Publishing, 2005.

[3] Mason Woo, Jackie Neider, Tom Davis, OpenGL Programming Guide, 2nd Ed., Addison Wesley Developers Press, 1997.

[4] Donald Hearn and M. Pauline Baker, Computer Graphics with OpenGL, 3rd Ed., Prentice-Hall, Inc.

[5] Hong Hua, Leonard D. Brown, Chunyu Gao, N. Ahuja, "A New Collaborative Infrastructure:SCAPE", Proceedings of the IEEE Virtual Reality 2003 (VR'03), pp.171-179, 2003.

[6] ISO/IEC 19775:2004, "X3D Abstract", Web3D Consortium, 2005.

[7] ISO/IEC CD 19775-1, "X3D Architecture and base components", Web3D Consortium, 2006.

[8] ISO/IEC 19776:2005, "X3D encodings: XML and Classic VRML", Web3D Consortium, 2005.