Interior Designing using Virtual Reality

Someswara Siripuram[1], Atul Shukla[2], Rishikant Dubey[3], Suchet Shetty[4]
[1,2,3,4,]- Terna Engineering College,Nerul.
University Of Mumbai.

Guided By- Prof. Mrs.Varsha Wahane

Abstract: Virtual Reality is an emerging field of applied science. Virtual Reality is an immersive, interactive system based on computable information.
Virtual reality interfaces are a natural method for presenting computer-based design by merging graphics with a view of the real world. The users interactively control their local view, try out design options, and communicate design proposals. They can use virtual graphical objects that substitute for real objects which are not yet physically created or are not yet placed into the real design environment. We describe the underlying virtual reality system and in particular how it maps the virtual objects onto the real world when viewed through a head mounted display. An interior design application is used as an example to demonstrate the advantages of our approach. The aim of this paper is to implement the principles of Virtual Reality in developing Interior Design Software.

Keywords : OpenGL, Walkthrough, Interior design

I. INTRODUCTION

FEATURES OF VIRTUAL ENVIRONMENT SYSTEM

- Are easily reconfigurable by changes in software.
- Can be used to create highly unnatural environments as well as a wide variety of natural ones.
- Are highly interactive and adaptive.
- Make use of a wide variety of human sensing modalities and human sensory motor systems.
- The user becomes highly immersed in the computer synthesized environment and experiences a strong sense of presence in the artificial environment.

The main goal of the software is to provide the user with following options:
- Design the structure of the house
- Design the interiors of the house
- Select the textures of the objects
- Provide a walkthrough of the designed house

If you've ever stood in a home store, struggling to combine your interior design choices with hastily scrawled floor plan measurements, you'll have lamented the fact that there is no easy way to pick and choose home decor items that would fit perfectly in your house. Our software aims at bringing the home plans in life in a stunning 3-D. It will put you in complete control of creating blueprints that match your home exactly, allowing you to add realistic flooring, paint, wallpaper, appliances and furniture. It turns your mobile device into a virtual reality environment as it pushes the limits of the 3D engine. When you choose dollhouse and walkthrough modes, the accurate detail and accessibility of your home layout will make you feel as though you've just walked in the front door.

There is a need for an application that would assist architects, interior designers, and design hobbyists. The combination of floor planning and interior design is unmatched by any other application, and the 3D viewpoint puts it leaps and bounds above the rest.

Whether you are waiting to move into your new home and want to start decorating right now, or you are an architect trying to show your clients a real time house plan of a potential home, our software is the home design software you can rely on to get you through the front door long before you have the keys in hand.

The remainder of this paper is structured as follows: in Section II, we explain the model and the methodology used for the software development, Section III describes the environment used for generating 3-D view i.e. OpenGL, Section IV describes the selection of front end language for the software, Section V shows a brief overview of how the system would work and finally Section VI discusses future work and conclusion.

II . METHODOLOGY OF SOFTWARE DEVELOPMENT

The application software will be designed using Component Based Development Model & SDLC. SDLC is a classical development method used as requirements are known in advance. In this model if existing classes are available their properties can be inherited. If the components are not available they can be designed as per requirement. Thus this model implements one of the most important properties oObject Oriented Methodology that is Software Reusability**.**

ISBN: 978-981-08-7932-7

III .SOFTWARE ENVIRONMENT

3.1 OPENGL

OpenGL is a software interface to graphics hardware. OpenGL is a cross-platform standard for 3D rendering and 3D hardware acceleration. OpenGL is designed as a streamlined, hardware-independent interface to be implemented on many different hardware platforms. With OpenGL, you must build up your desired model from a small set of geometric primitive - points, lines, and polygons.

3.2 BASIC OPENGL FUNCTION

The Fig. 1 shown below gives an abstract, high-level block diagram of how OpenGL processes data. In the diagram, commands enter from the left and proceed through what can be thought of as a processing pipeline. Some commands specify geometric objects to be drawn, and others control how the objects are handled during the various processing stages.

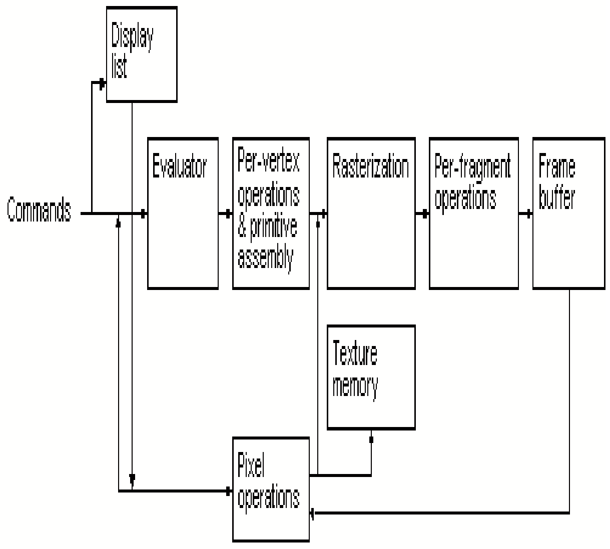

Fig. 1: OpenGL Block Diagram

The processing stages in basic OpenGL operation are as follows:

- **Display list:** Rather than having all commands proceed immediately through the pipeline, you can choose to accumulate some of them in a display list for processing later
- **Evaluator:** The evaluator stage of processing provides an efficient way to approximate curve and surface geometry by evaluating polynomial commands of input values.
- **Per-vertex operations and primitive assembly:** OpenGL processes geometric primitives' points, line segments, and polygons all of which are described by vertices. Vertices are transformed and lit, and primitives are clipped to the viewport in preparation for rasterization.
- **Rasterization:** The rasterization stage produces a series of frame-buffer addresses and associated values using a two-dimensional description of a point, line segment, or polygon. Each fragment so produced is fed into the last stage, per-fragment operations.
- **Per-fragment operations:** These are the final operations performed on the data before it's stored as pixels in the frame buffer.

.

3.3 PRIMITIVES AND COMMANDS

OpenGL draws primitives—points, line segments, or polygons—subject to several selectable modes. You can control modes independently of each other; that is, setting one mode doesn't affect whether other modes are set (although many modes may interact to determine what eventually ends up in the frame buffer). Primitives are specified, modes are set, and other OpenGL operations are described by issuing commands in the form of function calls.

Primitives are defined by a group of one or more vertices. A vertex defines a point, an endpoint of a line, or a corner of a polygon where two edges meet..Commands are always processed in the order in which they are received, although there may be an indeterminate delay before a command takes effect.

3.4 EVALUATORS

At the lowest level, graphics hardware draws points, line segments, and polygons, which are usually triangles and quadrilaterals. Smooth curves and surfaces are drawn by approximating them with large numbers of small line segments or polygons. However, many useful curves and surfaces can be described mathematically by a small number of parameters such as a few control points.

Evaluators provide a way to specify points on a curve or surface (or part of one) using only the control points. The curve or surface can then be rendered at any precision. In addition, normal vectors can be calculated for surfaces automatically. You can use the points generated by an evaluator in many ways - to draw dots where the surface would be, to draw a wire frame version of the surface, or to draw a fully lighted and shaded version.

We can use evaluators to describe any polynomial or rational polynomial splines or surfaces of any degree.

3.5 OpenGL Processing Pipeline

Many OpenGL functions are used specifically for drawing objects such as points, lines, polygons, and bitmaps.Some functions control the way that some of this drawing occurs(such as those that enable antialiasing or texturing). Other functions are specifically concerned with framebuffer manipulation.

The Fig. 2 details the OpenGL processing pipeline. For most of the pipeline, you can see three vertical arrows between the major stages. These arrows represent vertices and the two primary types of data that can be associated with vertices: color values and texture coordinates. Also note that vertices are Assembled into primitives, then into fragments, and finally pixels into frame buffer.

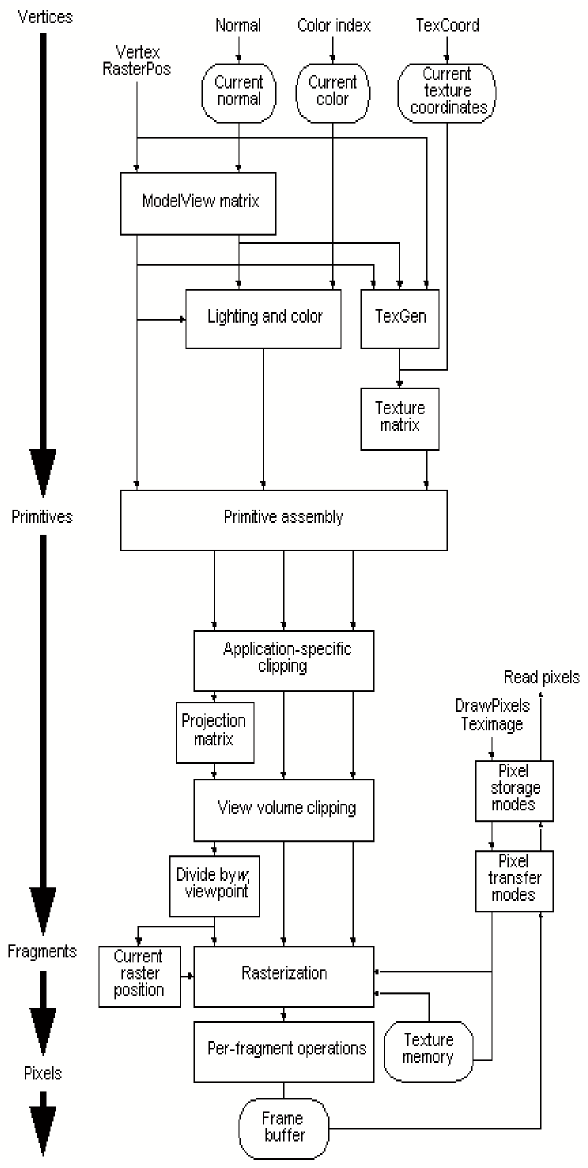

Fig. 2: OpenGL Processing Pipeline

IV. VISUAL BASIC 6.0

Visual Basic 6 is an ideal programming language for developing sophisticated professional applications for Microsoft Windows. It makes use of Graphical User Interface for creating robust and powerful applications. The Graphical User Interface as the name suggests, uses illustrations for text, which enable users to interact with an application. This feature makes it easier to comprehend things in a quicker and easier way.

Coding in GUI environment is quite a transition, traditional, linear programming methods where user is guided through a linear path of execution & is limited to a small set of operation. In a GUI environment the number of options open to the user is much greater allowing more freedom to the user & developer. Features such as easier comprehension, user-friendliness, faster application development & many other aspects such as introduction to ActiveX technology & Internet features. Make VB an interesting tool to work with.

4.1 Visual Basic Development Overview

All VB developers follow the following steps as they prepare programs destined for the user's desktop. Each of these steps described in detail in the following list:

- Design & build the user interface.
- Write code that responds to events.
- Create & call other procedures as needed.
- Test & debug.
- Convert to runtime version.
- Prepare distributable set of files.

These steps are not symmetrical. Some steps take longer than others do, & we will repeat several steps as initial application design is refined & enhanced. We will spend most of your development time in the first four steps. The VB package & deployment Wizard helps us in the last step, which normally takes less than an hour or two.

V. SYSTEM DESIGN

Initially, using the Design Toolkit, user designs the environment of his choice by specifying the characteristics, relative locations of various objects that are of his interest in the room. User is provided with a 3-D user interface that allows him to Drag-and-Texture objects like tables, sofas, etc. that are of his interest. He can also specify the shading parameters.

The 3D Modeling Software makes use of this information to generate the 3D image that is superimposed on the real-world image. This generated image is passed to user.

5.1 SYSTEM FLOWCHART

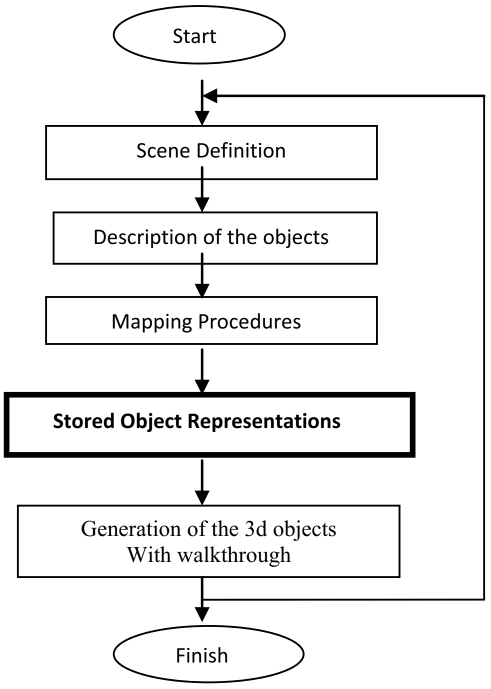
Start
Scene Definition
Description of the objects
Mapping Procedures
Stored Object Representations
Generation of the 3d objects
With walkthrough
Finish

Fig. 3 : Flowchart

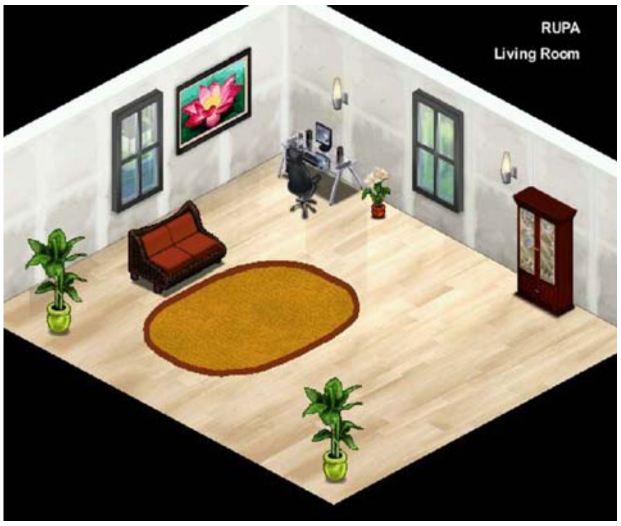

Fig. 4: 3D VIEW-The expected overview

VI. DISCUSSION AND CONCLUSION

6.1 CONCLUSION

The Literature survey conducted in order to gather information for this paper has helped us gain an insight into the challenges faced by the developers for successful implementation of any project in a new and developing field. Due to various advantages of Virtual Reality over other Graphics, this paper would have significance importance in the field of Architecture and Interior Designing.

This paper is an attempt to produce a platform for designers to explore the tremendous world of virtual reality and provide the clients with realistic view of the real world.

6.2 FUTURE DEVELOPMENT

- This paper can be implemented for Augmented reality by use of HEAD MOUNT DISPLAY and completely eliminate the need of a computer to monitor the user's movements in the room and also do away with the **Lag** which is present in the current system due to the use of keyboard and the mouse.
- Currently the software does not attempts to include objects needed to build a full-fledged house. This can be improved upon by providing the user with more objects.
- The software is currently meant to work only for a flat. This can be updated in future for a multistoried complex structure.

VII .REFERENCES

1. Azuma, R. (1997), A survey of Augmented Reality, Presence,.
2. Billinghurst, M. and Kato, H. and Proupyrev, I. (2001), The MagicBook-Moving Seamlessly between Reality and Virtual,
IEEE Computer Graphics and Application, Vol.21, No.3,
3. Billinghurst, M. and Proupyrev, I. and Kato, H. and May, R. (2000), Mixing Realities in Shared Space:
An Augmented Reality Interface for Collaborative Computer, Proceeding of ICME 2000.
4. Dias, J. M.S. and Santos, P. and Nande, P. (2003), InYour Hang Computing: Tangible Interfaces for Mixed Reality, Proceeding of 2nd IEEE International Augmented Reality Toolkit Workshop, WasedaUniv, Tokyo, Japan.
5. Kensek, K. and Noble, D. and Schiler, M. and Tripathi, A. (2000), Augmented Reality: An application for architecture, Proceeding of 8th International Conference on Computing in Civil and Building Engineering, ASCE, Stanford, CA, 294-301.
6. Milgram, P. and Takemura, H. and Utsumi, A. and Kishimo,
F. (1994), Augmented Reality: A class of displays on the reality-virtual continuum, Proceeding of Telemanipulator and Telepresence
Technologies: 2351-34, Retrieve 2007-03-15.
7. Pasman, W. and Woodward, C. (2003), Implementation of an Augmented Reality System on a PDA,
Symposium of Mixed and Augmented Reality, ISAR 2003, Tokyo, Japan.