REAL TIME RENDERING

1RUTUJA JOSHI, 2NEHA NANDE, 3PARESH KHALATE, 4AKSHAY YEWALE

Department of Computer Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune
Email: mybeproject14@gmail.com

Abstract—Rendering is an important arena in the domain of the Computer vision aims in communication with the user through output screen. “Real Time Rendering” is an inquisitive field of technology which is an event driven architecture that deals with creation of virtual environment. This paper gives a gist of the advancements in the “Real Time Rendering Techniques” along with a proposed system for the futuristic approach. We present literature survey for past approaches, along with advancements using some generic examples that are found commonly in various spectrums like simulations, gaming, advanced softwares, etc. We have provided with the some generic examples that are common with the various scenarios to describe about the improvements in the Real Time Rendering. We also provide refinement in the current technology along with description of approach to overcome their drawbacks. The proposed system so formulated is a concept of creation of special effects library that include certain generic effects with greater realism (in attributes of detail, rendering speed, frame size, etc.) Here, the system is specified to a certain theme, but can be extended to develop a complete scenarios. Thus it paves a platform to develop various other aspects of the virtual environment.

Keywords— OpenGL, Graphic Processing Unit, Real Time Rendering

I. INTRODUCTION

Computer Vision is an important field that directly affects the human perseverance of the computer world. Now, it is completely indulged in the interaction with the human through physical or virtual medium. Physical Medium can be sound, papers or effectors that are directly in contact with the real world. However, Virtual Medium is more versatile and cannot be standardized on a piece of paper. Further, Rendering is an important aspect of the Virtual Medium that aims in communication with the user. Rendering is the process of generating an image from a model or models in what collectively could be a scene file, by means of computer programs. Offline Rendering and Real Time Rendering have helped to improve the virtual environment in terms of realism, details, interaction, understanding etc. Realtime Rendering includes creating synthetic images fast enough, for the viewer to interact with a virtual environment. Real Time Rendering has improved user experience in the fields of Gaming, real life situation simulations and certain advanced softwares leading to better clarity, realism, interaction, and experience. Thus advancements in the Real Time rendering emphasis on stepping towards realism in the virtual environment that is created. In the recent year’s, real-time rendered effects have an edge over normal visual effects due to the advance in film industry as well as the domain of Computer Vision. The industry has rapidly emerged in recent decade due to a wide range of applications in films, games, tutorials, etc. Various effect development softwares are available in the market like 3D MAX, MAYA and Final Cut X. But in case of Real Time Rendering, these effects need to be created on the go. Now OpenGL provides this functionality under one library and using this, complicated realistic scenes can be rendered. Some of the effects include Generating and Rendering Procedural Clouds in Real Time on Programmable 3D Graphics Hardware, Fur Effects which can be used for Teddy, Cats and Hair and convenient simulation of snowflakes based on OpenGL. Thus, stimulation of the ‘effect’ as an output of a running program is developed using OpenGL. The main objective is that a program could serve as a collection of special effects enhanced to best possible realistic view and which could be used in application development.

II. LITERATURE SURVEY

The primary aim of Real Time Rendering is creation of interactive 3D world with an immense, believable environment for the user to navigate through. In the recent years, Real Time Rendering arena has uplifted itself throughskies, due to increased use in Movies, Gaming, and Simulations.

A. Movies
Movies, Feature Films have evolved and thus more and more use of special effects are a part of them. ‘Life of Pi’, Ang Lee film, is a...
brilliant example of the kind. Richard Parker, the Tiger, was an output of the code that included various different special effects.

B. Gaming Today, a multi-billion dollar industry ‘GAMES’ wholly concentrates on the user experience of the games so created. We have evolved from the traditional games and moved up to certain games that are full of storyline, special effects, stunts, and decision making capacities. This evolution is ongoing and every day we encounter new advancements in gaming technologies. Games like Battlefield3, Crysis 2 are examples of games that determine the amount of the evolution that has taken place.

C. Real Life Situation Simulations This is one of the most important field wherein a complete virtual environment is been created for the user to navigate. Consider Flight Simulations. Here the pilot can be trained to various condition that are subjected on field, without any monetary loss. Certain hazardous, untoward situations can be created without any man loss or monetary loss.

D. Advanced Softwares Certain softwares need to deliver a perfect and clear view to the user. For instance, creation of the interiors of the flat can be rendered on the output screen and can be navigated across various colour range, sizes, designs, arrangements etc. It is important in this softwares that the content shown is closer to the reality, and provide a complete and perfect view of the virtual environment so created.

III. OPEN GRAPHIC LIBRARY: A EFFICIENT API

OpenGL is a software interface to graphics hardware. It is designed as a streamlined, hardware-independent interface to be implemented on many different hardware platforms, consisting of more than 700 distinct commands (about 670 specified in OpenGL Version 3.0 and 50 in OpenGL Utility Library). The commands are used to specify object and operations required to produce three-dimensional application. OpenGL does not provide any commands for performing windowing tasks or obtaining user input, it is done by system controls of the respective hardware used. The commands provided, allow specification of relatively complex figures such as parts of body, air-planes, molecules etc. With the help of OpenGL, the desired model is built, with small set of primitive geometric figures like points, lines, polynomials etc. The OpenGL Utility Library (GLU) provides advanced effects like quadric surfaces and NURBS curves and surfaces. OpenGL is a multi-platform application programming interface (API) majorly used for interacting with the Graphics Processing Unit (GPU). It is created as an open alternative to Iris GL, the proprietary graphics API on Silicon Graphics workstations. OpenGL 1.0 authored by Mark Segal and Kurt Akeley formalized definition of using graphics API. Texture object was a notable omission in version 1.0 form the API. Revision in OpenGL, were significant in every version like OpenGL 1.1 included glBindTexture extension to core API. OpenGL Shading Language (GLSL) was incorporated in OpenGL 2.0 which enabled programming of transformation and fragment shading stages of the pipeline. OpenGL 3.0, the next version released removed the deprecated concepts, GL 3.1 removed most deprecated features while GL 3.2 created notion of core and compatibility OpenGL contexts. OpenGL 4.1 enables query and load a binary blob for program objects. It also has an ability to bind programs individually to programmable stages. It has multiple view ports for same rendering surface or one per surface. OpenGL 4.2 allows shaders to read and write images with few restrictions. It also allows unpacking 16-bit floats from a 32-bit unsigned integer value and use of certain advanced compression formats. OpenGL 4.3 includes key features like debug messaging and GLSL multi-dimensional arrays. Arbitrary image coping, arbitrary compute shaders along with texture, buffer object and frame buffer invalidation are remarkable features included in the version. The current version in use is OpenGL 4.4 released in 2013 has extended its features; it consists of a fixed storage for buffer object. It also enables direct clearing of a texture image. Locations can be set on input/output interface blocks for packing purpose. Texture can be made by use of stencil-only formats and 8-bit stencil is required format. Some standard books provided by the officials regarding the concepts and advancements in OpenGL include 1) ‘OpenGL Superbible Fourth Edition’ also known as ‘OpenGL Blue Book’. 2) OpenGL Programming Guide Seventh Edition, Dave Shreiner, The Khoronos OpenGL ARB Working Group.

IV. A GENERIC EXAMPLE

Fig. 2 Example of an image for advanced software to show use in building construction
Various scenarios are to be painted while implementation of a virtual environment for any game or even simulations. Now different kinds of scenes may be rendered such as terrains, roads, valleys, houses, walls, sky or atmosphere, fogs etc. but amongst them the hardest and so called the renderer’s enemy is commonly referred to a Water Body. Water body is hardest to render due to many reasons. This may include immense amount of physics involved in the process as simple as reflection or refraction of light. Also various rules of fluid dynamics, motions, light, and flow need to be considered while rendering a simple water body. Further we need to consider the thermodynamics as the factor of heat affects the water state. Consider a scenario in a first person action game where the person is wandering among the ice laded water body. Here all the above factors will affect the scene to be rendered in order to make the water body look authentic and more realistic.

Darwin Peachey presented a variation on this approach using basis shapes other than sinusoids. The later approach taken to create surface wave was close to the one outlined by Masten, Watterberg, and Mareda, This approach synthesizes a patch of ocean waves from a Fast Fourier Transform (FFT) prescription, with user-controllable size and resolution, and which can be tiled seamlessly over a larger domain. The patch contains many octaves of sinusoidal waves that all add up at each point to produce the synthesized height. The mixture of sinusoidal amplitudes and phases however, comes from statistical, empirically-based models of the ocean.

V. REFINEMENT PROVIDED BY PROPOSED SYSTEM

Objective of the proposed system is to develop a library of special effects which will be used to create a snippet. The effect included in library will be used frequently by programmer during development of real-time rendering applications. For each application effects are created separately and customised accordingly when required. But in proposed system effects can be imported into application and parameterised, so the programmer’s efforts will be reduced. The proposed library will allow a developer to create templates of some basic effects like water, cloud, etc. And user or developer making use of this library will import these Templates and customize them according to user’s requirements and application. Consider the generic scenario of rendering the water body. We have seen gist of all the methodologies so used in the Process. Recently NVIDIA PhysX’s position based fluids research is considered to be the best algorithm to render a water body. In our proposed system we will design and implement various effects based on the water body using the position based
fluid algorithms. Various other effects that are included in the process can be light source, wind, transparency, shallow and depth of the water body. We attempt to create the maximum details in the process in order that the rendered virtual environment so created be completely realistic. We will also try to implement various additional effects in other to create a new effect although for the desired library. These new effects will be a part of the desired library and can directly be used for the further programming purposes.

CONCLUSIONS

Thus this paper gives a detailed survey about the literature present in the field of Real Time Rendering along with the concepts and techniques used in OpenGL. We also have documented research conducted on our proposed system. We thereby have concluded that our proposed system has an edge over the other developing software's and animation software. This is an attempt made to reduce coding time and changes the output as per requirement and compatibility. The paper hereby enlightens us about the various fields and spheres of Real Time Rendering.

ACKNOWLEDGMENT

We sincerely acknowledge Mrs Geeta Navale, Head of Department, SITS, for her valuable support. We also express our sincere gratitude to Prof. Mr Prashant S. Dhotre, Project Guide, for his worthy guidance in the process also, to Mr. Yogeshwar Shukla, Eaglet, Pune for his advising at every step of the Project.

REFERENCES

[PDF], Available: Internet