

# Exploring Genomes with a Game Engine

Jeremiah J. Shepherd  
University of South Carolina  
315 Main St.  
Columbia, SC, 29208  
shephejj@email.sc.edu

Bill Arndt  
University of South Carolina  
315 Main St.  
Columbia, SC, 29208  
arndt@email.sc.edu

Jim Zheng  
MUSC  
135 Cannon Street, Suite 303  
Charleston, SC 29425  
zhengw@musc.edu

Yun Zhu  
University of South Carolina  
315 Main St.  
Columbia, SC 29208  
zhu5@mailbox.sc.edu

Jijun Tang  
University of South Carolina  
315 Main St.  
Columbia, SC 29208  
jtang@cec.sc.edu

## ABSTRACT

Studying genomes continues to be fruitful for prognosticating and diagnosing many genetic disorders and diseases. Most of these studies have used systems that view the DNA in a linear structure. However having this information is only a small part of fully understanding genomes. Visualizing genomes in real time 3D can give researchers better understanding, but the genome contains a lot of information that cannot be processed at once. By using a game engine and sophisticated video game visualization techniques, we were able to construct a multi-platform real-time 3D genome viewer.

## Categories and Subject Descriptors

H.5.2 [User Interfaces];

## Keywords

Genome, Game Engine, Octree

## 1. INTRODUCTION

Researchers continue to explore various genomes using diverse tools and techniques. Most software simply looks at the DNA sequence. This, however, is only a small part of the bigger picture. There exists much more information, such as nucleosome position, histone methylation, transcription factory, etc., that could lead to new discoveries. We believe through visualizing genomes in 3D will provide more insight than just viewing the sequence.

In order for this to be effective, a system must be designed to visualize the genome, but also allow for fluid interactivity. However, creating such a system is not a trivial task, since this would visualize any number of genomic models. Furthermore, the data that is used to represent this infor-

mation is big, so being able to load the entire model into memory is infeasible.

However, the answer to this problem lies in video games. Most modern video games have to visualize a large amount of information, while also being interactive. More often than not, game developers make games using game engines, which have been effective in simulation research [8]. Along with simulations, biology is no stranger to having games assist in new discoveries. Games like 'Fold It' have been used to solve difficult protein folding problems[6]. However, using a game engine and game visualization techniques in this domain has never been attempted. We show in this paper how the use of a game engine along with game visualizing techniques were used to create a real time 3D visualization of genomes.

## 2. RELATED WORK

Biological researchers often use a genome as a scaffold to analyze and disseminate their data, which requires new data to be merged into the existing framework. To address this challenge, the University of California Santa Cruz team developed a genome browser that not only integrates genome information, but also allows users to integrate their own data [5]. Such genome browsers have become very popular and played essential roles in several important, large-scale genome projects[1][9]. On the other hand, all these genome browsers are sequence based and visualize genomes in 2D space based on sequence coordinates. This type of visualization is limited in displaying some important epigenomic and structural information generated from recent technologies.

"Genome3D" was the first to facilitate integration and visualization of large genomic in three dimensions [2]. This framework enabled us to infer new knowledge about structure/function of genomes that would have been difficult to accomplish by sequence-based browsers. For example, phosphate groups of different base pairs in DNA strand can be either exposed to the outside or sandwiched in between the histone proteins and DNA backbone. This information has important implications as exposed phosphate groups can be easily accessible by DNA binding proteins [3]. While new sequencing technologies allow researchers to map individual nucleosomes across the whole genome [7], it is a significant challenge for current genome browsers to capture such important epigenomic information. Visualizing and inferring

such information from 3D genome model may provide additional evidence. However even with this system it could not handle large amounts of information, nor could the user explore this information in real time. We solve these problems by using a game engine and some sophisticated visualization techniques.

### 3. THE SYSTEM

The greatest problem facing this type of system was how to managed the large amount of information and having interactivity in real time. Fortunately, game engines, as opposed to other visual tools, allows real time user interaction coupled with sophisticated visualization techniques. Even with this in place, rendering this great amount of data is a challenge. One visualization features is the level of detail (LOD) system, and it is used to cut down the amount of processing power needed to display a scene. Different parts of the scene are presented at different LOD's generally based on a criteria like distance. An object further away from the scene's camera can be rendered at a lower resolution without much notice from the viewer. By doing this the system is not required to process as much information and can focus on objects closer to the camera. To best address the problem of rendering the genome we used some special data structures commonly used in games, and also processing different LOD's in a specialized manner.

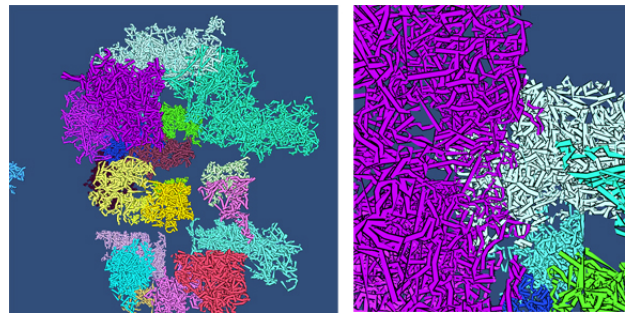
#### 3.1 The Octree

To determine at which LOD to show we first group together the information into a data structure called an octree. This structure subdivides 3D space into eight sections which represent nodes in the tree, which can be further subdivided making more nodes. Instead of using a static octree we incorporated a dynamic version[4] to reduce space and processing speed. This version of the data structure automatically subdivides into eight parts when the data contained in said node exceeds a given tolerance. In this way, we avoided having an unnecessarily big tree(s) in memory.

Preserving the structure of the sequence in the octree was key, so finding a way to split up the ordered information was a challenge. Instead of a mesh that had to be divided, the model is generated given a series of files that contain control points of a curve. We dissect the model it by its control points, so that whenever the curve intersects an octree node boundary a new control point on the curve is constructed and placed into both nodes. While this increases memory usage minimally, it preserves the content.

#### 3.2 Further Levels of Detail

Along with the octree, the information given is grouped into three different levels: fiber, nucleosome, and atomic. The fiber level represents the lowest level of detail where each chromosome's 30nm fiber is displayed as seen in Figure 1. Diving deeper, the nucleosome stage holds the next highest level of detail. At this level, nucleosome base-pair positions and the DNA that wraps around and links to them are displayed. In most cases the linking DNA is procedurally generated, but the system also takes into account when that linker data is known. That information, in these special cases, is loaded a displayed. Finally, the highest level of detail is at the atomic scale. From the information provided at the nucleosome level the atomic level is procedurally generated. In order to transition from one level to another,



**Figure 1: Human Genome Screen Shot.** The left shows the human genome at the 30nm scale where each chromosome is color coded. The right shows a zoomed in version on one particular chromosome

the user presses buttons that represent the levels of detail, and are able to fly around the different biological structures freely. Furthermore at each level, researchers can also view, edit, and create annotation on the structure.

### 4. CONCLUSION

In this paper we presented the problem of visualizing genomes in 3D, and the potential benefits that this could provide. We solve this problem by applying techniques and tools that are commonly used in video games. These include using an octree and other forms of LOD analysis. With the implementation of the system we have provided researchers with a tool that could be used for more genomic work.

### 5. REFERENCES

- [1] The encode (encyclopedia of dna elements) project. *Science*, 306:636–640.
- [2] T. Asbury, M. Mitman, J. Tang, and W. Zheng. Genome3d: a viewer-model framework for integrating and visualizing multi-scale epigenomic information within a three-dimensional genome. *BMC Bioinformatics*, 11:444.
- [3] K. Einfeld, R. Candau, M. Truss, and M. Beato. Binding of nfl to the mmtv promoter in nucleosomes: influence of rotational phasing, translational positioning and histone h1. *Nucleic Acids Res*, 25:3733–3742, 1997.
- [4] D. Ginsburg. Octree construction. *Game Programming Gems*, pages 439–443, August 2000.
- [5] W. Kent, C. Sugnet, T. Furey, K. Roskin, and T. Pringle. The human genome browser. *Genome Res* 12, 12:996–1006, November 2002.
- [6] F. Khatib, F. Dimaio, S. Cooper, M. Kazmierczyk, M. Gilski, and et al. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature Structural Molecular Biology*, 18(10):1175, November 2011.
- [7] D. Schones, K. Cui, T. Roh, and A. Barski. Dynamic regulation of nucleosome positioning in the human genome. *Cell*, 132:887–898, 2008.
- [8] J. Shepherd, R. Dougal, and J. Tang. Visualization and simulation potential of microsoft's xna. *Grand Challenge in Simulation and Modeling (GCSM)*, pages 306–310, July 2010.
- [9] N. Siva. 1000 genomes project. *Nat Biotechnol*, 26:256.