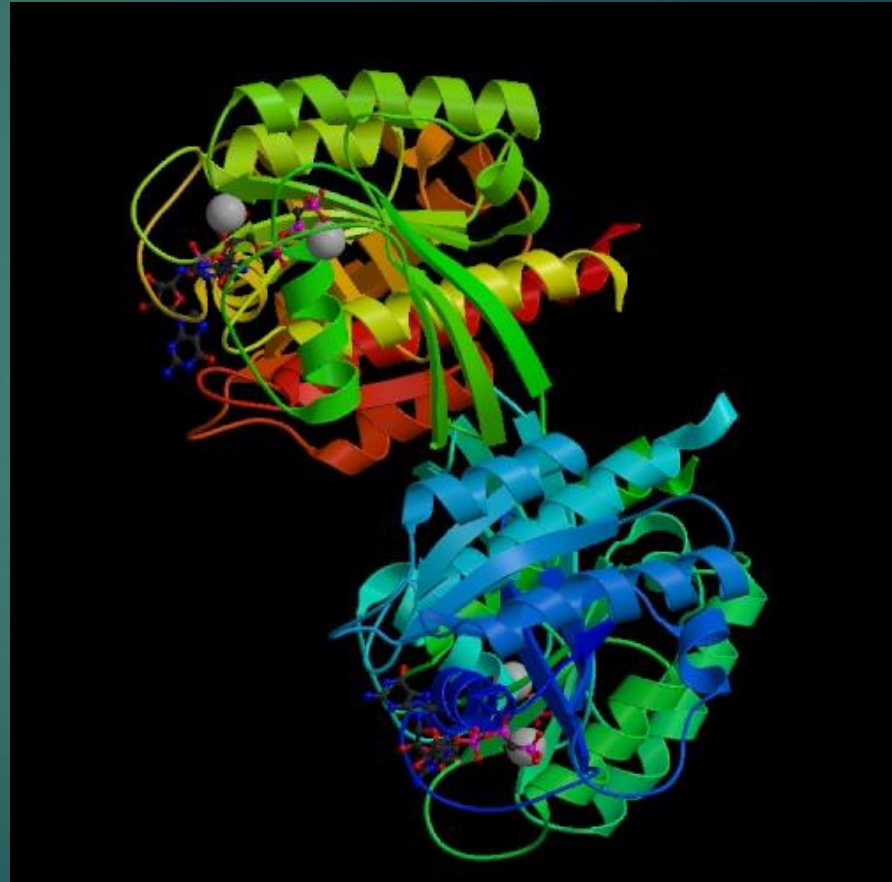
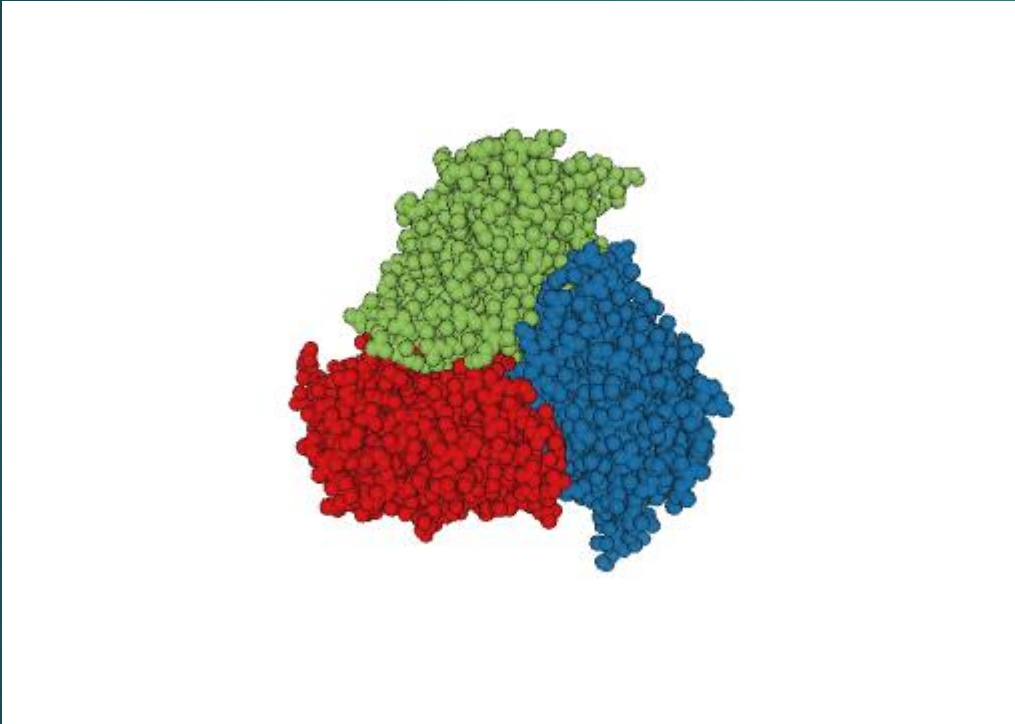


XLPM Map

A PROTEIN-PROTEIN INTERACTION 3D MODEL VIEWER

Proteins



Protein Data Bank

- ▶ Protein data bank is a database for all known existing proteins
- ▶ PDB is a source of information for protein sequencing, modeling and mapping
- ▶ You can download data files for any protein: PDB files, Fasta Files, Mapping files

What is XLPM?

- ▶ XLPM stands for X-Linked Peptide Mapping Algorithm
- ▶ Its primary goal is to analyze the data produced by cross linking two proteins
- ▶ It Aims at analyzing the interaction between two proteins

Previous work

- ▶ The XLPM visualization is part of the XLPM project at the Department of Bioinformatics
- ▶ Previously I worked on visualizing XLPM data files in 2D graphs using WebGL and D3 SVG
- ▶ I took the data files for several analyses of proteins and started thinking about ways to view these data both in 2D as a graph like visual and 3D as on actual protein structures highlighting the interacting atoms

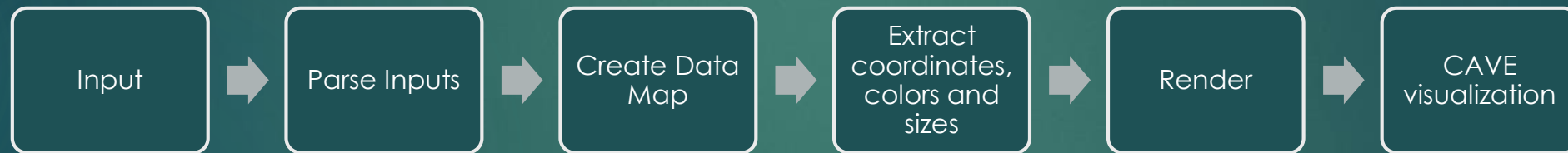
XLPM Map Viewer

- ▶ <http://www.i3akef.net/xlpm>
- ▶ Inputs to this system are FASTA files and XLPM Data files
- ▶ The visualization is a representation of the data file on three levels including the atomic level
- ▶ The data file here is the same I am using in this project

The aim of the project

- ▶ The aim of this project is to make it easy for biologists or any body who is interested in studying the interaction between 2 proteins to understand the analysis by just visualizing 2D and 3D data
- ▶ The research is part of a bigger effort to enhance the development and discovery of drugs for several organisms
- ▶ Primary users for these applications are biologists

Methodology



Methodolgy

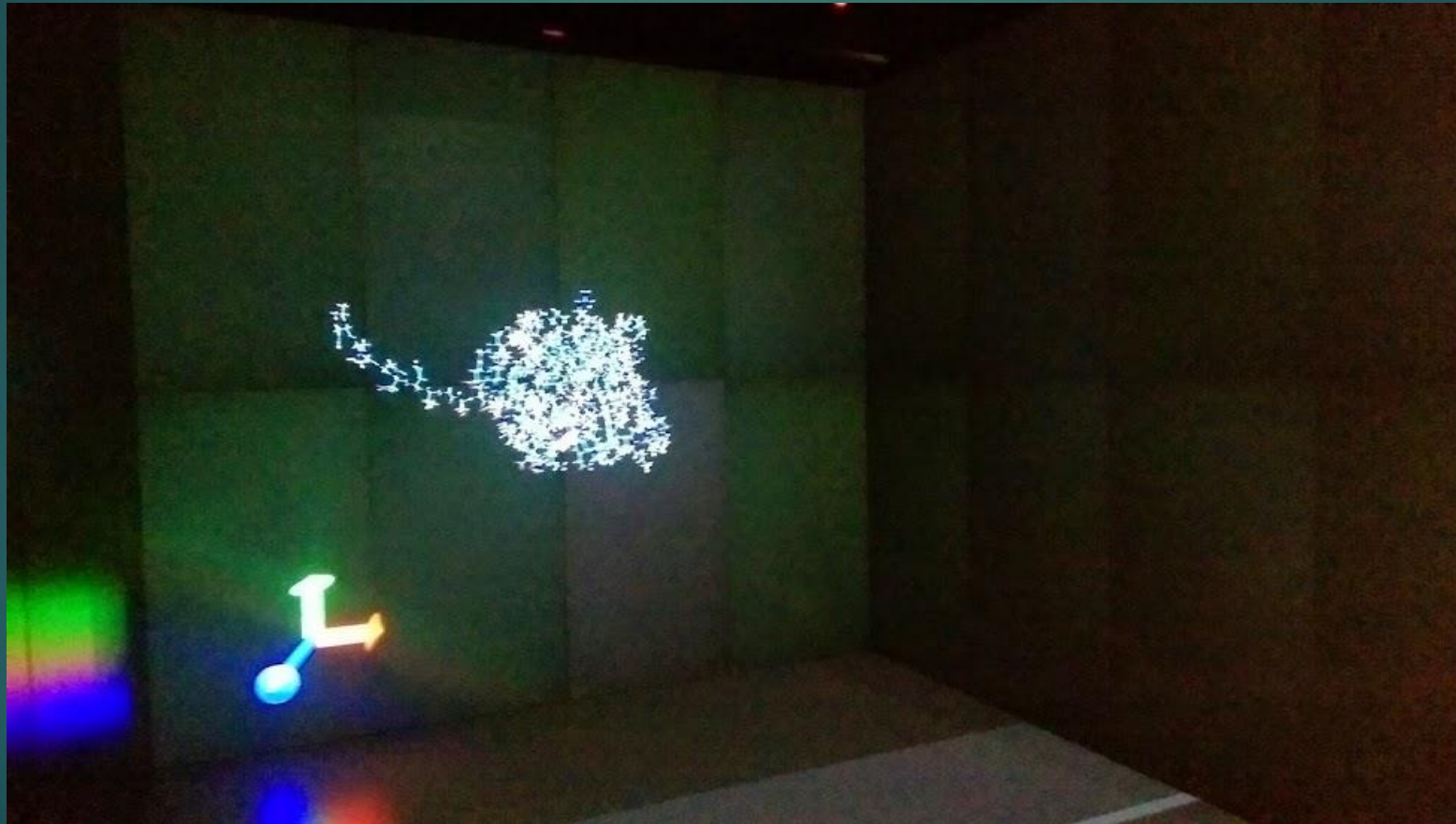
- ▶ Understanding PDB files and FASTA files is so important to be able to parse it
- ▶ PDB files, FASTA files, Mapping file ,XLPM Data files
- ▶ Parsing the PDB files and extracting coordinates, colors and sizes of Atoms
- ▶ Creating a map of interacting atoms in the two proteins by analyzing the PDB -> FASTA -> MAP
- ▶ Recoloring the interacting atoms
- ▶ Create Vertex Array
- ▶ Draw the vertex array

Rendering

- ▶ Drawing the vertex array as point sprites
- ▶ Applying a vertex and fragment shaders to draw the spheres with their colors and position
- ▶ Use clamp and diffuse lighting inside the shader to give good effects
- ▶ I pass the Model View matrix and the Projection Matrix to the shaders
- ▶ I use GLSL specification 330

Virtual Reality Environments (CAVE)

11



Virtual Reality Environments (CAVE)

12

- ▶ What virtual reality environments really means is two things:
Clustering and Tracking
- ▶ So VR libraries for CAVE like environments basically manages running the code on a cluster and provides input tracking for devices like joy sticks
- ▶ The main idea behind these libraries is that they basically takes over the projection and model view matrix so this makes it impossible for GLSL 330 with generic uniforms to work with these libraries
- ▶ I am still researching this part
- ▶ However I was able to show one molecule in the cave

Problems and Challenges

- ▶ The main problem when I started drawing the atoms using normal `glutSolidSpheres` was performance issues → really slow
- ▶ After some research I had some choices like: using geometry shaders or drawing point sprites then writing a fragment shader that would draw a sphere around the point
- ▶ Debugging the shader so I used recompiling
- ▶ Loading the vertex array is very slow
- ▶ Loading the program in the cave is very challenging because the library uses OpenGL 2.2 with 120 shader specification
- ▶ Point sprites do not automatically position themselves relative to the camera it had to be done in the shader

Future Work

- ▶ Put the program completely on the cave
- ▶ Using leap motion to handle the molecules
- ▶ Enhancing XLPM data visualization to appear on different levels
- ▶ Adding several molecular rendering options like Sticks and Ribbons
- ▶ Adding support for android platforms using OpenGL ES

Demo

Source code

- ▶ Source code available as a Git repository on Github
- ▶ <https://github.com/iebeid/xlpmol.git>

Refernces

- ▶ Marco Tarini, Paolo Cignoni, Claudio Montani:
Ambient Occlusion and Edge Cueing for Enhancing Real Time Molecular Visualization
IEEE Transactions on Visualization and Computer Graphics Volume 12 , Issue 5 , Pages 1237-1244 , **2006** , ISSN:1077-2626
- ▶ Video Production. (n.d.). Retrieved April 3, 2015, from <http://sol.gfxile.net/>

End