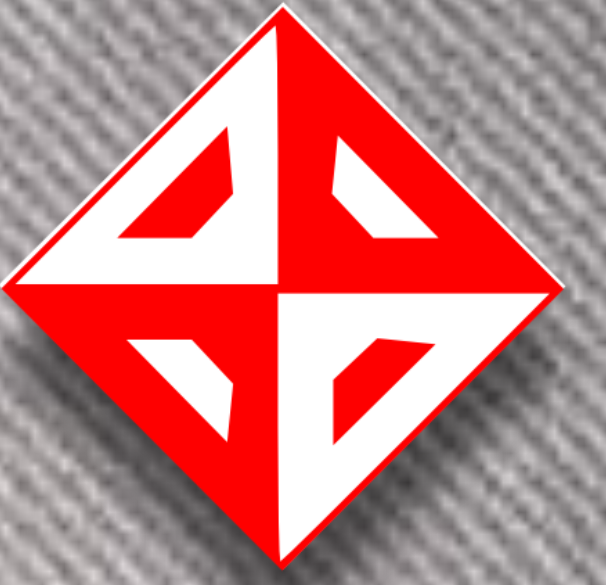




CEREBRA

a project of Simple Labs



SCOPE

The aim of this project is visualizing the fMRI data on a 3D graph to increase the understandability of the complex data. The fMRI data includes the brain response of a human in response to some particular circumstances as showing the picture of a red apple.

The main role of the final 3D graph will be visualizing voxels and edges which show the related voxels. Besides graph may include a brain image as background and five main lobes of the brain, which are frontal, parietal, occipital, limbic, temporal lobes, in different colors.

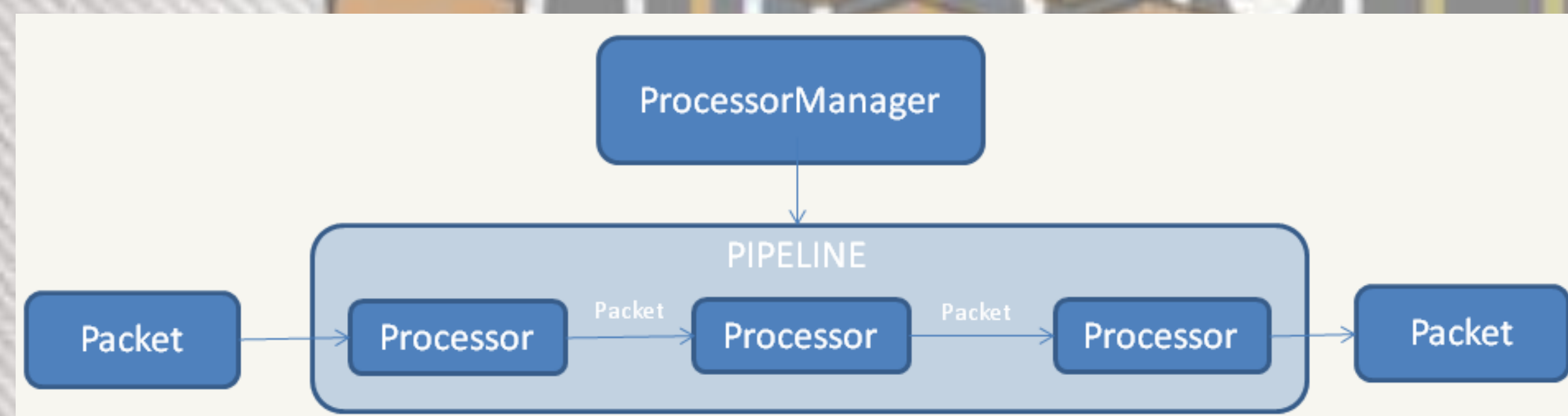
Since the fMRI data is very large and complex, time and space will be main constraints.

The target audience of this project is mostly academicians and medical institutes. Cognitive state representation and visualization of human brain is fundamentally important in neuroanatomy, neurodevelopment, cognitive neuroscience and neuropsychology.

This project will be implemented in Unity3D Game Engine with using its OpenGL libraries. C# will be used as the programming language.

IMPLEMENTATION

Extensibility is a must for the project. Since this project will be used in a highly active research area, it is essential that novel ideas be implemented easily. Project team aims to achieve this with a highly algorithm and data agnostic approach.



PACKET CLASS

This class is used as an immediate data format between two Processors. Packet class encapsulates all data needed by Processors: voxel coordinates, edge values, etc... It also offers a way to pass named extra data between Processors.

PROCESSOR INTERFACE

This interface defines outlines of Processors and how Processors should be implemented.

PIPELINE CLASS

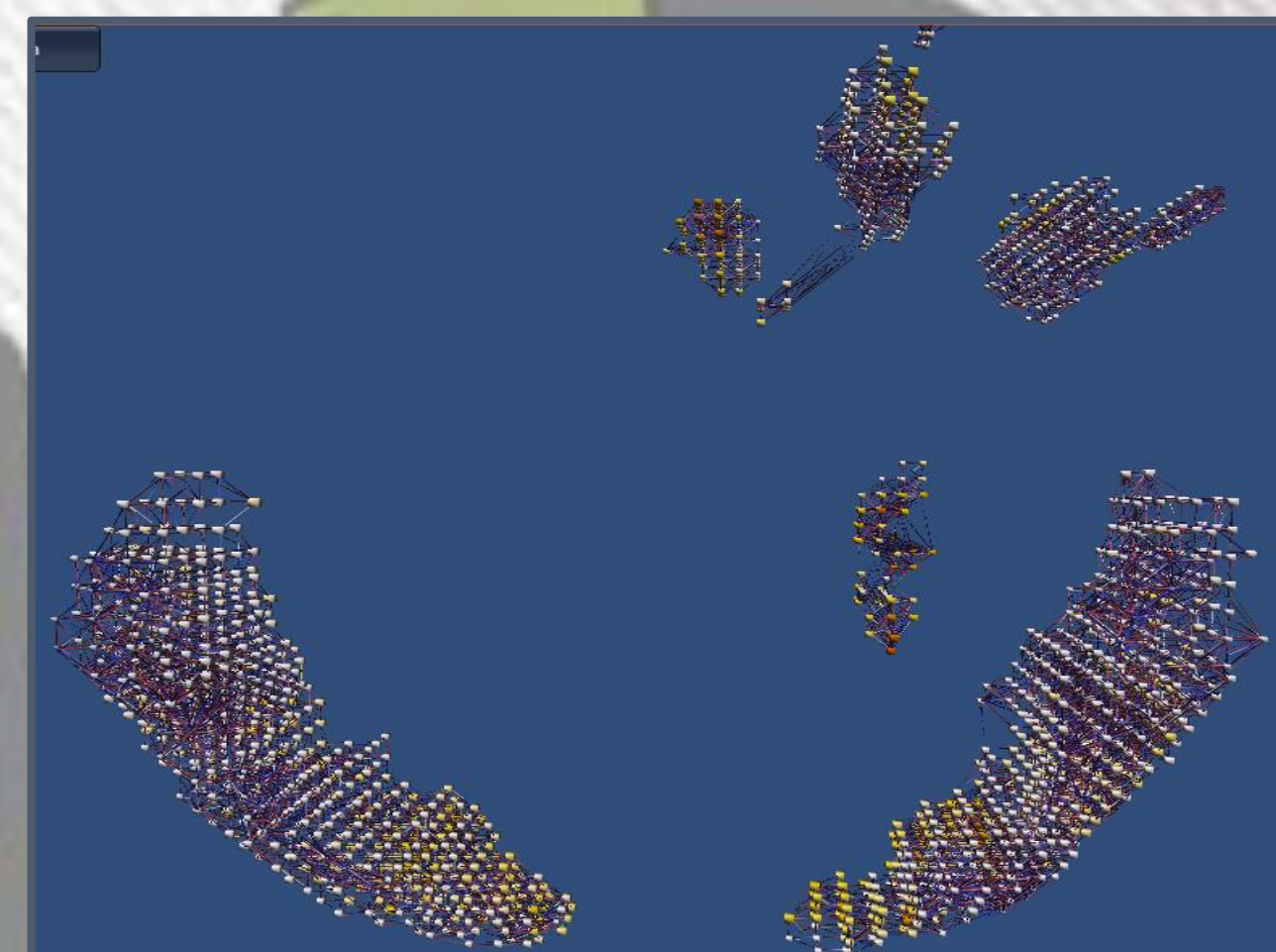
This class is responsible for chaining Processor operations. A pipeline is an object that the user can save to or load from it a file. Thus, it also enables the user to create his/her own presets. When a Processor is added to the Pipeline, Pipeline object checks whether it is the first Processor to be added, and if it is, is it an input type Processor.

PROCESSOR MANAGER CLASS

This class is responsible for managing Processor selection and generation. This is a static class and it's members are all static. Each Processor must register itself with the ProcessorManager. C++ doesn't allow static constructors, which would be used when registering. This is a problem the team is working on.

FINAL PRODUCT

Here are some screenshots of our program.



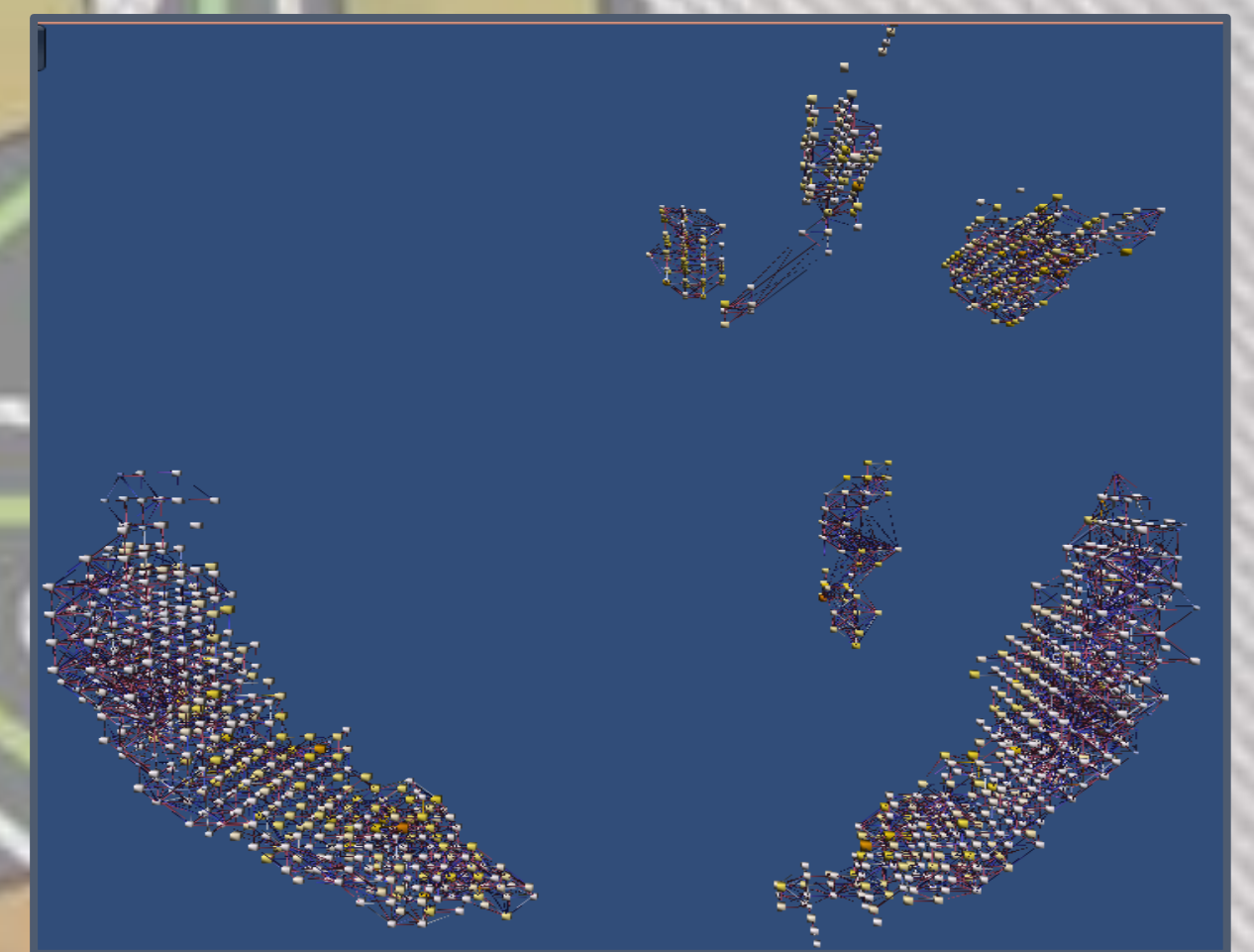
This image is the pure representation of brain data. There is no filtering applied. It is clear that the image is too complex and hard to analyze.

Cubes represent voxels.

Lines represent arcs. These are relationships between voxels.

You may also adjust the layer depth. You may want to do this because it is hard to see the inner parts of the brain. By adjusting layer depth you can easily see the inner parts.

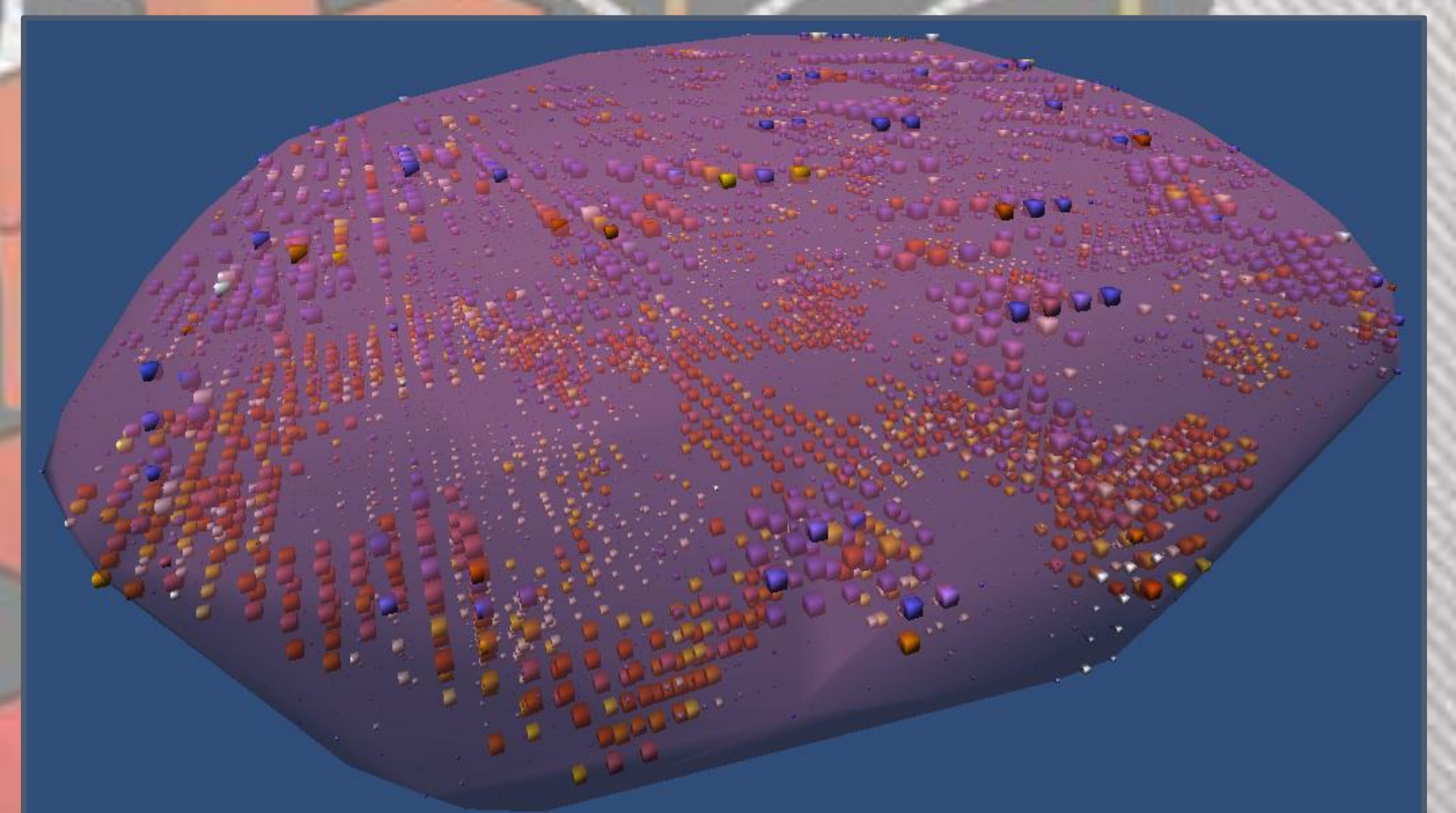
All you have to do is to slide the layer depth slider on the filters panel.



You can apply filters and simplify the image. On the right, image has been simplified by finding the maximum spanning tree.

You can even develop your own filter!

You can also add a brain shell according to your data like below.



ACKNOWLEDGEMENTS

We thank to Prof. Fatoş Yarman Tunay Vural and her colleagues at the Image Processing Laboratory for their help throughout the project.