



**Computer Graphics (CSE 528)**  
**Project Report on**  
**Mesh Segmentation and Labeling:**  
**A Data Driven Approach**

By: Rohit Rao P

SBU ID: 111447392

Guided By: Professor Hong Qin.

## Contents

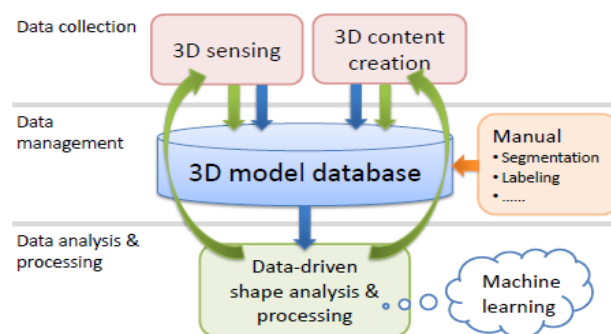
Introduction:.....	2
I) Mesh Features: .....	3
I.1) Geodesic distance: .....	3
I.2) Shape diameter function: .....	4
I.3) Curvature: .....	5
I.4) Volumetric shape images: .....	6
I.5) Shape Context: .....	7
II) Implementation: .....	8
II.1) User Interface using keyboard and mouse. ....	8
II.2) Mesh Feature Computation. ....	8
II.3) Mesh Feature Visualization: .....	9
II.4) Mesh Segmentation: .....	10
Results:.....	11
III.1) Mesh feature Visualization and Analysis: .....	11
Average geodesic distance .....	11
Shape Diameter function:.....	11
Curvature: .....	11
Volumetric Shape Images: .....	11
Shape context: .....	11
III.2) Mesh feature extraction: .....	12
III.3) Mesh Classification Analysis: .....	13
Future work: .....	14
References:.....	14

## Introduction:

One of the fundamental problems of shape understanding and processing is segmentation of the shape. Many of the problems require labelled segmentations where parts of the mesh are mapped to identified parts i.e every face of the mesh is assigning a part object label. Segmentation of 3D meshes has extensive applications in mesh editing, deformation, modelling, manufacturing, animation and many other fields.

However, manually labeling a mesh can be labour intensive, time intensive and error prone. This serves a motivation to devise methods to automate this process. Existing solution to these problems can broadly classified into knowledge-driven methods and data driven methods. In the knowledge-driven method, geometric and structural patterns are extracted and interpreted with the help of explicit rules. Although these approaches successful, they have the major limitation i.e. it is extremely difficult to generate rule to that can handle the enormous geometric and structural variability.

Alternatively, data-driven techniques learn representations and parameters from data. In data driven approach an ensemble of features is used to establish a correlation between the source and the target shapes and transfer the interesting information from the source to the target. The description acts an effective and compact representation of the mesh and is used to train a classifier such as SVM, Random forest or fully connected neural network.



*Overview of data driven shape analysis and processing.*

This project attempts to understand the surface and volume properties of a mesh by implementing and visualizing the various mesh feature algorithms. Moreover, utilize these features to train a classifier to automatically segment and label the mesh.

Overview of the project report is as follow:

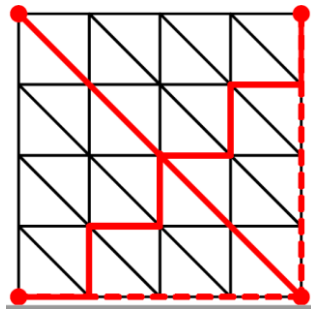
- 1) Brief description of mesh feature descriptor algorithms which include Average geodesic distance, Curvature, Shape diameter function, Volumetric shape images, Shape context.
- 2) Brief description of Mesh segmentation and labeling pipeline using a classifier trained on the extracted features
- 3) Details of implementation outlining the system architecture, class structures of feature generation, mesh classification and visualization.

## I) Mesh Features:

3D Meshes have variations in resolutions, scale, orientation, and structure. Hence, it is very difficult to devise a common parameterization or alignment. For this purpose, a variety of shape descriptors have been developed. This project implements, few local feature descriptors which generates effective representation of mesh properties such as mesh surface and mesh volume. This project implements Average Geodesic distance, Shape Diameter function, Curvature, Volumetric shape images and shape context.

### I.1) Geodesic distance:

Geodesic distance between two points on the surface of the mesh is the shortest distance between without leaving the manifold. The shortest path algorithms like Dijkstra's algorithm do not converge on the geodesic distance but can provide a good approximation.



Fig(a)



Euclidean

Fig(b.1)

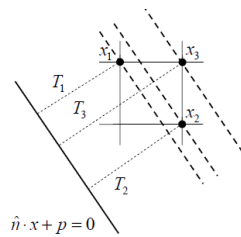


Geodesic

Fig(b.2)

Fig(a) shows the errors of dijkstras algorithm Fig(b) Difference between Euclidean and Geodesic distance.

One of the methods used to compute the geodesic distance is using fast marching algorithm(FMA). Geodesic distance computation using FMA assumes that propagation plane on the surface is planar. This is a reasonable assumption as further from the point from source, more planar the propagation plane becomes.



Planal front propagation

Fast marching algorithm is as follows:

Initialization:

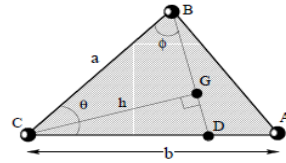
- 1) Choose a source point  $T(X_0) = 0$ , Mark point  $X_0$  as Alive
- 2) Mark all neighbours of  $T(X_0)$  as Close. The T values of points are initialised as the Euclidian distance.
- 3) Mark all other points as Far and initialized T value to infinity.

Loop:

- 1) Find point P in Close set with the smallest T values rename as Trail.
- 2) Add Trail set to Alive, remove from Close set.
- 3) Move all the adjacent vertices of to Close set. Also consider points in the Far set.
- 4) Update the T value of the trail.
- 5) Loop until there are no vertices in Far and Close set.

Update equation:

Considering a triangulated mesh, the equation to update the T values is given by the solution to the quadratic equation. Here the Trial C is being update and vertices A and B are in Alive set.



*Triangular face of mesh*

$$T = (a^2 + b^2 - 2ab * \cos\theta)t^2 + 2bu(acos\theta - b)t + b^2(u^2 - a^2 * \sin\theta * \sin\theta) = 0$$

## I.2) Shape diameter function:

Shape diameter functions is a poise oblivious scalar feature, which does not change on any rigid body transformation. Any objects volume from surface is defined as the distance from the surface to the medial axis. However, computing the medial axis is complex and error prone. SDF is an effective approximate to object's volume from its surface which is also simple to compute.



*Shape diameter function on different points on the mesh*

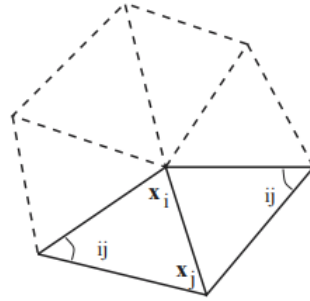
Algorithm to compute the SDF for any point on the surface of the mesh is given as follows

- 1) Construct cone of a given angle, with the centre as the around inward normal. The inward normal is the opposite direction of surface normal.
- 2) Shoot predefined number of rays into the cone.
- 3) Compute intersection of the ray and mesh.
- 4) Filter ray which lie outside one standard deviation from the median of all lengths.
- 5) Allocate weights inversely proportional to angle between the ray and face normal.
- 6) SDF of the point is the weighted average of all the ray lengths.

The ray-mesh intersection can be computed using algorithm like Möller–Trumbore ray - triangle intersection algorithm.

### I.3) Curvature:

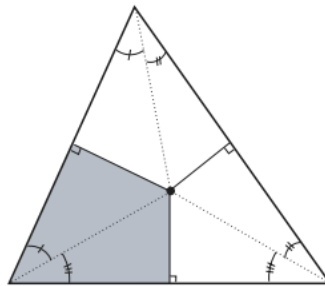
As triangular meshes are approximately represent the continuous surfaces, defining continuous normal vectors and curvature on the piecewise linear meshes is challenging.



*1-ring neighbour vertices of vertex i, with angles as  $\alpha, \beta$*

Area of the voronoi around the point, assuming the triangle is a non-obtuse triangle is given by

$$A_{Voronoi} = \frac{1}{8} \sum_{j \in Ni(i)} (Cot \alpha_{ij} + Cot \beta_{ij})(x_i - x_j)$$



*Voronoi region of a triangle.*

The algorithm to compute the area in the 1-ring neighbourhood x, which may contain obtuse triangle is given as follows.

- 1) Initialize  $A_{mixed} = 0$
- 2) Loop for each triangle in the 1 ring neighbourhood of point P
  - if T is non-obtuse
  - $A_{mixed} += \text{Voronoi region of } x$
  - Else
  - If the angle at x is obtuse
  - $A_{mixed} += (\text{Voronoi region of } x)/2$
  - Else
  - $A_{mixed} += (\text{Voronoi region of } x)/4$

### I.3.1) Mean Curvature:

From the area of 1-ring neighbourhood the mean curvature K can be computed using the equation

$$k(x_i) = 1/(2 * A_{mixed}) \sum_{j \in Ni(i)} (Cot \alpha_{ij} + Cot \beta_{ij})(x_i - x_j)$$

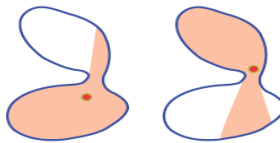
### I.3.2) Gaussian Curvature:

Similarly, the gaussian curvature can be computed

$$k_G(x_i) = (2 * \pi - \sum_{j=1}^{\#f} \theta_j) / A_{mixed}$$

### I.4) Volumetric shape images:

Volumetric shape image is motivated that change in part mostly corresponds to distinct change in volume.

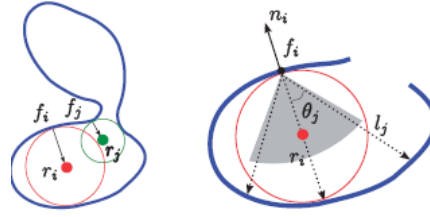


*Change in part has change in volume.*

Matching full volumes of a model is an expensive process. Hence to capture the shape, the volume is sampled across multiple faces.

The algorithm to compute the VIS of a mesh is given as follows

#### I.4.a) Computing the reference point $R_j$

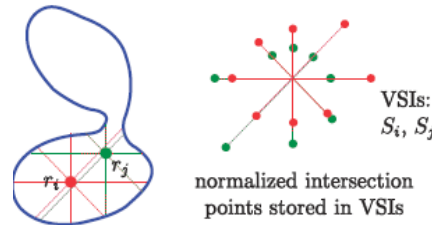


Computing the VIS using SDF

The volume of the shape can be best captured from the medial centre. However, computing the medial centre can be expensive. Hence, medial centre is approximated using SDF.

Using SDF algorithm, the longest ray intersecting the mesh is identified. The mid-point of the ray  $\theta_j$  can be considered as good approximate to the medial centre.

#### I.4.b) Computing VIS

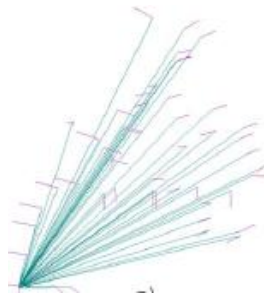


Computing the VIS

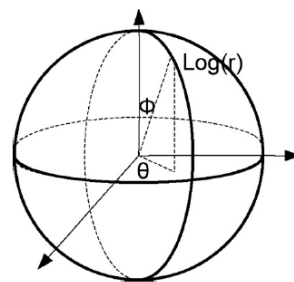
With medial centre as a reference point,  $m$  rays are sent out from this point onto a uniformly sampled gaussian sphere. The length of ray from the reference point to the intersection point on the gaussian sphere is called the volumetric shape image.

#### I.5) Shape Context:

Shape context feature couples the surface property of average geodesic distance with the orientations of the face normal. For a given face centre  $P_i$ , A histogram is computed but considering the angle between the face normal and the vector to other  $P_{i-1}$  face centres. This histogram is called the shape context of the face. The Shape context value are binned in uniform log-polar space, hence this descriptor is more sensitive to closer points.



Fig(1) Direction of normals wrt to the point



Fig(2) Log-Polar bin histogram.



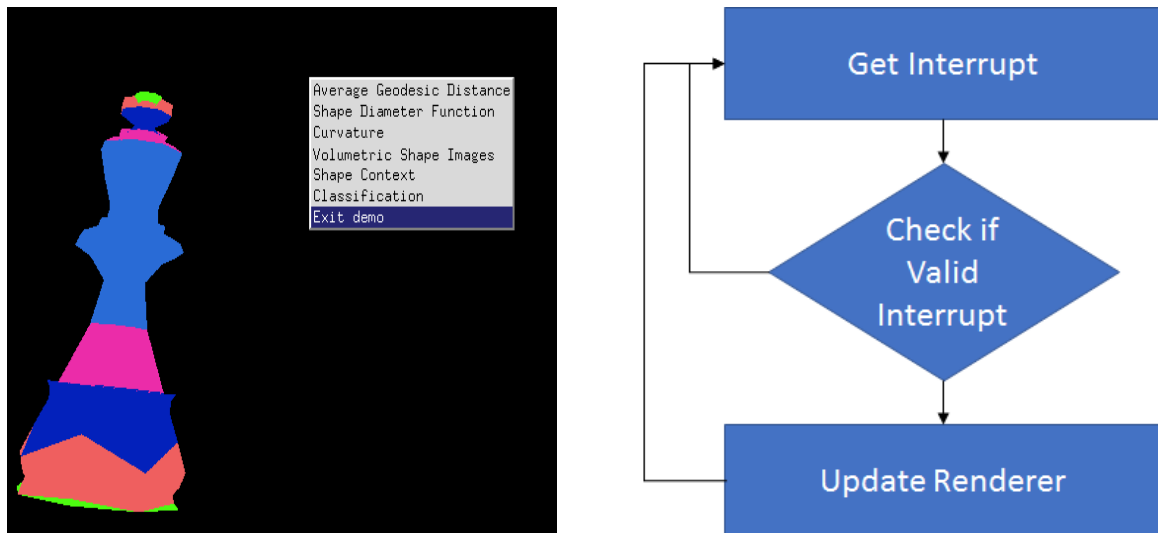
## II) Implementation:

The Implementation of the project can be classified into 4 parts:

- 1) User Interface using keyboard and mouse.
- 2) Mesh feature computation.
- 3) Mesh feature visualization.
- 4) Mesh segmentation.

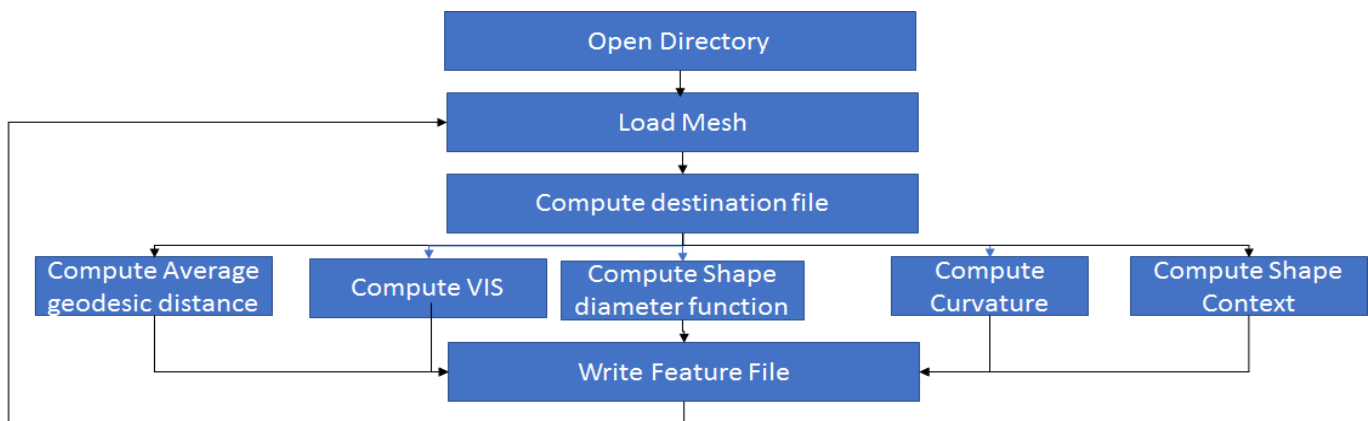
### II.1) User Interface using keyboard and mouse.

A simple UX is consisting of mouse operations to choose the mesh features and keyboard operations for navigation is created. The Keyboard and Mouse interrupts given by the user was intercepted using GLUT Library call-backs as static functions. On Keyboard and Mouse interrupt the input is checked if its valid and the renderer is updated.

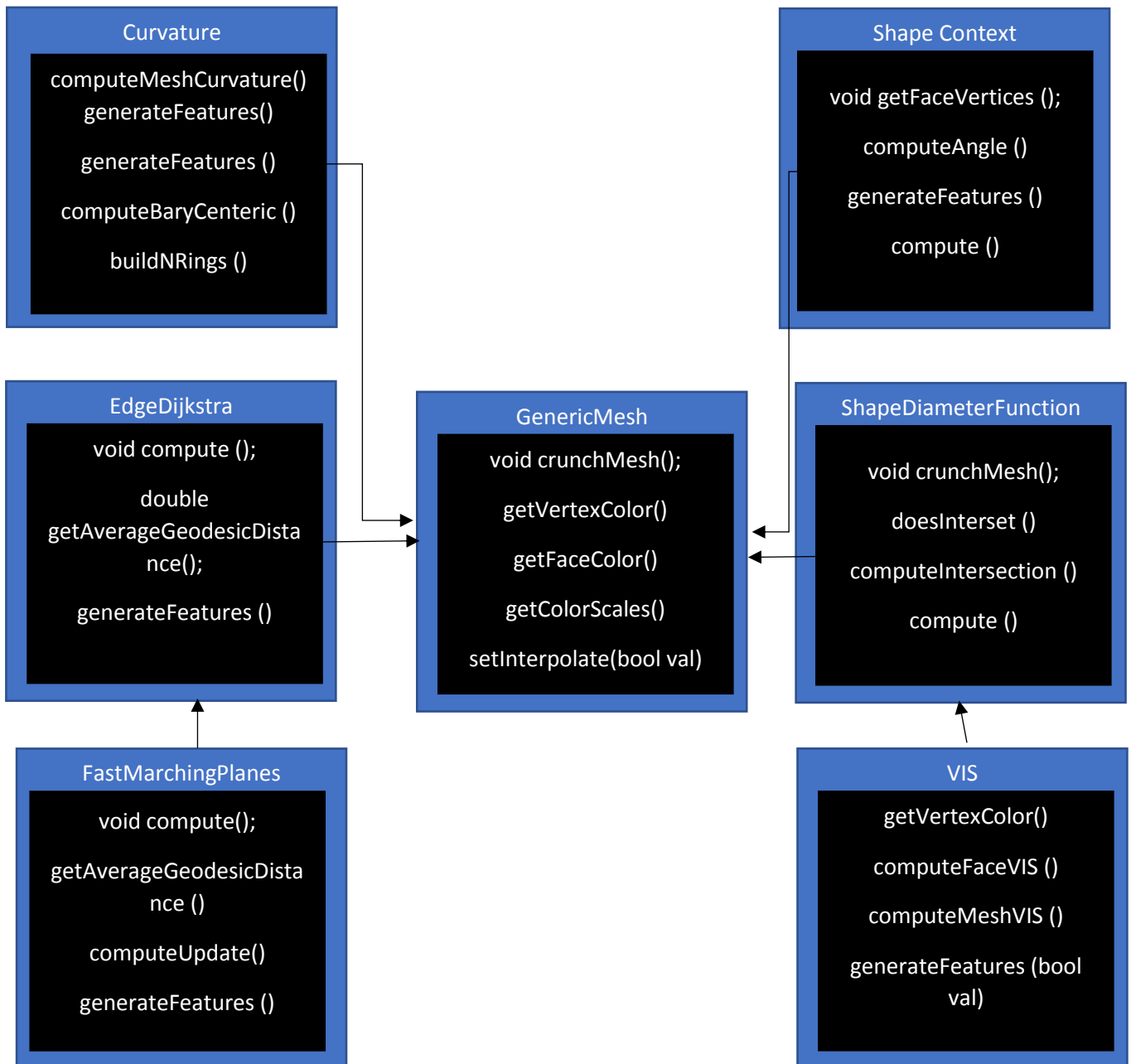


### II.2) Mesh Feature Computation.

This module deals with generating all the features of the mesh. It takes the path of the mesh directory and path to the destination where the features would be written. The features are modular in nature and can be executed independently.



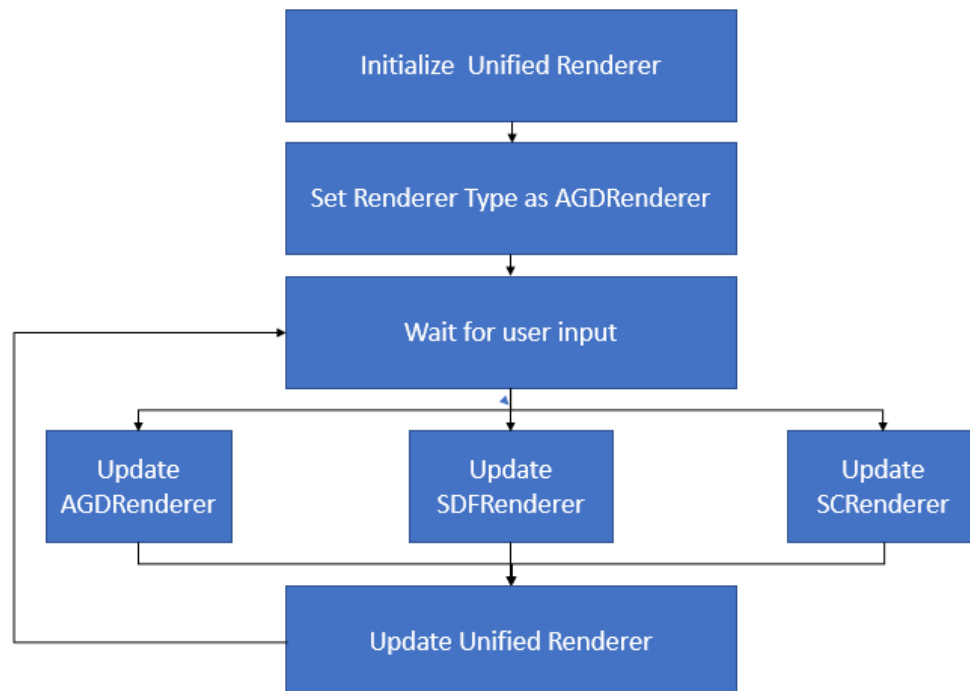
The feature computation is abstracted in a class called GenericMesh, it encapsulates the mesh handling, feature computation and feature representation. The class diagram of the program is shown below.



### II.3) Mesh Feature Visualization:

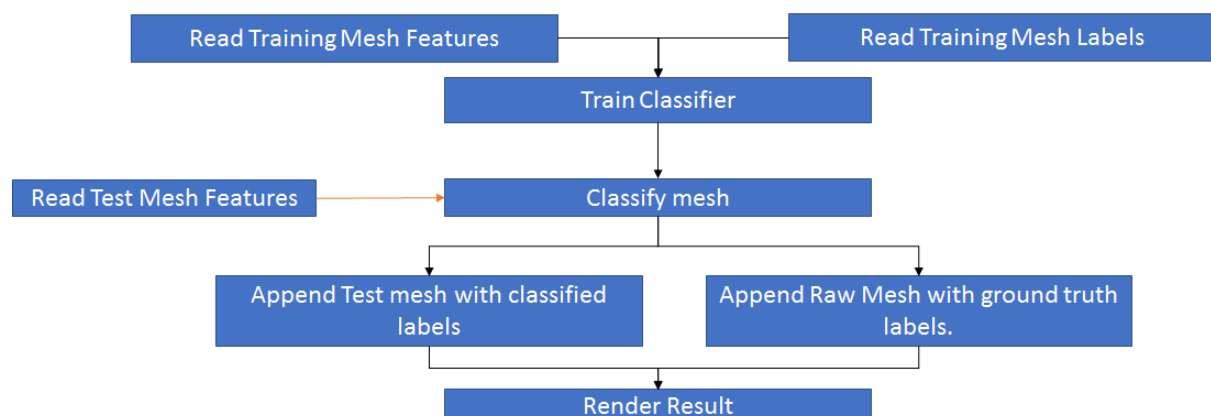
Mesh and feature abstraction from previous stage is rendered using OpenGL and GLSL apis. The renderers of individual features can be executed independently or in a unified renderer. However, each rendered has its own implementation with respect to the rendering method, Vertex and fragment shader and colouring schemes.

The rendering pipeline is as given below:



#### II.4) Mesh Segmentation:

One of the key aspect of data driven approach is the training on labelled data. This project makes use of Plier labelled data set as a target for mesh classification. The dataset contains both the raw mesh and part label for each face of the mesh. The features are extracted for training and validation data and a classifier is trained using the extracted mesh features. Finally, the classification data is made part of mesh data for rendering. Random forest classifier was used to classify the mesh segments.



## Results:

The application was developed in Pentium i7 3.0 Ghz processor on windows operating system, Nvidia GTX740 graphics card with support OpenGL4.0 and GLSL.

### III.1) Mesh feature Visualization and Analysis:

**Average geodesic distance:** Average geodesic distance is a local surface feature of the mesh which captures the distance of mesh face with respect to all the other mesh faces. Based on all the geodesic values, each of mesh face is placed in a colour bin. The colour bins equally space the range of the geodesic values obtained from the mesh. This feature type generates 6 features distinct features.

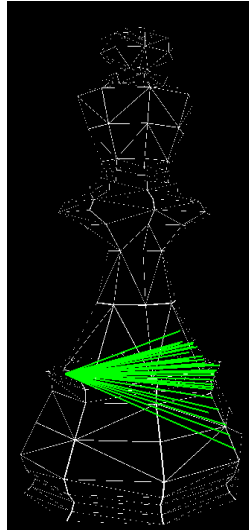
**Shape Diameter function:** Shape diameter function is a volume feature which is also poise invariant. The weighted average of length of the all the rays constitutes the shape diameter function. We generate 104 features from SDF.

**Curvature:** Curvature is a surface feature. Using the mean and the principle curvatures 30 features are computed.

**Volumetric Shape Images:** Volumetric shape image is a volume feature, which provides better approximation to the volume of the mesh at any given point. We generate 104 features from VIS.



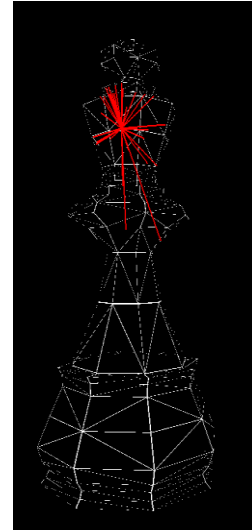
Average Geodesic



Shape Diameter

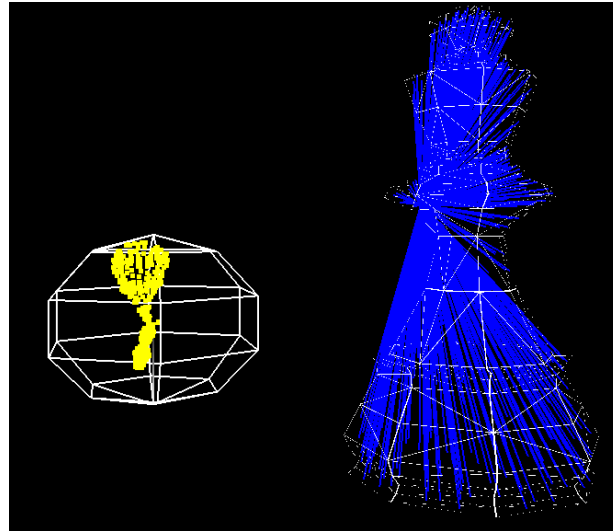


Curvature



Volumetric Shape Image

**Shape context:** Shape context feature is a 3D histogram which couples the geodesic distance of the orientation of the face with respect to the other faces of the mesh. we make use of 30 feature bins to represent the shape context.



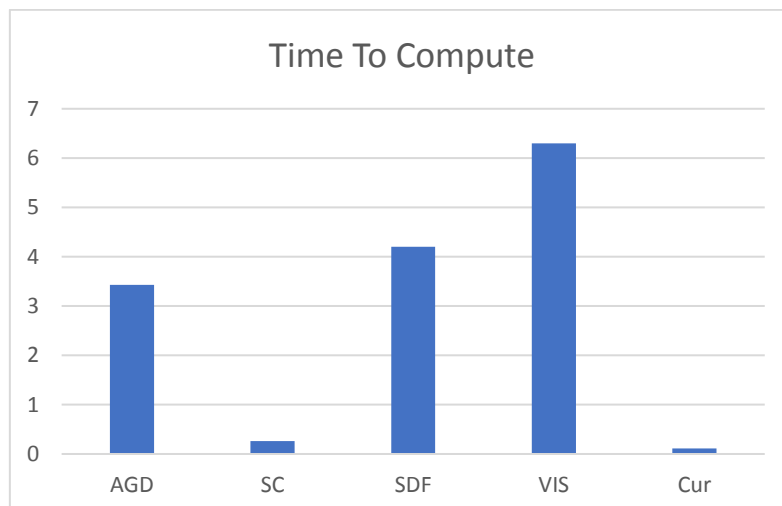
Shape Context.

	AGD	SC	SDF	VIS	CUR
Feature Count	6	30	104	104	45

### III.2) Mesh feature extraction:

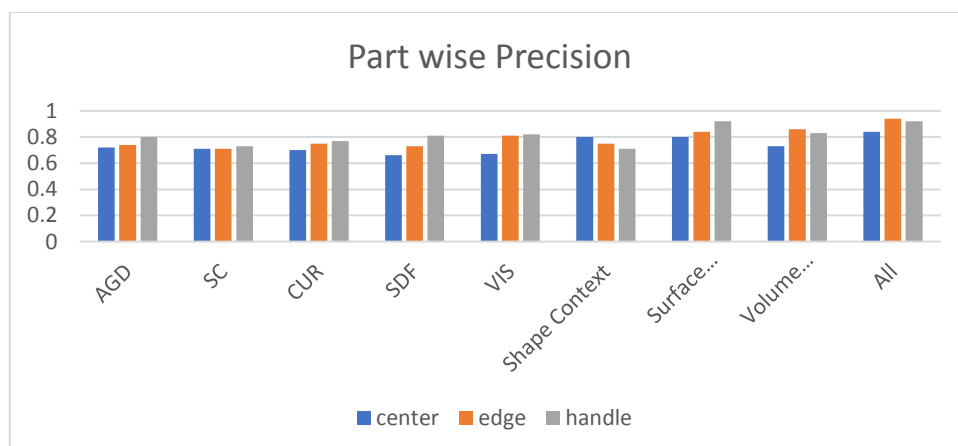
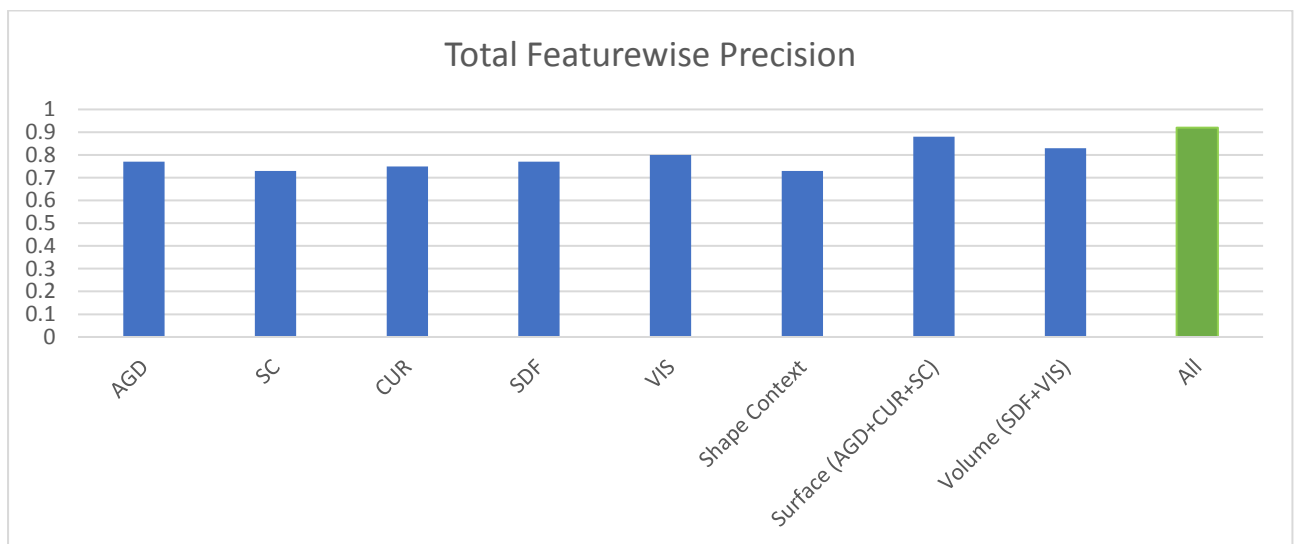
The process of Mesh feature extraction involves executing feature algorithm and recording the data the data. Below is a snapshot of time required to extract the feature from King.off having 314 vertices and 624 faces.

	AGD	SC	SDF	VIS	CUR
Time in sec	3.43	0.26	4.2	6.3	0.112



### III.3) Mesh Classification Analysis:

The features extracted from the test mesh's is used to train a classifier. In the project, a Random forest classifier was trained and the following results are based about the classification results. The Classifier was trained on test dataset of 20 labelled meshes, and the mesh had 3 partwise labels. Using -- features the classifier was able to classifies with a precision of 92%. The Sample of the classification result is shown below.



## Future work:

While the project gave a good insight into the mesh properties and application of the features, many ideas and aspects of the mesh segmentation problem must be studied. The future goal of the projects is as follows.

- 1) Implement a pairwise feature and apply graph cut algorithm to better segment the mesh.
- 2) Improve execution time of Average geodesic distance using idea of heat kernels.
- 3) Improve execution time of Shape diameter function using octree.
- 4) Explore deep learning to improve classification, use convolution neural network to combine similar feature types.
- 5) Identify “cheap to compute” feature which can be used to drive a fast classifier such as extreme learning machine to handle online segmentation.

In conclusion, A Data driven approach to mesh segmentation provides an excellent solution to automatic segmentation. However, the underlying features extracted from the mesh provides the fuel to drive the classifier forward. Hence, developing and improving mesh feature algorithms should be further researched.

## References:

- [1] SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.* 24, 4, 249–259.
- [2] LIU, R. F., ZHANG, H., SHAMIR, A., AND COHEN-OR, D. 2009. A Part-Aware Surface Metric for Shape Analysis. *Computer Graphics Forum, (Eurographics 2009)* 28, 2.
- [3] KIMMEL, R., AND SETHIAN, J. A. 1998. Computing geodesic paths on manifolds. *Proc. of National Academy of Sci.* 95(15) (July), 8431–8435
- [4] MEYER, M., DESBRUN, M., SCHROEDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III (Proceedings of VisMath 2002)*, Springer Verlag, Berlin (Germany), 35–54.
- [5] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM Trans. On Graph (SIGGRAPH)* 29, 4 (2010), 102:1–102:12. 1, 2, 3, 5, 7, 8