

Tutorial n°4 3D Mesh processing

1 Mepp - 3D MESH Processing Platform

MEPP is a platform development environment based on the class "Polyhedron" of CGAL, for processing and visualization of mesh and mesh sequences. It allows the loading of multiple meshes or mesh sequences, processing and visualization. It is intended for engineers, researchers but also to students with a quick start, facilitated by the proposed architecture.

Based on CGAL, Qt, libQGLViewer, OpenGL, Boost, Assimp, FFmpeg and CMake, MEPP aims at building a framework around novel mesh processing techniques. Oriented towards modularity, it targets developers as well as GUI users.

Features :

- C++, Open Source (GNU GPL v3),
- multiplatform (Windows, Linux, Mac OS X),
- compilation with CMake,
- fast and easy installation,
- installation documentation, user manual,
- VMware virtual machine with Ubuntu Linux "ready to start",
- drag and drop objects from one application window or from a browser.

Supported formats :

- obj (Wavefront),
- off (Object File Format),
- ply (Polygon File Format),
- smf (3D World Studio),
- x3d (XML based royalty-free open standard file format),
- 3ds (3ds Max),
- dae (Collada),
- lwo (LightWave).

Exports :

- image capture,
- video capture.

Display features :

- management of multiple objects in one or more windows,
- accelerated display with "Display Lists".

MEPP platform allows two types of loading :

- mode "space" in which several objects are loaded into the same scene,

- mode "time" in which several objects are loaded as a sequence of meshes.

It is then possible to visualize the sequence through a VCR 3D + t.

Components :

- implemented as plugins with automatic detection and loading,
- selection of components at compile time,
- easy creation of new components from an example.

Available components :

- basic mesh processing : triangulation, subdivision, simplification,
- curvature analysis,
- boolean operations : union, intersection, subtraction,
- segmentation (Variational Shape Approximation),
- progressive compression/watermarking,
- visual quality metrics,
- remeshing : canonical simplification,
- mathematical morphology,
- mesh repairing...

Possibility to contribute to the platform by creating new components.

2 Exercice

1. Install the source code of MEPP from <https://projet.liris.cnrs.fr/mepp/developers.html>
2. Contribute to any one of the components by adding a function that apply Taubin smoothing on a mesh with λ and μ equal to 0.6307 and -0.6732 , respectively (values suggested by Taubin).