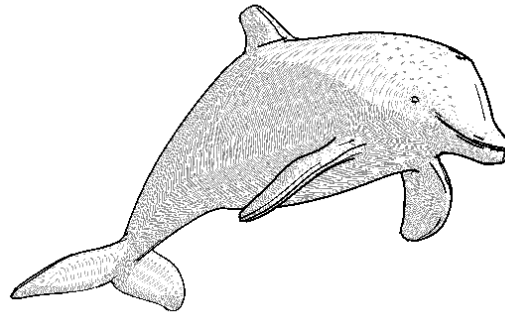
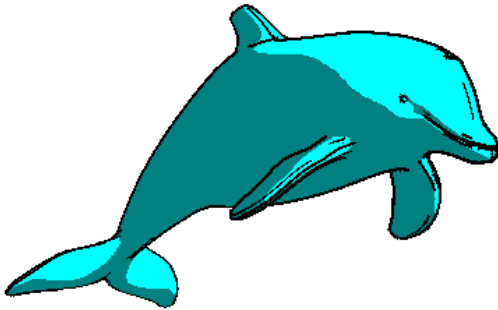


## Assignment 2

Due: 11:55pm, Friday, 24 Sep 2021

Maximum Marks: 20



### Non-Photorealistic Rendering Using OpenMesh and OpenGL-4

In this assignment, you will develop methods for non-photorealistic rendering of three-dimensional mesh models. The OpenMesh library will be used for loading mesh objects and for creating the definitions of triangle adjacency primitives (GL\_TRIANGLES\_ADJACENCY). The triangle adjacency primitives are used for the computation of silhouette and crease edges.

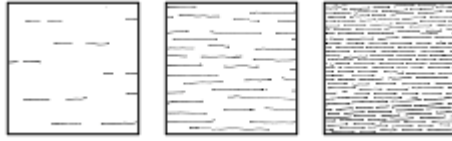
#### I. Non-Photorealistic Rendering (NPR):

In this assignment, we will focus on two types of non-photorealistic rendering methods: two-tone shading and pencil shading. Two-tone rendering uses only two shades of a material colour, while pencil shading uses a set of pencil stroke textures. Both methods require highlighting or darkening of important edges of the output of the mesh model. The main requirements of this assignment are outlined below:

##### **I(a): Basic NPR (Max. 12 Marks)**

- Your program must use the OpenMesh library for loading mesh models and for generating the element array for triangle adjacency primitives. You may use the program “MeshViewer.cpp” in the Programming Exercises section as the base code for this assignment, and add the required features to this program. A set of mesh models is also provided.
- You must include a geometry shader in the program. Most of the computations will be performed in this shader.
- The silhouette and crease edges of the model must be highlighted in the output.

- You must create at least three pencil stroke textures corresponding to three different shade levels. Hand drawn sketches or procedurally generated images may be used for this purpose. A sample set is shown in the figure below.



- Your program must be capable of producing both types of NPR outputs (two-tone rendering and pencil shading). Please include the following keyboard bindings for user interaction:
  - ‘ ’ (space) : Toggle between two-tone rendering and pencil shading
  - Arrow keys: Rotate the model about x and y axes
  - Page up/down: Zoom in/outYou may include additional keyboard/mouse functions as necessary (see next section).

### **I(b): Extra features (Max. 5 Marks)**

Some of the additional features that you could implement to gain extra marks up to a maximum of 5 marks are listed below:

- [2 marks] A mipmap set of textures could be used to improve the quality of rendering of pencil stroke textures. You will require three or four mipmap levels for each texture (e.g., 64x64, 32x32, 16x16, 8x8).
- [1 mark] You may provide the functionality to independently vary the thicknesses of silhouette edges and crease edges. Please use keyboard bindings ‘q’, ‘a’ to increase and decrease the thickness of silhouette edges, and ‘w’, ‘s’ keys to modify the thickness of crease edges.
- [1-2 Marks] You may be able to improve the methods described in lecture slides (Slides [6]:26-28) for rendering silhouette edges and crease edges. For example, you could render the edges as view oriented billboards. The lengths of the edges could be increased to overlap with neighboring edges, to remove/reduce discontinuities along edges.
- [1 Mark] The texture coordinates for mapping pencil textures to triangular faces of a model could be specified based on the direction of local curvature so that the stroke lines approximately follow the direction of curvature. Please provide a screenshot in the report showing this feature.
- [2 marks]. Border edges are edges belonging to only one triangle. Border edges can also be easily detected and highlighted in the output. You will need to use a model containing border edges.

- [1-2 Marks] Certain types of texture features or operations could enhance the rendering quality of pencil shading. For example, the stroke lines could be made lighter or darker based on illumination conditions.

The list given above should not be taken as the complete set of features that can be implemented.

## II. Report (3-4 pages; Max. marks: 3):

Please prepare a brief report containing a description of the following:

- A brief outline of the extra features implemented, and any model-specific characteristics (e.g., large number of crease edges, presence of border edges, regions of large gradient/curvature, material colours etc.)
- A brief description of the methods used for texture generation, mapping and edge rendering (if different from the methods given in lecture slides).
- Any problems encountered and how you attempted to solve them.
- The complete list of keyboard/mouse functions defined for user interaction. Please also include references to the sources of textures, models etc. used in your program, if different from what was provided in the course.

## III. Program Development:

You may use math library functions (eg. GLM), mesh models, and images that are available on the Internet or obtained from other sources such as books. Please acknowledge the source in your report. You may also use programs and other supplementary materials provided in this course. If any part of your implementation is based on a method described in a paper, book etc., please give full details of the source in the list of references. Please use only C/C++ as the programming language for application development. Demo programs found on the Internet and other OpenGL resources should not be submitted as part of the assignment. Please do not use OpenGL Extensions (ARB, EXT etc) or third-party mesh processing libraries other than OpenMesh in your program.

## IV. Assignment Submission

Submit your files using the assignment link on Learn ([learn.canterbury.ac.nz](http://learn.canterbury.ac.nz)) before midnight on 24 Sep 2021. Your submission must contain:

1. The source code(s) and all supplementary files (textures, model files, shader code) needed to run your program. Please do not include freeglut, OpenGL, GLEW, OpenMesh or GLM library files with your submission.
2. Your report in Word or PDF format.

Miscellaneous

1. Check regularly on the *Learn* system forums for spec updates and clarifications.
2. You may submit up to one week late for a 15% penalty.
3. This is not a group project. Your assignment must represent your own individual work. In particular, students are not permitted to share program source code in any way. However, you may discuss ideas, implementation issues etc using the class forum on Learn.
4. Standard departmental regulations regarding dishonest practices and late submissions apply.