CS4102 Computer Graphics: Practical 2
3D Rendering

170004680

University of St. Andrews

April 2021

Contents

1 Compilation, Execution and Usage Instructions

1.1 Compilation and Execution

From within the `src/` directory, the program can be compiled and executed using (see figure 1):

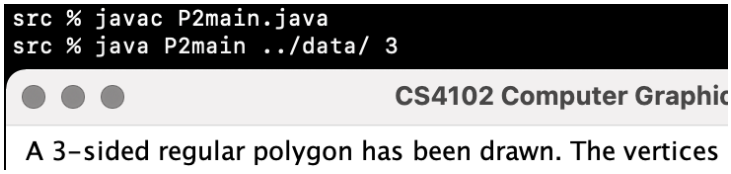
javac P2main.java

java P2main <data_dir> <num_reference_faces> [-fs|-is|-wf] [-l|-nl]

, where:

- '<data_dir>' is a relative path from the `src/` directory to a directory containing the reference face files.
- '<num_reference_faces>' is an integer representing the number of reference faces that should be used on program initialisation (between 3 and 199 inclusive).
- The '[-fs|-is|-wf]' arguments are optional and specify the technique to render 3D faces: '-fs' refers to flat shading (default), '-is' refers to interpolation (Gouraud) shading, and '-wf' refers to wire-frame (used for debugging).
- The '[-l|-nl]' arguments are optional and specify whether to use directional lighting in 3D face rendering: 'l' instructs the program to use directional lighting (default), and '-nl' instructs not to use directional lighting (rendered faces will be evenly lit).

Figure 1: Screenshot showing the program being compiled and executed with the given instructions; the GUI is correctly presented. Attempts to check the working order of the program on the lab machines was unsuccessful; the GUI program could not be executed via SSH. However, there should be no problems executing the program on a lab machine.



1.2 Usage Instructions

Instructions on how to use the graphical user interface (GUI) are provided within the GUI. A fuller description is given here for clarity:

- *Left-clicking* on one of the points (i.e., vertices) of the polygon in the main window will cause the reference face associated with the clicked point to be rendered in the rendering window.
- *Left-clicking* anywhere within the polygon of the main window causes an interpolated face to be rendered in the rendering window. The location of the clicked point specifies the degree to which reference faces are weighted in the interpolation; the closer the clicked point is to a vertex, the more the corresponding reference face is weighted.
- *Left-clicking* outside of the reference face polygon in the main window does not enact any action (i.e., invalid input).
- *Pressing* the 'left' button, when a face has been rendered, will rotate the rendered face counter-clockwise around the y-axis. Likewise, *pressing* the 'right' button will rotate the rendered face clockwise around the y-axis. *Pressing* the 'auto' button toggles the automatic rotation of the 3D model.

2 Basic Specification - Implementation And Discussion

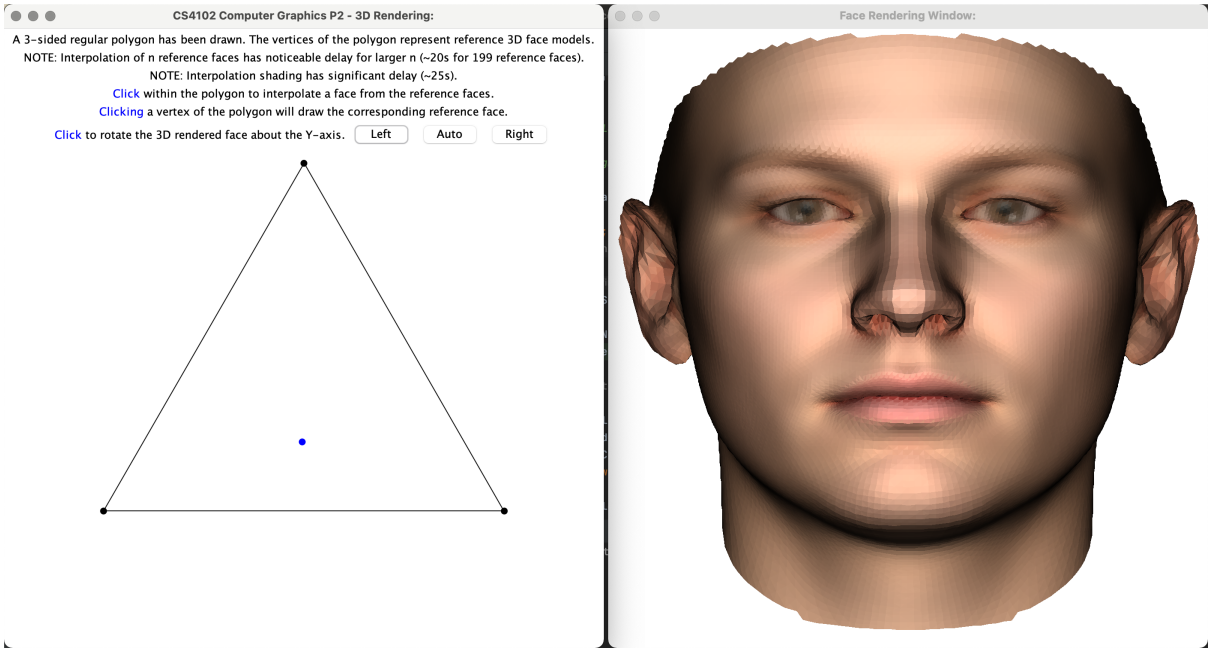
2.1 Reference Face Interpolation

The interpolation of n reference faces, represented by the vertices of an n-sided regular polygon, is achieved using inverse distance weighting. In the implementation, the top vertex of the polygon represents the first reference face, the second (clockwise) vertex represents the second reference face, and so on up to the specified n reference faces ($n = 3$ for the basic specification).

The distance between a specified point and each of the vertices (within the area of the n-sided polygon) determines the weighting of each reference face in the interpolation of a new 3D face. The further the specified point is from a vertex, the less the corresponding reference face is weighted in the interpolation, and vice versa. For example, specifying a point at the centre of the n-sided polygon is equivalent to interpolating with each of the n reference faces weighted equally. Selecting a vertex of the n-sided polygon is equivalent to interpolating with a single reference face being used, so said reference face is rendered (i.e., reference faces can be previewed).

Inverse distance weighting is beneficial as it generalises easily to n reference faces, so all 199 provided reference faces can be used to interpolate a new 3D face. However, this naive technique limits the range of interpolated faces that can be created. Specifying a point on an edge of the polygon should ideally interpolate only between the two reference faces defining the edge, but all other reference faces will have at least a small influence on the interpolated face with this technique. Barycentric co-ordinates were researched to be able to give the desired interpolation property but could not be implemented in its general form (to use an arbitrary number of reference faces).

Figure 2: Example interpolation of a 3D face for the basic specification, which uses 3 reference faces. The specified point (blue) causes interpolation that roughly evenly weights the second and third reference faces, with the first reference face also being weighted but to a lesser extent. The result is produced in the rendering window (right).



2.2 3D Face Rendering

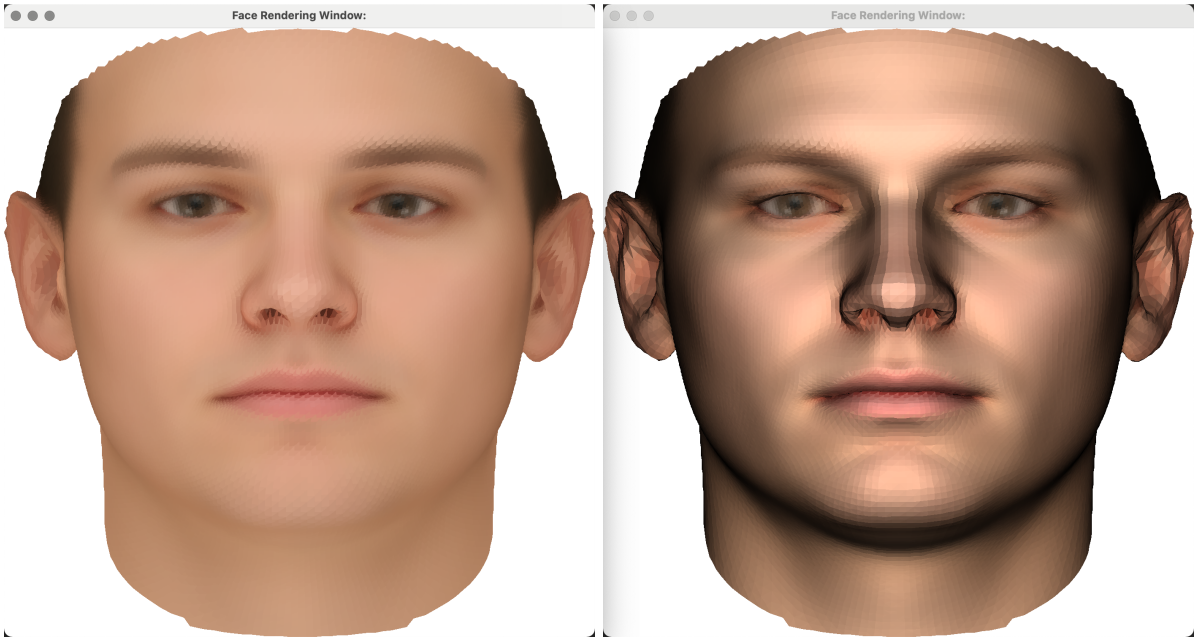
For the basic specification, 3D faces are rendered using painter's algorithm, flat shading, and Lambert's illumination model.

In painter's algorithm, triangles forming the 3D face are sorted by depth (z co-ordinate). Then, triangles are painted in descending depth order, so background triangles are painted over by foreground triangles. This simple algorithm is disadvantaged by the fact that ordering polygons by depth is ambiguous, which can create artefacts. Multiple polygons can overlap such that they all have a point behind each other (i.e., at a greater depth). This issue is not noticeably produced by the implementation, likely because of the connected (and non-overlapping) formation of triangles in the mesh.

In flat shading, each triangle forming the 3D face is assigned a single colour across its surface. In the implementation, the colour of the first point of each triangle was chosen arbitrarily as the colour used. Whilst this form of shading is relatively fast compared to other shading techniques (Gouraud, Phong, etc.), it is disadvantaged by its non-realistic colouring; the polygons of the mesh are distinctly visible in the produced rendering, so the curved surfaces of the 3D face look faceted (see figure 3).

For the basic specification, the 3D faces are assumed to be perfectly matte (i.e., no specularity) and have a unity defuse co-efficient. With these assumptions, Lambert's illumination model was implemented. In this illumination model, light is reflected equally in all directions. Also, the intensity of reflected light is proportional to how parallel the polygon surface normal vectors are with the direction of the light source (single directional light in the viewing direction). Therefore, polygons of the face in regions such as the nose and cheeks are well lit, whilst polygons in regions like crevices, the sides of the head, and under the chin are shadowed (see figure 3). Naturally, human faces are not perfectly matte; they have imperfections, such as grease, that create specular reflections. A more sophisticated illumination model that takes these factors into account (e.g. Phong) will produce more realistic results.

Figure 3: Example of 3D face rendering using flat shading with (right) and without (left) Lambert's illumination model for the first reference face. In both cases, the triangles forming the 3D face are clearly visible (artefact of flat shading), especially in regions where there are larger triangles or sharp colour changes (e.g. eyebrows, lips, and nose). The illumination model provides decent shadowing but is unrealistic; faces are not perfectly matte.



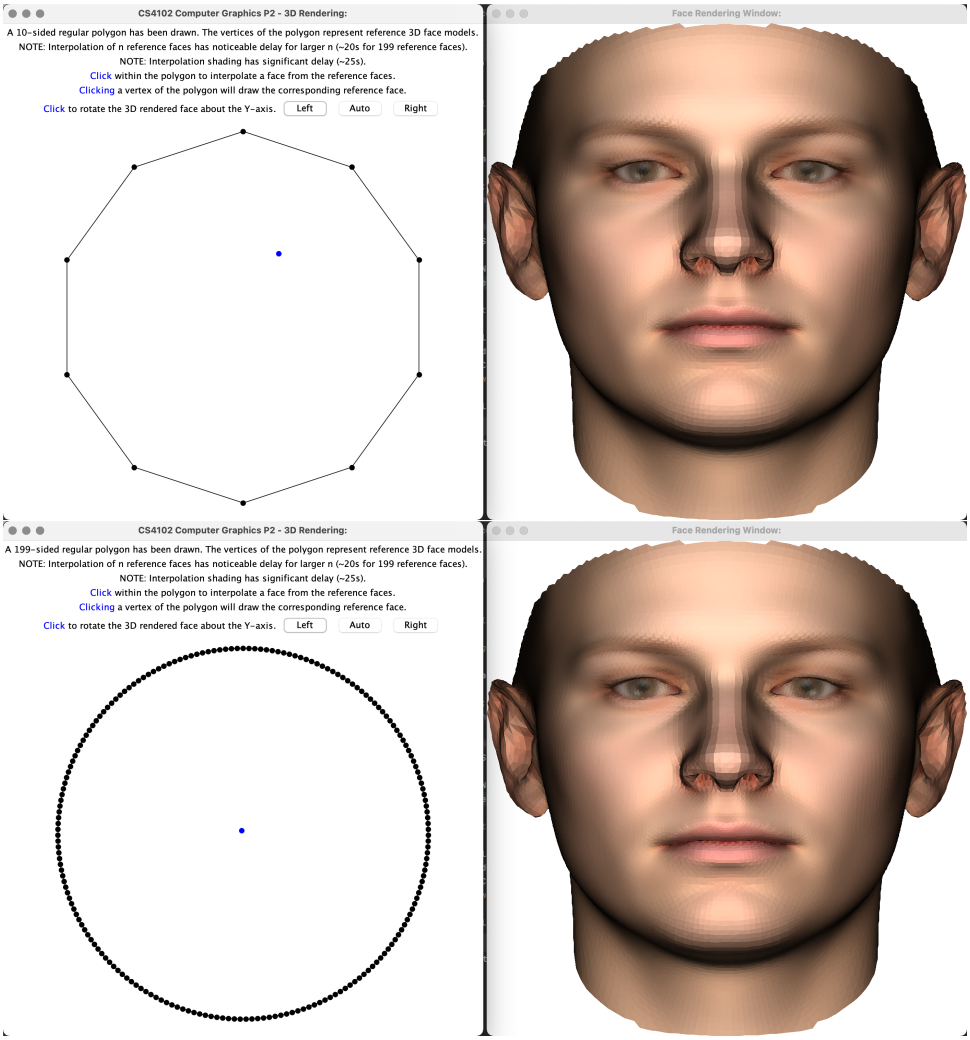
3 Advanced Specification - Interpolation Of 199 Reference Faces

For the advanced specification, the implementation has been extended to allow for the interpolation of up to 199 reference faces (see figure 4). Reference faces can be previewed before an interpolation is initiated by selecting the corresponding vertex of the reference face polygon, as before.

Differences between interpolated faces when using up to 199 reference faces become increasingly subtle, which is expected; all of the reference faces are variants of a common average face. This allows for increasingly fine-grained editing of the rendered 3D face. An improvement to the implementation is one which allows for differences between rendered faces to be shown.

The time taken to interpolate a face is proportional to the number of reference faces specified, n. In the implementation, interpolating amongst n reference faces requires the disk file reading of n CSV files (the '`sh_00n.csv`' files). This is the main bottleneck of the implementation but was chosen as the memory requirements otherwise were too excessive. Despite this, the time taken for the implementation to interpolate a face (at 199 reference faces) is not too significant (20 seconds maximum), and the running time of the program is not a focus of the assessment.

Figure 4: The following screenshots illustrate the successful extension of the basic implementation to allow for interpolation of up to 199 reference faces. In the top screenshot, 10 reference faces was used in the interpolation of the rendered face. In the bottom screenshot, the maximum of 199 reference faces was used.

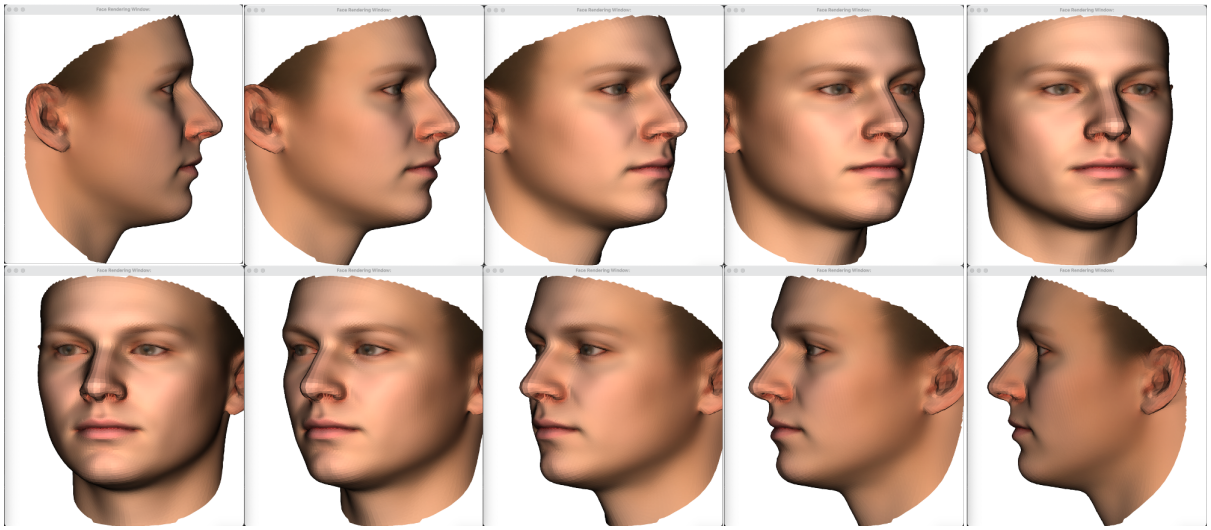


4 Highly Advanced Specification

4.1 3D Model Rotation

As part of the highly advanced specification, functionality has been implemented that allows the user to rotate a rendered 3D model about the y-axis (see figure 5). The rotation is around the geometric centre of the rendered 3D face. Rendering aspects, such as lighting, are re-calculated as the face is rotated (i.e., directional lighting moves with the viewing direction as opposed to being fixed on the front of the face).

Figure 5: The following set of screenshots illustrate the rotation functionality; clicking on the appropriate button creates a clockwise or counter-clockwise rotation about the y-axis of the 3D face. Here, rotation from one side-profile to the other is shown, though the user may rotate completely around the model if desired.



4.2 Gouraud (Interpolation) Shading

As part of the highly advanced specification, another shading technique has been implemented: Gouraud shading (see figure 6).

In Gouraud shading, normal vectors are approximated at each vertex of each triangle comprising the 3D face. Then, colour information given by evaluating the illumination model at the triangle vertices is linearly interpolated across each triangle surface. This is different to the flat shading technique, where a single surface normal is computed per triangle and is used to uniformly shade the triangle surface.

Gouraud shading significantly reduces the faceted appearance visible with flat shaded rendering (see figures 7 and 8). The discontinuities at triangle borders created with flat shading is what produces the faceted artefact. However, Gouraud shading produces smoother transitions between triangles; colour/illumination information is interpolated from the triangle vertices, and two adjacent triangles share two vertices. Overall, Gouraud shaded models look smoother and are more realistic in appearance. However, Gouraud shading is more computationally complex than flat shading, so it is a considerably slower technique. In this implementation, flat shading takes less than a second, whilst Gouraud shading takes minutes. This difference is not solely due to the change in shading technique (the implementation can be greatly optimised), but it is a significant factor.

Note that Gouraud shading often fails to produce realistic shading in the presence of specular reflections. However, artefacts related to this are not visible as the 3D faces are assumed to be perfectly matte. Future work with more sophisticated lighting models could better show the relative performance of common shading techniques (e.g., flat, Gouraud, and Phong shading).

Figure 6: Example of 3D face rendering using interpolation (Gouraud) shading with (right) and without (left) Lambert's illumination model for the first reference face.

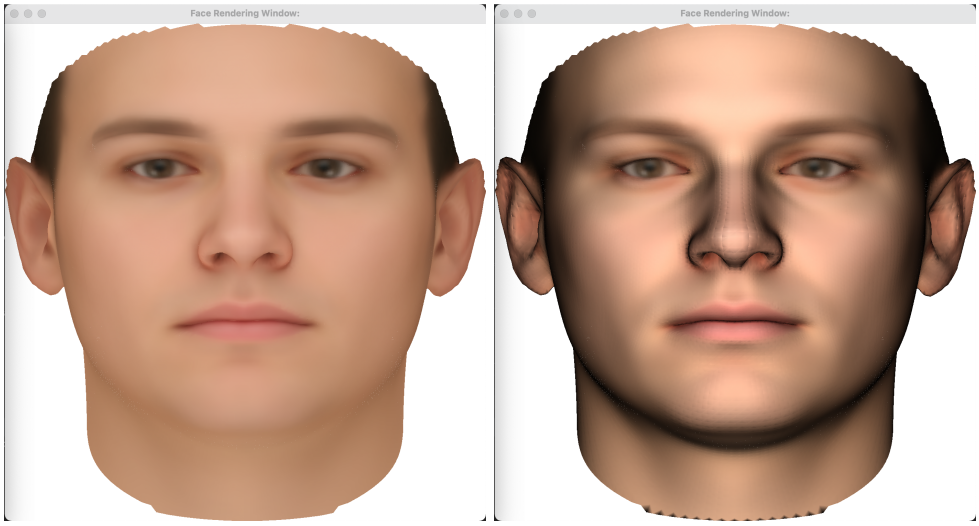


Figure 7: Comparison of rendered faces without directional lighting when using flat shading (left) versus interpolation (Gouraud) shading (right). As seen, colour interpolation across the surface of the triangles removes the artefact experienced with flat shading where the triangles comprising the 3D face are clearly visible.

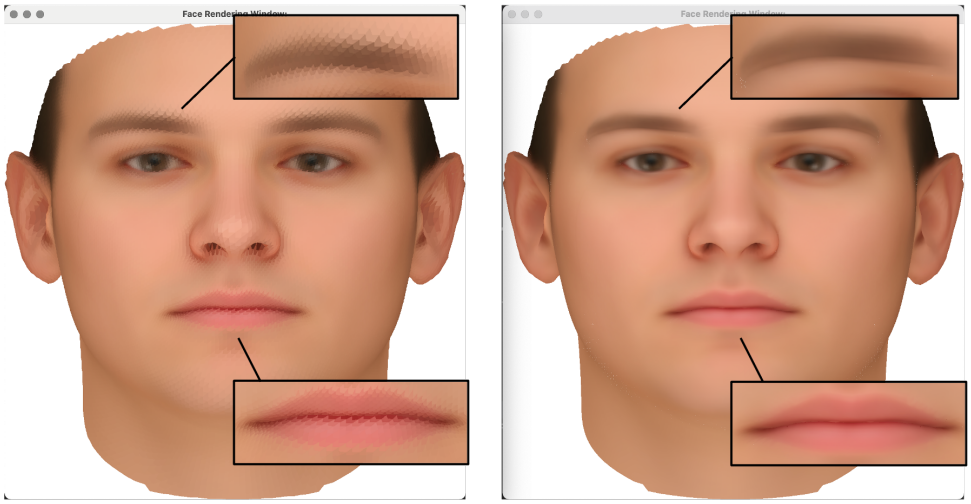


Figure 8: Comparison of rendered faces with directional lighting using Lambert's illumination model with flat shading (left) versus interpolation (Gouraud) shading (right).