COMPSCI 716 S2 T - Assignment 3

Due Date: Monday 2nd October 2006, 9.00pm

Computer Science

This assignment is worth 10% of your final mark for COMPSCI 716 S2 T.

1. *Introduction* *(12 Marks)*

(i) Read the paper “Top Scientific Visualization research Problems” (see 716 web page, Chapter 1) and write the answers to the following questions into the Word document `Ass3Answers.doc`.

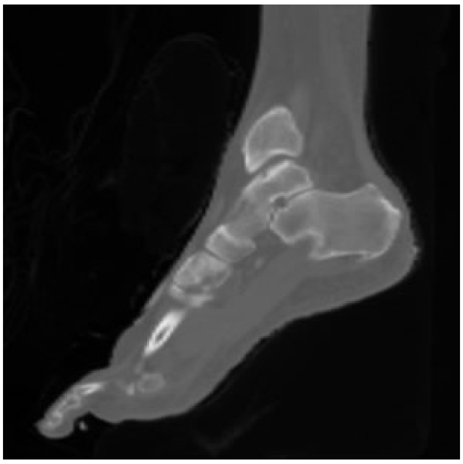
(a) [2 Marks] An important research problem is to quantify the effectiveness of a visualization algorithm. In question 3 of this assignment you have to visualize a sorting algorithm. Suggest a way to evaluate the effectiveness of your visualization of the sorting algorithm, e.g. how can you measure whether your visualization is better than that of a classmate? Please explain your answer.

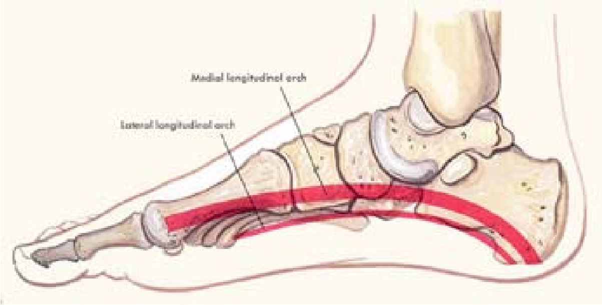
(b) [2 marks] Find at least three references for techniques to visualize uncertainty on the web. Write the URLs with a short explanation of the method (1-2 sentences per URL) into the answer sheet.

(c) [2 marks] Give three examples of applications where it is useful to display visualizations on a PDA or mobile phone.

(ii) Read page 69-83 of the handout “Chapter 4 - Scientific Visualization” (see 716 web page) and write the answers to the following questions into the Word document `Ass3Answers.doc`.

(a) [2 marks] Read section 4.1 of the handout and have a look at the two images below (for colour illustrations please view the electronic copy of this handout on the 716 web page). The image on the left shows a CT (Computed Tomography) slice image of a food and the image on the right shows an illustration of an anatomy textbook. Explain why CT images alone represent a bad visualization and what information must be added to the visualization in order to make it more informative.



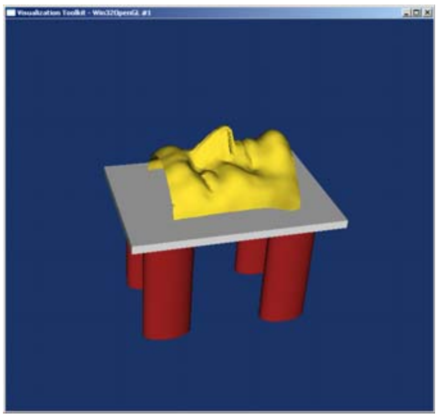


(b) [2 marks] Suggest techniques to visualize a 2D deformation field, which gives for each point of an object the corresponding location of the deformed object.

(c) [2 marks] Chernoff faces (see page 94 of the handout “Chapter 4 - Scientific Visualization”) utilize the brain’s ability to quickly recognize small changes in facial expressions. Think about how you recognize facial expressions and then identify 4-5 basic facial features which can be use to classify facial expressions.

2. Introduction to VTK (10 Marks)

[10 Mark] Download the .NET solution Ass3Q2.zip and complete the file Ass3Q2.cxx so that it renders a scene similar to the one below. The scene consists of a yellow face which lies (approximately centred) on a table. The table has a white plate and four reddish legs with an ellipsoidal cross section. The render window has a size of 300x300 pixels and the camera has an associated trackball (see the `VTK_Cone_Tutorial.zip` for an example).



3. Visualisation of a Sorting Algorithm (23 Marks)

This question is a **peer programming task**. You are allowed to choose **one partner** and you can both submit the same solution. If you decide to do this task together with another student clearly state in the document `Ass3Answers.doc` who your partner is (Name, UPI). You will both get the same mark for this part of the assignment. If you prefer to work alone you may do so.

(a) [5 marks] Implement a sorting algorithm using the C/C++ language (you are allowed to use existing code, e.g. code from the web or from COMPSCI 220, as long as you reference it). Use as input an array of 30 random Integers between 0 and 99 and use an OpenGL window to output the original and the sorted array.

Note that you can output text in OpenGL by using something like this:

```
#include <gl/gl.h>
#include <gl/glu.h>
#include <gl/glut.h>

char text[10];
glRasterPos2i(20,20);
for(char *s=text;*s!='\0';*s++) glutBitmapCharacter(FONT, *s);

// use for FONT one of the following predefined constants
// GLUT_BITMAP_9_BY_15, GLUT_BITMAP_8_BY_13, GLUT_BITMAP_TIMES_ROMAN_10,
// GLUT_BITMAP_TIMES_ROMAN_24, GLUT_BITMAP_HELVETICA_10,
// GLUT_BITMAP_HELVETICA_12, GLUT_BITMAP_HELVETICA_18
```

(b) [5 marks] Think about what “features” of the sorting algorithm you need to visualize in order to demonstrate to the viewer the basic idea behind the algorithm. For example, QuickSort uses a “divide-and-conquer” principal, so the visualization should clearly indicate how the array is divided into smaller sub-arrays and how these are merged subsequently. How can you use knowledge from “Visual Perception” research in order to make your visualization more efficient (see section 4.4 of the handout “Chapter 4 - Scientific Visualization”)? Write the answer to this question into the Word document `Ass3Answers.doc`.

(c) [10 marks] Implement an effective visualization of the sorting algorithm chosen by you. Please submit both a .NET solution (without binaries) and a working executable. Think carefully how you can make the best use of the screen space, color, shape, animations, markers etc. in order to clearly visualize the working of the sorting algorithm. If you complete this part of the question you do not need to submit a separate program for part (a).

(d) [3 marks] Think about how your program could be modified in order to visualize the sorting of an array with 10,000 elements. Write the answer to this question into the Word document `Ass3Answers.doc` (no implementation is necessary).

☺ *Enjoy!*

Assignment submission

All files should include your name, UPI and ID in a comment at the beginning of the file. All your files should be able to be compiled in the CS lab under .NET 2005 without requiring any editing. In particular they should include the correct VTK and/or OpenGL libraries. All your files should include adequate documentation.

The assignment due date is Monday the 2nd October 2006 (time: 9.00pm).
Please submit your assignments using the assignment dropbox. If the dropbox server is down please create a zip-file which contains all the files you want to submit and email it to me. The name of the zip-file must be "Ass3_`<your_UPI>.zip`" (e.g. `Ass3_bwue001.zip`).

Please Submit:

- A Word document `Ass3Answers.doc` with the answers to the questions 1, 3 (b-d)
- the file `Ass3Q2.cxx` with the missing code completed.
- If you chose a partner for question 3 clearly state his/her name and UPI.
- A .NET solution (.sol, .vcproj, .cpp and .h files) and a working executable for question 3.