

# Visualisation Course 2008, UG4 – L10 (ONLY)

## Assignment 2 – Visualisation of Glider Flight Path

This assignment is the second of two assessed practical exercises for the visualisation module. The aim of the practical is to visualise a set of GPS co-ordinates recorded from a glider in flight. This practical is again open-ended in its nature. You are supplied with the data, a description of its nature and a few suggested methods of visualisation. Your task is to produce a suitable visualisation of the data.

### Background

The file `vtkgps.txt` contains data from a satellite global positioning system (GPS) during the flight of a glider.

Each point in the data file is defined by 4 quantities:

- X position (in metres)
- Y position (in metres) (both defined on the Swedish grid)
- Height (altitude) above datum (reference point) in metres.
- Time in seconds

Position of the glider was obtained from satellite GPS measurements that are accurate to about 10m. Altitude was obtained from a barometric altimeter and is accurate to a metre or better. Occasionally spurious points can be generated by GPS when the signal is weak. You will need to identify these spurious points and remove them. There are 1339 data points in the file, spaced between 2 and 3 seconds between each sample.

*The glider was flying in the Swedish mountains in conditions glider pilots refer to as wave lift. The glider took off at 10:35am from a frozen lake at an altitude of 500m above sea level and was towed by a tow plane to 950m and released. As can be seen from the data the glider then climbed in the wave lift to a maximum of 1782m at 10:52am. This wave lift is formed in the lee of mountains when the air bounces up and down following its ascent over the mountain. A glider flying in these conditions can gain altitude by flying in the rising air and avoiding the sinking air. Gliders can reach extraordinary heights – the world record is over 15km – much higher than commercial airliners that fly at around 10-12km.*

### Task

This data consists of 3D dimensional points arranged in time. Your task is to visualise the pattern of rising and sinking air in the data corresponding to the rise and fall of the glider. *Can the 'wave pattern' of rising and sinking air be seen over the ground? Can it be seen to change with time or with height?* These are the visualisation questions – by visualising the data we hope to be able to determine the answers. Since the glider is continuously moving you will need to make assumptions about the data in order to grid it for plotting – for example assume the pattern of lift is stationary over time. This is not an unreasonable assumption, but you should try to test any assumptions you make, if possible.

You are required to try **two different visualisation techniques and compare them**. Ideally one would be scalar method and one a vector method. Glyphs are perhaps the most obvious for unstructured data but you could also use splatting, streamlines, streamtubes or volume rendering. Consider also the use of implicit modelling, colour mapping and other techniques from lectures in your visualisation. You can also use filtering to pick out salient features of the data.

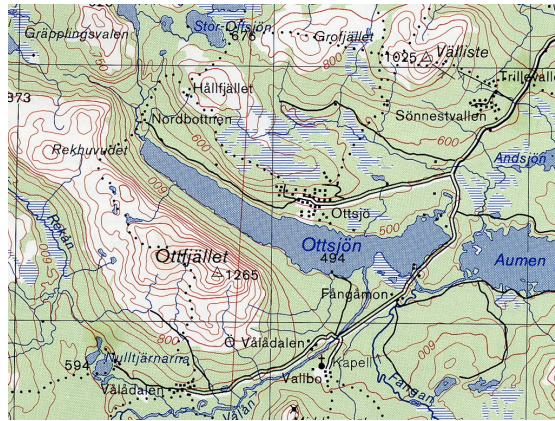
### GPS Data & Map Image

You will need to read the data into vtk. To make things simpler a vtk dataset object called `glider_gps.vtk` is also provided. This can be read in using a `vtkDataSetReader` object. It is polydata that contains glider positions, speeds and vector directions for each data point. These data are the raw recorded points, no re-sampling in time has been performed.

There is a map image (`glider_map.jpg`) also provided that you can use to texture map the ground beneath the glider and relate the position of the lift to the ground topography.

The coordinates of the corners of the map image in the Swedish grid are:

- Top left: 1349340 7022573      Bottom right: 1371835 7006362
- Top right: 1371573 7022967      Bottom left: 1349602 7005969



If you have time you could also digitise a few spot heights off the map and make a 3D landscape for the glider to fly over.

## Requirements & Submission

For this assignment, you are required to submit the following (with marks awarded as shown):

- **A visualisation using your first method (30%):**
- **A visualisation using your second method (30%):** *(must be significantly different from 1)*
  - **For both:** a presentation of your visualisation method with 2 representative JPEG still images, an MPEG movie (of an animation or interaction) and a working (on DICE) demonstration program that can be run to show your visualisation.
- **Code (20%):** Commented TCL script(s), Java or C++ code(s) for both visualisation methods. The TCL script should be usable in the form `vtk scriptname.tcl` and read all data/texture files from the current directory (assume file names as per web). If you use Java or C++ you **must** also submit a binary that works under DICE with instructions on how to run it.
- **A README file (20%):** containing anything not obvious from the comments in the source and details of how to run your scripts (e.g. “for part 1 run `vtk fileX.tcl`”). Additionally a brief (< 500 words) description of what your visualisation methods show, an evaluation of their relative merits for the visualisation task in hand and future improvements you would make given more time. You will get marks for understanding and appreciating the problem, even if you haven’t managed to complete the implementation.

**DO NOT submit more than 2 movie files and ensure all movies are no longer than 1 minute in length.** JPEG still images can be captured from VTK itself or using a screen/window capture tool.

*Submission procedure:*

Place your files in a single directory, and call the informatics electronic submission script for visualisation as follows (man submit for further details):

```
UG4 :      submit cs4 vis-4 cw2 your_directory_name
```

Submission deadline for this part is **Wednesday 9nd April at 5pm**. The usual lateness penalties will apply. *This part carries 50% of the final practical mark for the course.*

**Remember to leave enough time to test your program on the DICE machines.**

**Please ensure your submission complies with the school policy on plagiarism:**

<http://www.inf.ed.ac.uk/teaching/plagiarism.html> A failed assignment will not ruin your career but a plagiarised one may do. Acknowledge all other sources of code you use in your submission.

**Taku Komura 29/02/08 (based on previous exercise by Gordon Watson)**