Malte Schwarzkopf, St John's College, ms705@cam.ac.uk

Proteus – Project Progress Report – Week 12

Summary

Source code restructuring to deal with increased code base size; implementation of the approximation of the line integral for edge snapping. General bugfixing and cleanup, added interactive line and node placement.

What I intended to do

- Actually implement line snapping according to the method discussed in the final supervision of term (use summation of gradient magnitudes along projected line as approximation to the line integral)
- Implement annotation copying and linear interpolation between frames
- If time permits, try to experimentally integrate the KLT tracker for proper feature tracking between frames

What I ended up doing

- Restructured source code into hierarchy of directories to make it more manageable. Changed makefiles and workspace scripts accordingly.
- Implemented rudimentary snapping by having nodes snap to the point of maximum gradient in a 10x10 pixel window
- Implemented the line integral approximation using a summation of the gradient magnitude values along the line (based on Bresenham's line drawing algorithm), normalised by the line length
- Changed the GUI to display an ad-hoc projection of the line/node that is about to be placed
- General attempt to catch up after delays due to the supervision workload at the end of term and the week lost to writing my PhD research proposal.

Where I am in the timetable

It turned out that the PhD application process was a lot more time consuming than I had expected; getting ill at the same time certainly didn't help. Consequently I didn't do very much project work over the last week, and am now about 3 weeks behind on the time table. I hope to be able to catch up over the holidays, or alternatively I will have to use the catchup time in February/March to make up the time lost.

Problems/Issues encountered

The GUI code to display the projected line turned out to be quite fiddly, as did the implementation of various other GUI features. The edge snapping as implemented does not always work perfectly; occasionally, especially with "dark" areas (i.e. small gradient values that originally corresponded to high-magnitude negative numbers), a projected line that crosses noise gets a higher rating than the correct line. A solution for this has been suggsted, but its implementation is delayed until later as it is non-critical at this point.

What I intend to do

- Finish the last bits of the edge snapping implementation
- Get started on code to compute the fundamental matrix and perform the first steps of the structure-from-motion process
- If time permits, try to experimentally integrate the KLT tracker for proper feature tracking between frames (carried over from before)
- Read up on more theory behind the fundamental matrix and camera matrix computations
- Feature copying and linear interpolation between frames are still missing

Further remarks

I unfortunately managed to leave my copy of the Hartly & Zisserman book in Cambridge – I have ordered a new one, but it will take a few days to deliver, so that any work based on that is stalled until then.