

Proteus – Project Progress Report – Week 7

Summary

This week was spent consolidating and extending the work on the graphical annotation interface. Some more work on it remains to be done before line-snapping can be attempted as the next step.

What I intended to do

- Finish basic annotation framework, have per-frame annotation working
- Sort out annotation overdrawing bug (conceptual problem with deleting annotations)
- Add annotation IDs (integers) and facility to display them
- Change annotation drawing code so that it draws from the underlying data structure rather than on mouse events
- Change mouse interaction code to mutate the annotation data structure
- Start looking at interpolation and line snapping

What I ended up doing

- Basic annotation framework is done, annotations are stored and displayed per-frame
- Fixed annotation overdrawing bug, fixed z-ordering of annotations
- Added annotation IDs that can optionally be displayed on the GUI as an overlay
- Annotations are drawn from the underlying data structure rather than on mouse events; mouse events mutate the annotation data structure
- Added a zooming feature to allow more precise annotation by zooming in

Where I am in the timetable

About a week behind; I finished last week's milestone (plus a bit more) at the end of this week. This is not too surprising given the time in term and my current supervision workload, but I should aim to try and catch up this week.

Problems/Issues encountered

None really, except for a general inability to stretch the day to have more than 24 hours :)

What I intend to do

- Change the data structure representing annotations to support explicit edges
- Change the mouse interaction semantics to allow live update of edge position before setting the end point (and point creation on double click)
- Implement convolution with kernel for entire image, write output to file, compare the reference output
- Implement mechanisms for copying annotations between frames (no interpolation yet, just straight copying)
- If enough time, implement edge detection around current mouse position/around current projected line
- In order to be able to do the above, read papers on Shi-Tomasi feature tracker, investigate line integral calculation and mean-squared error calculation

Further remarks

None.