

Proteus – Project Progress Report – Week 14

Summary

Integration of the KLT feature tracker, multithreaded decoding-tracking implementation, identified and fixed sign error bug with computation of F , further reading on the next steps after computation of F .

What I intended to do

- Continue implementation of the fundamental matrix computation, investigate the sign error on the first and second columns
- Add normalisation for fundamental matrix
- Investigate how to implement camera matrix computation
- If time permits, try to experimentally integrate the KLT tracker for proper feature tracking between frames (carried over from before)
- Read the chapter on camera matrix computation and point extraction (triangulation) in Hartley & Zisserman

What I ended up doing

- Found and fixed the bug that caused the sign flip on the fundamental matrix after the rank constraint being applied: When computing $U * \text{diag}(3) * V^T$, the `transform()` function cannot be applied as part of a larger expression due to the way the temporary result is treated.
- Integrated the KLT feature tracker: Interfacing with the code, conversion between `KLT_FeatureList` and `AnnotationList` data types.
- Added a tracker thread that is spawned on-demand when using the KLT feature tracker – it gets decoded frames provided by the decoder thread and then identifies/tracks features on them and returns the resulting features to the main thread so that it can convert them into annotations.
- Designed and implemented the automated tracking system.
- Reading: KLT documentation / reference manual / tutorials.
- Reading: Hartley & Zisserman, chapter 10 and 11

Where I am in the timetable

Even though significant progress was made this week, I am still 2-3 weeks behind on the timetable. This is mainly due to delays in the core implementation; I spent a number of days

on integrating the KLT tracker this week as I was stalled on core work. Ongoing revision in parallel with project work also limited the amount of time I had to spend on project work. Focus for next week will be to make significant progress on the core implementation of the structure-from-motion process.

Problems/Issues encountered

The multithreaded implementation of automated feature tracking using KLT turned out to be quite tricky in order to sequentialise things as it is required and a lot of debugging effort went into getting this to work in a thread-safe manner. Further issues were encountered with the fundamental matrix computation, although the most prominent one is now fixed, so work should be able to continue making progress on this now.

What I intend to do

- Finish off the fundamental matrix computation
- Add normalisation for points and denormalisation for fundamental matrix
- Implement camera matrix computation
- If time permits, try to further improve the KLT implementation and add a better conversion mechanism that preserves feature relationships so that tracked features can be used as input to the structure-from-motion process
- Investigate the methods for point extraction / triangulation and get started on implementing a simple one in order to get first results out
- Hopefully I should be getting out the first set of 3D points by the end of the week, so that I have a very simple structure-from-motion process working before the start of term

Further remarks

The current implementation of the structure-from-motion process is a very minimal one (normalised 8-point algorithm, camera matrix computation according to formula, simple triangulation method). The results will have to be evaluated to see if they are satisfactory - most likely, more advanced techniques will have to be considered. Additionally, the current implementation only considers two (adjacent) frames. While it is trivial to extend it to use non-adjacent frames, the consideration of more than two frames in the sequence will require significant modifications (but should hopefully improve the results).