CSC 262 **Independent Project** **Fall 2026**

Assigned: Thursday 15 October

Deadlines Project Selection: 5 PM Friday 6 November

Project Proposal: 5 PM Sunday 15 November

Progress Report: 5 PM Wednesday 2 December

Final Report: 5 PM Wednesday 16 December

Presentation: 9 AM Tuesday 15 December

Collaboration: You are *strongly* encouraged to work in pairs, but you may work individually. You must get permission to work in a group of three, which will be allowable only if there is an odd number out.

1 Project

By now we have examined a variety of problems in computer vision, and there are still a few more to come. With this grounding, it is time for you to study a topic more in-depth or explore something new that we have not covered yet. You will have four weeks to learn some new techniques, develop an implementation, and test it on data of your choosing. Start early and don't wait until the last minute! Computers are notoriously bad at seeing, and it will be your job to take the time to make them better.

Your project should focus on implementing a substantial portion of a technique for solving some computer vision problem. Select something that seems small and then set intermediate goals. For the most part, you should plan to develop all of the code you need yourself (with the exception of the tools built-in to Matlab). If you wish to work on one aspect of a system that is deeper in a pipeline and depends on a tool you have not authored, you must get prior approval from me. (Do not let the need for approval dissuade you, it is simply a means to make sure your own work is substantial.)

The overall project is broken down into the following stages.

Selection: A few sentences indicating the task and method you hope to pursue.

Proposal: A detailed report describing the task, development milestones, evaluation data, metrics, and results of preliminary experiments.

Progress: A follow-up report describing accomplishments, challenges (past and present), next actions, and help needed.

Report: A final write-up explaining, exploring, and summarizing your project work (and accompanying code).

Presentation: An exciting show-and-tell for all your hard work.

These stages are detailed in the following sections.

2 Selection

Deliverable: Send me an e-mail with your group member's names and two or three sentences identifying the task and method you plan to implement.

Your first task it to propose a project. I will include some examples, but they are primarily intended to get your thinking started. You should strive to find something that is interesting you. Here are some suggestions for finding a project area:

- Read over our many textbooks and look for interesting applications that we have not covered in great depth; several more books are on reserve for this course in the library. You can also follow some of the references to the technical literature given in the texts.
- Scan the proceedings of recent computer vision conferences for interesting work you might be able to replicate (or do a bare-bones approximation of). Some examples include:
  - CVPR (Computer Vision and Pattern Recognition)
  - ICCV (International Conference on Computer Vision)
  - ECCV (European Conference on Computer Vision)
  - BMVC (British Machine Vision Conference)

  You can typically find a list of all the papers from a conference at the conference web site for a particular year (e.g. web search "CVPR 2018" and look for "Program"). Most all CS conference papers are posted by their authors on the web and can be found with additional targeted sleuthing (via GoogleScholar). If you cannot find it, see your local librarian about acquiring a copy.
- If you have a vague idea but are not sure where to go, I am happy to talk with you and discuss possible avenues. This will likely require follow-up work on your part, but I am glad to get you started.

3 Proposal

Deliverable: Submit a PDF writeup on Gradescope.

Once you have found a topic, you must decide what exactly it is you will do. Be specific! Your proposal should address several primary concerns.

- What algorithm will you implement?
- What will you need to read/study/examine to give yourself sufficient context and background?
- What (type(s) of data) will your system run on?

- How will you evaluate your results (quantitatively and qualitatively)?

Next you should think about how to break your chosen problem into smaller pieces that you can test or demonstrate along the way. In writing your proposal, you must outline four benchmarks or milestones for you to structure your work around.

The first should be dead simple; so simple that you can (and will) complete it for your proposal to indicate the viability of your project and data. You can think of this milestone as the "*If we can do this, we should at least pass the project with a D*" benchmark.

The next milestone should also be relatively easy, something that you know you can do and that provides the core foundation or ground work for your system. You can think of this milestone as the "*If we don't get this, we really don't deserve to get a C on the project*" benchmark, and it should be completed for your progress report.

The third milestone should be more substantial, an important piece of the system that gets you most of the way there. Perhaps your system is implemented at this point but only works robustly under some stringent assumptions. Maybe the last piece that would tie everything together still needs to be done. You can think of this milestone as the *"If we get this, we should get at least a B on the project*" benchmark, and work should be underway for your progress report and completed by the final report.

Finally, you should outline a task that, if completed, you would be proud to show your friends, your parents, and sell on the app store for $4.99 (ok, maybe $0.99). Perhaps your system works reliably on a large image database. Maybe you finished replicating the results in the winner of last year's CVPR Best Paper award. You can think of this milestone as the *"If we get this, we are definitely getting an A on the project*" benchmark.

Your benchmarks are mostly for your benefit in helping you to develop your project and structure your work. Thus, you should reflect on them carefully and critically. I will take them as suggestions and give you feedback about whether they are at the right level, but I may or may not use them in assessing your final grade.

Finally, you are also encouraged to include in your proposal a scaffolded hierarchy of data that you will use to test your system. That is, perhaps you want to test with some easy, synthetic images first, moving on to more constrained images, and finally testing your system in harder, more general situations. There are a variety of data sets for many computer vision problems available on the web, and you can also capture images yourself from the cameras in our lab or your own digital camera.

Your proposal should be 2–3 pages, include details of:

- what you plan to implement (with appropriate references),
- your four (or more) clear benchmarks,
- what data you will use to test your system,
- how you plan to evaluate it, and
- first-milestone feasibility results.

You should also clearly indicate what you will need to learn about along the way, and what the biggest risks are.

Read the Proposal Rubric for further details about what is expected.

4 Progress Report

Deliverables: Submit a PDF writeup on Gradescope. Orally present your progress to the class.

By week 13, you will have had almost three full weeks to work on your project. It is hoped that by this point you have completed your second milestone and are at least halfway to the third (possibly more). This is a good point to report on your progress to everyone. In 2-3 pages, you should

- recapitulate your goal,
- describe what you have accomplished so far (including any figures as appropriate),
- share what challenges you faced and overcame,
- explain what hurdles you currently need help with, and
- say what you plan to do next.

On the same day, each project group will take 5–10 minutes in class (depending on the number of groups) to give an oral report of the same substance, giving others an opportunity to ask questions, comment, and make suggestions on your current problems.

5 Final Report

Deliverables: Submit a PDF writeup and complete code on Gradescope.

The final report is due during finals week. It should explain the algorithm you implemented, general comments on how your implementation is organized, and your approach to setting any important parameters of the system. Your report should demonstrate and evaluate your system's operation with tables and/or figures as appropriate. You may also wish to briefly report on shortcomings, difficulties still faced, or other work left to do. Tell a complete story with an introduction that provides context and a conclusion that unites all the ideas and issues you have covered. In short, it should read like a concise technical report.

In addition, you must submit all of the code necessary to run your system and any data you used in your evaluation. The same documentation standards used in labs apply here; it is especially important to clearly document the inputs and outputs to your functions. Include a short `README.m` script with a documented example of how to run your system on some (or all) of the data you provided.

Your report should be written to the College's highest standards of writing. Therefore, you should make an appointment for week fourteen with the writing lab ***right now***. Even if you do not have all of your results by that time, you should have an outline for those sections and a nearly complete paper.

During our final exam period, each group will take 10–15 minutes (depending on the number of groups) to give a presentation of their results with at least 3–5 minutes for questions from the audience. Plan to hit the high points: what is the general method, how

you did it, and (most exciting) what are the results. Assuming you use slides to present your work, 7–9 slides should suffice. Well-rehearsed live demonstrations are also strongly encouraged.

6 Assessment

The proposal will count toward 20% of the overall project grade. Likewise, the progress report (written and oral together) will also contribute 20% of the overall project grade. The remaining 60% will be assessed roughly equally on the quality of your report and the substance of your implementation, with a portion based on the clarity of your code as well.

See the project rubrics for more details on expectations and assessment.

7 Examples

The following is a very short list of project ideas to get you started. You are highly encouraged to think beyond these and do something interesting to you. If you have questions about any of the vague suggestions below, do a little research and/or visit my office hours to discuss them.

- Complete a multi-scale, rotation aligned MOPs descriptor and matching algorithm with an image stitcher that uses optimal affine transformations
- Match features across multiple frames to build a robust orthographic structure-from-motion system
- Smoothly animate from one view of a 3-D scene to another with view interpolation (see Szeliski 7.2.3)
- Build a stereo vision system that does rectification and reconstruction to build a 3-D model. (We could 3-D print your face!)
- Build a texture representation to classify items from a large texture database and perform texture-based image segmentation
- Use color and shape features (or others) to segment and recognize images of fruits for an automated grocery checkout
- Learn to use a simple parametric binary (two-class) classifier to train an X detector, where $X \in \{\text{face}, \text{text}, \text{car}, \ldots\}$; make it scale invariant
- Implement an "image scissors" or outline tracker using active contours (aka snakes)
- Learn a model for doing image super-resolution
- Implement a robust optical flow algorithm for tracking

Copyright ©2010, 2012, 2015, 2019 Jerod Weinman.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 4.0 International License.