

CS 461 Computer Vision: Final Project

Prof. Gregory D. Hager
Department of Computer Science

Due: Dec. 11, 2008 (Final Demonstration)
Location: TBA
Time: TBA

The objective of the final project is to make use of what you have learned during this course, and to develop some creative new applications of computer vision. The project will be demo'ed for your fellow classmates and the Johns Hopkins community

You will work in groups of one or two. I would like most of you to work in one of two project areas:

- The photosorter project
- The interactive display project

Both of these are further described further below.

Grading will be based on the completeness of the project, the clarity of the writeup, and the level of functionality attempted. Further specifics on grading are given for each project below.

The Photosorter Project

Over the last decade, digital cameras have overtaken film. As a result, there are ever larger consumer archives of photos as well as large online repositories such as Flickr, Facebook, and Picasa. Facebook is currently the largest photo online storage site, with over 2-3 Terabytes of photos uploaded daily, and 15 billion photos served served each day(<http://www.facebook.com/press/info.php?statistics>). (You should show a demo of Picasa's ability to automatically tagging <http://www.youtube.com/watch?v=teeGF-w5Cpw> and sorting by color).

Digital cameras have tended to alter people's photo-taking behavior. Since images are now "free," most people take more pictures. They also take many duplicates in the hope of getting one "really good" shot. But, as these photos come off the camera, they tend to pile up on the drive without and useful sorting or filtering.

This is where you come in. I'd like you to look into developing some photo sorting tools. While the idea is to focus on personal tools, in fact some of what you develop could be used to help organized the gazillions of photos on public sites as well.

The project ground rules are as follows:

- You will be supplied with a folder of digital images. Our test sets will be lossless, but obviously it will be better if your techniques work with lossy (e.g. jpg) images as well.
- Your job will be to detect commonalities among photos, and to create "clusters" of similar photos. Optionally, you can also apply labels to the clustered photos.
- We will supply a test set of photos for your to work with. We will also create a "hidden" evaluation set that you submit for algorithm testing. We will tell you how well your algorithm works on this evaluation set, but we won't show you the images.
- You will also create an application interface that can be demonstrated. It is up to you whether to use Matlab or to use OpenCV in your application.

There are three levels of grading for this project:

- **Baseline:** The baseline project is to detect duplicate photos and to group them. Duplicates may differ in small details, but the overall composition is the same. A project that does this well will get 80% credit.
- **Preferred:** The preferred project will perform baseline operations, but also be able to detect common portions of images and group based on those. It will detect that scenes contain the same people or objects (but might differ in composition) and will group based on those objects. This does not need to be a general-purpose functionality. That is, it suffices, for example, to recognize some small set of people, and it is ok to assume the people have not altered their appearance. A project that exhibits these capabilities would get 100% credit.
- **Deluxe:** A deluxe project would add additional functionality, a nice user interface, interactivity, or interfaces to other tools (Facebook, Flickr, Photoshop etc.) Credit for deluxe items will be given on a case-by-case basis.

Some References — reference some of the recent photo-sorting papers

The Interactive Display Project

The use of live video to provide interactive displays has been a topic for several years in the research community, and is now also gaining ground in the commercial world (see references below).

This project will allow you to explore the development of interactive display ideas ranging from intelligent billboards to interactive digital art. As such, it is more open-ended than the project above. It also will demand real-time performance, so OpenCV is the preferred approach.

For this project, I will provide you with an OpenCV example program to start with. This program does a very simple optical flow algorithm to “animate” a screen when an object is at a rough distance. I will also provide access to two 50” display monitors equipped with stereo cameras and computing hardware capable of performing “interesting” vision operations.

The project ground rules are as follows:

- The focus of the project should be on computer vision, not on graphics. A really good looking project that doesn’t contain interesting vision work will not be graded as well as a clunky-looking project with very good vision algorithms.
- You are welcome to make some assumptions about the environment (e.g. a black background) if you want. However, the more assumptions necessary, the weaker the project.
- The project does not have to be completely real-time, but should at least be reasonably interactive — waiting more than 1/2 sec for a response starts to look very sluggish.

The grading for this project will be a little more subjective compared to the project above, but there is also a lot more room for fun and creativity. As above there are three levels of grading for this project:

- **Baseline:** The baseline project is to detect motion and react to it in an interesting way. It should either make use of both cameras in one form or another, or it should significantly enhance the baseline system that has been supplied (e.g. by segmenting and tracking the motion). Such a project will get 80% credit.
- **Preferred:** The preferred project will perform baseline operations, but also be able to detect and track people in some form. For example, it might detect and track motion, then run a face-finder to decide if someone is looking at the display, and change the display if this is the case. The system must have some 3D awareness by using both cameras. A preferred project will get 100% credit.
- **Deluxe:** A deluxe project would add additional functionality, better graphics, or other enhancements (e.g. sound output).

Some References — reference some of the recent digital art or gesture papers

Schedule

You will be required to report on your progress each week until the final demo. These checkpoints are as follows:

- Nov. 11: An approx. one-page project description that includes: 1) Team members; 2) Project description; 3) Proposed final functionality; 4) Project components and team member assignments; and 5) Rough project timeline.

Nov. 18: By this date, you should have a basic skeleton app up and running that minimally reads in data and puts something on the screen. You should submit an update to your one-page description indicating what you've accomplished, and what (if any) problems you've run into.

Nov. 25: Mid-term demonstration! You should have some basic functionality running by this time, or at least be close to it. You should again update your project description with progress and unanticipated problems. You will also be required demonstrate/discuss your project with me or Raphael in the coming week.

Dec. 4: Final one-page progress update. You should be nearing completion. This will be your last chance to ask for help!

Dec. 11: Final projection writeup (described below) and final demonstration are due on this date.

Report Format and Grading

Your final report should document the following:

- Your initial project goals.
- The goals you achieved.
- The computer vision methods you used to achieve these goals.
- Known limitations and possible future extensions.
- The two or three things you learned during this project.
- Advice you would give to next year's vision students.

GRADE BREAKDOWN (tentative):

Concept - 20%
Implementation - 20%
Intermediate submissions - 20%
Final Application - 20%
Final Paper - 20%

Additional Resources

TBD