CS691A: Computer Vision. Project ideas

Erik G. Learned-Miller
Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA 01003

November 7, 2011

Abstract

This document contains short description of project ideas. Some of them have more detail than others. You can pick a project which is not on this list, or modify one of the ideas here. This is just some ideas to get you going.

1 Camera within a camera: Building a hybrid pinhole camera

NOTE: Unfortunately I cannot provide materials for this project. In particular, I do not have a digital camera that you can use for this project, but you should be able to recover your own digital camera (if you have one) once you disassemble your final project.

Build a pinhole camera, and use a digital camera inside it to take pictures. As discussed in class, the interesting thing about this camera is that you can take pictures with infinite depth of field.

You can find a plan for building a pinhole camera with a camera inside at this web site:

`http://www.foundphotography.com/PhotoThoughts/archives/2005/04/pinhole_camera_2.html`

Don't be too disappointed if the pictures don't come out crystal clear. It is not necessarily easy to get sharp clear pictures.

2 Near infrared photography

2.1 Project Idea 1

WARNING: You may need to order some (inexpensive) parts for this project, so if you want to do it, you need to get started right away.

Modify a standard web camera to take cameras that include an infrared component.

You can follow the basic plan outlined at this web site:
`http://www.alexschreyer.net/programming/explorations-in-the-infrared-spectrum/`

2.2 Project Idea 2

Note that you can also do infrared photography with a camera phone like the iPhone. Check out this link:

`http://www.flickr.com/photos/matt_brock/sets/72157624198109814/`

If you have an idea for a project like this, propose it to me. One interesting thing you could do is try to form images by subtracting images taken with an IR filter and with no filter to try to get narrow band infrared images. This would require image registration and a few other elements.

3 Face identification (challenging)

Read this paper:

http://lear.inrialpes.fr/pubs/2009/GVS09/verbeek09iccv2.pdf

and implement the face recognition algorithm it describes.

4 Digit classification using congealing

Make a full-scale digit classification system using congealing. You can use the code from your previous assignment if you want. The steps would be something like:

1. Get digits from each category of digit (0-9). I can get these for you.
2. Congeal each class of digits, storing a "funnel" model for each digit.
3. Build a distribution over transforms from your congealing results. (I can help you with this.)
4. To test a new digit, try congealing it with each funnel. When you're done, compare the congealed version of the digit, and the transform associated with it, to stored versions of the congealed training data. You could use nearest neighbor, for example.
5. Pick the class with the best congealed character and transform.

5 Panoramas with image stitching

Write software that can be used to stitch together images into panoramas. You should be able to take some number of photos of a scene (say 10), and automatically put them together into one large image by finding point correspondences.

While theoretically you could use gradient descent methods to achieve this, it would probably be better to use something like SIFT features and RANSAC, as described in Szeliski's computer vision book. If that's not enough detail to get you started, ask me for help with this project.

6 Build a material classifier

Implement the ideas discussed in class about building a classifier for materials such as chrome, wood, pearls, or matte surfaces.

If you do this project, I would ask that you develop a training set and a test set, where the training set shows images of objects made of various materials in certain lighting conditions. The test set would then show different objects in different lighting conditions, but using the same basic set of materials.

The goal would be to come up with features that gave the highest accuracy on the training set, and then evaluate the ability of the algorithm on the test test. Ask me for more details if you're interested.

7 Face Recognizer from SIFT features

Use SIFT features and a "bag of words" model to build a face recognizer.
There are many ways to do this, but here is one possibility.

1. Assume your goal is to build a face recognizer that distinguishes you from everyone else. This could be used to control access to your computer, for example.
2. Take a large number of pictures of yourself under various conditions (say, 100 pictures).
3. Take a large number of pictures of other people (say, 300 pictures).
4. Find code online to compute the SIFT features for an image.
5. Take all of the detected SIFT features for all of the images and cluster them into some number of clusters (say 20). You can do this using k-means clustering in matlab.
6. Now, every SIFT feature will be closest to some cluster, so you can replace a SIFT feature (128 numbers) with the number of cluster it is closest to (in the L2 sense).
7. Every image can now be converted into a probability distribution over SIFT cluster values in the following way.
   (a) Compute all the SIFT features for an image.
   (b) Convert those SIFT features into cluster IDs.
   (c) Compute the percentage of each cluster ID that is present in the image. This will give you a set of 20 numbers that add up to 1. This can be your "feature vector" representing an image.
8. Use a support vector machine with this feature representation to train a "you versus the rest of the world" classifier.