Review for Final
CMSC 426 – Fall 2012

General comments

There are five key technical ideas in this class. The goal of the class is for you to master these ideas and to see how they can be used to solve problems in vision. So hopefully the final exam will test both your understanding of basic problems in vision and your mastery of these techniques, and how they can be used to solve vision problems. Below is an outline of the class. I'm including some sample problems that you can work if you want practice on some of these problems. The practice problems from the midterm are also appropriate.

Correlation/Convolution and Multiscale

The first key idea of the course is to understand correlation and convolution.

1. You should understand how to perform these operations.
2. You should also understand some of their basic properties. Convolution is associative, and this will allow you to combine operations, or sometimes to split operations. For example, we can combine smoothing and a derivative filter into one operation. Or we can break smoothing with a box filter into two convolutions with smaller, 1D filters.
3. You should know how to construct a filter so that you can use convolution and correlation to perform a few different operations, including smoothing an image or taking a derivative.
4. You should understand how to construct and use multiscale representations of images. You should understand how to detect blobs as local extrema in images as they are smoothed.
5. * The Fourier series representation of functions might figure in a challenge problem.

SAMPLE PROBLEMS

1) Compute the convolution of the 1D image: (0,0,1,1,2,2,3,3,4,4,4,4) with the convolution kernel: (1/4,1/2,1/4). Treat the boundary in any reasonable way.

Let's use a repeating boundary. We get:
(0,¼,3/4,5/4, 7/4, 9/4,11/4,13/4,15/4,16/4,16/4,16/4).

2) Suppose I convolve an image first with the filter [1; 0; -1] and then with the filter [-1 0 1]. Give a single, 2D filter that will accomplish the same thing. What is the mathematical operation this filter approximates?

To get a single filter that combines both, we can convolve the first filter with the second one. This gives us: [-1 0 2 0 -1]. The first filter computes the derivative times 2. The *second filter computes the derivative times -2. Together, these compute a second derivative times -4.*

The Image Gradient

The image gradient is perhaps the most fundamental way we have of representing images. It captures how the image changes. You should understand:

1. How to compute the gradient.
2. Given a gradient, find the magnitude and direction of the gradient.
3. Use the gradient to find boundaries. This includes understanding whether a gradient is a local maximum in the direction of the gradient (eg., Canny edge detection).
4. Use the gradient and the temporal derivative of an image to compute the optical flow.

SAMPLE PROBLEMS

1) Consider the following 1D image:
(0 0 …. 0 3 3 3 3 3 3 3 3 3 3 3 3 3 0 0 0 10 10 10 0 0 0 0 …)
Where would you expect a 1D edge detector to find edges?

It could find an edge every time there is a sharp change in intensity, in this case between 0 and 3 or 0 and 10.

An edge detector has a parameter that allows it to ignore weak edges. As we increase this parameter, which edges disappear first?

The transition from 0 to 3 (or 3 to 0) produces a smaller derivative than the transition from 0 to 10, so that disappears first.

If we smooth the image before detecting edges, how would you expect this to affect the position and number of the edges?

As the image is smoothed the region of 3s and the region of 10s can blend together, and the edges between them can disappear.

2) For the point 16, in bold face, find the gradient. Give the direction and magnitude of the gradient.

5	8	11	14	17	20
6	9	12	15	18	21
7	10	13	**16**	19	22
8	11	14	17	20	23
9	12	15	18	21	24
10	13	16	19	22	25

The x derivative is (19-13)/2 and the y derivative is (17-15)/2, so the gradient is (3,1). The magnitude is sqrt(10) and the direction is (3,1)/sqrt(10).

3) I is an image described by the equation I(x,y) = x*x + y*y. J is the same image translated 1 pixel in the positive x direction. Suppose I and J are consecutive images in a motion sequence. Write the optical flow equation in its general form. Using this equation, give the equation for a line in the second image on which we expect to find the point that was at (5,5) in the first image. Perform the same computation for a point at (5,0). Combine these two equations, and show that this yields the translation of the image.

The gradient of this image is (2x,2y). So the gradient at (5,5) is (10,10). If we translate one pixel in the x direction we will compute a time derivative of (x-1)(x-1)+y*y – (x*x+y*y) = 1-2x. So at(5,5) this will be -9. So the optical flow equation will give us 0 = -9 + (10,10).(u,v), or 9 = 5u+5v.*

Performing the same computation at (5,0) we get:
0 = -9 + (10,0).(u,v), or 9 = 10u. Combining these together we get u = 9/10, and v = 0. This is approximately right; the discrepancy is

Histograms and Statistical Modeling

We have looked a lot at histograms. You should realize that when we normalize a histogram so that it sums to one, we can treat it as a probability distribution.
There are two key parts to probabilistic modeling. First, you must use some prior knowledge to construct a statistical model. Second, you must use this model, typically to compare distributions, classify new data or to generate new examples.

1. Building statistical models
   a. Given some independent samples, use a histogram to build a distribution for them.
   b. Kernel Density Estimation: We can smooth a histogram to get a less noisy estimate of the distribution.
   c. Two particularly relevant examples of histograms are histograms of image intensities, and SIFT descriptors. You should be familiar with these. You should also know about histogram equalization.
2. Comparing distributions using:
   a. SSD
   b. Chi-Squared
3. Classifying new examples: in background subtraction, we classify a pixel as foreground or background.

SAMPLE PROBLEMS

1 Suppose we are performing background subtraction. For one particular pixel, we see the following intensities: 1 1 2 2 3 3 3 3 1 1 2 2 3 3 3 3 1 1…. If we use a

histogram of these values and no Gaussian smoothing what would you estimate is the probability that the next pixel would have an intensity of 2?

This is just the fraction of pixels we've seen that had the value 2, which is ¼.

2 Consider the following statement: "When performing background subtraction with kernel density estimation, we should use a smaller and smaller value for sigma as the size of our training set increases." Do you think this is true? Why or why not?

Yes, the more data you have, the less you have to smooth it. Try using Matlab to generate random integers from 1-10, and make a histogram, normalized so all the entries add up to 1. Do this with 10 random numbers, 100, 1,000 and 1,000,000. See how much smoother the histograms get.

Optimization

Optimization is a very large area. There is a wide array of optimization methods that have been applied to vision problems. You should understand the few of these methods that we have studied, and how they can be used to solve vision problems.

1. Structure of optimization
    a. Define a set of possible solutions.
    b. Define a cost function that defines the value of each solution.
    c. Find a search algorithm for exploring solutions, picking the best one found.
2. Optimization methods
    a. Brute-force search. For many problems, at least as a baseline, the starting point is to try every possible solution.
    b. Dynamic Programming.
    c. Graph cuts
    d. K-means algorithm.
    e. RANSAC
3. Applications
    a. Finding boundaries.
    b. Clustering pixels.
    c. Stereo matching
    d. Image matching and Mosaicing.

SAMPLE PROBLEMS

1. **Kmeans:** Suppose we have points at the locations: (1,3) (4,7) (2,9) (2,2) (3,6), and we pick centers at (1,4) (4,4). Which points will be assigned to each center? What will be the location of the new centers?

Points closest to (1,4) will be (1,3), (2,9) and (2,2). The new center will be at their average, (5/3, 14/3). The others points, (4,7) and (3,6) will form a second cluster, which has a new center at (7/2, 13/2).

Suppose we have five points and two centers, as above. Place an upper bound on the number of possible iterations that k-means can perform before it converges.

A simple bound is that there are only 2^5=32 possible ways of clustering 5 points into 2 clusters, so we can't have more iterations than that. A better bound could be obtained in a more complex way; might be a good challenge problem.

2. Suppose we are performing stereo matching, and we want to add a penalty, D, which we have to pay whenever there are any changes in disparity. Explain how this would change the dynamic programming stereo algorithm.

3D Geometry

This all comes down to forming linear subspaces and intersecting them. For example, given two points, find the line that they form. Or given three points, find the plane that they form. Or intersect a line with a plane, or two planes to find a line. With these tools, we can solve a variety of problems:

1. Perspective Projection: form a line from two points and intersect it with an image plane.
2. Locating a 3D point from its appearance in a 2D image: form a line from two points and intersect it with any other possible constraint.
3. Locating a 3D point from its appearance in two 2D images (stereo): form a line from two points twice, and intersect these lines. For a standard stereo set-up, this can be solved more simply using similar triangles.
4. Motion Flow. You should understand possible flow patterns for simple motions.
5. Epipolar geometry:
    a. Given a point in a 2D image, delimit its possible location in a second image (the epipolar constraint): form a plane with three points (two focal points and an image point) and intersect it with an image plane.
    b. You should also understand how to find the epipole, by forming a line between the two focal points and intersecting this with an image plane.
6. Image rectification: project an image plane onto a new image plane.
7. 2D Image Transformations using a Matrix
    a. Represent a 2D point as a vector.
    b. Represent a 2D translation using a matrix.
    c. Represent scaling using a matrix.
    d. Rotations are a bit more complicated. To be more explicit, you should know how to:
        i. Represent a rotation with a matrix.

ii. Determine whether a matrix represents a rotation.

e. Represent a 2D affine transformation

SAMPLE PROBLEMS

1) If using perspective projection, how can you tell whether a line in the world will project to a single point in the image?

If the line goes through the focal point, than all points on that line will project to the same image point.

2) Consider a camera in the standard position for perspective, with the image plane the z=1 plane, and the focal point at the origin. Consider a rectangle with corners at: (0,0,2), (1,1,2), (0,0,3), (1,1,3). Give its image with perspective projection. Prove that parallel lines in the world don't necessarily project to parallel image lines.

Using (x,y) = (X/Z,Y/Z)f, it's image is just found with corners at (0,0), (½ ½), (0,0), (1/3, 1/3).

This gives a trivial example in which parallel lines don't project to parallel image lines, since one line just projects to a point. For a less trivial example, take the parallel line segments going from (-10,10,2) to (-10, 10, 10) and going from (10,10,2) to (10,10,10). This produces line segments in the image going from (-5,5) to (-1,1) and from (5,5) to (5,1). These aren't parallel in the image.

3) Suppose we have two parallel lines. One goes through (0,0,2) to (0,0,3) and the other through (1,0,2) to (1,0,3). What will be their vanishing point? (assume the standard camera configuration).

As above, figure out where these lines appear in the image. Then take the intersection of these lines.

4) Suppose we have a camera with a focal point at (0,0,0) and an image plane of x=1. There is a point in the world with coordinates (X,Y,Z). Where will this appear in the image?

(X/Z, Y/Z)

5) Suppose we take two pictures. In the first, the camera has a focal point at (0,0,0) with an image plane of z=1. In the second, the focal point is at (0,1,1) and the image plane is z = 2. Where is the epipole in the second image? Give an example of conjugate epipolar lines in the two images. That is, give a pair of lines, one in each image, so that every point in line 1 matches some point in line 2.

The epipole is the intersection of the image plane and a line connecting the two focal points. The line connecting the two image points can be described as: (X,Y,Z) = t(0,1,1).

This intersects the z=2 plane at (0,2,2), and in the first image, it intersects the image plane at (0,1,1). Notice that these points will both have coordinates (0,1), since (0,2,2) actually has coordinates (0,1).

All epipolar lines go through these points. To find conjugate epipolar lines, let's pick a point in the world, like (5,5,5). This appears in the first image at (1,1) and in the second image at (5/4, 1). So we get conjugate epipolar lines if we make a line in the first image connecting (1,1) and (0,1), and make a line in the second image connecting (5/4,1) and (0,1). These happen to be the same horizontal line. What about other conjugate epipolar lines?

6) You have a camera with a focal point at (7,3,4) with an image plane of x+z=12. Suppose there is a scene point with coordinates (100,100,100). Where would this appear in the image plane?

We can get this by first forming a line between the focal point and the scene point. This is:
(X,Y,Z) = (7,3,4) + t(93,97,96)
Then we intersect this with X+Z = 12. We have:
X = 7 + 93t
Z = 4 + 96t
So we get:
7 + 93t + 4 + 96t = 12
This means:
189t = 1. t = 1/189.
So the point is in the image plane at:
(7,3,4) + (1/189)(93,97,96).

Equations

There are a few equations that are so important you should know them by heart, and understand how to use them. These include:

Chi-Squared

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{m=1}^{K}\frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

Correlation

$$F \circ I(x) = \sum_{i=-N}^{N} F(i)I(x+i)$$

$$F \circ I(x,y) = \sum_{j=-N}^{N}\sum_{i=-N}^{N} F(i,j)I(x+i, y+j)$$

CONVOLUTION:

$$F * I(x) = \sum_{i=-N}^{N} F(i) I(x-i) \qquad F * I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x-i, y-j)$$

$$\nabla J = (J_x, J_y) = \left(\frac{\partial J}{\partial x}, \frac{\partial J}{\partial y} \right) \quad \|\nabla J\| = \sqrt{J_x^2 + J_y^2} \text{ tells how fast image changes}$$

$$\frac{\nabla J}{\|\nabla J\|} \text{ is the direction of fastest change}$$

Translation, rotation and similarity transformation matrices

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \textit{translation}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \textit{rotation}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ -b & a & t \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \textit{similarity}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \textit{affine}$$

Perspective Projection

$$(x,y) = f(X/Z, Y/Z).$$

Disparity

$$Z = f \frac{T}{d}$$

Optical Flow

$$0 \approx I_t + \nabla I \cdot [u \ \ v]$$