Practice Midterm

CMSC 426

The midterm will cover material up to and including human perceptual grouping. Here are some topics it will be helpful to master for the midterm:

- Correlation
  - How do you perform correlation in 1D or 2D? What is a box filter? How do you create a Gaussian filter? How do you create filters to compute derivatives? Fourier transforms (only for challenge problems).
- Edge detection
  - Image gradients
    - How to compute the image gradient for a continuous or discrete image.
    - What is the direction and magnitude of a gradient?
    - How to compute the directional derivative of an image.
    - What does the gradient tell you about the direction in which the image changes most rapidly, and how rapidly it changes in that direction.
  - Non-maximum suppression
- Human perceptual grouping
  - Basic gestalt grouping cues.
- There will not be questions about Matlab coding.

In particular, you should understand a few equations:

1D and 2D correlation

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

1D and 2D Convolution

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

Definition of a gradient:

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

Formula for using the gradient to determine how the image changes as you move in a particular direction

$$I(x+\Delta\cos\theta, y+\Delta\sin\theta) - I(x,y) \approx \langle v, \nabla I\rangle \text{ where } v \equiv (\Delta\cos\theta, \Delta\sin\theta)$$

Definition of the partial derivative.

$$\frac{\partial I(x,y)}{\partial x} = \lim_{\Delta x\to 0}\frac{I(x+\Delta x, y) - I(x,y)}{\Delta x}$$

I'm asking you to "memorize" these formulas because I feel that if you really understand what is in the formula, you will have it memorized.

Below are some practice problems. These are not guaranteed to cover everything you need to know, but should be helpful.

Practice Problems

1. Show the result of convolving the kernel, *k*, with the image, *I*.

1/4	1/2	1/4

k

0	0	...	0	1	2	...	10	0	0	...

I

The output looks like:

0, 0, ..., 0, ¼, 1, 2, 3, 4, 5, 6, 7, 8, 9, 7 ¼, 2 ½, 0, 0 , 0...

2. Give an example that shows that correlation is not associative

Try combining the derivative filter with a smoothing filter.

To keep things as simple as possible, use [0, -1, 1] as the derivative filter, and [1/3, 1/3, 1/3] as the smoothing filter.

3. Suppose we have an image whose intensities are described by the equation $x^3-2xy+7$.
   a) What is the magnitude of the image gradient at the point (7,4)?

The gradient is ($3x^2$-2y, -2x). At (7,4) this is (138, -14). The magnitude of that is sqrt(147^2+16). If I asked a question like that on the test, the numbers would be kinder to you. But it is fine to just write the answer as I just did, without simplifying.

b) Suppose we move in the image to the point (7+δ, 4 + 2δ). Use the image gradient to predict what the intensity will be at this point.

138δ - 14δ=124δ.

c) **Challenge problem:** What is the actual intensity at this point. What does this tell you about the accuracy of your solution using the gradient?

If you work this out, you'll find it's the same as the answer you got for (b) plus some terms that are quadratic or cubic in δ. When δ is small, these terms become very small.

4. Consider the following, small image:

4	6	9	12	15
3	4	6	*9*	12
3	3	**4**	6	9
2	*3*	3	4	6
2	1	3	3	4

a) Compute the magnitude and the direction of the gradient for the central point (**4**).

Let's use filters [-1/2, 0, ½] and [-1/2; 0; ½] for the x and y derivatives. Then we get for the gradient (3/2, -3/2). The magnitude of this is sqrt(18)/2. The direction is the gradient divided by the magnitude (a unit vector in that direction). Or you could say the direction is a diagonal vector pointing towards the 9.

b) Would the Canny edge detector consider this point to be an edge? Explain why or why not in detail, describing what conditions are needed to decide this.

You need to compare the gradient magnitude to the gradient magnitude at the diagonals with values of 9 and 3 (the italicized numbers here). You can tell just by looking that the magnitude of the gradient at the 9 will be larger, so no, it's not an edge.

Workshop Questions

1. Consider an image in which the intensities are given by the equation I(x,y) = x^2+2y.
   a. What is the gradient of the image?

(2x, 2).

b. What is the gradient at the location (x,y) = (2,3)?

(4, 2)

c. At the location (x,y) = (2,3), what is the directional derivative in the direction of (3,2)? That is, if you were to move a small amount in this direction, how much would you expect the image to change?

If you move a little in this direction, say moving by (3δ, 2δ), the change in intensity is 14δ. The distance you moved is sqrt(13)δ. So the direction derivative is: $14\delta/\mathrm{sqrt}(13)\delta = 14/\mathrm{sqrt}(13)$.

2. Make part of a discrete image with intensities chosen according to the equation in 1. That is, for example, make an image so that the pixel at location (7,3) has the value $7^2+2*3 = 55$.

3	6	11	18	27
5	8	13	20	29
7	10	15	22	31
9	12	17	24	33
11	14	19	26	35

a. If you were to compute the discrete image gradient at each pixel using derivative filters, what values would you get for the image gradient everywhere?

I'll use filters like [-1/2, 0, ½]. At the point with the value 15 (x=3, y=3), the gradient is: (6, 2).

b. How do these compare to the gradients you get in problem 1? If there is a difference, why is this?

The formula predicted (6,2). They are the same. Notice this isn't always true.

When this will be true is a little tricky. Here's a quick discussion. The y component of the gradient will always match the formula when we compute it discretely, because the 2nd derivative of the image with respect to y is 0. Notice that the 2nd derivative with respect to x isn't 0, but it's constant. This means that using the filter [-1/2, 0, ½] we get the same result as the formula, but this would not be true for the filter [0, -1, 1]. Understanding why this is true is a little more advanced, but interesting.

3. Suppose you have an image in which the intensities are given by the equation $I(x,y) = 100 - x^2 - y^2 + xy + x + 4y$. What is the intensity of the brightest point in the image?

The key insight here is that at the brightest point, the gradient will be (0,0). Otherwise, there would be a direction we could move in to make it brighter. This is just like the 1D case, where the derivative is 0 at the extremal values of a function.

The gradient I $(-2x + y + 1, -2y + x + 4)$. Setting these both to 0 gives the equations:

$-2x+y+1 = 0$
$-2y+x+4 = 0.$

So we get $y + 1 - 4y + 8 = 0$.
$9 = 3y$. $y = 3$. $x = 2$. *Intensity* $= 100 - 4 - 9 + 6 + 2 + 12 = 107$.

4. Suppose g is a Gaussian filter with a standard deviation of 1.
   a. If I filter an image twice with g, show why this is equivalent to filtering it once with a larger Gaussian.
   b. What should the standard deviation of the larger Gaussian filter be?

I'm going to leave this one as a challenge problem. Here are some hints. If you're interested, first try combining the filters using Matlab. Plot the result, and see that it looks like a Gaussian. Use Matlab to see what the variance of the new filter is.

To show this analytically, you need to write convolution as a continuous operation, using integrals, Then you can play around with this expression, removing one of the integrals you get, and putting it in the form of a single convolution with a larger Gaussian.

5. Using Matlab, create a filter that has the same effect as applying a 1x3 averaging filter 20 times. What is the relationship between this filter and a Gaussian filter?

```
a = ones(1,3)*(1/3);
f = a;
for I = 1:19 f = imfilter(a,f); end
figure;
plot(f);
```

6. Create a 1D filter that smooths an image and takes a derivative at the same time.
   a. Do this using an averaging filter to smooth the image.

*[0, -1, 1] * [1/3, 1/3, 1/3] looks like [-1/3, 0, 0, 1/3]. Not so smooth.*

b. Do it using a Gaussian to smooth.

Do this in Matlab.

c. What do you think about the differences?

Looks a lot smoother with the Gaussian. This gets even worse when you use a very wide averaging filter, and a wide Gaussian.