Bruce A. Maxwell
Review Homework Questions for Week 1

1. What is the difference between an interleaved image representation and a planar image representation in memory?

An interleaved representation stores all of the color channels at a pixel together. For example, for an RGB image, the data is laid out in memory as R_(0,0) G_(0,0) B_(0,0) R_(0,1), G_(0,1), B(0,1), ...

A planar representation stores all of the first color channel information before all of the second color channel information. For example, it would be laid out in memory as R_(0,0) R(0,1) R(0,2) ... G_(0,0) G_(0,1) G(0,2) ... B_(0,0) B_(0,1) B_(0,2) ...

2. Give a couple of subjective explanations of what a filter can do.

Some examples include:

- A filter can blur the appearance of an image
- A filter can enhance the contrast of an image
- A filter can change the color balance of an image
- A filter can identify areas where the image is changing quickly
- A filter can remove noise in an image

3. What are a couple of ways of handling boundary issues (the edges of the image) when applying a filter?

Some examples include:

- Valid convolution: computing a result only where the filter fits completely inside the image
- 0-padding: treat the area outside the image as though it contains 0 values
- reflection: treat the area outside the image as though it is a reflection of the actual image across the boundary
- special handling: adjust the filter size so it adapts to the area that overlaps the image

4. If you have a 3x3 filter, how many operations per pixel (assume a greyscale image) are required for convolution? What about a 5x5 filter? What about a 7x7 filter? What about a 1x11 filter? (What I'm really asking, is what is the growth in computation as filter size increases?)

3x3 filter: filter size is 3x3 = 9 pixels, 9 multiplications, 8 additions, possibly 1 division for normalization = 18

5x5 filter: filter size is 5x5 = 25 pixels, 25 multiplications, 24 additions, 1 normalization = 50

7x7 filter: filter size is 7x7 = 49 pixels, 49 multiplications, 48 additions, 1 normalization = 98

1x11 filter: filter size is 1x11 = 11 pixels, 11 multiplications, 10 additions, 1 normalization = 22

The computational complexity is O(N) if N is the number of pixels, or O(N^2) if N is the width of a square filter.

5. Full convolution is convolution where the result contains all possible ways the two filters could overlap. What happens if you do full convolution between the following two filters (treat ∗ as the convolution operator)?

```
[1   2   1]     | −1 |
            ∗   |  0 |
                |  1 |
```

The left matrix is symmetric, so flipping it horizontally and vertically results in the same filter.

Consider all possible ways of overlapping the two filters. Let the position of the leftmost filter define the position in the output, with (0, 0) being the most upper left configuration.

The first overlap is [1 ∗ −1]. Moving the left filter to the right one pixel at a time, the result will be [−1, −2, −1] for the first row of the output.

The second row of the output will be [0 0 0], because the overlapping pixels always include a zero.

The third row of the output will be [1 2 1], as the left matrix overlaps the 1.

The final output will be

```
| −1  −2  −1 |
|  0   0   0 |
|  1   2   1 |
```

Now think about how this affects the run-time of a Sobel filter. Recall that convolution is associative. Therefore:

(f ∗ g) ∗ h = f ∗ (g ∗ h)

If a Sobel can be built from two 1x3 filters, lets let f = [1 2 1], g

= [−1 0 1]^t and h = the image.

The computational cost of (f ∗ g) ∗ h is 9 operations to generate (f ∗ g) and then 18∗image size to apply the 3x3 filter to the image.

The computational cost of f ∗ (g ∗ h) is 6∗image size to apply the 3x1 filter to the image, then 6∗image to apply the 1x3 filter to the image. Therefore, the total cost of the second approach is 12 x image size.

This is an example of what we call a separable filter. Separable filters can be implemented much more quickly by factoring them into two 1xN filters instead of one NxN filter. Think about the gains when filters get large (e.g. 21x21).