

## Slide 1 – Title Slide

Hello and welcome to Week 11, part 2 of EGM310: Image Enhancements. In this lesson, we will learn about different ways we can enhance images. We will also learn about how we can use filters to reduce noise or enhance certain features like edges.

## Slide 2 – Image Enhancement

Why do we want to enhance images? Please note – I’m not talking about being able to zoom in infinitely on an image with increasing clarity, because that doesn’t actually exist. By enhancement, we normally mean increasing contrast, or improving the distribution of color values within the image. Sometimes, it’s to improve the appearance of the image, in order to aid visual interpretation, aid other analysis, or for aesthetic reasons – we might want to make an image look nice for a publication or other purpose. Image enhancement can be done in a number of different ways, but there are three broad classes that we will touch on in this lesson: contrast stretching, histogram manipulation, and filtering.

## Slide 3 – Low-contrast image

Let’s start with a low-contrast image. Overall, this image is rather dark – you might be able to pick out some features – you can probably tell that this image shows Portrush and Portstewart, for example. But it’s hard to get much information out of this image, and doing any sort of mapping or visual interpretation would be difficult. Part of why this is so, is because most of the image is very low-intensity. I’ve included a histogram of the pixel values in the image to help illustrate this. In the histogram, the normalized intensity – that is, how bright the pixel appears – is along the x-axis, and the frequency, or how often a particular value appears, is along the y-axis. So this histogram shows a large peak of values around 0.1, then another peak just above that, but very few pixels have values above 0.2. So most of the pixels have a low intensity, and the image appears quite dark as a result. Most of the image is squished into this range between 10% and 20% of the total range, so we would say that the “dynamic range” of this image is quite small – it doesn’t make use of the whole range of possible values. The image thus has a very low contrast – the difference between the brightest areas and darkest areas is fairly small.

## Slide 4 – Linear contrast stretch

One way that we can improve the appearance is by using something called a linear contrast stretch. By “contrast stretch,” I mean that we are taking the values in the image and stretching them so that they fill out more of the range of possible values. By “linear,” I mean that to do this, the transformation that we use follows that of a line – we multiply by a slope, or scale, and optionally add a constant value, in order to calculate the values in the transformed image. Most GIS software have some way of applying a contrast stretch to an image – in ArcMap, you can find it in the symbology tab within the image properties window. To calculate the scale, I’ve used the 2<sup>nd</sup> percentile for the low end – this is the value where 2% of the pixels in the image are darker than that value. At the high end, I’ve used the 98<sup>th</sup> percentile – the value where only 2% of the pixels in the image are brighter than that value.

## Slide 5 – Linear contrast stretch

This is not the only way to determine the slope, though. We can choose the maximum and minimum values in the image, or the maximum and minimum possible values for the type of image that it is (in other words, whether it's an 8-bit image, a 16-bit image, or so on). Outlier values can throw off the scaling, though, which we see here. The second image, labeled "min/max," shows that we've stretched the histogram somewhat – it now fills more of the 0-25% intensity range, but we still don't have many values above 25% or so. This is because there's a small number of abnormally bright pixels in the image that are "outliers" – they don't fit within the range of most of the pixels in the image. To help counteract this, we can take a percentile instead – for example, using the bottom 2% and top 2% of values. We can also use the mean pixel value plus/minus a multiple of the standard deviation of pixel values, as shown on the far right here. This doesn't always work if the pixel values in the image aren't normally-distributed, though – you can see that the values in our image are bimodal, and so we have a bit of a lopsided distribution here. This type of transformation, linear contrast stretching, can be great because it doesn't distort the image – it just rescales it. But, it can also mean that we 'wash out' our brightest or darkest pixels, as you can see here.

## Slide 6 – Gamma adjustment

Another type of contrast stretching that won't wash out the brightest or darkest pixels is what's known as a gamma adjustment, or gamma correction. In this type of stretch, the pixel values are raised to a power, gamma. If gamma is greater than 1, this has the effect of shifting the histogram to the left, darkening the image. If gamma is less than 1, this has the effect of shifting the histogram to the right – brightening the image. In the example here, I've first done a min/max stretch on our original, low-contrast image – you can see this here in the bottom left. Then, I applied a gamma adjustment of  $\frac{1}{2}$ , which shifted the histogram closer to the middle of the range. The image appears brighter, and we can see more detail in some of the darker pixels on the water, for instance, that we couldn't see with the contrast stretch alone. It also allows us to see more contrast without saturating the brightest values – unlike with the 2/98% contrast stretch, the beaches aren't completely washed out, and we can see a lot more brightness variation there, as well.

## Slide 7 – Histogram equalization

Sometimes, we want to be able to better see local, small-scale variations in an image. To do this, we can use a technique called histogram equalization. In the plot at the upper left here, you can see the histogram of our original image, as well as the "equalized" histogram. We've stretched out, or equalized, the histogram so that the values are fairly flat – we don't have pronounced peaks around a given intensity value. This has worked particularly well for the darker parts of the image – we can really see the variations in intensity over the water here, and the differences between a lot of the agricultural fields are greatly enhanced – brighter fields have become much brighter, while darker fields have stayed the same, or gotten a bit darker. This can have the effect of decreasing contrast in some areas, though – the brighter areas, such as the beaches or the towns of Portrush and Portstewart, have sort of washed out in this image, and it can also amplify noisier parts of an image while

decreasing the signal, making some things harder to discern. To help counteract this, we can use modified techniques such as local, or adaptive, histogram equalization, which does this over smaller sections of the image. Another variation, called contrast-limited adaptive histogram equalization, limits the amount of contrast in each of the subregions of the image, in order to prevent noise from being amplified too much.

## Slide 8 – Filtering

Filters are a technique that can either help enhance or smooth image data. They can highlight certain features in an image, or they can be used to suppress features. We use them for things like edge detection, sharpening images, or reducing noise. The way a filter normally works is by convolving a kernel – moving it around the image – to calculate new values. For each pixel in the input image, we can then calculate the value in the output image using the values that fall within our window, or kernel. To see how this works, I have a short example here showing a 3x3 mean filter. That is, we're going to take a window of 3 pixels by 3 pixels, and calculate the mean of the values from our image – that's the value that goes into the output image. So, we start with our image here. Our kernel looks like this – an array of numbers called weights. As we move the window around our image, each of the pixels within our window gets multiplied by its corresponding weight and then added together. For a 3x3 mean filter, the weights are all the same, and they are equal to  $1/9$  – the 9 coming from the fact that we have 9 pixels in the kernel. Here we have our window, which is centered on the blue pixel here. If we then multiply each of the values in the window by  $1/9$  and add them together, we get 16 back out – so this goes into our output image here. And we can keep going to get all of the values in the new image. You might notice that the edges, where we can't get a full window, aren't filled in – normally, we need to specify what to do here – you'll need to check the particular software package to see how it handles these edge cases.

## Slide 9 – Low-pass (smoothing) filter

One common type of filter is what is known as a low-pass, or smoothing, filter. One example of this is the mean filter that we showed on the previous slide. The term low-pass comes from the fact that sharp features – high-frequency changes – are filtered out. Only low-frequency changes are left. You might also hear this referred to as a “blur” filter, because the end result is that the image looks blurry. One of the most common implementations of this is by using something called a Gaussian kernel. A Gaussian kernel takes the shape of a 2-d bell curve – if we look at a 3x3 example here, we see that the largest weight is in the middle of the kernel, with a drop-off as we move away. Each of the weights is multiplied by  $1/16$  so that the end result is still an average of the pixels that fall within the window. We can have other variations on this, as well – if we want to increase the weight on the center, we might use a 2-sigma, or 2 standard deviation, version, rather than the 1-sigma, or 1-standard deviation, that is shown here.

## Slide 10 – Example: noise reduction

On the left-hand side of this slide, I have an example of a satellite image that has a significant amount of noise in it, due to some errors in transmission between the satellite and the ground station. As a result, a number of pixels appear darker or brighter than their neighbors, resulting in a noisy pattern – you can especially see this with the line running down the image. By applying a small median filter, we can smooth out and remove a lot of this noise – it's not perfect, but this does help reduce the impact of some of the noisier parts of the image. This does have the effect of removing some of the detail of the image, some of which is real – in using some of these tools, we have to decide which of these things are more important: removing the noise, or preserving as much of the image as possible.

## Slide 11 – High-pass (sharpening) filter

To remove the low-frequency variation in an image (essentially, the color), and just look at the high-frequency variation, we use what is called a high-pass, or sharpening, filter. The way that we calculate this is by subtracting the low-pass filtered image from the original, leaving only the high-frequency variation. This has the effect of bringing out the sharper edges of the scene – the beaches here really stand out, for example – but the color differences between different surfaces are gone – the water here looks essentially the same color as most of the other parts of the image. This kind of filter can be useful for automatically identifying objects in between two images where the illumination changes dramatically – the shapes and patterns are preserved and will look fairly similar, even if the color or brightness level changes by a lot.

## Slide 13 – Edge filters

A more extreme application of this kind of filter is using something called an edge filter. The purpose of these kind of filters is, as you might have thought, to find sharp boundaries or edges in an image. Two examples are shown here, something called a Sobel filter on the left, and a Prewitt filter on the right. These types of edge filters work by approximating the first derivative, or gradient, of the image in either the horizontal (x) or vertical (y) directions. If you think back to trigonometry, we can calculate the magnitude of the gradient by adding the square of the x component to the square of the y component, and taking the square root. The result is shown in the two images here. Anywhere the filtered image is bright, we have a sharp, well-defined edge in the original image; anywhere it is dark, we don't have a very well-defined edge. In a high-resolution image, you might be able to use this to help pick out buildings or roads or other human-created features, or to find boundaries between different landcover types – not just visually, or digitizing by hand, but also automatically using the computer.

## Slide 14 – Summary

In this lesson, we have learned about how we can use enhancements to help us visually interpret images, and reduce noise in images. We learned about how we can use contrast stretching, a type of enhancement that increases contrast in images, helping us to see more detail and use more of the of the

range of possible values. We also learned about filtering, which we can use to reduce, or enhance, small-scale variation in images, we also saw some examples where we can use this technique to find the edges of objects in an image.

## **Slide 15 – Additional resources**

Once again, you can read more about the concepts we've covered in this lesson in the textbooks, Chapter 7 of Lillesand, Kiefer & Chipman; and Chapter 4 of Campbell & Wynne. I've added links to 3 videos below, the first two talk about some more of the mathematics and computer science behind image filters, while the third shows how you can apply contrast stretches in ArcGIS. That's all for this lesson – I hope you found it interesting, and if you have any questions, please don't hesitate to e-mail me or post in the discussion forum on blackboard. Bye!