L4 – Type of Image Feature and Segmentation

Feature Extraction

**Global Features** are features extracted from the entire image.

**Block-based Features** are where the image is split into blocks and it finds one feature in each box.

**Region-based Features** are where the image is split into regions and one feature is extracted per region.

**Local Features** are where interesting points are located and the features are extracted from the local area of the interesting point.

Global Features

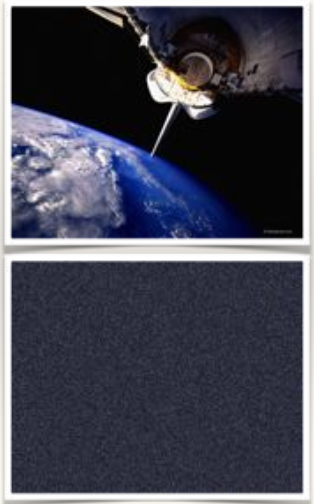
We looked at finding a global feature by taking the average of the colour bands but it wasn’t very robust. It is more common to produce a **histogram** of the pixel values.

A **Joint-Colour Histogram** measure the number of times a colour appears in an image.

The colourspace is quantised into bins, and we accumulate the number of pixels in each bin. (English: We take each pixel, figure out which colour bin it fits into and increment a counter for that bin)

This produces a multidimensional histogram so we unwrap it to make a feature vector.

Normalisation allows the histogram to be **invariant** to the image size. The choice of the colour space can also make it invariant to uniform lighting changes. It is also invariant to rotation but vastly different images can have the same histogram.



This is an example where the bottom picture is a randomisation of the pixels from the top picture but because they contain the same pixels the histogram is identical

Image Segmentation

You create region based descriptions of the image. You partition the pixels into these regions(segments). Typically, pixels in a segment have similar visual characteristics e.g colour.

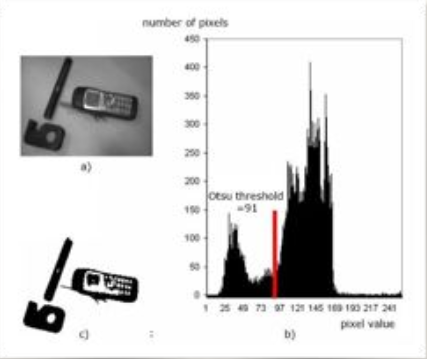
**Thresholding** is the simplest way to segment, it creates a binary image (two segments) based on a threshold. This is very fast but requires a manually set threshold. It works very well in applications with lots of physical constraints.

Otsu’s Thresholding

It automatically finds the threshold and is optimal.

It assumes there is a foreground and a background, creates a histogram and expects two peaks, one for the colour of the background and the other for the foreground.

It then exhaustively searches for the threshold inbetween the peaks that maximises the variance. The threshold will tend towards the largest mass, hence the threshold in the image.



Adaptive Thresholding

**Adaptive/Local Thresholding** choses a separate threshold for every pixel based on it’s neighbours.

In **Mean Adaptive Thresholding** if the pixel value is less than the mean of it’s neighbours plus a **constant** it sets it to 0 or otherwise sets to 1.

Adaptive has good invariance to uneven lighting and contrast but is computationally expensive. It can also be difficult to choose the window size because if the distance between the image and the camera changes then the window size should change too.

K-Means Segmentation

You can use K-means clustering to cluster the colour vectors and assign each pixel to the segment bases on the closest cluster centroid. This is only works if the colour-space and distance function are compatible. In **LAB** colour space it is designed so that the Euclidean distances are proportional to perceptual colour differences.

The problem is that you might end up with odd pixels.

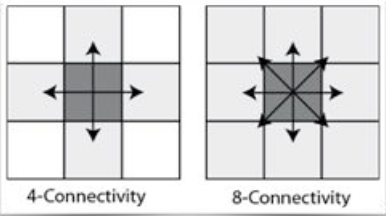
You can encode the spatial position in the vectors that are being clustered [r,g,b,x,y]. You normalise x and y by the width and height of the image so different image sizes don’t effect it. You can then scale x and y so that you can control the effect they have compared to the colour components.

Advance Segmentation

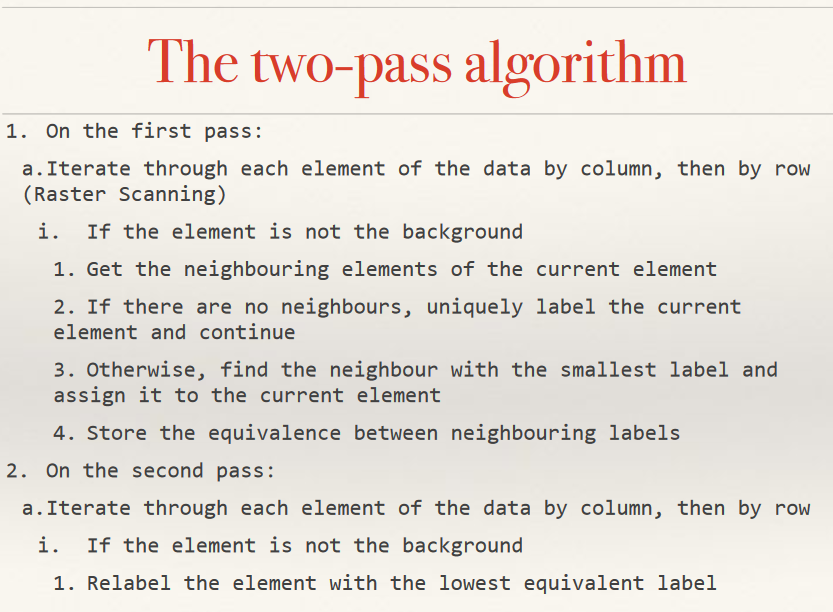
There is a lot of ongoing research in this field. There are techniques to automatically determine the number of segments or techniques that it objects in the scene based on training examples.

**Felzenswalb-Huttenlocher Segmentation**

Connected Components

A pixel is connected with another if they are spatially adjacent to each other. There is 4-connectivity and 8-connectivity.  


Therefore a connected component is a set of connected pixels. **Connected Component Labelling** is where you find all the components within a binary image. There are lots of algorithms to do this and the main tradeoff is memory vs time.



L5 – Shape Description and Modelling

Borders

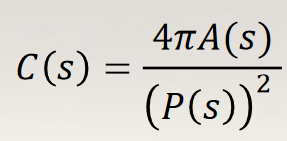
A border can either be inner or outer, an inner border will use the edge pixels of a component as a border whereas an outer border will create new pixels surrounding a component as the new border.

You can describe an object by its **region** or its **border**.

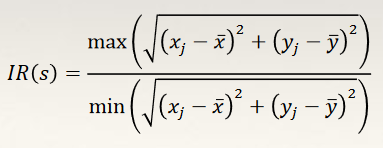
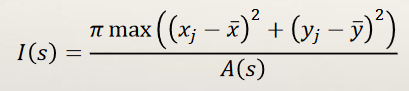
Region Description

This is things such as area and perimeter.

It is also **compactness**, compactness measures tightly packed the pixel in a component are. It is often computed as the weighted ration of area to perimeter squared.



You can also describe the **centre of mass** which is just the point at the mean x and mean y co-ordinate.

You can describe the **irregularity/dispersion**; this is a measure of how spread out it is.  


Moments

**Moments** describe the distribution of pixels in a shape. They can be computed for any grey-level image.