Computational Vision
Revision Notes

James Brown

April 5, 2017

Contents

1 Introduction

These are notes I have written in preparation of the 2017 Computation Vision exam. This year the module was run by Hamid Deghani (H.Dehghani@cs.bham.ac.uk).

Computational vision is the acquisition of knowledge about objects and events in the environment through information processing of light emitted or reflected from objects. In short - we want to make a computer know what is where, by looking through information. We can also use computational vision to do automatic inference of properties of the world from images.

2 Human Vision

As humans we have evolved eyes which perceive the visible section of the electromagnetic spectrum, which falls between the wavelengths of 380nm - 760nm. Red light lies at the longer end (760nm) of visible light, and purple at the shorter end (380nm). Visible light is strongly absorbed in the human eye because it can cause an electron to jump to a higher energy levels - yet it does not have enough energy to ionize cells. The evolutionary process of evolving eyes began more than 3 billion years ago with the formation of photopigments. These are molecules where light incident upon them will trigger a physical or chemical change. Photopigments capture photons which lead to the release of energy in the photopigment. This is may be used for photosynthesis or a behavioural reaction (a nerve reaction). A single photocell contains multiple layers to catch light, not just one. This increases the chance of catching any one individual photon - if it's not caught by the first layer it's much more likely to be caught by the second and so on.

2.1 Image Formation

Photocells contain a light sensitive patch of photopigments. Using a single cell we can capture light in 1 dimension, but we can't really 'see'. All we can do is tell if the light is on, or off. With multiple cells we can have better direction resolution and with multiple cells we have a very wide aperture - we can't tell exactly where the image is. The image formed will be very bright but extremely fuzzy. This became curved over time and this which really helped with direction resolution as light incident on the left side of the curve must have come from the right. Images formed this way are still very blurry and will result in multiple projections of the same image. Over time eyes evolved to become pinhole cameras which form sharp yet dim images by allowing light to come from a single source (the pinhole) - effectively throwing away loads of potential information about the image. The solution to form sharp and bright images was to use a lens at the front of the eye. The lens focuses all incoming light to a single point and from there we can use our simple pinhole camera for forming images. It should be noted that the image formed is upside down to what exists in reality.

2.2 Retina Processing

The human retina contains two kinds of photoreceptors which respond to incident light - **rods** (around 120 million) and **cones** (around 6 million). Rods are extremely sensitive photosensors and respond to a single photon of light. Multiple rod cells converge to the same ganglion cell and therefore neuron within the which results in poor spatial resolution. Rods are responsible simply for detecting the presence of light, and as a result make up the entirety of our night-vision. On the other hand, cones are active at much higher light levels and responsible for the detection of different colours of light. Cones have a much higher resolution as they are processed by several different neurons. Within the eye we have a receptive field, which is the area on which light must fall for a neuron to be stimulated. It should be noted that receptive field in the center of the eye is much smaller than it is for the periphery of the eye.

2.3 The Visual Pathway

Vision is generated by photoreceptors in the retina. All the information captured leaves the eye by way of the optic nerve. There is a partial crossing of axons at the optic chiasm. This is only partial as information from both eyes is sent to both sides of the brain - this allows us to process depth. After this chiasm, the axons are called the optic tract. The optic tract wraps around the midbrain to get to the lateral geniculate nucleus (LGN). The LGN axons fan out through the deep white matter of the brain and ultimately to the visual cortex.

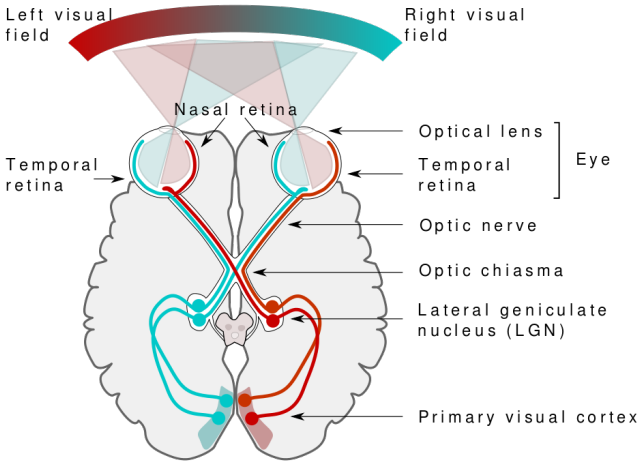

Figure 1: The visual pathway in the human brain

2.4 Colour Processing

Objects in the world selectively absorb some wavelengths (a colour) and reflect other wavelengths. Human retinas contain three different types of cones to respond to different colours. This gives us the ability to distinguish different forms of the same object - for example: unripe, ripe and off fruit. Many theories of colour vision have been proposed and some have been hard to disprove until recently.

In 1802 Young proposed that the eye has three different types of receptors, each of which are sensitive to a single hue. From this he proposed that any colour can be produced by appropriate mixing of the three primary colours. This is known as the trichromatic theory.

Hering suggested that colour may be represented in a visual system as 'opponent colours' in the 1930's.

3 Edge Detection

3.1 Intensity Images

An intensity image is a data matrix whose values represent intensities within some range. Each element of the matrix corresponds to one image pixel. An indexed image consists of a data matrix, X, and a colour map matrix, *map*. *map* is a m-by-3 array of double containing floating point values in the range $[0, 1]$. Each row in the map specifies the red, green an blue components of a single color. Each cell in the indexed image then specifies the corresponding colour from the colour map. In an intensity image can be thought of as a function $f(x, y)$ mapping coordinates to intensity. From this we can calculate an intensity gradient.

$$\overrightarrow{G}[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$$

This is a vector which we can think of as having an x and y component. We can calculate both the magnitude and direction for this gradient of intensity.

$$M(\overrightarrow{G}) = \sqrt{G_x^2 + G_y^2} \qquad \alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

When calculating the magnitude, figuring out a square root can be very computationally expensive. It's possible to replace this with an approximation: $M(\overrightarrow{G}) = |G_x| + |G_y|$

3.2 Approximating the Intensity Gradient

In order to approximate the gradient we may use a variety of different masks over the image, such as the Roberts and Sobel detectors. These masks use the idea of convolution in order to calculate the rate of change of intensity at each pixel. Convolution is the computation of weighted sums of image pixels. For each pixel in the image, the new value is calculated by translating the mask to the pixel and taking the weighted sum of all the pixels in the neighbourhood. Once we have a value for the rate of change of the intensity gradient we can threshold our image to produce the locations of edges.

4 Noise Filtering

When taking images we also gather a lot of noise which results in fake edges once we apply our edge detectors. We would like to remove this noise and we many filters which can be implemented by the idea of convolution. The most widely used of these filters is the Gaussian filter, although there are other simpler filters such as the mean filter.

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

Figure 2: An example mean filter

$$\begin{bmatrix} 0 & .01 & .02 & .01 & 0 \\ .01 & .06 & .11 & .06 & .01 \\ .02 & .11 & .16 & .11 & .02 \\ .01 & .06 & .11 & .06 & .01 \\ 0 & .01 & .02 & .01 & 0 \end{bmatrix}$$

Figure 3: An example Gaussian filter

The mean filter averages a pixels value with all of the surrounding intensities for a new value. A Gaussian filter follows the same principle as this but uses a weighting, valuing pixels closer to the original more than pixels farther away. The Gaussian filter has an extra bonus of being able to be applied as 2 individual 1D Gaussian filters in sequence rather than one large 2D filter as shown below.

$$\begin{bmatrix} 0.0545 & 0.2442 & 0.4026 & 0.2442 & 0.0545 \end{bmatrix} \qquad \begin{bmatrix} 0.0545 \\ 0.2442 \\ 0.4026 \\ 0.2442 \\ 0.0545 \end{bmatrix}$$

5 Advanced Edge Detection

Intensity changes can be caused by geometric events such as surface orientation discontinuities, depth discontinuities, color discontinuities and texture discontinuities. It can also be caused by non-geometric events such as illumination changes, specularities, shadows and inter-reflections. All of these events can be used to try to find edges in the image. When we try to detect edges, we are trying to produce a line 'drawing' of a scene from an image of that scene. We use this to

extract important features of an image, such as corners and curves. These features are then used by higher-level computer vision algorithms.

5.1 Main steps in Edge Detection

Regardless of methods, there a few major steps to edge detection:

1. **Smoothing**: suppress as much noise as possible, without destroying any of the true edges.
2. **Enhancement**: apply differentiation to enhance the quality of edges (i.e. sharpening).
3. **Thresholding**: determine which edge pixels should be discarded as noise and which should be retained (i.e. threshold edge magnitude).
4. **Localization**: determine the exact edge location.

Most of the time, points that lie on an edge are detected by

1. Detecting the local *maxima* or *minima* of the first derivative.
2. Detecting the *zero-crossings* of the second derivative

There are a few practical issues that come with this. When smoothing an image, the smoothing effect achieved depends on the mask size (for example, with a Gaussian filter it depends on σ). A larger mask will reduce noise more, but it also worsens localization and adds uncertainty to the location of the edge.

Based on this we can draw up some criteria for an optimal method of edge detection:

1. **Good detection**: Minimize the probability of false positives and false negatives (spurious edges and missing real egdes).
2. **Good localization**: Detected edges must be as close as possible to the true edges.
3. **Single response**: Minimise the number of local maxima around the true edge.

Canny showed that the first derivative of Gaussian closely approximates the operator that optimizes the product of signal-to-noise ratio and localization.

5.2 Hysteresis

Hysteresis thresholding uses two thresholds - a low threshold t_l and a high threshold t_h (usually this is $2t_l$). This makes the assumption that important edges should be along continuous curves in the image. It allows us to follow a faint section of a given line and to discard a few noisy pixels that do not constitute a line but have produced large gradients. We begin by applying the high threshold - this marks edges that we can be fairly sure that are genuine. Starting from these points, and using the directional information derived earlier, edges can be traced throughout the whole image. While tracing an edge, we apply the lower threshold, allowing us to trace faint sections of edges as long as we find a starting point.

6 Hough Transform

7 Scale Invariant Feature Transform

8 Face Recognition

9 Motion

10 ROC Analysis

Receiver operating characteristic analysis provides us tools to select possibly optimal models and to discard suboptimal ones. In order to carry out ROC analysis we need to count the different

kind of errors that are possible. When classifying if a pixel is an edge or not, there are 4 possible situations - **true positive** (predicted it was an edge and it was), **false positive** (predicted it was an edge and it wasn't), **false negative** (predicted it wasn't an edge and it actually was) and **true negative** (predicted it wasn't an edge and it wasn't).

Once classifying each pixel, we then count the total number of each label (TP, FP, TN, FN) over the large dataset. In ROC analysis we use two statistics:

$$\text{Sensitivity} = \frac{TP}{TP+FN} \qquad \text{Specificity} = \frac{TN}{TN+FP}$$

Sensitivity is the likelihood of spotting a positive case when presented with one or the proportion of edges we find. Specificity is the likelihood of spotting a negative case when presented with one or the proportion of non edges that we find. To define a ROC space we only need the true positive rate (TPR) and the false positive rate (FPR). The TPR is simply the sensitivity and the FPR is $1 -$ specificity. We use the FPR and TPR as x and y axes respectively of the ROC space.

All of the optimal detectors lie on the convex hull - the best possible position is for a detector to lie in the top left corner. It should be noted that if the edge detector lies below the dotted line then we can instantly make the detector better by simply just flipping the output of the algorithm!

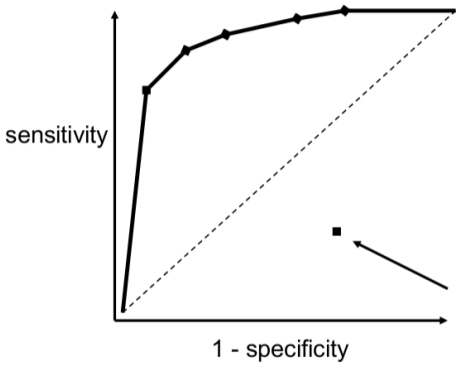

Figure 4: A sample ROC space

11 Object Recognition

12 Model Based Object Recognition

List of Figures

Index