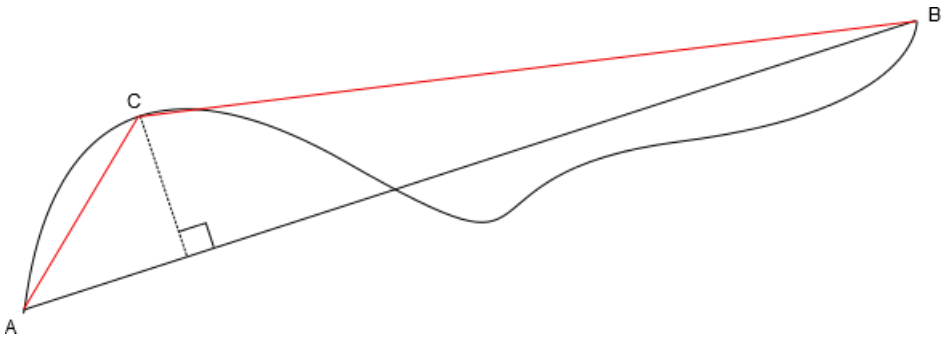
EEE6219 – Computer Vision – Solutions – 2015/16

Q1:

(a) Assume an edge detector produces a string (or linked list) of edgels. Describe a possible recursive algorithm to approximate such an edge string with a series of straight line segments. How would such an algorithm be terminated? [5 marks]

Given some edge string A-B, we would begin by approximating this string with the line AB:



At this point, we would calculate the normal distances from the line to each of the edgels in the string. We determine that the largest deviation occurs at point C. Consequently, we would add a 'knot' at C, now approximating the edge string with AC and CB (shown in red). Would then repeat the above process on the lines AC, and CB independently, partitioning each interval into AC → AD/DC, and CB→ CE/EB, and so on. Thus partitioning proceeds *recursively*, which produces a very elegant and compact algorithm.

As we proceed, we will approximate the edge string increasing numbers of straight line segments, hence producing an increasingly accurate approximation. The obvious termination criterion is to continue the recursive splitting until the largest deviation between and edgel and its straight line approximant is less than some user-defined threshold.

Approximating an edge string with a series of straight line segments will obviously lead to some error. In the algorithm you have described above, what bounds the error? [3 marks]

The recursive splitting procedure continues until the largest error is below the user-defined threshold, at which point the algorithm terminates. Consequently, the largest error is bounded by user-defined termination threshold.

Explain why it may be desirable to approximate an edge string using straight line segments. Give a possible example of such a use. [3 marks]

In general, computer vision is concerned with representing an image using increasingly high-level, and abstract representations. Describing an edge string with a series of straight line segments is just such an abstraction. One potential application is in identifying an object from its outline, where this outline is not easily expressible as a parametric curve. One could, for example, infer the presence of the object (using a Hough transform) constructed using straight-line approximations; this would be more computationally efficient than using the individual edgels.

(b) Suppose you are given a grey-level medical image containing views of a complex network of blood vessels. (Assume the image does not contain any other anatomical features other than blood vessels.) Suggest how you would go about automatically detecting the blood vessels in the image as

a series of disconnected vessel points. Clearly explain the principles underlying your approach. [6 marks]

Assuming we can model a blood vessel as an infinitely long tube of circular cross-section, in considering any arbitrary point on the tubular object, it should be clear that the curvature of the grey level along the vessel is (effectively) zero while the curvature at right angles to the vessel will be large with a magnitude determined by the radius of the vessel; the sign of the curvature will be dictated by the contrast of the original image – that is, depending on whether the vessels present as dark objects on a light ground, a light object on a dark ground. Assuming an ideal tubular object, we can compute the Hessian matrix at every individual point in the image:

$$\boldsymbol{H} = \begin{pmatrix} \dfrac{\partial^2 I}{\partial x^2} & \dfrac{\partial^2 I}{\partial x \partial y} \\ \dfrac{\partial^2 I}{\partial x \partial y} & \dfrac{\partial^2 I}{\partial y^2} \end{pmatrix}$$

which is also known as the curvature matrix. For a blood vessel of arbitrary orientation, the elements of this matrix will vary. However, if we perform on eigen-decomposition on $\boldsymbol{H}$, we will, in effect, rotate the variations in intensity into a frame in which the eigenvector associated with the smallest eigenvalue runs along the vessel, and the eigenvector associated with the largest eigenvalue is normal to the vessel. (These two directions will be orthogonal since $\boldsymbol{H}$ is a real, symmetric matrix.)

At this stage, we can examine the relative values of the eigenvalues λ_1 and λ_2 ; highly asymmetric values of $\lambda_{1,2}$ can be associated with the ridge of the vessel structure, whereas small values of each eigenvalue will be associated with the featureless background region. (In practice, either the determinant of trace of the Hessian matrix are used for this comparison.)

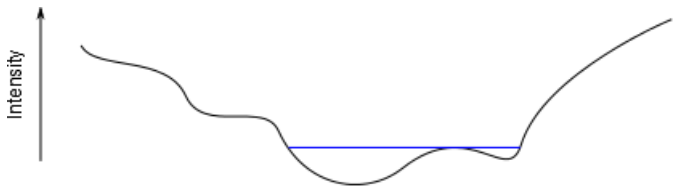
Further suppose that you wanted to extract the *network* of vessels by tracking along the detected points. Are there any particular points in the image which will prove difficult for the algorithm you have described above? (It is not necessary to describe the tracking algorithm.) [3 marks]

Real vessel network bifurcated – that is a vessel splits into two vessels. (In fact, the angle between the two resulting vessels varies depending on the anatomical feature being imaged.) Nonetheless, the above detection method based on the Hessian matrix assumes infinitely-long tubular object. Clearly bifurcations deviates from this assumption, and consequently, you should expect difficulties in detecting junctions between vessels simply because they do not conform to the fundamental assumptions made in the algorithm.

Q2:

(a) Describe the physical analogy underlying the *watershed segmentation* of a grey-level image interconnected regions. Explain how a watershed segmentation algorithm would be implemented in practice. How would the segmentation process terminate? [5 marks]

The physical analogy underlying the watershed algorithm is of a landscape, which can be represented in cross-section with the following figure:



Rain is presumed to fall on this landscape and begins to gather in the hollows, filling these and creating lakes of increasing surface area as more rain accumulates. If we imagine water that gathers in each hollow as a different colour, the point at which two lakes of different colours merge is the point at which we have segmented a given hollow.

In terms of practical implementation, we begin by selecting pixels with the lowest local values and giving each of these pixel a unique "colour". We then examine the 8-neighbours of each filled pixel and if any of the 8-neighbours is unfilled, we fill it with the "colour" of the central cell. We continue to cycle round each of the seed points from which we have initiated colour filling, and examine each of the 8-neighbours of the 8-neighbours, and so on. In this way, the seeded regions grow; the growth of a region ceases when we examine the 8-neighbours of its boundary, and the boundary pixel is filled by a different colour. At this point, we have identified the boundary, or extent, of a segmentable region.

What is the main shortcoming watershed segmentation? How can this problem be reduced? Are there any further techniques that could be employed to improve the quality of the segmentation result? [2 marks]

The principal shortcoming of the watershed segmentation algorithm described above is that the boundary regions identified tend to be sensitive to noise. This problem can be ameliorated by Gaussian-smoothing the image before applying the watershed segmentation algorithm. Determining the scale of the Gaussian smoothing, of course, is problematic. In addition to smoothing the image, it is also possible to improve the quality of watershed segmentation by:

1. Merging any very small regions with their surrounding regions

2. Merging adjacent regions with very similar grey levels.

Both these approaches, of course, require the setting of thresholds, which in its turn, is problematic.

(b) The image in Figure 1 was a acquired with a CMOS camera. Explain why the blades of the helicopter appear to be bent. [4 marks]

The artefact shown in the image is known as the 'rolling shutter' artefact is caused by the fact that low-cost CMOS sensors are typically constructed without row buffers. Pixel values are therefore read directly on the imaging sensor, one row at a time. Consequently, if the image changes on the scale of the readout time, strange artefacts like that shown occur. In this particular case, when the first row

of the image is read out, the blades of the helicopter are in some given position. When the second row of the image is read out, the helicopter's blades have moved slightly. Row one thus corresponds to one view of the scene, and the second row corresponds to a slightly different view. This process continues with the view of the scene changing slightly on the scale of the readout time for each row. When the image is reassembled from its constituent rows, the artefact issues hybrid image shown in Figure 1 results.

(c) A solid-state sensor comprises a grid of light-sensitive regions of dimensions $W \times W$. Explain the effect of the finite size, W of the pixel sensor on the frequency content of the acquired image. [5 marks]

A pixel of finite size represents an aperture, which is used to sample the image. The effect is to convolving the δ-sample acquired with infinitesimally small sampling aperture (i.e. ideal sampling) with the rectangular aperture of finite pixel width. In the Fourier domain, the rectangular aperture produces a sinc response, which is a low-pass filter since high frequencies are attenuated. For an aperture of width W , the first zero of the sinc occurs at a frequency of $1/W$, meaning that larger pixels attenuate more high-frequency information. This results in less fine detail in the image signal. In summary, a larger pixel size translates to less high-frequency content in the acquired image.

In practice, the pixels on a camera sensor do not butt together exactly – rather, there is a small gap between adjacent pixels. What happens when you reduce the pixel size W relative to the pixel pitch, say, Y – namely, when the pixel fill factor W/Y is reduced from unity to less than one? [4 marks]

If the pixel pitch is Y but the active area of the pixel is W , where $W < Y$, this will have the effect of reducing the width of the sampling aperture. This, in turn, will have the effect of increasing the high-frequency detail in the image signal since the width of the sinc function in the Fourier domain will increase.

The downside of reducing the aperture (pixel) size for a fixed pitch is at the sensitivity of the sensor will decrease because each pixel will now comprise a smaller light-sensitive area.

Q3:

(a) Briefly describe the pyramid representation of an image. For what purposes are pyramids useful in computer vision? [6 marks]

A pyramid representation generates a hierarchical structure by taking the original image, filtering it, and down sampling it. This process is repeated to form the next level in the pyramid, and so on. Consequently, a pyramid is a parallel representation of the image with successive levels have lower resolution and fewer high-frequency components.

Pyramids useful for searching over scale space. Locating the appropriate scale which to process or interpret an image can be very time-consuming, but searching over a (precomputed) pyramid at different scales is much more efficient. In addition, performing coarse-to-fine search is straightforward over a pyramid; we can start from the highest (lowest resolution) level, quickly locate a small number of candidate matches and 'pursue' these down the pyramid to higher resolution levels, discarding those search paths which turn out to be false. Eventually, we will arrive at the highest resolution level (original) of the pyramid with a valid match.

In terms of the frequency domain, how do the levels of a Laplacian pyramid differ from each other? What image processing operation does this implement? Describe the practical use of having such a representation. [3 marks]

In terms of the frequency domain, the different levels of a Laplacian pyramid on bandpass-filtered versions of the original. This allows each level of the pyramid to be 'probed' the signals at different scales.

(b) In the context of object recognition, outline the *bag of visual words* approach. What is the fundamental limitation of the bag of words approach? [5 marks]

The bag of visual words approach derives its name from the analogy to text recognition, in which the occurrence of a set of words is taken as a signature, regardless of the order in which the words occur. It is thus computationally easy to identify a set of visual features and simply infer the presence of an object in the presence of some fraction of the expected features on that object.

The drawback with the bag of words approach is, much like its text recognition origins, that the spatial ordering and geometrical relationships of the features are ignored leading to a lack of discriminatory power and ambiguous matching.

(c) What is a level set? What is the advantage of using a level set for segmenting an arbitrary shape from an image compared to, say, parameterising the shape with some curve? [5 marks]

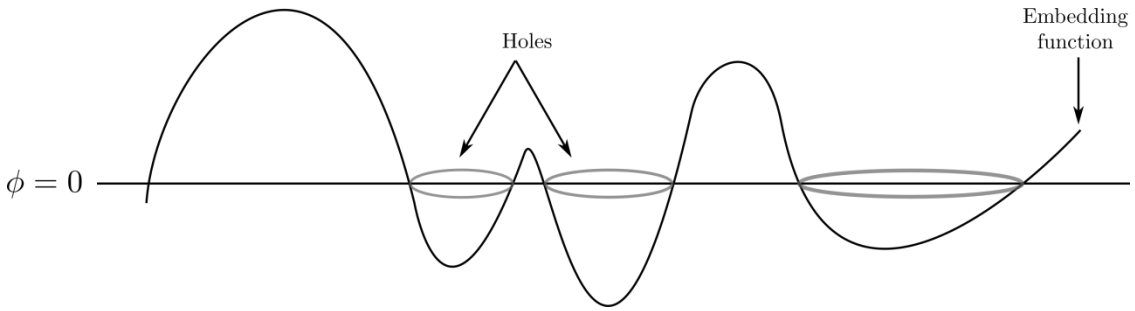
A level set is defined as the set of points on an embedding function for which the value of the embedding function is zero. Thus:

$$C = \{x | \varphi(x) = 0\}$$

Compared to an explicitly-parameterised curves, a level set provides much greater flexibility to describe naturally-occurring shapes. In particular, a single embedding function can describe a shape with several disjoint holes.

Sketch an illustration of how a level set can be used to segment an object from an image which contains holes. [2 marks]

An illustration of the way in which a level set can be used to describe an object with disjoint holes is:



Notice that the boundary of the object is defined by the set of points which the embedding function has a value of zero. If the embedding function oscillates suitably around a zero value, this can generate an appropriate segmentation of an object which includes disjoint holes.

Q4:

(a) In motion estimation, what is the *aperture problem*? What fundamental limitation does it impose? How can it be overcome? [4 marks]

The aperture problem is characterised by trying to estimate 2-dimensional motion where the only available information is intrinsically one-dimensional. This is often described in terms of viewing an edge through an *aperture* – hence the name – posing the question how is it possible to estimate motion parallel to the edge.

The estimation problem is, of course under-determined, and cannot be solved without additional information. In practice, it is often possible to identify the fact that the equations of 2-D motion are under-determined (for example, via rank deficient matrices) and switch to estimating 1D motion instead since this is all that it is possible to estimate.

(b) In tracking, what is the *data association* problem? What is the fundamental approach to dealing with this? [3 marks]

In tracking, the data association problem is how to correctly associate corresponding points in successive frames. Since the root of this problem is ambiguity, most straightforward approach to dealing with it is to impose some constraint, namely an explicit model of motion.

Explain the operation of the Kalman filter – in words – without recourse to any mathematical formulae. Illustrate how the Kalman filter can be used to address the data association problem in tracking. [5 marks]

Under the assumption of linear dynamics and Gaussian noise, a Kalman filter performed two steps:

1. Prediction

2. Correction or updating

Starting from a set of state equations, the prediction step estimates the state at time k based on the measurements up to, but not including time k. The state prediction is then updated using the information from the actual measurement at time k by carefully waiting the two sources of information – prediction and measurement – based on their respective uncertainties.

Tracking with a Kalman filter solves the data association problem is by forming an optimal prediction of the new system state together with an uncertainty on that prediction. This information typically provides a measure of plausibility on the set of associations, thereby greatly reducing the number of false matches.

Describe the main processing steps involved in the Random Sampling for Consensus (RANSAC) algorithm. [4 marks]

Given some set of data $\{x_1, x_2, \dots, x_n\}$, some of the points can be assumed to fit the function $f(x)$ where a remainder do not. In essence, the dataset is (heavily) contaminated by outliers. Taking the function to be $f(x, \Theta)$, where Θ is some unknown parameter vector, the RANSAC procedure randomly selects q points, performs a least-squares fit to $f()$ and determines the size of a "consensus" set lies within a "tolerable" band around $f()$. The procedure is repeated the some number of trials and the solution with the largest consensus set is adopted.

In using the RANSAC algorithm, how many points would you use to calculate a candidate fit to, say, a straight line? Justify your answer. [4 marks]

The size of the sample required to perform fitting should be the minimum number required to determine Θ since this provides the most efficient route to identifying the parameters. The fitting sample is required to comprise only *inliers* and the probability of this happening is maximised if the sample used for fitting has the minimum size required to determine Θ.

What is the major disadvantage of the RANSAC algorithm? [2 marks]

The major disadvantage of the RANSAC algorithm is the need to select a tolerance band, which is essentially a threshold on the decision whether a point is or is not an outlier. Setting this tolerance band can be problematic.

How many iterations should it take for the RANSAC algorithm to converge on the globally-correct answer? [2 marks]

RANSAC is a stochastic algorithm so there is always some non-zero probability of it not finding the desired solution, even for huge numbers of iterations. So the answer is: there is no answer to the question of how many iterations should be used. In practice, if the fraction of outliers in the data is known, it is possible to estimate the probability of finding the solution and adjust the number of trials to make this some acceptably large number, for example, 99%.