Exercises 1–10 for Computer Vision – with solutions

Exercise 1

Explain why inferring object surface properties from image properties is, in general, an ill-posed problem: some of Hadamard's criteria for well-posed problems are not satisfied. In the case of inferring the colours of objects from images, how does knowledge of the properties of the illuminant affect the status of the problem and its solubility? More generally, illustrate how addition of ancillary constraints or assumptions, even metaphysical assumptions, allows an ill-posed problem to be converted into a well-posed problem.

Exercise 2

In human vision, photoreceptors (cones) responsible for colour are numerous only near the fovea, mainly in the central ± 10 degrees. High spatial resolution likewise exists only there. So then why does the visual world appear to contain colour information everywhere in the field of view? Why does it also seem to have uniform spatial resolution? Why does the world appear stable despite all our eye movements? Discuss some implications for computer vision principles that might be drawn from these observations.

Exercise 3

Present five experimental observations about human vision that support the thesis that "vision is graphics:" what we see is explicable only partly by the optical image itself, but is more strongly determined by top-down knowledge, model-building and inference processes.

Exercise 4

The binary image pixel array on the left below is convolved ($*$) with what operator $\boxed{?}$ to give the result on the right? Specify the operator by numbers within an array, state its relationship to finite difference operators of specific orders, and identify what task this convolution accomplishes in computer vision.

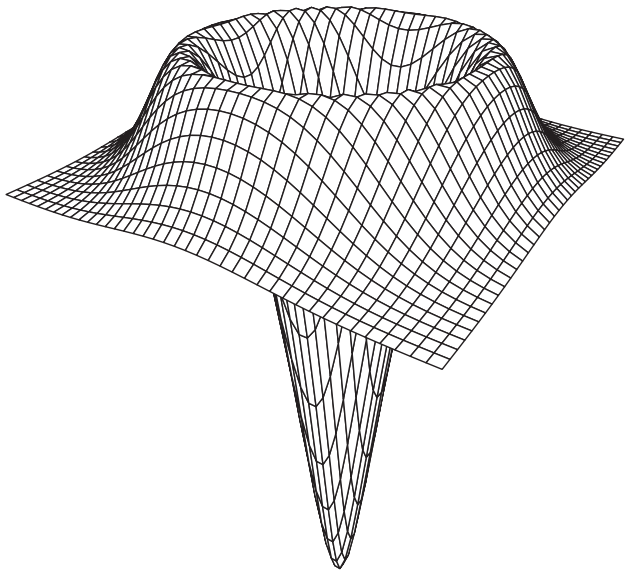
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0
0	0	0	**1**	**1**	**1**	**1**	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$* \boxed{?} \Rightarrow$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	**-1**	**1**	0	0	**1**	**-1**	0
0	**-1**	**1**	0	0	**1**	**-1**	0
0	**-1**	**1**	0	0	**1**	**-1**	0
0	**-1**	**1**	0	0	**1**	**-1**	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Exercise 5

The following operator is often applied to an image $I(x, y)$ in computer vision algorithms, to generate a related function $h(x, y)$:



$$h(x,y) = \int_{\alpha}\int_{\beta} \nabla^2 e^{-\left((x-\alpha)^2+(y-\beta)^2\right)/\sigma^2} I(\alpha,\beta)\; d\beta\; d\alpha$$

where

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$$

(a) Give the general name for the type of mathematical operation that computes $h(x, y)$, and the chief purpose that it serves in computer vision.

(b) What image properties should correspond to the zero-crossings of the equation, *i.e.* those isolated points (x, y) in the image $I(x, y)$ where the above result $h(x, y) = 0$?

(c) What is the significance of the parameter σ? If you increased its value, would there be more or fewer points (x, y) at which $h(x, y) = 0$?

(d) Describe the effect of the above operator in terms of the two-dimensional Fourier domain. What is the Fourier terminology for this image-domain operator? What are its general effects as a function of frequency, and as a function of orientation?

(e) If the computation of $h(x, y)$ above were implemented entirely by Fourier methods, would the complexity of this computation be greater or less than the image-domain operation expressed above, and when? What would be the trade-offs involved?

(f) If the image $I(x, y)$ has 2D Fourier Transform $F(u, v)$, provide an expression for $H(u, v)$, the 2D Fourier Transform of the desired result $h(x, y)$ in terms of only the Fourier plane variables (u, v), the image transform $F(u, v)$, and the parameter σ.

Answer to Exercise 1

Most of the problems we need to solve in vision are *ill-posed,* in Hadamard's sense that a *well-posed* problem must have the following set of properties:

- its solution exists;
- its solution is unique;
- its solution depends continuously on the data.

For example, inferring depth properties and 3D surface shape from image data is ill-posed because an image is a two-dimensional optical projection, but the world we wish to make sense of visually is three-dimensional. In this respect, vision is *"inverse optics:"* we need to invert the 3D $\longrightarrow$ 2D projection in order to recover world properties (object properties in space); but the 2D $\longrightarrow$ 3D inversion of such a projection is, strictly speaking, mathematically impossible. This violates Hadamard's 2nd criterion.

Inferring object colours in an illuminant-invariant manner is ill-posed because the wavelength mixture reaching a video camera (or the eye) is the product of the wavelength distribution of the illuminant (which may be multiple, extended, or a point source; narrowband or broadband; etc.) with the spectral reflectances of objects. We wish to infer the latter, i.e. object pigment properties, but in order to decompose the product we would need to know the wavelength distribution of the illuminant. Usually we don't have that information. This violates Hadamard's 1st criterion.

In many respects, computer vision is an "AI-complete" problem: building general-purpose vision machines would entail, or require, solutions to most of the general goals of artificial intelligence. But the intractable problems can be made tractable if metaphysical priors such as "objects cannot just disappear; they more likely occlude each other;" or "objects which seem to be deforming are probably just rotating in depth;" or "head-like objects are usually found on top of body-like objects, so integrate both kinds of evidence together;" etc. can resolve the violation of one or more of Hadamard's three criteria. Bayesian priors provide one means to do this, since the learning (or specification) of metaphysical principles ("truths about the nature of the world") can steer the integration of evidence appropriately, making an intractable problem soluble.

Answer to Exercise 2

The fact that the cone population subserving both high resolution and colour vision is numerous only near the fovea, yet the world appears uniformly coloured and uniformly resolved, reveals that our internal visual representation is built up and integrated somehow from multiple foveated "frames" over time. The stability of the visual world despite eye movements, and our unawareness of retinal blood vessels or blind spots, also suggest that human vision may have more to do with graphics than with merely image analysis. What we see may arise from a complex graphical process that is *constrained* by the retinal image as a rather distal initial input. It also shows the importance of integrating information over time, from multiple views. All of these are features that could be used as design principles in computer vision.

Answer to Exercise 3

The five supporting observations might include items from this list of ten:

1. The front of the retina is covered with a dense tree of blood vessels, creating an arborising silhouette on the image, but we do not see that.
2. Each retina has a large black hole (or "blind spot") where the 1 million fibres forming an optic nerve exit through the retina, about 17 degrees to the nasal side of the fovea; but we do not see these two large black holes.
3. Colour-sensitive cones are found mainly near the fovea, while colour-insensitive rods predominate elsewhere. Yet somehow we build up a representation of the visual world that seems to have colour everywhere.
4. High spatial resolution exists only near the fovea; yet our representation of the world does not seem to become blurry outside the fovea.
5. We constantly move our eyes about; but the world appears stable, and it does not seem to dart around (as it would if video cameras darted about like that).
6. As the Gestaltists showed in many demonstrations, what we see depends on context, expectations, and grouping principles, more than on just the literal image.
7. We can have rivalrous percepts, bi-stable visual interpretations that flip back and forth (like the Necker Cube), despite no change in the retinal image itself.
8. We experience many visual illusions: percepts not supported by the image itself.
9. We are capable of inferring the 3-dimensional structure of objects even from just a still picture, and can for example perform mental 3-D rotations of them into different poses or viewing angles, when solving tasks such as face recognition.
10. In human brain anatomy, there is a massive neural feedback projection from the cortex to the LGN.

Answer to Exercise 4

The mystery convolution operator $\boxed{?}$ is the following (1×3) array:

-1	2	-1

It corresponds to the second finite difference, the discrete form of a second derivative. It serves as a detector of *vertical edges* within images, localisable to the transitions between -1 and $+1$ in the output. (It could also be used to enhance the contrast of vertical edges.)

Answer to Exercise 5

(a) The operator is a convolution. Image $I(x, y)$ is being filtered by the Laplacian of a Gaussian to emphasize edges of a certain scale, and it can be used to detect them.

(b) The zero-crossings of the equation, isolated points where $h(x, y) = 0$, correspond to edges (at any angle) within the image $I(x, y)$. Thus this operator serves as an isotropic (non orientation-selective) edge detector. (Note that extended areas where the image is completely uniform, i.e. constant pixel values, will also be regions where $h(x, y) = 0$.)

(c) Parameter σ determines the scale of image analysis at which edges are detected. If its value were increased, there would be fewer edges detected, i.e. fewer zeroes of $h(x, y)$, but also fewer false edge detections related to spurious noise.

(d) In the 2D Fourier domain, the operator is a bandpass filter whose centre frequency is determined by σ. Low frequencies are attenuated, and also high frequencies are attenuated, but middle frequencies (determined by the value of σ) are emphasized. However, all orientations are treated equivalently: the operator is isotropic.

(e) The operation can be easier to implement via Fourier methods, because convolution is achieved by the simple multiplication of the Fourier transforms of the two functions being convolved. (In the case in question, these are the image and the Laplacian of a Gaussian filter.) In contrast, image-domain convolution requires a double integral to be computed in order to evaluate $h(x, y)$ for each point (x, y). But a Fourier cost is the requirement first to compute the Fourier transform of the image, and then to compute the inverse Fourier transform of the result after the multiplication, in order to recover the desired $h(x, y)$ function. The computational complexity (execution speed) of using Fourier methods becomes favourable for convolution kernels larger than about 5×5.

(f) By application of the 2D Differentiation Theorem, and the fact that the Fourier transform of a Gaussian of scale σ is also a Gaussian but with reciprocal scale $1/\sigma$:

$$H(u, v) = -(u^2 + v^2)\ e^{-(u^2+v^2)\sigma^2} F(u, v)$$

(We are ignoring constants 2 and π that would appear if the Gaussian were normalised to have unit volume, as would be necessary if it were a probability distribution.)

Exercise 6

(a) Extraction of visual features from images often involves convolution with filters that are themselves constructed from combinations of differential operators. One example is the Laplacian $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ of a Gaussian $G_\sigma(x, y)$ having scale parameter σ, generating the filter $\nabla^2 G_\sigma(x, y)$ for convolution with the image $I(x, y)$. Explain in detail each of the following three operator sequences, where $*$ signifies two-dimensional convolution.

(i) $\nabla^2 \left[G_\sigma(x, y) * I(x, y) \right]$

(ii) $G_\sigma(x, y) * \nabla^2 I(x, y)$

(iii) $\left[\nabla^2 G_\sigma(x, y) \right] * I(x, y)$

(b) What are the differences amongst them in their effects on the image?

Exercise 7

(a) For some image $I(x, y)$, define its gradient vector field $\vec{\nabla} I(x, y)$.

(b) Why is this vector field a useful thing to compute?

(c) Define the gradient magnitude that can be extracted over the image plane (x, y).

(d) Define the gradient direction that can be extracted over the image plane (x, y).

(e) Explain how the gradient vector field is used in the Canny edge detector, what are the main steps in its use, and its advantages over alternative approaches.

Answer to Exercise 6

(a) (i) Operation $\nabla^2 [G_\sigma(x,y) * I(x,y)]$ first smooths the image $I(x,y)$ at scale σ by convolving it with the low-pass filter $G_\sigma(x,y)$. Then the Laplacian of the result of this smoothing operation is computed.

(ii) Operation $G_\sigma(x,y) * \nabla^2 I(x,y)$ first computes the Laplacian of the image itself (sum of its second derivatives in the x and y directions), and then the result is smoothed at a scale σ by convolving it with the low-pass filter $G_\sigma(x,y)$.

(iii) Operation $[\nabla^2 G_\sigma(x,y)] * I(x,y)$ first constructs (off-line) a new filter by taking the Laplacian of a Gaussian at a certain scale σ. This new band-pass filter is then convolved with the image as a single operation, to band-pass filter it, isotropically.

(b) By commutativity of linear operators, all the above are equivalent. Their effect is an isotropic band-pass filtering of the image, extracting edge structure within a certain band of spatial frequencies determined by σ, while treating all orientations equally.

Answer to Exercise 7

(a) The gradient vector field $\vec{\nabla} I(x,y)$ of an image $I(x,y)$ is a tuple of partial derivatives associated with each point in the image: $\vec{\nabla} I(x,y) \equiv \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$

(b) This vector field can be used to detect local edges in the image, estimating both their strength and their direction.

(c) The gradient magnitude, estimating edge strength, is: $\|\vec{\nabla} I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

(d) The gradient direction (orientation of an edge) is estimated as: $\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right)$

(e) In the Canny edge detector the following steps are applied, resulting in much cleaner detection of the actual boundaries of objects, with spurious edge clutter eliminated:

1. First the image is smoothed with a Gaussian filter to reduce noise.
2. Then the gradient vector field $\vec{\nabla} I(x,y)$ is computed across the image. The partial derivatives in $\vec{\nabla} I(x,y)$ can be estimated as the first finite differences in x and in y.
3. An "edge thinning" technique, non-maximal suppression, eliminates spurious edges. An edge is represented by a single pixel at which the gradient is maximal.
4. Applying a double threshold to the gradient magnitude enables a triage of edge data, labelling it as strong, weak, or suppressed.
5. A connectivity constraint is applied, by "tracking" detected edges across the image. Edges that are weak and not connected to strong edges are eliminated.

Exercise 8

Consider the following pair of filter kernels:

-1	-1	-1	-1	-1	-1
-1	-3	-4	-4	-3	-1
2	4	5	5	4	2
2	4	5	5	4	2
-1	-3	-4	-4	-3	-1
-1	-1	-1	-1	-1	-1

1	1	1	1	1	1
-1	-2	-3	-3	-2	-1
-1	-3	-4	-4	-3	-1
1	3	4	4	3	1
1	2	3	3	2	1
-1	-1	-1	-1	-1	-1

1. Why do these kernels form approximately a quadrature pair?
2. What is the "DC" response of each of the kernels, and what is the significance of this?
3. To which orientations and to what kinds of image structure are these filters most sensitive?
4. Mechanically how would these kernels be applied directly to an image for filtering or feature extraction?
5. How could their respective Fourier Transforms alternatively be applied to an image, to achieve the same effect as in (4) above?
6. How could these kernels be combined to locate facial features?

Exercise 9

Explain the method of *Active Contours*. What are they used for, and how do they work? What underlying trade-off governs the solutions they generate? How is that trade-off controlled? What mathematical methods are deployed in the computational implementation of Active Contours?

Answer to Exercise 8

1. The two kernels form a quadrature filter pair because they have a 90 degree phase offset. The first is even-symmetric (in fact a cosine-phase discrete Gabor wavelet), and the second is odd-symmetric (in fact it is a sine-phase discrete Gabor wavelet). The two kernels are orthogonal to each other (their inner product = 0).

2. The DC response of each kernel is 0. This means they give no response to uniform areas of an image (where brightness is constant).

3. These filters are most responsive to horizontal structures such as edges, or other modulations (such as fingers) that are horizontal.

4. The kernels would be used by convolving them with an image. Positioned over each pixel in the image, the sum of the products of each tap in the filter with each corresponding pixel in the image would become the new pixel at that point in a new image: the filtered image. (But a DC offset must be added to make it a positive image).

5. Alternatively, the same result could be obtained just by multiplying the discrete Fourier Transform of each kernel with the discrete Fourier Transform of the image, and then taking the inverse discrete Fourier Transform of the product.

6. Taking the modulus (the sum of the squares, pixel by pixel) of the two images that result from convolving a facial image with the two kernels, yields peaks of energy at locations corresponding to the eyes and the mouth when the scale is appropriate, as such facial features are local wavelet-like undulations.

Answer to Exercise 9

Active contours are deformable models for object shapes, with admissibility constraints that implement high-level goals about shapes such as geometry, complexity, classification, and smoothness. The trade-offs in deformable models are parametrically controlled.

We compute a shape model M by minimising an energy functional that is a linear combination of two terms: an *external energy* (measuring how poorly the model fits the data), and an *internal energy* M_{xx}^2 (measuring how squiggly and frenzied the model is):

$$\text{argmin}_{\{M:\lambda\}} \int \left((M - I)^2 + \lambda (M_{xx})^2 \right) d\mathbf{x}$$

where M is the solution and I is the shape data (reduced to vector form $\mathbf{x}$ for simplicity). The first term inside the integral seeks to minimise summed-squared-deviations between the model and the data. The constraints imposed by the second ("smoothness") term cause the model to be more or less willing to bend itself to every invagination of the data. Parameter λ gives us, in effect, a knob to turn for setting how stiff or flexible should our active contour model be. Iterative numerical methods for gradient descent, such as PDEs or annealing, are used to converge upon an optimal (minimal energy) shape model M.

Exercise 10

Give three examples of methodologies or tools used in Computer Vision in which Fourier analysis plays a role, either to solve a problem, or to make a computation more efficient, or to elucidate how and why a procedure works. For each of your examples, clarify the benefit offered by the Fourier perspective or implementation.

Answer to Exercise 10

Any three from the following list would do:

1. Convolution of an image with some operator, for example an edge detection operator or feature detecting operator, is ubiquitous in computer vision. Convolution is computationally costly and slow if done "literally," but it is very efficient if done instead in the Fourier domain. One merely needs to multiply the Fourier transform of the image by the Fourier transform of the operator in question, and then take the inverse Fourier transform to get the desired result. For kernels larger than about (5×5), the benefit is that the Fourier approach is vastly more efficient.

2. The Fourier perspective on edge detection shows that it is really just a kind of frequency-selective filtering, usually high-pass or bandpass filtering. For example, applying the ∇^2 second-derivative operator to an image is equivalent to multiplying its Fourier transform by a paraboloid, $\mu^2 + \nu^2$, which discards low frequencies but emphasises high frequencies, in proportion to their square.

3. Texture detection, and texture segmentation, can be accomplished by 2D spectral (Fourier) analysis. Textures are well-defined by their spatial frequency and orientation characteristics, and these indeed are the polar coordinates of the Fourier plane.

4. Motion can be detected, and its parameters estimated, by exploiting the "Spectral co-planarity theorem" of the 3-D spatio-temporal Fourier transform.

5. Active contours as flexible boundary descriptors ("snakes") can be implemented through truncated Fourier series expansions of the boundary data.