Computer Graphics & Image Processing: Exercises

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Part IB
Part II(General)
Diploma in Computer Science

Introduction

These exercises were, in the main, written by three computer graphics supervisors in 1997. All the credit for these questions goes to Chris Faigle, James Gain, and Jonathon Pfautz. Any mistakes remain the problem of Dr Neil Dodgson, who should have checked them more carefully.

There are four sets of exercises, one for each of the four parts of the course. Each set of exercises contains four sections:

W. Warmups

These are short answer questions that should only take you a few minutes each.

P. Problems

These are geared more towards real exam questions. You should make sure that you can answer these.

E. Old Exam Problems

These are problems from old exams. You should definitely make sure that you can answer these.

F. Further Problems

These are harder problems that will be worth your while to answer if you find the other stuff too easy.

Information for supervisors

Be selective: do not set your supervisees every single question from each part — that would overload them.

Don't feel that you have to give one supervision on each part: choose your questions with regard to when your supervisions fall relative to the lectures.

You can specify questions as: Part/Section/Question. For example: 2/W/1.

Starred exercises (★) have model answers (kept in Jenni Cartwright's office). Doubly starred exercises (★★) (these are all exam questions) have model answers marked according to the official marking scheme.

Part 1: Background

W. Warmups

1. ★ Why is the sky blue? [Hints: Why *might* it be blue? Why are sunsets red? Are the red of a sunset and the blue of the sky related?]
2. ★ Suppose you are designing a user interface for someone who is colorblind. Describe how some user interface of your choice should be suitably modified.
3. ★ Why is it better to look at stars and comets slightly off-center rather than looking directly at them?
4. In a CAD system blue would be a poor choice for the colour of an object being designed. Why is this?

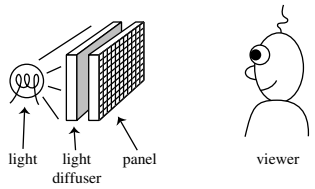
P. Problems

1. ★ **Colour Spaces**. Explain the use of each of the following colour spaces: (a) RGB; (b) XYZ; (c) HLS
2. **Monitor Resolution**. Calculate the ultimate monitor resolution (i.e. colour pixels/inch) at which point better resolution will be indistinguishable.
3. **Display Basics**. The Rainbow Graphics group has developed an Autostereo Device capable of creating "true" three dimensional views by displaying slightly shifted images to each eye. It creates 16 of these views. Assume that the device has the following charecteristics:
   vertical flyback time = 200μs
   image aspect ratio (horizontal:vertical) = 4:3
   pixel clock = 144MHz
   views = 16
   flicker threshold = 50Hz
   uses video interlacing
   (a) Explain each of the terms above.
   (b) Calculate the maximum resolution per view consistent with the above characteristics.
4. **Pixels**. Why do we use square pixels? Would hexagonal pixels be better? What about triangles? Do you see any difficulties building graphics hardware with these other two schemes?
5. **Additive *vs* Subtractive**. Explain the difference between additive colour (RGB) and subtractive colour (CMY). Where is each used and why is it used there?

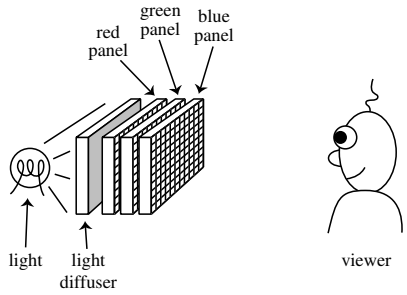
E. Old Exam Problems

1. ★ [94.3.8] (a) Explain how a shadow mask cathode ray tube works. What do you think some of the manufacturing dificulties might be? (b) What might be the point of extending the scheme to accommodate five electron guns?
2. ★★ [97.6.4 *first part*] It is convenient to be able to specify colours in terms of a three-dimensional co-ordinate system. Three such co-ordinate systems are: RGB, HLS, L*a*b*.
   Choose *two* of these three co-ordinate systems.
   For *each* of your chosen two:
   (a) describe what each of the three co-ordinates represents
   (b) describe why the co-ordinate system is a useful representation of colour
3. ★★ [98.6.4 *first part*] An inventor has recently developed a new display device: it is a transparent panel with a rectangular array of square pixels. The panel is tinted with a special ink which allows each pixel to range from totally transparent to transmitting only the colour of the ink. Each pixel has an 8-bit value. For example,

if the ink is blue then a pixel value of 0 would be totally transparent, 255 totally blue (only blue light transmitted) and 100 a light blue.



The inventor has recently found that he can make the special ink in *any* colour he likes, but that each panel can be tinted with only one of these colours. He proposes to use three inks in three panels to make a 24-bit colour display: a red-tinted panel, a green-tinted panel and a blue-tinted panel will be stacked up to make a full-colour display (see picture). A value of (0,0,0) will thus be white (transparent), (255,0,0) red and (255,255,255) black.



Explain why this will not work. [4]
Modify the three-panel design so that it will work. [3]
In common with other 24-bit "full-colour" displays (for example CRT, LCD), your display *cannot* display *every* colour which a human can perceive. Why not? [3]

F. Further Problems

1. ★ **Liquid Crystal Displays**. Explain how a liquid crystal display works.
2. ★ **Head Mounted Display**. Ivan Sutherland described, in 1965, the "Ultimate Display". This display was to match all of the human senses so that the display images were indistinguishable from reality. He went on to build the world's first head-mounted display (HMD) over the next few years. Needless to say, he was far from accomplishing his goal.
   You are to work out the flat-screen pixel resolution (height and width) necessary for an Ultimate HMD (that is, so that the display seems to match the real world) given the following information:
   viewing distance: 10 cm
   human visual acuity: 1 minute of visual arc
   human vertical FOV: 100 deg
   human horizontal FOV: 200 deg
   Give at least five assumptions that have been made in your calculations (and in the question itself!).

Part 2: 2D Computer Graphics

W. Warmups

1. ★ **GUIs**. How has computer graphics been affected by the advent of the graphical user interface?
2. **Matrices**. Give as many reasons as possible why we use matrices to represent transformations.
3. **BitBlt**. What factors do you think affect the efficiency of BitBlt's ability to move images quickly.

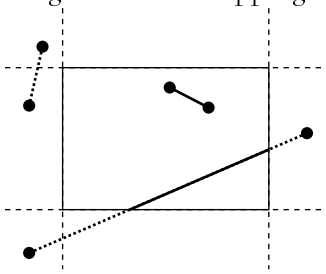
P. Problems

1. ★ **Circle Drawing**. Give the circle drawing algorithm for the *first* octant.
2. **Circle Drawing**. (a) In slide 74 of the 2D Computer Graphics notes Dr. Dodgson writes that one must use points of 45 degree slope to divide the oval into eight sections. Explain what is meant by this and why it is so.
3. ★ **Line Drawing**. Compare the efficiency of Bresenham's and the Midpoint line drawing algorithms for lines of 5, 10, and 20 pixels length (assume that a floating point operation takes 10 times as long as an integer operation).
4. ★ **Subdivision**. If we are subdividing Bézier curves in order to draw them, how do we know when the curve is within a given tolerance? (i.e. how do we know when to stop subdividing?)
5. **Bezier cubics.** Derive the conditions necessary for two Bezier curves to join with (a) just *C0*-continuity; (b) *C1*-continuity; (c) *C2*-continuity. Why would it be difficult (if not impossible) to get three Bezier curves to join in sequence with *C2*-continuity at the two joins?

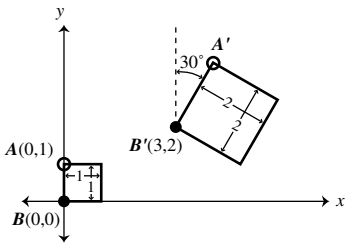
E. Old Exam Problems

1. [96.5.4] Consider the control of detail in a curve represented as a sequence of straight line segments. Describe how Douglas and Pücker's algorithm might be used to remove superfluous points.
   Describe how Overhauser interpolation can be used to introduce additional points.
2. ★ [95.5.4] Why are matrix representations used to describe point transformations in computer graphics?
   Describe how to represent three different 2D transformations as matrices.
   Explain how to derive a sequence of transformations to achieve the overall effect of performing a 2D rotation about an arbitrary point.
3. ★★ [97.4.10] Describe an algorithm to draw a straight line using only integer arithmetic. You may assume that the line is in the first octant, that the line starts and ends at integer co-ordinates, and that the function *setpixel(x,y)* turns on the pixel at location *(x,y)*.
   Explain how straight lines can be used to draw Bezier cubic curves.

4. ★★ [98.5.4] Describe an algorithm for clipping a line against a rectangle. [9]



Show that it works using the above three examples. [3]



The above diagram shows a complicated 2D transformation applied to a unit square. The overall transformation can be described in terms of a number of simpler transformations. Describe each of these simple transformations and give a matrix representation of each using homogeneous coordinates. [6]

Use the matrices from the previous part to find the (x,y) coordinates of point A', the image of point A under the overall transformation. [2]

F. Further Problems

1. **Bernstein Polynomials**. Prove that the Bernstein polynomials sum to 1.
2. **Polygon Filling**. Explain *in detail* how a polygon filling algorithm works.

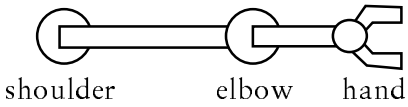
Part 3: 3D Computer Graphics

W. Warmups

1. **Texture Mapping**. Are there any problems texture mapping onto a sphere?
2. *<3/W/2 deleted in 1999 because the course no longer covers this material>*
3. ★ **Shading Model**. Do you see any problems with the naïve shading model? Can you suggest anything to improve its faults?
4. ★ **Illumination**. Describe the following in terms of illumination: ambient, diffuse, specular.
5. **Phong**. Describe how Phong's specular reflection models real specular reflection. Why is it only a rough approximation? Why is it useful?
6. **Level of Detail**. You are given several representations of each object in your 3D scene. When you render the scene, you know all of the world/viewing parameters necessary. Describe several (at least three) ways to determine which representation to choose for each object.
7. ★ **Coordinate Systems**. Draw pictures to show what is meant by:
   object coordinates,
   world coordinates,
   viewing coordinates,
   screen coordinates.
8. ★ **Projections**. Illustrate the difference between orthographic (parallel) and perspective projection.
9. ★ **Triangle mesh approximations**. We use a lot of triangles to approximate stuff in computer graphics. Why are they good? Why are they bad? Can you think of any alternatives?

P. Problems

1. **Projection**. Draw and explain two different scenes which have the same projection as seen by the viewer. What other cues can you give so that the viewer can distinguish the depth information.
2. **3D Transformations**. The Rainbow Graphics Group has available a simple robot arm. This robot arm has, for the purposes of this question, the following characteristics:
   • Two joints, a shoulder joint and an elbow joint. The shoulder joint is connected by a 2-unit length upper arm to the elbow joint, which in turn is connected by a single unit lower arm to the hand.



   • The joints can only rotate in the xy-plane.
   • The shoulder joint is attached to a z-axis vertical slider, which can raise or lower the entire robot arm, but is not able to translate it in the xy-plane.
   • The initial position of the arm is:
      shoulder joint: position (0,0,0), rotation 0°
      elbow joint: position (2,0,0), rotation 0°
      hand: position (3,0,0)

   There is a soft drink can located at position (1,1,1). The robot hand must touch this can. Specify the transformation matrices of the joints needed to achieve this.
3. **Perspective Projection Geometry**. Develop the Perspective Projection Geometry for stereo views. Assume that the left and right eye-points are at $(-e/2, 0, 0)$ and $(e/2, 0, 0)$ respectively.

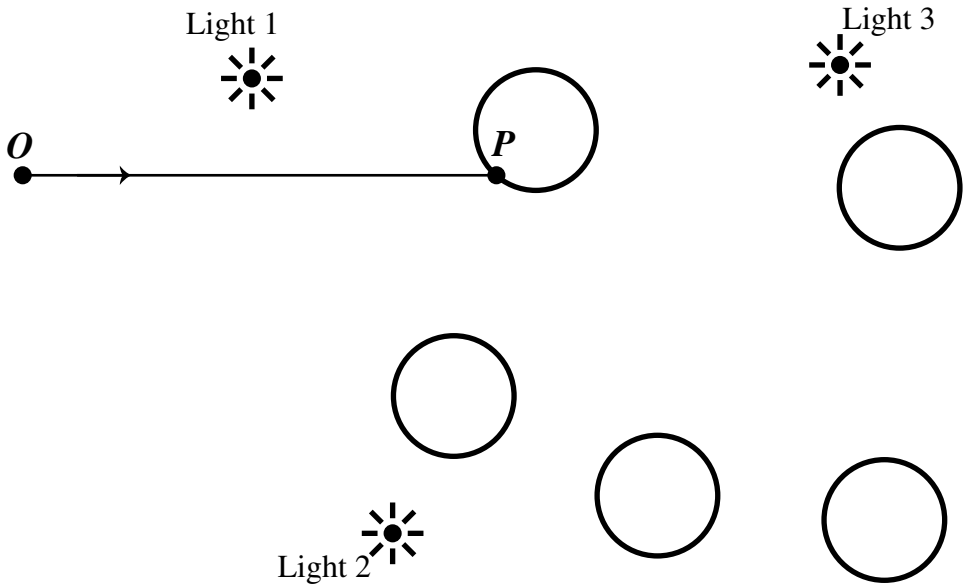
4. ★ **Bounding Volumes**. For a cylinder of radius 2, with endpoints (1,2,3) and (2,4,5), show how to calculate: (a) an axis-aligned bound box, (b) a bounding sphere.
5. **Depth Interpolation**. Prove that depth interpolation is correct as given in the notes.
6. ★ **BSP Tree**. Break down the following (2D!) lines into a BSP-tree, splitting them if necessary.
   (0, 0)→(2, 2), (3,4)→(1, 3), (1, 0)→(-3, 1), (0, 3)→(3, 3) and (2, 0)→(2, 1)
7. ★ **Sphere Subdividing**. We often use triangles to represent a sphere. Describe two methods of generating triangles from a sphere.
8. ★ **Compare and Contrast**. Compare and contrast: *texture mapping*, *bump mapping* and *displacement mapping*.
9. **Shading**. Develop pseudocode for algorithms that shade a triangle according to:
   (a) Gouraud shading.
   (b) Phong shading.
   You can use the scan line polygon fill algorithm as a starting point. Detail your inputs and outputs explicitly.
10. **Rendering**. Compare the computation and memory requirements of the following rendering algorithms:
    (a) z-Buffer
    (b) Ray tracing
    State explicitly any assumptions that you make (e.g. the resolution of the screen). Present your results both numerically and in graph form (with number of polygons on the x-axis).
11. ★ **Bézier Joins**. Explain the three types of joins for Bézier curves and Bézier patches.
12. ★ **3D Clipping**. (a) Compare the two methods of doing 3D clipping in terms of efficiency. (b) How would using bounding volumes improve the efficiency of these methods?
13. ★ **Rotation**.
    Show how to perform 2D rotation around an arbitrary point.
    Show how to perform 3D rotation around an arbitrary axis parallel to the x-axis.
    Show how to perform 3D rotation around an arbitrary axis.

E. Old Exam Problems

1. ★ [96.6.4] What are *homogeneous coordinates*? How can they be used in computer graphics to model (a) translation? and (b) simple perspective?
2. ★ [95.4.8] (a) Explain the purpose of the A-buffer in rendering a sequence of images into the framestore. (b) Exhibit an example that shows an advantage over the use of a z-buffer.
3. ★ [95.6.4] In ray tracing a large computational cost is associated with determining ray-object intersections. Explain how the use of bounding volumes and space subdivision methods may reduce this cost.
4. ★ [94.5.4] (a) Discuss sampling artifacts and their effect on image quality on a raster display. (b) What can be done to reduce or eliminate them?
5. ★★ [97.5.2] Describe the z-buffer polygon scan conversion algorithm. Explain how the A-buffer improves on the z-buffer.

6. ★★ [98.4.10] In ray tracing, *ambient*, *diffuse* and *Phong's specular* shading can be used to define the colour at a point on a surface. Explain what each of the three terms refers to, and what real effect each is trying to model. [9]
The diagram below represents a scene being ray traced. The circles may be taken to represent the cross-sections of spheres. In answering the remaining parts of this question you should use copies of the diagram below (see page 10).
A particular ray from the eyepoint ***O*** has been found to have its closest intersection with an object at point ***P***. Show, on a diagram, all subsequent rays and vectors which must be found in order to calculate the shading at point ***P***. Explain the purpose of each one.
Assume that:
   - each object has ambient, diffuse and specular reflections, but is *not* a perfect reflector
   - each object is opaque
   - all rays and vectors lie in the plane of the paper
   - we are *not* using distributed ray tracing [8]

   Assume now that all of the objects are perfect reflectors (in addition to having ambient, diffuse and specular reflection). Show, on a separate diagram, the extra rays which need to be calculated and explain the purpose of each one. [3]



[Note: in 1998 about half of the attempts at this question were wildly wrong. Many assumed (although they did not explicitly say so) that the computer performed some sort of magic in order to work out the shading at point ***P***. Please, therefore, check your answer to ensure that a computer would actually be able to perform the calculations that you draw on your diagrams.]

F. Further Problems

1. **Snell's Laws**. Look up Snell's laws and address how they relate to computer graphics.
2. **Ray Tracing Bézier Patches**. Can you suggest how we might ray-trace a Bézier patch?
3. **Bézier Patches**. Describe how you would form a *C1* continuous cylinder from Bézier patches. Draw the patches and their control points and give the co-ordinates of the control points.
4. **Bézier Patches**. Given the following sixteen points, calculate the first eight of the next patch joining it as t increases so that the join has continuity *C1*. Here the

points are listed with *s=0*, *t=0* on the bottom left, with s increasing upwards and *t* increasing to the right.

(-.2, 3.4, .3)	(1, 3.1, -.2)	(2, 2.6, -.2)	(3.1, 2.8, .2)
(0, 1.2, .4)	(1.2, 2.0, 1.2)	(1.4, 1.9, -.2)	(2.7, 1.8, .2)
(.2, 1, -.2)	(1.1, .8, .5)	(1.4, 1.0, 0)	(3.1, 1.1, -.2)
(0, 0, 0)	(1, 0, .5)	(2, .2, .4)	(2.7, 0, -.2)

5. **Web 3D Language**. Describe some features that you think might be important in a language designed to describe 3D scenes and be used across the web.
6. **Rotations**. Define and then compare and contrast the following methods of specifying rotation in 3D. [you will need to look these up]
   (a) Quaternions
   (b) Euler Angles
7. **DOOM-Style Rendering**. Describe the enhancements and restrictions that the DOOM rendering engine employs to improve efficiency in the interests of interactivity. Define any terms (such as texture-mapping) that you might use. A useful starting point for your research is <http://www.gamers.org/dEngine/doom/>.

Part 4: Image Processing

P. Problems

1. **Image Coding Schemes**. Describe the following image encoding schemes:
   (a) GIF
   (b) JPEG
   Show their operation with a suitable small sample image. Compare and contrast them (note that GIF file encoding is not in the notes so this requires some research).
2. **Image Coding**. Explain each of the three stages of image coding, as presented in the notes, and why each helps to reduce the number of bits required to store the image.
3. **Error Diffusion**. Compare the two methods of Error Diffusion described in the notes, with the aid of a sample image.

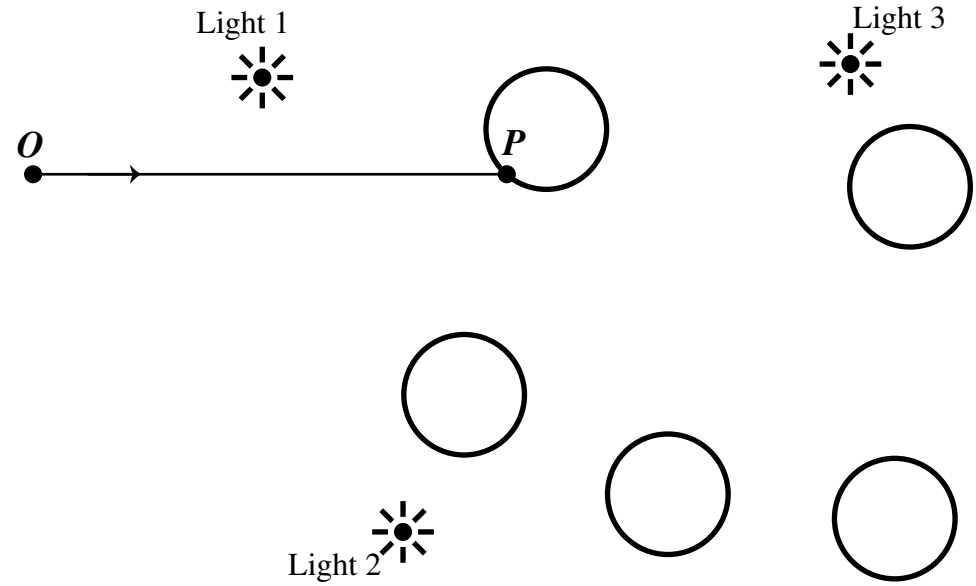
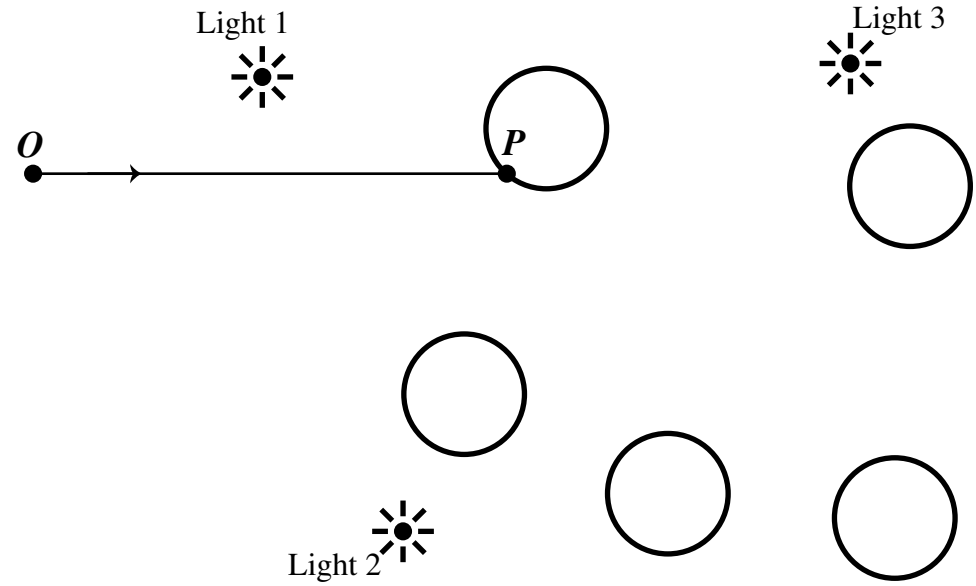
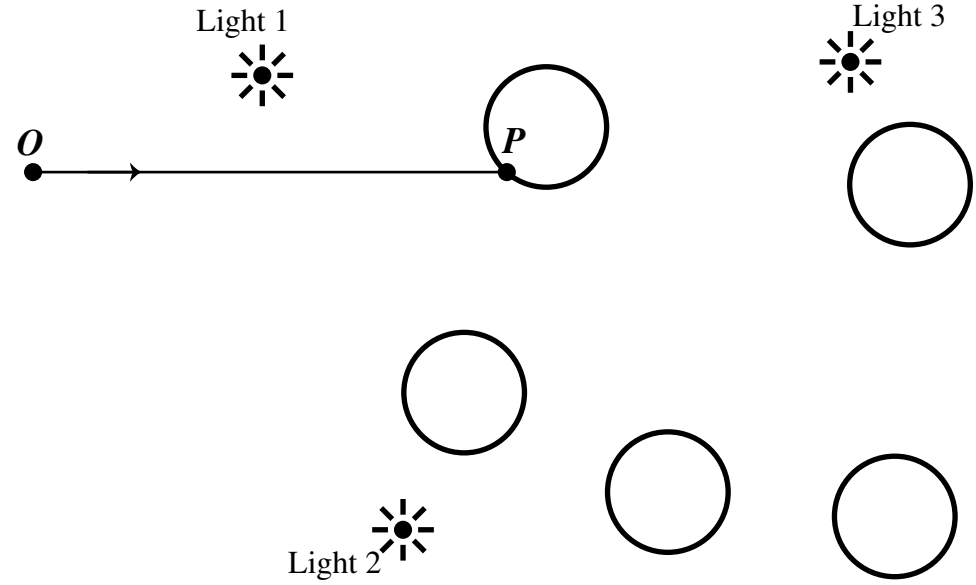
E. Old Exam Questions

1. ★ [96.4.10] An image processing package allows the user to design 3×3 convolution filters. Design 3×3 filters to perform the following tasks:
   (a) Blurring
   (b) Edge detection of vertical edges
   Choose one of the two filters (a) or (b) from the previous part. Explain how it works, using the following image as an example (you may round off any calculated values to the nearest integer).

100	100	100	0	0	0
100	100	100	0	0	0
100	100	100	0	0	0
100	100	100	100	100	100
100	100	100	100	100	100
100	100	100	100	100	100

2. ★★ [97.6.4 *second part*] Draw *either*:
   (i) the first eight one-dimensional Haar basis functions
   or
   (ii) the first eight one-dimensional Walsh-Hadamard basis functions
   Calculate the co-efficients of your eight basis functions from the previous part for the following one-dimensional image data:

   12 16 20 24 24 16 8 8

   Explain why, in general, the Haar or Walsh-Hadamard encoded version of an image is preferable to the original image for storage or transmission.
3. ★★ [98.6.4 *second part*] In image compression we utilise three different mechanisms to compress pixel data:
   (a) mapping the pixel values to some other set of values
   (b) quantising those values
   (c) symbol encoding the resulting values
   Explain each mechanism, why it helps us to compress the image, and whether (giving reasons) the resulting image noticeably differs. [10]

Copies of the diagram for answering 3/E/6 [96.4.10]