

## Mid-Level Vision: Segmentation (Artificial)

### 1. Briefly describe the role of mid-level vision.

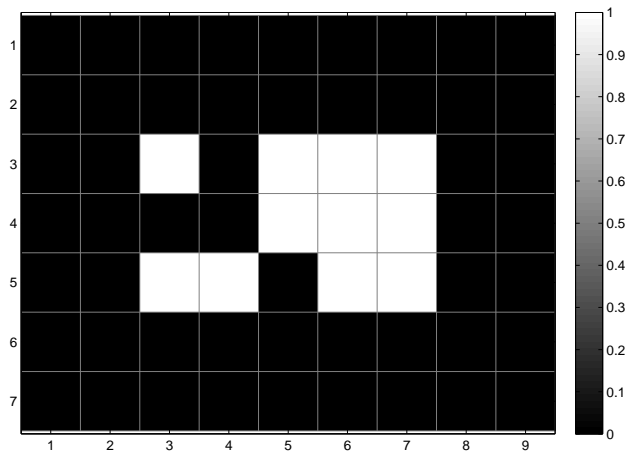
*To group together image elements that belong together, and to segment them from all other image elements.*

### 2. One simple method of segmentation is thresholding. (a) Briefly explain how an appropriate threshold might be obtained by using an intensity histogram. (b) Explain the procedure used when applying hysteresis thresholding.

*(a) This method assumes that the histogram of pixel intensities will have two peaks. The threshold is placed (mid-way) between those peaks.*

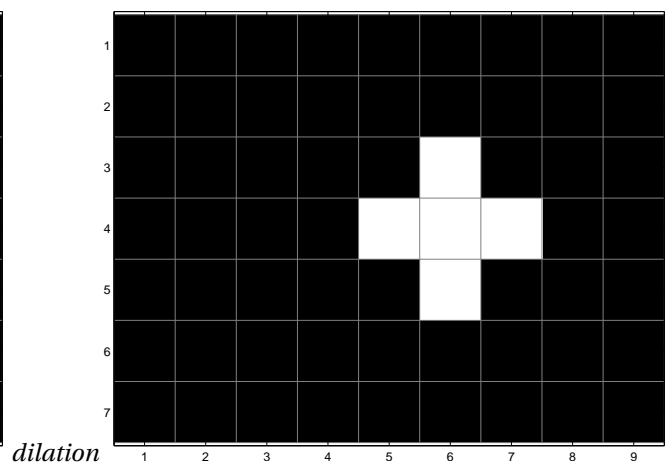
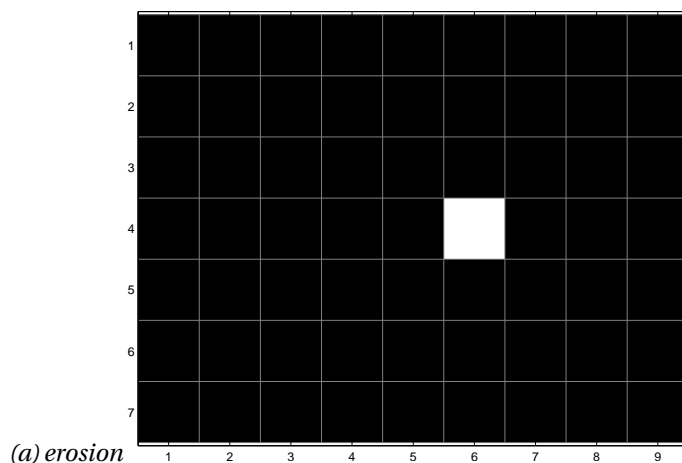
*(b) This method requires two thresholds to be defined (a high threshold and a low threshold). These may be obtained using the intensity histogram by placing the thresholds on either side of the valley between the two peaks. Pixels above the high threshold are classified as figure and below the low threshold as background. Pixels between the low and high thresholds are classified as figure only if they are adjacent to other figure pixels.*

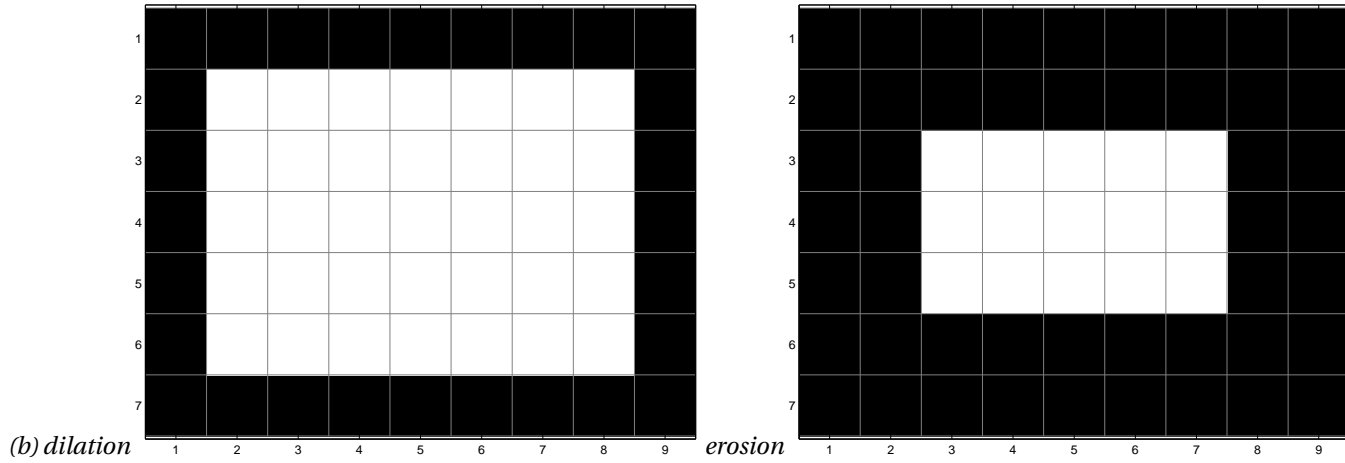
### 3. The figure shows a 7 by 9 pixel binary image.



(a) Draw the image that results from performing erosion followed by dilation on this image, assuming each pixel has 4 (horizontal and vertical) neighbours.

(b) Draw the image that results from performing dilation followed by erosion on this image, assuming each pixel has 8 (horizontal, vertical and diagonal) neighbours.





**4. Write pseudo-code for the region growing algorithm.**

1. While not all pixels have been assigned a region label
2. Choose an unlabelled pixel at random
3. Assign chosen pixel the next unused region label
4. For all neighbouring pixels which haven't been assigned a label
5. Calculate similarity to chosen pixel
6. For each pixel within the similarity threshold
7. Give that pixel the same region label as the chosen pixel
8. Make that pixel the chosen pixel and repeat from step 4
9. Repeat from step 1

**5. The array below shows feature vectors for each pixel in a 3 by 3 image.**

$$\begin{bmatrix} (5, 10, 15) & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) & (5, 20, 15) & (10, 5, 30) \\ (5, 5, 15) & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Apply the region growing algorithm to assign each pixel to a region. Assume that (1) the method used to assess similarity is the sum of absolute differences (SAD), (2) the criteria for deciding if elements are similar is that the SAD is less than 12, (3) the seed pixel is the top-left corner, (4) pixels have horizontal, vertical and diagonal neighbours.

Label top-left pixel 1. Calculate SAD values for pixels neighbouring the top-left pixel:

$$\begin{bmatrix} (5, 10, 15) : 1 & \begin{matrix} 25 \rightarrow \\ 5 \downarrow \\ 10 \searrow \end{matrix} & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) : 1 & & (5, 20, 15) : 1 & (10, 5, 30) \\ (5, 5, 15) & & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Calculate SAD values to unlabelled pixels from middle-left pixel:

$$\begin{bmatrix} (5, 10, 15) : 1 & \begin{matrix} 25 \rightarrow \\ 5 \downarrow \\ 10 \searrow \end{matrix} & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) : 1 & & (5, 20, 15) : 1 & (10, 5, 30) \\ \begin{matrix} 10 \downarrow \\ 30 \searrow \end{matrix} & & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Calculate SAD values to unlabelled pixels from middle-middle pixel:

$$\begin{bmatrix} (5, 10, 15) : 1 & \begin{matrix} 25 \rightarrow \\ 5 \downarrow \\ 10 \searrow \end{matrix} & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) : 1 & & (5, 20, 15) : 1 & \begin{matrix} 25 \uparrow \\ 35 \rightarrow \end{matrix} \\ \begin{matrix} 10 \downarrow \\ 30 \searrow \end{matrix} & & \begin{matrix} 45 \downarrow \\ 40 \searrow \end{matrix} & (10, 5, 30) \\ (5, 5, 15) : 1 & & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Calculate SAD values to unlabelled pixels from left-bottom pixel:

$$\begin{bmatrix} (5,10,15):1 & \mathbf{25} \rightarrow & (10,15,30) & (10,10,25) \\ \mathbf{5} \downarrow & \mathbf{10} \searrow & \mathbf{25} \uparrow & \mathbf{25} \nearrow \\ (10,10,15):1 & & (5,20,15):1 & \mathbf{35} \rightarrow (10,5,30) \\ \mathbf{10} \downarrow & \mathbf{30} \searrow & \mathbf{45} \downarrow & \mathbf{40} \searrow \\ (5,5,15):1 & \mathbf{40} \rightarrow & (30,10,5) & (30,10,10) \end{bmatrix}$$

Region 1 cannot grow further. Choose another seed pixel: middle-top:

$$\begin{bmatrix} 1 & (10,15,30):2 & \mathbf{10} \rightarrow & (10,10,25):2 \\ & & \mathbf{10} \searrow & \\ 1 & 1 & & (10,5,30):2 \\ 1 & (30,10,5) & & (30,10,10) \end{bmatrix}$$

Calculate SAD values to unlabelled pixels from right-middle pixel:

$$\begin{bmatrix} 1 & (10,15,30):2 & \mathbf{10} \rightarrow & (10,10,25):2 \\ & & \mathbf{10} \searrow & \\ 1 & 1 & & (10,5,30):2 \\ 1 & (30,10,5) & \mathbf{50} \nearrow & \mathbf{45} \downarrow \\ & & & (30,10,10) \end{bmatrix}$$

Region 2 cannot grow further. Choose another seed pixel: bottom-right:

$$\begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & (30,10,5):3 & \mathbf{5} \leftarrow (30,10,10):3 \end{bmatrix}$$

Region labels:  $\begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & 3 & 3 \end{bmatrix}$

## 6. Write pseudo-code for the region merging algorithm.

1. Give every pixel a unique region label
2. Randomly choose a region which has not been marked final
  3. Compare the chosen region's properties with those of its neighbours
  4. For all regions with properties which are within the similarity threshold
    5. Re-label them with the region label of the chosen region
    6. Calculate properties of the merged region as mean of all its constituents
  7. If no regions can be merged with chosen region, mark it as final
8. Repeat from step 2 until all image regions are marked final.

## 7. The array below shows feature vectors for each pixel in a 3 by 3 image.

$$\begin{bmatrix} (5,10,15) & (10,15,30) & (10,10,25) \\ (10,10,15) & (5,20,15) & (10,5,30) \\ (5,5,15) & (30,10,5) & (30,10,10) \end{bmatrix}$$

Apply the region merging algorithm to assign each pixel to a region. Assume that (1) the method used to assess similarity is the sum of absolute differences (SAD), (2) the criteria for deciding if regions are similar is that the SAD is less than 12, (3) the first chosen region is the top-left corner, (4) regions have horizontal, vertical and diagonal neighbours.

Give each pixel a unique label. Calculate SAD values for regions neighbouring the top-left region:

$$\begin{bmatrix} (5,10,15):1 & \mathbf{25} \rightarrow & (10,15,30):2 & (10,10,25):3 \\ \mathbf{5} \downarrow & \mathbf{10} \searrow & & \\ (10,10,15):4 & & (5,20,15):5 & (10,5,30):6 \\ (5,5,15):7 & & (30,10,5):8 & (30,10,10):9 \end{bmatrix}$$

Merge similar regions, and calculate properties of the merged region:

$$\begin{bmatrix} (6.67, 13.33, 15) : 1 & (10, 15, 30) : 2 & (10, 10, 25) : 3 \\ (6.67, 13.33, 15) : 1 & (6.67, 13.33, 15) : 1 & (10, 5, 30) : 6 \\ (5, 5, 15) : 7 & (30, 10, 5) : 8 & (30, 10, 10) : 9 \end{bmatrix}$$

Randomly choose region 1, and calculate SAD values for neighbouring regions:

$$\begin{bmatrix} (6.67, 13.33, 15) : 1 & (10, 15, 30) : 2 & (10, 10, 25) : 3 \\ (6.67, 13.33, 15) : 1 & \mathbf{20} \uparrow & \mathbf{16.67} \nearrow & (10, 5, 30) : 6 \\ \mathbf{10} \downarrow & (6.67, 13.33, 15) : 1 & \mathbf{26.67} \rightarrow & \\ \mathbf{36.67} \downarrow & (30, 10, 5) : 8 & \mathbf{31.67} \searrow & (30, 10, 10) : 9 \end{bmatrix}$$

Merge similar regions, and calculate properties of the merged region:

$$\begin{bmatrix} (6.25, 11.25, 15) : 1 & (10, 15, 30) : 2 & (10, 10, 25) : 3 \\ (6.25, 11.25, 15) : 1 & (6.25, 11.25, 15) : 1 & (10, 5, 30) : 6 \\ (6.25, 11.25, 15) : 1 & (30, 10, 5) : 8 & (30, 10, 10) : 9 \end{bmatrix}$$

Randomly choose region 1, and calculate SAD values for neighbouring regions:

$$\begin{bmatrix} (6.25, 11.25, 15) : 1 & (10, 15, 30) : 2 & (10, 10, 25) : 3 \\ (6.25, 11.25, 15) : 1 & \mathbf{22.5} \uparrow & \mathbf{15} \nearrow & (10, 5, 30) : 6 \\ \mathbf{35} \downarrow & (6.25, 11.25, 15) : 1 & \mathbf{25} \rightarrow & \\ \mathbf{30} \searrow & (30, 10, 5) : 8 & & (30, 10, 10) : 9 \end{bmatrix}$$

Mark region 1 as final. Randomly choose region 2 and calculate SAD values for neighbouring regions:

$$\begin{bmatrix} 1 & (10, 15, 30) : 2 & \mathbf{10} \rightarrow & (10, 10, 25) : 3 \\ & & \mathbf{10} \searrow & \\ 1 & 1 & & (10, 5, 30) : 6 \\ 1 & (30, 10, 5) : 8 & & (30, 10, 10) : 9 \end{bmatrix}$$

Merge similar regions, and calculate properties of the merged region:

$$\begin{bmatrix} 1 & (10, 10, 28.33) : 2 & (10, 10, 28.33) : 2 \\ 1 & 1 & (10, 10, 28.33) : 2 \\ 1 & (30, 10, 5) : 8 & (30, 10, 10) : 9 \end{bmatrix}$$

Randomly choose region 2, and calculate SAD values for neighbouring regions:

$$\begin{bmatrix} 1 & (10, 10, 28.33) : 2 & (10, 10, 28.33) : 2 \\ 1 & 1 & (10, 10, 28.33) : 2 \\ 1 & (30, 10, 5) : 8 & \mathbf{43.33} \nearrow & \mathbf{38.33} \downarrow & (30, 10, 10) : 9 \end{bmatrix}$$

Mark region 2 as final. Randomly choose region 8 and calculate SAD values for neighbouring regions:

$$\begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & (30, 10, 5) : 8 & \mathbf{5} \rightarrow & (30, 10, 10) : 9 \end{bmatrix}$$

Merge similar regions, and mark region 8 as final:

$$\begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & 8 & 8 \end{bmatrix}$$

**8. Write pseudo-code for the split and merge algorithm.**

1. Label every pixel with the same region label
2. For each region
  3. If all pixels are not similar
    4. Assign the four quadrants different region labels
5. Repeat from 2 until all regions homogeneous
6. For each region
  7. Compare the region's properties with those of its neighbours
  8. For all regions with properties which are within the similarity threshold
    9. Re-label them with the region label of the chosen region
    10. Calculate properties of the merged region as mean of all its constituents
11. Repeat from step 6 until no more regions can be merged.

**9. The array below shows feature vectors for each pixel in a 3 by 3 image.**

$$\begin{bmatrix} (5, 10, 15) & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) & (5, 20, 15) & (10, 5, 30) \\ (5, 5, 15) & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Apply the split and merging algorithm to assign each pixel to a region. Assume that (1) the method used to assess similarity is the sum of absolute differences (SAD), (2) the criteria for deciding if regions are similar is that the SAD is less than 12, (3) regions have horizontal, vertical and diagonal neighbours.

Give each pixel the same label. Determine if SAD value for any pair of pixels within the the region exceeds 12:

$$\begin{bmatrix} (5, 10, 15) : 1 & \mathbf{25} \rightarrow & (10, 15, 30) : 1 & (10, 10, 25) : 1 \\ (10, 10, 15) : 1 & & (5, 20, 15) : 1 & (10, 5, 30) : 1 \\ (5, 5, 15) : 1 & & (30, 10, 5) : 1 & (30, 10, 10) : 1 \end{bmatrix}$$

Split the region into quadrants (pad image to do so):

$$\begin{bmatrix} (5, 10, 15) : 1 & (10, 15, 30) : 1 & (10, 10, 25) : 2 & (10, 10, 25) : 2 \\ (10, 10, 15) : 1 & (5, 20, 15) : 1 & (10, 5, 30) : 2 & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ (5, 5, 15) : 3 & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

For region 1, determine if SAD value for any pair of pixels within the region exceeds 12:

$$\begin{bmatrix} (5, 10, 15) : 1 & \mathbf{25} \rightarrow & (10, 15, 30) : 1 & (10, 10, 25) : 2 & (10, 10, 25) : 2 \\ \mathbf{5} \downarrow & \mathbf{10} \nearrow & \mathbf{25} \downarrow & & \\ (10, 10, 15) : 1 & \mathbf{15} \rightarrow & (5, 20, 15) : 1 & (10, 5, 30) : 2 & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ (5, 5, 15) : 3 & & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

Split region 1 into quadrants:

$$\begin{bmatrix} (5, 10, 15) : 1 & (10, 15, 30) : 5 & (10, 10, 25) : 2 & (10, 10, 25) : 2 \\ (10, 10, 15) : 6 & (5, 20, 15) : 7 & (10, 5, 30) : 2 & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ (5, 5, 15) : 3 & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

For region 2, determine if SAD value for any pair of pixels within the region exceeds 12:

$$\begin{bmatrix} (5, 10, 15) : 1 & (10, 15, 30) : 5 & (10, 10, 25) : 2 & \mathbf{0} \rightarrow & (10, 10, 25) : 2 \\ & & \mathbf{10} \downarrow & \mathbf{10} \nearrow & \mathbf{10} \downarrow \\ (10, 10, 15) : 6 & (5, 20, 15) : 7 & (10, 5, 30) : 2 & \mathbf{0} \rightarrow & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & (30, 10, 5) : 3 & (30, 10, 10) : 4 & & (30, 10, 10) : 4 \\ (5, 5, 15) : 3 & (30, 10, 5) : 3 & (30, 10, 10) : 4 & & (30, 10, 10) : 4 \end{bmatrix}$$

For region 3, determine if SAD value for any pair of pixels within the region exceeds 12:

$$\begin{bmatrix} (5, 10, 15) : 1 & (10, 15, 30) : 5 & (10, 10, 25) : 2 & (10, 10, 25) : 2 \\ (10, 10, 15) : 6 & (5, 20, 15) : 7 & (10, 5, 30) : 2 & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & \mathbf{40} \rightarrow & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ & \mathbf{0} \downarrow & \mathbf{40} \searrow & \mathbf{0} \downarrow & \\ (5, 5, 15) : 3 & \mathbf{40} \rightarrow & (30, 10, 5) : 3 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

Split region 3 into quadrants:

$$\begin{bmatrix} (5, 10, 15) : 1 & (10, 15, 30) : 5 & (10, 10, 25) : 2 & (10, 10, 25) : 2 \\ (10, 10, 15) : 6 & (5, 20, 15) : 7 & (10, 5, 30) : 2 & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & (30, 10, 5) : 8 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ (5, 5, 15) : 9 & (30, 10, 5) : 10 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

For region 4, determine if SAD value for any pair of pixels within the region exceeds 12:

$$\begin{bmatrix} (5, 10, 15) : 1 & (10, 15, 30) : 5 & (10, 10, 25) : 2 & (10, 10, 25) : 2 \\ (10, 10, 15) : 6 & (5, 20, 15) : 7 & (10, 5, 30) : 2 & (10, 5, 30) : 2 \\ (5, 5, 15) : 3 & (30, 10, 5) : 8 & (30, 10, 10) : 4 & \mathbf{0} \rightarrow & (30, 10, 10) : 4 \\ & & \mathbf{0} \downarrow & \mathbf{0} \searrow & \mathbf{0} \downarrow \\ (5, 5, 15) : 9 & (30, 10, 5) : 10 & (30, 10, 10) : 4 & \mathbf{0} \rightarrow & (30, 10, 10) : 4 \end{bmatrix}$$

Can not split any further. So perform merge operation for region 1 (using mean properties for each region):

$$\begin{bmatrix} (5, 10, 15) : 1 & \mathbf{25} \rightarrow & (10, 15, 30) : 5 & (10, 7.5, 27.5) : 2 & (10, 7.5, 27.5) : 2 \\ & \mathbf{5} \downarrow & \mathbf{10} \searrow & & \\ (10, 10, 15) : 1 & & (5, 20, 15) : 1 & (10, 7.5, 27.5) : 2 & (10, 7.5, 27.5) : 2 \\ (5, 5, 15) : 3 & & (30, 10, 5) : 8 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ (5, 5, 15) : 9 & & (30, 10, 5) : 10 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

Continue merging region 1:

$$\begin{bmatrix} (6.67, 13.33, 15) : 1 & (10, 15, 30) : 5 & & (10, 7.5, 27.5) : 2 & (10, 7.5, 27.5) : 2 \\ & \mathbf{20} \uparrow & \mathbf{21.67} \nearrow & & \\ (6.67, 13.33, 15) : 1 & (6.67, 13.33, 15) : 1 & \mathbf{21.67} \rightarrow & (10, 7.5, 27.5) : 2 & (10, 7.5, 27.5) : 2 \\ & \mathbf{10} \downarrow & \mathbf{36.67} \downarrow & \mathbf{31.67} \searrow & \\ (5, 5, 15) : 1 & (30, 10, 5) : 8 & & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ (5, 5, 15) : 9 & (30, 10, 5) : 10 & & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{bmatrix}$$

Continue merging region 1:

$$\left[ \begin{array}{cccc} (6.25, 11.25, 15) : 1 & (10, 15, 30) : 5 & (10, 7.5, 27.5) : 2 & (10, 7.5, 27.5) : 2 \\ & \mathbf{22.5} \uparrow & \mathbf{20} \nearrow & \\ (6.25, 11.25, 15) : 1 & (6.25, 11.25, 15) : 1 & \mathbf{20} \rightarrow & (10, 7.5, 27.5) : 2 \\ & \mathbf{35} \downarrow & \mathbf{30} \searrow & \\ (6.25, 11.25, 15) : 1 & (30, 10, 5) : 8 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ & \mathbf{7.5} \downarrow & \mathbf{35} \searrow & \\ (5, 5, 15) : 1 & (30, 10, 5) : 10 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{array} \right]$$

Continue merging region 1:

$$\left[ \begin{array}{cccc} (6, 10, 15) : 1 & (10, 15, 30) : 5 & (10, 7.5, 27.5) : 2 & (10, 7.5, 27.5) : 2 \\ & \mathbf{24} \uparrow & \mathbf{19} \nearrow & \\ (6, 10, 15) : 1 & (6, 10, 15) : 1 & \mathbf{19} \rightarrow & (10, 7.5, 27.5) : 2 \\ & \mathbf{34} \downarrow & \mathbf{29} \searrow & \\ (6, 10, 15) : 1 & (30, 10, 5) : 8 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \\ & \mathbf{34} \searrow & & \\ (6, 10, 15) : 1 & (30, 10, 5) : 10 & (30, 10, 10) : 4 & (30, 10, 10) : 4 \end{array} \right]$$

Mark region 1 as final. Continue merging with another region...

###### 10. Write pseudo-code for the k-means clustering algorithm.

1. Randomly choose  $k$  points to act as cluster centres
2. For each data point
  3. Calculate its similarity to each cluster centre
  4. Allocate data point to closest cluster centre
5. Repeat from step 2 for each data point
6. For each cluster centre
  7. Calculate its new position as the mean position its elements
8. Repeat from step 6 for each cluster centre
9. Repeat from step 2 until the cluster centres are unchanged

###### 11. The array below shows feature vectors for each pixel in a 3 by 3 image.

$$\begin{bmatrix} (5, 10, 15) & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) & (5, 20, 15) & (10, 5, 30) \\ (5, 5, 15) & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Apply the k-means clustering algorithm to assign each pixel to a region. Assume that (1) the method used to assess similarity is the sum of absolute differences (SAD), (2) there are two clusters, and the cluster centres are initially positioned at these positions in feature space: (5, 10, 15) and (10, 10, 25).

Calculate distance of each data point to each cluster centre:

$$\text{distance from } (5, 10, 15): \begin{bmatrix} 0 & 25 & 15 \\ 5 & 10 & 25 \\ 5 & 35 & 30 \end{bmatrix} \quad \text{distance from } (10, 10, 25): \begin{bmatrix} 15 & 10 & 0 \\ 10 & 25 & 10 \\ 20 & 40 & 35 \end{bmatrix}$$

$$\text{Allocate each data point to closest cluster centre: } \begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

Calculate new cluster centres as mean of elements belonging to that cluster:

$$c_1 = [(5, 10, 15) + (10, 10, 15) + (5, 20, 15) + (5, 5, 15) + (30, 10, 5) + (30, 10, 10)] / 6 = (14.167, 10.833, 12.50)$$

$$c_2 = [(10, 15, 30) + (10, 10, 25) + (10, 5, 30)] / 3 = (10, 10, 28.33)$$

Calculate distance of each data point to each cluster centre:

$$\text{distance from } (14.167, 10.833, 12.50): \begin{bmatrix} 12.5 & 25.834 & 17.5 \\ 7.5 & 20.834 & 27.5 \\ 17.5 & 24.166 & 19.166 \end{bmatrix}$$

$$\text{distance from } (10, 10, 28.33): \begin{bmatrix} 18.33 & 6.67 & 3.33 \\ 13.33 & 28.33 & 6.67 \\ 23.33 & 43.33 & 38.33 \end{bmatrix}$$

Allocate each data point to closest cluster centre:  $\begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$

Allocation is unchanged, so cluster centres are also unchanged and algorithm has converged.

## 12. Write pseudo-code for the agglomerative hierarchical clustering algorithm.

1. Assign each data point to a unique cluster
2. Compute the similarity between each pair of clusters (store this in a proximity matrix)
3. Repeat
  4. Merge the two closest clusters
  5. Update the proximity matrix
6. Until only a single cluster remains

## 13. There are a number of methods for calculating the distance between clusters in the agglomerative hierarchical clustering algorithm. Briefly describe each of the following methods. (a) single-link clustering, (b) complete-link clustering, (c) group-average clustering, (d) centroid clustering.

(a) single-link clustering

Distance between clusters is shortest distance between elements (MIN distance)

(b) complete-link clustering

distance between clusters is longest distance between elements (MAX distance)

(c) group-average clustering

distance between clusters is average distance between all pairs of elements (AVERAGE distance)

(d) centroid clustering distance between clusters is distance between their averages.

## 14. The array below shows feature vectors for each pixel in a 3 by 3 image.

$$\begin{bmatrix} (5, 10, 15) & (10, 15, 30) & (10, 10, 25) \\ (10, 10, 15) & (5, 20, 15) & (10, 5, 30) \\ (5, 5, 15) & (30, 10, 5) & (30, 10, 10) \end{bmatrix}$$

Apply the agglomerative hierarchical clustering algorithm to assign pixels in to three regions. Assume that (1) the method used to assess similarity is the sum of absolute differences (SAD), (2) centroid clustering is used to calculate the distance between clusters.

Each point is a separate cluster initially. Compute the distance between each pair of clusters:

$$\begin{bmatrix} & c1: & c2: & c3: & c4: & c5: & c6: & c7: & c8: & c9: \\ c1: (5, 10, 15) & 0 & & & & & & & & \\ c2: (10, 15, 30) & 25 & 0 & & & & & & & \\ c3: (10, 10, 25) & 15 & 10 & 0 & & & & & & \\ c4: (10, 10, 15) & 5 & 20 & 10 & 0 & & & & & \\ c5: (5, 20, 15) & 10 & 25 & 25 & 15 & 0 & & & & \\ c6: (10, 5, 30) & 25 & 10 & 10 & 20 & 35 & 0 & & & \\ c7: (5, 5, 15) & 5 & 30 & 20 & 10 & 15 & 20 & 0 & & \\ c8: (30, 10, 5) & 35 & 50 & 40 & 30 & 45 & 50 & 40 & 0 & \\ c9: (30, 10, 10) & 30 & 45 & 35 & 25 & 40 & 45 & 35 & 5 & 0 \end{bmatrix}$$

Merge closest clusters (there are 3 equally close, so merge them all), and update the proximity matrix:

$$\begin{bmatrix} & c1 + c4 + c7: & c2: & c3: & c5: & c6: & c8 + c9: \\ c1 + c4 + c7: & (6.67, 8.33, 15) & 0 & & & & \\ c2: & (10, 15, 30) & 25 & 0 & & & \\ c3: & (10, 10, 25) & 15 & 10 & 0 & & \\ c5: & (5, 20, 15) & 13.33 & 25 & 25 & 0 & \\ c6: & (10, 5, 30) & 21.66 & 10 & 10 & 35 & 0 \\ c8 + c9: & (30, 10, 7.5) & 32.5 & 47.5 & 37.5 & 42.5 & 47.5 & 0 \end{bmatrix}$$



Merge closest clusters (there are 3 equally close, so merge them all), and update the proximity matrix:

$$\begin{bmatrix} & c1 + c4 + c7: & c2 + c3 + c6: & c5: & c8 + c9: \\ c1 + c4 + c7: & (6.67, 8.33, 15) & 0 & & \\ c2 + c3 + c6: & (10, 10, 28.33) & 18.33 & 0 & \\ c5: & (5, 20, 15) & 13.33 & 28.33 & 0 \\ c8 + c9: & (30, 10, 7.5) & 32.5 & 40.83 & 42.5 & 0 \end{bmatrix}$$

Merge closest clusters, and update the proximity matrix:

$$\begin{bmatrix} & c1 + c4 + c5 + c7: & c2 + c3 + c6: & c8 + c9: \\ c1 + c4 + c5 + c7: & (6.25, 11.25, 15) & 0 & \\ c2 + c3 + c6: & (10, 10, 28.33) & 18.33 & 0 \\ c8 + c9: & (30, 10, 7.5) & 32.5 & 40.83 & 0 \end{bmatrix}$$

We have three regions, so stop.

### 15. Write pseudo-code for the Normalised Cuts algorithm.

1. Form a graph such that each vertex represents an image element and each edge represents the similarity ( $e_{ij}$ ) between the connected elements
2. For two sets of vertices  $A$  and  $B$  calculate:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

Where:

$$cut(A, B) = \sum_{p \in A, q \in B} e_{pq}$$

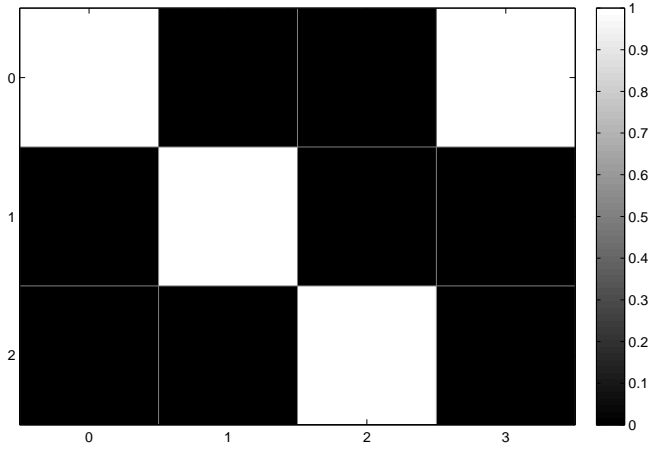
$$assoc(A, V) = \sum_{p \in A, q \in V} e_{pq}$$

3. Repeat from step 2 for every possible combination of nodes in  $A$  and  $B$
4. Cut the edges between the sets  $A$  and  $B$  for which  $Ncut(A, B)$  is minimum
5. Repeat from step 2 for all subgraphs until minimum value of  $Ncut(A, B)$  exceeds a threshold for each subgraph or maximum number of segments have been reached.

### 16. Write pseudo-code for the Hough transform for straight lines

1. Initialise all elements in the accumulator array with a value of zero.
2. For each "edge" pixel
  3. For each value of  $\theta$ 
    4. Calculate the corresponding value of  $r$  (where  $r = y \cos(\theta) - x \sin(\theta)$ )
    5. Increment by one the value stored in the accumulator array nearest to the coordinates  $(\theta, r)$
6. Repeat from Step 3 for all  $\theta$ .
7. Repeat from Step 2 for all pixels.

### 17. Following some initial processing, a 4 by 3 pixel region of an image has been converted into the following binary image:



Perform the Hough transform for straight lines on this image region, using the following values for  $\theta$ : [0,30,60,90,120,150], and quantizing  $r$  to the nearest whole number.

The absolute value of  $r$  will not be greater than the length of the diagonal (i.e  $\sqrt{3^2 + 2^2} = 3.6$ , so rounding up, maximum absolute value of  $r$  is 4. Hence, initialise the accumulator array to:

4	4	0	0	0	0	0	0
	3	0	0	0	0	0	0
	2	0	0	0	0	0	0
	1	0	0	0	0	0	0
	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0
	-2	0	0	0	0	0	0
	-3	0	0	0	0	0	0
	-4	0	0	0	0	0	0
		0	30	60	90	120	150
							$\theta$

For pixel at (0,0), the following lines are possible:

$$\theta = 0, \quad r = 0\cos(0) - 0\sin(0) = 0$$

$$\theta = 30, \quad r = 0\cos(30) - 0\sin(30) = 0$$

$$\theta = 60, \quad r = 0\cos(60) - 0\sin(60) = 0$$

$$\theta = 90, \quad r = 0\cos(90) - 0\sin(90) = 0$$

$$\theta = 120, \quad r = 0\cos(120) - 0\sin(120) = 0$$

$$\theta = 150, \quad r = 0\cos(150) - 0\sin(150) = 0$$

Update accumulator array:

$r$	4	0	0	0	0	0	0
	3	0	0	0	0	0	0
	2	0	0	0	0	0	0
	1	0	0	0	0	0	0
	0	1	1	1	1	1	1
	-1	0	0	0	0	0	0
	-2	0	0	0	0	0	0
	-3	0	0	0	0	0	0
	-4	0	0	0	0	0	0
		0	30	60	90	120	150
							$\theta$

For pixel at (1,1), the following lines are possible:

$$\theta = 0, \quad r = 1\cos(0) - 1\sin(0) = 1$$

$$\theta = 30, \quad r = 1\cos(30) - 1\sin(30) = 0.366 \approx 0$$

$$\theta = 60, \quad r = 1\cos(60) - 1\sin(60) = -0.366 \approx 0$$

$$\theta = 90, \quad r = 1\cos(90) - 1\sin(90) = -1$$

$$\theta = 120, \quad r = 1\cos(120) - 1\sin(120) = -1.366 \approx -1$$

$$\theta = 150, \quad r = 1\cos(150) - 1\sin(150) = -1.366 \approx -1$$

Update accumulator array:

$r$	4	0	0	0	0	0	0
	3	0	0	0	0	0	0
	2	0	0	0	0	0	0
	1	1	0	0	0	0	0
	0	1	2	2	1	1	1
	-1	0	0	0	1	1	1
	-2	0	0	0	0	0	0
	-3	0	0	0	0	0	0
	-4	0	0	0	0	0	0
		0	30	60	90	120	150
							$\theta$

For pixel at (2,2), the following lines are possible:

$$\theta = 0, r = 2\cos(0) - 2\sin(0) = 2$$

$$\theta = 30, r = 2\cos(30) - 2\sin(30) = 0.732 \approx 1$$

$$\theta = 60, r = 2\cos(60) - 2\sin(60) = -0.732 \approx -1$$

$$\theta = 90, r = 2\cos(90) - 2\sin(90) = -2$$

$$\theta = 120, r = 2\cos(120) - 2\sin(120) = -2.732 \approx -3$$

$$\theta = 150, r = 2\cos(150) - 2\sin(150) = -2.732 \approx -3$$

Update accumulator array:

$r$	4	0	0	0	0	0	0
	3	0	0	0	0	0	0
	2	1	0	0	0	0	0
	1	1	1	0	0	0	0
	0	1	2	2	1	1	1
	-1	0	0	1	1	1	1
	-2	0	0	0	1	0	0
	-3	0	0	0	0	1	1
	-4	0	0	0	0	0	0
		0	30	60	90	120	150
							$\theta$

For pixel at (3,0), the following lines are possible:

$$\theta = 0, r = 0\cos(0) - 3\sin(0) = 0$$

$$\theta = 30, r = 0\cos(30) - 3\sin(30) = -1.5 \approx -2$$

$$\theta = 60, r = 0\cos(60) - 3\sin(60) = -2.598 \approx -3$$

$$\theta = 90, r = 0\cos(90) - 3\sin(90) = -3$$

$$\theta = 120, r = 0\cos(120) - 3\sin(120) = -2.598 \approx -3$$

$$\theta = 150, r = 0\cos(150) - 3\sin(150) = -1.5 \approx -2$$

Update accumulator array:

$r$	4	0	0	0	0	0	0
	3	0	0	0	0	0	0
	2	1	0	0	0	0	0
	1	1	1	0	0	0	0
	0	2	2	2	1	1	1
	-1	0	0	1	1	1	1
	-2	0	1	0	1	0	1
	-3	0	0	1	1	2	1
	-4	0	0	0	0	0	0
		0	30	60	90	120	150
							$\theta$

18. Repeat the previous question using the following values for  $\theta$ : [0,45,90,135]

Initialise the accumulator array to:

$r$	4	0	0	0	0
	3	0	0	0	0
	2	0	0	0	0
	1	0	0	0	0
	0	0	0	0	0
	-1	0	0	0	0
	-2	0	0	0	0
	-3	0	0	0	0
	-4	0	0	0	0
		0	45	90	135
					$\theta$

For pixel at (0,0), the following lines are possible:

$$\theta = 0, r = 0\cos(0) - 0\sin(0) = 0$$

$$\theta = 45, r = 0\cos(45) - 0\sin(45) = 0$$

$$\theta = 90, r = 0\cos(90) - 0\sin(90) = 0$$

$$\theta = 135, r = 0\cos(135) - 0\sin(135) = 0$$

Update accumulator array:

$r$	4	0	0	0	0
	3	0	0	0	0
	2	0	0	0	0
	1	0	0	0	0
	0	1	1	1	1
	-1	0	0	0	0
	-2	0	0	0	0
	-3	0	0	0	0
	-4	0	0	0	0
		0	45	90	135
					$\theta$

For pixel at (1,1), the following lines are possible:

$$\theta = 0, r = 1\cos(0) - 1\sin(0) = 1$$

$$\theta = 45, r = 1\cos(45) - 1\sin(45) = 0$$

$$\theta = 90, r = 1\cos(90) - 1\sin(90) = -1$$

$$\theta = 135, r = 1\cos(135) - 1\sin(135) = -1.414 \approx -1$$

Update accumulator array:

$r$	4	0	0	0	0
	3	0	0	0	0
	2	0	0	0	0
	1	1	0	0	0
	0	1	2	1	1
	-1	0	0	1	1
	-2	0	0	0	0
	-3	0	0	0	0
	-4	0	0	0	0
		0	45	90	135
					$\theta$

For pixel at (2,2), the following lines are possible:

$$\theta = 0, r = 2\cos(0) - 2\sin(0) = 2$$

$$\theta = 45, r = 2\cos(45) - 2\sin(45) = 0$$

$$\theta = 90, r = 2\cos(90) - 2\sin(90) = -2$$

$$\theta = 135, r = 2\cos(135) - 2\sin(135) = -2.828 \approx -3$$

Update accumulator array:

$r$	4	0	0	0	0
	3	0	0	0	0
	2	1	0	0	0
	1	1	0	0	0
	0	1	3	1	1
	-1	0	0	1	1
	-2	0	0	1	0
	-3	0	0	0	1
	-4	0	0	0	0
		0	45	90	135
					$\theta$

For pixel at (3,0), the following lines are possible:

$$\theta = 0, r = 0\cos(0) - 3\sin(0) = 0$$

$$\theta = 45, r = 0\cos(45) - 3\sin(45) = -2.121 \approx -2$$

$$\theta = 90, r = 0\cos(90) - 3\sin(90) = -3$$

$$\theta = 135, r = 0\cos(135) - 3\sin(135) = -2.121 \approx -2$$

Update accumulator array:

$r$	4	0	0	0	0
	3	0	0	0	0
	2	1	0	0	0
	1	1	0	0	0
	0	2	3	1	1
	-1	0	0	1	1
	-2	0	1	1	1
	-3	0	0	1	1
	-4	0	0	0	0
		0	45	90	135
					$\theta$

Note, choice of quantisation makes big difference to result.

**19. Briefly describe the two terms that form the energy function that an active contour attempts to minimise.**

*Energy = Internal energy + External energy*

*Internal energy is a function of the shape of the contour, it is reduced if the curve is short and smooth.*

*External energy is a function of the image features near the contour, it is reduced if the intensity gradient is high.*