Lecture-03

Segmentation

Introduction to image segmentation

- The purpose of image segmentation is to partition an image into *meaningful* regions with respect to a particular application
- The segmentation is based on measurements taken from the image and might be *grey-level*, *colour*, *texture*, *depth* or *motion*

Introduction to image segmentation

- Usually image segmentation is an initial and vital step in a series of processes aimed at overall image understanding
- Applications of image segmentation include
  - Identifying objects in a scene for object-based measurements such as size and shape
  - Identifying objects in a moving scene for *object-based video compression (MPEG4)*
  - Identifying objects which are at different distances from a sensor using depth measurements from a laser range finder enabling path planning for a mobile robots

Introduction to image segmentation

- Example 1
  - Segmentation based on greyscale
  - Very simple ‘model’ of greyscale leads to inaccuracies in object labelling

Histogram-based thresholding

- the simplest and most commonly used method of segmentation.
- Given a single threshold, t, the pixel located at lattice position $[i, j]$, with greyscale value $f(i, j)$, is allocated to category 1 if

$$f(i, j) \leq t$$

Otherwise, the pixel is allocated to category 2

Histogram-based thresholding

- In many cases t is chosen manually by the scientist, by trying a range of values of t and seeing which one works best at identifying the objects of interest.
- Although pixels in a single thresholded category will have similar values (either in the range 0 to t, or in the range (t+1) to 255), they will not usually constitute a single connected component.

Histogram-based thresholding

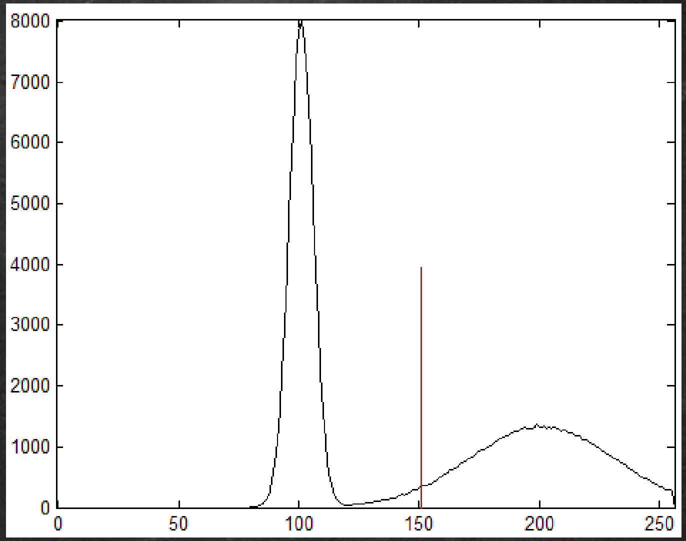
- More than one threshold can be used, in which case more than two categories are produced.
- Thresholds can be chosen automatically. (next slide)

OTSU Automatic thresholding

- Otsu’s method selects the threshold by minimizing the within-class variance of the two groups of pixels separated by the thresholding operator.

OTSU Automatic thresholding

- Formulation
  - Considering, the pixels of a given picture be represented in L gray levels $[1, 2, \dots, L]$. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + \cdots + n_l$.



OTSU Automatic thresholding

- Formulation
  - In order to simplify the discussion, the gray-level histogram is normalized and regarded as a probability distribution:

$$p_i = \frac{n_i}{N} \qquad p_i > 0 \qquad \sum_{i=1}^{L} p_i = 1$$

OTSU Automatic thresholding

- Formulation
  - We divide the pixels into two classes C_0 and C_1 (background and objects, or vice versa) by a threshold at level k;
  - C_0 denotes pixels with levels [1, , k], and C_1 denotes pixels with levels [k + 1,…. , L].

OTSU Automatic thresholding

- Formulation
  - Then the probabilities of class occurrence and the class mean levels, respectively, are given by:

$$\omega_0 = Pr(C_0) = \sum_{i=1}^{k} p_i = \omega(k)$$

$$\omega_1 = Pr(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k)$$

OTSU Automatic thresholding

- Formulation
  - Class means

$$\mu_0 = \sum_{i=1}^{k} ip_i(i|C_0) = \frac{1}{\omega_0}\sum_{i=1}^{k} ip_i$$

$$\mu_1 = \sum_{i=k+1}^{L} ip_i(i|C_1) = \frac{1}{\omega_1}\sum_{i=k+1}^{L} ip_i$$

  - the total mean level of the original picture.

$$\mu_T = \sum_{i=1}^{L} ip_i = \omega_0\mu_0 + \omega_1\mu_1$$

OTSU Automatic thresholding

- The variance formula is

$$\sigma^2 = \sum_{i=1}^{k} (i-\mu)^2 p_i$$

- The class variance is given by,

$$\sigma_0^2 = \sum_{i=1}^{k} (i-\mu_0)^2 \frac{p_i}{\omega_0}$$

$$\sigma_0^2 = \sum_{i=k+1}^{L} (i-\mu_1)^2 \frac{p_i}{\omega_1}$$

OTSU Automatic thresholding

- Within-class variance(Cost function or Objective function)

$$\sigma_w^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2$$

- the between-class variance

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$$

$$\sigma_B^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2$$

OTSU Automatic thresholding

- Then our problem is reduced to an optimization problem to search for a threshold k that maximizes between-class variance object functions