Some Exercises

You are using k-means clustering in color space to segment an image. How-ever, you notice that although pixels of similar color are indeed clustered together intothe same clusters, there are many discontiguous regions because these pixels are often notdirectly next to each other. Describe a method to overcome this problem in the k-meansframework.

Solution: Concatenate the coordinates (x,y) with the color features as input to the k-means algorithm.

Recursive filtering techniques are often used to reduce the computational complexity of a repeated operation such as filtering. If an image filter is applied to each location in an image, a (horizontally) recursive formulation of the filtering operation expresses the result at location $(x+1, y)$ in terms of the previously computed result at location (x, y).

A box convolution filter, B, which has coefficients equal to one inside a rectangular window, and zero elsewhere is given by:

$$B(x, y, w, h) = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} I(x+i, y+j)$$

where $I(x, y)$ is the pixel intensity of image I at (x, y). We can speed up the computation of arbitrary sized box filters using recursion as described above. In this problem, you will derive the procedure to do this.

(a) The function J at location (x,y) is defined to be the sum of the pixel values above and to the left of (x,y), inclusive:

$$J(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$

Formulate a recursion to compute $J(x, y)$. Assume that $I(x, y) = 0$ if $x < 0$ or $y < 0$. *Hint: It may be useful to consider an intermediate image to simplify the recursion.*

(b) Given $J(x, y)$ computed from an input image, the value of an arbitrary sized box filter (B_J) applied anywhere on the original image can be computed using four references to $J(x, y)$.

$$B_J(x, y, w, h) = aJ(?, ?) + bJ(?, ?) + cJ(?, ?) + dJ(?, ?)$$

Solution

(a) We can compute each row in one pass, and each column in a second pass. Given an intermediate image $P(x, y)$, we can compute:

$$P(x, y) = P(x-1, y) + I(x, y)$$
$$J(x, y) = J(x, y-1) + P(x, y)$$

(b)

$$B_J(x, y, w, h) = J(x+w, y+h) - J(x-1, y+h) - J(x+w, y-1) + J(x-1, y-1)$$

How does SIFT achieve rotational invariance?

One can make descriptors rotationally invariant by assigning orientations to the key points and thenrotating the patch to a canonical orientation. In SIFT this is done by constructing Histograms of Gradients ina neighborhood around the feature point, and assigning the largest bin as the corresponding direction of thekeypoint. Later, all detected features are rotated so that the corresponding orientations are vertically aligned

9. [3 points] The pixel value recorded from a particular point on a camera's sensor array depends on several factors, such as the position and orientation of the object in the world that gets imaged at that point. Name at least three other factors that determine the pixel value of the point.

Exposure time, illumination, aperture, focus, …

12. [5 points] What 3D point $p = (x, y, z)$ is produced by adding the following 3D points, represented in homogeneous coordinates:

$$p_1 = \begin{bmatrix} 2 \\ 3 \\ 6 \\ 2 \end{bmatrix} \qquad p_2 = \begin{bmatrix} 4 \\ 3 \\ 0 \\ 4 \end{bmatrix}$$

$(2, 2.25, 3)^T$ – first convert each to (x, y, z, 1)

In difference images, pixel values greater than zero (or greater than a certain threshold) are thought to reflect moving objects in the scene. However, this does not always have to be the case. There are other factors besides object motion that can cause non-zero values in a difference image. List as many such factors as you can think of.

- Change in lighting
- Noise in the images varying over time
- Motion of the camera