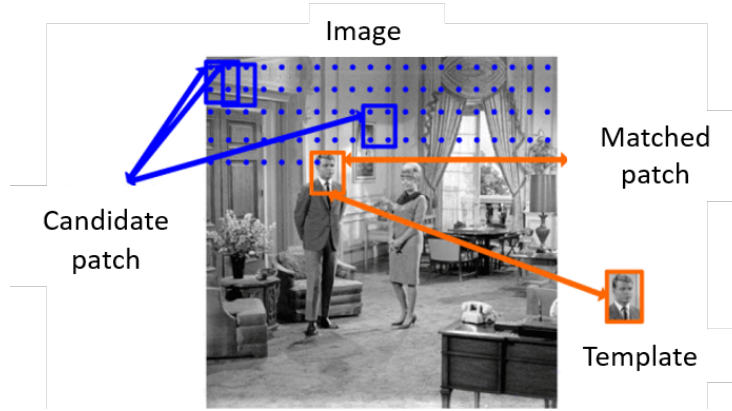


# CoE 3SK3: Project Part 1

Due: February 1, 2017

Template matching, a fundamental problem in computer vision and image processing, is the task of locating the best matched patch in reference image to a given query template. For example, given a face image (the query template), find the face in an input image, if it exists.



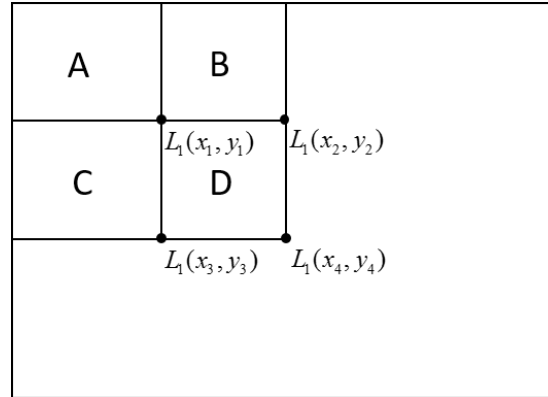
The most popular strategy for solving the template matching problem is to compare some statistical features of the template with each patch of the input image and output the patch with the most similar features to the query template. The mean pixel value of a patch, for instance, is a commonly used patch feature. The computation of the mean values for all patches in an input image can be greatly sped up by using a so-called summed area table. Each element of the summed area table records the sum of the pixel values in the rectangular area enclosed by the top-left corner and the current pixel position.

$$L(x, y) = \sum_{i=1}^x \sum_{j=1}^y I(i, j). \quad (1)$$

With the summed area table of an image, the sum  $S(D)$  of an arbitrary area  $D$  can be calculated as follows,

$$\begin{aligned} S(D) &= S(A + B + C + D) - S(A + B) - S(A + C) + S(A) \\ &= L(x_4, y_4) - L(x_2, y_2) - L(x_3, y_3) + L(x_1, y_1) \end{aligned} \quad (2)$$

where, as shown in the following figure, the sums of areas  $A$ ,  $A + B$ ,  $A + C$ ,  $A + B + C + D$ , i.e.,  $L(x_1, y_1)$ ,  $L(x_2, y_2)$ ,  $L(x_3, y_3)$ ,  $L(x_4, y_4)$ , are already known in the summed area table.



Your task is to implement this idea for template matching using Matlab. Given a query template  $X$ , find the patch of the same size of  $X$  in an input image that has the closest mean value to the mean of  $X$ . Besides the source code with detailed comments, you need to submit a report to analyze the complexity of your implemented algorithm.

Suppose that the value of each pixel is an integer from 0 to 255. Is it a good idea to store the summed area table using IEEE 32-bit float format? What would happen if you do so? What about using unsigned 32-bit integer format? Justify your answer in your report.