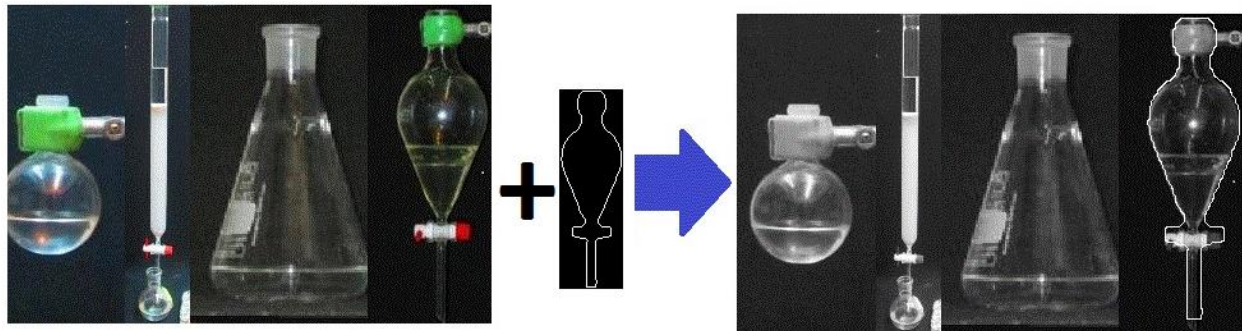


## Simple template match with variable image to template size ratio



Find Template  $Itm$  in image  $Is$ . The template does not have to fit the size of the object in the image. The program scans various of size ratios of template to image to find the location and size ratios of the best match. The template match is done crosscorrelation of the resized template  $Itm$  to the edge image of the target image  $Is$ .

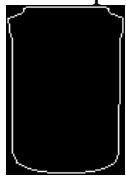
**MAIN\_find\_object\_in\_image** ( $Itm, Is$ ) is the main function. The output is the boundary and location and size of the template in the image with the object boundary marked on it.

### Input:

**$Is$ :** Color image with the object to be found.



**$Itm$ :** A template of the object to be found. The template is a binary image with the boundary of the template marked 1 (white) and all the rest of the pixels marked 0.

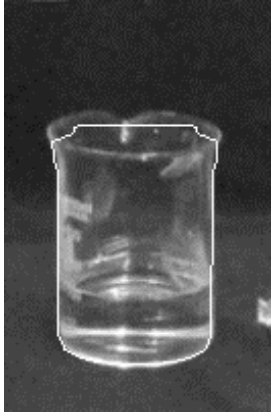


Template of object could be created by extracting the object boundary in image with uniform background, this could be done (for symmetric objects) using the code at:

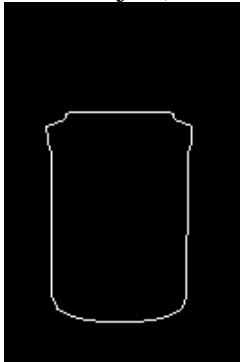
<http://www.mathworks.com/matlabcentral/fileexchange/46887-find-boundary-of-symmetric-object-in-image>

## Output

***Ismarked:*** The image with the template marked upon it in the location of and size of the best match.



***Iborders:*** Binary image of the borders of the template in for the best match (borders of the found object)



***Xbest,Ybest:*** Location on the image (in pixels) where the template best match the image for the upper left corner of the template

.

***BestScore:*** Score of the best match found in the scan (the score of the output).

## How to use

Run the function: **MAIN\_find\_object\_in\_image**(*Is,Itm*) With the above parameters.

The output is the parameters

[*Ismarked,Iborders,Ybest,Xbest, ItmSize, BestScore*]

Described above.

## Optional Input:

***Itm\_dilation:*** The amount of dilation for of the template. How much the template line will be thickened (in pixels) for each side before crosscorelated with the image. The thicker the template the better its chance to overlap with the edge of the object in the image and more rigid the recognition process. However thick template can also reduce recognition accuracy. The default value for this parameter is 1/40 of the average dimension size of the template *Itm*.



## Algorithm:

The function scans the various of ratios of image *Is* to the template *Itm* and for each ratio search for the template in the image. The size ratio and location in the image that gave the best match for the template are chosen.

This is done by resizing the template *Itm* to match the size of the image *Is* and then shrinking it in jump 0.5% until it reaches 3% of the image size. For each size the template *Itm* is matched to the image the match with the best score for all size ratios is chosen

## Template\_match(*Is*,*Itm* ,*Itm\_dilation*)

Find template *Itm* in greyscale image *Is* using various of forms of template match specified by the optional input parameters.

Return the *x,y* coordniates of the best match.

Also return the *score* of the best match.

(there is no resizing or rotating of the template or image during scan).

Based on crosscorrelation between the template *Itm* and the image or edge image of *Is*.

### INPUT (essential)

*Is*: Greyscale picture were the template *Itm* should be found.

*Itm*: Binary edge image of the template with edges marked 1 and the rest 0.

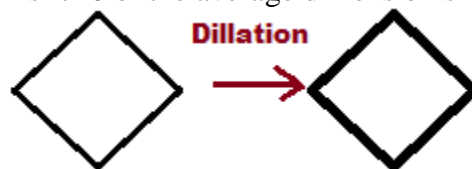
### OUTPUT

*x,y*: Coordinatines of template *Itm* in image *Is* for the best match (Location the edge point [1,1]) of the template *Itm* in *Is* for the match with the highest score.

*score*: Score of the best match.

## Optional input

*Itm\_dilation*: The amount of dilation for of the template. How much the template line will be thickened (in pixels) for each side before crosscorelated with the image. The thicker the template the better its chance to overlap with the edge of the object in the image and more rigid the recognition process. However thick template can also reduce recognition accuracy. The default value for this parameter is 1/40 of the average dimension size of the template *Itm*.



## **[*mat*]=Resize\_binary\_edge\_image(*I*,*Sy*,*Sx*)**

Resize binary/logic image that consists of lines and curves of thickness of one pixels (i.e edge images, graphs, contours..) to new size binary image while maintaining line connectivity and line thickness of one pixel.

### **Input**

***I*:** Binary edge image (logical type) consist of lines and curves with a thickness of one pixel (such as curves, contour line, template, or edge images)

***Sy,Sx*:** The size of the resized image

OR

***Sy*** (no ***Sx***) the scale ratio of the resized image to the original image

### **Output**

***mat*:** Resized version of the input image ***I***, also binary edge image, the connectivity/topology of the edges/curves in ***I*** is maintained and also the line thickness remains one pixel

### **Note:**

In addition to enlarging/shrinking of images can also be used to stretch edge images by using different proportions (***Sx,Sy***) of output image dimension to input image dimension.