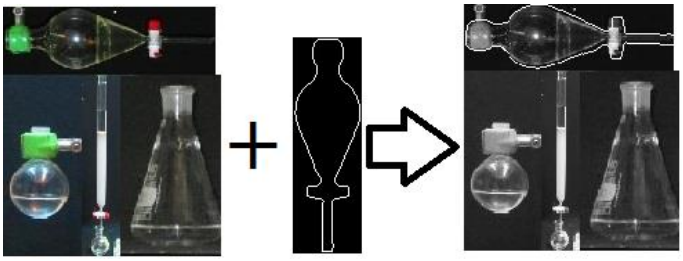
Find object in image using template match with rotation



Find an object that fit Template *Itm* in image *Is*.The orientation of the template and the object in the image does not have to be the same as that as the template. The template *Itm* is matched to the image *Is* in various of rotations and the best match is chosen. The function can use various of methods to trace the template in the image, including: Generalized Hough transforms, Normalize crosscorrelation to edge image and other forms of template match.

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.

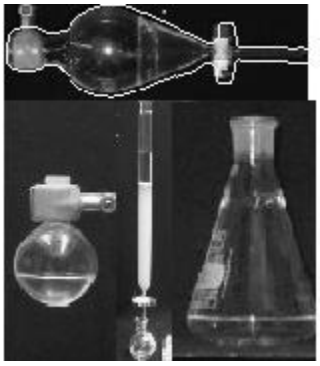
Itm: A template of the object to be found. The template is a binary image with the boundary of the template marked 1(white) and all the rest of the pixels marked 0.

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

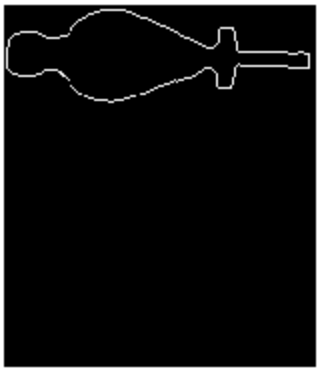
Additional optional input parameters are given in the next page.

Output

Ismarked: The image with the template marked upon it in the location of and rotation angle 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

ItmRot: The angle of rotation of the template (*Itm*) that gave best results.

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, ItmRot, BestScore]*

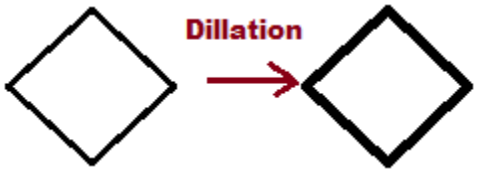
Described above.

Optional Input:

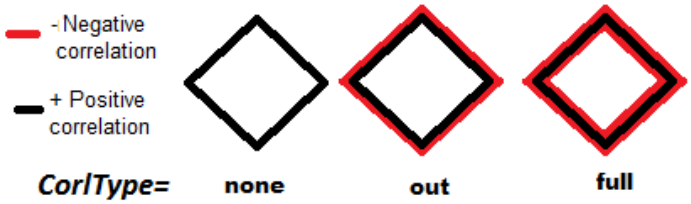
Search_Mode: The method by which template *Itm* will be searched in image *Is:*
Search_Mode='*hough*': use generalized hough transform to scan for template *Itm* in image *Is*.
Search_Mode='*template*': Use crosscorellation to scan for template *Itm* in the edge image of Is (default).

Edge_Type: Only relevant case of *Search_Mode*='template'. This parameter determine the type of image to which the template will be matched (edge, gradient, grayscale).
Edge_Type='sobel': Template Itm will be matched (cross-correlated) to the 'sobel' gradient map of the image *Is*.
Edge_Type ='canny': Template Itm will be matched (cross-correlated) to the to 'canny' binary edge map of the image *Is* (default).
Else Template *Itm* will be matched (crosscorrelated) to the grayscale version of the image *Is*.

Itm_dilation: Only relevant case of *Search_Mode*='template'. 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*.



CorlType: Only relevant case of *Search_Mode*='template'. Matching the template to the edge image is likely to give high score in any place in the image with high edge density which can give high false positive, to avoid this few possible template match option are available:



CorlType='out': Use negative template (negative correlation) surrounding the template contour (in small radius around the template line) but only on the outside of the template.
Crosscorrelation of this area with edges in the canny image will reduce the match score of the template in this location (default).
CorlType ='full': Use negative template (negative correlation) around the template contour line (in small radius around the template line) for both the inside and outside of the template.
Crosscorrelation of these areas with edges in the canny image will reduce the match score of the template in this location (default).
CorlType ='none': Use the template as it is.

Other functions:

Generalized_hough_transform*(Is, Itm)*

Find template *Itm* in greyscale image *Is* using generalize hough transform
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).

INPUT
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: Coordninates of template *Itm* in image *Is* for the best match (Location the edge (point [1,1]) of the template *Itm* in I*s* for the match with the highest score).

score: Score of the best match.

Template_match*(Is,Itm,CorlType,Edge_Type,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).
For the various of working mode of the function see the optional INPUT (optional) section. All method are in general 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

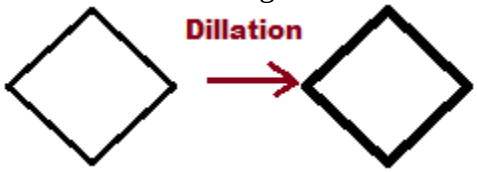
x,y: Coordninates of template *Itm* in image *Is* for the best match (Location the edge point [1,1]) of the template *Itm* in Is for the match with the highest score.

score: Score of the best match.

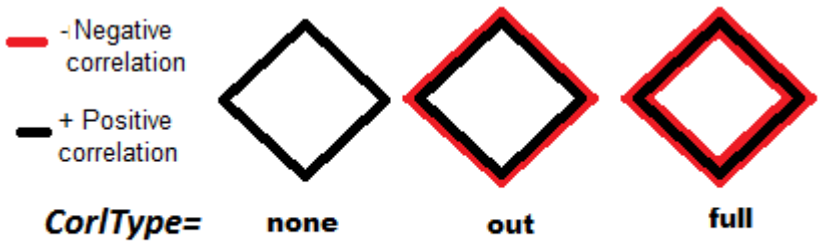
Optional input

Edge_Type: Only relevant case of *Search_Mode*='template'. This parameter determine the type of image to which the template will be matched (edge, gradient, grayscale).
Edge_Type='sobel': Template Itm will be matched (cross-correlated) to the 'sobel' gradient map of the image *Is*.
Edge_Type ='canny': Template Itm will be matched (cross-correlated) to the to 'canny' binary edge map of the image *Is* (default).
Else Template *Itm* will be matched (crosscorrelated) to the grayscale version of the image *Is*.

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*.

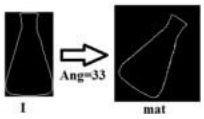


CorlType: Only relevant in case of *Search_Mode*='template'. Matching the template to the edge image is likely to give high score in any place in the image with high edge density which can give high false positive, to avoid this few possible template match option are available:



CorlType='out': Use negative template (negative correlation) surrounding the template contour (in small radius around the template line) but only on the outside of the template. Crosscorrelation of this area with edges in the canny image will reduce the match score of the template in this location (default).
CorlType ='full': Use negative template (negative correlation) around the template contour line (in small radius around the template line) for both the inside and outside of the template. Crosscorrelation of these areas with edges in the canny image will reduce the match score of the template in this location (default).
CorlType ='none': Use the template as it is.

Rotate_binary_edge_image(*I,Ang*)



Rotate edge image (*I*) in (*Ang*) degrees
The rotated output image will also also be a binary edge image.
The connectivity/topology of all edges/curves in the input image (*I*) will be maintained and the line thickness of the curves in the output image (*mat*) will remain 1 pixel.
The center of rotation is the center of the image
The dimensions of the output image (*mat*) will be different from the input image and will be set such that the rotated image is fully within the image frame.

Input

***I*:** Binary edge image (logical type) consist of lines and curves with a thickness of one pixels (such as curves, contour line, template, or edge images)
***Ang*:** Rotation angle of the image in Degrees

Output

mat: Rotated version of the input image (I), also binary edge image, the connectivity/topology of the edges/curves in input image (I) is maintained and also the line thickness remain one pixel.