COMS 4731 HW2

Xiaoning Wan

xw2501

WalkThrough1:

threshold is set to 100/255 based on the histogram of 'img'

dilations/erosions time k is set to 5 based on experiment, when k=5, the output will satisfy the requirement, if k is set too small, the holes won't be cleaned, if too big, the image will change

erosions/dilations time k is set to 10 based on experiment, when k=10, the output will satisfy the requirement, if k is set too small, the rice won't be removed, if too big, the image will change

Program Challenge:

as multiple images will be shown when run this program, please be sure to check the title of each image

generateLabeledImage:

function used:

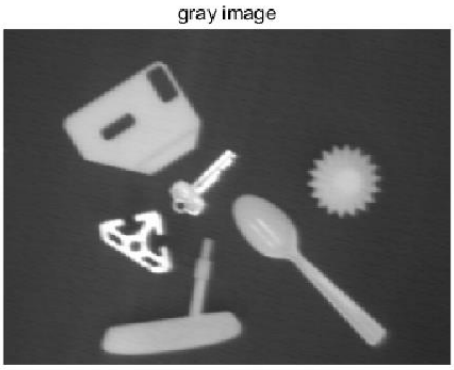
im2bw

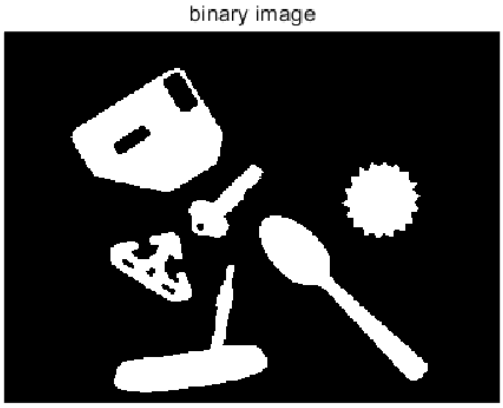
bwlabel

threshold is set to 125/255 based on histogram and experiment

in this function, the gray image and binary image will also be shown in figures, one is titled 'gray image', the other is titled 'binary image'

running result example:





compute2DProperties:

function used:

no image processing function used

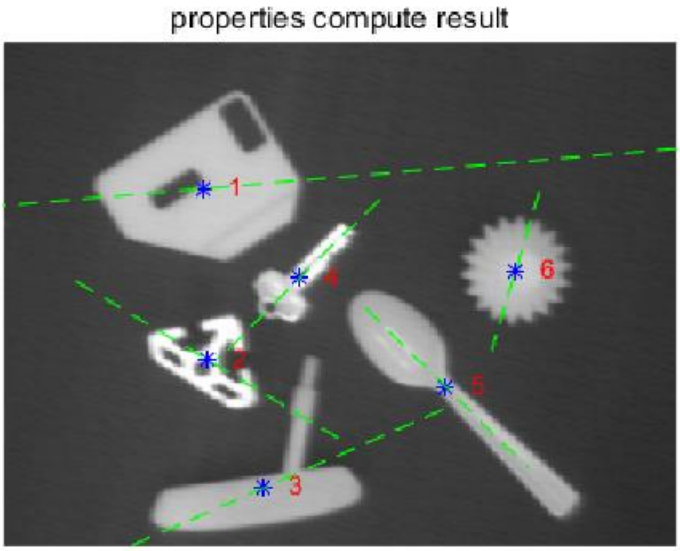
7 properties are computed in this function, besides the required 6 properties, I also computed the value of (minimum moment of inertia)/(area), this property is used for the match function. The reason for computing this property is roundness itself is not enough to show all the features of an object, and properties such as center position, area, orientation will change as the object scales, moves or rotates. Thus, I came up with this feature, along with roundness to identify an object.

the orientation is computed according to the coordinate given in the lecture, thus there is a 90-degree difference from that computed by MATLAB image processing function, besides, the orientation is given in radian measure

an image titled 'properties compute result' will be shown in figure, the center of each objects is marked in blue star(*), the orientation is drawn in green dot line(--), the red number next to each objects indicates the ID of this object

as I choose to have all the lines have the same x-axis projection length, several lines are longer while some are shorter

running result example:



recognizeObjects:

function used:

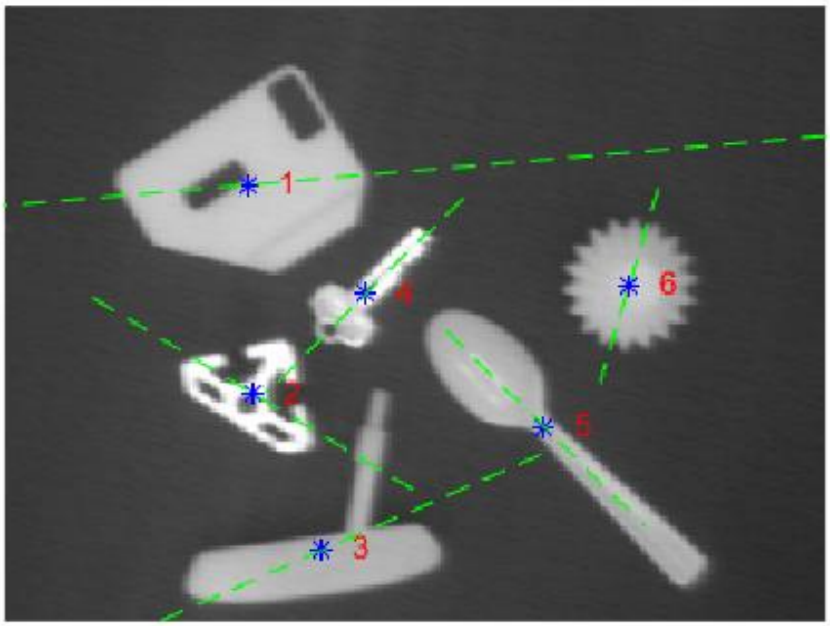
no image processing function used

the recognize is processed based on the roundness and property 7(minimum moment of inertia/area), I wrote two ways for matching objects.

If you would like to test both of them, change the value of **'choose1'** in recongizaObjects.m, by default it is set to false which means method 2 is used.

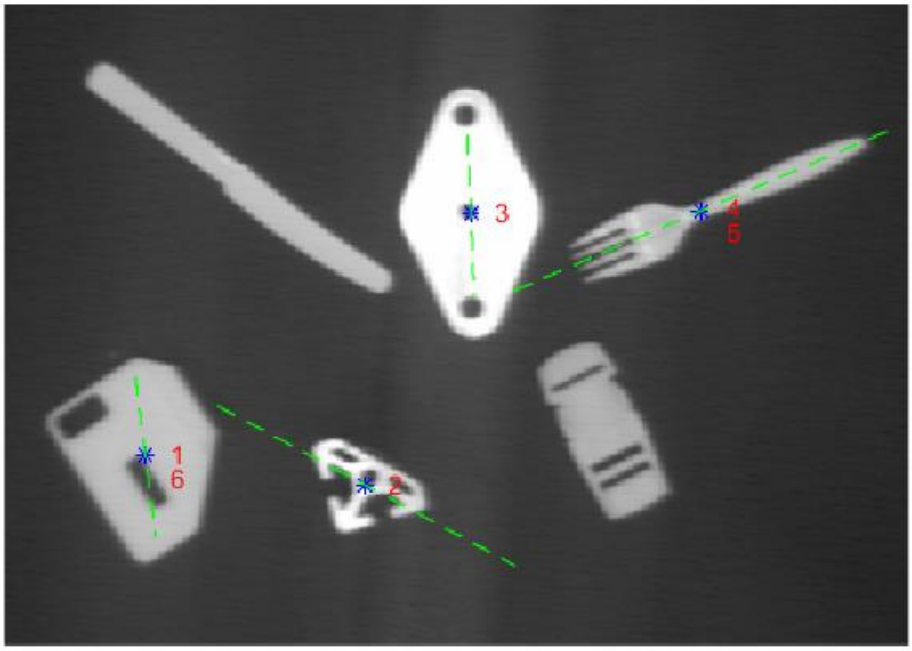
i. The first method is to compare these two properties between target object and all the potential objects, then choose the most similar one as matched object.
In this method, we need to compute both the roundness and property 7 difference between current object and target object, when current difference in both these two properties are smaller than previous minimum, we set this object as temporary matched object and set current difference as new previous minimum.
Running result example:

properties compute result



This is the image of many_objects_1, 6 objects are labeled and have their properties computed, then we will match these 6 objects in many_objects_2

match result

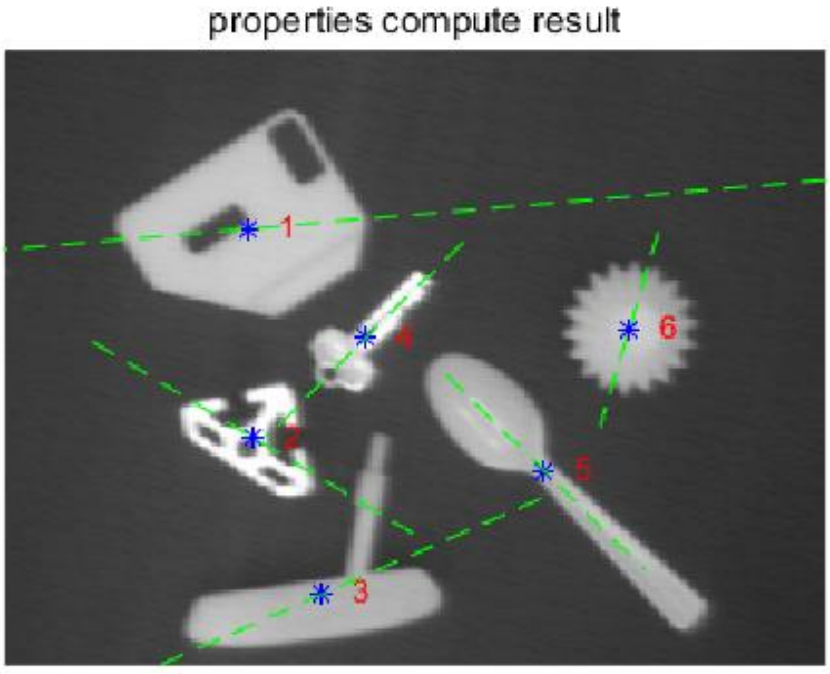


This is the match result of many_objects_1 and many_objects_2. This image shows the match result by object number. The red number next to the objects no longer indicates the ID of object in this image, but it means the ID of matched object in the former image.

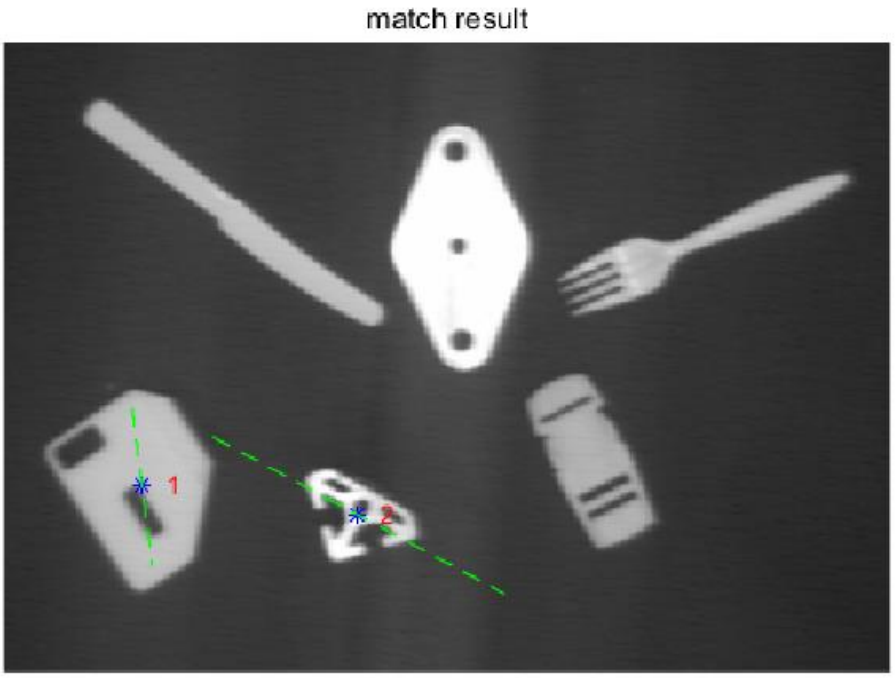
In this case, the ‘fork’ is labeled 4 and 5, it means that it matches the ID 4 and 5 object in the former image which is the ‘key’ and ‘spoon’ respectively.

According to the result, we can see that this method could match all the objects whether there is actually a matched object or not. It chooses the most ‘like’ one as a matched object.

ii. The second method is to compute the percentage differential of roundness and property 7 between objects, the one with a differential smaller than the threshold is a match.
In this program, I choose 10% as the threshold, I came up with this number by experiment.
Running result example:



This is the image of many_objects_1, 6 objects are labeled and have their properties computed, then we will match these 6 objects in many_objects_2



This is the match result of many_objects_1 and many_objects_2. This image shows the match result by object number. The red number next to the object also means the ID of matched object in the former image.

According to this result, this method is more accurate in matching, as it won't match different objects. However, the result is subject to the value of threshold, if threshold is set too small, there will not be match at all, and if is too big, the output will suffer inaccuracy caused by it. Thus the choose of threshold is the key to this method.

From here to the end is the test result of

Matching between **two_objets** and **many_objects_1**,

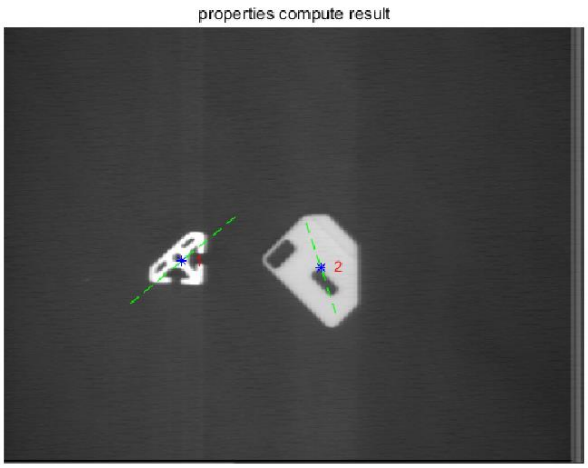
two_objets and **many_objects_2**,

many_objects_1 and **many_objects_2**

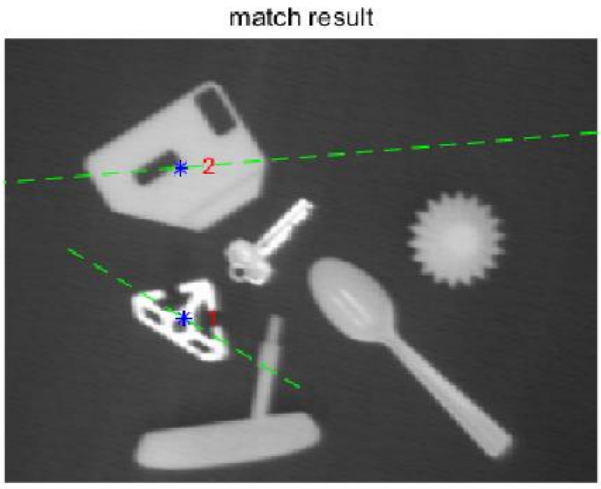
two_objets and **many_objects_1**

method 1 and method 2 has the same result.

ID of objects in two_objects:



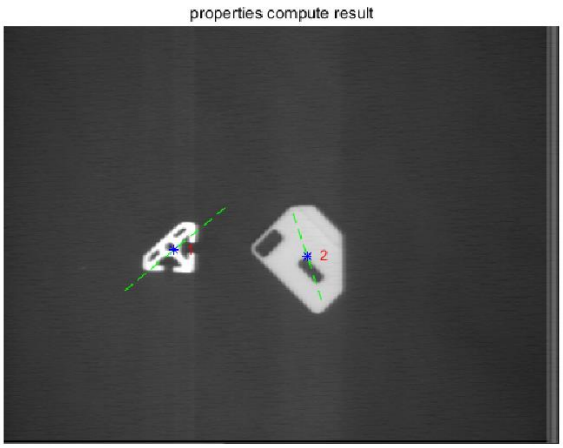
Matching result:



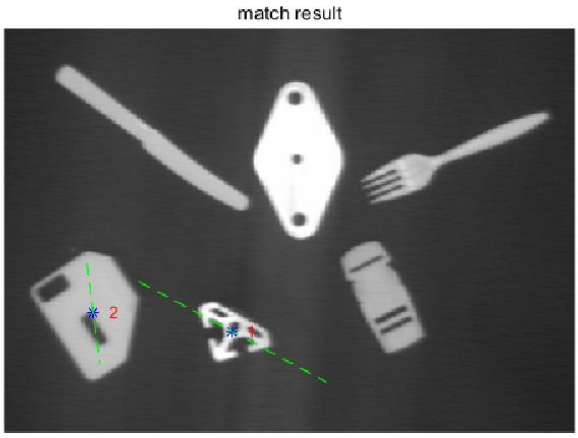
two_objets and **many_objects_2**

method 1 and method 2 has the same result.

ID of objects in two_objects:



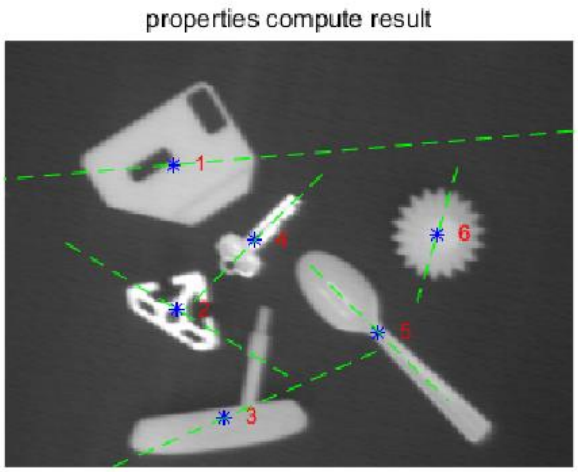
Matching result:



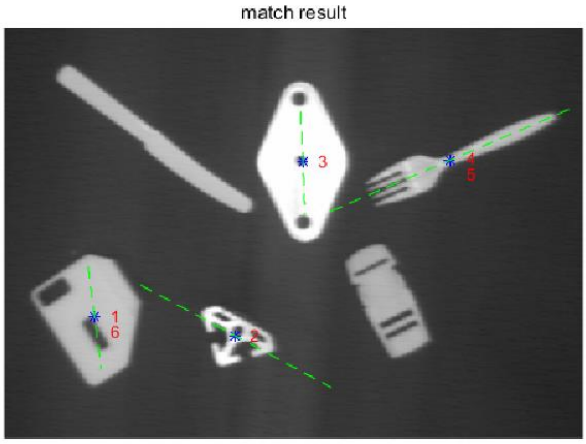
many_objects_1 and **many_objects_2**

method 1

ID of objects in many_objects_1:



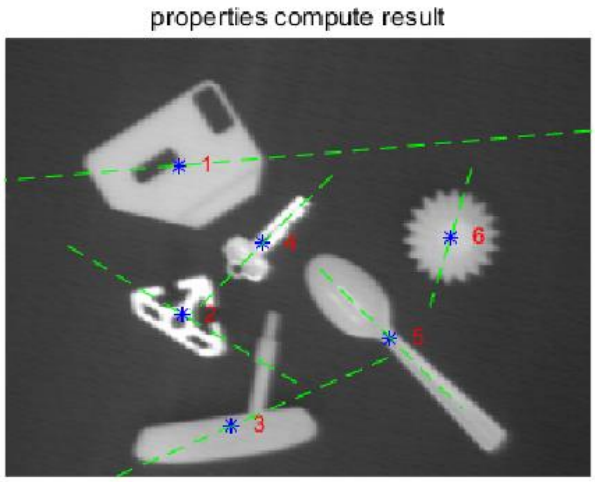
Matching result:



many_objects_1 and **many_objects_2**

method 2

ID of objects in many_objects_1:



Matching result: