Please write a visual C++ program for the following three problems. Please send back the source code, the executable file, and a document to discuss the basic idea and to illustrate how to use the executable file. In attached, there are 12 pairs of images for testing purpose.

Problem1:

Requirement: There are a bunch of non-overlapping white circles in Figure(a). Please detect the position and measure the radius of each circle.

Input: an image

Output: a file containing a list of circles with their positions and sizes

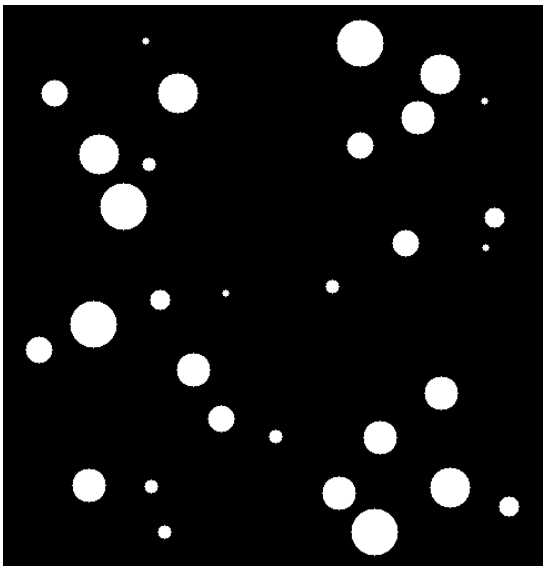

Figure (a)

Problem2:

Each circle in Figure(b) corresponds to a circle in Figure(a). For a pair of corresponding circles, their sizes are same and their center positions are correlated by a global similarity transformation:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = s \begin{bmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix}$$

where s, a, u, and v are rigid transformation parameters. (x_0 ,y_0) is the center of the circle in Figure a and (x_1 ,y_1) is the center of the circle in figure (b). Please design and implement an efficient algorithm to estimate the global transformation. Please analyze the time complexity of the algorithm.

Input: a pair of image

Output:

a. the global rigid transformation

Auckland Bioengineering Institute
Correspondence: Mahyar Osanlouy

b. an image A' obtained by warping A based on the estimated rigid transformation
c. a file containing a list of circles with their positions in figure(a) and figure(b)

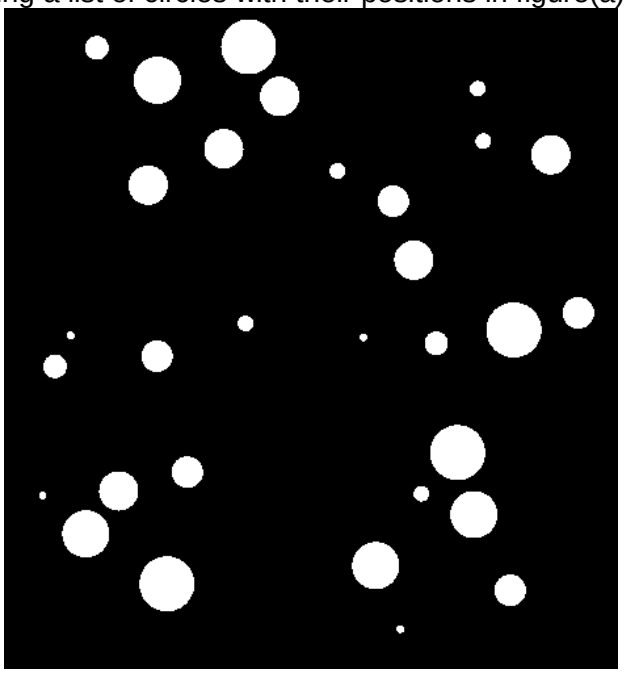

Figure (b)

Problem3:

The requirement is the same as in the problem2: estimate the global rigid transformation between image A and image B. However, a circle in one image **may not** exist in the other image, even though there is an overall correspondence between two images. Please develop a robust algorithm to solve this problem.

Actually, only one algorithm is needed to solve both problem 2 and problem 3. However because there is no missing circles in problem 2, it is possible to develop a faster algorithm.

Testing data:
No missing circles:
pair1, pair2, and pair3

Missing circles in B:
pair 4: 10% of the circles in A are missing in B
pair 5: 20% of the circles in A are missing in B
pair 6: 40% of the circles in A are missing in B
pair 7: 60% of the circles in A are missing in B

Missing circles in both A and B:
pair 8: 5%
pair 9: 10%
pair 10: 20%
pair 11: 30%
pair 12: 40%

Auckland Bioengineering Institute
Correspondence: Mahyar Osanlouy