

MP#2 Report

- **Target:**

The purpose of this MP is to implement 5 morphological operations: erosion, dilation, opening, closing and boundary.

- **Technique:**

I use C++ with the OpenCV library to solve this problem. I use one data structure called Mat to store the data of picture and conduct operations.

- **Algorithm description:**

For erosion function, we just traverse the picture twice from left to right and line-by-line. Firstly, I traverse the whole picture to find all points that are according with the definition and record them into a vector. Then, mark each value as 0 of that picture. After that, traverse the vector and mark the value of the corresponding points in the picture as 1. For the dilate function, it is similar as the erosion function. To begin with, we traverse the picture to find the valid points and store them into a vector. Then set value of surrounding points into 1 according to the structure elements. For opening function, we first do erosion and then do dilate. For the closing function, we first conduct dilate then erode it. For the boundary function, just the original image deducts the eroded image.

- **Result:**

Palm.bmp (the SE is 3*3 matrix, the origin is (0, 0) and the iterations is 1):



dilate



erode



close



open

palm.bmp (the SE is 4*4 matrix, the origin is (-1, 1) and the iterations is 1):



dilate



erode



close



open

gun.bmp (the SE is 4*4 matrix, the origin is (-1, 1) and the iterations is 1):



dilate



erode

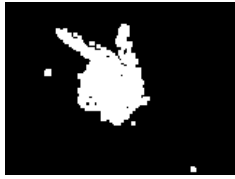


close



open

gun.bmp (the SE is 3*3 matrix, the origin is (0, 0) and the iterations is 1):



dilate



erode



close



open

Boundary:



● Result Analyze:

Dilate function is to expand an image so it looks fat. Erode function is to shrink an image so it looks thin. Close function is to smooth corner but it can fill up small holes so that image has fewer holes than the image operated by open function. Open function is also to smooth corner but it can eliminate small area so the image has no small origin but the close function will still keep these spots. For the boundary result, due to there a great number of holes in these original images so there are many boundaries in the final image.