Computer Vision HW7

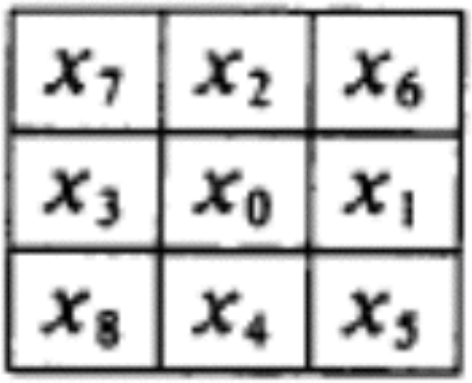
National Taiwan University

R02944002 王瀚宇

Description

I implemented Zhang-Suen Thinning Algorithm, This algorithm is used for thinning binary image. Binary image by definition, consists of only black and white pixels.

8 neighborhood:



- Part A : Let the region pixel = 1, background = 0.
  - If $2 \leq \sum_{n=1}^{8} x_n \leq 6$, TRUE.
  - Go through clockwise from the top pixel, once pixel turn into 1 from 0, counter++ If counter = 1, TRUE.
  - If $P_2 \cdot P_1 \cdot P_4 = 0$, TRUE.
  - If $P_1 \cdot P_4 \cdot P_3 = 0$, TRUE.
  - If all above true, DELETE the pixel.

- Part B : Let the region pixel = 1, background = 0.
  - Same as Part A.
  - Same as Part B.
  - If $P_3 \cdot P_2 \cdot P_1 = 0$, TRUE.
  - If $P_4 \cdot P_3 \cdot P_2 = 0$, TRUE.
  - If all above true, DELETE the pixel.

Note: Extra spurs (short branches) caused by thinning must be minimized.

parameters

```
String fileName = "./assets/lena.im";
int headerLength = 172;
int imageWidth = 512;
int imageHeight = 512;
int threshold = 128;
```

principal code

```
 ArrayList<Integer> bytes = GetByteData(fileName);
            ArrayList<Integer> binerized =
Binarize(bytes,threshold,headerLength,imageWidth,imageHeight);

            ArrayList<Integer> origin = binerized;
            ArrayList<Integer> result;

            result = thinning(origin,headerLength,imageWidth,imageHeight);
            while(!identical(result,origin))
            {
                  origin = copy(result);
                  result =
thinning(origin,headerLength,imageWidth,imageHeight);
            }

            WriteOut(result,”./assets/result.im");
```

Result