Computer vision homework 5

B04902028 資工三 洪浩翔

Original Lena image

1. Gray-scale morphological dilation:

The result after dilation

Main code about dilation:

```
#dilation
map_record = [[-1 for i in range(h+4)] for j in range(w+4)]
for i in range(h):
  for j in range(w):
    map_record[i+2][j+2] = image.getpixel((i , j))

for i in range(h):
  for j in range(w):
    tmp = max(map_record[i+1][j] , map_record[i+3][j] , map_record[i+2][j] , map_record[i][j+1] , map_record[i+1][j+1] , map_record[i+2][j+1] ,
              map_record[i+3][j+1] , map_record[i+4][j+1] , map_record[i][j+2] , map_record[i+1][j+2] , map_record[i+2][j+2] , map_record[i+3][j+2] ,
              map_record[i+4][j+2] , map_record[i][j+3] , map_record[i+1][j+3] , map_record[i+2][j+3] , map_record[i+3][j+3] , map_record[i+4][j+3] ,
              map_record[i+1][j+4] , map_record[i+3][j+4] , map_record[i+2][j+4])
    dilation.putpixel((i , j) , tmp)

dilation.show()
dilation.save("gray_dilation.bmp")
```

2. Gray-scale morphological erosion:

The result after erosion

Main code about erosion:

```
#erosion
for i in range(h):
  for j in range(w):
    tmp = min(map_record[i+1][j] , map_record[i+3][j] , map_record[i+2][j] , map_record[i][j+1] , map_record[i+1][j+1] , map_record[i+2][j+1] ,
              map_record[i+3][j+1] , map_record[i+4][j+1] , map_record[i][j+2] , map_record[i+1][j+2] , map_record[i+2][j+2] , map_record[i+3][j+2] ,
              map_record[i+4][j+2] , map_record[i][j+3] , map_record[i+1][j+3] , map_record[i+2][j+3] , map_record[i+3][j+3] , map_record[i+4][j+3] ,
              map_record[i+1][j+4] , map_record[i+3][j+4] , map_record[i+2][j+4])
    erosion.putpixel((i , j) , tmp)

erosion.show()
erosion.save("gray_erosion.bmp")
```

3. Gray-scale morphological opening:

The result after opening

Main code about opening:

```
#opening
map_record = [[-1 for i in range(h+4)] for j in range(w+4)]
for i in range(h):
  for j in range(w):
    map_record[i+2][j+2] = erosion.getpixel((i , j))

for i in range(h):
  for j in range(w):
    tmp = max(map_record[i+1][j] , map_record[i+3][j] , map_record[i+2][j] , map_record[i][j+1] , map_record[i+1][j+1] , map_record[i+2][j+1] ,
              map_record[i+3][j+1] , map_record[i+4][j+1] , map_record[i][j+2] , map_record[i+1][j+2] , map_record[i+2][j+2] , map_record[i+3][j+2] ,
              map_record[i+4][j+2] , map_record[i][j+3] , map_record[i+1][j+3] , map_record[i+2][j+3] , map_record[i+3][j+3] , map_record[i+4][j+3] ,
              map_record[i+1][j+4] , map_record[i+3][j+4] , map_record[i+2][j+4])
    opening.putpixel((i , j) , tmp)

opening.show()
opening.save("gray_opening.bmp")
```

4. Gray-scale morphological closing:

The result after closing

Main code about closing:

```
#closing
map_record = [[-1 for i in range(h+4)] for j in range(w+4)]
for i in range(h):
  for j in range(w):
    map_record[i+2][j+2] = dilation.getpixel((i , j))

for i in range(h):
  for j in range(w):
    tmp = min(map_record[i+1][j] , map_record[i+3][j] , map_record[i+2][j] , map_record[i][j+1] , map_record[i+1][j+1] , map_record[i+2][j+1] ,
              map_record[i+3][j+1] , map_record[i+4][j+1] , map_record[i][j+2] , map_record[i+1][j+2] , map_record[i+2][j+2] , map_record[i+3][j+2] ,
              map_record[i+4][j+2] , map_record[i][j+3] , map_record[i+1][j+3] , map_record[i+2][j+3] , map_record[i+3][j+3] , map_record[i+4][j+3] ,
              map_record[i+1][j+4] , map_record[i+3][j+4] , map_record[i+2][j+4])
    closing.putpixel((i , j) , tmp)

closing.show()
closing.save("gray_closing.bmp")
```