Computer Vision I

Homework 7 - Thinning B04902090 資工四 施長元

Usage of the full code:

`python3 main.py [Image_path]`

After the code exit, output file will be in the directory where you execute the code.

Environment: Python3.7 on Windows Linux Subsystem (Ubuntu 18.04.1)

Contents:

1. Downsample the image (from last assignment) to 64x64 and binarize

```python
# Downsample to binarized 64x64
def downsample_binary(img_o):
    # 66x66 for boundaries handling
    img_t = np.zeros((68, 68), dtype=np.int32)
    for i in range(64):
        for j in range(64):
            img_t[i+2, j+2] = (0 if img_o[8 * i, 8 * j] < 128 else 1)
    return img_t
```

For each iteration, do:

(Save the original image)

2. Marked-interior/border-pixel operator

```python
# Mark-interior/border-pixel operator ((i, b) = (1, 2); 4-connected)
def operator_ib(img_o):
    img_t = np.zeros((68, 68), dtype=np.int32)
    for i in range(2, 66):
        for j in range(2, 66):
            if img_o[i, j] == 1:
                img_t[i, j] = 1
                for (x, y) in neighbor:
                    if img_o[i+x, j+y] != 1: img_t[i, j] = 2; break
    return img_t
```

3. Pair relationship operator

```python
# Pair relationship operator ((p = 3); 4-connected)
def operator_pr (img_o):
    img_t = np.zeros((68, 68), dtype=np.int32)
    for i in range(2, 66):
        for j in range(2, 66):
            if img_o[i, j] == 2:
                for (x, y) in neighbor:
                    if img_o[i+x, j+y] == 1: img_t[i, j] = 1; break
    return img_t
```

4. Marked-pixel connected shrink operator

```
def _h(b, c, d, e):
    if b == c and (b != d or b != e): return 1
    return 0

def yokoi(img_o, i, j):
    _f = [_h(img_o[i, j], img_o[i, j+1], img_o[i-1, j+1], img_o[i-1, j]),
          _h(img_o[i, j], img_o[i-1, j], img_o[i-1, j-1], img_o[i, j-1]),
          _h(img_o[i, j], img_o[i, j-1], img_o[i+1, j-1], img_o[i+1, j]),
          _h(img_o[i, j], img_o[i+1, j], img_o[i+1, j+1], img_o[i, j+1])]
    if np.array(_f).sum() == 1: return True
    return False
```

(Then delete the removable and marked pixels.)

(Compare the result to original one; If they are identical, break and output the image; else continue the loop.)

Result: