Computer Vision HW7, Thinning Operator Report

tags: `NTU CS` `Computer Vision` `Writeup` `Report`

NTU CSIE, R08922024, Alfons Hwu
Prequisites and env as the following

```
Ubuntu WSL for windows with jupyter notebook
Python3.6.7
OpenCV for image IO
Matplotlib for displaying image
```

a, thinning operator

6.2.8 Thinning Operator

- Step1
  - input : original symbolic image
  - marked-interior/border-pixel operator
  - output : interior/border image
- Step2
  - input : interior/border image
  - pair relationship operator
  - output : marked image
- Step3
  - input : original symbolic image +marked image
  - marked-pixel connected shrink operator
    - removable(by connected shrink operator on original symbolic image)
    - marked(by marked image)
    - delete those pixels satisfied the two conditions mentioned above
    - output : thinned output image
- use thinned output image as next original symbolic image
- repeat step1, step2, step3 until the last output never changed

Run as aforementioned step to thin image
Main driver functions shown as below, and the rest of detailed implementation of respective function, please check `utils.py` under this homework for more information.

```
def do_thinnig():
    img_bin = img_binarize(img)

    row, col = (img_bin.shape)
    row, col = row // 8, col // 8
    res_final = np.zeros((row, col), np.int)

    for i in range(row):
        for j in range(col):
            res_final[i, j] = img_bin[8 * i, 8* j]

    step_cnt = 0
    while True:
        # use numpy copy to prevent from changing to same memory block
        res_old = res_final
        if id(res_old) == id(res_final):
            res_old = np.copy(res_final)

        res_ib = do_ib(res_final)
        res_mp = do_mp(res_ib)

        res_yokoi = do_yokoi(res_final)
        res_to_delete = (res_yokoi == 1) * 1

        for i in range(row):
            for j in range(col):
                if res_to_delete[i, j] == 1 and res_mp[i, j] == 1:
                    res_final[i, j] = 0

        save_name = 'lena_thinned_step' + str(step_cnt) + '.png'
        cv2.imwrite(save_name, res_final)
        plt.imshow(res_final, cmap = 'gray')
        plt.show()
        step_cnt += 1

        if np.sum(res_old == res_final) == row * col:
            break
```

GIF version of compression progress among all iterations 👇.

PNG version of finished image.