Projet libre:

Segmentation of Corneal Basal Image

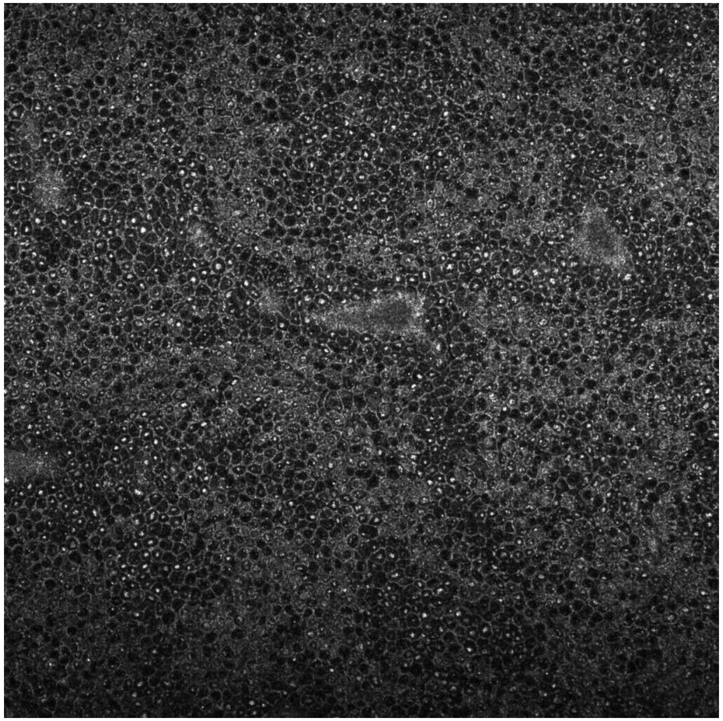
Sheng FANG
30/06/2014

Content

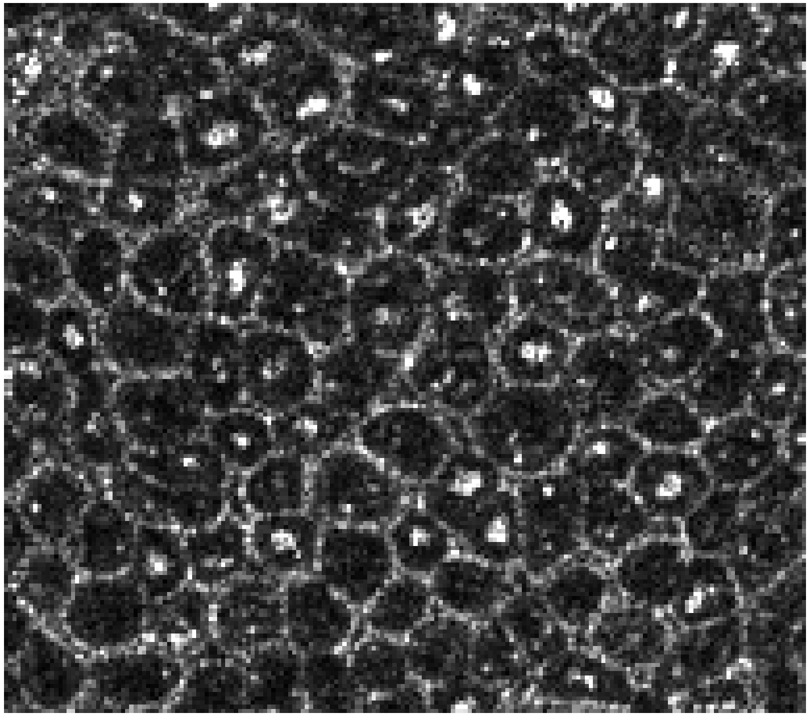
- Target
- Our method
- Failed methods
- Conclusion

Target

- Do segmentation on corneal basal image with lots of noise

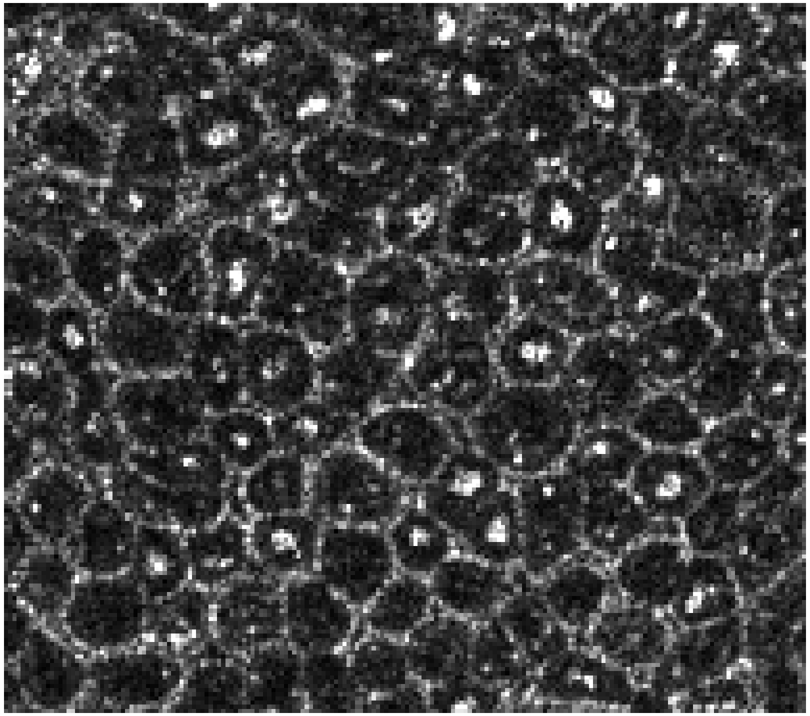


Original Image



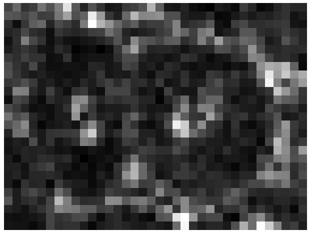
A small part

Target - Difficulties

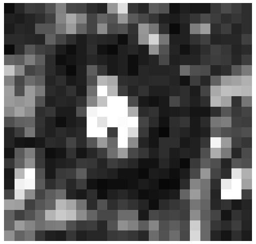


Difficulties:

1. The borders of the cells are noise and unclear.



2. Big noise in the middle of the cell



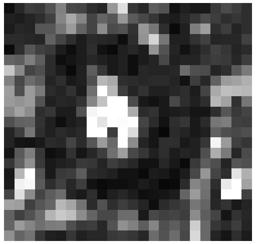
Target - Difficulties

- Difficulty 1 will cause unsuccessful segmentation of the cells
- Difficulty 2 will cause over-segmented

- Precise targets
- 1. Do noise reduction to get more clear border
- 2. Get rid of the noise part in the middle of the cell

Our method:
Preparation – Size measurement

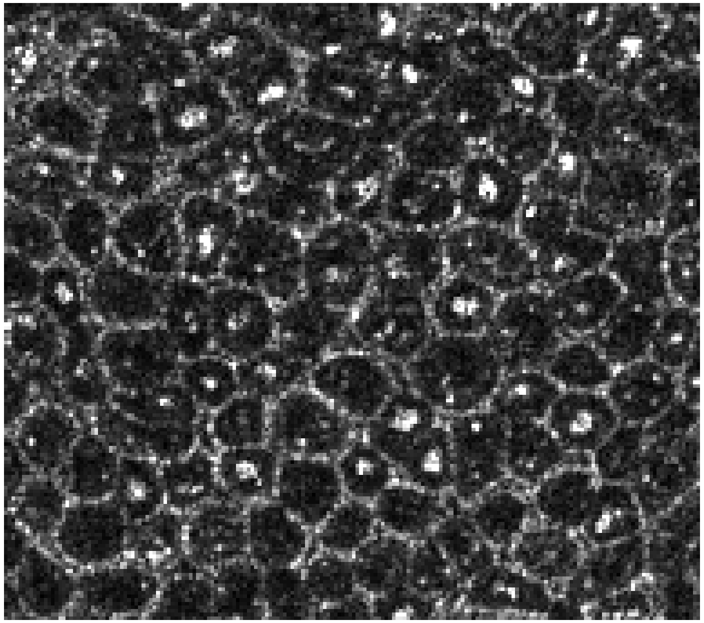
- Radius of the cell: 8 px
- Radius of the noise part: 3~4 px
- Width of the margin between cells: 3 px



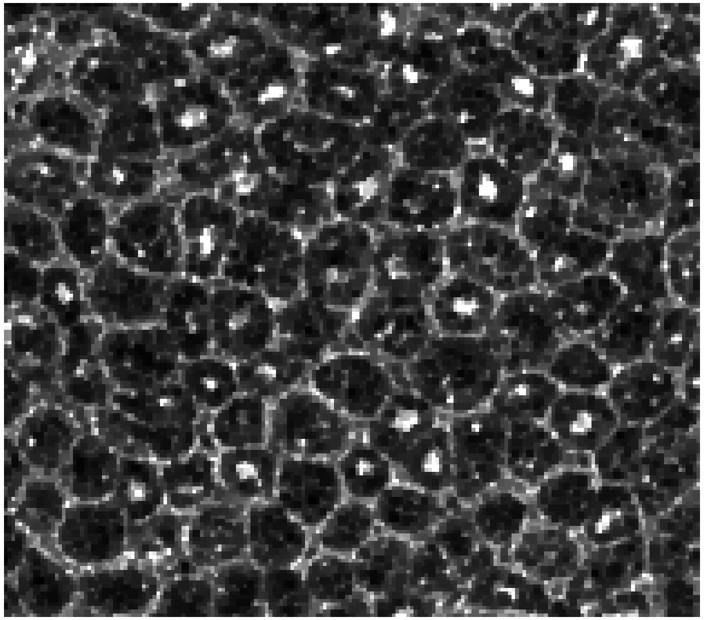
Our method:
Step 1: Make more connection

- A close operation with small size of construct element can make more connections of the borders.

`imc = imclose(im1,se);`



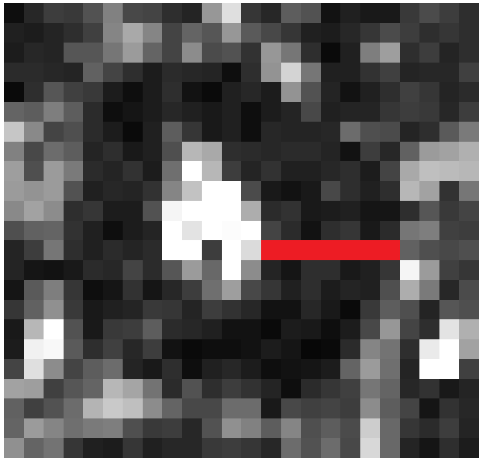
Original image



After close operation (imc)

Our method:
Step 1: Make more connection

- The size of the small construct element depends on the distance between the noise border and cell border.(The distance of red line)



If the size of the small construct element is bigger than the length of red line, this close operation will connect the margin and the noise part.

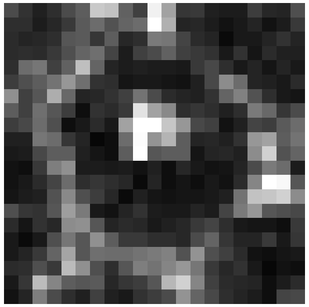
In this case, the size of the small construct should be smaller than (8 – 4 = 4) px. We choose 2 px in the code.

Our method:
Step 2: Get rid of the noise part

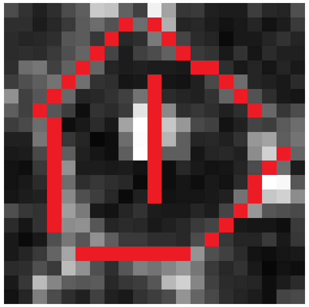
- In order to get rid of the noise part, we want to reconstruct the ‘imc’.
- We need a maker without the noise part.

- The method to get the maker is to filter the image with four construct element who have different orientations.
- Four orientations:
  - Horizon: — Vertical: |
  - 45 degree: / 135 degree: \

Our method:
Step 2: Get rid of the noise part



For example, apply erode operation with four direction construct element on the first image. Erode operation will set the smallest value in the construct element to the current pixel .
In this case, we can keep the border as white, while change the noise part to black.



The size of the construction element should be bigger than the size of noise part. In this condition, the smaller the size is, the better the performance is.
In our case, the size should be bigger than (4*2=8) px. We choose 10 px in the codes.

PS: Four direction is enough to build the maker, because we only want to keep rough borders and get rid of the noise part.

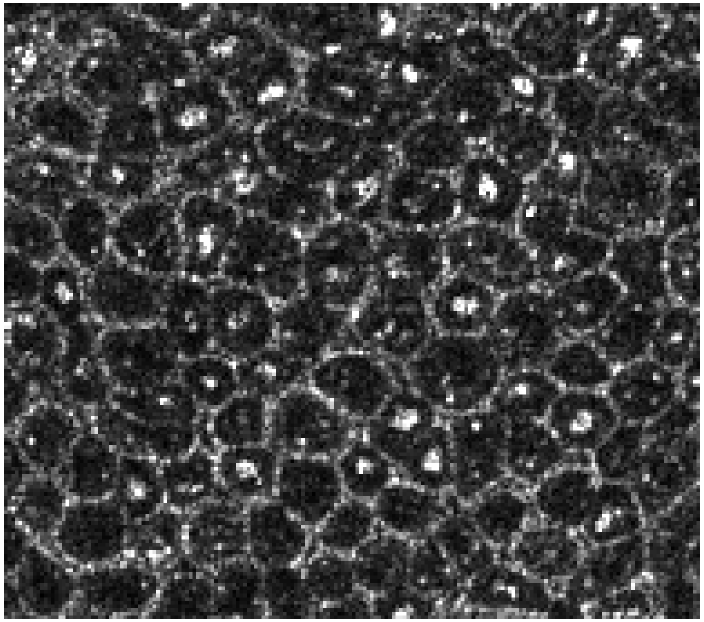
Our method:
Step 2: Get rid of the noise part

- The code to make the maker

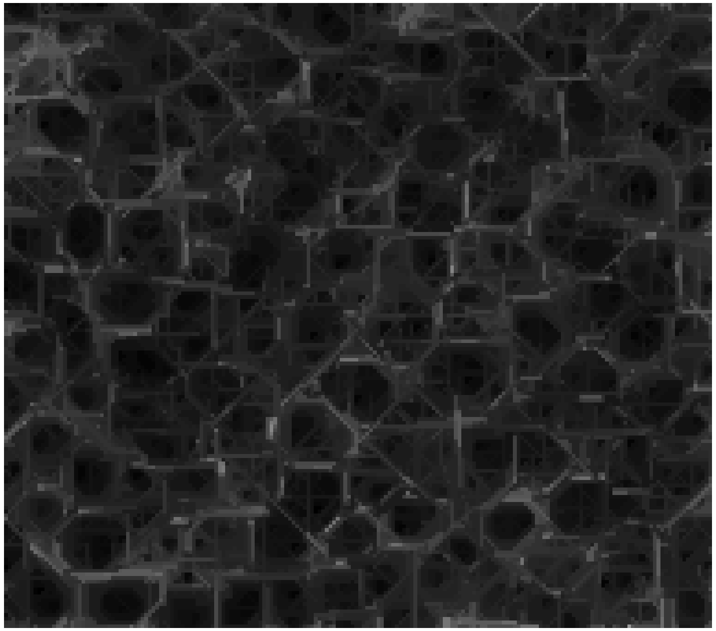
```
im1 = imerode(im,se1);
im2 = imerode(im,se2);
im3 = imerode(im,se3);
im4 = imerode(im,se4);
%se1, se2, se3, se4 are construct elements with different direction
%im1, im2, im3, im4 are the four images filter by the four elements
[row,colume,z] = size(im);

  for i = 1:row
    for j = 1:colume
      imc(i,j) = max(im1(i,j),max(im2(i,j),max(im3(i,j),im4(i,j))));
    end
  end
%Use 'for' loop to generate the maker pixel by pixel, using the maximum value of
pixel in same position of the four images.
```

Our method:
Step 2: Get rid of the noise part



Original image

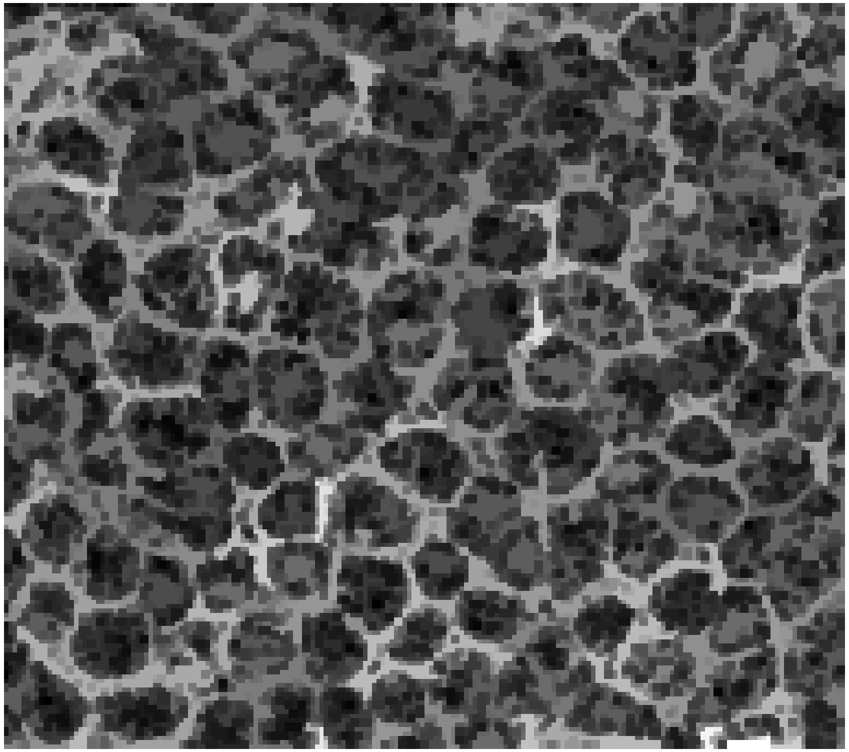


Maker(imc_rc)

Our method:
Step 2: Get rid of the noise part

- Reconstruct the image

```
imrecon = imreconstruct(imc_rc,imc);
```



Reconstructed image (imrecon)

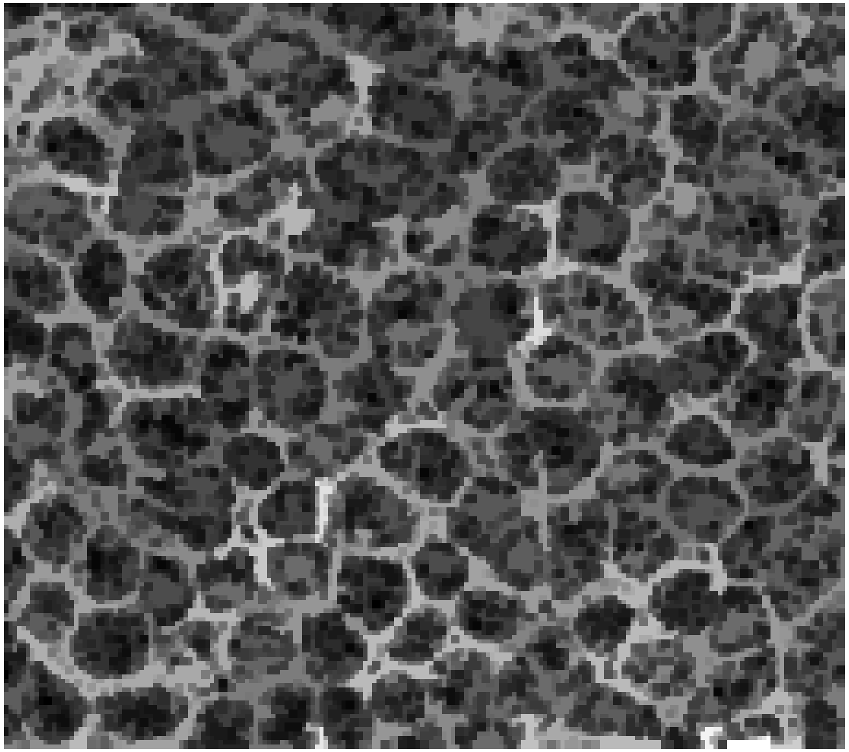
Our method:
Step 2: Get rid of the noise part

- The middle-noise part in image ‘imrecon’ is much weaker. But there are still many local minimum in the cells. This will still cause over-segmented. So we need to make the image more smooth.
- In our case, we use a close operation.

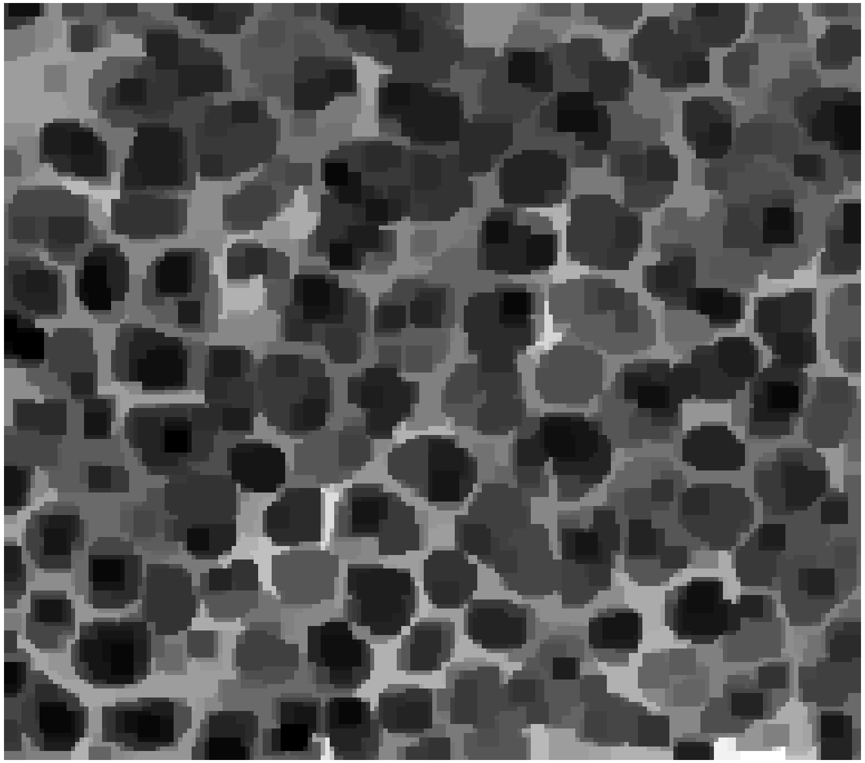
```
imbd = imclose(imrecon,se1);
```

The size of se1 should be smaller than the width of the margins in the image ‘imrecon’.

Our method:
Step 2: Get rid of the noise part



Reconstructed image(imrecon)

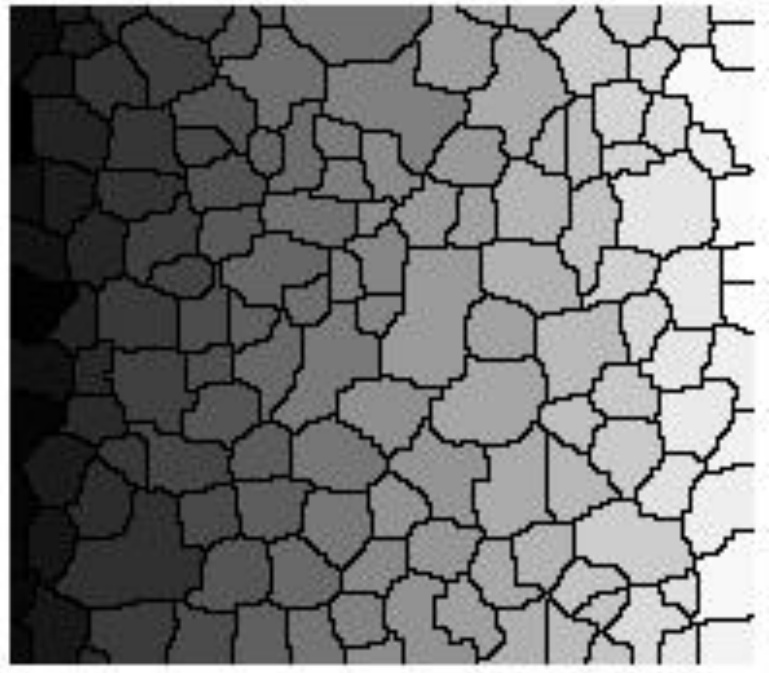


Reconstructed image after smooth (imbd)

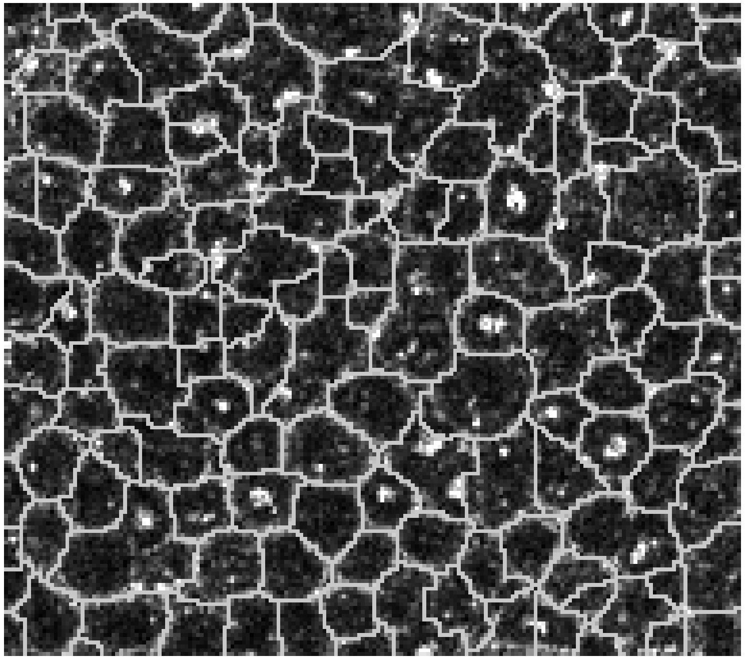
Our method:
Step 3: Watershed

Finally, we apply watershed on the image imbd

```
imf = watershed(imbd);
```

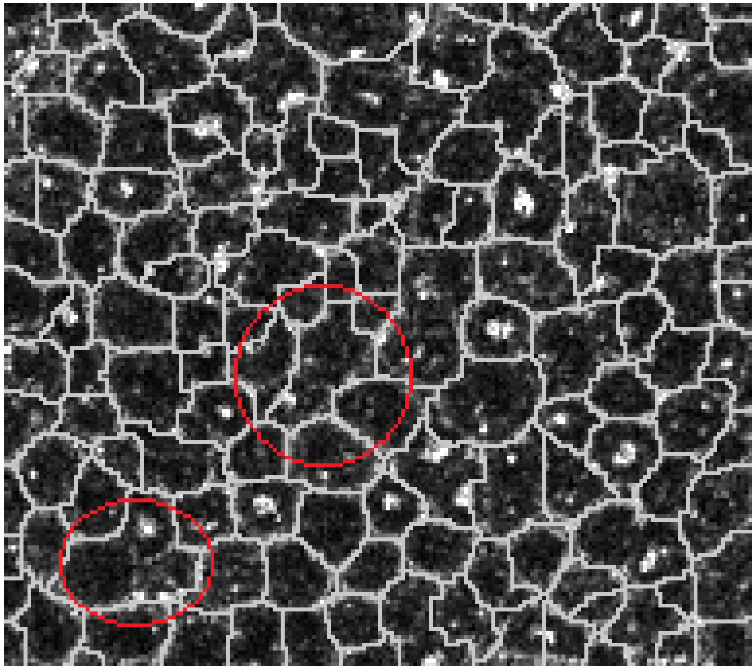


Segment result

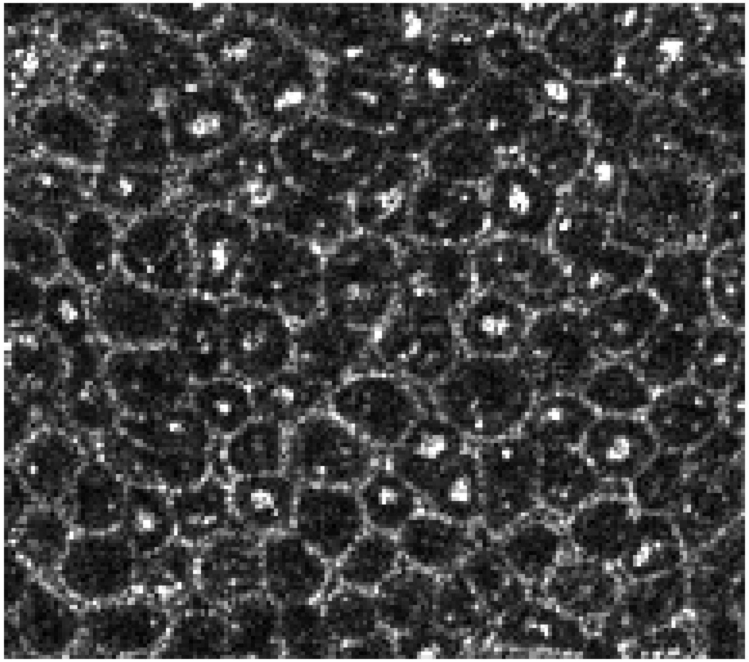


Merged result

Our method: Comparation



Merged result

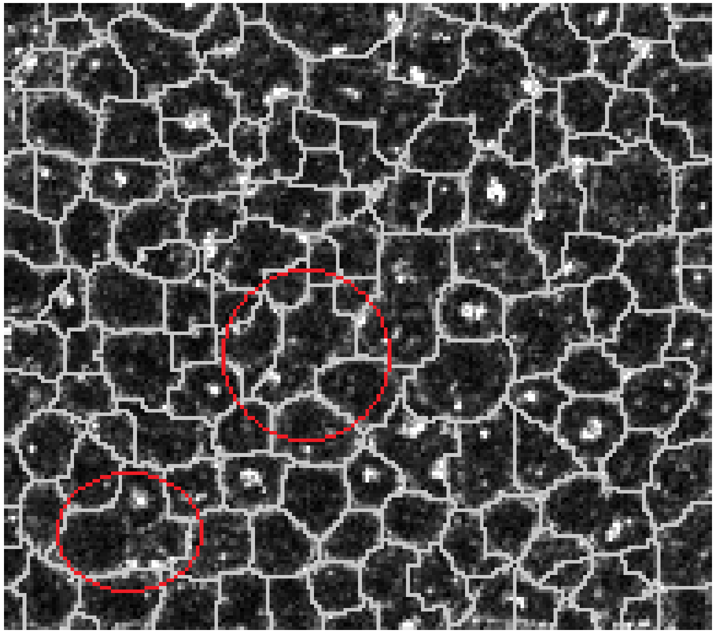


Original image

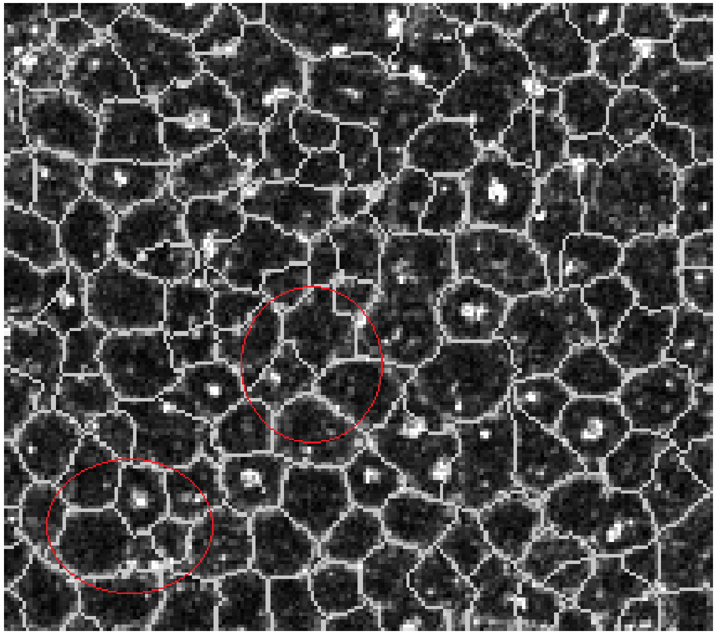
Look at the red circle, there exists some cells which are not segmented.

Our method: Improvement - 1

- imf = watershed(imbd,4);
- We add parameter 4 in the watershed function. 4 means the number of the neighbors when applying watershed



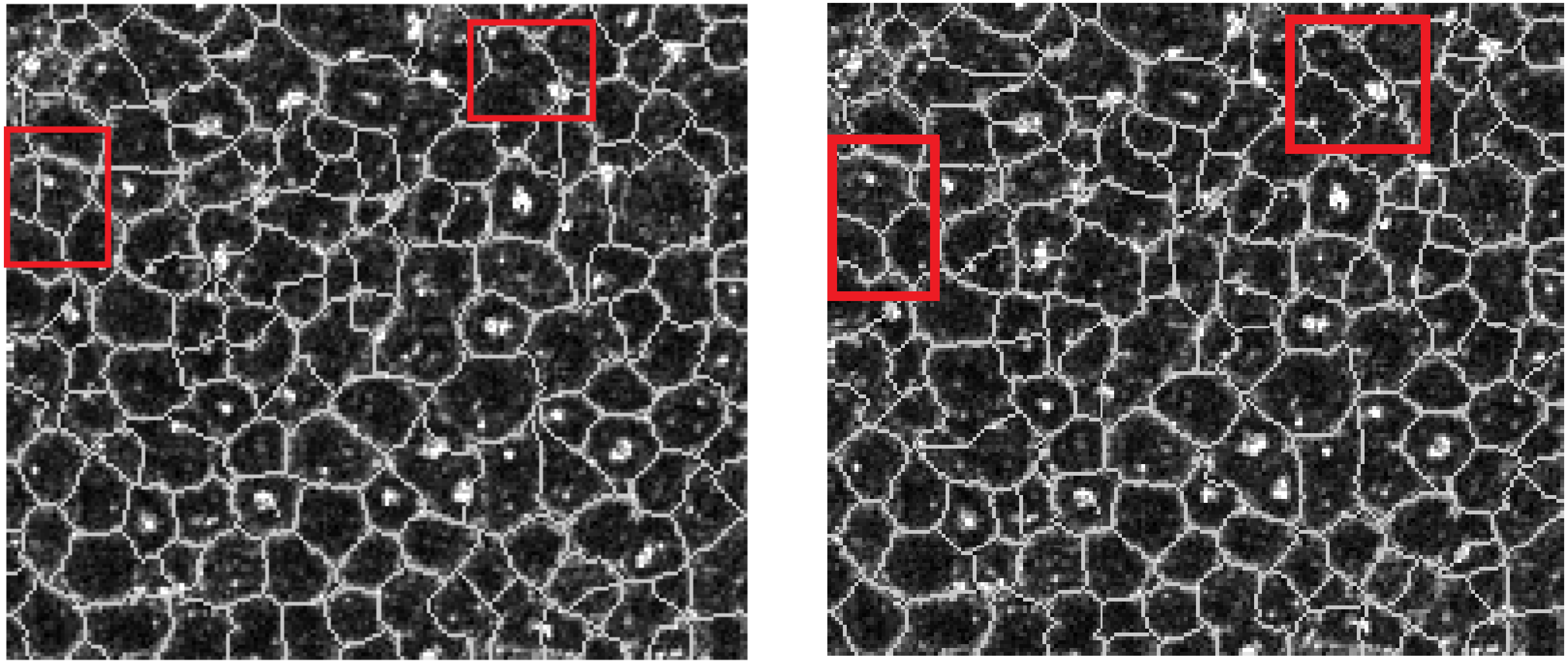
Before improvement



After improvement

Our method: Improvement - 2

- Add FFT Filter before the close operation(make more connection)

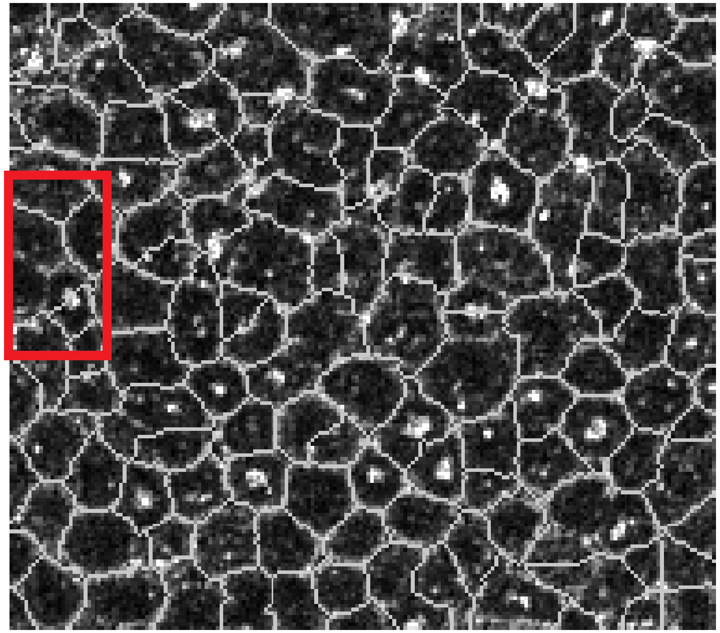


Our method: Improvement - 3

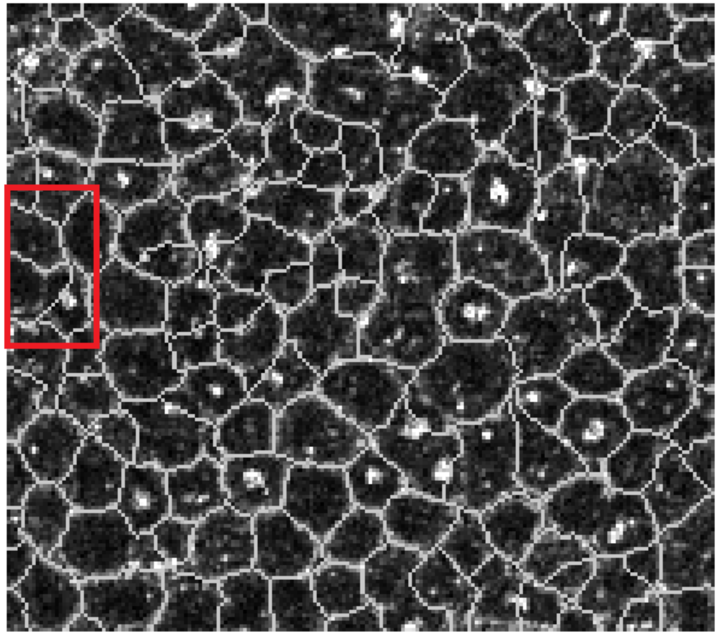
- In order to avoid the local minimum, we apply a close operation on the constructed image. In this step, the margins whose width are less than the size of se1 (5px in our case), are strongly weakened, so some cells will merge into one.
- In this case, we can set an interaction. We can let the customer choose the area with wrong results and apply a close operation with smaller size.

Failed methods - 1

- FFT Filter
- In order to get a clear border (noise reduction), I have tried FFT Filter instead of small close operation. We lost information of border, but not made more connection.



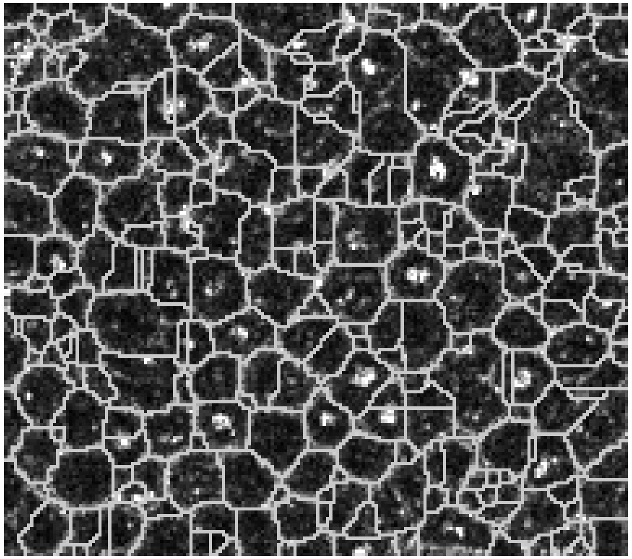
FFT filter



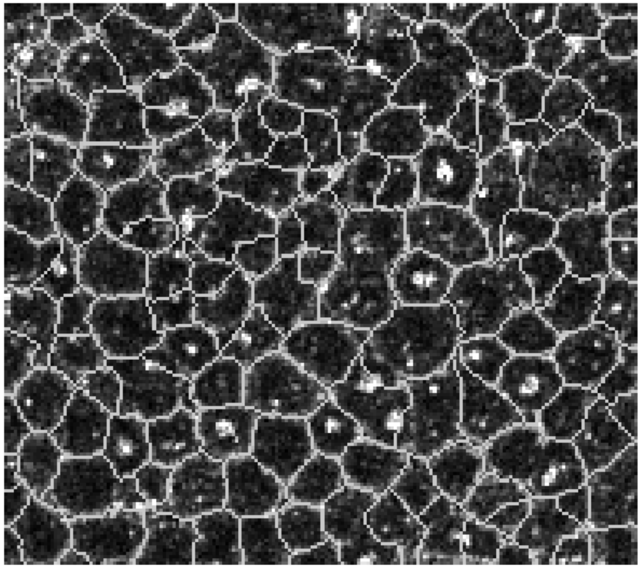
close operation

Failed methods - 2

- Binarilization
  - In order to get rid of the local minimum in reconstructed image, I have tried to apply binarilization. But it doesn't well separate the border and local minimum.



Binarilization operation



Close operation

Conclusion

- Our method can overcome the noise part in the middle of the cell
- However, some future work can be done to improve the result automatically.
    - For example, calculate the area of cell and then the average area. Re-apply the method with smaller se1(when smooth the reconstructed image) on the one whose area is much bigger than the average.

Segmentation of Corneal Basal Image

Thank you!