Medical Image Analysis

6. Medical image segmentation(1)

Taeyang Yang

Oct. 2020

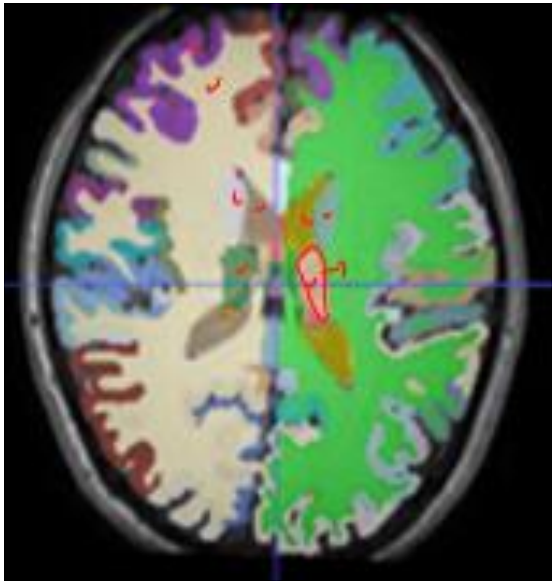
https://tyami.github.io/

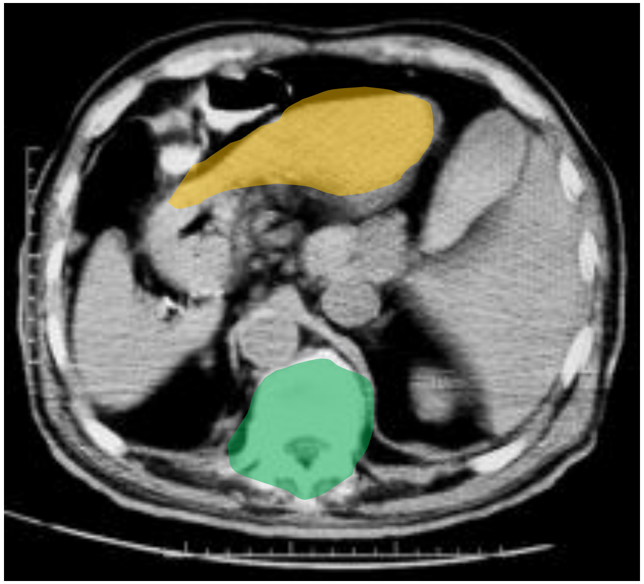
1. Introduction to medical image segmentation

Segmentation in medical image analysis

- Longitudinal study
  - 1년 후 특정 ROI가 어떻게 변화했는지 확인 가능

- 특정 ROI (e.g. 종양) 의 비율





1. Introduction to medical image segmentation

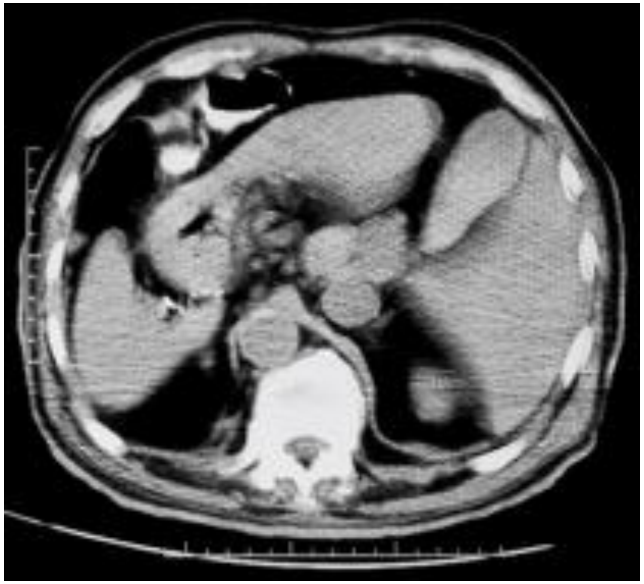
Contents

	Conventional methods	Deep learning methods
Segmentation	**Intensity/Color-based** - Thresholding - Region growing **Prior information-based** - Graph cut - Active contour model **Learning-based** - Active shape model	FCN U-Net Deep lab

2. Otsu thresholding

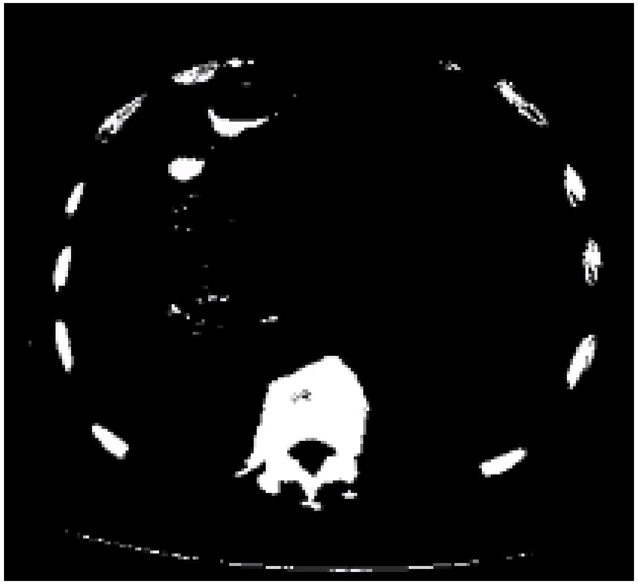
Manual threshold

(Manual) Voxel intensity > Threshold



0~255

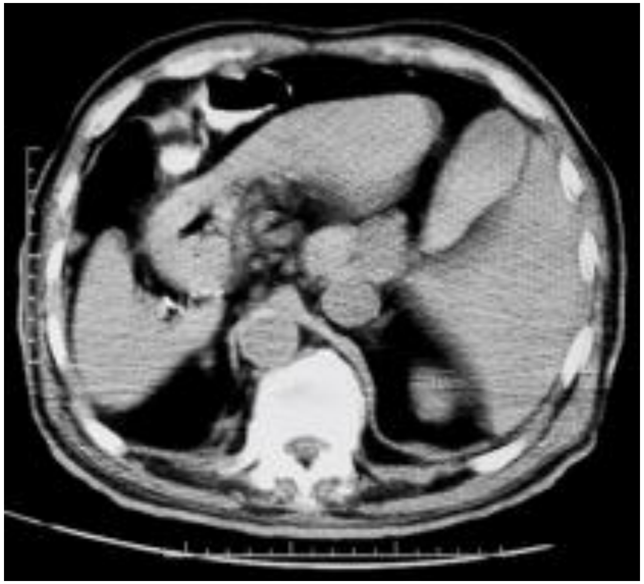
Threshold=200

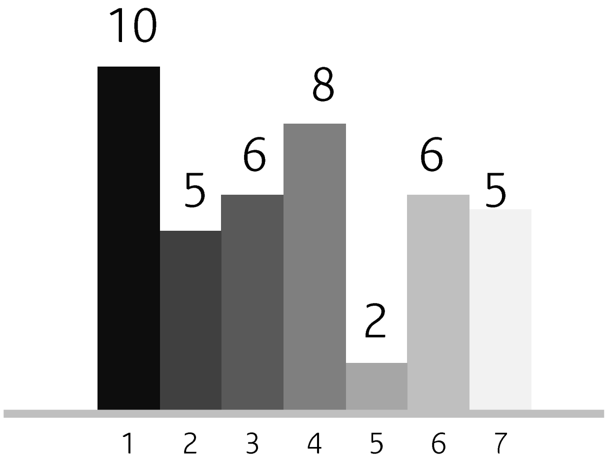


2. Otsu thresholding

Otsu thresholding: Automatic binarization

(Automatic) Voxel intensity > Threshold

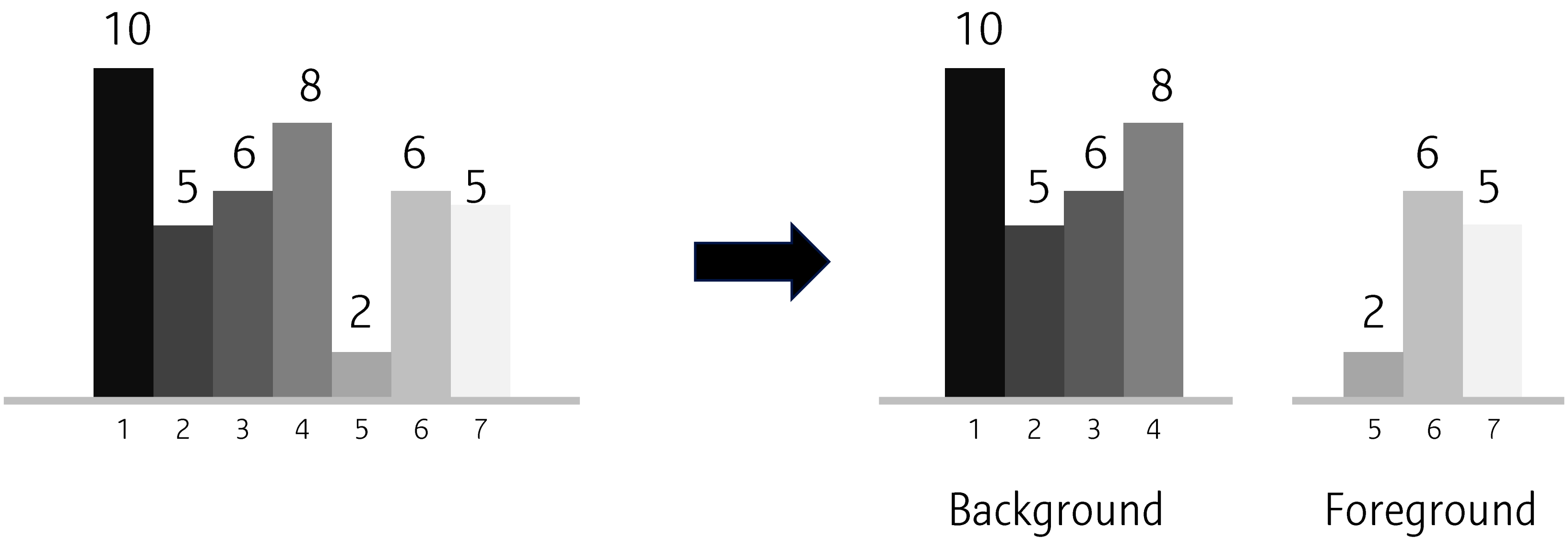




2. Otsu thresholding

Otsu thresholding: Procedure 1/5

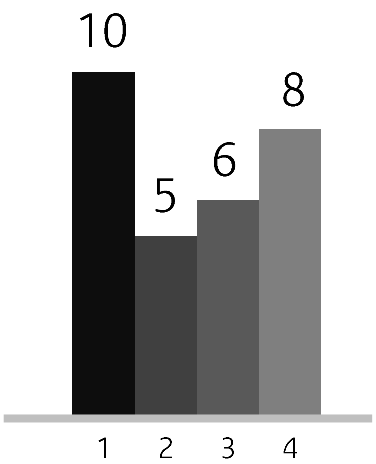
Split into two class (background & foreground)



2. Otsu thresholding

Otsu thresholding: Procedure 2/5

Calculate Weight, Mean, and Variance for each class



$$W_b = \frac{10+5+6+8}{10+5+6+8+2+6+5} = \frac{29}{42} = 0.690$$

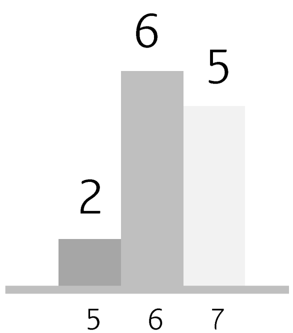
$$\mu_b = \frac{1\times 10 + 2\times 5 + 3\times 6 + 4\times 8}{10+5+6+8} = 2.414$$

$$\sigma_b = \frac{\left((1-2.414)^2\times 10\right) + \left((2-2.414)^2\times 5\right) + \left((3-2.414)^2\times 6\right) + \left((4-2.414)^2\times 8\right)}{10+5+6+8} = 1.484$$

2. Otsu thresholding

Otsu thresholding: Procedure 3/5

Calculate Weight, Mean, and Variance for each class



$$W_f = \frac{2+6+5}{10+5+6+8+2+6+5} = \frac{13}{42} = 0.310$$

$$\mu_f = \frac{5 \times 2 + 6 \times 6 + 7 \times 5}{2+6+5} = \frac{13}{42} = 6.231$$

$$\sigma_f = \frac{\left((5-6.231)^2 \times 2\right) + \left((6-6.231)^2 \times 6\right) + \left((7-6.231)^2 \times 5\right)}{2+6+5} = 0.485$$

2. Otsu thresholding

Otsu thresholding: Procedure 4/5

Calculate $\frac{\sigma_W^2}{\sigma_B^2}$

$$Maximize \frac{\sigma_B^2(T)}{\sigma_W^2(T)}$$

$$\sigma_W^2 = W_b \sigma_b^2 + W_f \sigma_f^2$$

$$\sigma_B^2 = \sigma^2 - \sigma_W^2 = W_b W_f \left(\mu_b - \mu_f\right)^2$$

$W_b = 0.690$

$\mu_b = 2.414$

$\sigma_b = 1.484$

$$\sigma_W^2 = 0.690 \times (1.484)^2 + 0.310 \times (0.485)^2 = 1.593$$

$$\sigma_B^2 = 0.690 \times 0.310(2.414 - 6.231)^2 = 3.114$$

$$\boldsymbol{\frac{\sigma_W^2}{\sigma_B^2} = \frac{1.593}{3.114} = 0.512}$$

$W_f = 0.310$

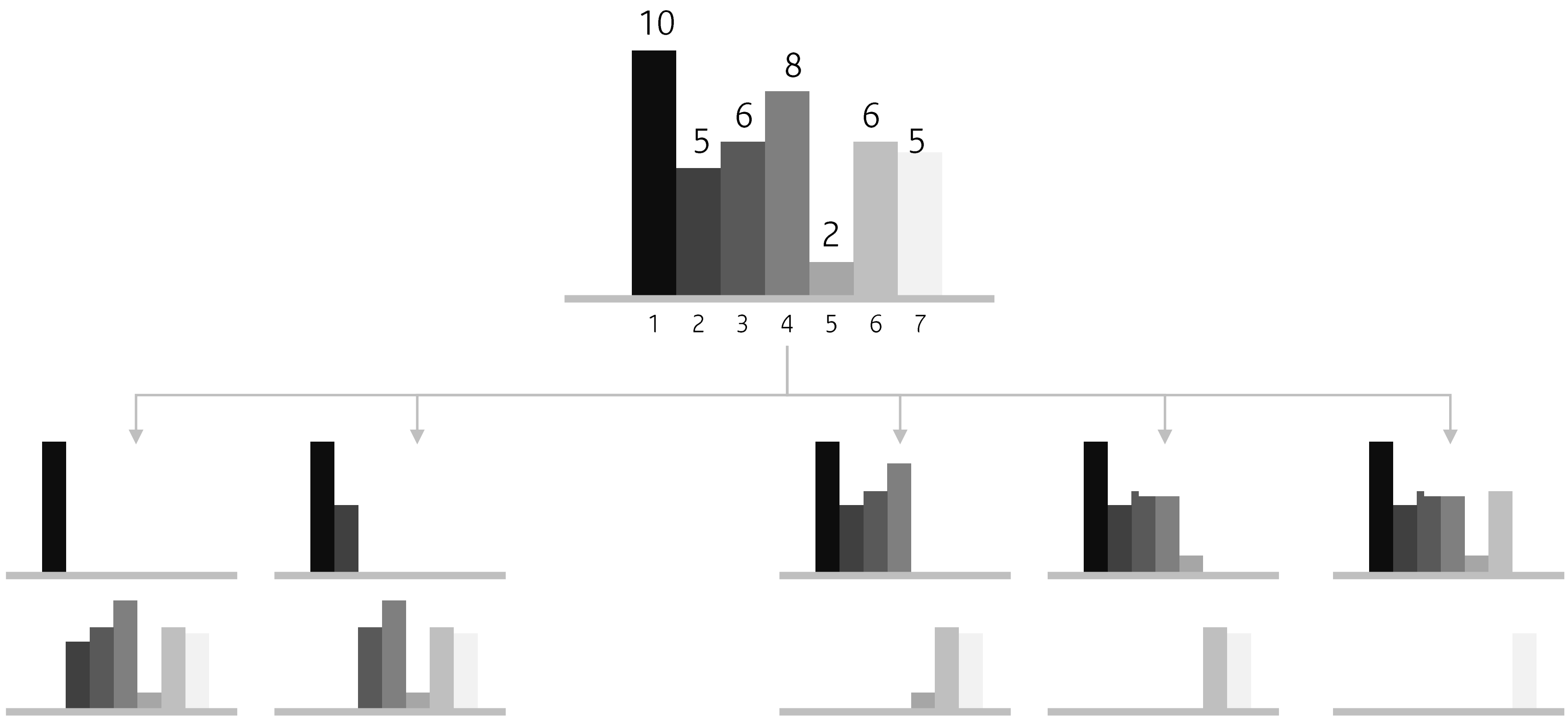
$\mu_f = 6.231$

$\sigma_f = 0.485$

2. Otsu thresholding

Otsu thresholding: Procedure 5/5

Find threshold with maximum $\frac{\sigma_W^2}{\sigma_B^2}$

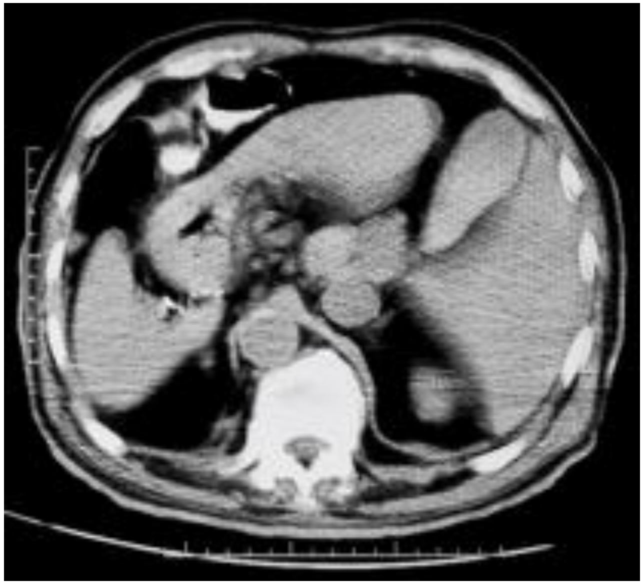


$\frac{\sigma_W^2}{\sigma_B^2}$ $\frac{\sigma_W^2}{\sigma_B^2}$ • • • $\frac{\sigma_W^2}{\sigma_B^2}$ $\frac{\sigma_W^2}{\sigma_B^2}$ $\frac{\sigma_W^2}{\sigma_B^2}$

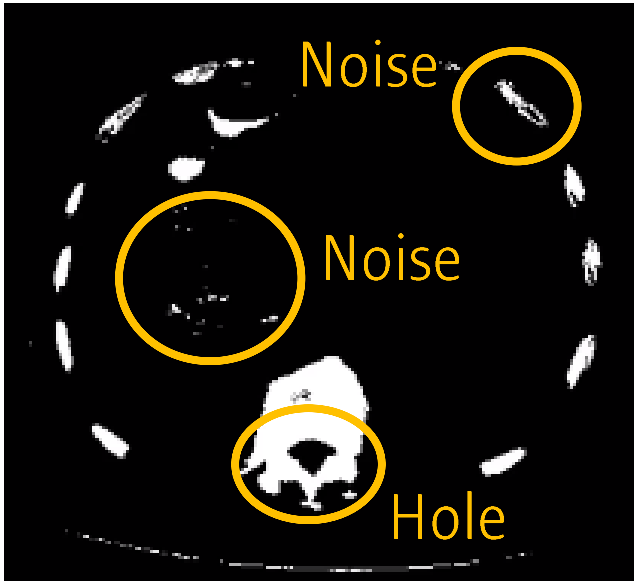
3. Morphological processing

Problem of threshold-based algorithms

Noise of threshold-based algorithms



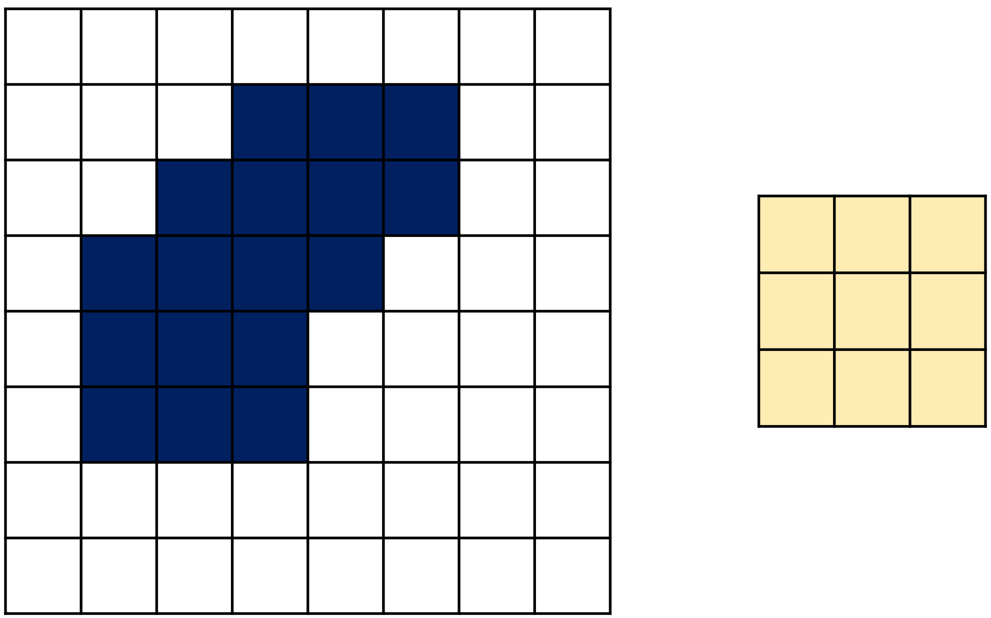
Threshold=200



3. Morphological processing

Morphological processing: Dilation

Dilation: Reverse of convolution



3. Morphological processing

(Revisit) Convolution

Data

9	9	9 (1/9)	0 (1/9)	0 (1/9)	0
9	9	9 (1/9)	0 (1/9)	0 (1/9)	0
9	9	9 (1/9)	0 (1/9)	0 (1/9)	0
9	9	9	0	0	0
9	9	9	0	0	0
9	9	9	0	0	0

Output

9	6	3	

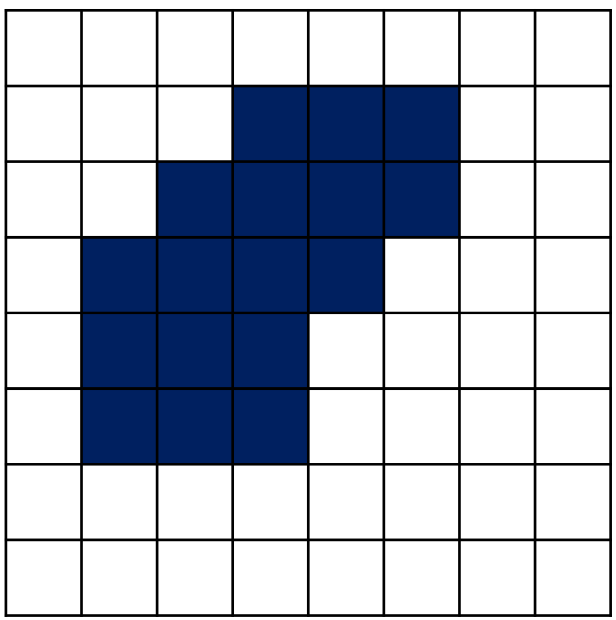
$$9\times\frac{1}{9}+9\times\frac{1}{9}+9\times\frac{1}{9}+0\times\frac{1}{9}+0\times\frac{1}{9}+0\times\frac{1}{9}+0\times\frac{1}{9}+0\times\frac{1}{9}+0\times\frac{1}{9}=6$$

3. Morphological processing

Morphological processing: Dilation and Erosion

Filters for dilation/erosion

Input



Diverse
Structural element
(filter)

0	1	0
1	1	1
0	1	0

0	1	1
1	1	1
0	1	1

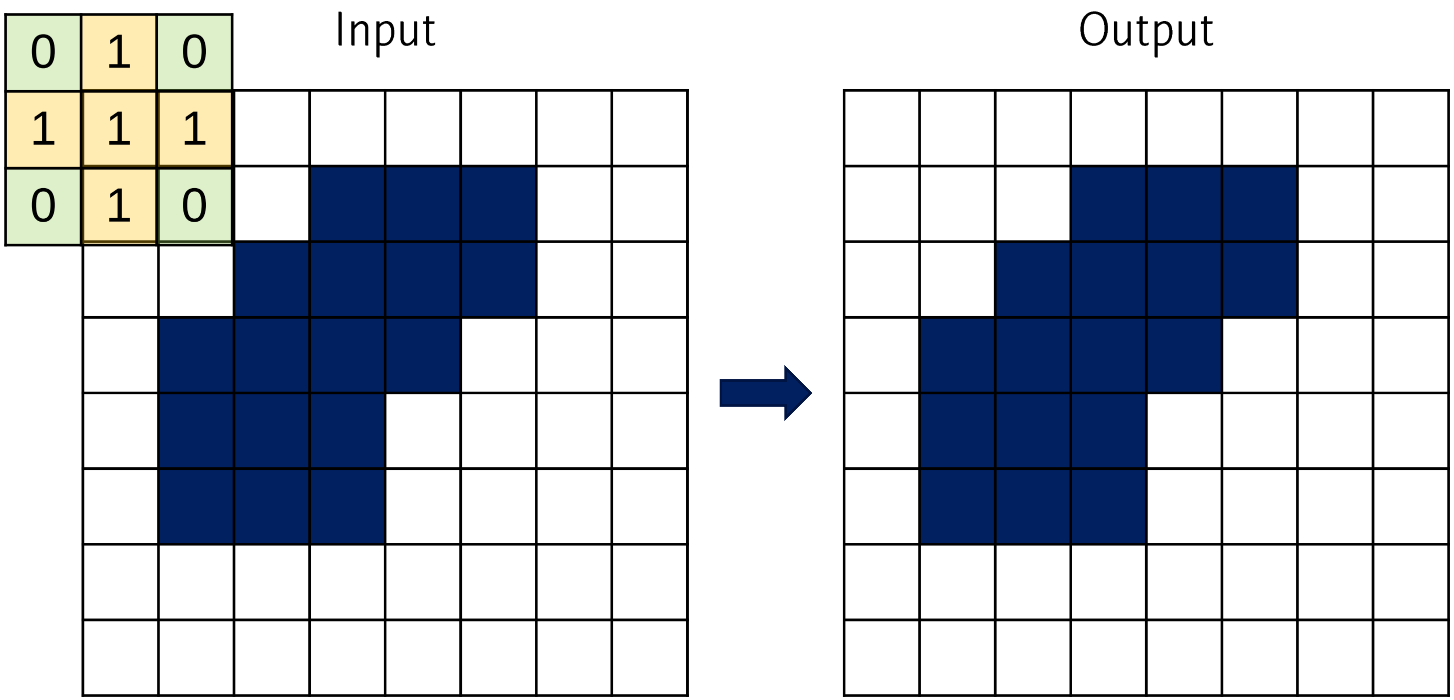
0	1	0
1	1	1
0	1	1

0	1	0
0	1	1
1	1	1

3. Morphological processing

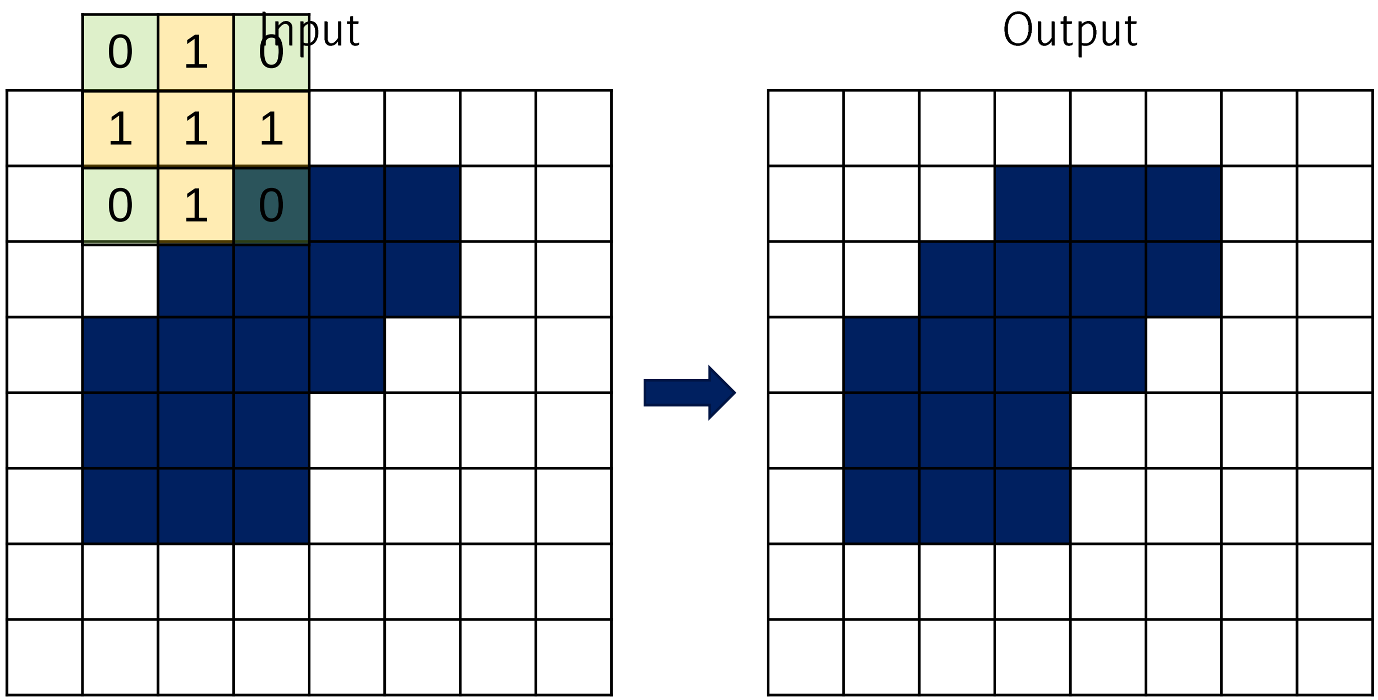
Morphological processing: Dilation

Find overlap between 1 and foreground (**navy** pixel)



Morphological processing: Dilation

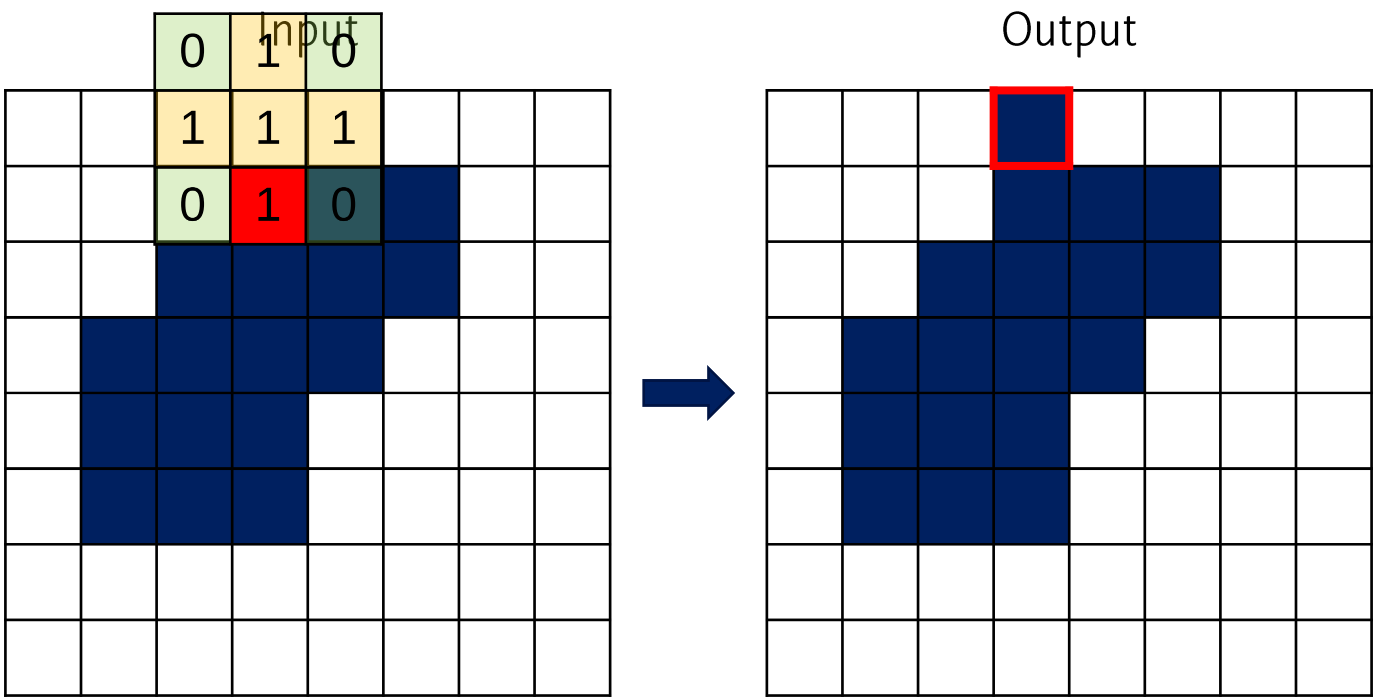
Find overlap between 1 and foreground (**navy** pixel)



3. Morphological processing

Morphological processing: Dilation

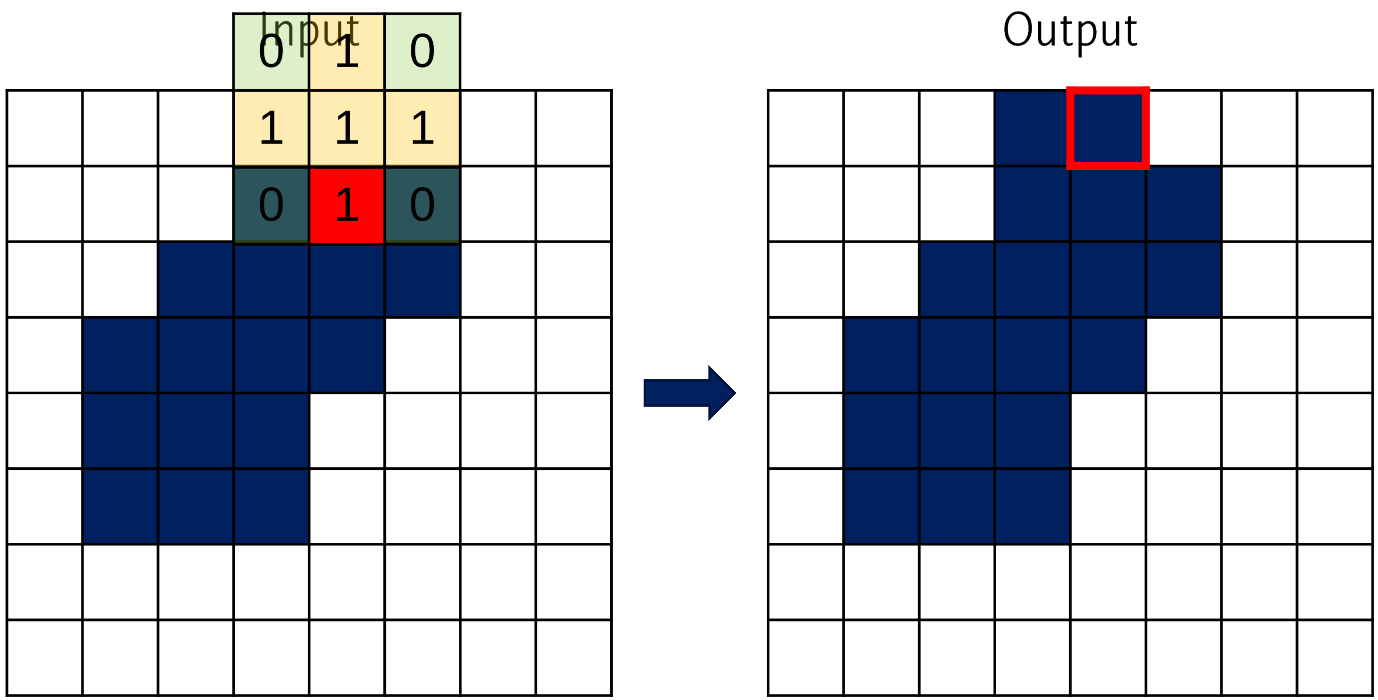
Find overlap between 1 and foreground (**navy** pixel)



3. Morphological processing

Morphological processing: Dilation

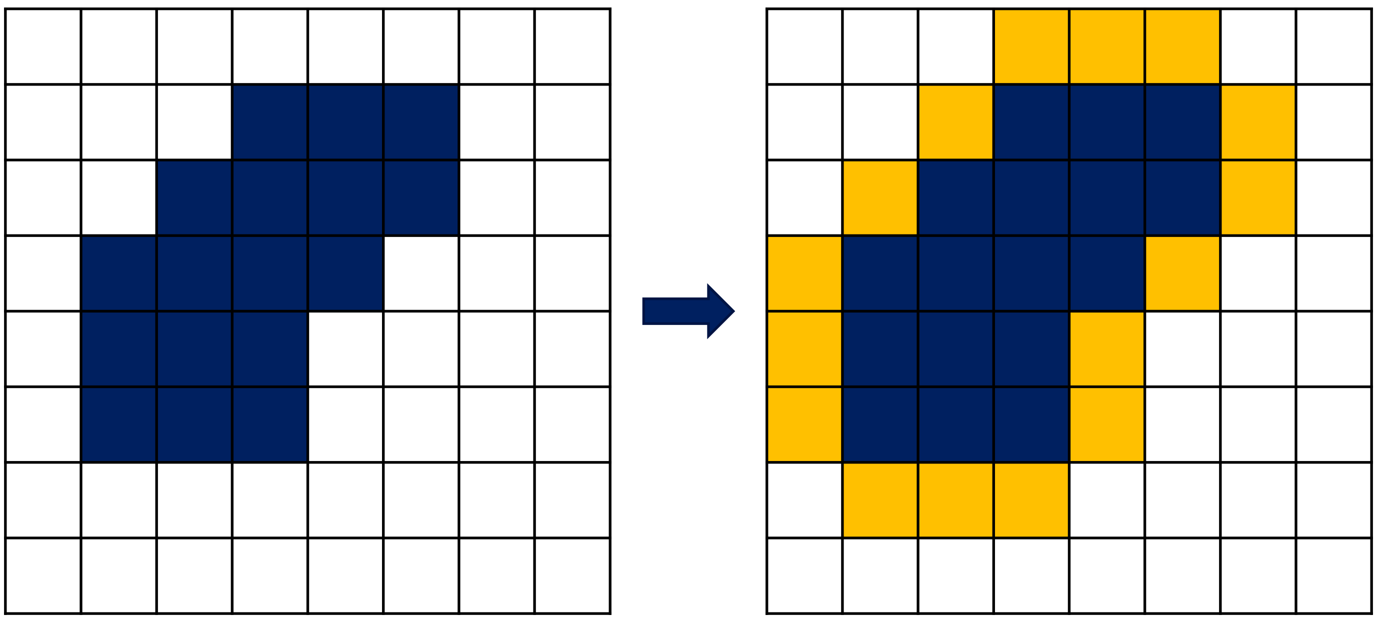
Find overlap between 1 and foreground (**navy** pixel)



Morphological processing: Dilation

Dilation: Increase of foreground

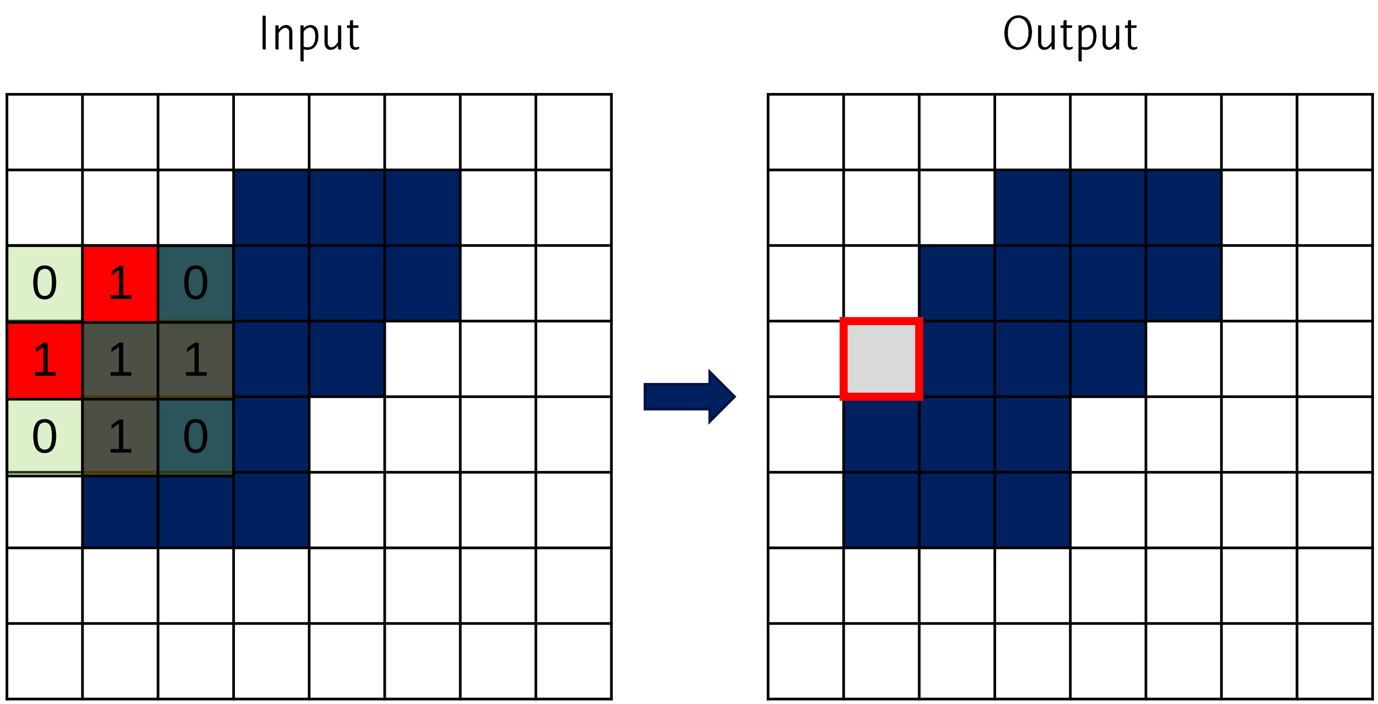
Dilation: Reverse of convolution



3. Morphological processing

Morphological processing: Erosion

Find overlap between 1 and background (white pixel)

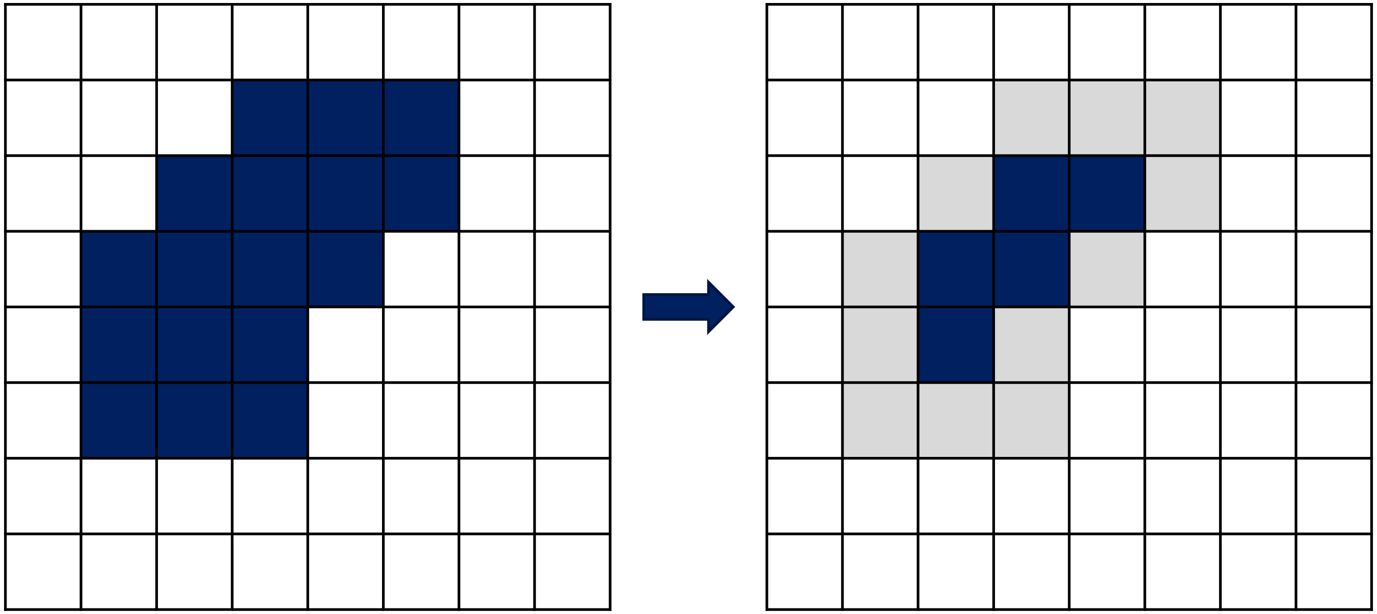


3. Morphological processing

Morphological processing: Erosion

Erosion: Increase of background

Dilation: Reverse of convolution

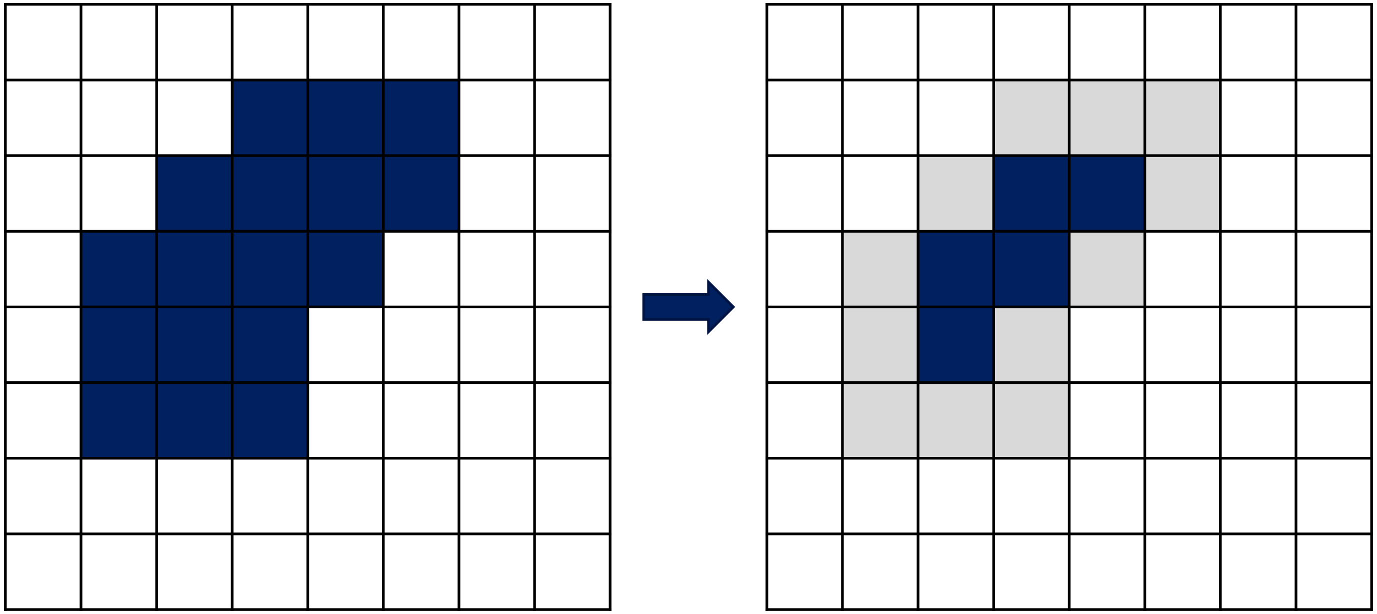


3. Morphological processing

Morphological processing: Erosion

Erosion: Increase of background

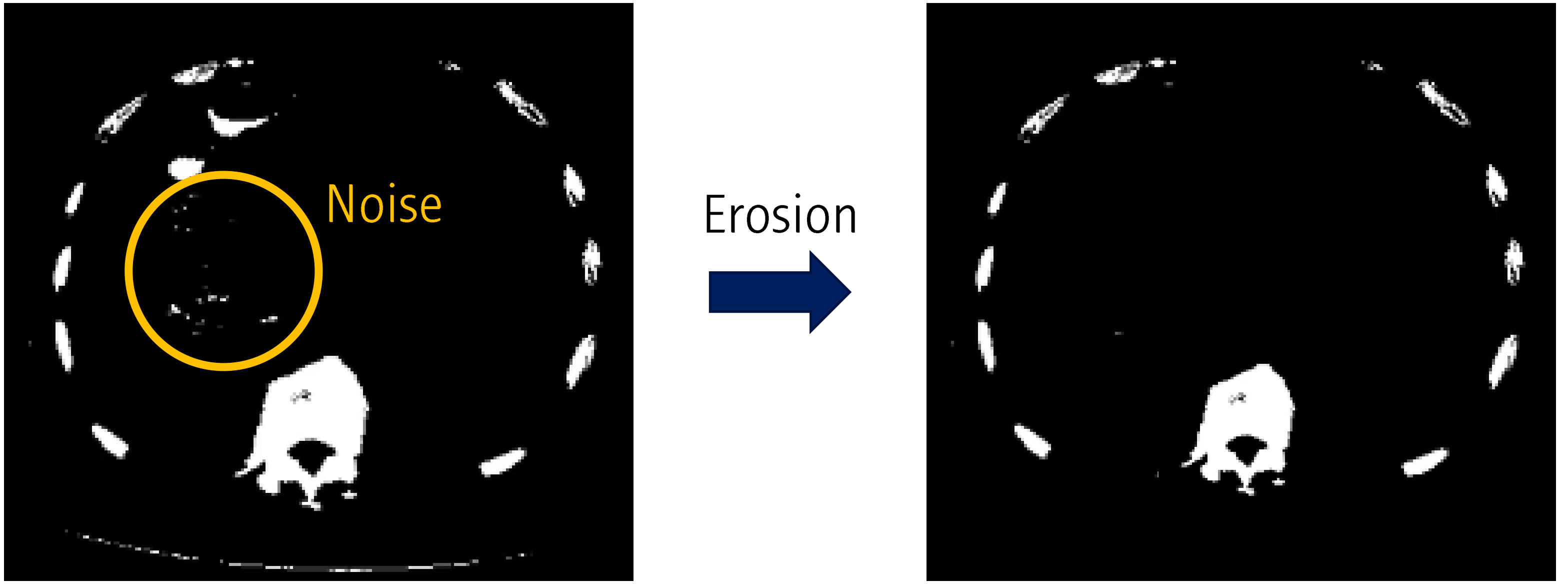
Dilation: Reverse of convolution



3. Morphological processing

Use of erosion

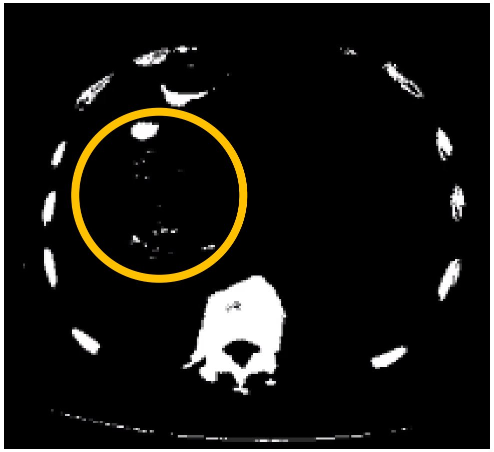
Reduction of noise, but decreased foreground size



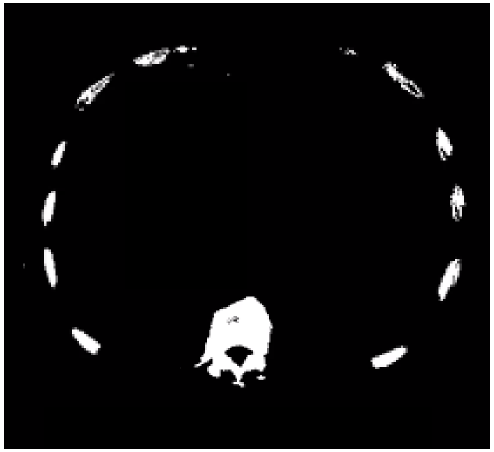
3. Morphological processing

Opening: Erosion ➔ Dilation

Opening

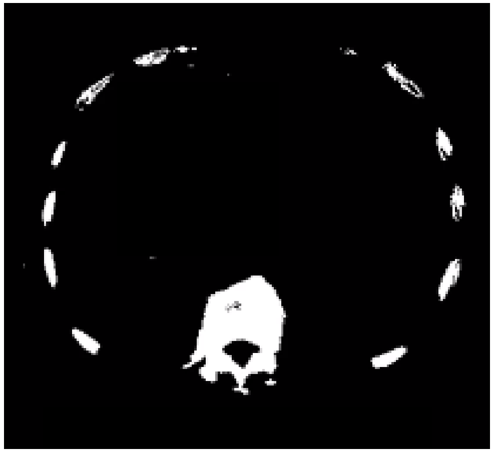


Erosion



Reduction of noise

Dilation

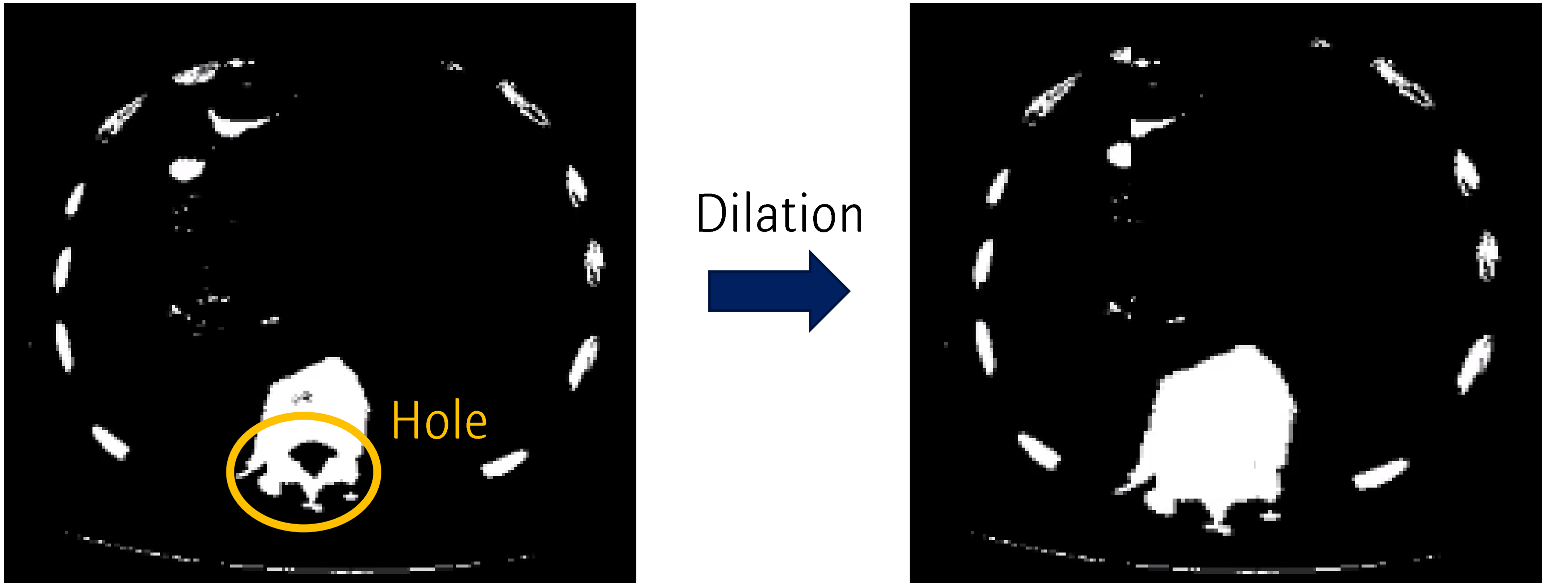


Resize of foreground

3. Morphological processing

Use of dilation

Reduction of hole, but increased foreground size

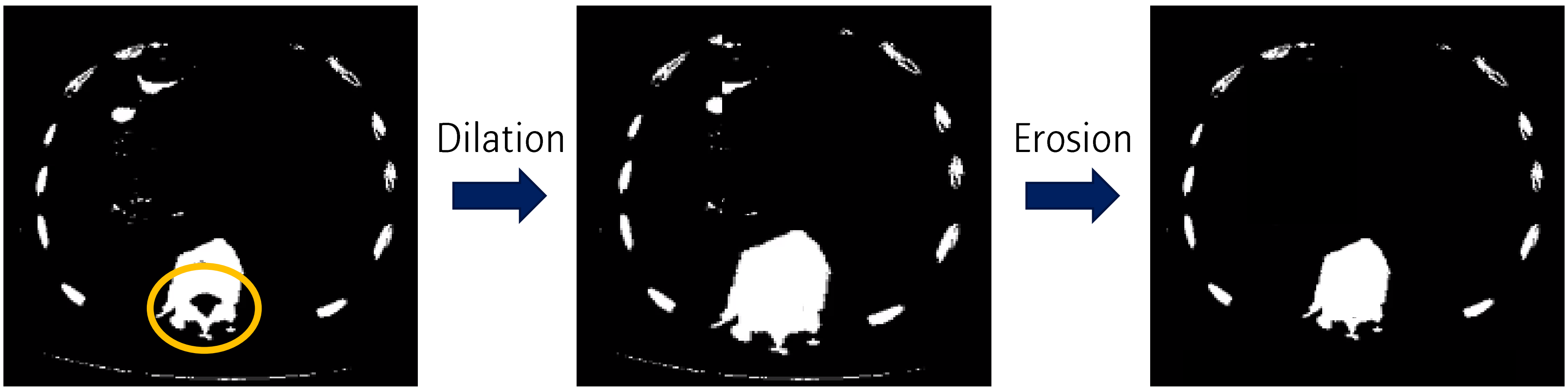


Reductio of hole

3. Morphological processing

Closing: Dilation → Erosion

Closing

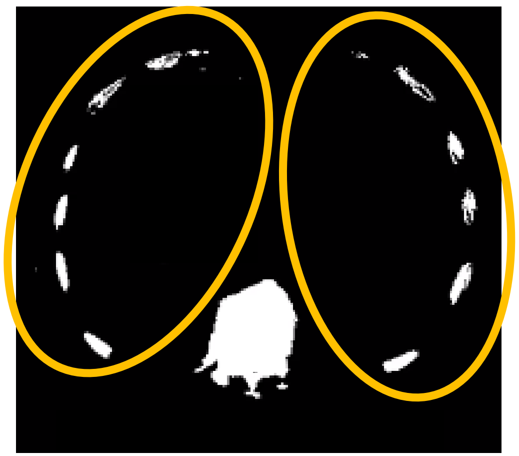


Reduction of hole

Reduction of noise
Resize of foreground

3. Morphological processing

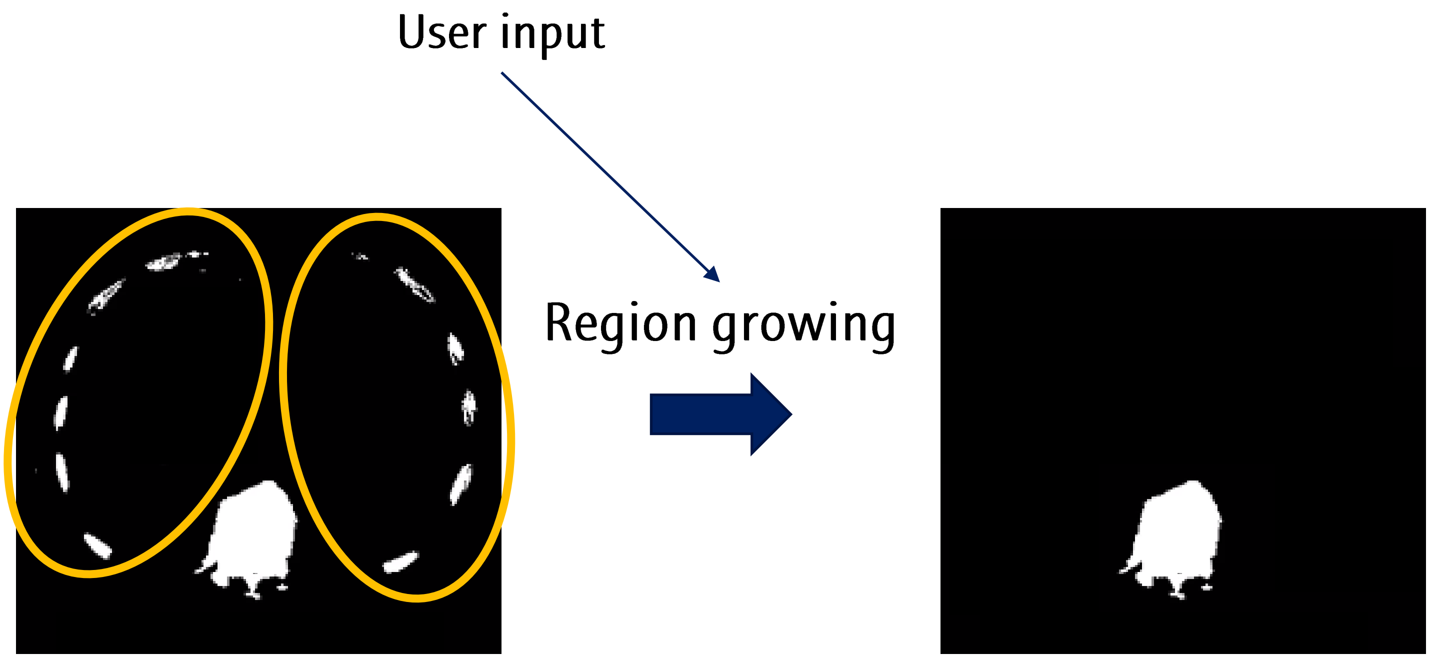
Drawback of morphological processing



Region growing

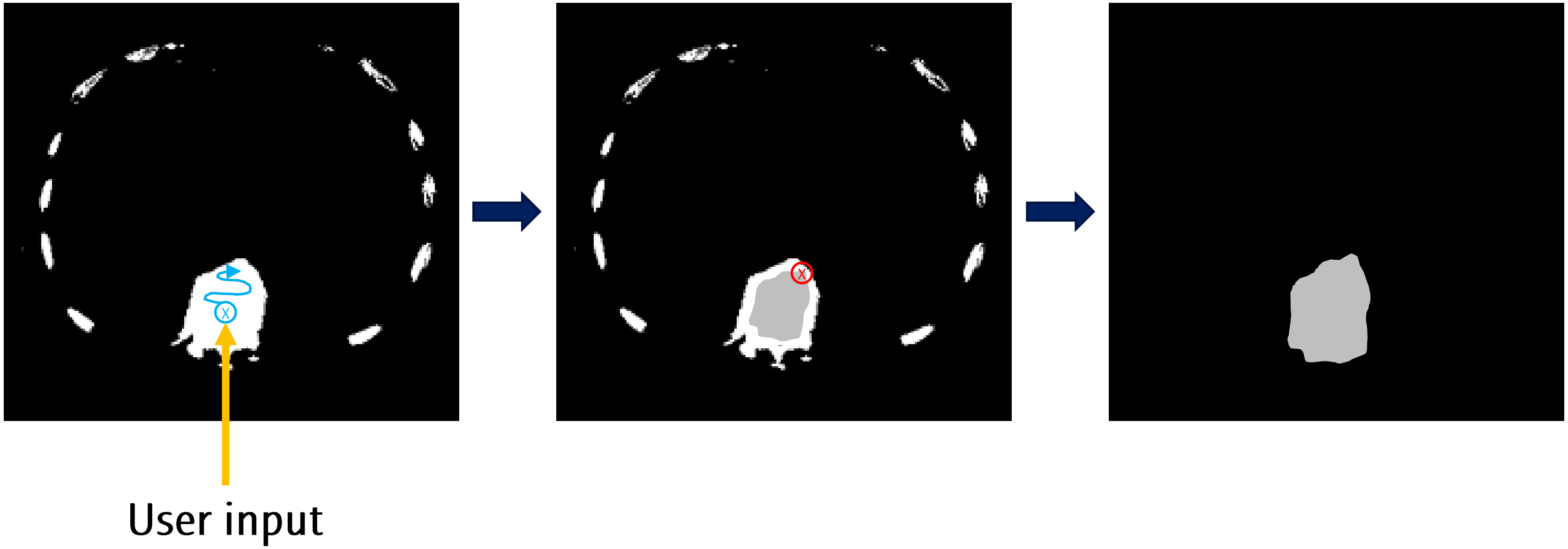
4. Region growing / Watershed algorithm

Region growing



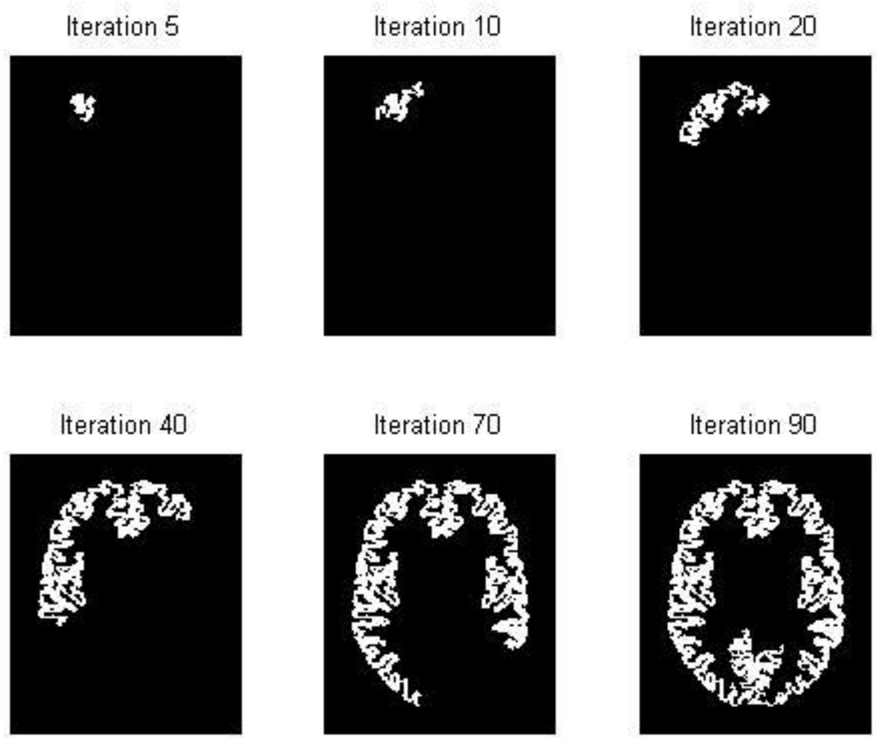
4. Region growing / Watershed algorithm

Region growing



4. Region growing / Watershed algorithm

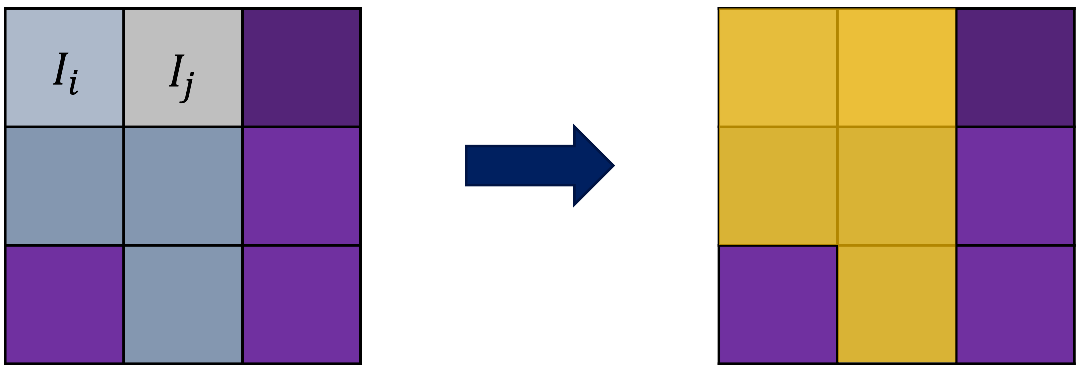
Region growing (brain image)



4. Region growing / Watershed algorithm

Region growing in RGB image

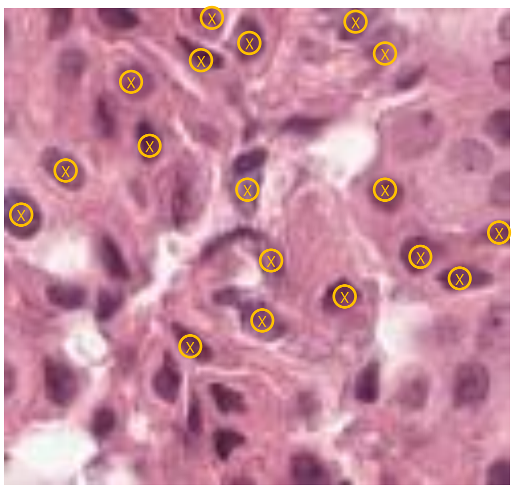
$$If\ \left|I_i - I_j\right| > threshold, expand$$



4. Region growing / Watershed algorithm

Drawback of region growing

Time consuming for input

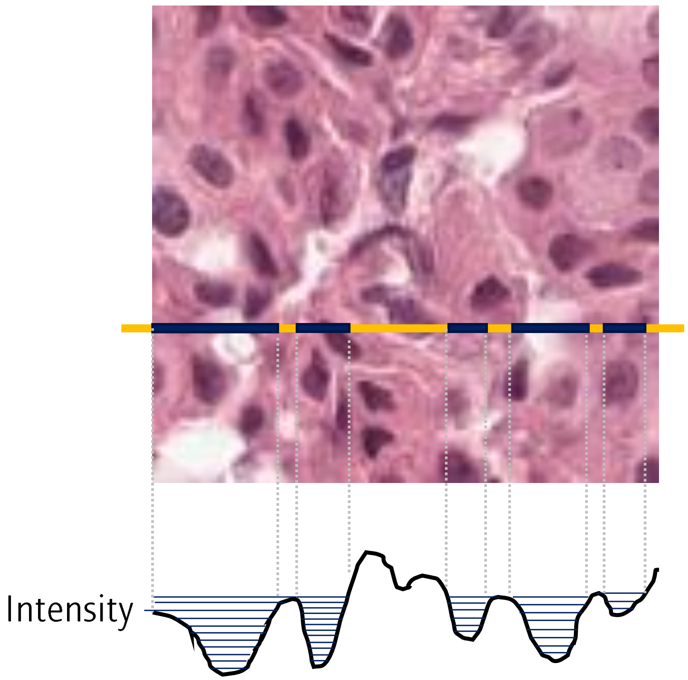


4. Region growing / Watershed algorithm

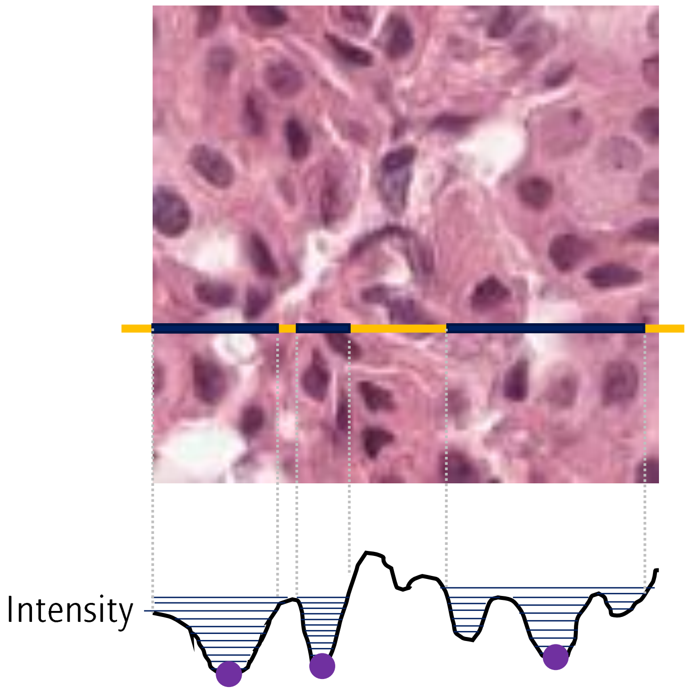
Watershed algorithm: Automatic region growing

Marker ➔ Region integrity

Without marker

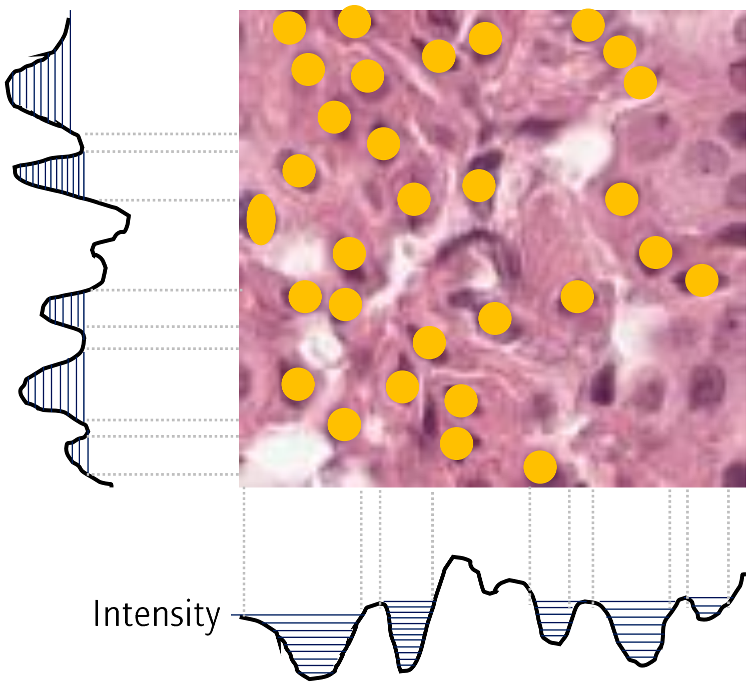


With marker



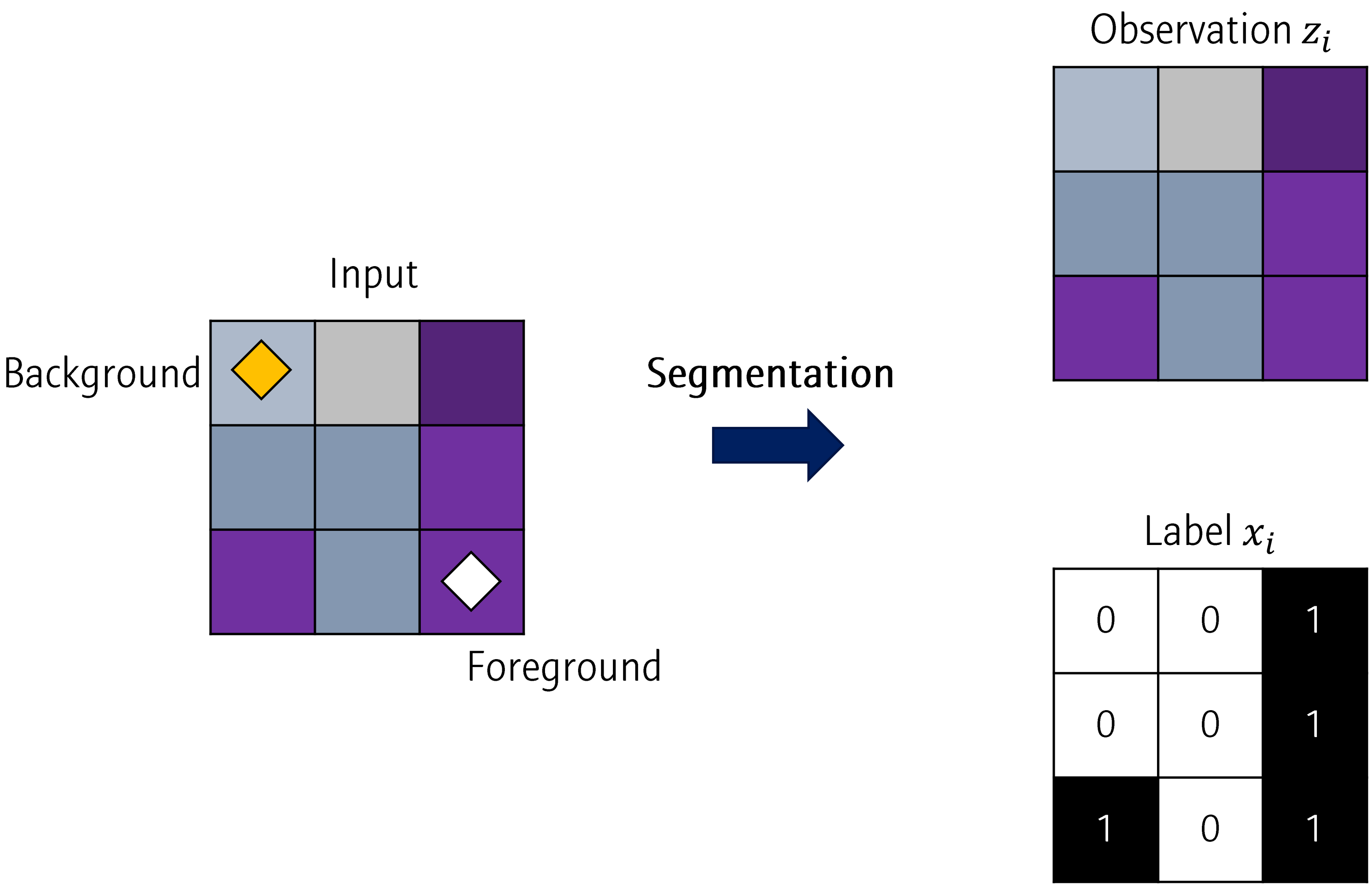
4. Region growing / Watershed algorithm

Watershed algorithm: 2D



5. Graph model

Mechanism of Graph model

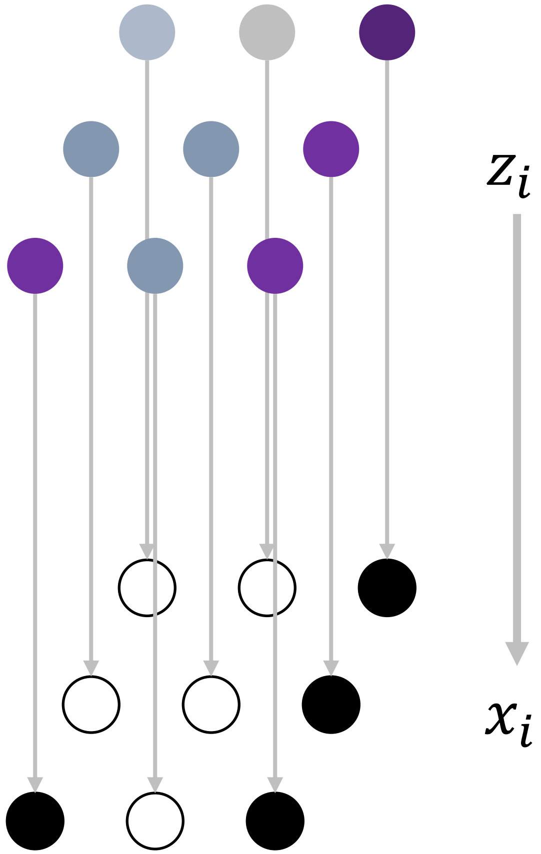


5. Graph model

Mechanism of Graph model

$$maximize\ P(x_1, x_2, \ldots, x_N | z_1, z_2, \ldots, z_N)$$

Posterior probability



5. Graph model

Mechanism of Graph model

$$maximize\ P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N)$$

Bayes rule

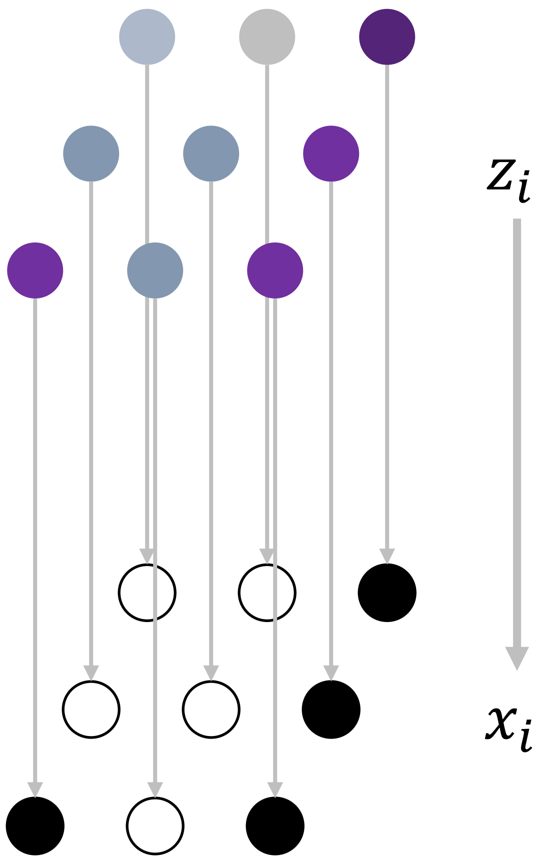
Posterior probability

$$\underbrace{P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N)}_{\text{Posterior probability}} = \frac{\overbrace{P(z_1, \dots, z_N | x_1, \dots x_N)}^{\text{Likelihood}}\overbrace{P(x_1, \dots, x_N)}^{\text{Prior probability}}}{P(z_1, \dots, z_N)}$$

Assumption 1: Naïve Assumption,
각 샘플들은 독립적이다 (=다른 픽셀 값을 고려하지 않는다)

$$P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N) \propto \prod_{i}^{N} P(z_i | x_i) \prod_{i,j} P(x_i, x_j)$$

Prior probability: 관찰된 값에 대한 분포
Likelihood: 관찰된 값을 바탕으로 예측된 Class의 확률



5. Graph model

Mechanism of Graph model

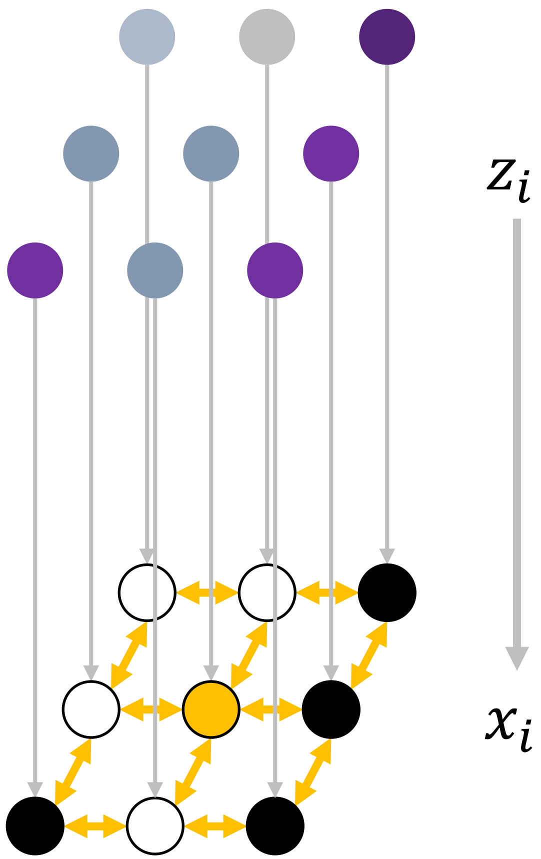
$$maximize\ P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N)$$

Bayes rule

Posterior probability *Likelihood* *Prior probability*

$$P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N) = \frac{P(z_1, \dots, z_N | x_1, \dots x_N) P(x_1, \dots, x_N)}{P(z_1, \dots, z_N)}$$

Assumption 2: Markov Random Field
주변부 픽셀 레이블들의 관계를 고려한다



5. Graph model

Mechanism of Graph model

Maximization problem

$$P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N) \propto \prod_i^N P(z_i|x_i) \prod_{ij} P(x_i, x_j)$$

Gibbs energy: Take negative logarithm

Minimization problem

$$E(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N) = \log\left(\prod_i^N P(z_i|x_i) \prod_{ij} P(x_i, x_j)\right)$$

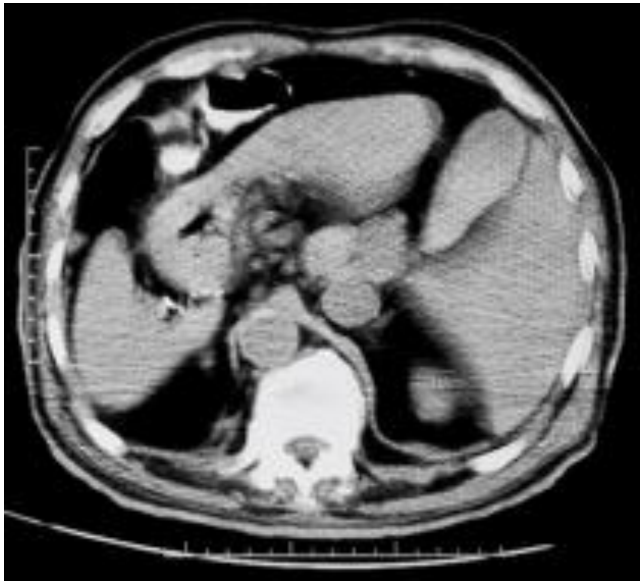
$$= \sum_i^N \theta_i(z_i|x_i) + \sum_{i,j} \theta_{i,j}(x_i, x_j)$$

Likelihood term　　　Prior term

Goal: x_i 조합 (2^N가지) Likelihood term과 Prior term을 모두 낮춰줄 수 있는 조건을 찾는 것

5. Graph model

λ



Input

$$E(x,z,\lambda) = \sum_{i}^{N} \theta_i(z_i|x_i) + \lambda \sum_{i,j} \theta_{i,j}(x_i, x_j)$$

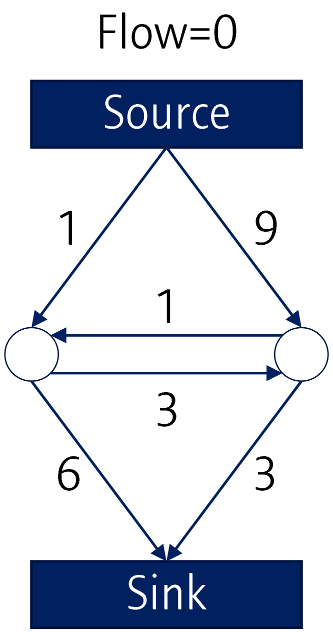
Likelihood

Prior

λ를 높이면, prior term을 고려함 (주변부를 고려하게 됨)

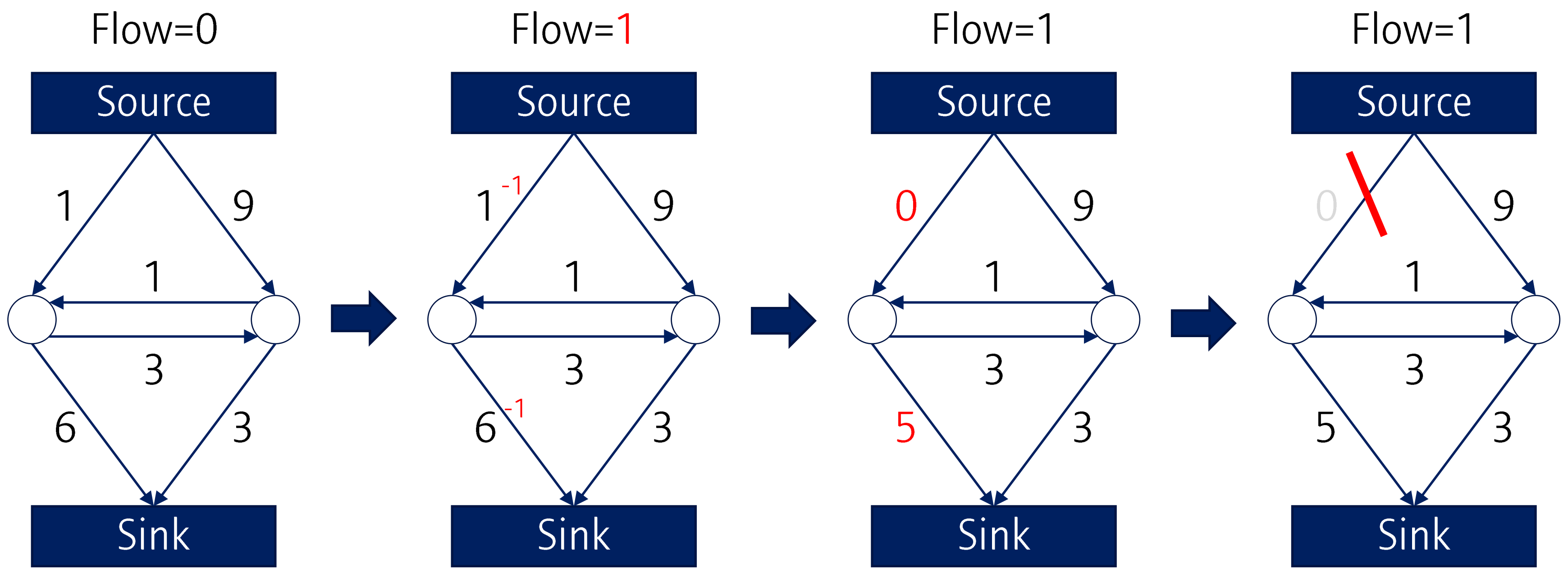
6. Graph cut optimization

Max-flow Min-cut: example data



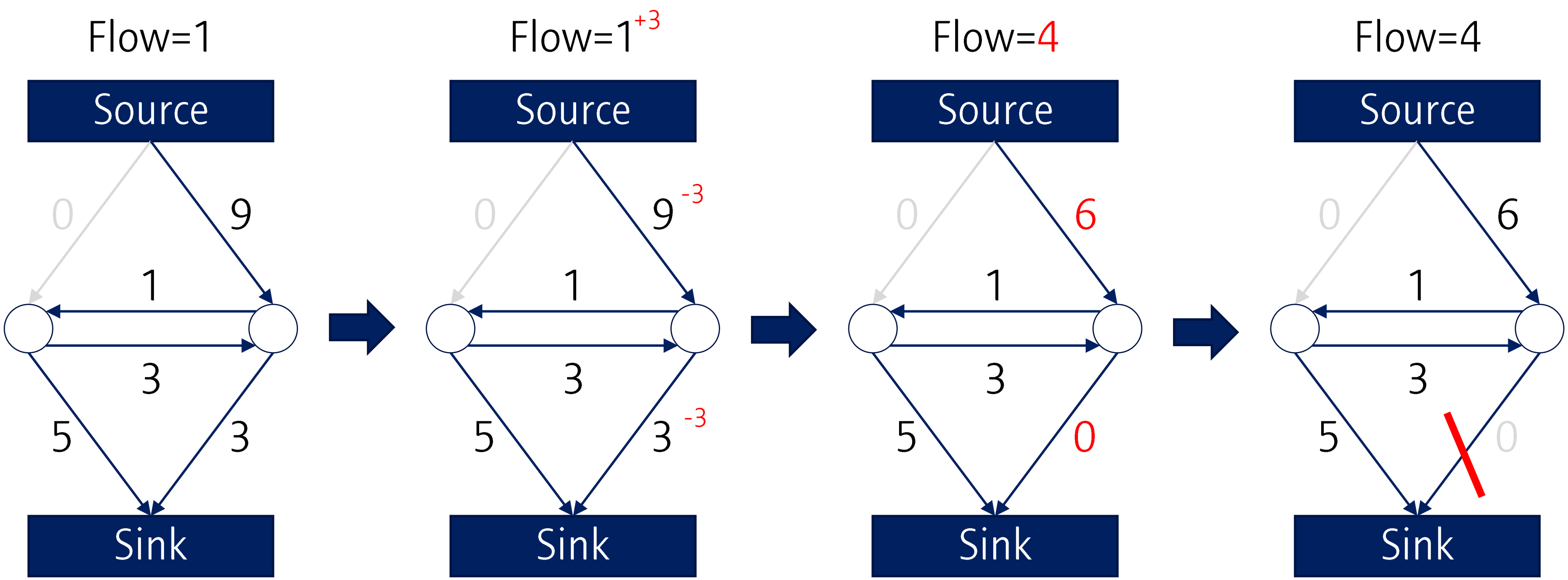
6. Graph cut optimization

Max-flow Min-cut: procedure 1/3



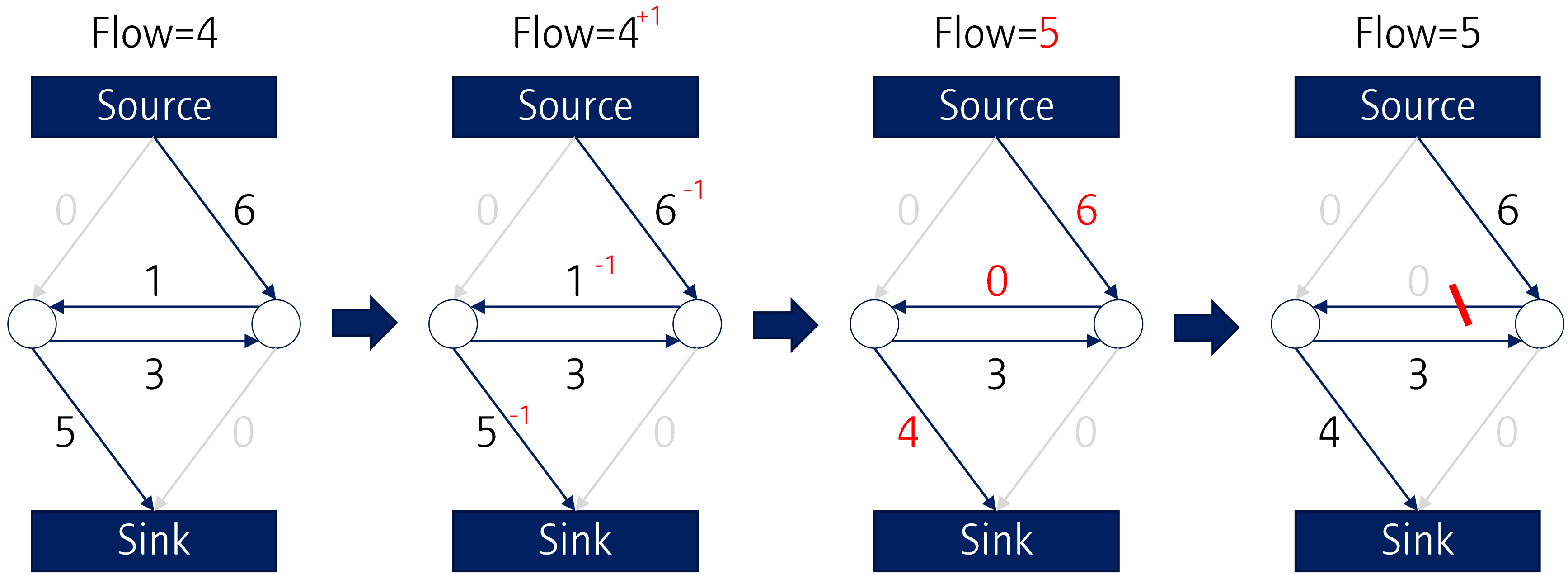
6. Graph cut optimization

Max-flow Min-cut: procedure 2/3



6. Graph cut optimization

Max-flow Min-cut: procedure 3/3



6. Graph cut optimization

Max-flow Min-cut: result