Point Operation

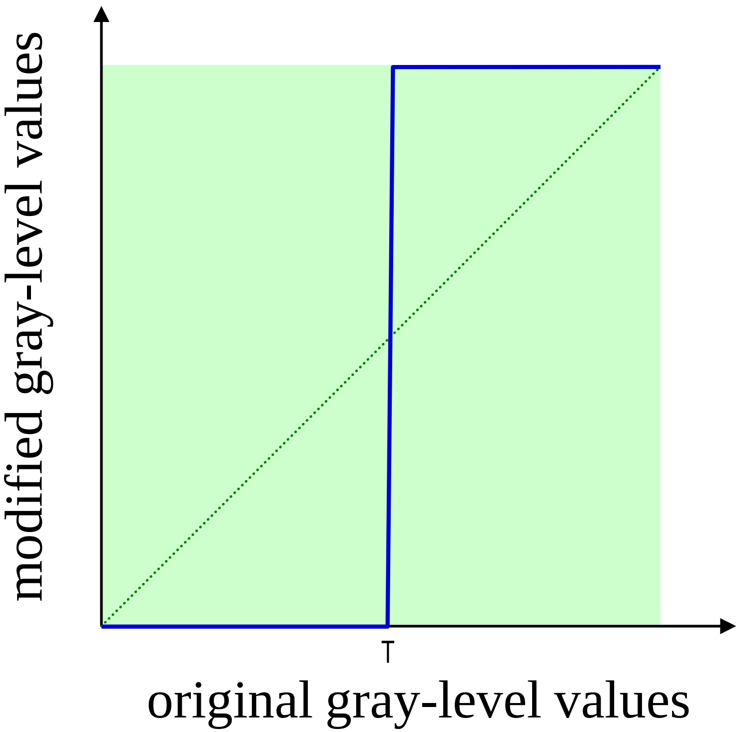
Grayscale Thresholding

김성영교수
금오공과대학교
컴퓨터공학과

학습 내용

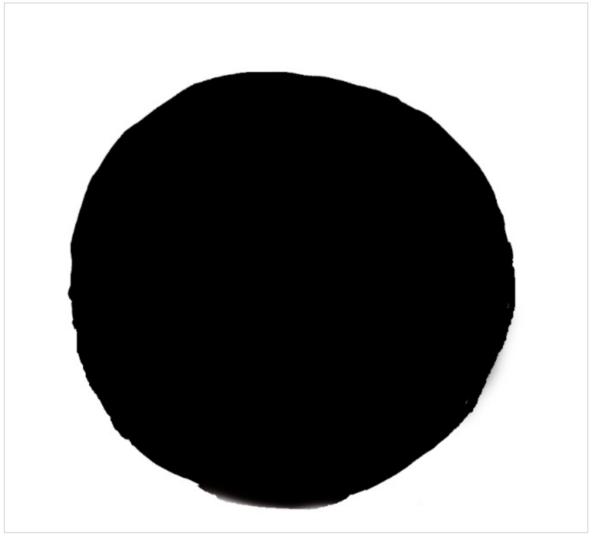
- Determining threshold in Thresholding
- Processing for Color Images

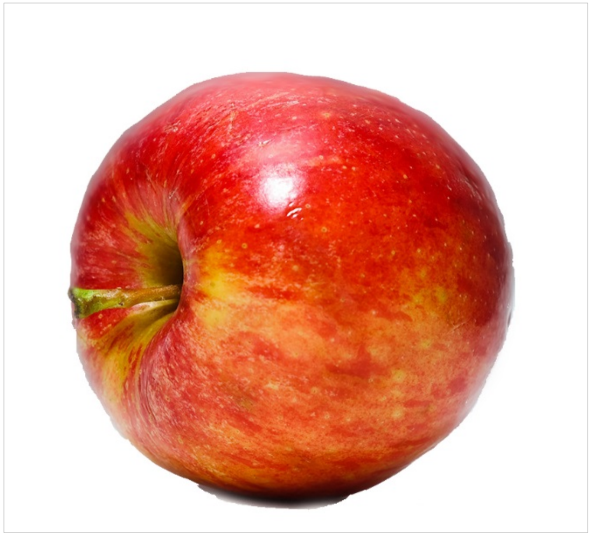
Gray-level Thresholding

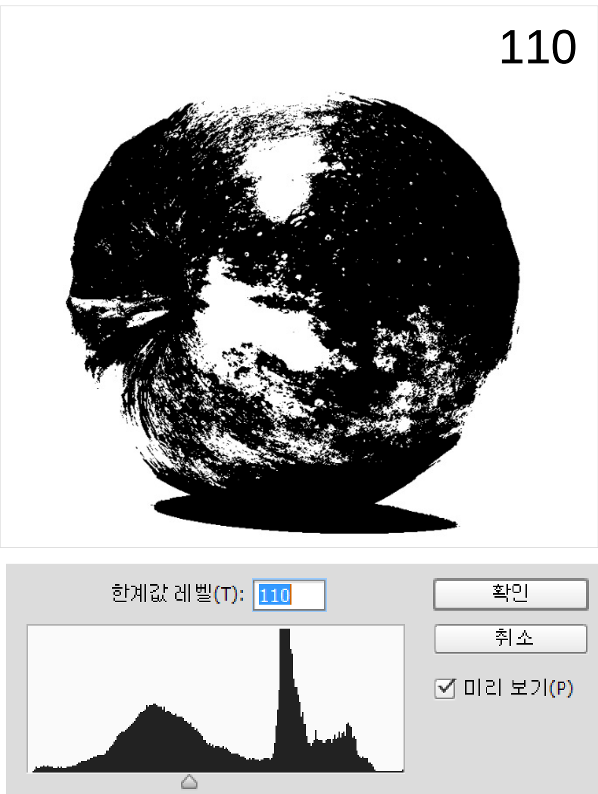


Binarization

이진화



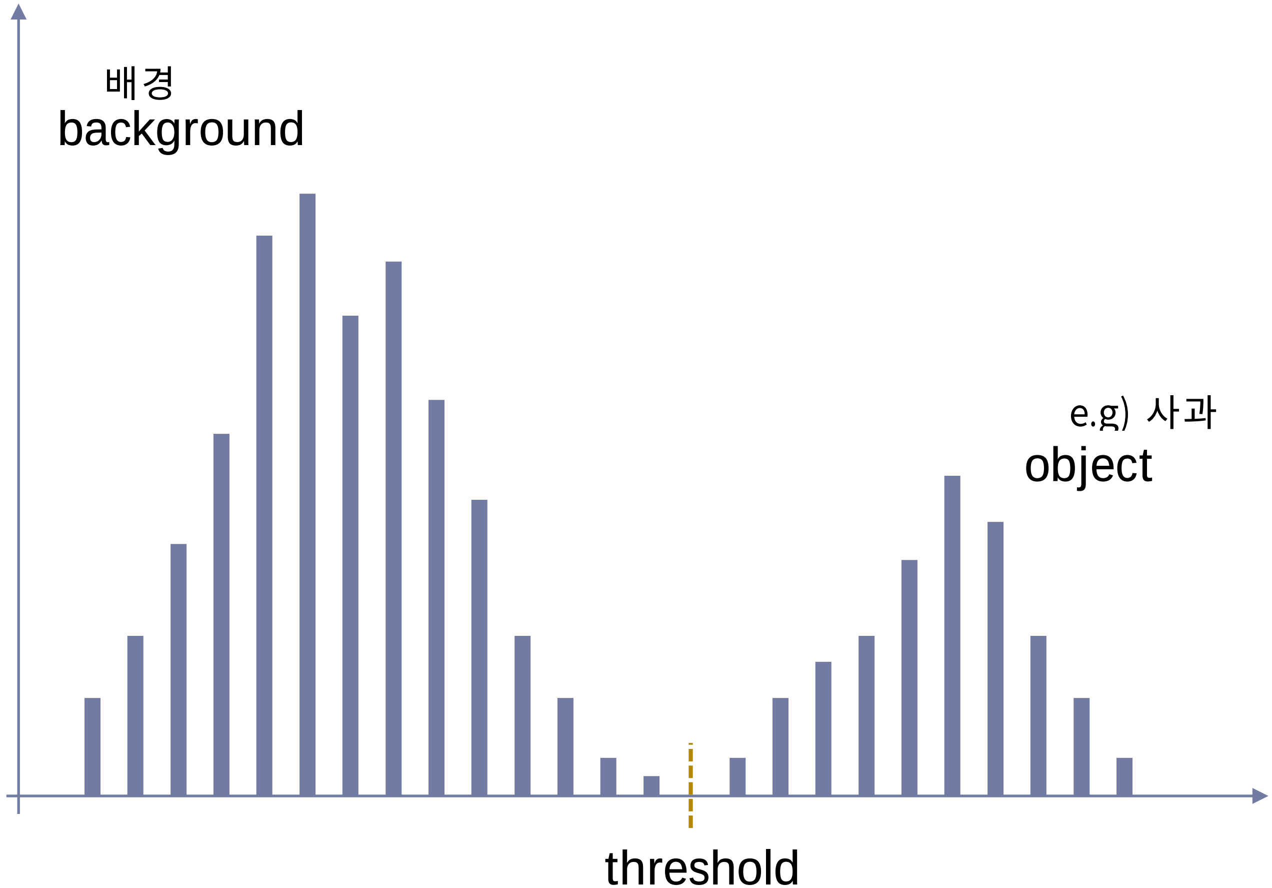



110
한계값 레벨(T): 110
확인
취소
미리 보기(P)

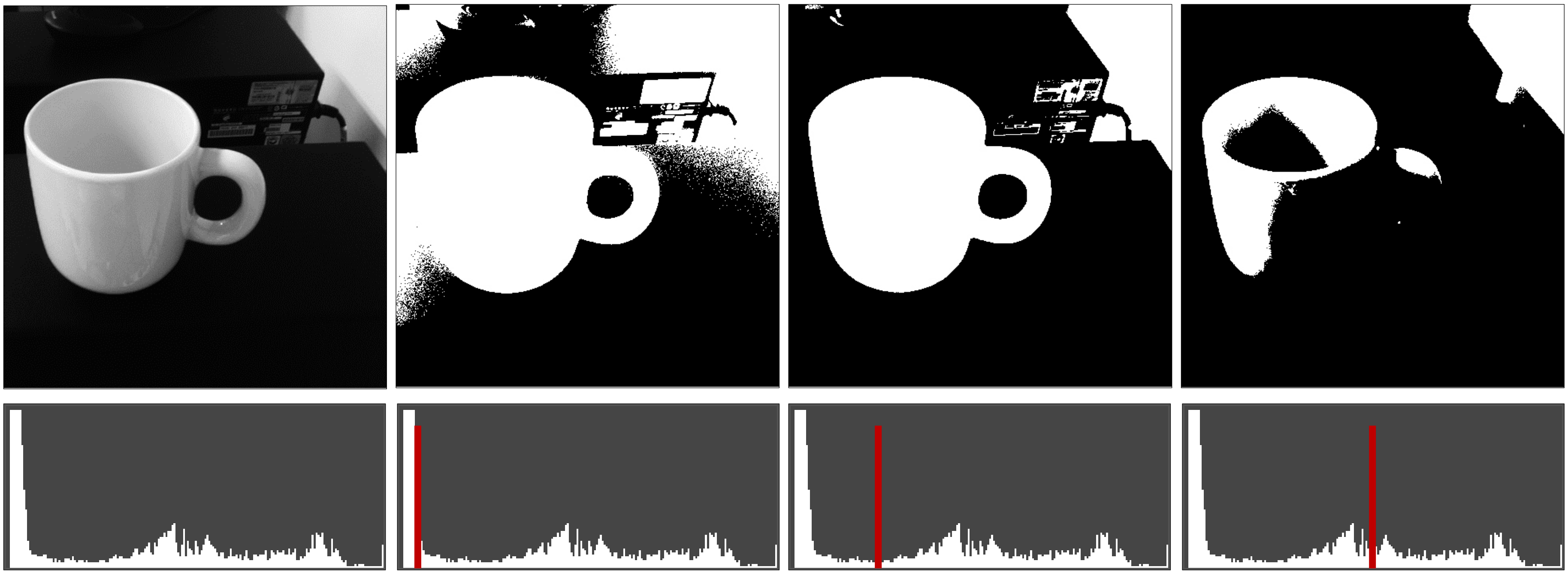

160
한계값 레벨(T): 160
확인
취소
미리 보기(P)


175
한계값 레벨(T): 175
확인
취소
미리 보기(P)

THRESHOLD의 결정



Bimodal histogram



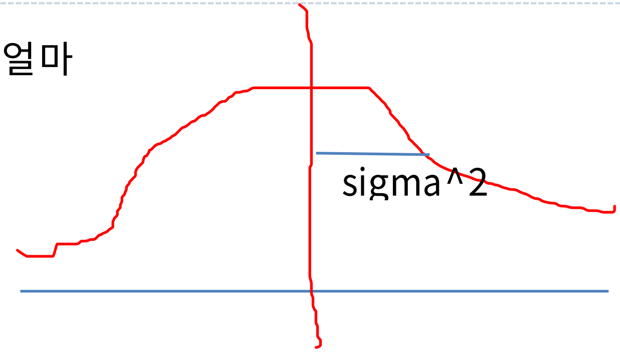
8 26 130

Optimal Threshold by Otsu (1)

- 기본 원리
  - 임계값 T를 기준으로 영역을 2개 그룹으로 나누었을 때 각 집합내의 명암 분포는 균일하고 집합 사이의 명암 차이는 최대화될 수 있도록 함
  - 모든 가능한 T에 대해 점수를 계산하여 가장 좋은 T를 최종 임계값으로 선택함 ⇨ 최적화 알고리즘 (optimization algorithm)
    - 낱낱 탐색 (exhaustive search), 언덕 오르기 (hill climbing) 등의 탐색 방법을 사용 가능
  - 최적화 알고리즘에서는 비용 함수 (cost function) 또는 목적 함수 (objective function)을 사용하여 점수 계산

Optimal Threshold by Otsu (2)

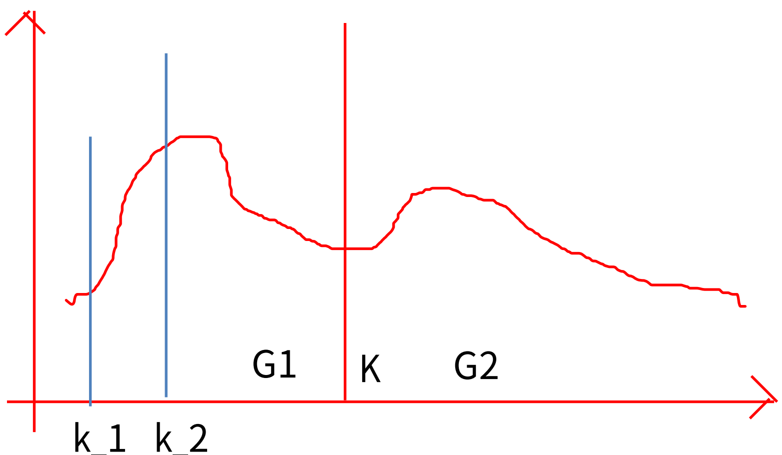
1. 분산 sigma^2 데이터가 평균으로부터 얼마
나 떨어져 있는 평균적인 거리
2 max - min = Range



목적 함수

$$T_{opt} = \mathrm{argmin}_{1 \le k \le L-1}(\sigma_W^2(k))$$

$$\sigma_W^2(k) = \omega_1(k)\sigma_1^2(k) + \omega_2(k)\sigma_2^2(k)$$



Optimal Threshold by Otsu (3)

-> 분산 분해 (수리 통계학)
Analysis Variance

1) 히스토그램 계산

2) $T = k(k \geq 1)$에서 클래스 분리를 위한 확률 및 평균 계산

$$C_1(k) = \sum_{i=0}^{k-1} N_i\,, \qquad C_2(k) = \sum_{i=k}^{L-1} P_i = N - C_1(k)$$

$$\boldsymbol{\omega_1(k) = \frac{C_1(k)}{N}}, \qquad \boldsymbol{\omega_2(k) = \frac{C_2(k)}{N} = 1 - \omega_1(k)}$$

$$\mu_{T1}(k) = \sum_{i=0}^{k-1} i \cdot N_i\,, \quad \mu_{T2}(k) = \sum_{i=k}^{L-1} i \cdot N_i\,, \quad \mu_T = \sum_{i=0}^{L-1} i \cdot N_i$$

$$\boldsymbol{\mu_1(k) = \frac{\mu_{T1}(k)}{C_1(k)}}, \qquad \boldsymbol{\mu_2(k) = \frac{\mu_{T2}(k)}{C_2(k)} = \frac{\mu_T - \mu_{T1}(k)}{N - C_1(k)}}$$

OPTIMAL THRESHOLD BY OTSU (4)

3) $T = k(k \geq 1)$ 에서 클래스 분리를 위한 분산 σ_W^2 계산

$$\sigma_1^2(k) = \sum_{\underset{i}{\cancel{n}}=0}^{k-1}[\overset{i}{\cancel{n}} - \mu_1(k)]^2 \frac{N_{\cancel{n}\,i}}{C_1(k)}, \quad \sigma_2^2(k) = \sum_{\underset{i}{\cancel{n}}=k}^{L-1}[\cancel{n} - \mu_2(k)]^2 \frac{N_{n\,i}}{C_2(k)}$$

$$\boldsymbol{\sigma_W^2(k) = \omega_1(k)\sigma_1^2(k) + \omega_2(k)\sigma_2^2(k)}$$

4) 모든 레벨에 대해 반복하여 최적 임계값 선택

$$\boldsymbol{T_{opt} = \mathrm{argmin}_{1 \leq k \leq L-1}\left(\sigma_W^2(k)\right)}$$

Optimal Threshold by Otsu (5)

$$\boldsymbol{\sigma^2 = \sigma_B^2 + \sigma_W^2}$$

Within class variance: $\boldsymbol{\sigma_W^2 = \omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}$

Between class variance: $\boldsymbol{\sigma_B^2 = \sigma^2 - \sigma_W^2}$

$$\boldsymbol{= \omega_1 (\mu_1 - \mu_T)^2 + \omega_2 (\mu_2 - \mu_T)^2}$$

$$\boldsymbol{= \omega_1 \omega_2 (\mu_1 - \mu_2)^2}$$

$$\boldsymbol{= \omega_1 (1 - \omega_1)(\mu_1 - \mu_2)^2}$$

Optimal Threshold by Otsu (6)

k = 0일 때

$$\omega_1(0) = N_0, \qquad \mu_1(0) = 0$$

$k \geq 1$일 때

$$\omega_1(k) = \omega_1(k-1) + N_k$$

$$\mu_1(k) = \frac{\omega_1(k-1)\mu_1(k-1) + kN_k}{\omega_1(k)}$$

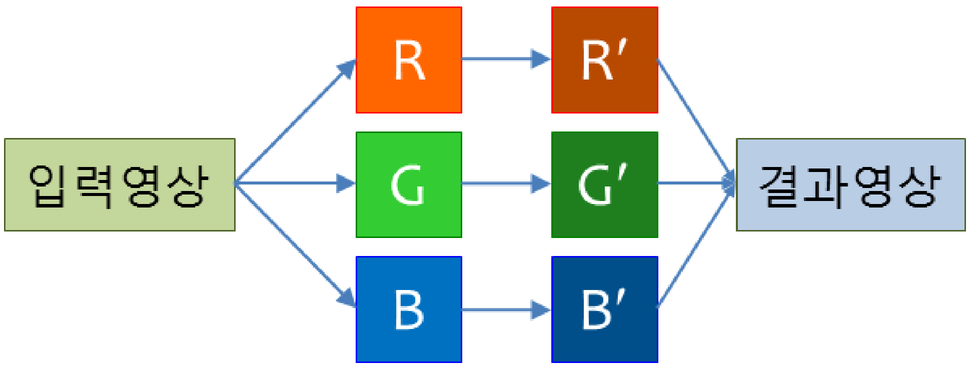
$$\mu_2(k) = \frac{\mu - \omega_1(k)\mu_1(k)}{1 - \omega_1(k)}$$

ALGORITHM

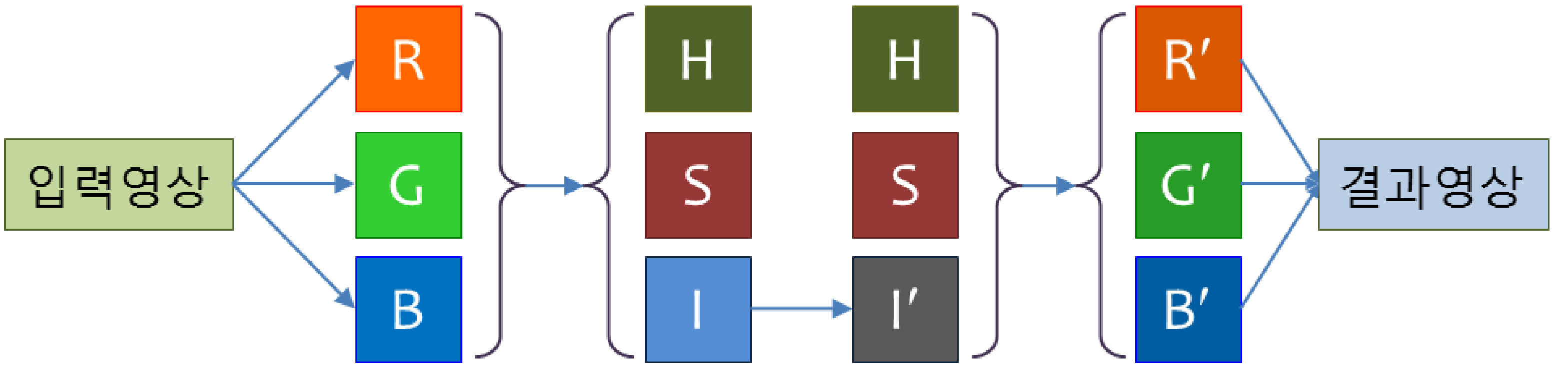
1. 히스토그램 계산
2. $\omega_1(0)$과 $\mu_1(0)$ 계산
3. 각 threshold $k(1 \leq k < L)$에 대해
   3-1. $\omega_0(k)$, $\mu_1(k)$, $\mu_2(k)$ 계산
   3-2. $\sigma_B^2(k)$ 계산
   3-3. 최대 σ_B^2와 비교하여 현재 $\sigma_B^2(k)$가 더 크면
      ① 현재 σ_B^2를 사용하여 최대 σ_B^2를 갱신
      ② threshold k를 optimal threshold (T_OPT)로 선택

※ Coarse to fine approach 사용 가능

PROCESSING FOR COLOR IMAGES



RGB 처리



HSI 처리

요약

- point operations
  - 이웃 픽셀과는 독립적으로 입력 영상의 각 픽셀 값을 변환한 후 결과 영상의 동일한 위치에 출력하는 연산
  - Improving image contrast and brightness
- Arithmetic operation
  - Scalar operation 및 Image operation
- Grayscale transformation
  - Improving image contrast and brightness by using mapping function
  - Brightness scaling by multiplication, Gray-level Thresholding, Gray-level Negative 등

Reference

- 오일석, **Computer Vision**, 한빛 아카데미, 2014
- Scott E Umbaugh, **Computer Imaging**, CRC, 2005
- Mark Nixon and Alberto Aguado, **Feature Extraction & Image Processing**, ELSEVIER, 2008
- Frank SHIH, **Image Processing and Pattern Recognition**, IEEE Press, 2010