Notes for Image Processing

Coordinate digitization $\mapsto$ sampling

Color digitization $\mapsto$ quantization

Spatial resolution is the smallest distinguishable detail.

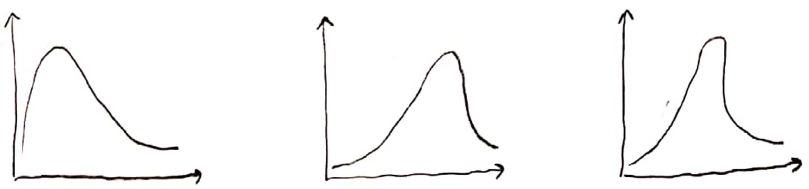
image negative $\mapsto$ $s = L - r - 1$

intensity level: $[0, L-1]$

brightness enhancement $\mapsto$ $J_k(r,c) = \begin{cases} I_k(r,c)+g, & \text{if } I_k(r,c)+g < 256 \\ 255 & \text{if } I_k(r,c)+g > 255 \end{cases}$

$g \geq 0$ and $k \in \{1, 2, 3\}$ is the band index.

Histogram



↳ dark ↳ light ↳ high contrast

RGB Image Histogram

- 3 histograms for both R, G and B.
- 3 bits R, 3 bits G, 2 bits B (MSB) $\mapsto$ 0-255 pseudo colors.
- Only intensity $(I = (R+G+B)/3)$

Histogram Equalization

- Transformation function

$$s = T(r) \quad 0 \le r \le 1$$

Conditions:

a) $T(r)$ is single valued and monotonically increasing.

b) $0 \le T(r) \le 1$ for $0 \le r \le 1$

- inverse function

$$r = T^{-1}(s) \quad 0 \le s \le 1$$

Example

r_k	n_k	$P_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

. intensity distribution and histogram values for a 3-bit, 64x64 image.

$$s_k = T(r_k) = (L-1) \cdot \sum_{j=0}^{k} P_r(r_j)$$

Discrete CDF

input	output	D. output
0	$\rightarrow$ 1.33	$\rightarrow$ 1
1	$\rightarrow$ 3.08	$\rightarrow$ 3
2	$\rightarrow$ 4.55	$\rightarrow$ 5
3	$\rightarrow$ 5.67	$\rightarrow$ 6
4	$\rightarrow$ 6.23	$\rightarrow$ 6
5	$\rightarrow$ 6.65	$\rightarrow$ 7
6	$\rightarrow$ 6.86	$\rightarrow$ 7
7	$\rightarrow$ 7.0	$\rightarrow$ 7

$$s_0 = T(r_0) = (8-1) \cdot \sum_{j=0}^{0} P_r(r_j) = 7 \cdot (0.19) = 1.33$$

$$s_1 = T(r_1) = (8-1) \cdot \sum_{j=0}^{1} P_r(r_j) = 7 \cdot (0.19 + 0.25) = 3.08$$

$s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, $s_7 = 7.0$

Image Enhancement

point processing vs mask/kernel processing

Image ⟼ filter ⟼ min/max normalization
[0, 255]

Smoothing (Low Pass) Spatial Filters

- Smoothing filters are used for blurring and for noise reduction.
- Blurring is used in preprocessing steps, such as removal of small details from an image prior to object extraction, and bridging of small gaps in lines or curves

There are 2 ways of smoothing spatial filters
↳ smoothing linear filters
↳ Order-statistics filters

Averaging Filter

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Standard average   weighted average

Gaussian Smoothing Filter

$$G_\sigma = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

1D Gaussian Function:

$$* \quad \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

2D Gaussian Function:

$$* \quad \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Example

3x3 Kernel, $\sigma = 5.5$

mostly 5x5 or 7x7

$$A = \begin{bmatrix} x=-1, y=-1 & x=0, y=-1 & x=1, y=-1 \\ x=-1, y=0 & \frac{1}{2\pi(5.5)^2} \exp\left(-\frac{0^2+0^2}{2\cdot(5.5)^2}\right) & x=1, y=0 \\ x=-1, y=1 & x=0, y=1 & x=1, y=1 \end{bmatrix}$$

- Each item should be multiplied by $1/\min(A)$
- Then, divide by $\mathrm{sum}(A)$.

- If $\sigma\uparrow$ then blurring $\uparrow$

Nonlinear Filters

- Based on ordering the pixels contained in the kernel.
- Replacing the value of the center pixel with the value determined by the ranking result.
- e.g., median filter, max filter, min filter.

Edge Detection

Edge normal $\mapsto$ unit vector in the direction of maximum intensity change

Edge direction $\mapsto$ unit vector to perpendicular to the edge normal.

Edge center $\mapsto$ the image position at which the edge is located

Edge strength $\mapsto$ related to the local image contrast along the normal.

Image Derivative and Sharpening

$$f'(x) = \frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$f''(x) = \frac{\partial^2 f}{\partial x^2} = \frac{\partial f}{\partial x} \cdot (f'(x) - f'(x-1)) = f(x+1) + f(x-1) - 2f(x)$$

Gradient of the Image

$$\nabla f = \left[\frac{\partial f}{\partial x} \; \frac{\partial f}{\partial y} \right]^T$$

$\longrightarrow$ $\nabla f = \left[\frac{\partial f}{\partial x} \; 0 \right]$

$\downarrow$ $\nabla f = \left[0 \; \frac{\partial f}{\partial y} \right]$

Magnitude of the Gradient

$$|\nabla f| = \left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right)^{1/2}$$

Direction of the Gradient

$$\angle(\nabla f) = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

The Prewitt Edge Detector

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

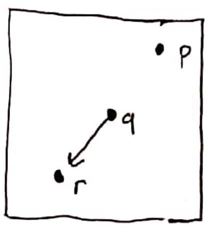
The Sobel Edge Detector

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Canny Edge Detector

- Filter image with derivative of Gaussian.
- Find magnitude and orientation of Gradient.
- Non-maximum supression
  → Thin multi-pixel wide "ridges" down to single pixel width.
- Linking of edge points
  → Hysteresis thresholding: use a higher threshold to start edge curves and a lower threshold to continue them.

Check if gradient magnitude at (i,j) is local maximum along gradient direction.



if q is larger than p and r, keep q.

Strong Edge Pixels	Low and High
Weak Edge Pixels	Thresholds

if $I(x,y) <$ Low threshold → not edge

if $I(x,y) >$ high threshold → edge

if low $\leq I(x,y) \leq$ high → Classified as edge iff they are connected to a sure edge otherwise discarded.

$\|\nabla f(x,y)\| \geq t_h \mapsto$ definitely

$t_\ell \geq \|\nabla f(x,y)\| < t_h \mapsto$ maybe

$\|\nabla f(x,y)\| < t_\ell \mapsto$ no

Image Segmentation

Meaningful and inconsistent regions of the image.

Examples ⟼ Segmentation by gray intensity ⟼ edge ⟼ discontinuity
Segmetation by texture features
Segmatation by depth

region, threshold ⟼ similarity

Segmentation by Thresholding

Global Thresholding

if $f(x,y) > T$, then $f(x,y) = 0$ else $f(x,y) = 255$

1- initialize T
2- divide as G1 and G2 with T
3- get average color values for both G1 and G2
4- get new T by
$$T = \frac{1}{2}(\mu_1 + \mu_2)$$
5- Repeat until $|T_{t-1} - T_t| < T0$

→ histogram should be bi-modal

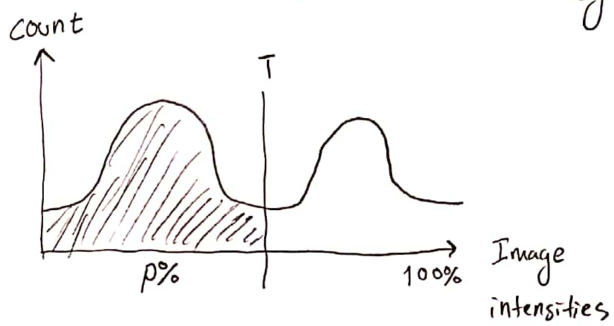
Scanned with CamScanner

Otsu with Global Thresholding

- variance of within class must be minimized
- Variance of between class must be maximized
- Grayscale histogram must be bi-modal

p-tile Thresholding

if the object occupies a known %p in the image, the threshold level is determined by the histogram.



Region Based Segmentation

- Based on sets
- Each image R is a set of regions R_i.
  - → Every pixel belongs to one region.
  - → One pixel can only belong to a single region.

$$R = \bigcup_{i=1}^{s} R_i \, , \quad R_i \cap R_j = \emptyset$$

Bottom-up Approach

entire image

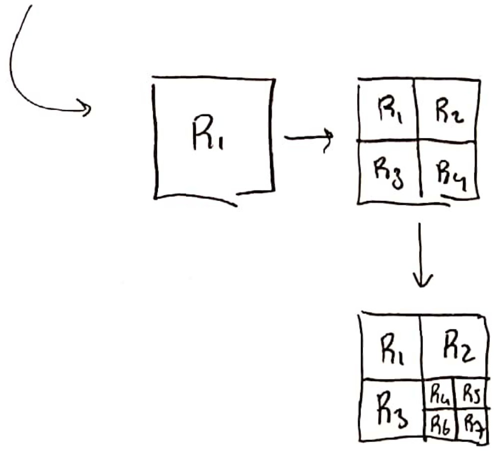
big regions

small regions

pixels

Top-down Approach

→ split and merge



K-Means Clustering

→ Reduce the number of colors (quantization), then segmentation.

Colors

$B+G=C$

$R+G=M$ → CMY Renk uzayı

$R+G=Y$

$C = 1-(B+G)$

$M = 1-(B+R)$

$Y = 1-(G+R)$

HSI

Hue → Renk tonu

Saturation → saflık miktarı

Brightness → Işığın yoğunluğu

Value → 3 rengin en büyüğünün değeri (HSV)

YUV & YIQ → renk ile parlaklık bilgisini ayırmak → renklerin daha az bilgi ile gösterilmesi avantaj

bilgi kaybı dezavantaj

↳ YCbCr
YPbPr
YDbDr

1 grayscale → Y

3 color → Cr Cg Cb

Chroma

4:4:4 → no compression

4:2:2 → enine yarı, boyuna tam

4:2:0 → sadece ilk çeyrek

↓

örnek sayısı

Morfolojik işlemler

$A - B = A \cap B'$

- Şekilsel değişiklik yapan filtreler.
- Binary görüntülerde kullanılır.
- Structured element
  - ↳ small set to probe the image under study
  - ↳ for each SE, define origo
  - ↳ Shape and size must be adapted to geometric properties for the objects.

- Erosion : shrink
- Dilation : grow

Erosion

$$A \ominus B = \{ z : (B)_z \cap A^c \neq \emptyset \}$$

A ↓ image

B ↓ structured element

			1		1		
		1	1	1	1	1	
			1		1		

A

$\ominus$

0	1	0
1	1	1
0	1	0

Eğer B, seçilen piksele tam oturuyorsa o piksel kalır oturmuyorsa silinir.

			0		0		
		0	1	0	1	0	
			0		0		

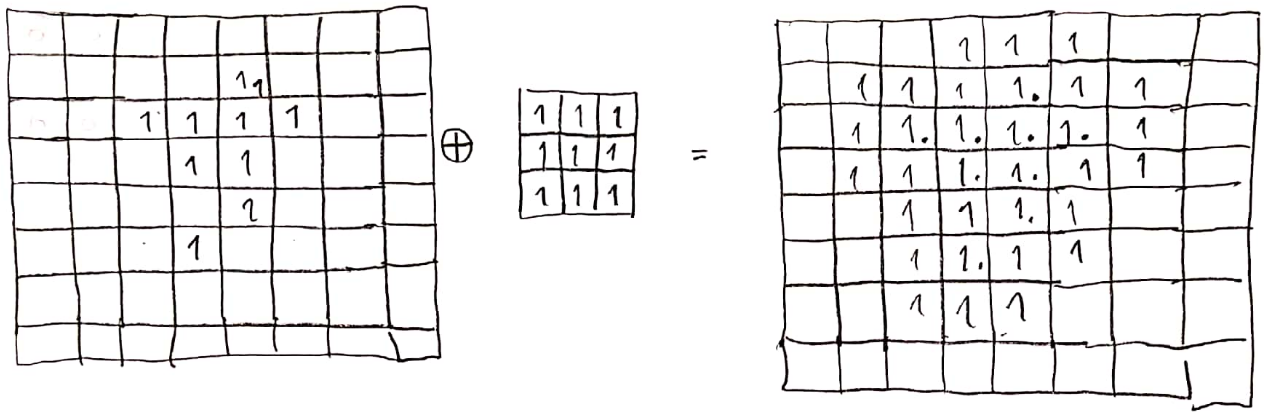
1 $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \ominus \begin{bmatrix} & 1 & \\ 1 & 1 & 1 \\ & 1 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Dilation

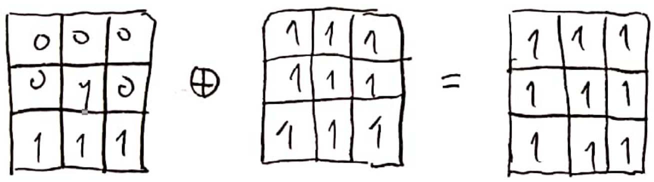
$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}$

A → image, B → SE

B'nin A ile kesişimi boş küme değilse koruyoruz.



1



Opening

Erosion → Dilation

$A \circ B = (A \ominus B) \oplus B$

Closing

Dilation → Erosion

$A \bullet B = (A \oplus B) \ominus B$

Hit or Miss Operation

Şekil tanıma için kullanılabilir.

0'lar da 1'ler de tam oturursa pikseli bırak. Olmuyorsa sıfırla.

Boundary Extraction

$$\beta(A) = A - (A \ominus B)$$

Region Filling

$$X_K = (X_{K-1} \oplus B) \cap A^c \quad K=1,2,3,\dots$$

Face Recognition

1) Preprocessing
2) Segmentation (face detection)
3) Feature extraction

difficulties:
- lightning
- Occlusion
- Viewpoint

Eigenvalue - Eigenvector

$$Av = \lambda v$$

$$Av - \lambda v = 0 \rightarrow (A - \lambda I)v = 0$$

$$\det(A - \lambda I) = 0 \rightarrow \begin{bmatrix} a_{11}-\lambda & a_{12} \\ a_{21} & a_{22}-\lambda \end{bmatrix} = 0$$

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \lambda \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = 0$$

$$\det\left(\begin{bmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix}\right) = 0 \longrightarrow (1-\lambda)^2 - 4 = 0 \longrightarrow \begin{matrix} \lambda_1 = -1 \\ \lambda_2 = 3 \end{matrix}$$

$$\lambda = -1 \longrightarrow \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = 0 \longrightarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix} \checkmark$$

$$\lambda = 3 \longrightarrow \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = 0 \rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix} \checkmark$$

Covariance Matrix

$$Cov(x,y) = \frac{\sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y)}{N-1}$$

$$C = \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix}$$

- reshape $A[N \times N]$ to $[N \times 1]$ N vectors

$$A[N \times N] \rightarrow \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N^2} \end{bmatrix} \cdots \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N^2} \end{bmatrix}$$

- Mean face : $\begin{bmatrix} a_1 + b_1 + \cdots + h_1 \\ \vdots \\ a_{N^2} + b_{N^2} + \cdots + h_{N^2} \end{bmatrix} * \frac{1}{M} = m$

- Mean-centered image : $\vec{a}_M = \begin{bmatrix} a_1 - m_1 \\ a_2 - m_2 \\ \vdots \\ a_{N^2} - m_{N^2} \end{bmatrix}$

- Covariance matrix:

$$A = \begin{bmatrix} a_1 & b_1 & \cdots & h_1 \\ a_2 & b_2 & \cdots & h_2 \\ \vdots & & & \\ a_{N^2} & b_{N^2} & \cdots & h_{N^2} \end{bmatrix} \qquad COV = A A^T$$

- Compute eigenvalues and eigenvectors. Select positive K eigenvalue in ascending orders

- $a_{Eigen} = V^T \vec{a}$

  $b_{eigen} = V^T \vec{b}$

- For unknown face X
  - ↳ mean center $\vec{X}$
  - ↳ $X_{eigen} = V^T \vec{X}$
  - ↳ $dist(\cdots, X_{eigen})$

Measuring devices → preprocessing → dimensionality reduction → prediction → model selection

öğrenme yöntemi seçimi → veri toplama → varsayımlar → özellik seçimi → model seçimi → değerlendirme

---

Basic Data Redundancies

- interpixel redundancy → Lossless
- psychovisual redundancy → Lossy
- Coding redundancy → Loseless

Compression Ratio: $C_r = \dfrac{n_1}{n_2}$

n_1: n of original bits

n_2: n of compressed bit

$I(x,y) \rightarrow \boxed{\text{encoder}} \xrightarrow{\text{compressed image}} \boxed{\text{decoder}} \rightarrow I'(x,y)$

- Lossy
- Lossless

Relative Data Redundancy: $R_D = 1 - \dfrac{1}{C_R}$

$R_D \approx 0 \rightarrow$ no redundancy

interpixel redundancy

The pixel values are not independent but correlated with their neighbors, both within the same frame and across frames.

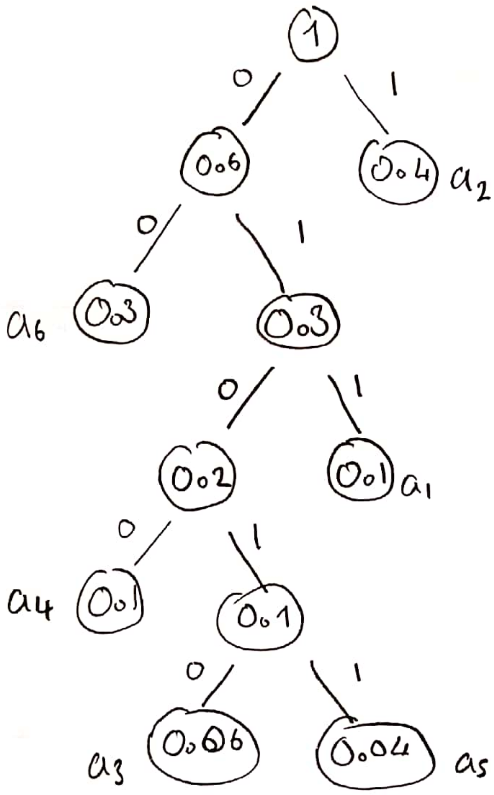
Run-length coding

Coding redundancy

For any non-random digitized signal, some code values occur more frequently than others.

Code the more frequently occuring values with shorter codes than the rare ones (Huffman Coding).

a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				



$a_2 \to 0.4 \to 1$

$a_6 \to 0.3 \to 00$

$a_1 \to 0.1 \to 011$

$a_4 \to 0.1 \to 0100$

$a_3 \to 0.06 \to 01010$

$a_5 \to 0.04 \to 01011$

$0.4\times1+0.3\times2+0.1\times3$
$+0.1\times4+0.06\times5$
$+0.04\times5=2.7$

Kullanılan ortalama bit sayısı

$$C_R=\frac{3}{2.7}=1.1$$

$$R_D=1-\frac{1}{2.7}=0.099$$

$$a_2a_4a_6 \to \underline{1}\ \underline{0100}\ \underline{00}$$

Psychovisual Redundancy

The limit of Spatial resolution is the ability of the eye to resolve the fine details in the image.

The limit of temporal resolution is the ability of the eye to track fast-moving images.

Human Sensitivity

Low frequencies → error in homogeneous regions

High frequencies → error in edge

Medium frequencies → textured areas

JPEG

RGB → YCbCr → DCT Transform → quantization → huffman