

ECE 172A: Introduction to Image Processing

Image Processing Tasks

Rahul Parhi
Assistant Professor, ECE, UCSD

Winter 2025

Outline

- Preprocessing
 - Histogram
 - Normalization
 - Combining Images
 - Spatial Averaging
- Matching and Detection
 - Correlation
 - Matched Filtering
- Feature Extraction
 - Contour/Edge Detection
- Segmentation
 - Variational Thresholding
 - Connected-Component Labeling

Preprocessing

- Histogram
- Normalization
- Combining Images
- Spatial Averaging
 - Linear Smoothing
 - Median Filtering

Graylevel Histogram

Input image: $r[k]$ with $k \in \Omega = \{0, \dots, K-1\} \times \{0, \dots, L-1\}$

Total number of pixels: $\#\Omega = KL$

- Graylevel distribution

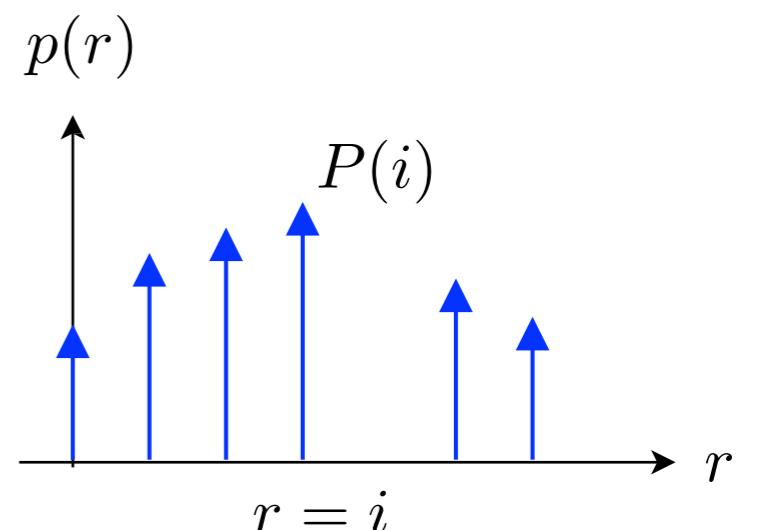
p.d.f. $p(r)$ with $\int_{-\infty}^{\infty} p(r) dr = 1$

- Histogram

Quantized graylevels: $\{0, 1, 2, \dots, N-1\}$

n_i : number of pixels with graylevel i

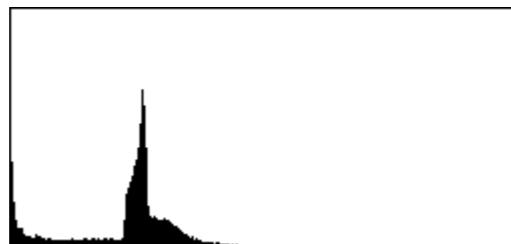
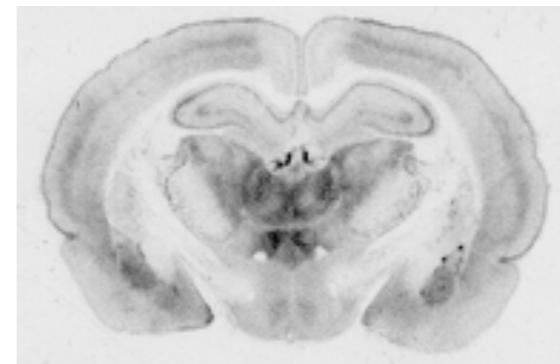
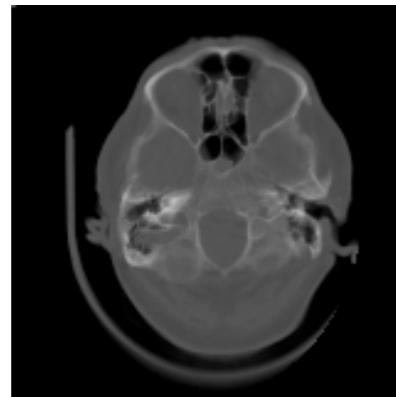
$P(i) = \frac{n_i}{\#\Omega}$: probability of graylevel i



$$p(r) = \sum_{i=0}^{N-1} P(i) \delta(r - i)$$

- Probability mass function (p.m.f.)

Examples of Histograms



What can we do with these histograms?

- Reading the histogram can tell us about:
 - Dynamic range
 - Potential saturation problems
 - Average intensities of background and objects

Normalization: Linear Contrast Adjustment

Linear transformation/system: $T\{f\}[\mathbf{k}] = \alpha(f[\mathbf{k}] - \beta)$ with parameters $\alpha, \beta \in \mathbb{R}$

How to we implement full dynamic-range contrast stretching?

$$\beta = \min\{f[\mathbf{k}] : \mathbf{k} \in \Omega \subset \mathbb{Z}^2\} \quad \alpha = \frac{255}{\max_{\mathbf{k}}\{f[\mathbf{k}]\} - \min_{\mathbf{k}}\{f[\mathbf{k}]\}}$$

- Image normalization

Average graylevel

$$\mu = \frac{1}{\#\Omega} \sum_{\mathbf{k} \in \Omega} f[\mathbf{k}]$$

Variance

$$\sigma^2 = \frac{1}{\#\Omega} \sum_{\mathbf{k} \in \Omega} (f[\mathbf{k}] - \mu)^2$$

Normalized image statistics: $T\{f\}[\mathbf{k}] = a \left(\frac{f[\mathbf{k}] - \mu}{\sigma} \right) + b$

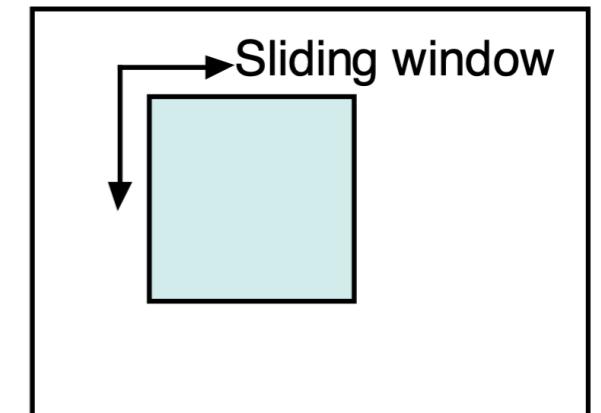
“zero mean and unit variance”

Localized Normalization

Compensation of non-uniformities across the image;
e.g., shading, nonuniform background, changes in illumination

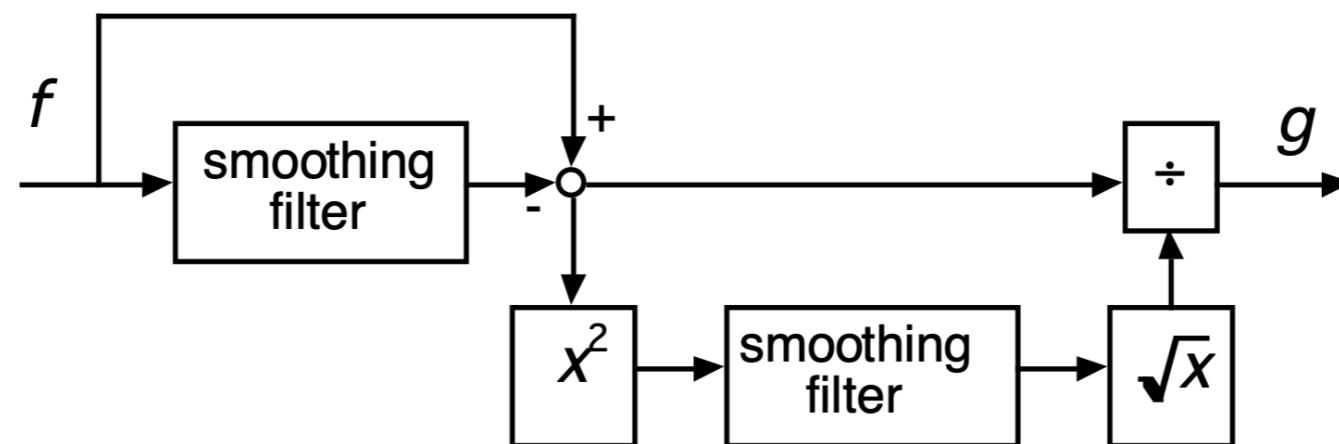
- Normalization over a sliding window:

$$g[\mathbf{k}] = a \left(\frac{f[\mathbf{k}] - \tilde{\mu}[\mathbf{k}]}{\tilde{\sigma}[\mathbf{k}]} \right) + b$$



$$\tilde{\mu}[\mathbf{k}] = \sum_{\mathbf{n}} w[\mathbf{n}] f[\mathbf{n} - \mathbf{k}]$$

$$\sum_{\mathbf{k}} w[\mathbf{k}] = 1$$



<https://bigwww.epfl.ch/demo/ip/demos/local-normalization/>

Smoothing filter implements local averaging \Rightarrow Estimation of local statistics

Combining Images

- Averaging for noise reduction:
 - Independent noisy observations: $f_i[\mathbf{k}] = s[\mathbf{k}] + n_i[\mathbf{k}], \quad i = 1, \dots, N$
 - Hypotheses:
 - (i) $\mathbf{E}[f_i[\mathbf{k}]] = s[\mathbf{k}] \Rightarrow \mathbf{E}[n_i[\mathbf{k}]] = 0$
 - (ii) i.i.d. noise at each location $\mathbf{k} \Rightarrow \text{var}(f_i[\mathbf{k}]) = \text{var}(n_i[\mathbf{k}]) = \sigma^2$
 - Noise reduction procedure: $\bar{f}[\mathbf{k}] = \frac{1}{N} \sum_{i=1}^N f_i[\mathbf{k}]$

Exercise: Determine the mean and variance of $\bar{f}[\mathbf{k}]$

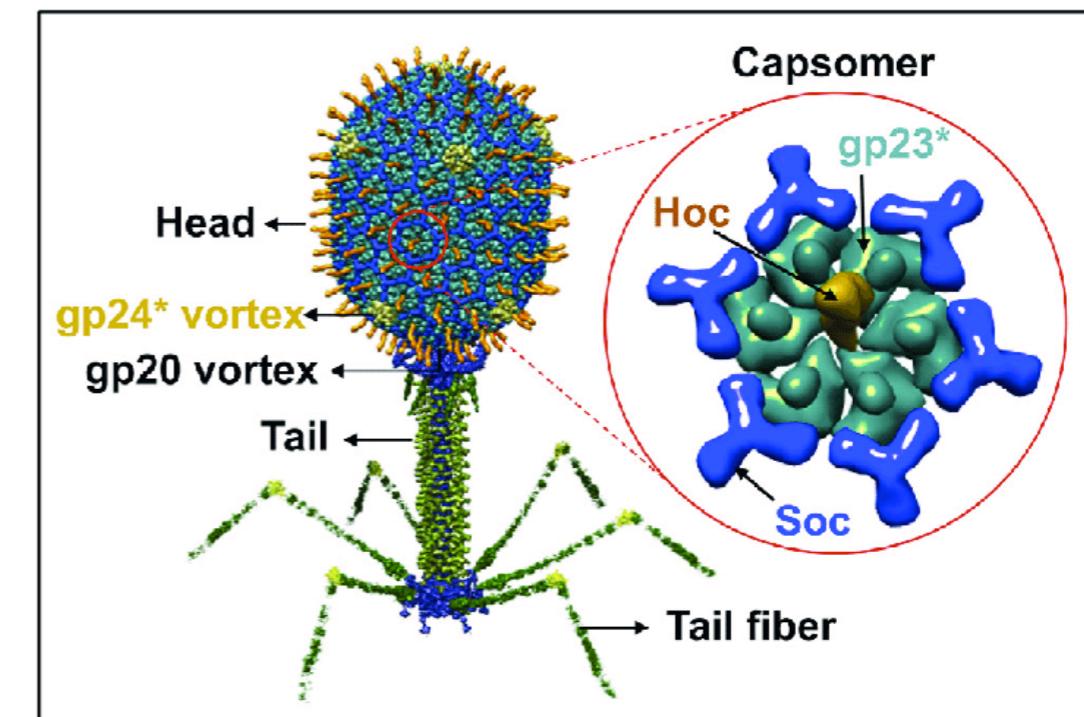
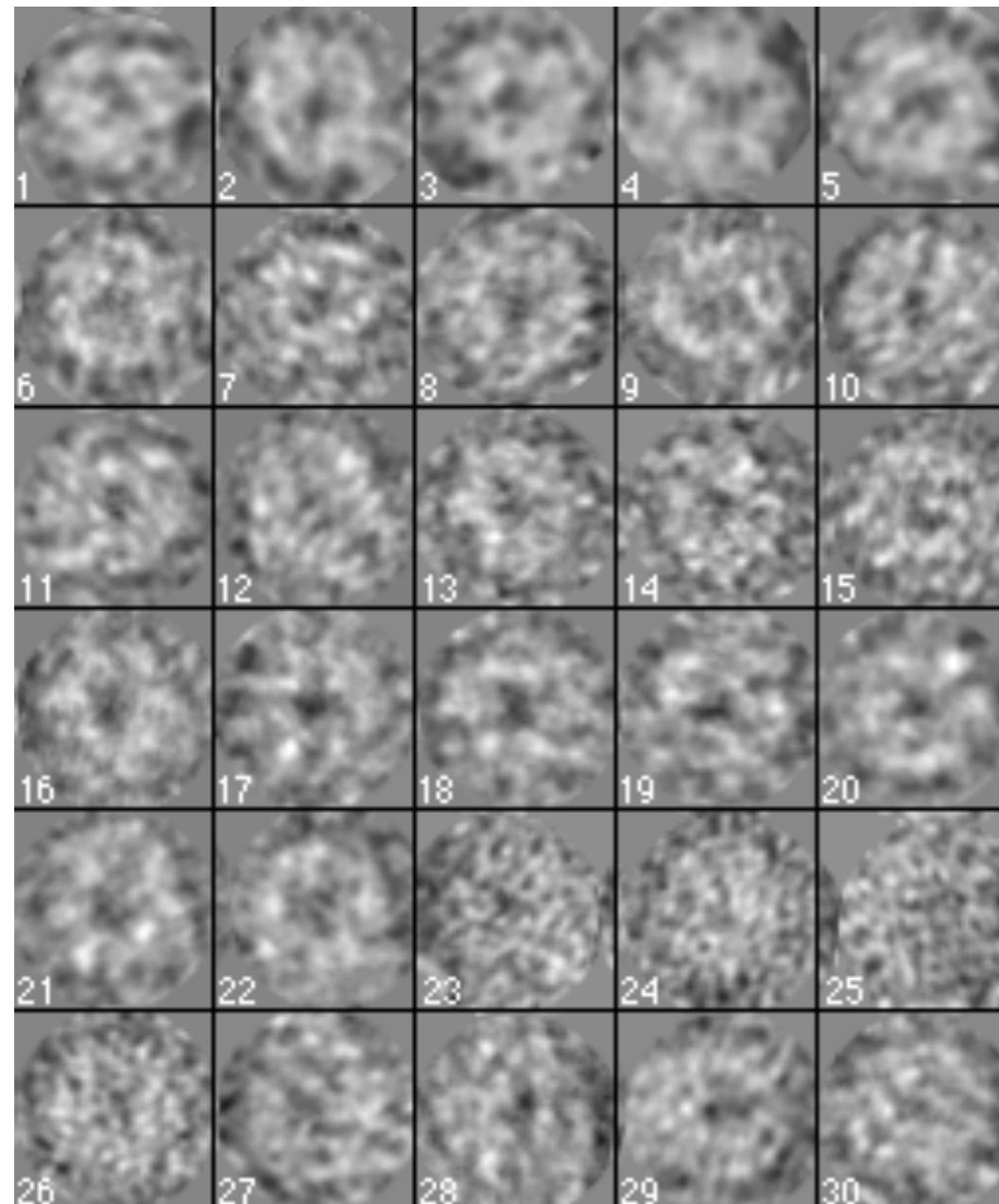
$$\text{Mean: } \mathbf{E}[\bar{f}[\mathbf{k}]] = s[\mathbf{k}]$$

$$\text{Variance: } \text{var}(\bar{f}[\mathbf{k}]) = \sigma^2/N$$

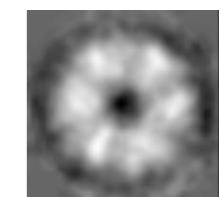
Central limit theorem: for large N , $\bar{f}[\mathbf{k}] \sim \mathcal{N}(s[\mathbf{k}], \sigma^2/N)$

Example: Noise Reduction

20 electron micrographs of a virus capsomere



Result of averaging:



Practical problems

- Image registration
- Detection of outliers

Spatial Averaging: Linear Smoothing

Linear smoothers = Low-pass filters

$$g = h * f \text{ with } \sum_k h[k] = 1$$

- Finite-impulse response (FIR)

Moving average

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & \boxed{1/9} & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1/8 & 0 \\ 1/8 & \boxed{1/2} & 1/8 \\ 0 & 1/8 & 0 \end{bmatrix}$$

- Limitations
 - Blurring of edges and image details

- Infinite-impulse response (IIR)

- Symmetric exponential
 - Gaussian filter

How do we get around this?

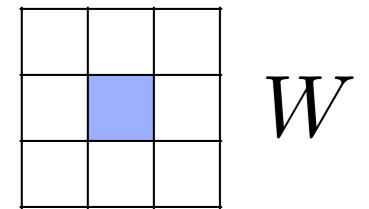
- Main uses

- noise reductions (high frequencies)
 - estimation of local statistics (mean, variance)

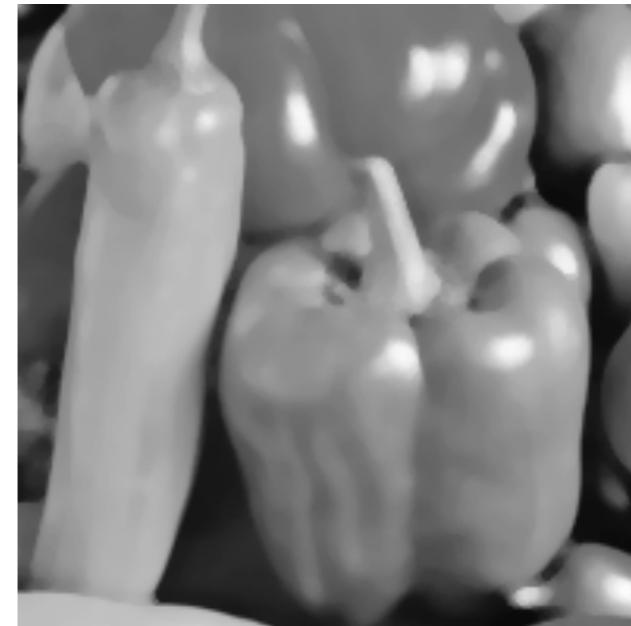
Nonlinear operations

Spatial Averaging: Median Filter

$$g[k] = \text{median}\{f[k-n] : n \in W\}$$



Input (200×200)



5×5 median filtered

- Advantages
 - Tend to preserve contours better than linear smoothers
 - Good for impulsive or heavy-tailed (non-Gaussian) noise
- Limitations
 - Computationally costly for large sizes of neighborhoods
 - Breaks down when there is a majority of noisy pixels

Impulsive-Noise Reduction Experiment



Input (200×200)



Input + impulsive noise



3×3 median



3×3 moving average



5×5 moving average



5×5 median

Matching and Detection

- Template Matching
 - Problem Definition
 - Correlation
- Matched-Filter Detection
- Application Areas
 - Object Detection
 - Automated Inspection
 - Data Fusion
 - Registration
 - Motion Compensation

Template Matching

- Problem definition
 - Reference pattern, target, or template: $f_r[\mathbf{k}], \mathbf{k} \in \Omega_r$
 - Test image: $f[\mathbf{k}], \mathbf{k} \in \Omega_f$
 - Common support $\Omega = \Omega_f \cap \Omega_r \neq \emptyset$
 - How do we decide whether or not f and f_r are similar?
 - Given a collection of templates $f_i, i = 1, \dots, N$ (e.g., shifted versions of some reference template), how do we select the best match?

Exercise: Come up with a concrete instantiation of this sort of problem

Correlation Measures

- Basic correlation

$$\sum_{\mathbf{k} \in \Omega} f[\mathbf{k}] f_r[\mathbf{k}] = \langle f, f_r \rangle$$

$\ell^2(\Omega)$ -inner product

How is maximizing the correlation related to the similarity between f and f_r ?

$$\text{Similarity} = \text{distance} = \|f - f_r\|_{\ell^2(\Omega)}$$

$$\|f - f_r\|_{\ell^2(\Omega)}^2 = \langle f - f_r, f - f_r \rangle$$

$$= \|f\|_{\ell^2(\Omega)}^2 + \|f_r\|_{\ell^2(\Omega)}^2 - 2\langle f, f_r \rangle$$

$$= \text{constant} - 2\langle f, f_r \rangle$$

increasing correlation decreases distance

$\|f - f_r\|_{\ell^2(\Omega)}^2$ is minimum $\Leftrightarrow \langle f, f_r \rangle$ is maximum

Correlation Measures (cont'd)

What if our template and test image have different intensity ranges?

- Centered correlation

$$\langle f - \bar{f}, f_r - \bar{f}_r \rangle = \sum_{\mathbf{k} \in \Omega} (f[\mathbf{k}] - \bar{f})(f_r[\mathbf{k}] - \bar{f}_r)$$

average value

$$\bar{g} = \frac{1}{\#\Omega} \sum_{\mathbf{k} \in \Omega} g[\mathbf{k}]$$

- Normalized correlation coefficient

$$-1 \leq \rho(f, f_r) = \frac{\langle f - \bar{f}, f_r - \bar{f}_r \rangle}{\|f - \bar{f}\|_{\ell^2(\Omega)} \|f_r - \bar{f}_r\|_{\ell^2(\Omega)}} \leq 1$$

Invariant to linear amplitude scalings: $af + b$

Matched-Filter Detection

- Measurement model (signal + noise): $f[\mathbf{k}] = s[\mathbf{k} - \mathbf{k}_0] + n[\mathbf{k}]$
 - s : known deterministic template or pattern
 - n : additive **white** noise with zero mean and variance σ^2
 - \mathbf{k}_0 : unknown template location $\mathbf{E}[f[\mathbf{k}]] = s[\mathbf{k} - \mathbf{k}_0]$
- Goal: Design a correlation-like detector

$$g[\mathbf{k}] = (h * f)[\mathbf{k}]$$

$$= \sum_{\mathbf{n} \in \mathbb{Z}^2} h[\mathbf{n}] f[\mathbf{k} - \mathbf{n}] = \sum_{\mathbf{n} \in \mathbb{Z}^2} w[\mathbf{n}] f[\mathbf{k} + \mathbf{n}]$$

“convolution” “correlation”

where $w[\mathbf{k}] = h[-\mathbf{k}]$

Matched-Filter Detection

- Optimal detector: Maximizes SNR at $\mathbf{k} = \mathbf{k}_0$

Solution: $w[\mathbf{k}] = s[\mathbf{k}]$ (matched filter)

(technically, $w[\mathbf{k}] = \alpha s[\mathbf{k}]$ is fine, for any $\alpha \in \mathbb{R}$)

Proof:

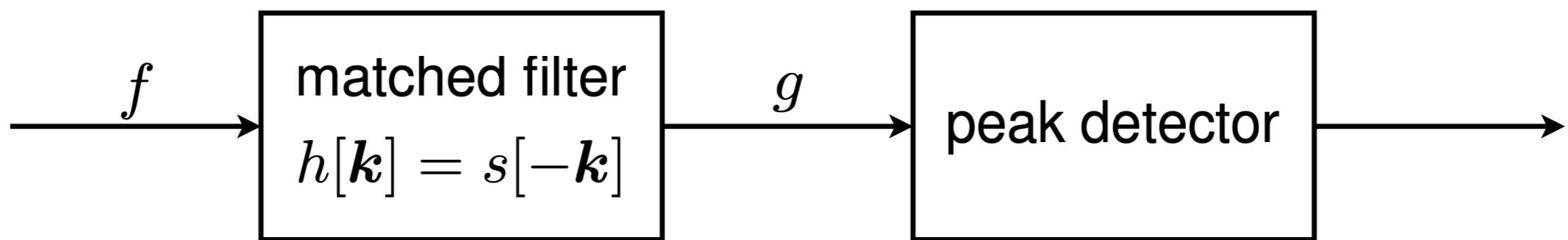
“signal” Expected output at $\mathbf{k} = \mathbf{k}_0$ $\mathbf{E}[g[\mathbf{k}_0]] = \sum_{\mathbf{n} \in \mathbb{Z}^2} w[\mathbf{n}] s[\mathbf{k}_0 - \mathbf{k}_0 + \mathbf{n}]$
 $= \langle w, s \rangle$

“noise” Variance output $\text{var}(g[\mathbf{k}]) = \sum_{\mathbf{n} \in \mathbb{Z}^2} w[\mathbf{n}]^2 \text{var}(n[\mathbf{k} + \mathbf{n}]) = \sigma^2 \|w\|_{\ell^2(\mathbb{Z}^2)}^2$

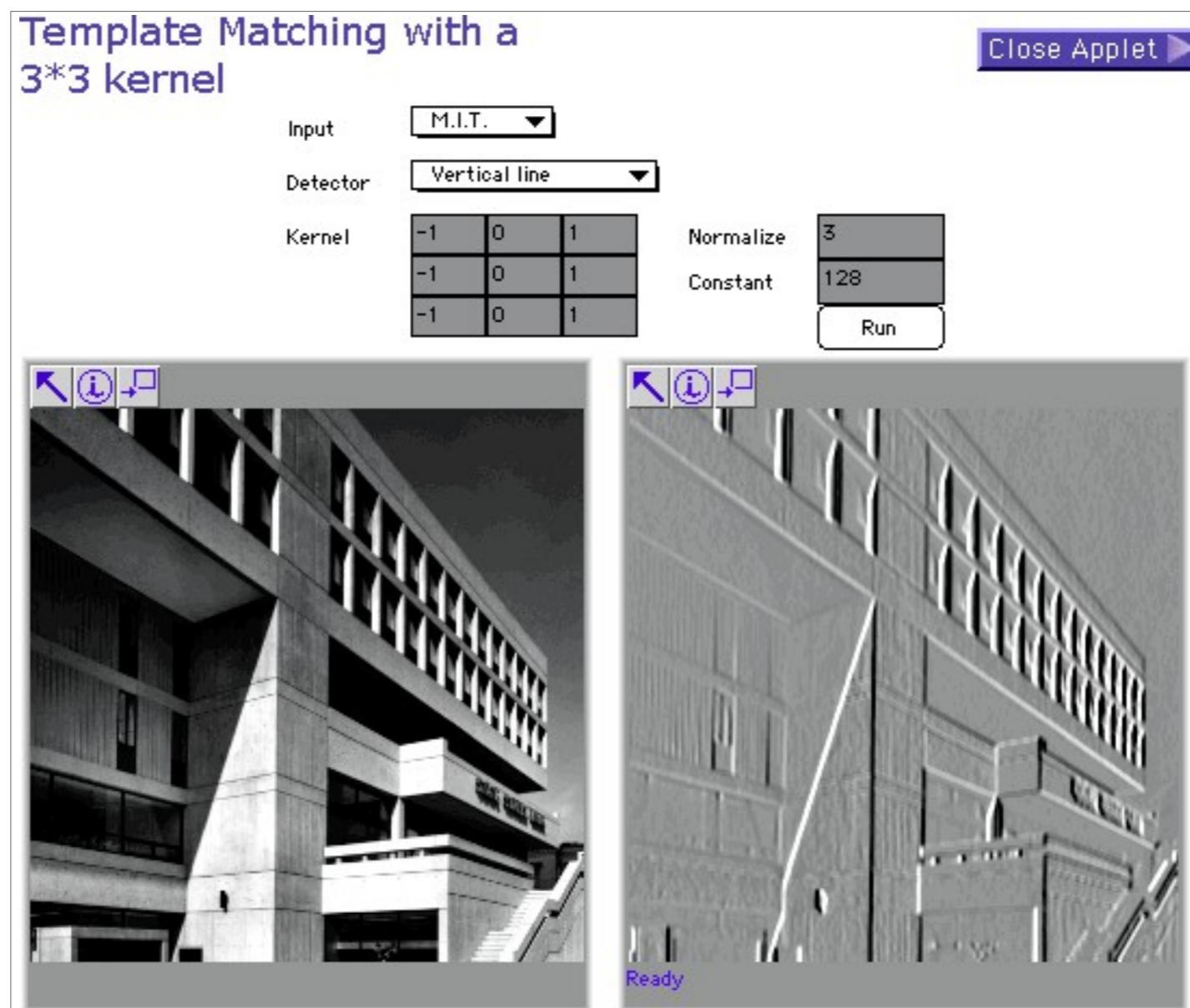
SNR at $\mathbf{k} = \mathbf{k}_0$: $\text{SNR} = \frac{\langle w, s \rangle}{\sigma \|w\|_{\ell^2(\mathbb{Z}^2)}}$

Maximized when $w[\mathbf{k}] = \alpha s[\mathbf{k}]$

Pattern Detection by Template Matching



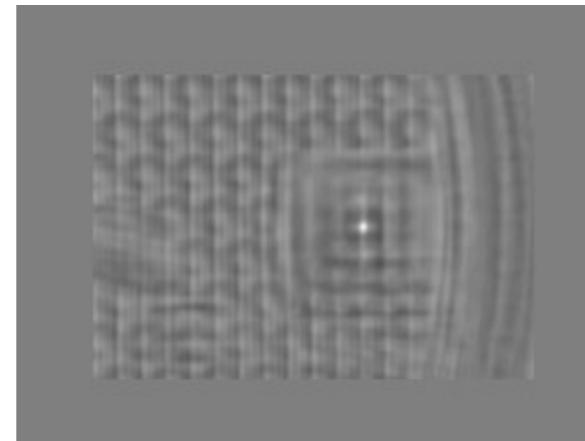
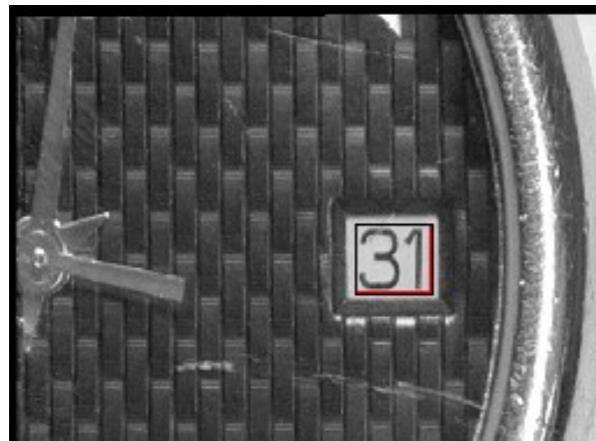
- Application: Line detector



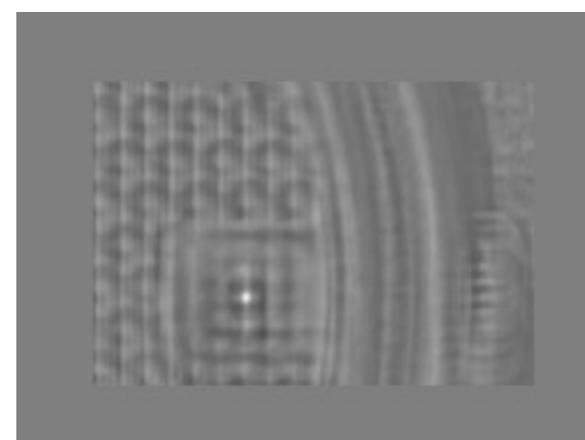
Pattern Detection by Template Matching

Reference template (33×31 pixels)

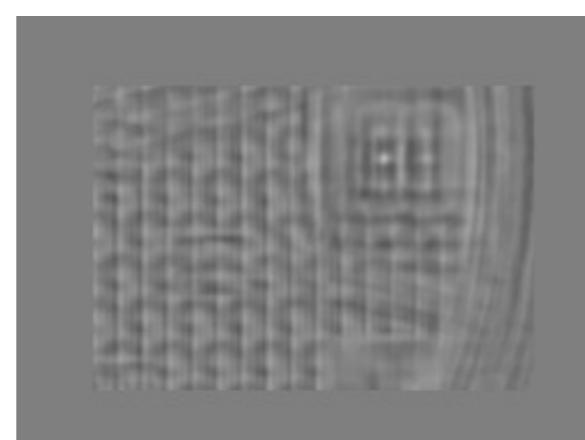
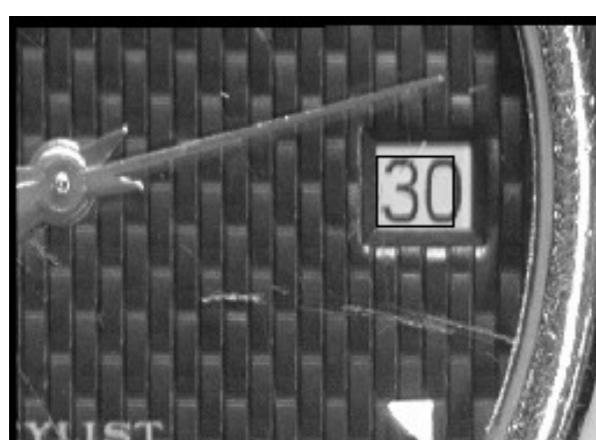
31



$$(x, y) = (149, 95)$$
$$\rho = 100\%$$



$$(x, y) = (98, 123)$$
$$\rho = 88\%$$



$$(x, y) = (58, 61)$$
$$\rho = 33\%$$

Feature Extraction

- Edge detection

Edges are important clues for the interpretation of images; they are essential to object recognition

- Edges: Analog formulation
- Gradient-based edge detection

Edges: Analog Formulation

What is an edge?

Definition: An edge point is a location of abrupt change in an image

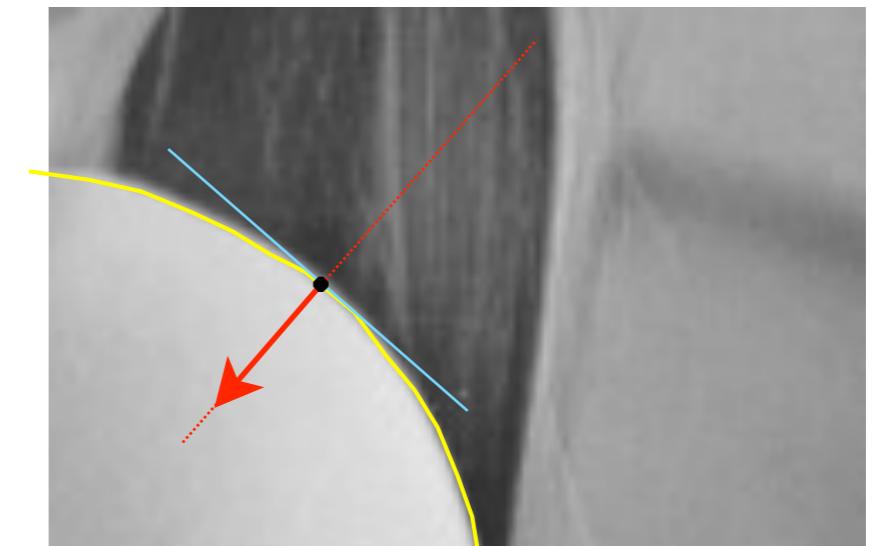
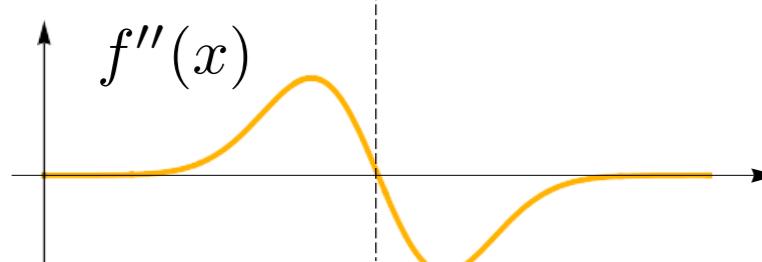
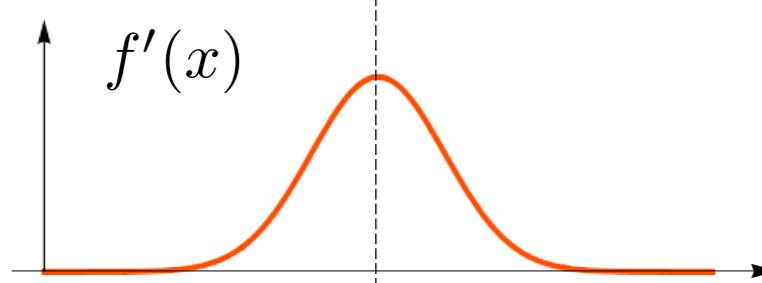
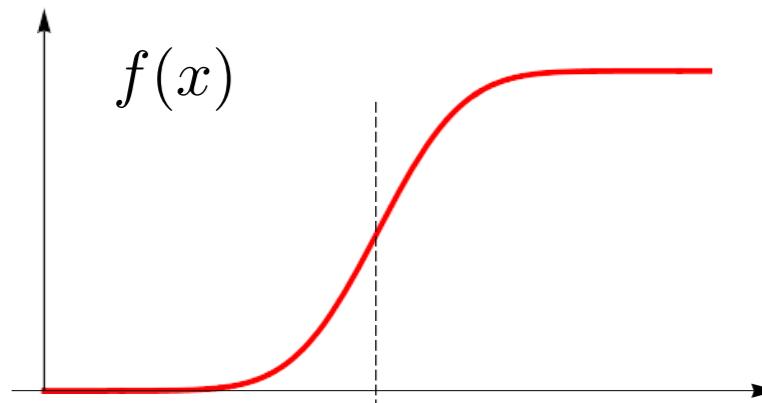


Image value at location x : $f(x)$

Normal vector: $\mathbf{n} = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|_2}$

\Rightarrow direction of maximum change

Gradient and Directional Derivatives

- Gradient of f at $\mathbf{x} = (x, y)$: $\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x}, \frac{\partial f(\mathbf{x})}{\partial y} \right) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$
- Directional derivative of f along the unit vector $\mathbf{u}_\theta = (\cos \theta, \sin \theta)$

$$D_{\mathbf{u}_\theta} f(\mathbf{x}) = \lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{u}_\theta) - f(\mathbf{x})}{\varepsilon}$$

Taylor-series argument:
 $f(\mathbf{x} + \varepsilon \mathbf{u}) = f(\mathbf{x}) + \varepsilon \mathbf{u}^\top \nabla f(\mathbf{x}) + O(\varepsilon^2)$

$$\begin{aligned} &= f_1(\mathbf{x}) \cos \theta + f_2(\mathbf{x}) \sin \theta \\ &= \mathbf{u}_\theta^\top \nabla f(\mathbf{x}) \end{aligned}$$

Exercise: What is $\max_\theta D_{\mathbf{u}_\theta} f(\mathbf{x})$?

$$\max = D_{\mathbf{n}} f(\mathbf{x}) = \|\nabla f(\mathbf{x})\|_2 = \sqrt{f_1(\mathbf{x})^2 + f_2(\mathbf{x})^2}$$

$$\theta^* = \angle(\nabla f(\mathbf{x})) = \arctan \left(\frac{f_2(\mathbf{x})}{f_1(\mathbf{x})} \right) + k\pi, k \in \mathbb{Z} \quad (\perp \text{ to edge})$$

General Criteria for Edge Detection

- Maximum of the gradient
- Zero crossings of the second-order (directional) derivative
- Combination of both

Remarks:

- Gradient magnitude and Laplacian are **rotationally invariant**, while gradient vectors and directional second-order derivatives are not
- Derivatives are usually estimated on a smoothed version of the image to improve robustness and/or reduce the effect of noise

Gradient-Based Edge Detection

How do we design discrete filters that mimic gradients?

- Discretized gradient operators

$$\text{Horizontal derivative: } g_1[\mathbf{k}] = (h_1 * f)[\mathbf{k}]$$

$$\partial_x \approx \begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$$

$$\text{Vertical derivative: } g_2[\mathbf{k}] = (h_2 * f)[\mathbf{k}]$$

$$\partial_y \approx \begin{bmatrix} \frac{1}{2} \\ 0 \\ -\frac{1}{2} \end{bmatrix}$$

$$g[\mathbf{k}] = \sqrt{g_1[\mathbf{k}]^2 + g_2[\mathbf{k}]^2}$$

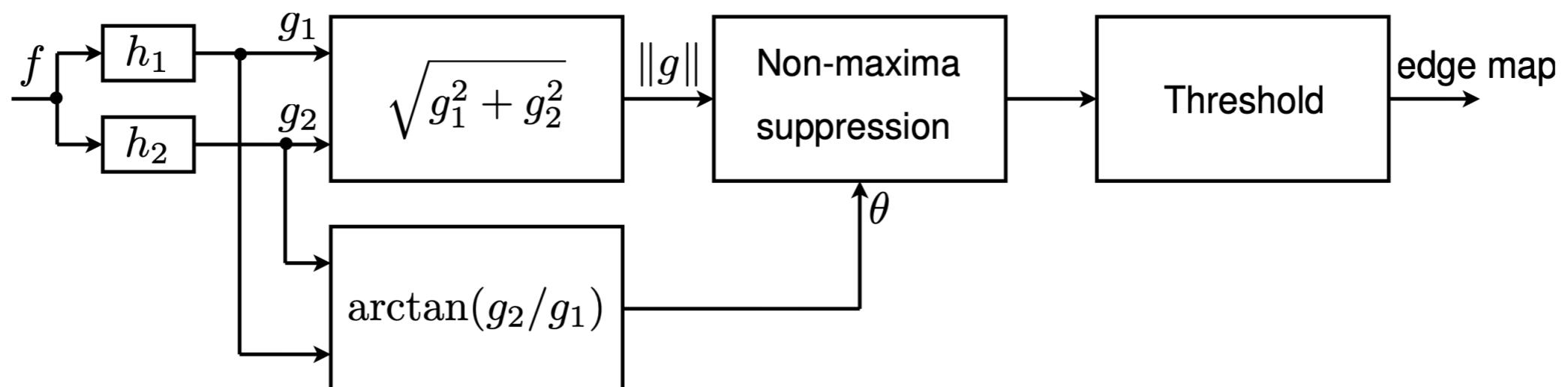
$$\theta[\mathbf{k}] = \arctan \left(\frac{g_2[\mathbf{k}]}{g_1[\mathbf{k}]} \right)$$

- Threshold-based edge detection

$$\text{edge}[\mathbf{k}] = \begin{cases} 1, & g[k_1, k_2] \geq T \\ 0, & \text{else} \end{cases}$$

Canny's Edge Detection Algorithm

- Refinements:
 - Non-maxima suppression: Using knowledge of $\theta[k]$
 - Intelligent thresholding



<https://bigwww.epfl.ch/demo/ip/demos/edgeDetector/>

Image Segmentation

- Segmentation: Art or Science?
- Amplitude Thresholding
 - Variational Thresholding
 - Statistical Thresholding
- Binary Segmentation Techniques

What is Segmentation?

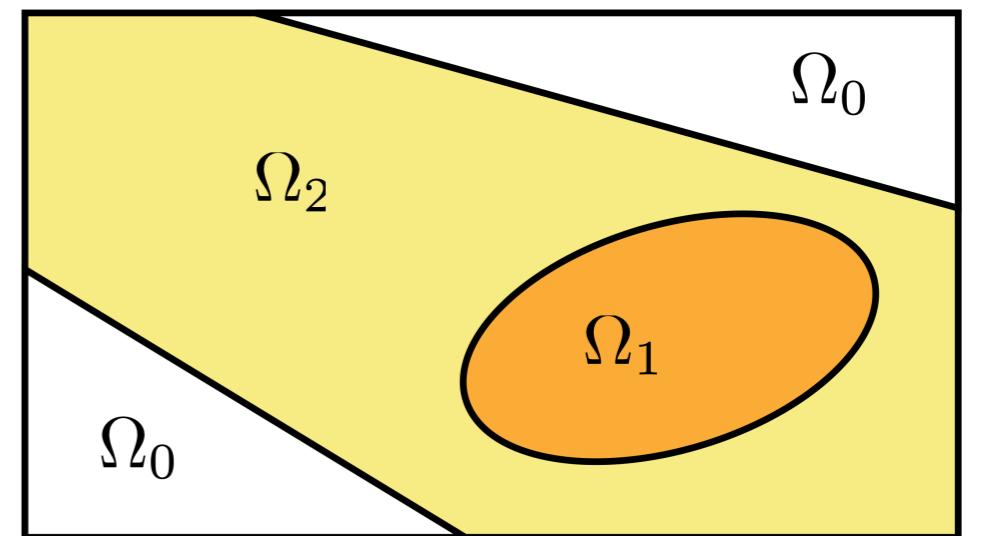
- Definition

Image $f[\mathbf{k}]$, with $\mathbf{k} \in \Omega$

Image segmentation: Find a partition of the support Ω of the image f , with

$$\Omega = \bigcup_i \Omega_i \quad \text{with} \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j$$

such that the regions Ω_i satisfy some homogeneity (and connectivity) criterion.



The total number of regions is not necessarily known

- Three main approaches (not based on deep learning)
 - Pixel classification
 - Region-based segmentation
 - Boundary-based segmentation \Rightarrow Edge detection

Segmentation: Art or Science?

Problem: lack of a universal definition of homogeneity
⇒ many application-specific approaches

- Approaches for specifying homogeneity
 - Empirical (e.g., similar graylevels; feature maps)
 - Statistical, based on some a priori model (e.g., constant mean + additive white noise)
- Approaches for enforcing connectivity (if required)
 - Prior information about object size or shape
 - Joint probability model for class labels
 - Contour length