T-61.6030 Special Course in Computer and Information Science
Multimedia Retrieval

Image Processing part 2.

László Kozma <Lkozma@cis.hut.fi>
Helsinki University of Technology
15.2.2008

Regions

- **Segmentation**: partition image plane into *disjoint*, *meaningful* regions
- ***top-down*** methods: restrictions regarding shape, position, orientation of the regions come from the application domain
- ***bottom-up*** methods: no restrictions, building from the pixel level
  - problems: over-/under- segmentation

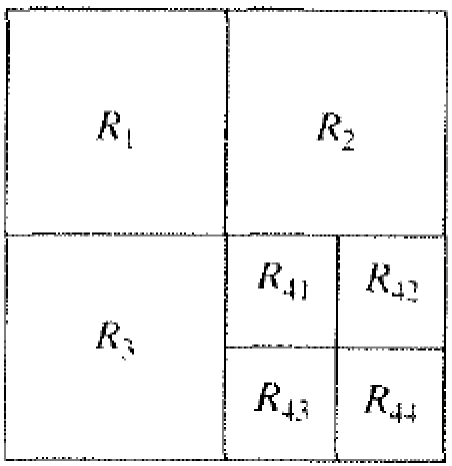
Segmentation

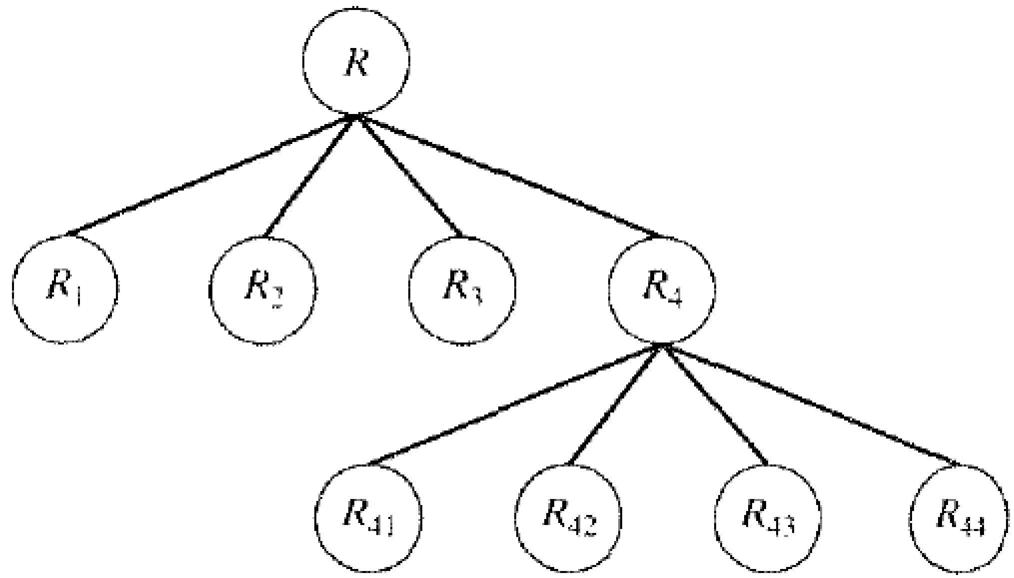
- ***Area-based***: group neighboring pixels if local homogeneity is satisfied
- ***Edge-based***: find boundaries between regions (where homogeneity is broken)

Area-based segmentation

- 1. Merge-and-split
  - initial partitioning of the image
  - if a region not locally homogenous, split further
  - do until all regions are homogenous
  - pick two neighboring regions
  - if together they form a homogenous region, merge
  - repeat until no merging possible

Splitting





source: Gonzales, Woods ...

Homogeneity

- Possible criteria:

$$\text{max-min criterion:} \max_{(n,m)\in region}(f_{n,m}) - \min_{(n,m)\in region}(f_{n,m}) < \text{threshold}$$

$$\text{squared error criterion :} \frac{1}{area_{region}} \sum_{(n,m)\in region} (f_{n,m} - \hat{\mu}_{region})^2 < \text{threshold.}$$

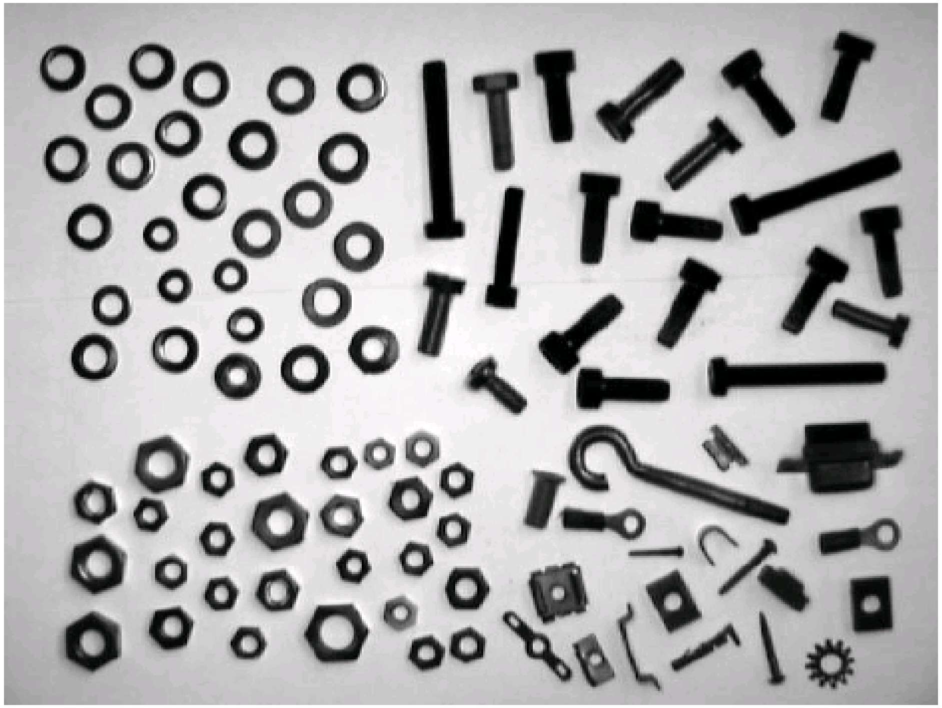
f – intensity, RGB component, etc.

Area-based segmentation

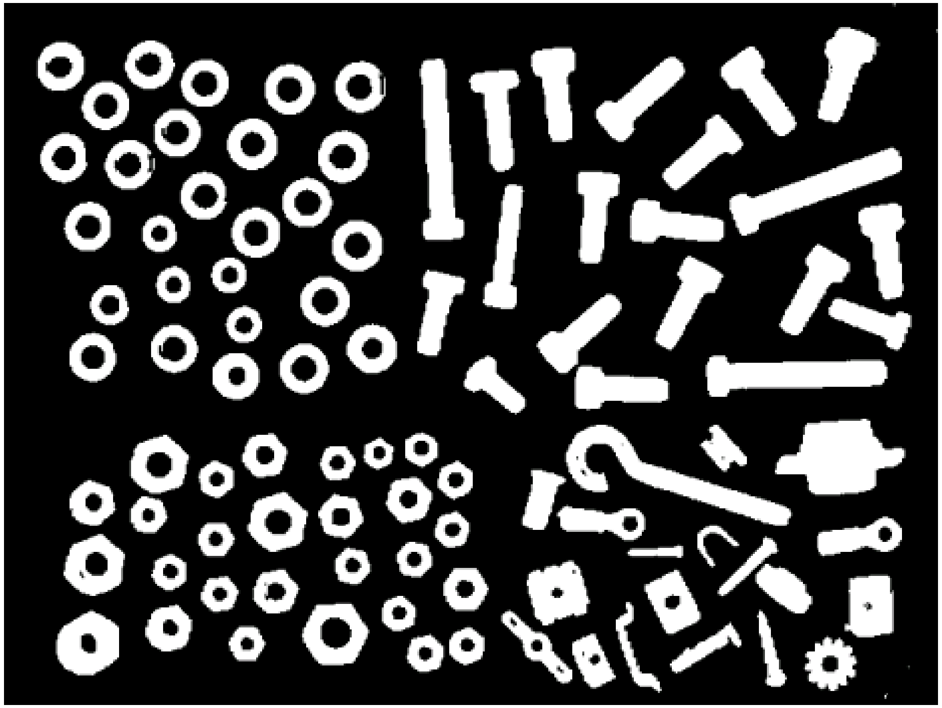
- 2. Pixel classification
  - merge neighboring pixels, if they have similar features
  - features: intensity, RGB components, texture, ...
  - prone to noise
  - works only with well-controlled illumination, uniform background, etc.
  - morphological corrections might be useful

Pixel classification

(a)



(b)



source: Blanken ...

Edge-based segmentation

- detect edge (line) elements
- connect neighboring edge elements (bridge gaps, etc.)
- problem: high frequency texture can cause spurious edge elements
- use domain knowledge to correct edges
- example:
  - tennis court
  - road lane detection

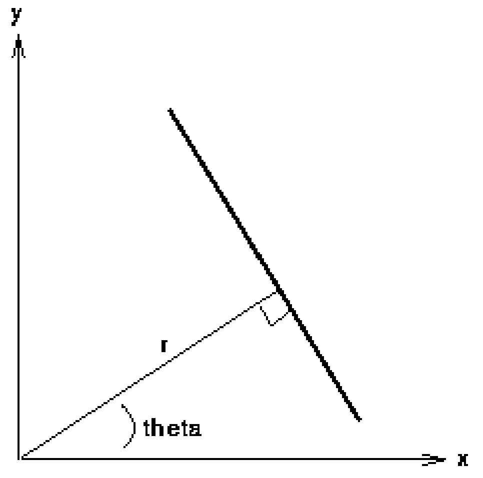
Hough transform

- detect features specified in parametric form (for example straight lines)
- most often used for lines and other simple parametric curves
- Line equation:

$$x * \cos(\Theta) + y * \sin(\Theta) = r$$

  - (x,y) edge elements from image (constant), find corresponding r and Θ
  - collinear points yield r and Θ that intersect in common point

Hough transform (straight line)



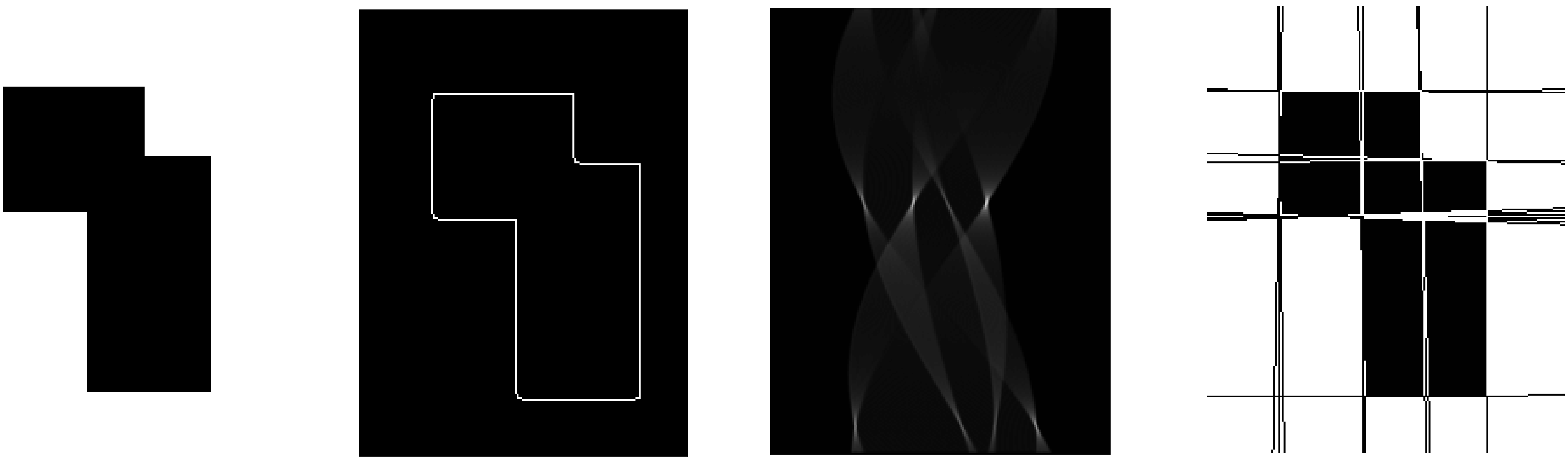
$$x * \cos(\Theta) + y * \sin(\Theta) = r$$

(why not y = m*x + n ?)

Hough algorithm

- 1. Detect edge elements
- 2. Transform to Hough-space
  - for each (x,y) edge element in image space find corresponding set of (r, Θ) in Hough-space
  - for different values of Θ calculate r (sinusoid)
  - add value to “accumulator” in Hough-space (vote for a line)
- 3. Detect maxima in Hough-space
  - thresholding, thinning
- 4. Transform maxima back to image space
  - each (r, Θ) describes a line
- 5. Find segments of the detected lines

Hough transform



Hough transform

- Resolution of parameters is a trade-off
- Can be used for any other parametric curve (circle, parabola, etc.) or generic object
- Not practical, unless we can restrict parameters (for ex. circles with a given radius)
- Computationally expensive
- Can be easily implemented in parallel
- Robust to noise

Regional description

- After segmentation:
- Describe regions:
  - **radiometry** (mean/variance of intensity, RGB components, texture description)
  - **geometry** (position, orientation, shape)
  - **relation** vs. other regions (adjacency, relative size, etc.)

Radiometry

- mean value of intensity, RGB component
- simple statistical measures:
  - variance, skewness, kurtosis
- higher order statistics (P – normalized co-occurence matrix):

$$\text{contrast} \quad \sum_{i,j} (i-j)^2 P_{\theta,\rho}(i,j)$$

$$\text{energy} \quad \sum_{i,j} P^2_{\theta,\rho}(i,j)$$

$$\text{correlation} \quad \sum_{i,j} \frac{(i-\mu)(j-\mu)}{\sigma^2} P_{\theta,\rho}(i,j)$$

Geometric properties

- Describing area:

centroid:

$$\bar{x} = \frac{1}{area} \sum_{(n,m)\in region} n \qquad \bar{y} = \frac{1}{area} \sum_{(n,m)\in region} m,$$

moments:

$$M_{pq} = \frac{1}{area} \sum_{(n,m)\in region} n^p m^q$$

centralized moments:

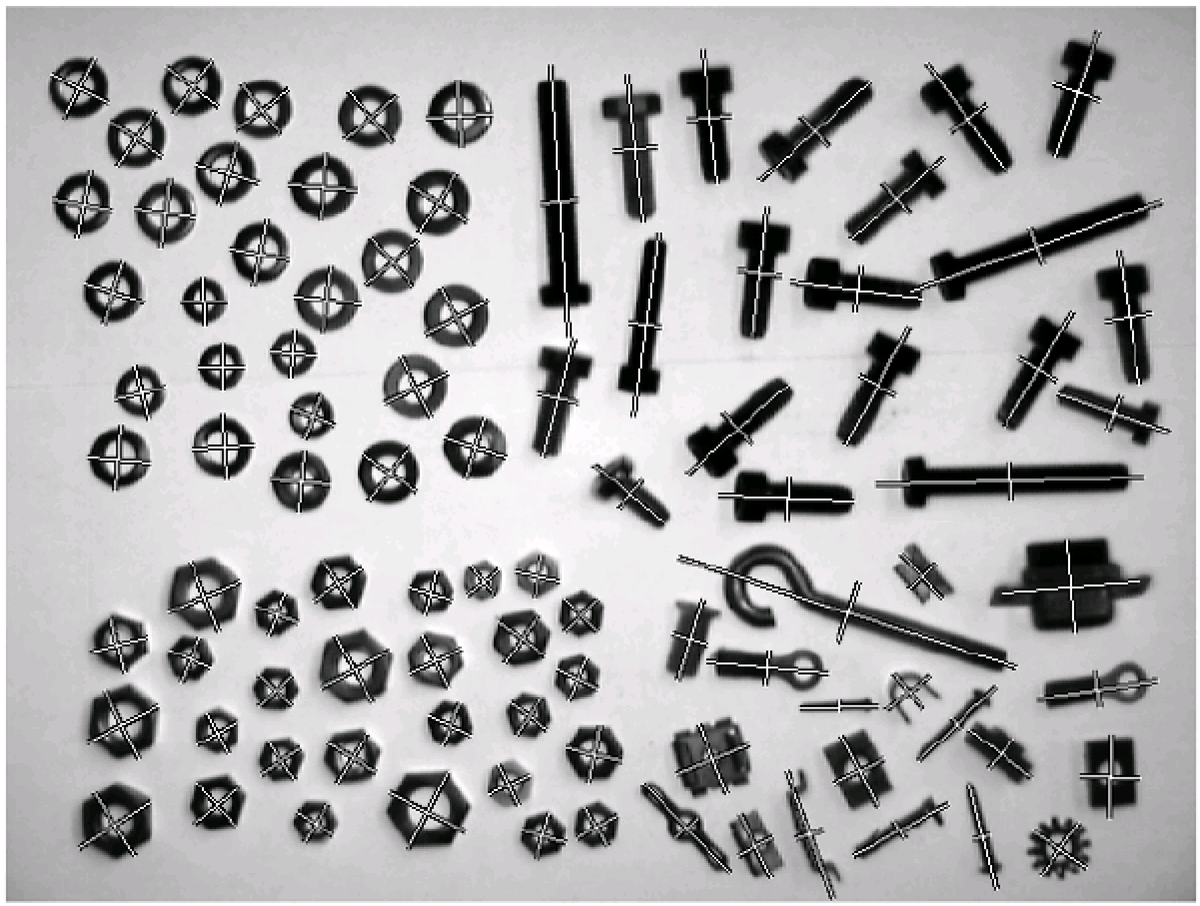
$$\mu_{pq} = \frac{1}{area} \sum_{(n,m)\in region} (n - \bar{x})^p (m - \bar{y})^q$$

second order moments

(principal axes spanned by eigenvectors):

$$\begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix}$$

Geometric properties



principal axes derived from second order moments

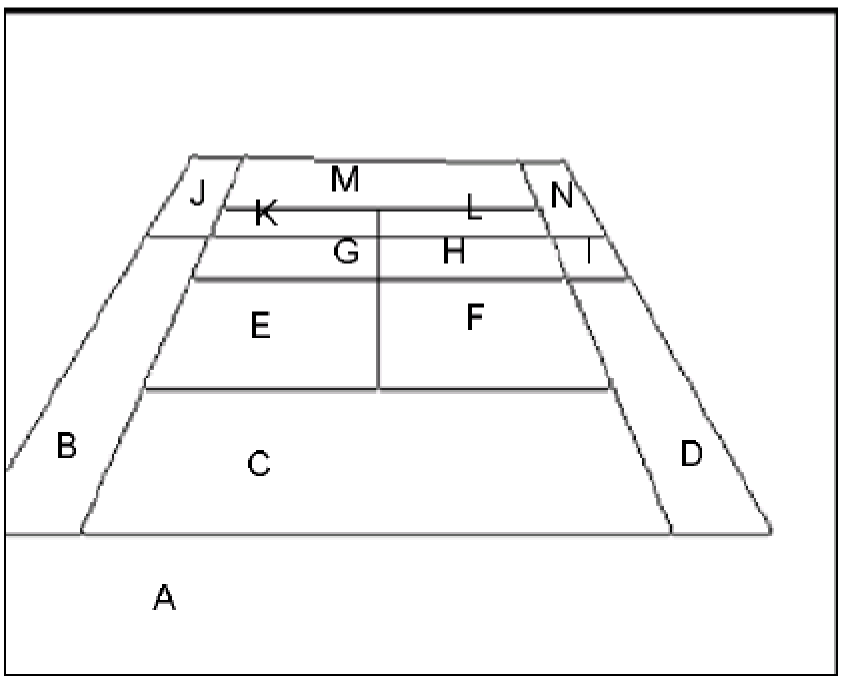
Geometric properties

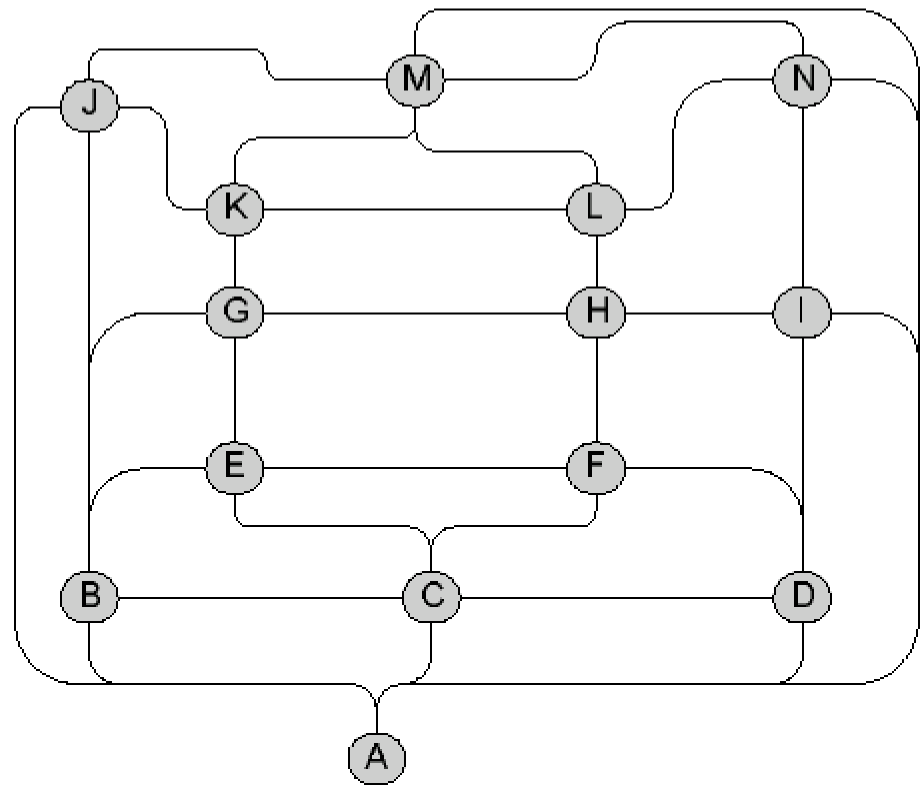
- Describing contour
  - $(x(s), y(s))$ - x,y periodic functions of s
  - $(x(0), y(0))$ selected arbitrarily
  - $x(s+P) = x(s)$, $y(s+P) = y(s)$
  - Fourier spectra can be obtained
  - Classify shapes based on Fourier descriptors

Relations

- Relations: properties *between* regions:
  - relative size
  - normalized distance (relative to size)
  - similarity (alikeness of shape based on Fourier)
  - adjacency (fraction of common boundary)

Relations: adjacency graph





Object recognition

- top-down approaches:
  - template matching
  - eigenobjects
  - statistical shape models

Template matching

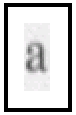
- Object template matched against the image in different positions
- Find locations with best match
- Difference functions:
  - Absolute difference:

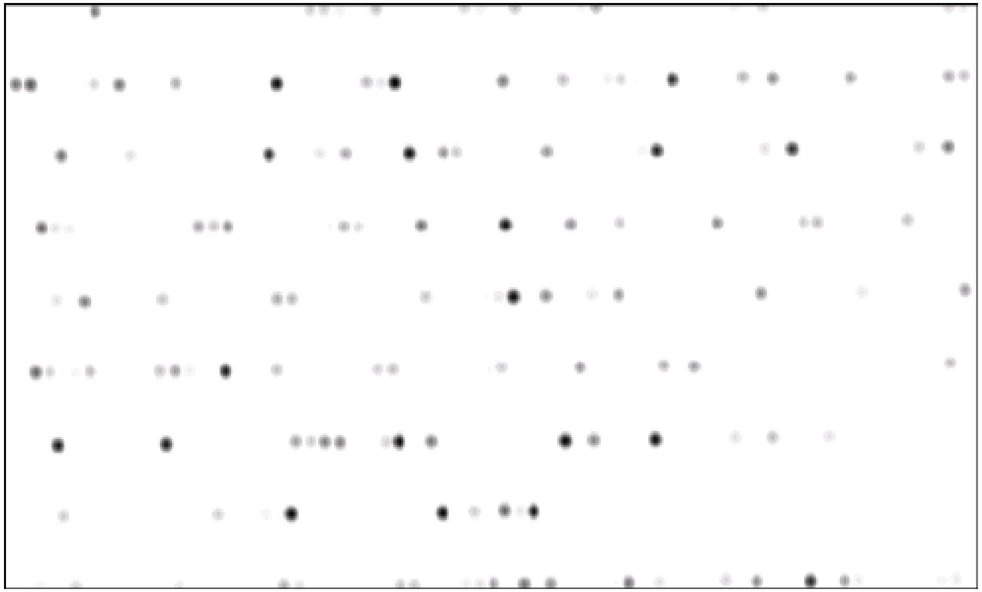
$$d_{T,I}(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} |T(u,v) - I(x+u, y+v)|$$

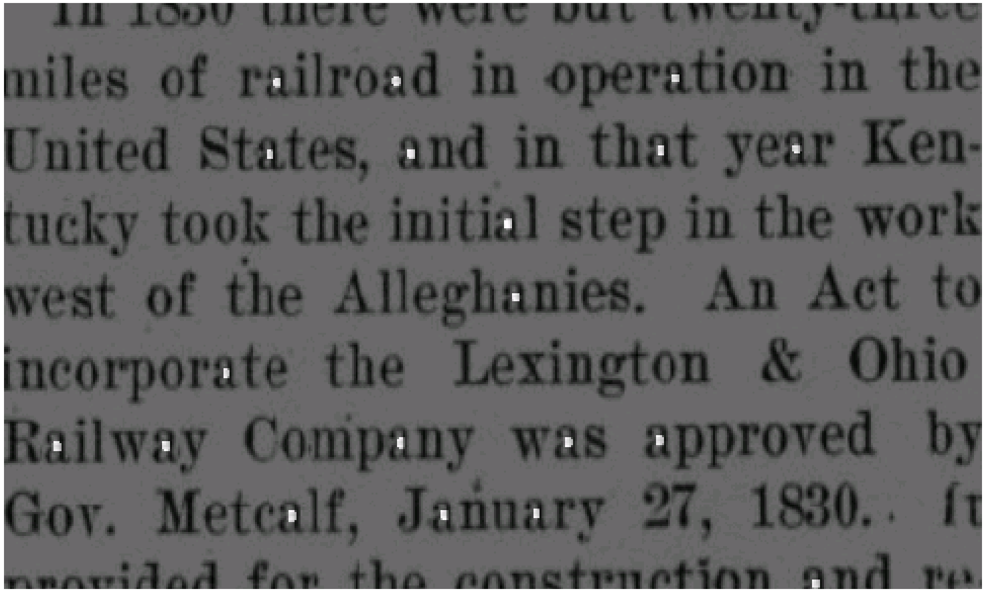
  - Cross-correlation (more robust, ex. different illumination):

$$C_{T,I}(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} T(u,v) \cdot I(u-x, y+v)$$

Template matching






miles of railroad in operation in the
United States, and in that year Ken-
tucky took the initial step in the work
west of the Alleghanies. An Act to
incorporate the Lexington & Ohio
Railway Company was approved by
Gov. Metcalf, January 27, 1830.

Eigenobjects

- Idea: any object can be represented as a mean plus a weighted sum of eigenobjects

$$O = \bar{O} + \sum_i w_i e_i.$$

- Object given as a vector of features: $O^i = \{u_1^{\ i}, u_2^{\ i}, \ldots, u_n^{\ i}\}$
- Eigenobjects found from training set using PCA
- A new object is described by weights w_i
- Object is identified using distance between its weights and the weights of known objects
- Other classification methods can be used

Eigenfaces

Statistical Shape Models

- Represent shape as a set of connected landmarks
  - example: on hand images tips of fingers
- Building model:
  - normalize training data (translate, rotate, scale)
    - “procrustes analysis”
  - detect landmarks
  - describe shape by coordinates of landmarks
  - PCA analysis can be used here too

Statistical Shape Models

$$\mathbf{x}_i = [x_{i1}, y_{i1}, x_{i2}, y_{i2}, .., x_{in}, y_{in}]^T$$

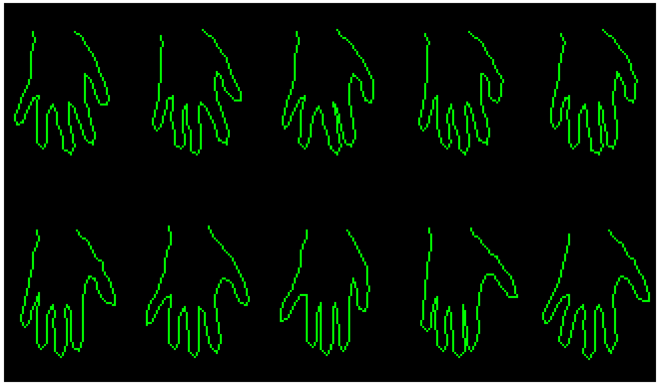
$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i,$$

Statistical Shape Models

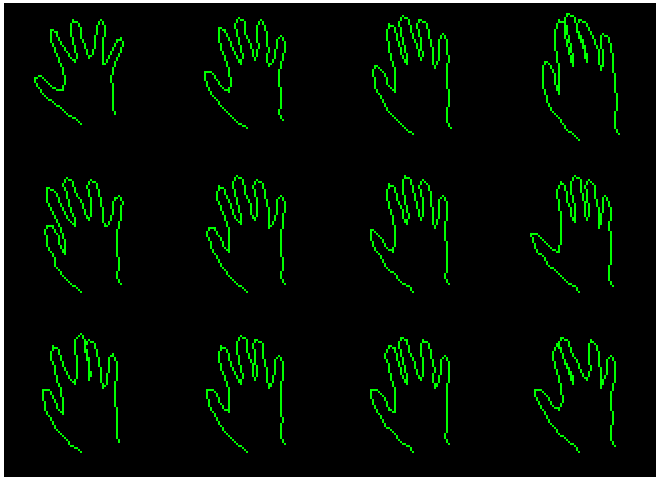
- Principal components describe modes of deformation
- By varying weights, we can generate new examples
- For a new object, we perform procrustes analysis
- Find weights of principal components
- Match weights to those of example shapes in the database

Statistical Shape Models

- training shapes:



- generated shapes:



Overview

- Regions
  - Segmentation
    - area-based
    - contour-based
  - Description
    - radiometry
    - geometric
    - relations
- Object recognition
  - Template matching
  - Eigenobjects
  - Statistical shape models

Sources

- Blanken, et. al. : Multimedia Retrieval
- Gonzales, Woods: Digital Image Processing
- Hough transform: http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm (University of Edinborough)
- Statistical shape models: http://www.isbe.man.ac.uk/research/Flexible_Models/pdms.html (University of Manchester)