Image Analysis - Lecture 15

Kalle Åström

15 October 2014

Lecture 14

Contents

- Image Analysis
- Gray-level operations
- Filtering
- Features
- Segmentation
- Classification
- Statistical Image Analysis
- Computer Vision

Image Analysis

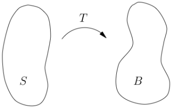

Image Processing: Enhance the image (image -> image)
Image Analysis: Interpret the image (image -> interpretation)
Computer vision: Mimic human vision, geometry, interpretation
Computer graphics: Generate images from models

Image models

- Continuous model

$$f : \Omega \mapsto R_+ \quad ,$$

where $\Omega = \{\,(x, y) \mid a \leq x \leq b,\ c \leq y \leq d\}$ and
$R_+ = \{x \in R \mid x \geq 0\}$.
$f(x, y)$ = intensity at point (x, y) = gray level

- A discrete model
Discretize x, y, called **sampling**
Discretize f, called **quantification** usually in $G = 2^m$ levels.

Interpolation

Given an image with **discrete** representation one can obtain an **continuous** version by interpolation.
Problem: (Interpolation)
Given $f(i,j), \quad i,j \in \mathbb{Z}^2$.
"compute" $f(x,y), \quad x,y \in \mathbb{R}^2$

$$f(x,y) = \sum_{i,j} h(x-i, y-j) f(i,j),$$

with different interpolation functions h.

- Nearest neighbor interpolation
- Linear/bilinear interpolation
- Cubic spline interpolation
- Sinc-interplation
- Gaussian interpolation

Gray-level transformation

A simple method for image enhancement
Let $f(x, y)$ be the intensity function of an image. A **gray-level transformation**, T, is a function (of one variable)

$$q(x, y) = T(f(x, y))$$
$$s = T(r) \quad ,$$

that changes from gray-level f to gray-level q. T usually fulfils

- $T(r)$ increasing in $L_{min} \leq r \leq L_{max}$,
- $0 \leq T(r) \leq L$.

■

Example: Thresholding

$$T(r) = \begin{cases} 0 & r \leq m \\ 1 & r > m, \end{cases}$$

Graylevel transformations and histograms

- Let $s = T(r)$ be a grayscale transformation ($r = T^{-1}(s)$)
- Let $p_r(r)$ be the frequency function for the original image.
- Let $p_s(s)$ be the frequency function for the resulting image.

Relation:

$$p_s(s) = p_r(r)\frac{dr}{ds} \quad (s = T(r)) \ .$$

Histogram equalisation

$$s = T(r) = \int_0^r p_r(t)dt$$

Linear Algebra

- Linear Algebra
- Projections, Pseudo-inverse
- Image matrix
- The Fourier transform in 1 and 2 dimensions
- The discrete Fourier transform in 1 and 2 dimensions
- The Fast Fourier Transform (FFT)

The discrete Fourier transform

$$f = (f(0), f(1), \ldots, f(N-1)) \ .$$

$$F(u) = \sum_{k=0}^{N-1} f(k)\omega_N^{ku}, \ u = 0, \ldots N-1 \ ,$$

$$\omega_N = e^{-i2\pi/N} \ .$$

$$\mathcal{F}_N = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega_N & \omega_N^2 & \ldots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \ldots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \ldots & \omega_N^{(N-1)(N-1)} \end{bmatrix} , F = \mathcal{F}_N f$$

The discrete Fourier transform

- Inversion

$$\overline{\mathcal{F}_N}\mathcal{F}_N = NI \quad \Rightarrow \quad {\mathcal{F}_N}^{-1} = \frac{1}{N}\overline{\mathcal{F}_N}$$

- FFT

$$\mathcal{F}_{2N} = \begin{bmatrix} I & D_N \\ I & -D_N \end{bmatrix} \begin{bmatrix} \mathcal{F}_N & 0 \\ 0 & \mathcal{F}_N \end{bmatrix} P_{2N} \ ,$$

- DFT and FFT in 2 dimensions
- FFT on images
- Computational complexity

Image enhancement and filters

- Degradation can be modelled by convolution (linear shift invariant filters)
- Frequency function
- High pass - low pass - band pass
- Differentiation
- Robert-, Prewitt-, Sobel-filter.

Discrete convolution and FFT

- Discrete convolution
- Convolution theorem 1D
- Convolution theorem 2D
- Computational complexity

Feature detection

- Technique:
  - Scale space (smoothing)
  - Interpolation
  - Differentiation
  - Noise
  - Scale space pyramid
- Edge detection
- Orientation tensor
- Corner/interest point detection
- Other detectors: Ridges

Edge detection

- Gradient methods (max of derivative = location of edge)

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

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \text{ (Roberts)}$$

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

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

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \text{ (Prewitt)}$$

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \text{ (Sobel)}$$

Other methods

- Laplace methods (zero point of second derivative = location of edge)
- Advanced methods (Canny-Deriche)
- Sub-pixel edge detection

What is segmentation?

Goal: Segment the image into pieces/segments, i.e. regions that belong to the same object or that has the same properties. Can also be seen as a as a problem of 'grouping' of pieces (pixels, regions) together.

Segmentation using clustering

- Segment images into pieces
- Fit lines to a set of points
- Fit a fundamental matrix to image pairs

In some cases it is easier to view segmentation as the problem of putting pieces together. This is usually called **grouping** (less precise) or **clustering** (which has a precise meaning in the field of pattern recognition).

Clustering

Goal: Partition a set on *n* feature vectors

$$x_1, \dots x_n$$

with *d* components, i.e. $x_i \in \mathbb{R}^d$, in groups (clusters) such that all examples in the same group are similar.

The Clustering Problem

Input: n examples $x_1, \ldots x_n$.
Output: A mapping

$$c : \{1, \ldots, n\} \mapsto \{1, \ldots, g\}.$$

Example: **K-means algorithm**
Choose g cluster centre $m_1, \ldots, m_g$ and a clustering function c that minimizes

$$f(c, m) = \sum_{i=1}^{n} |x_i - m_{c(i)}|^2.$$

K-means

$$\min_{c,m} f(c,m) = \min_{c,m} \sum_{i=1}^{n} |x_i - m_{c(i)}|^2$$

1. Randomly choose g cluster centres (e.g. g examples)
2. **Optimize** c**:** For every example x_k assign $c(k)$ such that $|x_i - m_{c(i)}|$ is minimized, i.e.

$$c = \text{argmin}_c f(c,m).$$

3. **Optimize** m**:** For every group j change m_j as the centre of mass of corresponding examples, i.e.

$$m = \text{argmin}_m f(c,m).$$

Hierarchical methods

A popular clustering method is hierarchical clustering.
Start with one cluster.
Reduce one after one.
Stop whenever you are finished.

1. Start with n clusters
2. Choose error criteria, e.g. $f(c, m)$ as above
3. Merge the two clusters that minimizes the error criteria
4. If the the number of clusters is more than g go to 3.

The result is a clustering for every number of clusters between g between 1 and n. This can often be represented in a so called dendrogram, where you map error criteria and clustering.

There are **many** other methods

- Graph theoretical clustering
- EM-method (Similar to k-means, but better (and slower))
- k-medoids
- adaptive resonance theory
- fuzzy clustering
- auto-class
- mode-separator
- Self Organizing Maps
- Agglomerative hierarchical clustering
- Divisive hierarchical clustering
- Isodata, Isomap

Segmentation using fitting

- Fitting
  - Hough transform
  - Lines
  - Curves (in particular conics)
  - as inference
- Robustness
  - M-estimation
  - RANSAC
- Example: Robust fitting of the fundamental matrix

The Hough transform

Represent lines as

$$x\cos(\theta) + y\sin(\theta) = \rho$$

$$-\infty < \rho < +\infty, \quad -\frac{\pi}{2} < \theta \leq \frac{\pi}{2}$$

gives $\rho\theta$-plane and can represent all lines.
Map point nr k on the curve

$$\Lambda_k = \{\, (\rho, \theta) \,|\, (x_k, y_k) \in l_{\rho,\theta} \,\}$$

Idea: Cover the $\rho\theta$-plane with squares (accumulator cells) and add 1 *to the cell that contain Λ_k.*

Segmentation using classification

- Classification
- Statistically optimal classification
- Nearest neigbour classification
- Artificial Neural Networks
- Support Vector Machines
- Dimensionality reduction
- Invariants

Scale space

$$V_j = \{\, f \in L^2 \mid f = \sum_{j' \geq j} c_{j',k} \Psi_{j',k} \,\}$$

Each function $f_j \in L^2$ can be written as

$$f_j = f_{j+1} + d_j \ \text{where} f_j \in V_j$$

Additionally $V_{j+1} \subset V_j$.
"Information" in V_j = "information" in V_{j+1} + "j-details".
"j-details" is represented by $\Psi_{j,k}$.

Classification

Example: OCR, Cell analysis, diagnostic support
Feature vectors:

$$\mathbf{x} = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}^T$$

Compare $\mathbf{x}$ with classes $\omega_1, \dots, \omega_M$.
Decision functions $d_i : \mathbf{x} \to \mathbb{R}$.
Methods

- Matching
- Optimal statistical classifier (plug-in classifiers)
- Artificial Neural Networks
- (k,l)-nearest neighbor
- Support vector machines

Matching

Mean prototype

$$m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x$$

Distance

$$D_j(\mathbf{x}) = ||x - m_j||$$

Artificial Neural Networks

Idea: Build a classifier function built on *perceptrons* $d(\mathbf{x}) = f(w^T\mathbf{x} + w_0)$ where f is a non linear function, e.g.

$$f(y) = \frac{1}{\pi}\,\text{atan}(y) + \frac{1}{2}$$

By comparing examples $\mathbf{x}_j$ with ground truth y_j one can find the parameters (v, w) by optimizing

$$\min_{(v,w)} \sum ||d(\mathbf{x}_j) - y_j||$$

Problems with local minima.

Statistical image analysis

Bayes methods

1. A priori probabilities

$$p(w), w \in \{1, \ldots, M\}$$

$$\sum_{w} Pr(w) = 1$$

2. Likelihood functions

$$p(x|w)$$

3. A posteriori probabilities

$$p(w|x) = \frac{p(x|w)p(w)}{p(x)} \sim p(x|w)p(w)$$

- Maximum likelihood estimation (ML)

$$\max_{w} Pr(x|w)$$

- Maximum A posteriori estimation (MAP)

$$\max_{w} Pr(w|x)$$

- Minimum Posterior Expected Loss

$$\min_{c} M(c|x)$$

$$M(c|x) = \sum_{w} L(c|w) Pr(w|x)$$

where $L(c|w)$ denotes loss when c is classified as the pattern w.

Computer vision

Can be seen as and attempt at mimicking human (biological) vision.

Some examples of research areas:

- Recognition
- Motion estimation
- 3D reconstruction from 2D images

Transformations

- Euclidean transformations

$$y = Rx + b, \quad RR^T = I$$

- Affine transformations

$$y = Ax + b$$

- Projective transformations

$$\lambda \begin{bmatrix} y \\ 1 \end{bmatrix} = T \begin{bmatrix} x \\ 1 \end{bmatrix} .$$

Camera matrix

$$\lambda \begin{bmatrix} y_1' \\ y_2' \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ 1 \end{bmatrix}, \quad K = \begin{bmatrix} f & sf & x_0 \\ 0 & \gamma f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic parameters:

- f: **focal length**
- (x_0, y_0): **principal point** (orthogonal projection of the camera centre in the image plane.)
- γ: **aspect ratio** (rectangular light sensitive elements instead of quadratic)
- s: **skew** (skewed light sensitive elements)

Extrinsic parameters:

- R: rotation
- t: translation
- λ: depth

Camera equation, epipolar constraint

Camera equation $\lambda x = PX$
Camera centre $PC = 0$
Two projections of same point

$$\lambda_1 x_1 = P_1 X$$

$$\lambda_2 x_2 = P_2 X$$

Rewrite as an equation system

$$\underbrace{\begin{pmatrix} P_1 & x_1 & 0 \\ P_2 & 0 & x_2 \end{pmatrix}}_{M} \begin{pmatrix} X \\ -\lambda_1 \\ -\lambda_2 \end{pmatrix} = 0$$

- Epipolar geometry, epipolar constraint $\det(M) = x_1^T F x_2 = 0$
- Eight point algorithm
- Epipoles $e_1^T F = 0$, $F e_2 = 0$, epipolar lines $l_2 = x_1^T F$

Multi-spectral image analysis

Gray level images gives only one number the represents the whole intensity spectrum.
Other methods can be used to measure several components of the intensity spectrum at each pixel.
Humans measure three channels. Some animals have 4, 5 or even 12 channels.
Multi-spectral images are measurements of many frequencies in each pixel.
Ex: Color images have three frequency bands (Red, Green, Blue)
Better for classification.
Principal component analysis important tool

System building and testing

Image analysis tools can be used to build small systems (image -> interpretation).

To understand the performance of the system it has to be evaluated on data with ground truth.

It is often hard work to obtain ground truth.

Preferably one would need also ground truth for different steps of the system.

Try to establish ground truth data and a benchmark routine early on.

Other courses

Mathematics:

- Medical Image Analysis
- Computer Vision
- Optimization
- Linear and combinatorial optimization

Mathematical Statistics:

- Statistical image analysis

Physics:

- Multispectral image analysis

Thank you for your contribution in this course

Good luck in the future with exam, projects, masters thesis!
Keep in touch and tell me about any interesting projects that you do with image analysis and computer vision.