Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 28: A walk through the lecture series

Winter Semester 2014/15

Dr. Benjamin Seppke
Prof. Siegfried Stiehl

Definition of Image Understanding

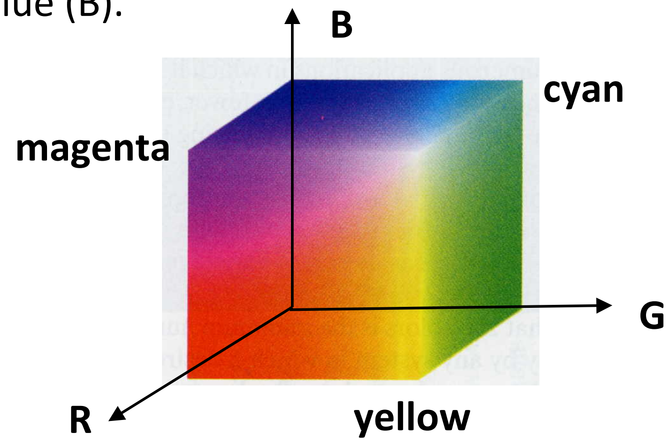
Image understanding is the task-oriented reconstruction and interpretation of a scene by means of images

- "scene": section of the real world
  - stationary (3D) or
  - moving (4D)
- "image": view of a scene
  - projection, density image (2D)
  - depth image (2 1/2D)
  - image sequence (3D)
- "reconstruction and interpretation“: computer-internal scene description
  - Quantitative
  - Qualitative
  - Symbolic
- "task-oriented": for a purpose, to fulfil a particular task
  - context-dependent,
  - supporting actions of an agent

Computer Vision Colour Models

RGB colour model – Inverse to CMY

Different colors are generated by adding different portions of red (R), green (G), and blue (B).



RGB is the most commonly used color space in Computer Vision.

Typical discretization:

8 bits per colour dimension

→ 16.777.216 colours

HSI colour model

Different colors are described by Hue (H), Saturation (S), and Intensity (I). Can be derived from RGB model:

$$H = \begin{cases} Q & \text{if} \quad B \leq G \\ 360 - Q & \text{if} \quad B > G \end{cases}$$

$$Q = \arccos\left(\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right)$$

$$S = 1 - \frac{3}{R+G+B}\min(R,G,B)$$

$$I = \frac{R+G+B}{3}$$

Closer to human perception

Better choice e.g. for selecting colors!

Sampling Theorem

Shannon´s Sampling Theorem:

> A bandlimited function with bandwidth W can be exactly reconstructed from equally spaced samples, if the sampling distance is not larger than $\frac{1}{2W}$.

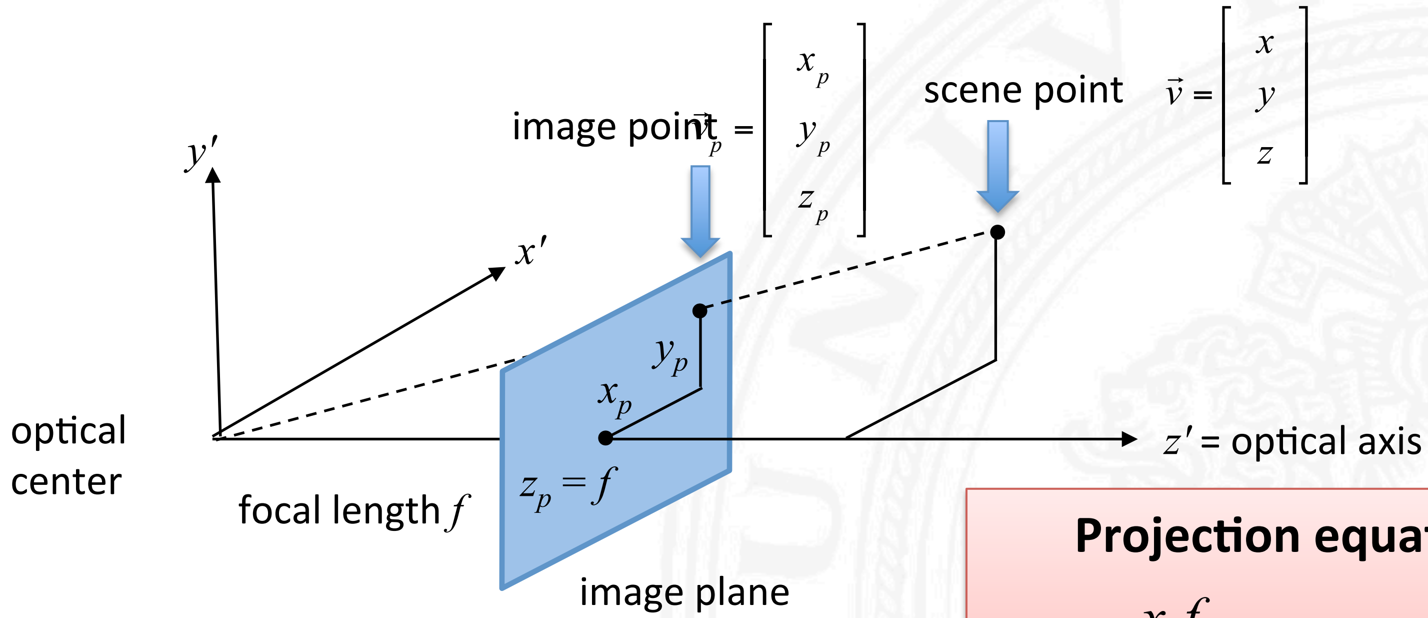
Bandwidth = largest frequency contained in signal
(=> Fourier decomposition of a signal)

Analogous theorem holds for 2D signals with limited spatial frequencies W_x and W_y

Perspective Projection Geometry

Projective geometry relates the coordinates of a point in a scene to the coordinates of its projection onto an image plane.

Perspective projection is an adequate model for most cameras.



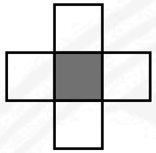
Projection equations:

$$x_p = \frac{x\,f}{z} \qquad y_p = \frac{y\,f}{z}$$

Connectivity in Digital Images

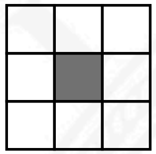
Connectivity is an important property of subsets of pixels. It is based on adjacency (or neighbourhood):

Pixels are 4-neighbours if their distance is $D_4 = 1$



all 4-neighbours of center pixel

Pixels are 8-neighbours if their distance is $D_8 = 1$



all 8-neighbours of center pixel

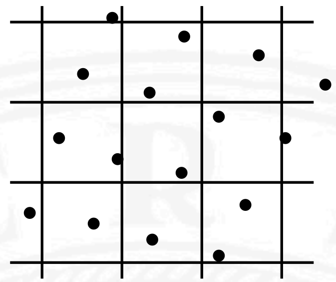
A path from pixel P to pixel Q is a sequence of pixels beginning at Q and ending at P, where consecutive pixels are neighbours.

In a set of pixels, two pixels P and Q are connected, if there is a path between P and Q with pixels belonging to the set.

A region is a set of pixels where each pair of pixels is connected.

Principle of Greyvalue Interpolation

Greyvalue interpolation = computation of unknown greyvalues at locations (u´v´) from known greyvalues at locations (x´y´)



Two ways of viewing interpolation in the context of geometric transformations:

A) Greyvalues at grid locations (x y) in old image are placed at corresponding locations (x´y´) in new image: g(x´y´) = g(T(x y))

→ interpolation in new image

B) Grid locations (u´v´) in new image are transformed into corresponding locations (u v) in old image: g(u v) = g(T^{-1}(u´v´))

→ interpolation in old image

We will take view B:

Compute greyvalues between grid from greyvalues at grid locations.

Global Image Properties

Global image properties refer to an image as a whole rather than components. Computation of global image properties is often required for image enhancement, preceding image analysis.

We treat

- empirical mean and variance
- histograms
- projections
- cross-sections
- frequency spectrum

Median Filter

Median of a distribution $P(x)$: x_m such that $P(x < x_m) = 1/2$

$$\hat{g}_{ij} = \max(a) \text{ with } g_k \in D \quad \text{and} \quad \left|\{g_k < a\}\right| < \frac{|D|}{2}$$

Median Filter:

1. Sort pixels in D according to greyvalue
2. Choose greyvalue in middle position

Example:

11	14	15
13	12	25
15	19	26

11
12
13
14
15
15
19
25
26

greyvalue of center pixel of region is set to 15

Median Filter reduces influence of outliers in either direction!

Discrete Fourier Transform (DFT)

Computes image representation as a sum of sinusoidals.

Discrete Fourier Transform:

$$G_{uv} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{mn} e^{-2\pi j \left(\frac{mu}{M} + \frac{nv}{N}\right)}$$

$$\text{for } u = 0,\ldots,M-1 \text{ and } v = 0,\ldots,N-1$$

Inverse Discrete Fourier-Transform:

$$g_{mn} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \mathrm{G}_{\mathrm{uv}} e^{2\pi j \left(\frac{mu}{M} + \frac{nv}{N}\right)}$$

$$\text{for } \mathrm{m} = 0,\ldots,M-1 \text{ and } \mathrm{n} = 0,\ldots,N-1$$

Notation for computing the Fourier Transform:

$$G_{uv} = F\left\{g_{mn}\right\}$$

$$g_{mn} = F^{-1}\left\{G_{uv}\right\}$$

Transform is based on periodicity assumption!

→ periodic continuation may cause boundary effects

Filtering in the Frequency Domain

A filter transforms a signal by modifying its spectrum.

$$G(u, v) = F(u, v)\, H(u, v)$$

F	Fourier transform of the signal
H	frequency transfer function of the filter
G	modified Fourier transform of signal

Typical filters:

- low-pass filter — *low frequencies pass, high frequencies are attenuated or removed*
- high-pass filter — *high frequencies pass, low frequencies are attenuated or removed*
- band-pass filter — *frequencies within a frequency band pass, other frequencies below or above are attenuated or removed*

Often (but not always) the noise part of an image is high-frequency and the signal part is low-frequency. Low-pass filtering then improves the signal-to-noise ratio.

Discrete Convolution Using the FFT

Convolution in the spatial domain may be performed more efficiently using the FFT.

$$g'_{ij} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{mn} h_{i-m,j-n} \qquad (MN)^2 \text{ operations needed}$$

Using the FFT and filtering in the frequency domain:

g_{mn} —**FFT**→ G_{uv} —**H_{uv}**→ G_{uv}' —**FFT^{-1}**→ g_{mn}'

MN log(MN) | **MN** | **MN log(MN)** | **# of operations**

Example with $M = N = 512$:

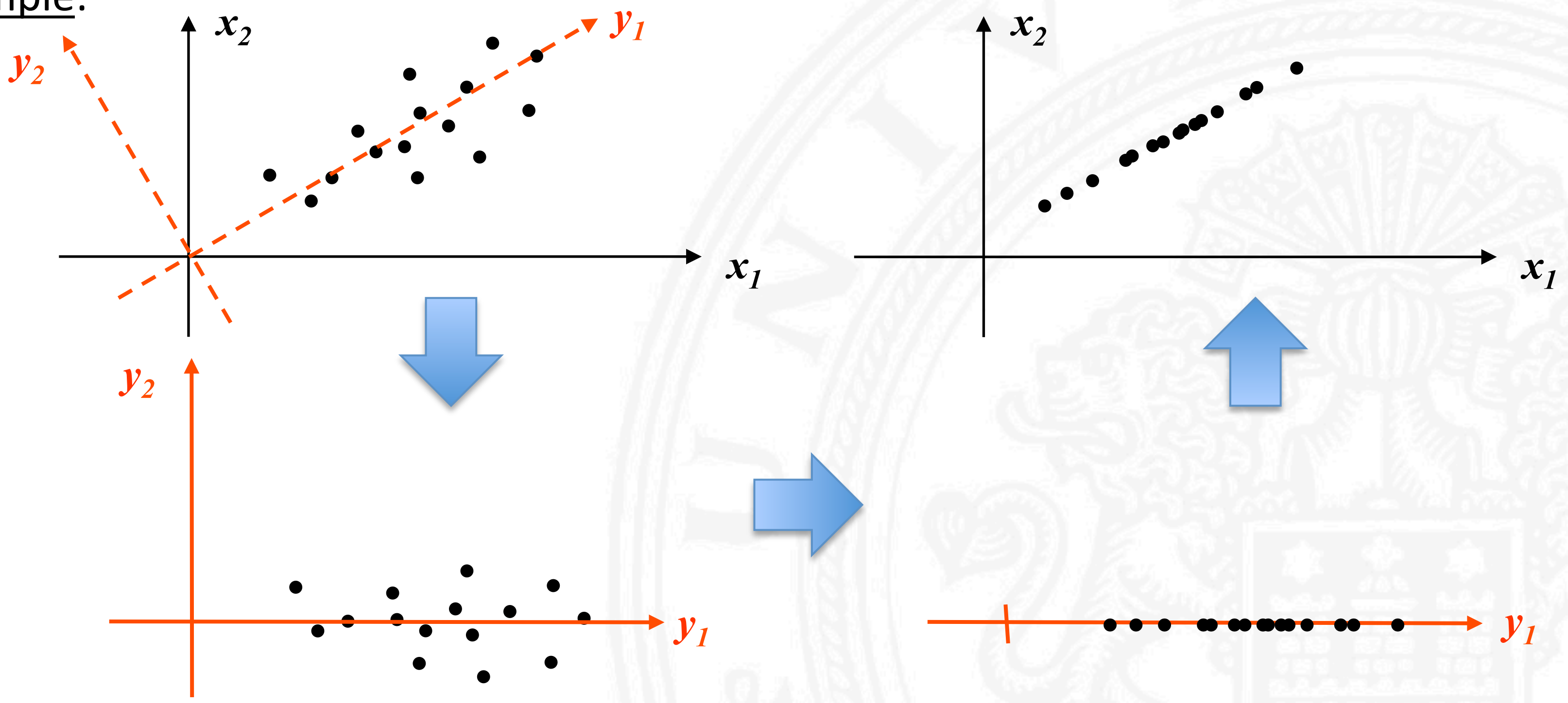
- straight convolution needs ~ 10^{10} operations
- convolution using the FFT needs ~10^7 operations

Illustration of Minimum-loss Dimension Reduction

Using the Karhunen-Loève transform, data compression is achieved by

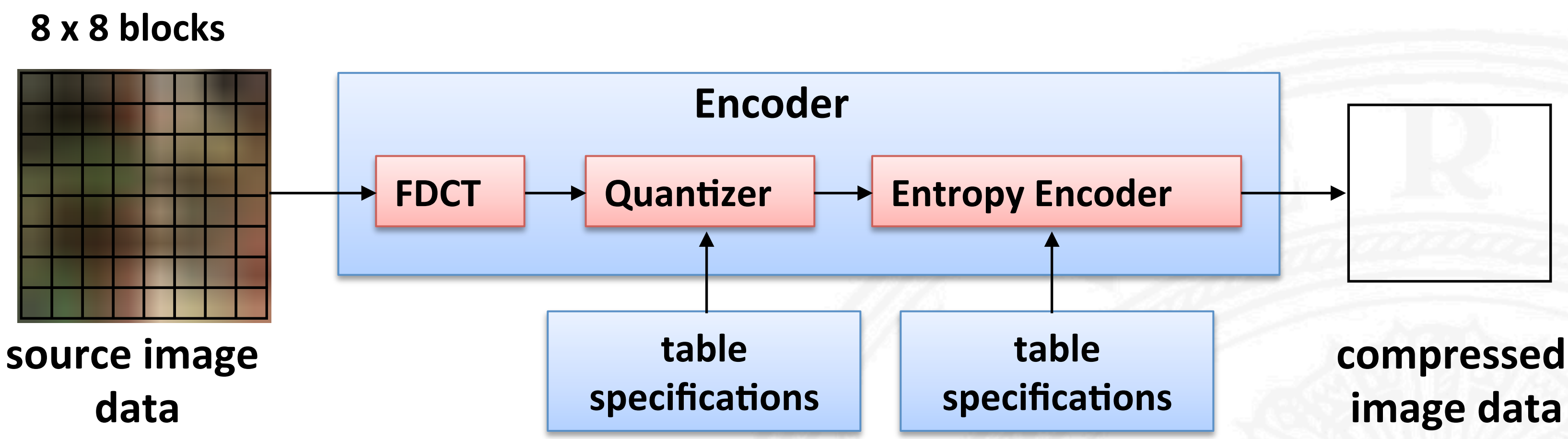
- changing (rotating) the coordinate system
- omitting the least informative dimension(s) in the new coodinate system

<u>Example</u>:



Principle of Baseline JPEG

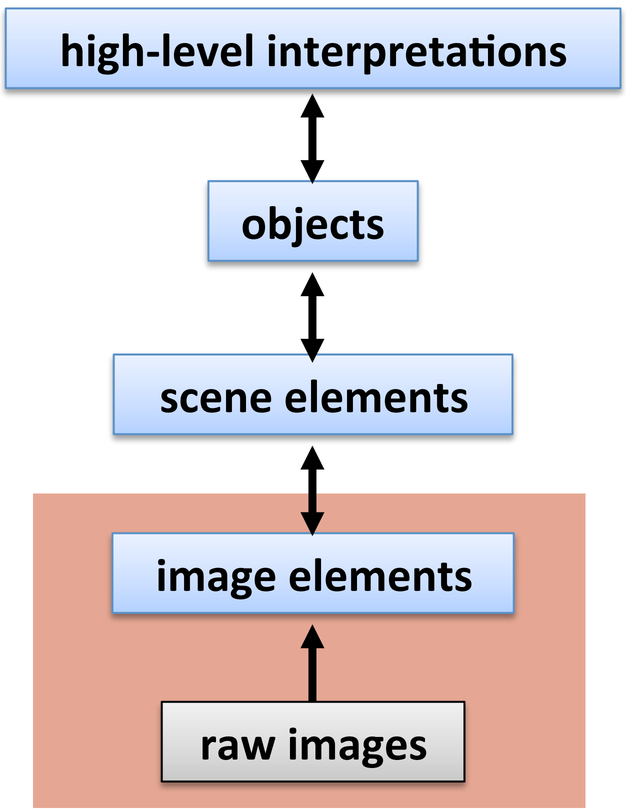
(Source: Gibson et al., Digital Compression for Multimedia, Morgan Kaufmann 98)



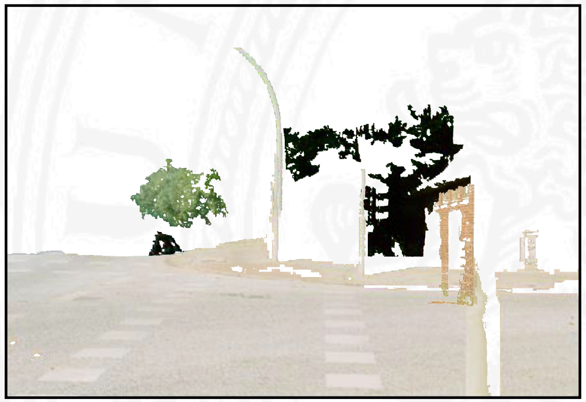
- transform RGB into YUV coding, subsample color information
- partition image into 8 x 8 blocks, left-to-right, top-to-bottom
- compute Discrete Cosine Transform (DCT) of each block
- quantize coefficients according to psychovisual quantization tables
- order DCT coefficients in zigzag order
- perform runlength coding of bitstream of all coefficients of a block
- perform Huffman coding for symbols formed by bit patterns of a block

Segmentation

Segmenting the image into image elements which may correspond to meaningful scene elements



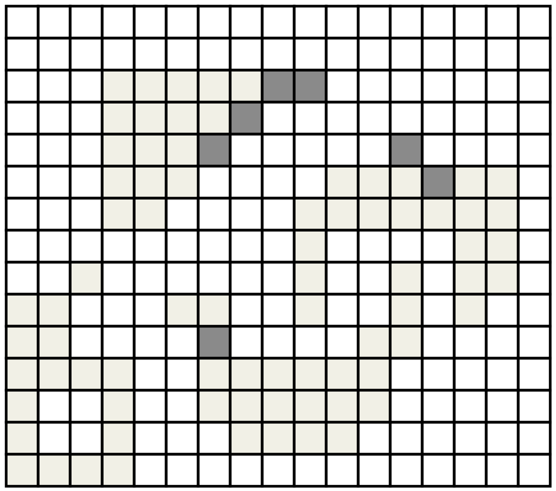
Example:
Partitioning an image into regions which may correspond to objects



Typical results of first segmentation steps

Representing Regions

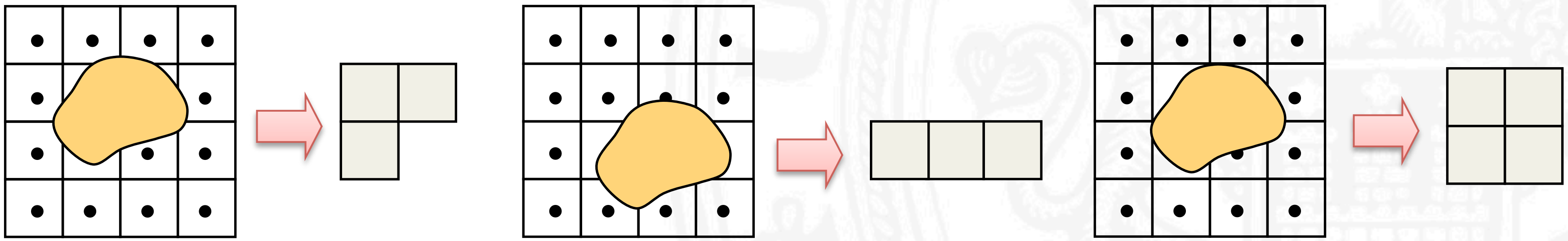
A region is a maximal 4- (or 8-) connected set of pixels.



Methods for digital region representation:

- grid occupancy
  - labelling
  - run-length coding
  - quadtree coding
  - cell sets
- boundary description
  - chain code
  - straight-line segments, polygons
  - higher-order polynomials

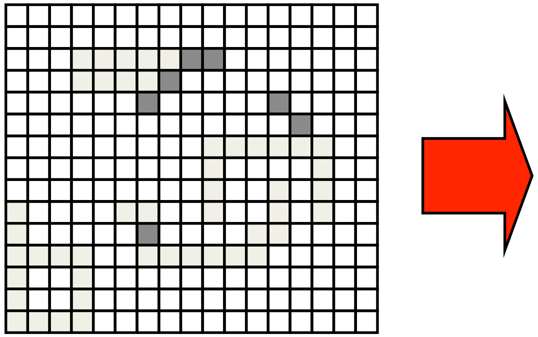
Note that discretizations of an analog region are not shift or rotation invariant:



Component Labelling

In this example: component descriptions using run-length coding

Determining connected regions in B/W images



Component 1
(2 3 9)(3 3 7)(4 6 6)

Component 2
(4 12 12)

Component 3
(5 13 13)(6 9 14)(7 9 9 14 14)(8 9 9 14 14)(9 9 9 14 14)

Component 4
(9 0 0)(10 0 0)(11 0 3)(12 0 0 3 3)(13 0 0 3 3)(14 0 0 3 3)

Component 5
(9 5 6 12 12)(10 6 6 11 12)(11 6 11)

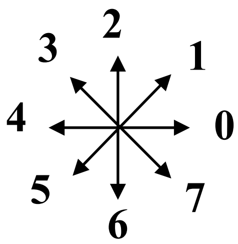
Component labelling of B/W images with 4-neighbourhood

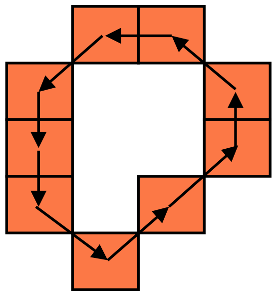
Scan image left to right, top to bottom:
- if pixel is white then continue
- if pixel is black then
  - if left neighbour is white and upper neighbour is white then assign new label
  - if left neighbour is black and upper neighbour is white then assign left label
  - if left neighbour is white and upper neighbour is black then assign upper label
  - if left neighbour is black and upper neighbour is black then
    - assign left label, merge left label and upper label

Chain Code

Chain code represents boundaries by "chaining" direction arrows between successive boundary elements.

Chain code for <u>8-connectivity</u>:





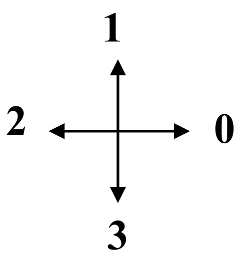
Arbitrary choice of starting point, chain code can be represented e.g. by

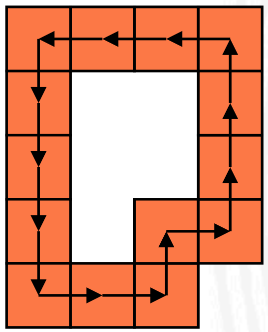
{456671123}

Normalization by circular shift until the smallest integer is obtained:

{112345667}

Chain code for <u>4-connectivity</u>:





Arbitrary starting point:

{22233330010111}

Normalized:

{00101112223333}

Canny Edge Detector I

Optimal edge detector for step edges corrupted by white noise.

Optimality criteria:

- **<u>Detection</u> of all important edges and no spurious responses**
- **Minimal distance between <u>location</u> of edge and actual edge**
- **<u>One response</u> per edge only**

1. Derivation for 1D results in edge detection filter which can be effectively approximated (< 20% error) by the 1st derivative of a Gaussian smoothing filter.
2. Generalization to 2D requires estimation of edge orientation:

$$\vec{n} = \frac{\nabla(f * g)}{|\nabla(f * g)|}$$

$\vec{n}$ normal perpendicular to edge
f Gaussian smoothing filter
g greyvalue image

Edge is located at local maximum of g convolved with f in direction $\vec{n}$:

$$\frac{\partial^2}{\partial \vec{n}^2} f * g = 0$$ "non-maximal suppression"

Canny Edge Detector II

Algorithm includes

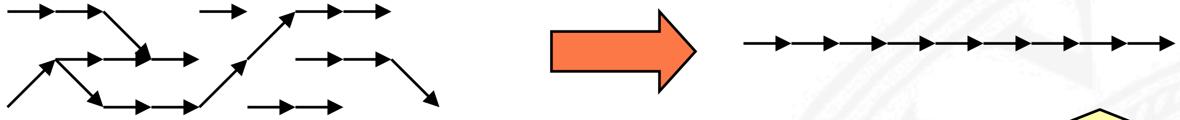
- choice of scale σ
- hysteresis thresholding to avoid streaking (breaking up edges)
- "feature synthesis" by selecting large-scale edges dependent on lower-scale support

1. Convolve image g with Gaussian filter f of scale σ
2. Estimate local edge normal direction γ for each point in the image
3. Find edge locations using non-maximal suppression
4. Compute magnitude of edges
5. Threshold edges with hysteresis to eliminate spurious edges
6. Repeat steps (1) through (5) for increasing values of σ
7. Aggregate edges at multiple scales using feature synthesis

Grouping

To make sense of image elements, they first have to be grouped into larger structures.

Example: Grouping noisy edge elements into a straight edge

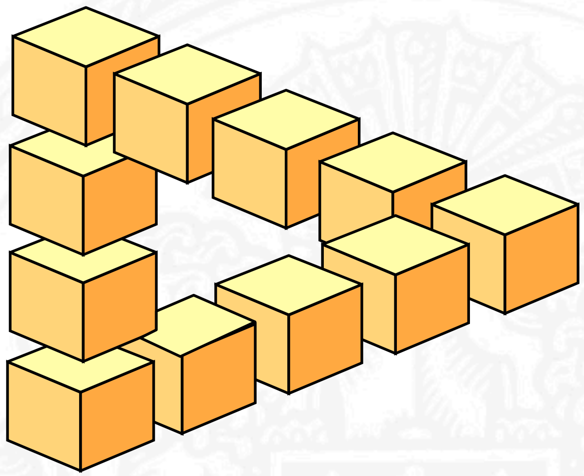


Essential problem:

Obtaining globally valid results by local decisions

Important methods:

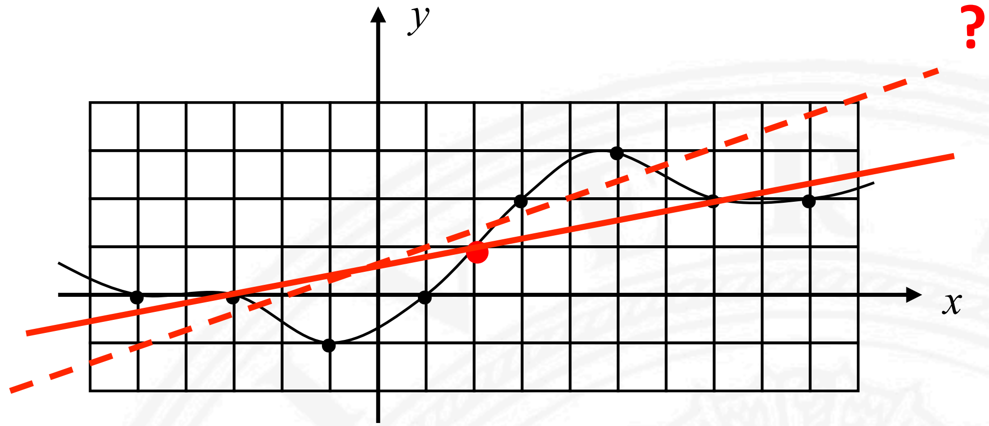
- Fitting
- Clustering
- Hough Transform
- Relaxation



- locally compatible
- globally incompatible

Example for Straight Line Fitting by Eigenvector Analysis

What is the best straight-line approximation of the contour?



Given points: { (-5 0) (-3 0) (-1 -1) (1 0) (3 2) (5 3) (7 2) (9 2) }

Center of gravity: $m_x = 2 \;\; m_y = 1$

Scatter matrix: $S_{11} = 168$, $S_{12} = S_{21} = 38$, $S_{22} = 14$

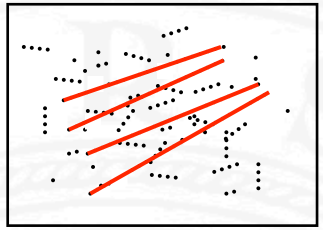
Eigenvalues: $\lambda_1 = 176.87$, $\lambda_2 = 5.13$

Direction of straight line: $r_y/r_x = 0.23$

Straight line equation: $y = 0.23\,x + 0.54$

Hough Transform I

Robust method for fitting straight lines, circles or other geometric figures which can be described analytically.

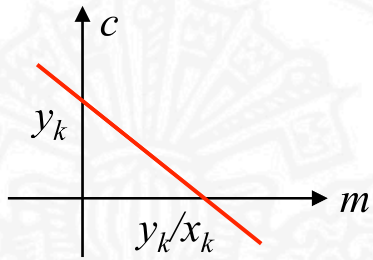


Given: Edge points in an image

Wanted: Straight lines supported by the edge points

An edge point (x_k, y_k) supports all straight lines $y = mx + c$ with parameters m and c such that $y_k = mx_k + c$.

The locus of the parameter combinations for straight lines through (x_k, y_k) is a straight line in parameter space.



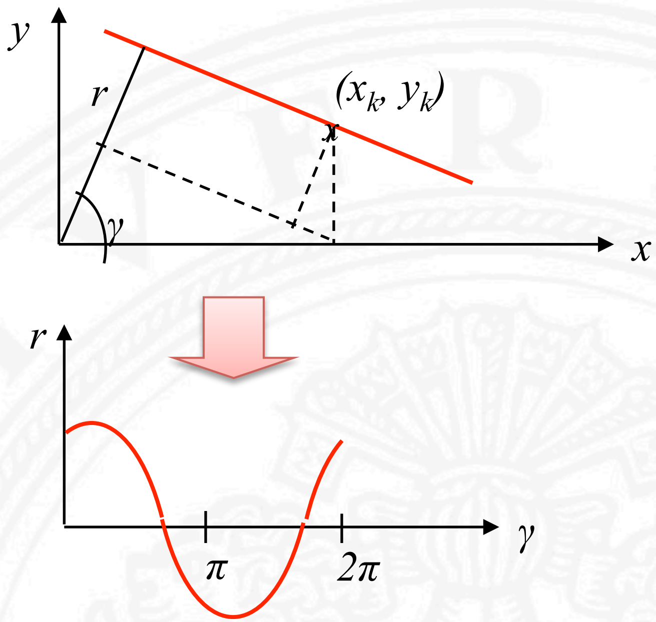
Principle of Hough transform for straight line fitting:

- Provide accumulator array for quantized straight line parameter combinations
- For each edge point, increase accumulator cells for all parameter combinations supported by the edge point
- Maxima in accumulator array correspond to straight lines in the image

Hough Transform II

For straight line finding, the parameter pair (r, γ) is commonly used because it avoids infinite parameter values:

$$x_k cos(\gamma) + y_k sin(\gamma) = r$$



Each edge point (x_k, y_k) corresponds to a sinusoidal in parameter space:

Important improvement by exploiting direction information at edge points:

(x_k, y_k, φ) ⇒ $x_k cos(\gamma) + y_k sin(\gamma) = r$ restricted to $\varphi - \delta \leq \gamma \leq \varphi + \delta$

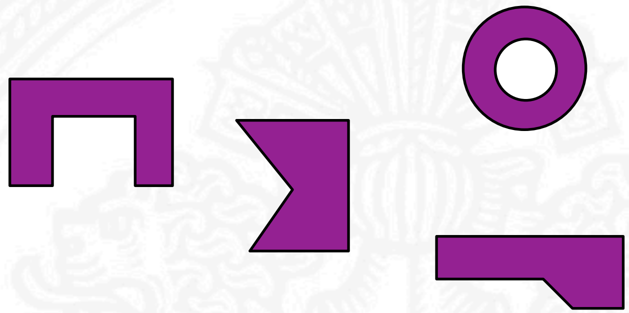
gradient direction (↑ φ)

direction tolerance (↑ δ)

Simple 2D Shape Features

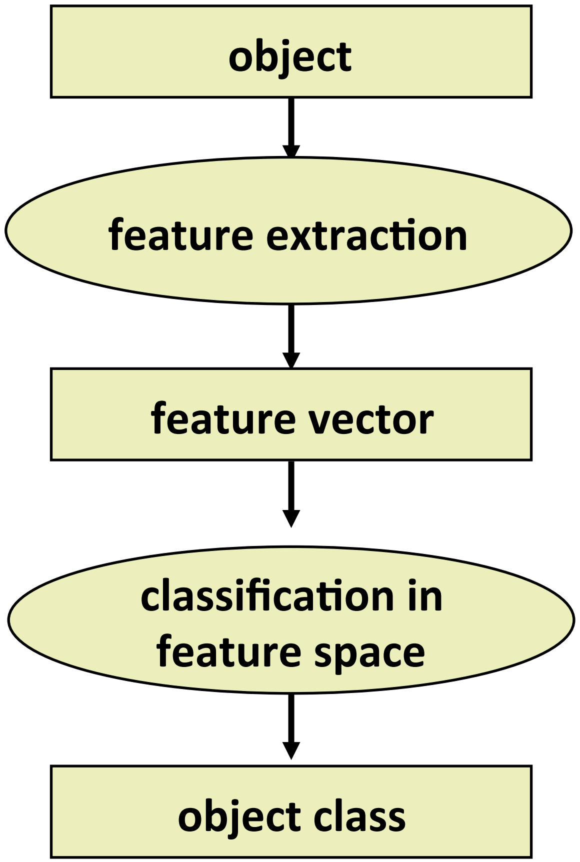
For industrial recognition tasks it is often required to distinguish
- **a small number of different shapes**
- **viewed from a small number of different view points**
- **with a small computational effort.**

In such cases simple 2D shape features may be useful, such as:
- area
- boxing rectangle
- boundary length
- compactness
- second-order momentums
- polar signature
- templates



Features may or may not have invariance properties:
- 2D translation invariance
- 2D rotation invariance
- scale invariance

Basic Terminology for Pattern Recognition



K	classes $\omega_1 \dots \omega_K$
N	dimension of feature space
$\vec{x}^T = (x_1 \; x_2 \; \dots \; x_N)$	feature vector
$\vec{y}^T = (y_1 \; y_2 \; \dots \; y_N)$	prototype (feature vector with known class membership)
$\vec{y}_i^{(k)}$	i-th prototyp of class k
M_k	number of prototypes for class k
$g_k(\vec{x})$	discriminant function for class k

Problem: Determine $g_k(\vec{x})$ such that

$$\underset{\vec{x}\in\omega_k}{\forall} \; \underset{k\neq j}{\forall} \; g_k(\vec{x}) > g_j(\vec{x})$$

Perceptron Learning Rule

A solution vector a can be determined iteratively by minimizing a criterion function $J(\vec{a})$ by gradient descent.

Perceptron criterion function:

$$J_p(\vec{a}) = \sum_{y \in B} \left(-\vec{a}^T \vec{y}\right)$$

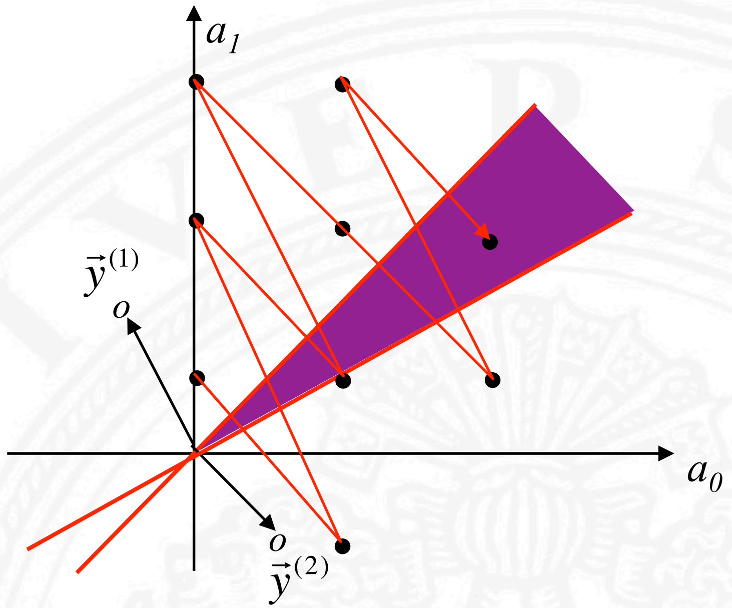
with B = {all misclassified prototypes}

Basic gradient descent algorithm:

Gradient: $$\nabla J_p(\vec{a}) = \sum_{y \in B} (-\vec{y})$$

Step: $$\vec{a}_{k+1} = \vec{a}_k + \rho_k \sum_{y \in B} (\vec{y})$$

Weight vector $\vec{a}$ is modified in negative gradient direction!



iterations viewed in weight space

Example (see illustration) with:

$\vec{y}_1 = (-1 \;\; 2)^T$, $\vec{y}_2 = (-1 \;\; 1)^T$, $\rho = 2$

k	0	1	2	3	4	5	6	7	8
$\vec{a}_k$	0 1	2 -1	0 3	2 1	0 5	2 3	4 1	2 5	4 3

solution

Statistical Decision Theory

Generating decision functions from a statistical characterization of classes (as opposed to a characterization by prototypes)

Advantages:

1. The classification scheme may be designed to satisfy an objective optimality criterion:
   Optimal decisions minimize the probability of error.
2. Statistical descriptions may be much more compact than a collection of prototypes.
3. Some phenomena may only be adequately described using statistics, e.g. noise.

General Framework for Bayes Classification

Statistical decision theory minimizes the probability of error for classifications based on uncertain evidence

$\omega_1 \dots \omega_K$	K classes
$P(\omega_k)$	prior probability that an object of class k will be observed
$\vec{x}^T = (x_1 \dots x_N)$	N-dimensional feature vector of an object
$p(\vec{x} \mid \omega_k)$	conditional probability ("likelihood") of observing $\vec{x}$ given that the object belongs to class ω_K
$P(\omega_k \mid \vec{x})$	conditional probability ("posterior probability") that an object belongs to class ω_K given $\vec{x}$ is observed

Bayes decision rule:

Classify given evidence $\vec{x}$ as class ω' such that ω' minimizes the probability of error $P(\omega \neq \omega' \mid \vec{x})$

→ Choose ω' which maximizes the posterior probability $P(\omega \mid \vec{x})$

$g_i(\vec{x}) = P(\omega_i \mid \vec{x})$ are discriminant functions.

Motion Analysis

Motion analysis of digital images is based on a temporal sequence of image frames of a coherent scene.

"sparse sequence"	→	few frames, temporally spaced apart, considerable differences between frames
"dense sequence"	→	many frames, incremental time steps, incremental differences between frames
video	→	50 half frames per sec, interleaving, line-by-line sampling

Motion detection

Register locations in an image sequence which have changed due to motion

Moving object detection and tracking

Detect individual moving objects, determine and predict object trajectories, track objects with a moving camera

Derivation of 3D object properties

Determine 3D object shape from multiple views ("shape from motion")

Optical Flow Constraint Equation

Optical flow is the displacement field of surface elements of a scene during an incremental time interval dt ("velocity field").

Assumptions:

- Observed brightness is constant over time (no illumination changes)
- Nearby image points move similarly (velocity smoothness constraint)

For a continuous image $g(x, y, t)$ a linear Taylor series approximation gives

$$g(x+dx,\ y+dy,\ t+dt)\ \approx\ g(x,\ y,\ t)\ +\ g_x dx\ +\ g_y dy\ +\ g_t dt\ =\ 0$$ with: $\nabla^T g = \begin{pmatrix} g_x & g_y & g_t \end{pmatrix}$

For motion without illumination change we have

$$g(x+dx,\ y+dy,\ t+dt)\ =\ g(x,\ y,\ t)$$

Hence $\frac{dx}{dt} g_x + \frac{dy}{dt} g_y = g_x u + g_y v = -g_t$

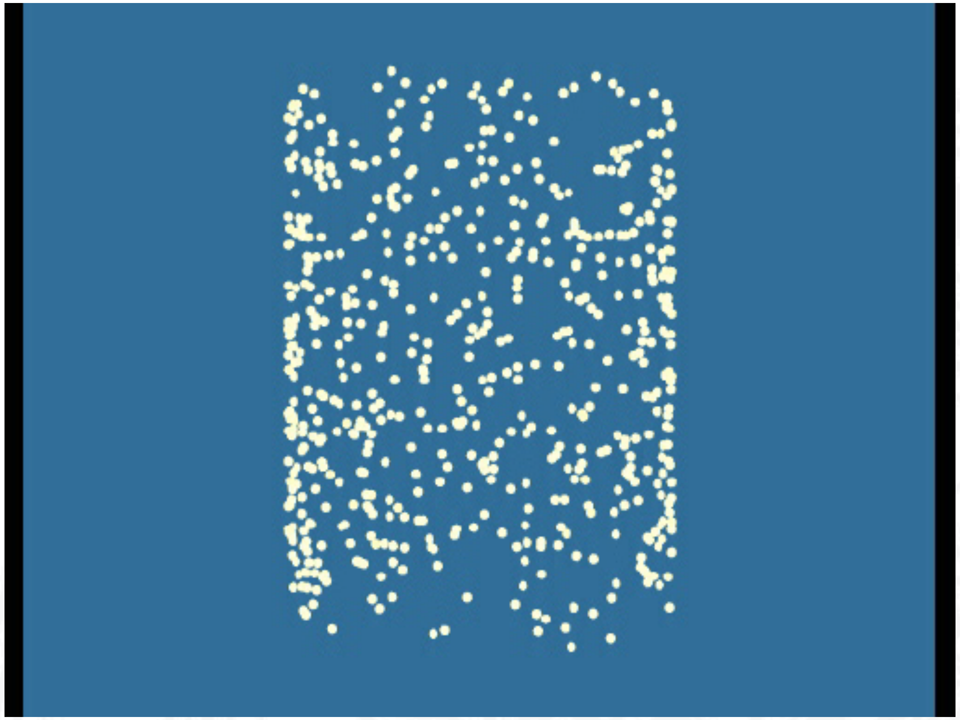
Optical Flow Constraint Equation (OFCE)
with: u, v velocity components

$g_x \approx \Delta g / \Delta x,\ \ g_y \approx \Delta g / \Delta y,\ \ g_t \approx \Delta g / \Delta t$ may be estimated from the spatial and temporal surround of a location *(x, y)*, hence the optical flow constraint equation provides **one** equation for the **two** unknowns u and v.

3D Motion Analysis Based on 2D Point Displacements

2D displacements of points observed on an unknown moving rigid body may provide information about

- the 3D structure of the points
- the 3D motion parameters



Rotating cylinder experiment by S. Ullman (1981)

Cases of interest:

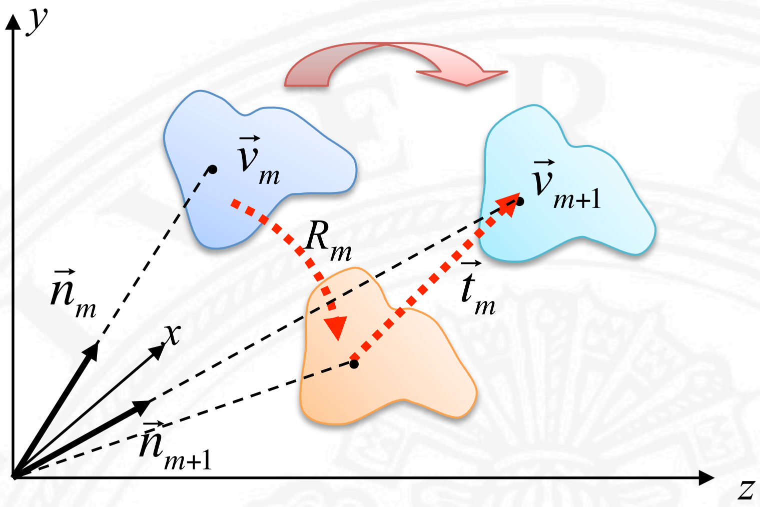
- stationary camera, moving object(s)
- moving camera, stationary object(s)
- moving camera, moving object(s)

camera motion parameters may be known

Essential Matrix

Geometrical constraints derived from 2 views of a point in motion

- motion between image m and $m+1$ may be decomposed into
  1. rotation R_m about origin of coordinate system (= optical center)
  2. translation $\vec{t}_m$
- observations are given by direction vectors $\vec{n}_m$ and $\vec{n}_{m+1}$ along projection rays.



$R_m\vec{n}_m$, $\vec{t}_m$ and $\vec{n}_{m+1}$ are coplanar: $\left(\vec{t}_m \times R_m\vec{n}_m\right)^T \vec{n}_{m+1} = 0$

After some manipulation: $\vec{n}_m E_m n_{m+1} = 0 \quad E =$ essential matrix

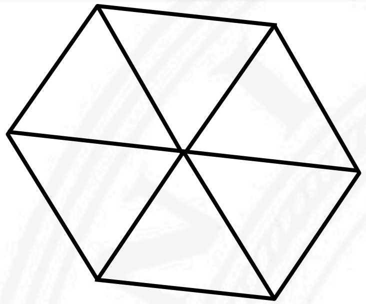
with $E_m = \begin{pmatrix} | & | & | \\ \vec{t}_m \times \vec{r}_1 & \vec{t}_m \times \vec{r}_2 & \vec{t}_m \times \vec{r}_3 \\ | & | & | \end{pmatrix}$ and $R_m = \begin{pmatrix} | & | & | \\ \vec{r}_1 & \vec{r}_2 & \vec{r}_3 \\ | & | & | \end{pmatrix}$

Generality Assumption

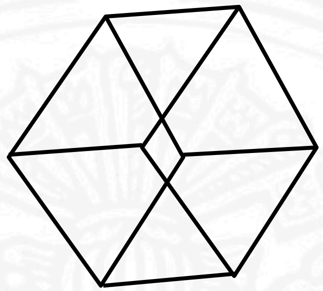
Assume that
- **viewpoint**
- **illumination**
- **physical surface properties**

are all general, i.e. do not produce coincidental structures in the image.

Example:
Do not interpret this figure as a 3D wireframe cube, because this view is not general.



General view:



The generality assumption is the basis for several specialized interpretation methods, e.g.

- shape from texture
- shape from shading

...

- "shape from X"

3D Line Orientation from Vanishing Points

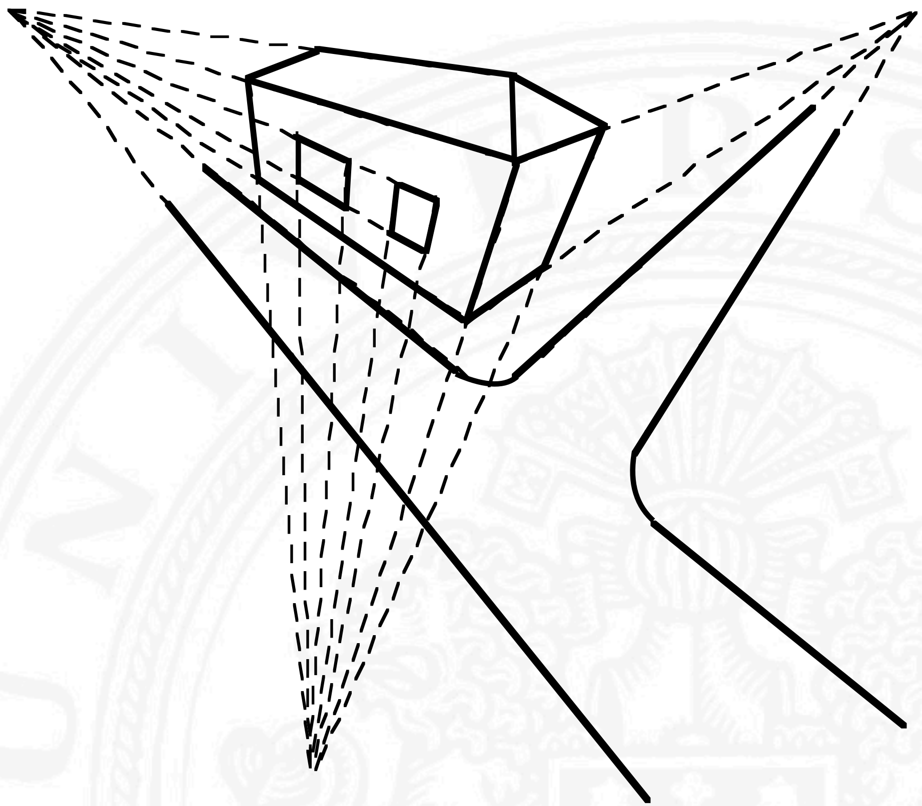
From the laws of perspective projection:

The projections of 3D parallel straight lines intersect in a single point, the vanishing point.

Assume that more than 2 straight lines do not intersect in a single point by coincidence

⇓

If more than 2 straight lines intersect, assume that they are parallel in 3D



Principle of Shape from Shading

See "Shape from Shading" (B.K.P. Horn, M.J. Brooks, eds.), MIT Press 1989

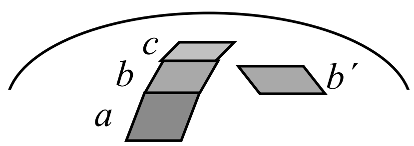
Physical surface properties, surface orientation, illumination and viewing direction determine the greyvalue of a surface patch in a sensor signal.

For a single object surface viewed in one image, greyvalue changes are mainly caused by surface **orientation changes**.

The reconstruction of **arbitrary** surface shapes is not possible because different surface orientations may give rise to identical greyvalues.

Surface shapes may be uniquely reconstructed from shading information if possible surface shapes are constrained by **smoothness assumptions**.

Principle of incremental procedure for surface shape reconstruction:



a: patch with known orientation

b, c: neighbouring patches with similar orientations

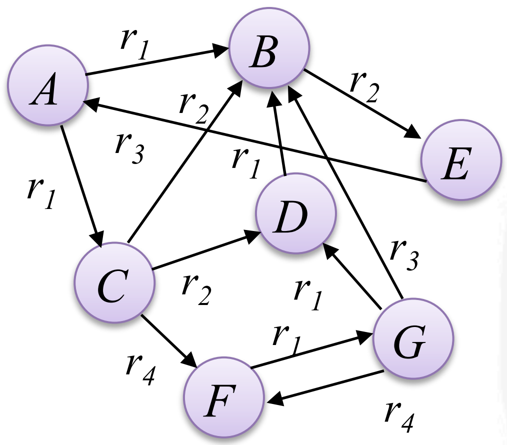
b´: radical different orientation may not be neighbour of a

Object Recognition by Relational Matching

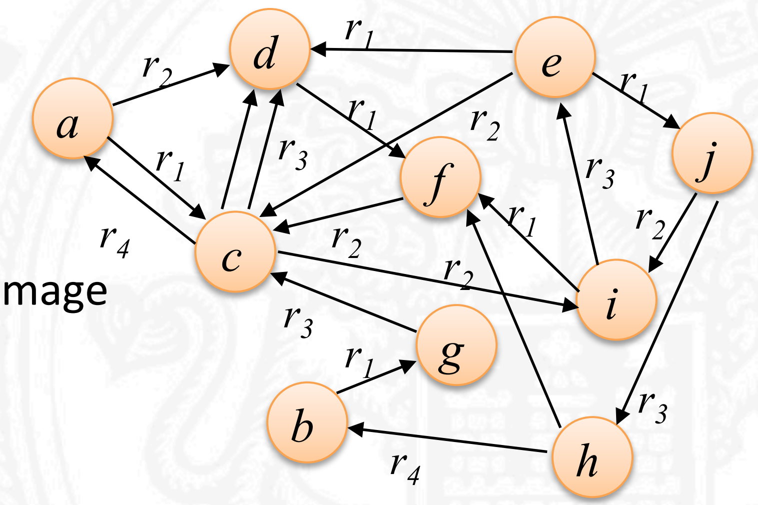
<u>Principle</u>:

- construct relational model(s) for object class(es)
- construct relational image description
- compute R-morphism (best partial match) between image and model(s)
- top-down verification with extended model

model



image



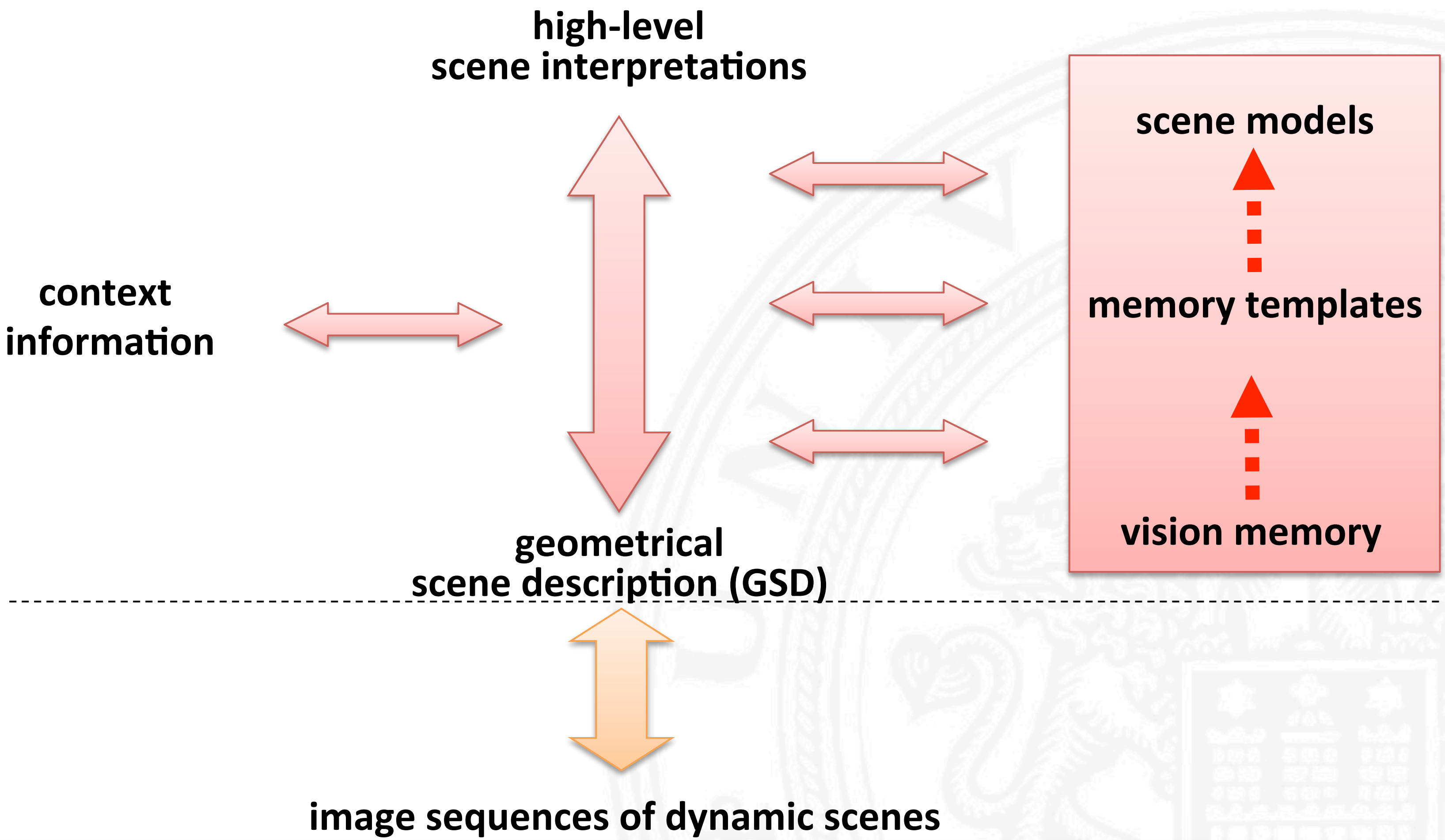
SIFT Method

David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints
International Journal of Computer Vision, 2004 (Protected by US patent)

Lowe developed specific methods for:

1. Determining invariant local descriptors at interest points
   - finding stable interest points ("keypoints")
   - computing largely scale-invariant features at interest points
2. Extracting stable descriptors for object models
3. Finding and recognizing objects based on local descriptors

Basic Building Blocks for High-level Scene Interpretation



Occurrence Models

Basic ingredients:

- relational structure
- taxonomy
- partonomy
- spatial relational language
- temporal relational language
- object appearance models

- An occurrence model describes a class of occurrences by:
  - properties
  - sub-occurrences (= components of the occurrence)
  - relations between sub-occurrences
- A primitive occurrence model consists of
  - properties
  - a qualitative predicate
- Each occurrence has a begin and end time point

Basic Interpretation Algorithm

```
Enter context information
Repeat
        Check for goal completion
        Check for new evidence
        Determine possible interpretation steps and update agenda
        Select from agenda one of
                { evidence matching,
                  aggregate instantiation,
                  aggregate expansion,
                  instance specialization,
                  parameterization,
                  constraint propagation }
        Check for conflict
end
```

Conflict = unsatisfiable constraint net

→need for backtracking or parallel alternative threads