

Einführung in die Erweiterte Realität

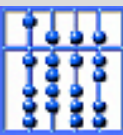
8. Basic Computer Vision

Prof. Gudrun Klinker, Ph.D.

Institut für Informatik, Technische Universität München

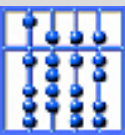
klinker@in.tum.de

Dec 21, 2004



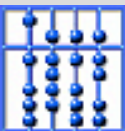
Reference

R. Jain, R. Kasturi, and B. Schunk.
Machine Vision,
McGraw Hill, 1995, ISBN 0-07-032018-7



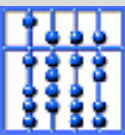
Overview

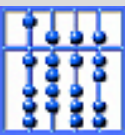
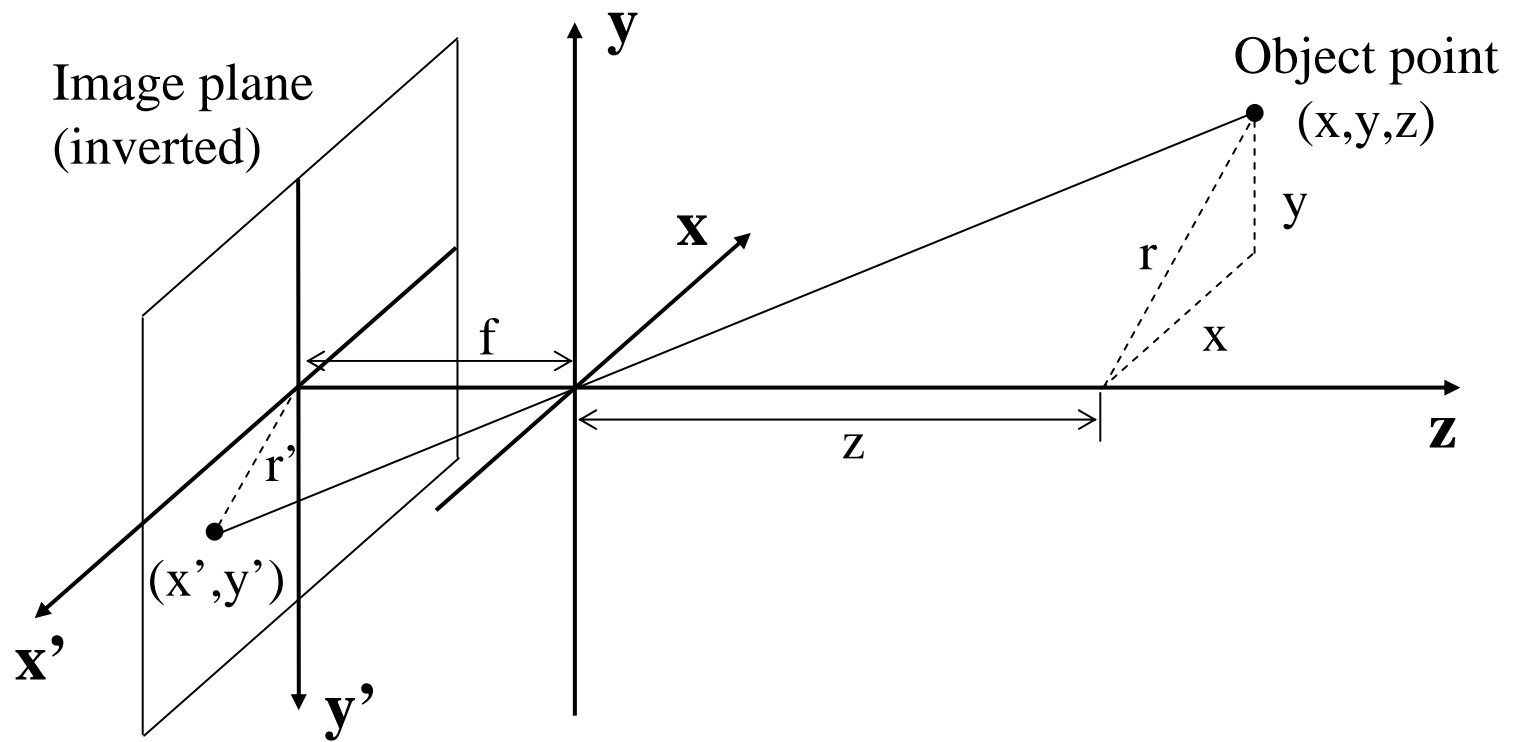
- Image Formation (Geometry, Radiometry)
- Feature Detection
 - Region(Similarity)-based Approaches
 - Edge(Difference)-based Approaches
- Feature Matching and Tracking



1. Image Formation

- Image geometry
- (Image radiometry)





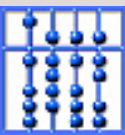
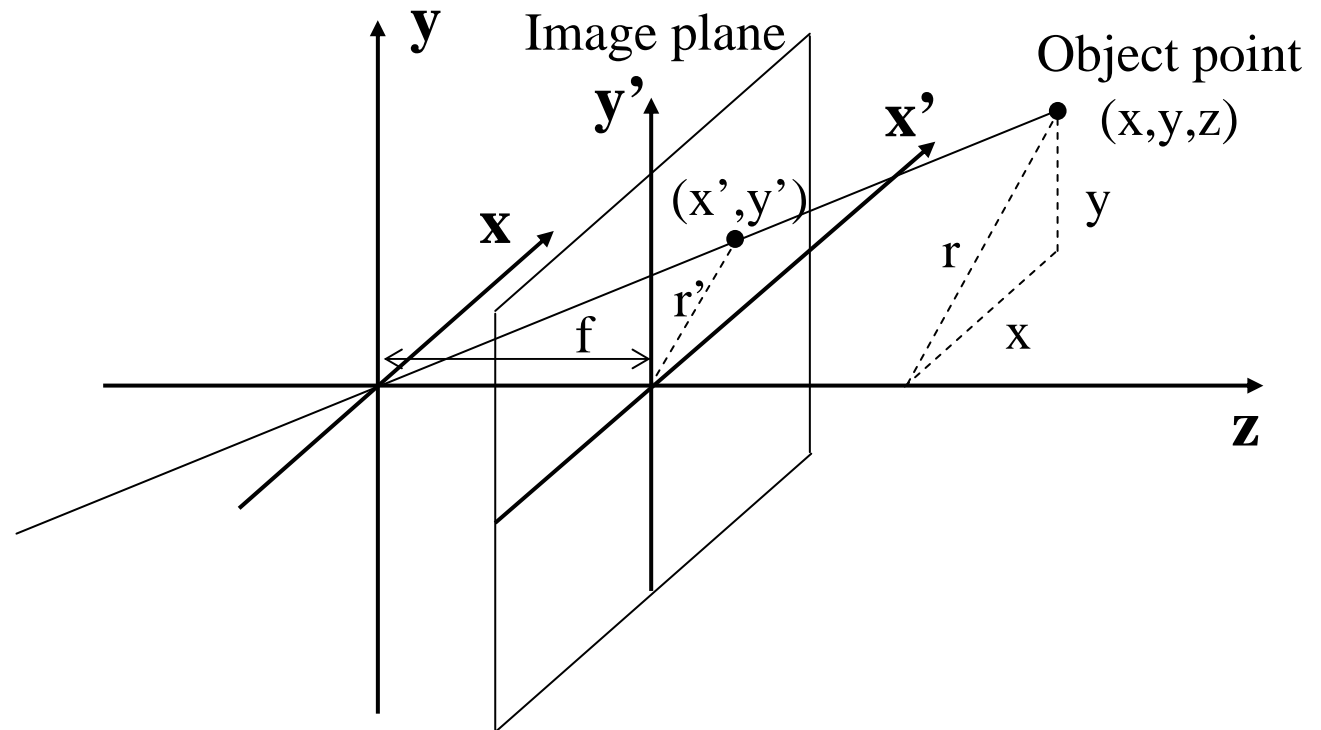
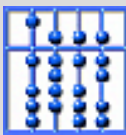
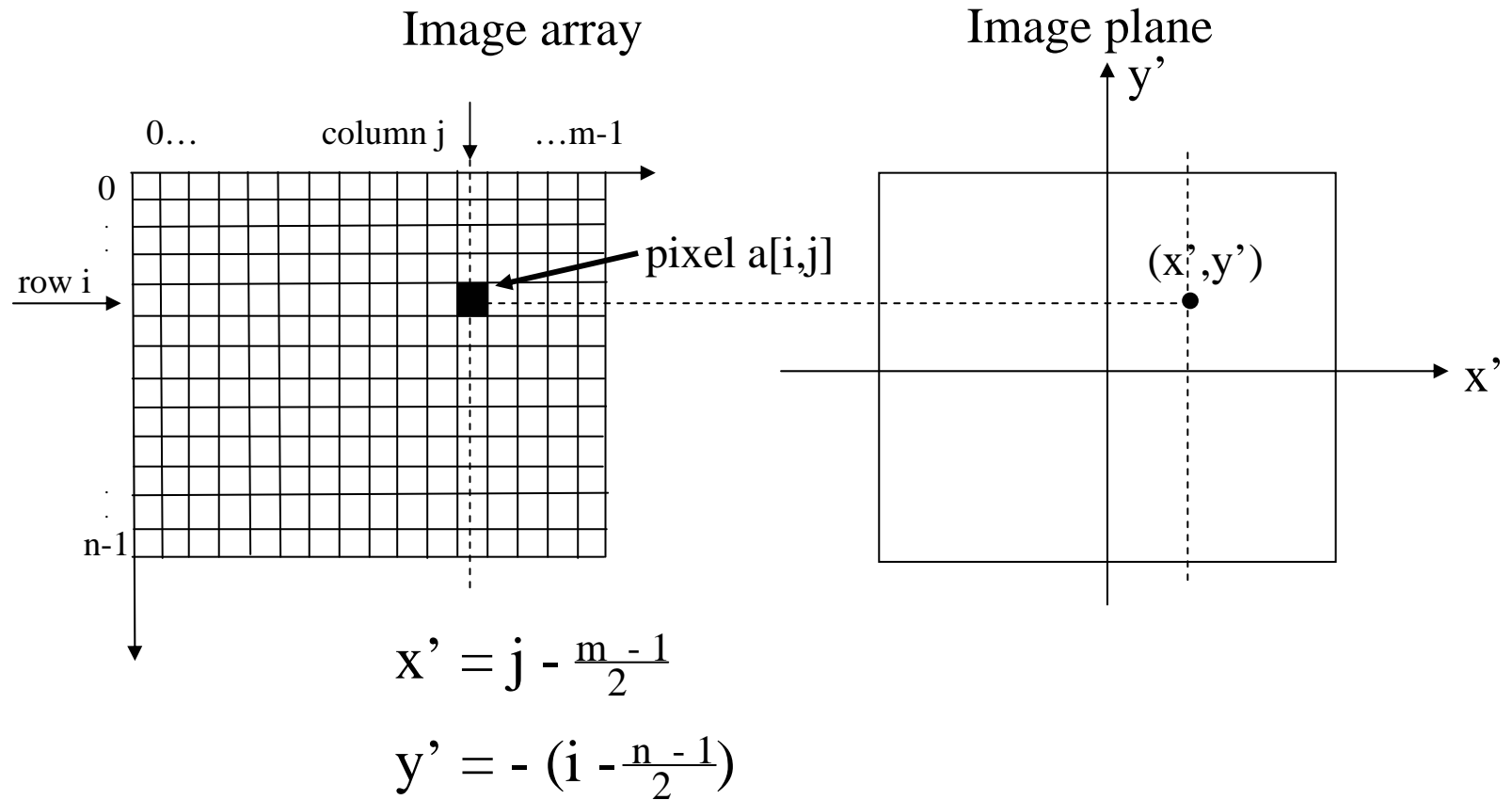
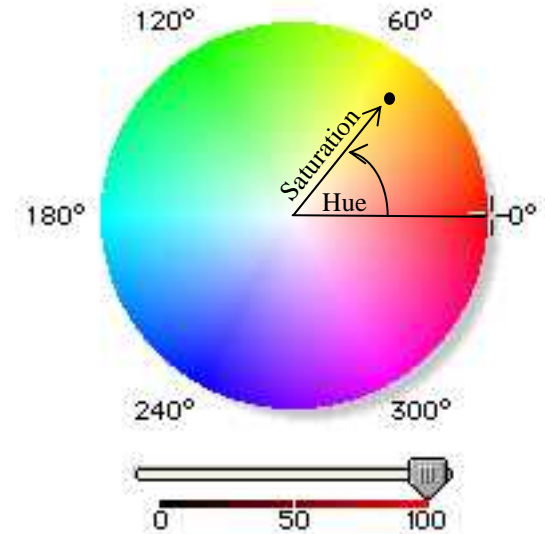


Image Geometry



Color pixel $[i,j] = \{R,G,B\}$ or $\{R,G,B,\alpha\}$

Grayscale pixel $[i,j] = \{I\}$



HSI color representation

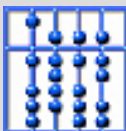
$$\cos(\text{Hue}) = (2R - G - B) / (2\sqrt{(R-G)^2 + (R-G)(G-B)})$$

$$\text{Saturation} = 1 - 3 / (R+G+B) * \min(R,G,B)$$

$$\text{Intensity} = (R+G+B)/3$$

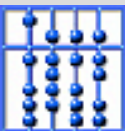
Common practice (for fast processing)

$$I = R \quad \text{or} \quad I = G$$



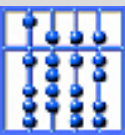
Overview

- Image Formation (Geometry, Radiometry)
- Feature Detection
 - Region(Similarity)-based Approaches
 - Edge(Difference)-based Approaches
- Feature Matching and Tracking



Overview: Region-based Approaches

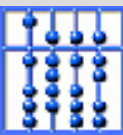
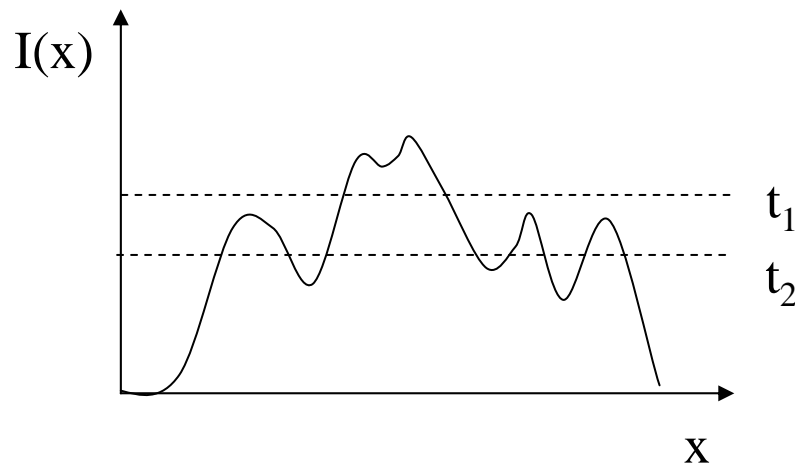
- Thresholding, Histograms
- Connected Components
- Region Boundaries
- Region Properties
- Morphological Operations



2. Image Thresholding

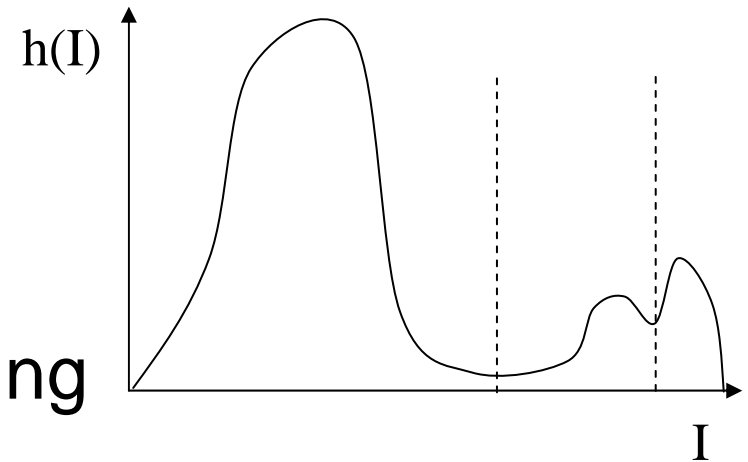
- Thresholding
- Lookup tables

$$f_B[i,j] = \begin{cases} 0, & \text{if } f_A[i,j] < t \\ 1, & \text{if } f_A[i,j] > t \end{cases}$$
$$f_B[i,j] = \text{lookup}[f_A[i,j]]$$

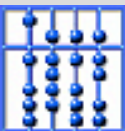


2.1 Histogram-based Region Segmentation

- P-tile method
- Mode method
- Iterative threshold selection
- Adaptive, variable thresholding
- Double thresholding

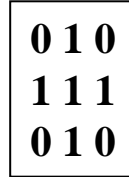


```
0 0 0 1 1 0 0 1 0
0 0 0 1 1 0 1 1 0
1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 0 0 0
0 0 0 1 1 0 1 1 0
```

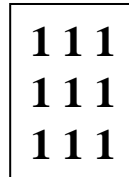


2.2 Connected Component Analysis -Pixel Neighborhoods-

- 4-neighbors



- 8-neighbors

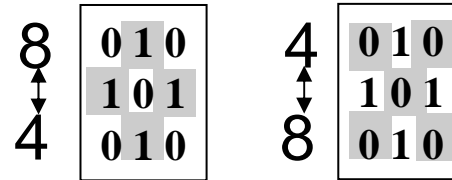


- 4-path, 8-path

- Connected components

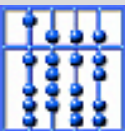
- Background and holes

- Boundary



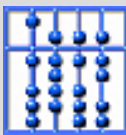
```

000110010
000110110
111111110
011111000
000110110
    
```



- Two-pass approach progressing in row-major order
- Recursive region growing
 - find seed
 - check each neighbor that hasn't been visited yet
 - if neighbor is inside the region,
 - label it
 - recurse

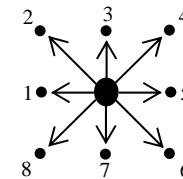
```
0 0 0 1 1 0 0 1 0
0 0 0 1 1 0 1 1 0
1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 0 0 0
0 0 0 1 1 0 2 2 0
```



2.3 Region Boundary Detection

- Boundary pixel:
A pixel that has 4-neighbors that don't belong to the region.
- Crab algorithm to follow a region boundary:
 - Find starting boundary pixel
 - Set starting search direction
 - Find next boundary pixel
 - Reset starting search direction
- Shape fitting (ellipsoid, polygon, ...)

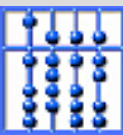
0	0	0	1	1	0	0	1	0
0	0	0	1	1	0	1	1	0
1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	0	0	0
0	0	0	1	1	0	2	2	0



2.4 Region Properties

- Area A and perimeter P
- Compactness
Relationship between A and P
- Boundary (Chain Code)

2 3 4
1 x 5
8 7 6



- Size (zeroth-order moment)

$$A = \sum \sum B[i,j]$$

- Position (first-order moments)

$$m_i = \sum \sum i B[i,j] / A$$

$$m_j = \sum \sum j B[i,j] / A$$

- Orientation (second-order moments)

$$a = \sum \sum (i - m_i)^2 B[i,j]$$

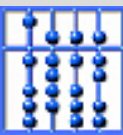
$$b = 2 \sum \sum (i - m_i)(j - m_j) B[i,j]$$

$$c = \sum \sum (j - m_j)^2 B[i,j]$$

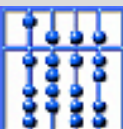
$$\tan 2\theta = b / (a - c)$$

```

000110010
000110110
111111110
011110000
000110110
    
```

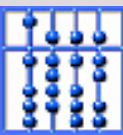


- Photometric Properties of a Region
 - Mean intensity (color)
 - Intensity variation
(color variation: 3 eigenvectors)
 - Repetitive patterns (textures)



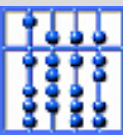
2.5 Morphological Operators on Region Masks

- Intersection of two masks
- Union of two masks
- Complement of a mask
- **Dilation** of a mask by a structuring element (simple case: region expansion)
- **Erosion** (simple case: region shrinking)
- **Opening**: erosion + dilation
- **Closing**: dilation + erosion



Overview

- Image Formation (Geometry, Radiometry)
- **Feature Detection**
 - Region(Similarity)-based Approaches
 - **Edge(Difference)-based Approaches**
- Feature Matching and Tracking



3. Edge Detection

- Maximal/Minimal Image Gradient

(Optima in the 1st derivative)

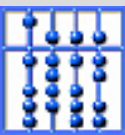
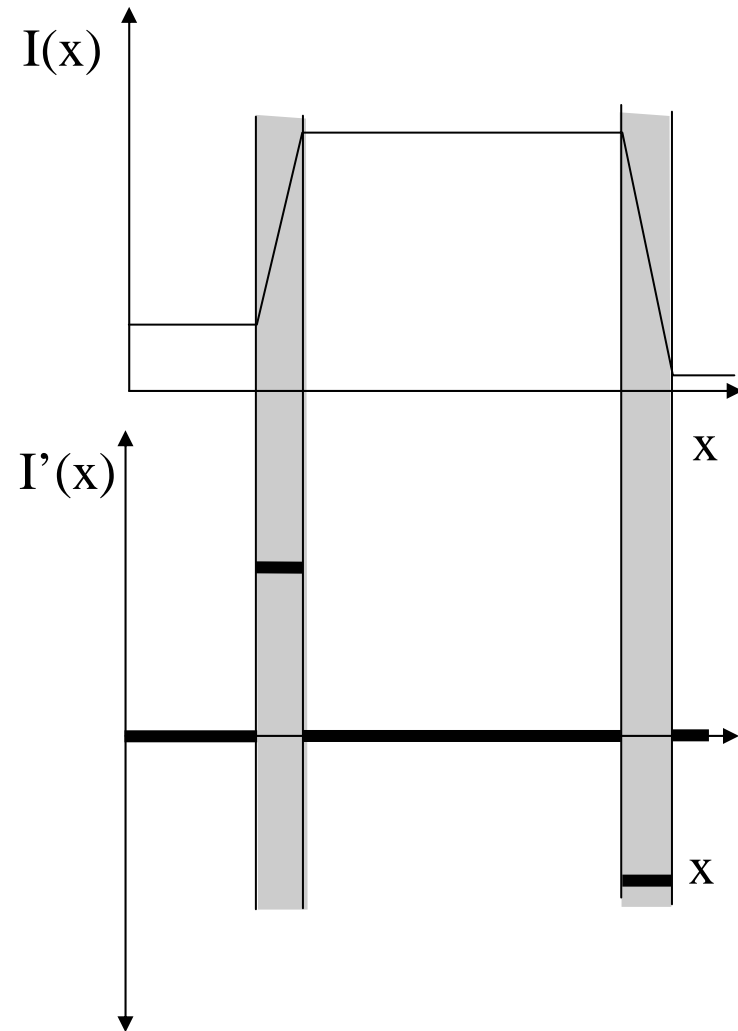


Image gradient (first derivative of $f(x,y)$)

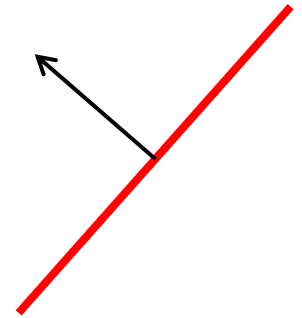
$$\mathbf{G}[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

- orientation (direction of steepest ascent)

$$\alpha(x,y) = \tan^{-1} (G_y / G_x)$$

- magnitude

$$\|\mathbf{G}[f(x,y)]\| = \sqrt{(G_x^2 + G_y^2)} \\ \sim \|G_x\| + \|G_y\|$$



3.1 Simple Edge Detectors -Discrete Approximations-

- Robert's Cross

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

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

- Sobel

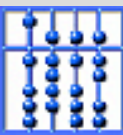
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

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

- Prewitt

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

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



Simple Edge Detectors -The Laplacian (2nd Derivative)-

Optima of 1. derivative of $f(x,y)$

Zero-crossings of 2. derivative of $f(x,y)$

$$\Delta^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

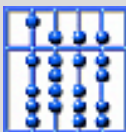
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{\delta^2 f}{\delta x^2} = f[i,j+1] - 2f[i,j] + f[i,j-1]$$

$$\frac{\delta^2 f}{\delta y^2} = f[i+1,j] - 2f[i,j] + f[i-1,j]$$

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

Problem: very sensitive to noise!

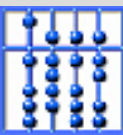


3.2 Image Convolution

$$h(x,y) = f(x,y) * g(x,y)$$

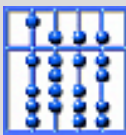
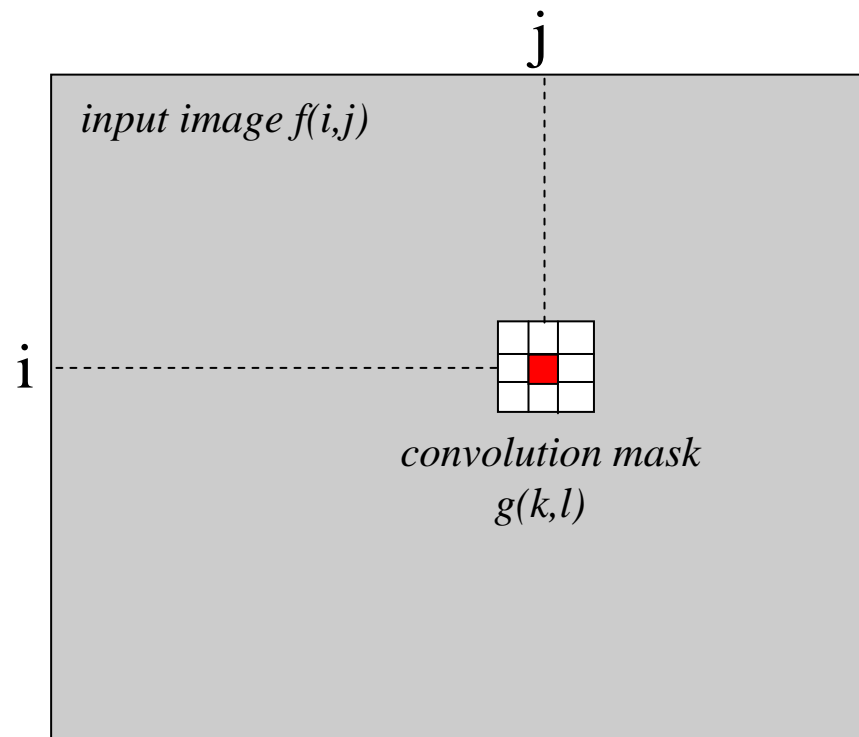
$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x + x', y + y') g(x', y') dx' dy'$$

$$h [i,j] = \sum_{k=0}^{n-1} \sum_{l=0}^{m-1} f [i - n/2 +k, j - n/2 +l] g[k, l]$$



Discrete Image Convolution

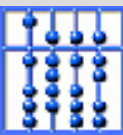
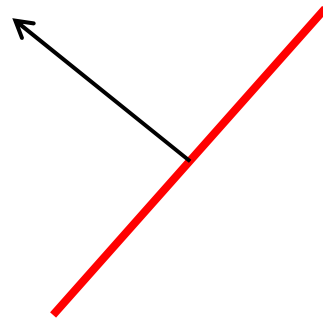
$$h[i,j] = \sum_{k=0}^{n-1} \sum_{l=0}^{m-1} f[i - n/2 + k, j - n/2 + l] g[k, l]$$



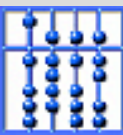
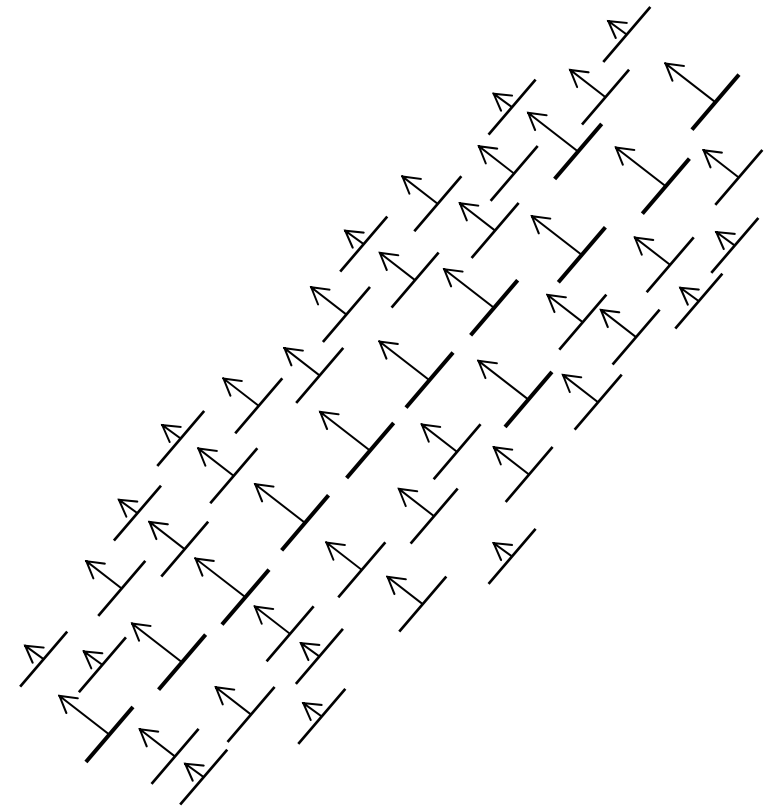
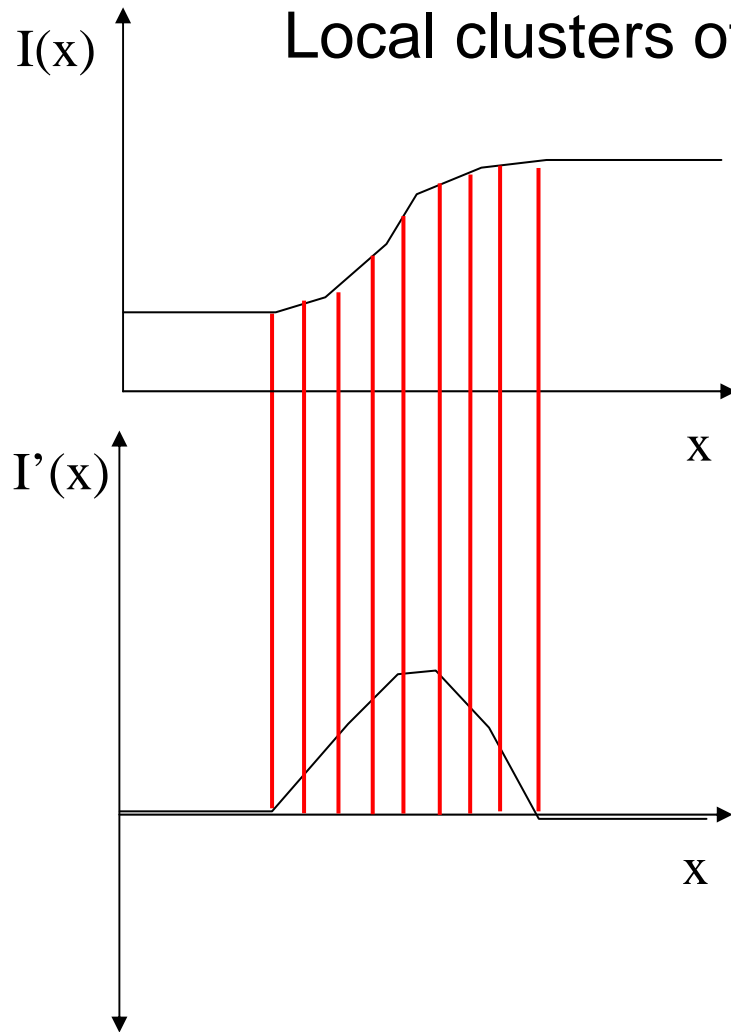
3.3 Result: Edgels

Edgels: pixelwise indicator of local

- edge strength
- dominant edge direction

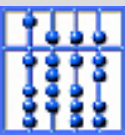


Edgels



3.4 Robust Edge Detection

- Detection problems (classification errors)
 - Lack of accuracy (position, orientation)
 - False edges (“false positives”)
 - Missing edges (“false negatives”)
- Combined filters for
 - Smoothing (noise reduction)
 - Enhancement (edge detection)
 - Detection (magnitude thresholding)
 - [Localization (subpixel precision)]



3.4.1 Smoothing Filters

- Local pixel averages

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

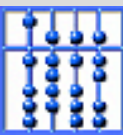
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * 1/16$$

- Gaussian low-pass filter

$$g[k,l] = e^{-\frac{(k^2+l^2)}{2\sigma^2}}$$

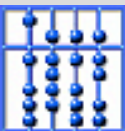
with
k: -n/2 .. n/2
l: -m/2 .. m/2

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 2 & 4 & 8 & 16 & 8 & 4 & 2 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 & 1 & 1 \end{bmatrix} * 1 / 144$$



3.4.2 Enhancement

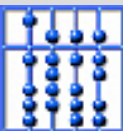
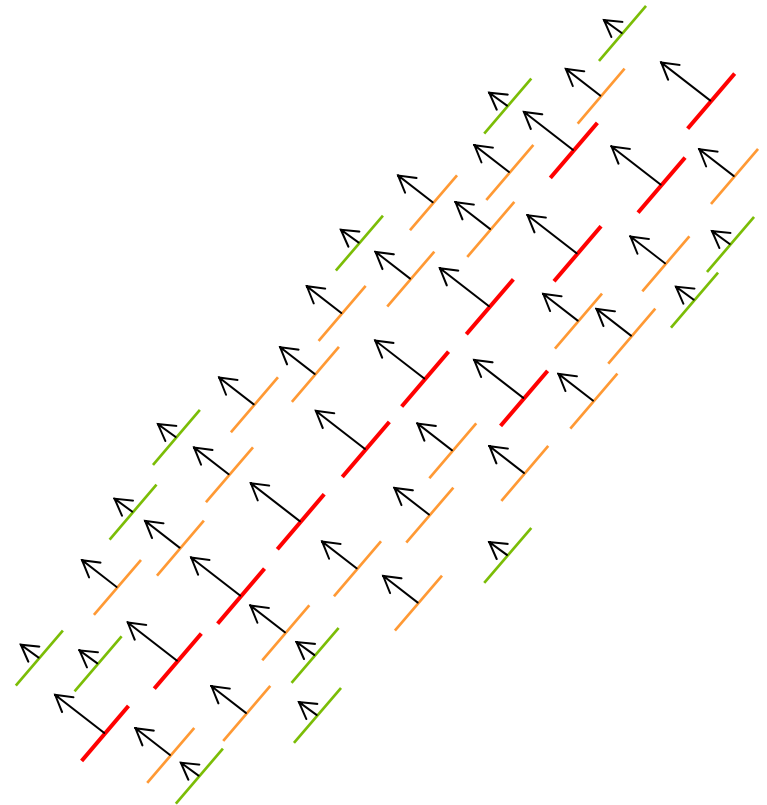
- Gradient-based edge detection
- Second derivative



3.4.3 Detection

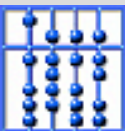
Thinning of edgel clusters:

- Determination of local maxima along the dominant edge direction
- Computation of zero-crossings in the 2nd derivative



3.4.4 Example: Mexican Hat Operator

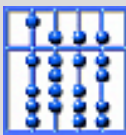
- Laplacian of Gaussian (LoG)
 - Smoothing with a Gaussian filter
 - Enhancement by 2. derivative edge detection
 - Detection of zero crossings in 2. derivative in combination with large peak in 1. derivative
 - Localization with subpixel precision using linear interpolation



- LoG-Operator $h(x,y) = \Delta^2[g(x,y) * f(x,y)]$
 $= [\Delta^2g(x,y)] * f(x,y)$

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	4	12	21	24	21	12	4	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0



3.4.5 Example: Canny Edge Detector

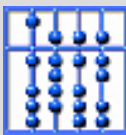
- First derivative of a Gaussian
(sensitive in the direction of steepest change,
insensitive along edge)

$$S[i,j] = G[i,j; \sigma] * I[i,j]$$

$$P[i,j] = \begin{matrix} -S[i,j] & +S[i,j+1] \\ -S[i+1,j] & +S[i+1,j+1] \end{matrix} \quad \begin{matrix} -1 & 1 \\ -1 & 1 \end{matrix}$$

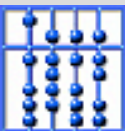
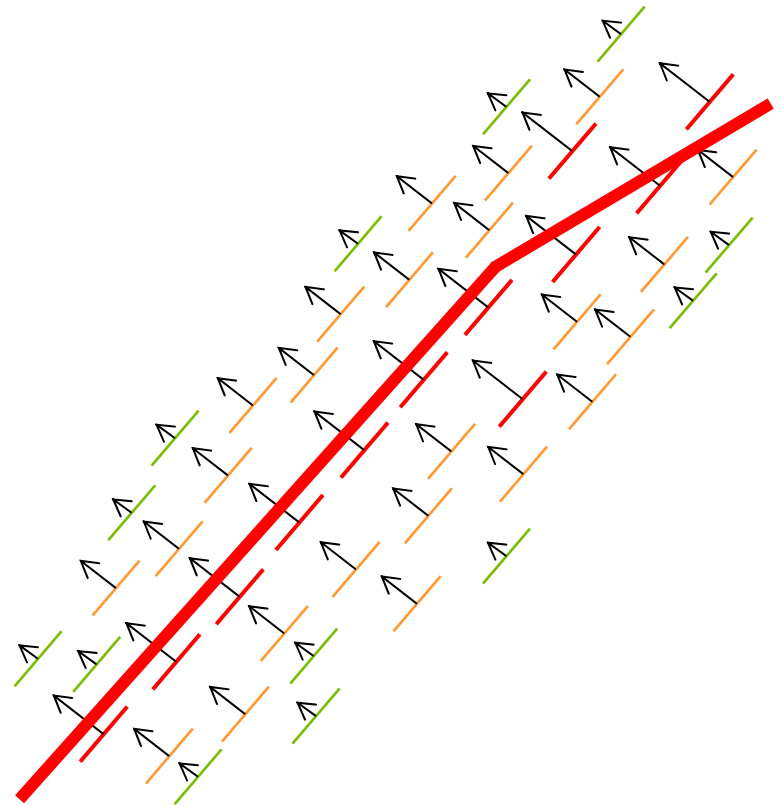
$$Q[i,j] = \begin{matrix} S[i,j] & +S[i,j+1] \\ -S[i+1,j] & -S[i+1,j+1] \end{matrix} \quad \begin{matrix} 1 & 1 \\ -1 & -1 \end{matrix}$$

- Nonmaxima suppression (ridge thinning)
- Double thresholding to detect and link edges

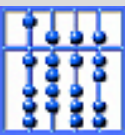


3.5 Shape Fitting

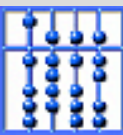
- Determination of consistent groups of edgels (clustering)
- Robust estimation of shape parameters (least squares, disregarding outliers)



- Goodness of fit
 - Maximum absolute error
 - Mean squared error
 - Normalized maximum error
 - Number of sign changes
 - Ratio of curve length to end point distance

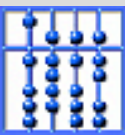


- Line segments (Polylines)
- Circular arcs
- Conic sections
- Cubic splines



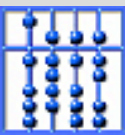
Polylines

- Sequence of line segments
 - implicit line representation $f(x,y) = 0$
 - distance from line $\rightarrow f(x,y) = d$
- Polyline splitting (recursive subdivision)
- Segment merging (bottom-up approach)
- Split and merge



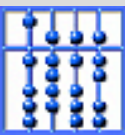
Overview

- Image Formation (Geometry, Radiometry)
- Feature Detection
 - Region(Similarity)-based Approaches
 - Edge(Difference)-based Approaches
- Feature Matching and Tracking



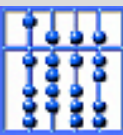
4. Feature Matching (3D to 2D)

- Photometric (Colors)
 - Color constancy problems: changing illumination, shadowing, ...
 - Invariant to changing image resolution / object size
- Geometric (Shapes)
 - One object at a time (unique recognition): Geometric invariants under 3D transformations and projections
 - Groups of objects in combination (graph matching)



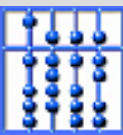
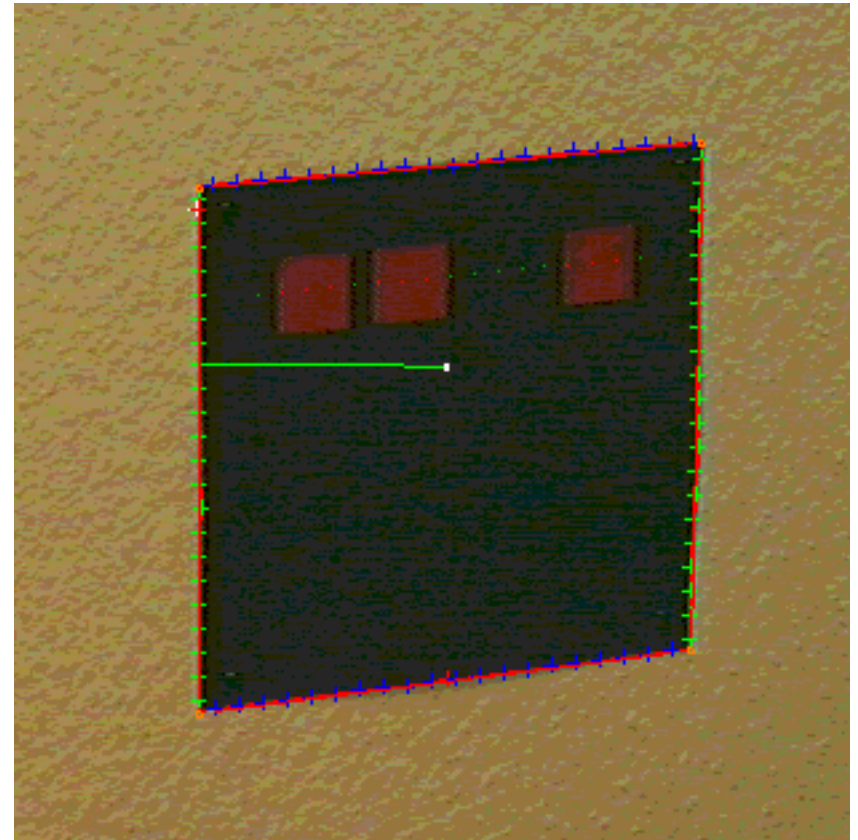
Feature Matching (2D to 2D)

- Init: known feature positions in Image_{1..i-1}
- Loop: (real-time)
 - Predict approximate position of each feature in Image_i
 - Search local image areas for features
 - Handle exceptions:
 - Disappearing feature
 - Partially visible feature
 - Reappearing feature
 - ...Update feature motion models...
 - ...Compute 3D interpretation...
- Continue loop



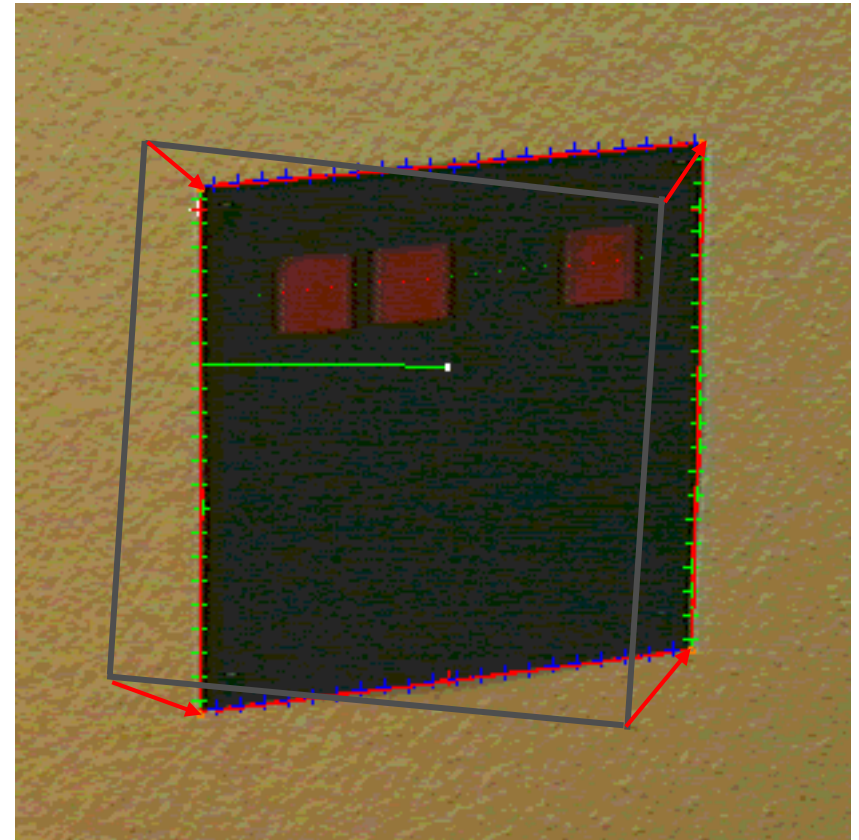
Example: - Target Detection and Identification -

- Find dark blobs on bright background
- Fit quadrilateral polygons
- Find corners
- Read ID label



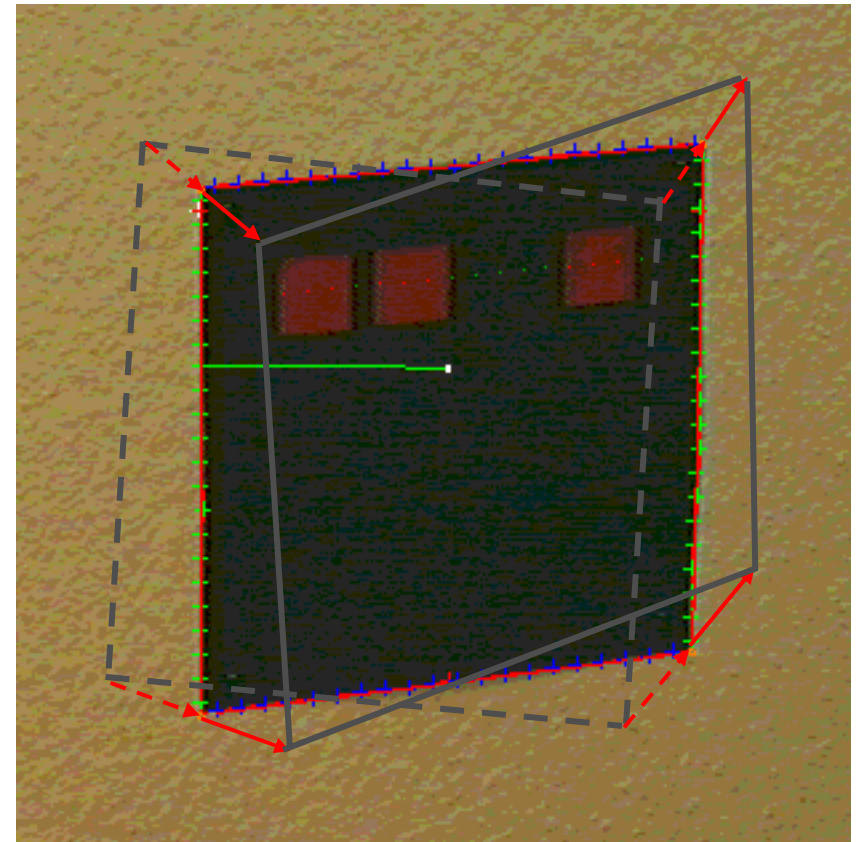
2D Motion Estimation

- Compute local motion vectors of every feature (images $n-1$, $n-2$)
- (more images to estimate higher-order motion models)



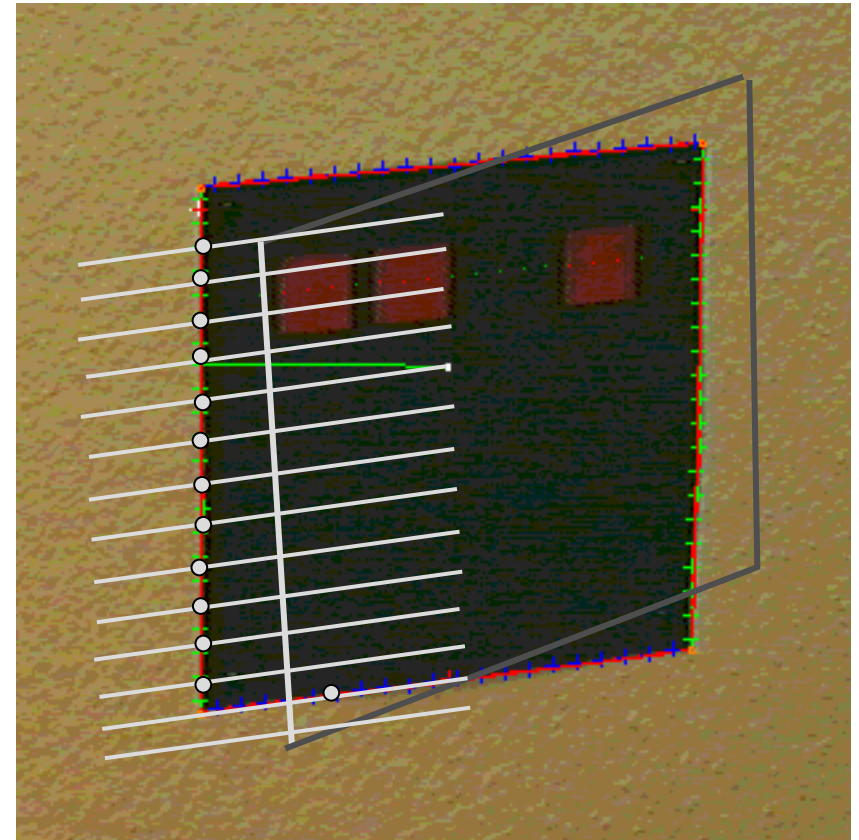
2D Motion Prediction

- Prediction of local feature motion for image n



Target Redetection

- Prediction of local feature motion for image n



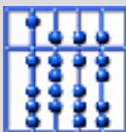
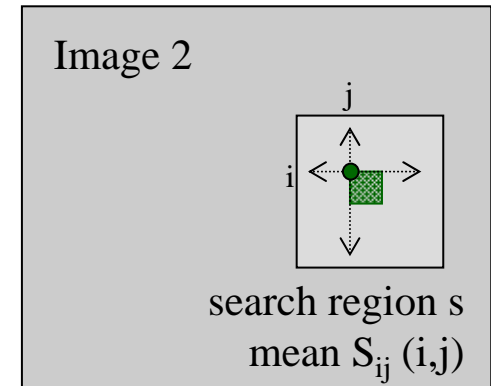
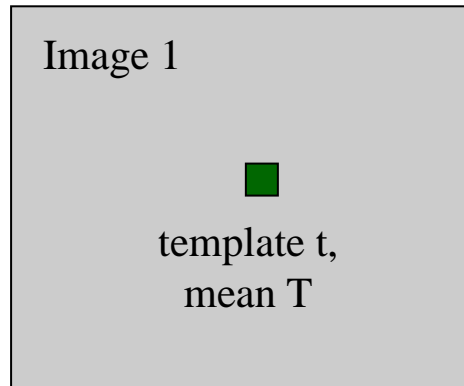
Other 2D Matching Techniques - Cross Correlation -

- unnormalized:

$$r(i,j) = \sum_{x,y} \{t(x,y) * s(i+x,j+y)\}$$

- normalized:

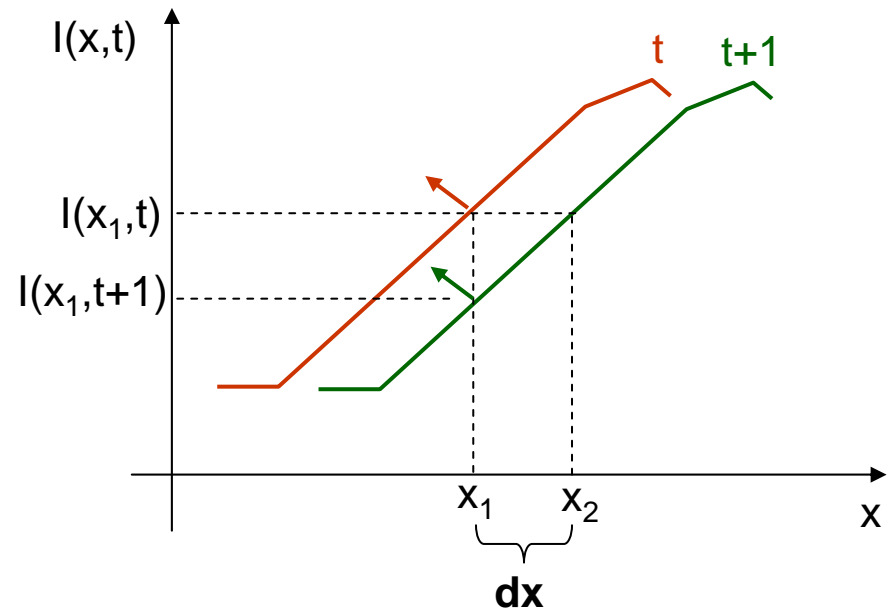
$$r(i,j) = \frac{\sum \{ [t(x,y)-T] * [s(i+x,j+y)-S_{ij}] \}}{\sqrt{\sum [t(x,y) - T]^2 * \sum [s(i+x,j+y) - S_{ij}]^2}}$$



Velocity field in the image plane due to the motion of the observer, objects or “apparent motion”

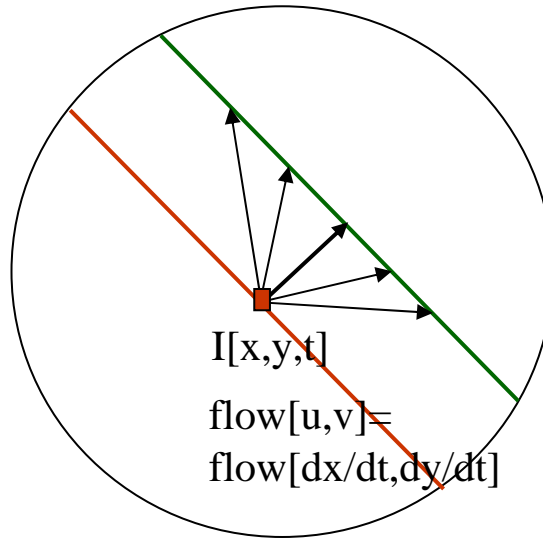
$$u = dx = \frac{I(x_1, t) - I(x_1, t+1)}{\| \text{image gradient} \|}$$

$$v = dy = \dots$$



Gradient-Based Image Flow

- Aperture problem



Two variables (u,v) per pixel; only one constraint

Gradient-Based Image Flow

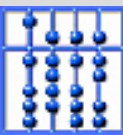
- Smoothness assumption:
The velocity field varies smoothly over an image.

$$u = u_{\text{average}} - I_x \frac{P}{D}$$

$$v = v_{\text{average}} - I_y \frac{P}{D}$$

$$P = I_x u_{\text{average}} + I_y v_{\text{average}} + I_t$$

$$D = \lambda^2 + I_x^2 + I_y^2$$



Gradient-Based Image Flow

- Variational calculus

- Image-flow constraint $flow(x,y,t)$:

$$E_x u + E_y v + E_t = 0$$

- Smoothness constraint:

$$sm(x,y,t) = \iint \left[\left[\frac{\delta u}{\delta x} \right]^2 + \left[\frac{\delta u}{\delta y} \right]^2 + \left[\frac{\delta v}{\delta x} \right]^2 + \left[\frac{\delta v}{\delta y} \right]^2 \right] dx dy$$

- Minimize:

$$\iint (\text{flow}(x,y,t)^2 + \lambda^2 \text{smoothness}(x,y,t)) dx dy$$

