Karlsruhe Institute of Technology

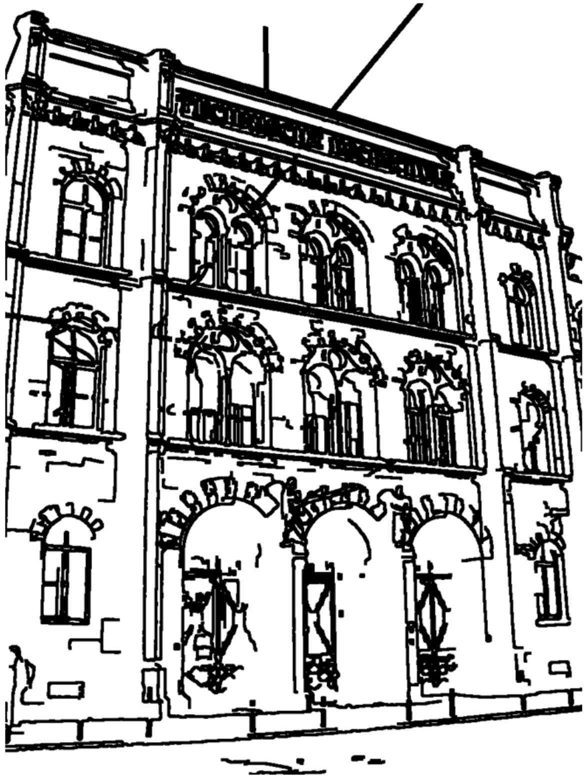
Machine Vision

Chapter 3: Edge and Corner Detection

Dr. Martin Lauer

Institut für Mess- und Regelungstechnik

Edge Detection



grey level edges:

- areas of hard changes between bright and dark areas
- typically occur at object boundaries
- occur at shadows and texture
- edges independent of image brightness
- many parts of human visual cortex are dealing with grey level edges

Finding Edges

- edges are areas of rapidly changing grey value

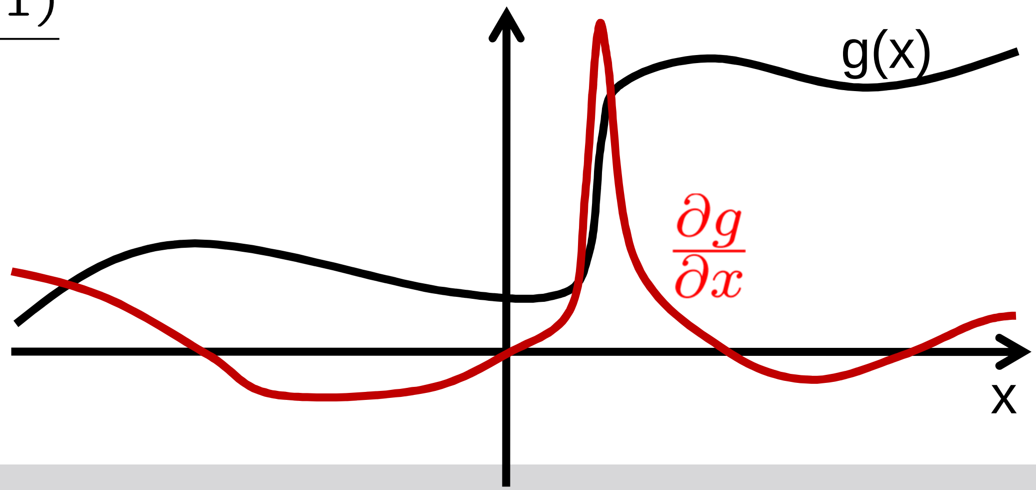
$$|g(x+\epsilon) - g(x-\epsilon)| \text{ large for small } \epsilon$$

- search areas with large derivative of g

$$\frac{\partial g}{\partial x} = \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon} = \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x-\epsilon)}{2\epsilon}$$

- approximating derivative by difference:

$$\frac{\partial g}{\partial x} \approx \frac{g(x+1) - g(x-1)}{2}$$



Finding Edges cont.

- approximating the derivative can be implemented as convolution with filter mask:

$$\frac{1}{2}\cdot\begin{array}{|c|c|c|}\hline 1 & \boxed{0} & -1 \\ \hline\end{array} \qquad \frac{\partial g}{\partial x} \approx \frac{g(x+1)-g(x-1)}{2}$$

- analogously:

$$\frac{1}{2}\cdot\begin{array}{|c|}\hline 1 \\ \hline \boxed{0} \\ \hline -1 \\ \hline\end{array} \qquad \frac{\partial g}{\partial y} \approx \frac{g(y+1)-g(y-1)}{2}$$

- noise reduction: additional averaging

Finding Edges cont.

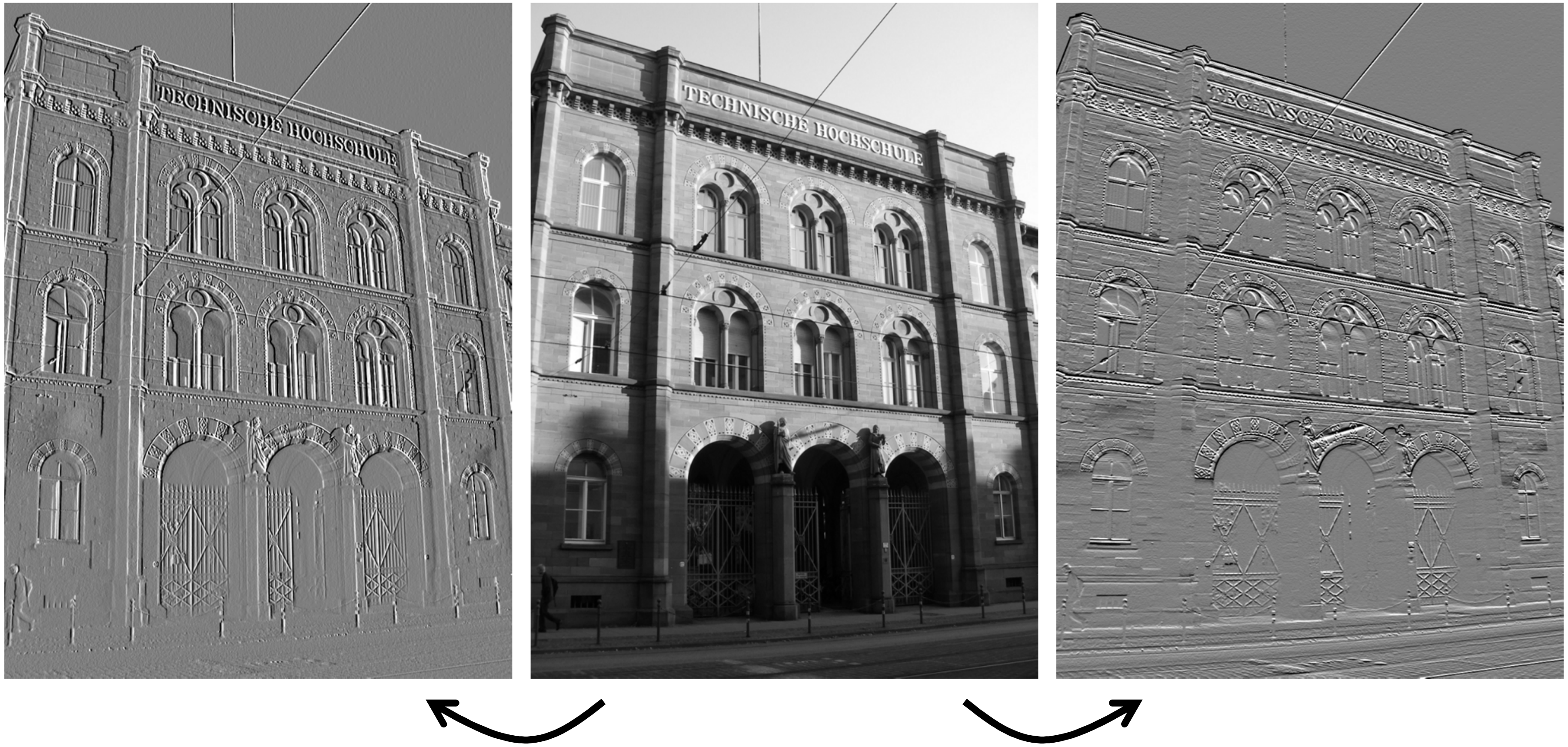
- Prewitt-operator:

$$\frac{1}{6}\cdot\begin{bmatrix} 1 & 0 & -1 \\ 1 & \boxed{0} & -1 \\ 1 & 0 & -1 \end{bmatrix} \qquad \frac{1}{6}\cdot\begin{bmatrix} 1 & 1 & 1 \\ 0 & \boxed{0} & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

- Sobel-operator:

$$\frac{1}{8}\cdot\begin{bmatrix} 1 & 0 & -1 \\ 2 & \boxed{0} & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \frac{1}{8}\cdot\begin{bmatrix} 1 & 2 & 1 \\ 0 & \boxed{0} & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Finding Edges cont.



$$\frac{\partial g}{\partial x}$$

$$\frac{\partial g}{\partial y}$$

Finding Edges cont.

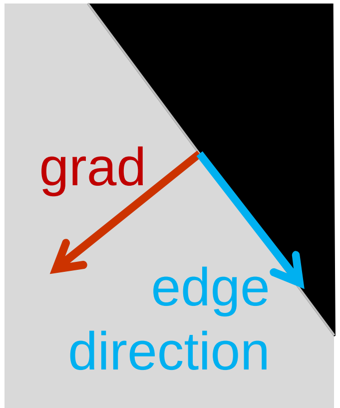
- edge orientation:

$$grad\ g = (\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y})$$

  - grey level gradient points to direction of maximal grey level ascend
  - orthogonal directions exhibit no change of grey level

$$grad\ g\ \perp\ (-\frac{\partial g}{\partial y}, \frac{\partial g}{\partial x})$$

  - length of gradient is proportional to grey level change rate



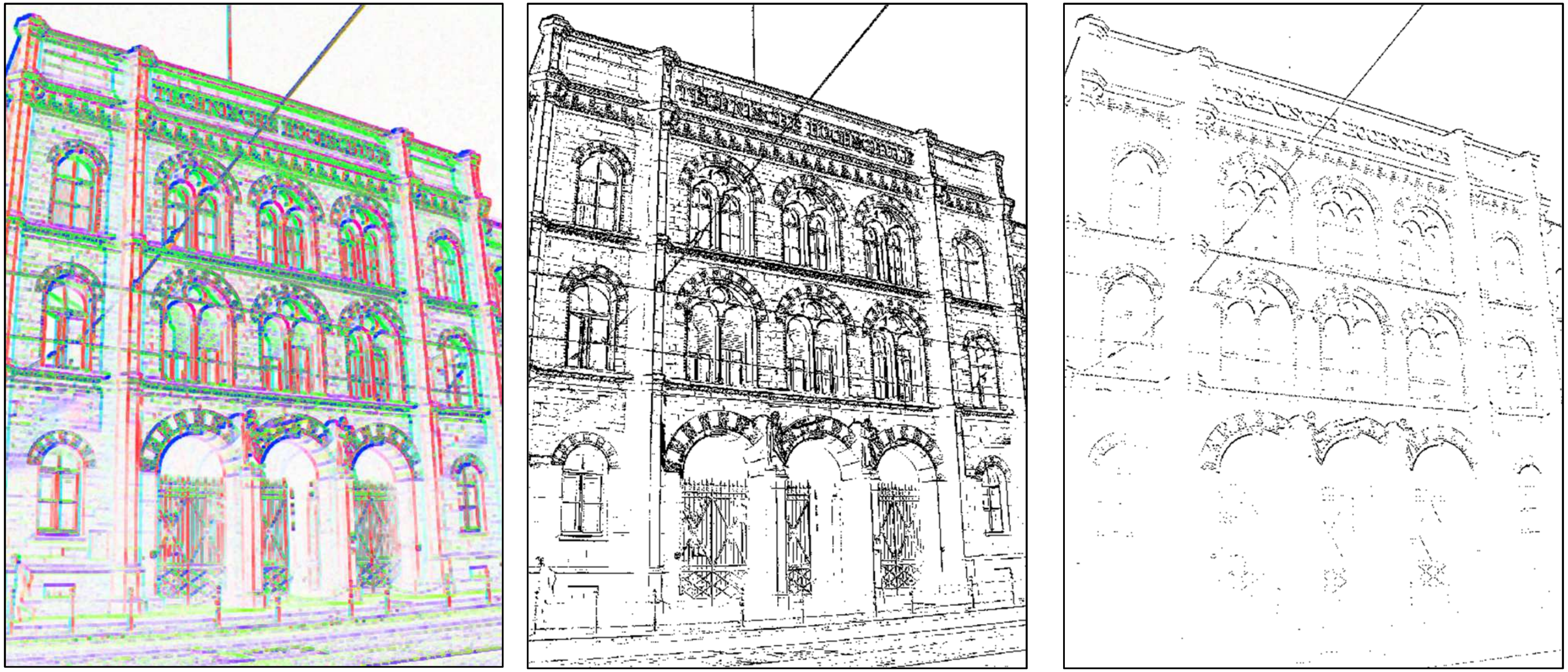
Finding Edges cont.

gradient plot:

- saturation: gradient length
- color: gradient direction (angle)

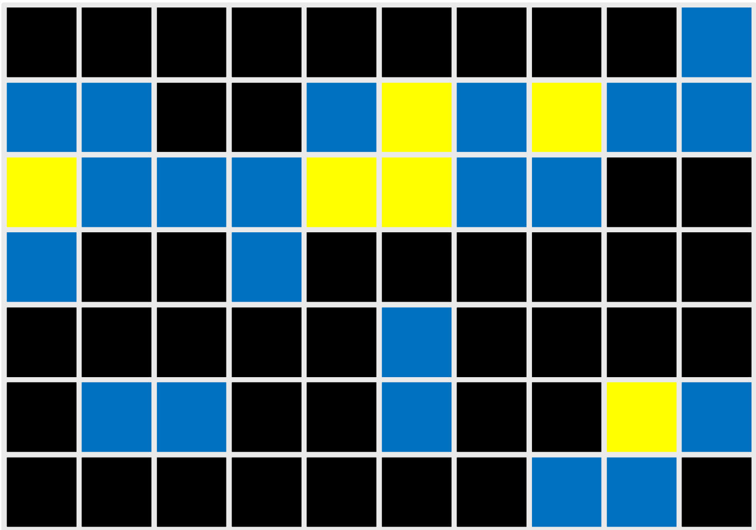
Finding Edges cont.

- from which gradient length on are edges relevant?
  - small threshold: too much noise remains
  - large threshold: contours not connected
  - idea: double thresholding



Finding Edges cont.

- double thresholding:
  - two thresholds: $\theta_L,\ \theta_U,\ \theta_L < \theta_U$
  - pixels are classified according to gradient length $||\vec{g}||$:
    - $||\vec{g}|| \leq \theta_L$ pixel is not edge element
    - $||\vec{g}|| \geq \theta_U$ pixel is edge element
    - $\theta_L < ||\vec{g}|| < \theta_U$ pixel is edge element if a neighboring pixel is edge element



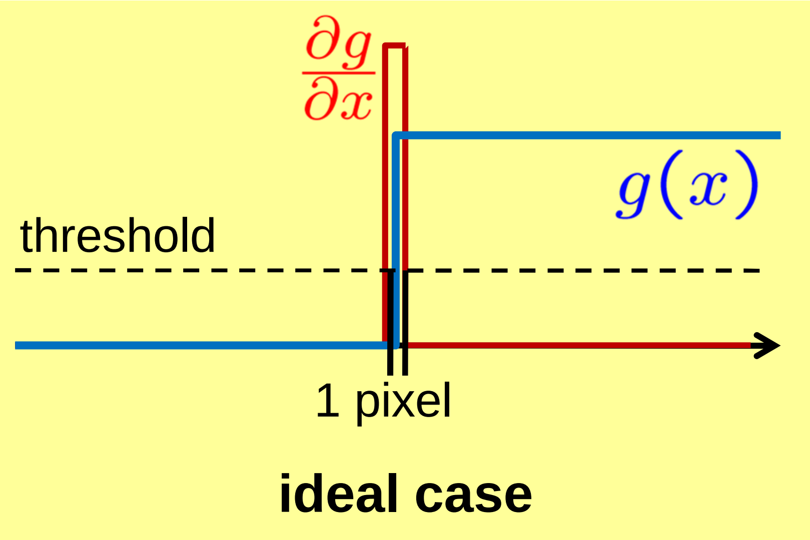
$||\vec{g}|| \leq \theta_L$

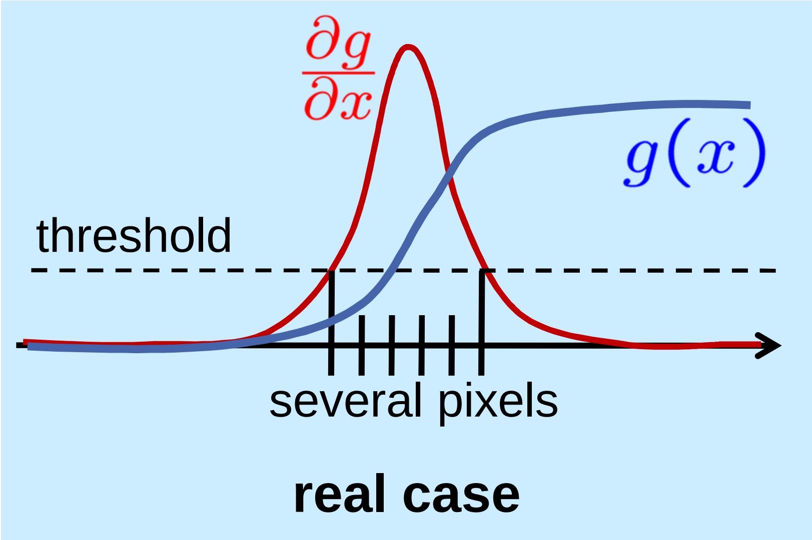
$\theta_L < ||\vec{g}|| < \theta_U$

$||\vec{g}|| \geq \theta_U$

Finding Edges cont.

- problem: thick lines

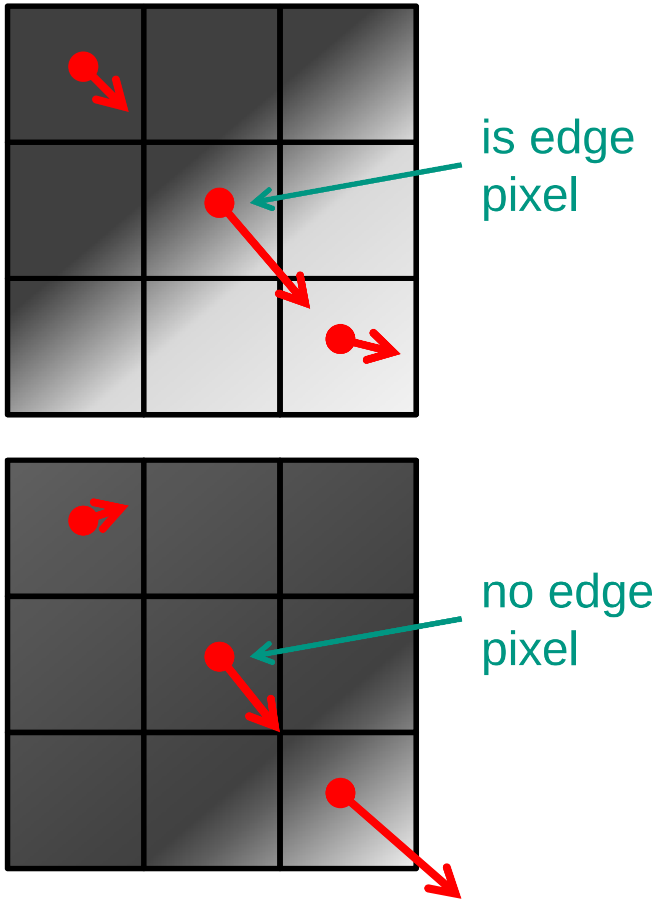




- non-maxima suppression
  - idea: among neighboring pixels, consider only the one with maximal gradient length
  - 2D case: take into account edge direction

Finding Edges cont.

- Non-maxima suppression

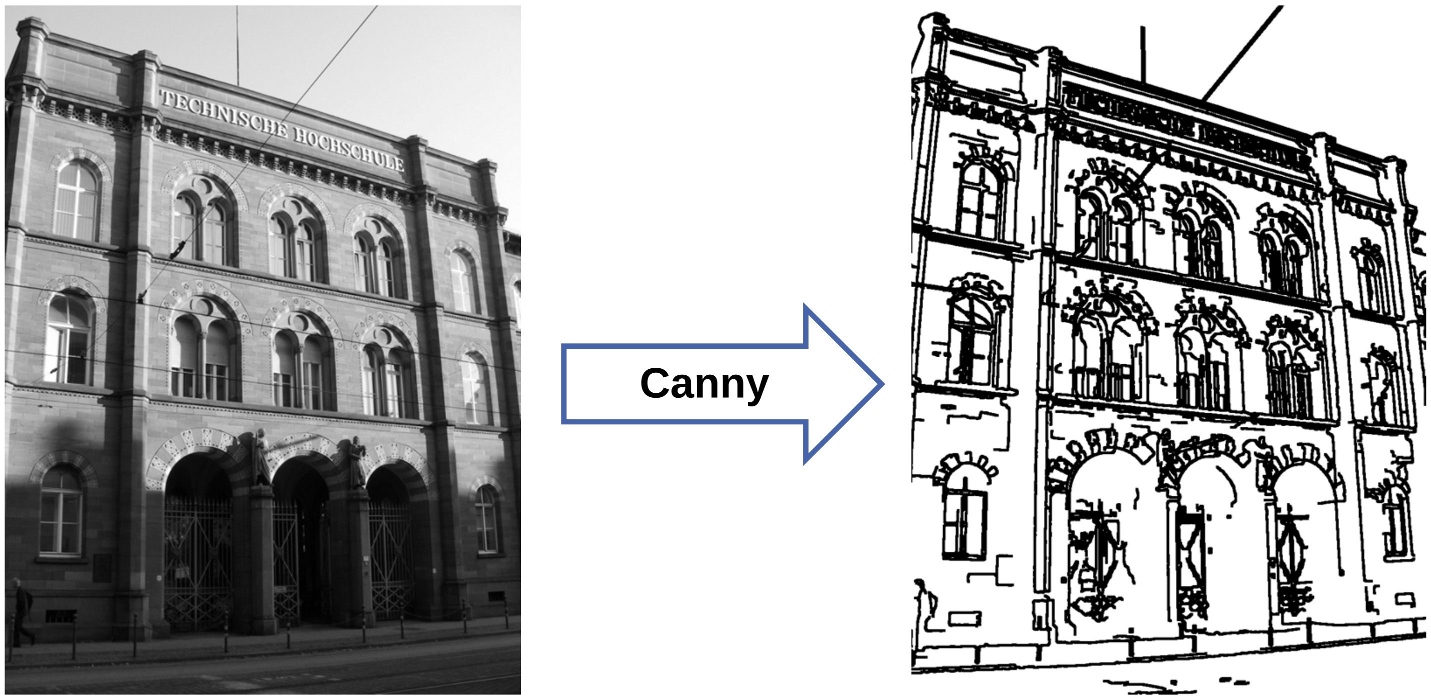


- check gradient direction
- select neighboring pixel in gradient direction and opposite gradient direction
- pixel is edge pixel if gradient length is larger than in those two neighboring pixels

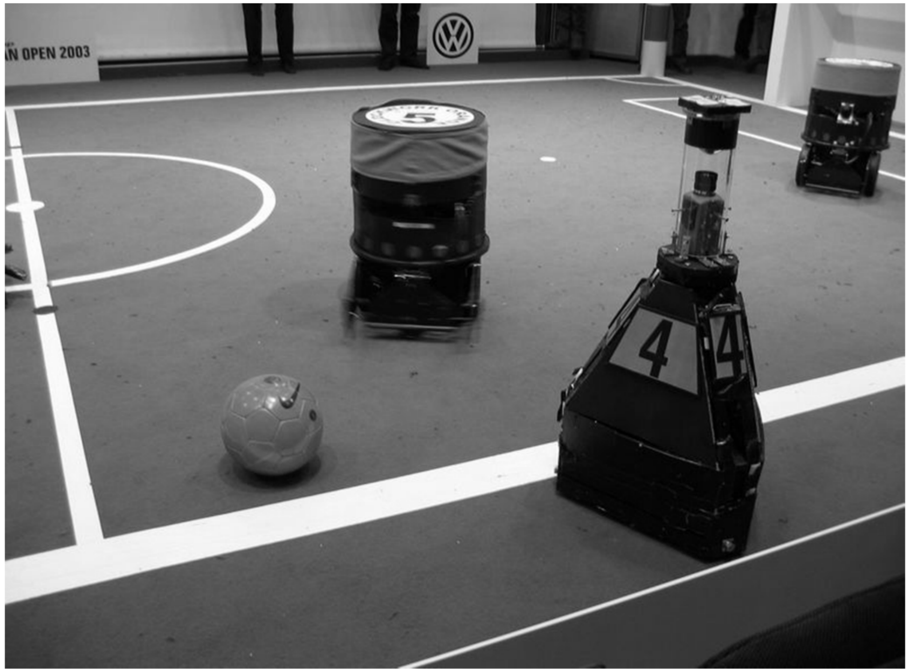
Canny Edge Detector

Canny edge detector combines the following techniques:

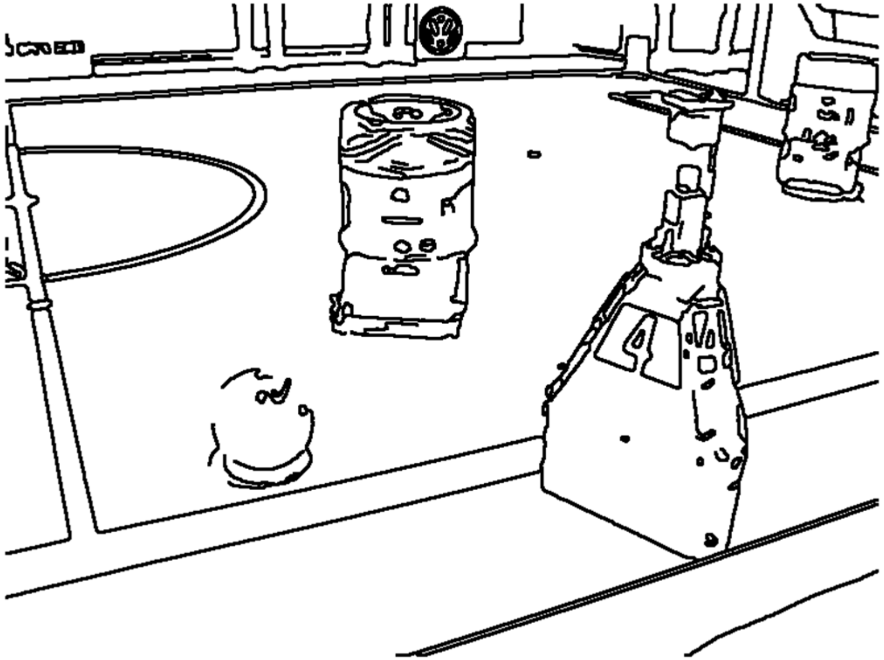
1. smooth image with Gaussian filter
2. compute grey level gradient with Sobel/Prewitt masks
3. apply non-maxima suppression
4. apply double thresholding



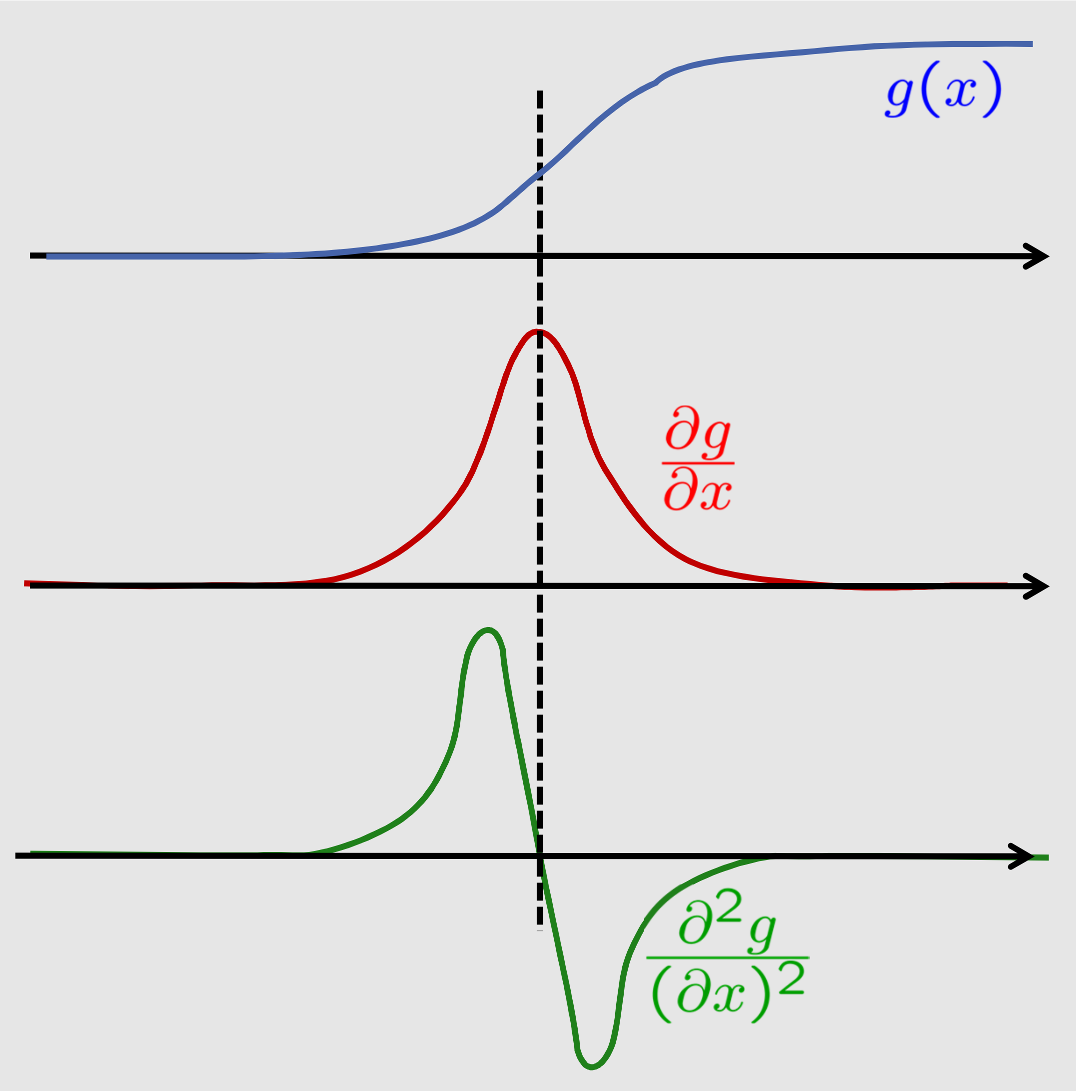
Canny Edge Detector



Canny



Finding Edges cont.



grey level edge:

- g rapidly changing
- maximum of $\frac{\partial g}{\partial x}$
- zero crossing of $\frac{\partial^2 g}{(\partial x)^2}$

- 2D analogon to 2nd order derivative is Laplace operator:

$$\nabla^2 g = \frac{\partial^2 g}{(\partial x)^2} + \frac{\partial^2 g}{(\partial y)^2}$$
$$= trace(H)$$
$$(H \text{ Hessian})$$

Laplace Operator

- Approximation to Laplace operator:

$$\frac{\partial g}{\partial x}(x,y) \approx g(x+1,y) - g(x,y)$$

$$\begin{aligned}\frac{\partial^2 g}{(\partial x)^2}(x,y) &\approx \frac{\partial g}{\partial x}(x,y) - \frac{\partial g}{\partial x}(x-1,y) \\ &\approx g(x+1,y) - 2g(x,y) + g(x-1,y)\end{aligned}$$

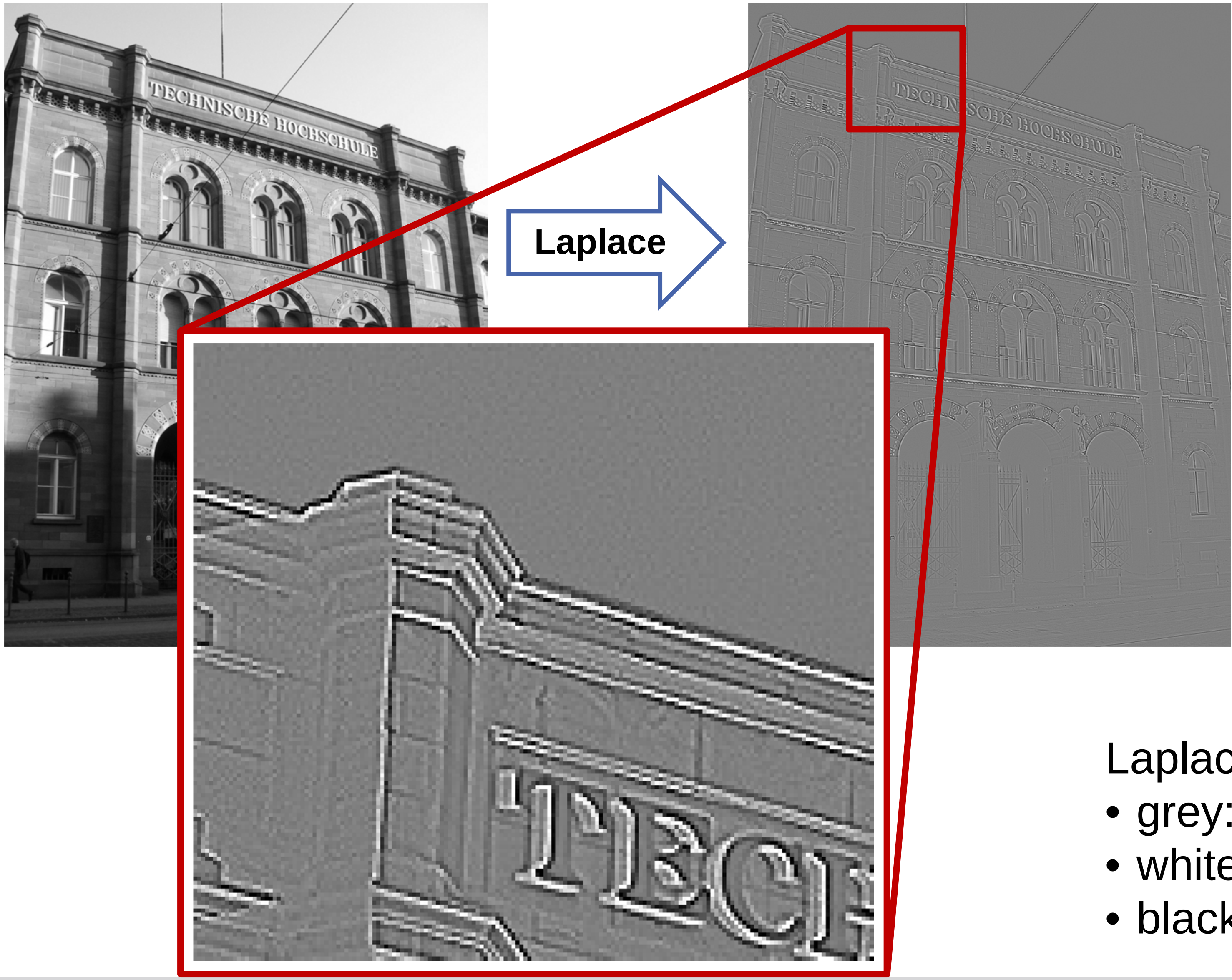
$$\frac{\partial^2 g}{(\partial y)^2}(x,y) \approx g(x,y+1) - 2g(x,y) + g(x,y-1)$$

$$\nabla^2 g \approx g(x+1,y) + g(x-1,y) + g(x,y+1) + g(x,y-1) - 4g(x,y)$$

- implementation as filter mask:

$$\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & \boxed{-4} & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array} \qquad \frac{1}{6}\cdot\begin{array}{|c|c|c|}\hline 1 & 4 & 1 \\ \hline 4 & \boxed{-20} & 4 \\ \hline 1 & 4 & 1 \\ \hline\end{array}$$

Laplace Operator cont.



Laplace operator
- grey: zero
- white: positive
- black: negative

Laplace Operator cont.

- 2nd order derivative is very noisy
- combine Laplacian with Gaussian smoothing

$$\nabla^2(G * g) = (\nabla^2 G) * g$$

$$(G \text{ Gaussian})$$

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(x^2+y^2)}$$

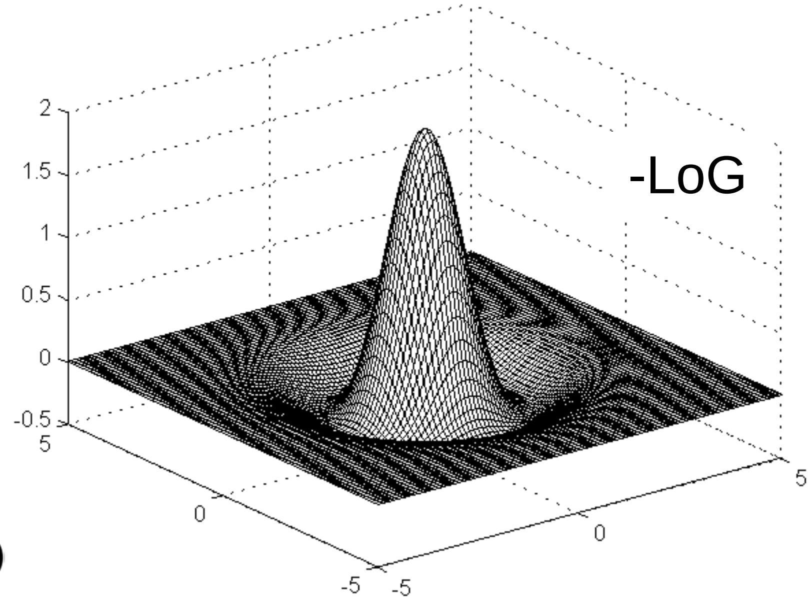
$$\frac{\partial G}{\partial x} = \frac{1}{2\pi\sigma^2}\left(-\frac{1}{2\sigma^2}\right) 2x e^{-\frac{1}{2\sigma^2}(x^2+y^2)}$$

$$= -\frac{x}{\sigma^2} G(x,y)$$

$$\frac{\partial^2 G}{(\partial x)^2} = -\frac{1}{\sigma^2} G(x,y) - \frac{x}{\sigma^2}\left(-\frac{x}{\sigma^2} G(x,y)\right)$$

$$= \frac{x^2 - \sigma^2}{\sigma^4} G(x,y)$$

$$\nabla^2 G = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} G(x,y)$$



"Laplacian of Gaussian" (LoG)

"mexican hat"

Laplace Operator cont.

- filter masks for LoG:

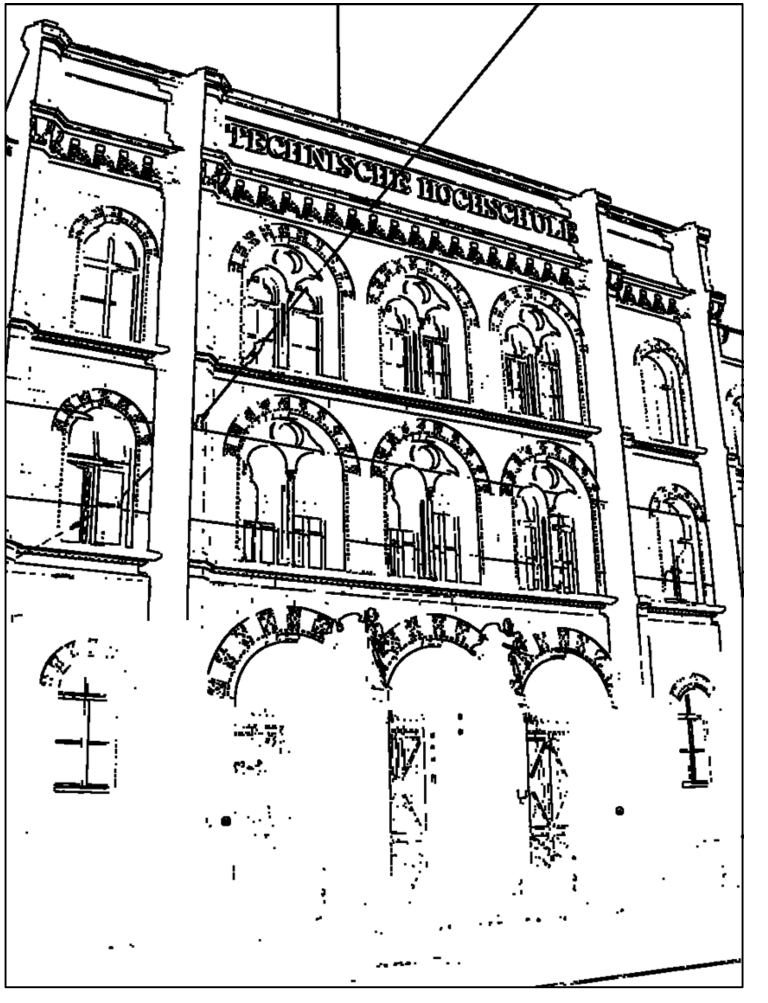
0	0	1	0	0
0	1	2	1	0
1	2	-16	2	1
0	1	2	1	0
0	0	1	0	0

- LoG can be approximated by DoG “Difference of Gaussian”

$$DoG(x,y) = G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y)$$

Laplace Operator cont.

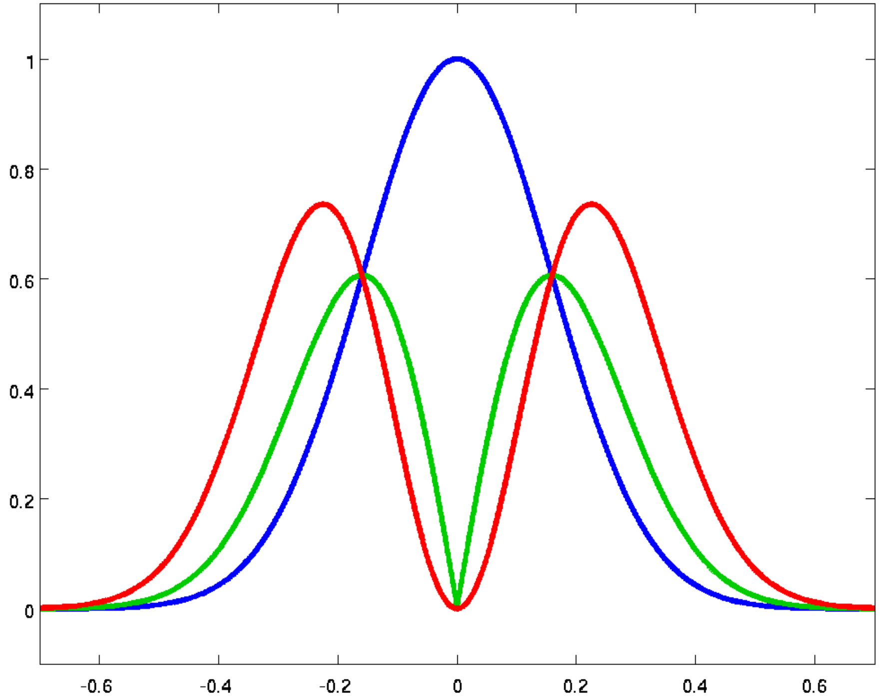
LoG +
Threshold



- edge detection approach according to Marr/Hildreth

Edge Detection cont.

- A brief look on the Fourier spectrum of $G, \nabla G, \nabla^2 G$:



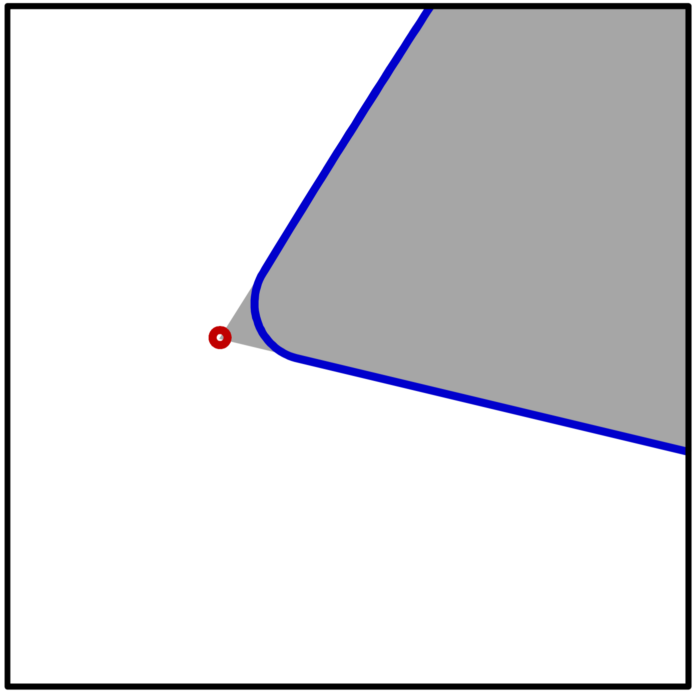
- Gauss function implements low-pass filter
- derivation implements high-pass filter
- derivatives of Gauss implements band-pass filter

CORNER DETECTION

Corner Detection

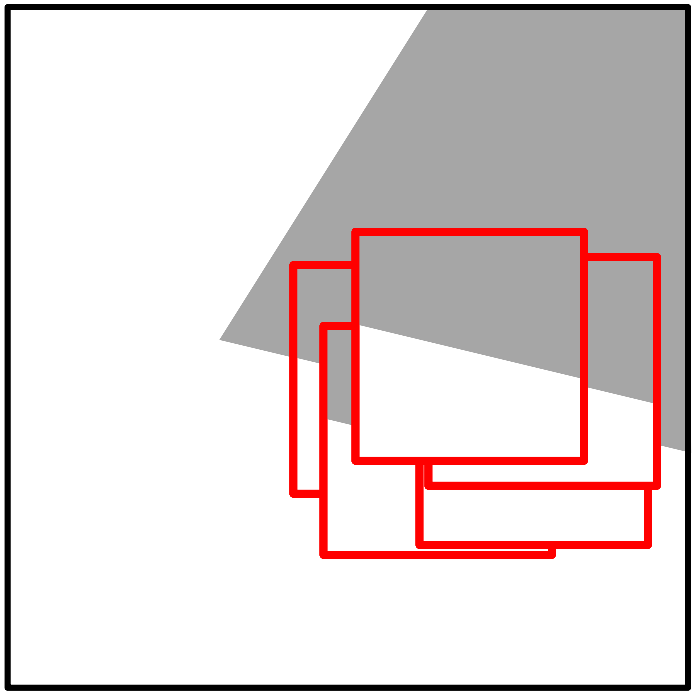
- graylevel corners important due to:
  - good features to find again in another image
  - corners as feature points, e.g. for calibration
  - edge detector usually round off corners

→ special filter to detect corners

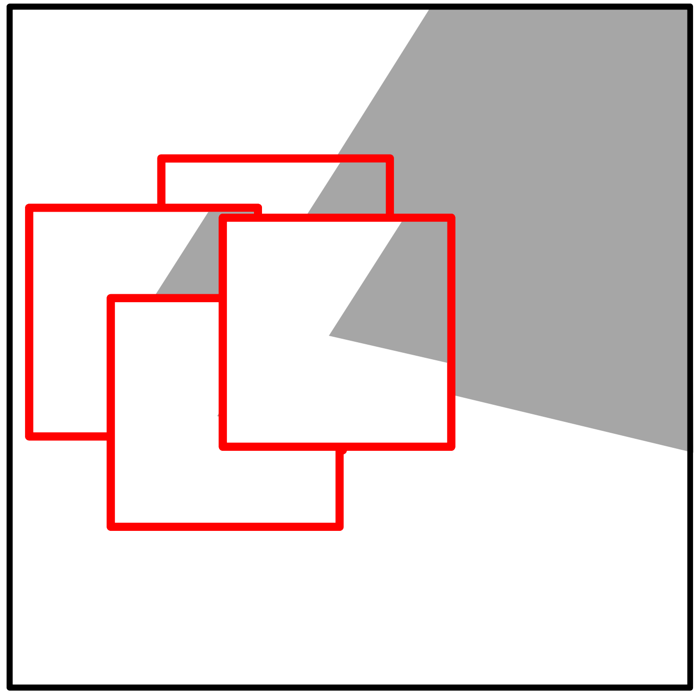


Corner Detection cont.

- Idea: find patches of maximal dissimilarity for local moves



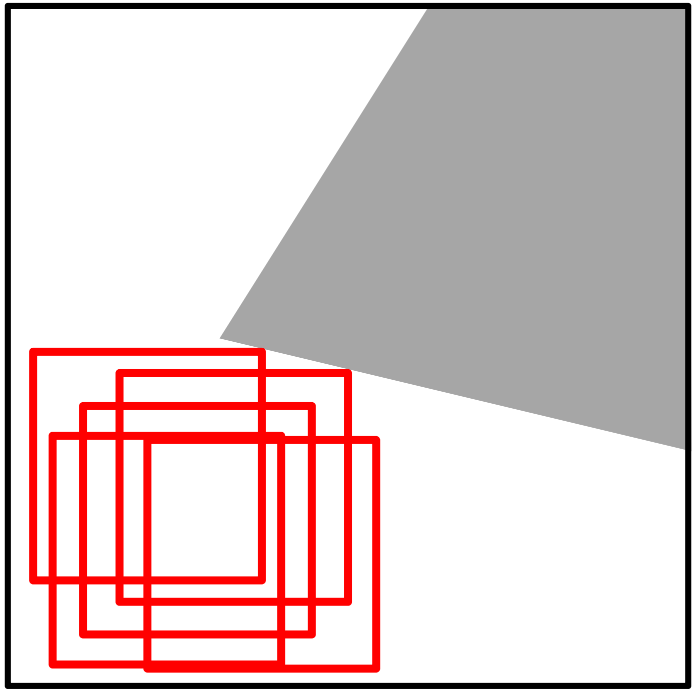
edge: similar when moving along edge, dissimilar when moving orthogonal



corner: dissimilar in all direction

Corner Detection cont.

- Idea: find patches of maximal dissimilarity for local moves



homogeneous areas:
similar in all directions

dissimilarity measure:

$$\sum_{(u,v)\in rectangle} (g(u+\Delta u, v+\Delta v) - g(u,v))^2$$

$$\approx \sum_{(u,v)\in rectangle} (g(u,v) + \Delta u \frac{\partial g}{\partial u} + \Delta v \frac{\partial g}{\partial v} - g(u,v))^2$$

$$= \sum_{(u,v)\in rectangle} ((\Delta u \frac{\partial g}{\partial u})^2 + 2\Delta u \frac{\partial g}{\partial u} \Delta v \frac{\partial g}{\partial v} + (\Delta v \frac{\partial g}{\partial v})^2)$$

$$= \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}^T \underbrace{\begin{pmatrix} \sum (\frac{\partial g}{\partial u})^2 & \sum \frac{\partial g}{\partial u}\frac{\partial g}{\partial v} \\ \sum \frac{\partial g}{\partial u}\frac{\partial g}{\partial v} & \sum (\frac{\partial g}{\partial v})^2 \end{pmatrix}}_{=:S} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}$$

Corner Detection cont.

- Dissimilarity measure:

$$d := \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}^T \underbrace{\begin{pmatrix} \sum(\frac{\partial g}{\partial u})^2 & \sum \frac{\partial g}{\partial u}\frac{\partial g}{\partial v} \\ \sum \frac{\partial g}{\partial u}\frac{\partial g}{\partial v} & \sum(\frac{\partial g}{\partial v})^2 \end{pmatrix}}_{=:S} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}$$

  - dissimilarity should be large for all unit vectors $(\Delta u, \Delta v)$
  - for special choice of coordinate system S becomes a diagonal matrix (Eigenvector coordinate system)

$$d := \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}^T \underbrace{\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}}_{=S} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \lambda_1 (\Delta u)^2 + \lambda_2 (\Delta v)^2$$

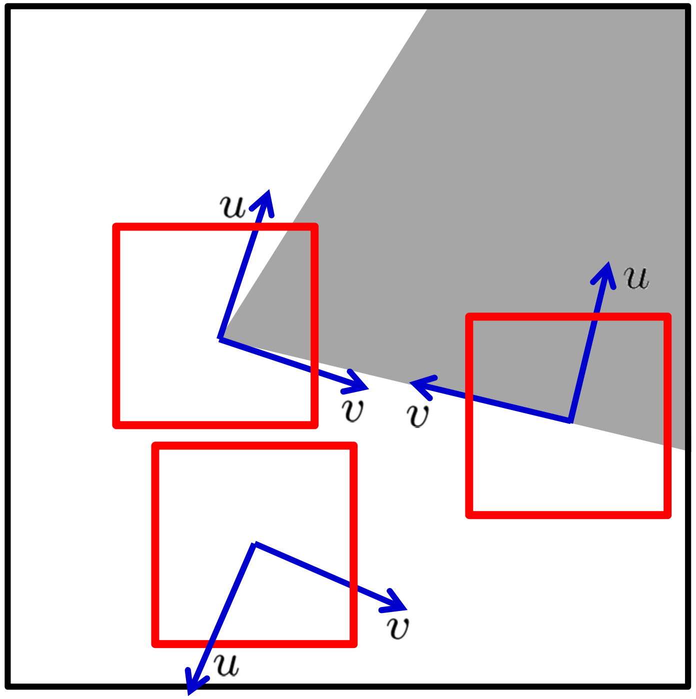
$$\text{w.l.o.g. } \lambda_1 \geq \lambda_2 \geq 0$$

Corner Detection cont.

$$d := \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}^T \underbrace{\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}}_{=S} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \lambda_1 (\Delta u)^2 + \lambda_2 (\Delta v)^2$$

w.l.o.g. $\lambda_1 \geq \lambda_2 \geq 0$

- typical cases:



	λ_1	λ_2
edge	large	small
corner	large	large
homogeneous area	small	small

Corner Detection cont.

- decision rule
  - pixel is in homogeneous area if

$$trace(S) = \lambda_1 + \lambda_2 < \theta$$

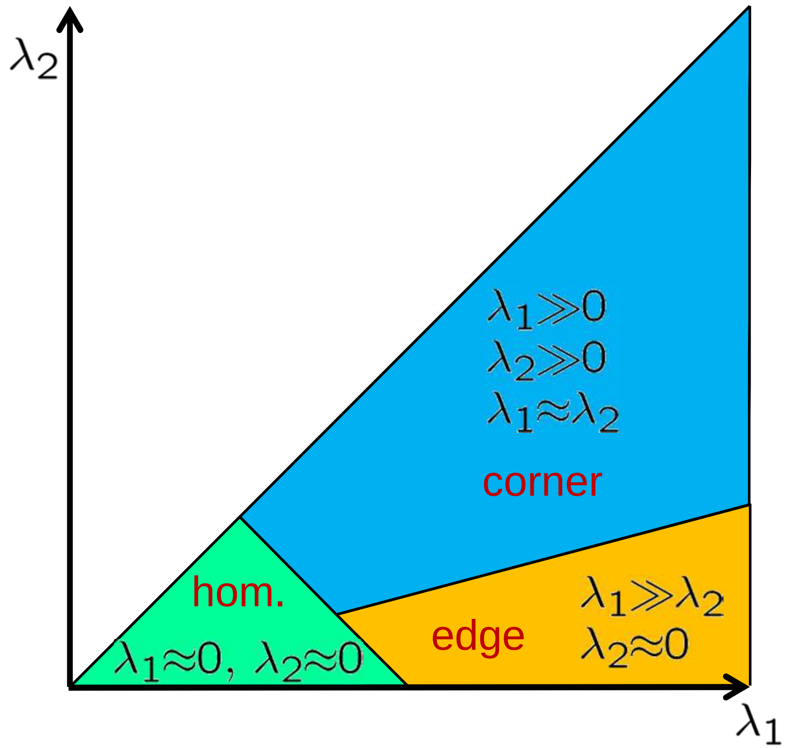
  - otherwise, pixel is corner if

$$\lambda_2 > \alpha\lambda_1$$

$$\Leftrightarrow det(S) - \frac{\alpha}{(1+\alpha)^2}(trace(S))^2 > 0$$

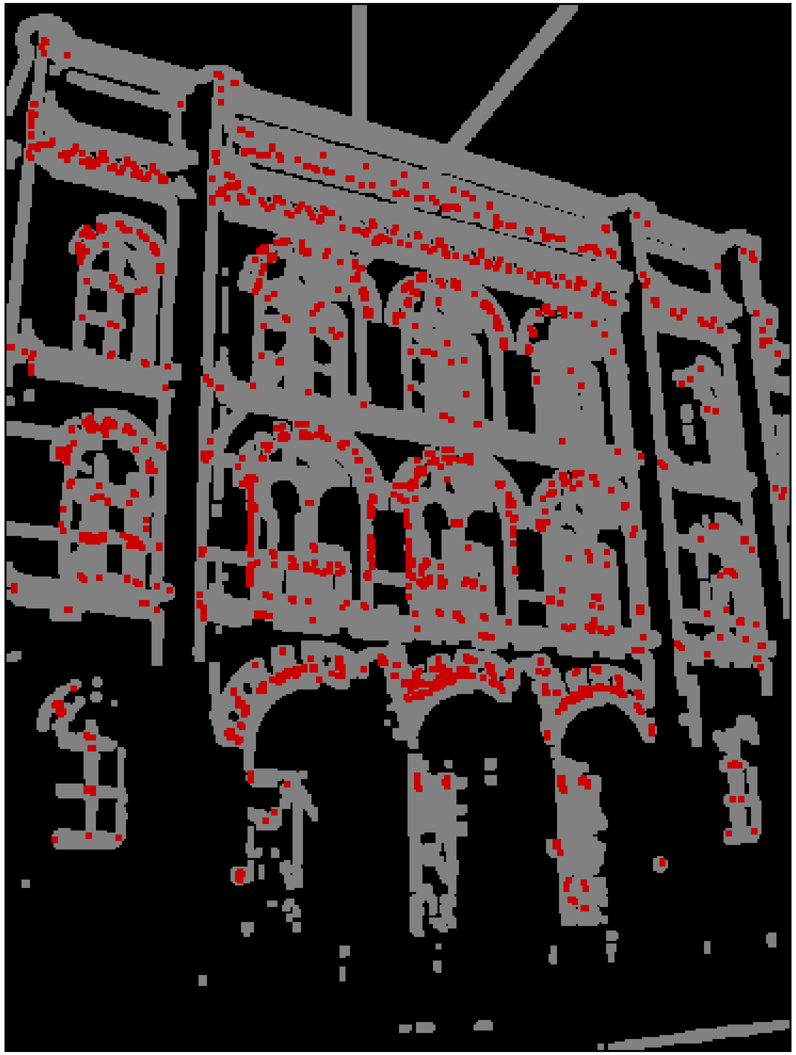
  - otherwise, pixel is edge

- parameters θ and α have to be tuned manually



Harris corner detector

Corner Detection cont.



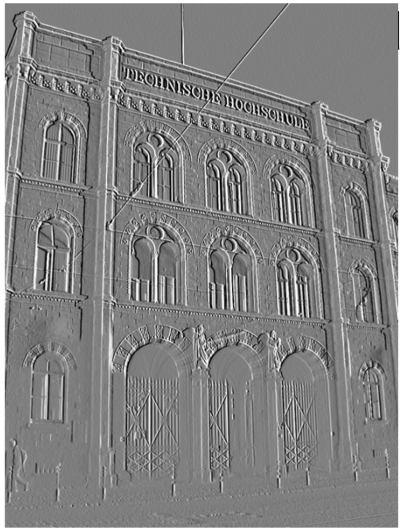
SUMMARY:
EDGE AND CORNER DETECTION

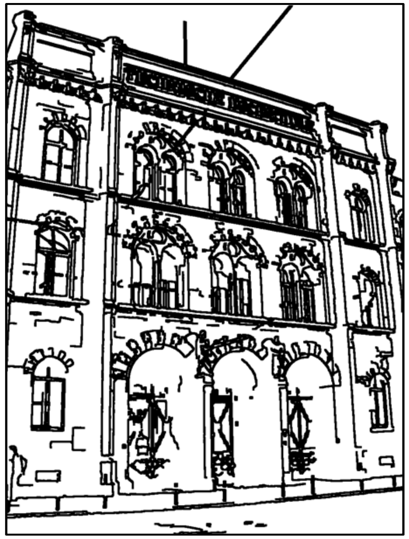
Summary

- **edge detection**
  - gradient: Sobel, Prewitt
  - thresholding, double thresholding
  - non-maxima suppression
  - Canny operator
  - Laplace operator, Marr/Hildreth approach

1	0	-1
2	$\boxed{0}$	-2
1	0	-1

1	4	1
4	$\boxed{-20}$	4
1	4	1



Summary cont.

- **edge detection**
- **corner detection**
  - Harris corner detector