Image Analysis

Filter Design

Technical University
of Applied Sciences
Würzburg-Schweinfurt

Linear Filters

- Average Filters
- Derivative Filters

Nonlinear Filters

- Sorting: Min, Max, Median
- Morphological Operators

Steerable Filters

- Anisotropic Averaging
- Directional Derivatives

Geometric Transformations

- Forward- and Backward Transformation
- Affine and Projective Transformations
- Interpolation

Point Operators

- Definition
- Histogram Methods

Averaging - Desired Properties

The aim of averaging is

- decreasing fine structures

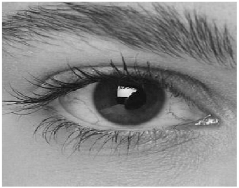
in image space. This is equivalent to the

- suppression of high wavenumbers

in frequency space.

Thus, an averaging is supposed to act like a 2D low-pass filter and can be used for noise reduction or image smoothing (blurring).

Original



Locally Averaged

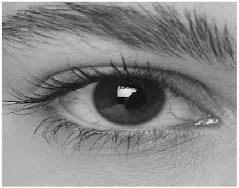
Linear Filters

Averaging - Ideal Properties

For ideal averaging, the filter must have the following properties:

- no displacement: No change of object positions
  $\rightarrow$ symmetric filter masks.
- preservation of mean value: No change of brightness
  $\rightarrow$ sum of all coefficients of the mask equal to one.
- Smoothing property: Finer structures are attenuated more than coarser ones
  $\rightarrow$ Monotonic decrease of the transfer function.
- isotropy: Directional independence of the smoothing
  $\rightarrow$ isotropic transfer function.

Original



Locally Averaged

Linear Filters

Box filter - Properties

Point spread function **R** of the box filter:

$$\mathbf{R} = \frac{1}{r^2}\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}}_{r\text{-times}} = \frac{1}{r}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \frac{1}{r}\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}.$$

3x3 box filter

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

Advantages:

- separable
- recursively computable: three computations per pixel independent of filter size

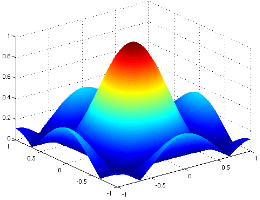
Disadvantages:

- no isotropic transfer function
- no monotonic decrease of the transfer function

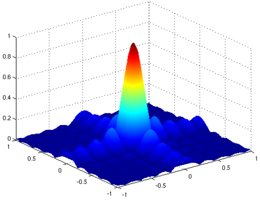
Linear Filters

Box filter - Transfer function

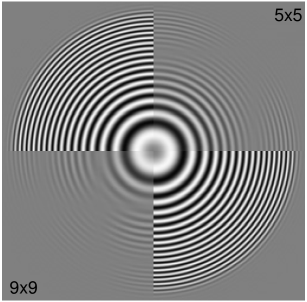
$|\hat{\mathbf{R}}(\tilde{\mathbf{k}})|$ of 5x5 filter



$|\hat{\mathbf{R}}(\tilde{\mathbf{k}})|$ of 9x9 filter



Test image



Transfer function:

$$\hat{\mathbf{R}}(\tilde{\mathbf{k}}) = \frac{\sin(\pi r \tilde{k}_x/2)}{r\sin(\pi \tilde{k}_x/2)} \frac{\sin(\pi r \tilde{k}_y/2)}{r\sin(\pi \tilde{k}_y/2)}$$

whereas $\tilde{k}_x \in [-1;1]$ and $\tilde{k}_y \in [-1;1]$ are normalised wavevector components.

Linear Filters

Binomial filter - Properties

Point spread function **B** of the binomial filter:

$$\mathbf{b} = \frac{1}{2^r} \underbrace{[11] * [11] * \cdots * [11]}_{r\text{-times}}, \quad \mathbf{B} = \mathbf{b}\mathbf{b}^\top .$$

3x3 Filter

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

Computation scheme: Pascal's triangle

r	s		σ^2
0	1	1	0
1	1/2	1 1	1/4
2	1/4	1 2 1	1/2
3	1/8	1 3 3 1	3/4
4	1/16	1 4 6 4 1	1
5	1/32	1 5 10 10 5 1	5/4
6	1/64	1 6 15 20 15 6 1	3/2
7	1/128	1 7 21 35 35 21 7 1	7/4
8	1/256	1 8 28 56 70 56 28 8 1	2

Advantages:

- separable, almost isotropic
- strictly monotonically decreasing transfer function towards zero

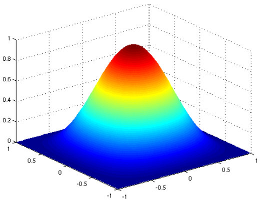
Disadvantages:

- slight anisotropy along the diagonals
- variance not arbitrarily selectable

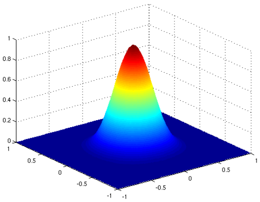
Linear Filters

Binomial filter - Transfer function

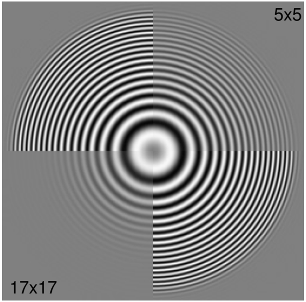
$|\hat{\mathbf{B}}(\tilde{\mathbf{k}})|$ of 5x5 filter



$|\hat{\mathbf{B}}(\tilde{\mathbf{k}})|$ of 9x9 filter



Test image



Transfer function:

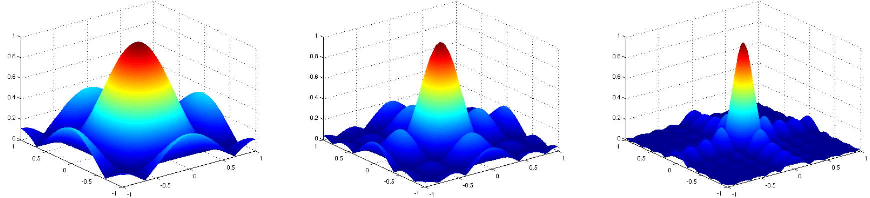
$$\hat{\mathbf{B}}(\tilde{\mathbf{k}}) = \cos^r(\pi \tilde{k}_x/2) \cos^r(\pi \tilde{k}_y/2)$$

whereas $\tilde{k}_x \in [-1;1]$ and $\tilde{k}_y \in [-1;1]$ are normalized wave vector components.

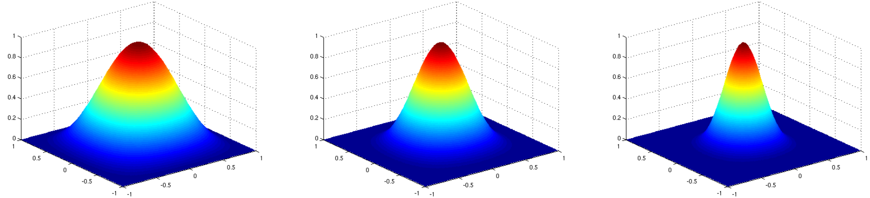
Linear Filters

Comparison: Box filter - Binomial filter

Box filter $|\hat{\mathbf{R}}(\tilde{\mathbf{k}})|$ 3x3 filter 5x5 filter 9x9 filter



Binomial filter $|\hat{\mathbf{B}}(\tilde{\mathbf{k}})|$ $\sigma^2 = 0.5$ $\sigma^2 = 1$ $\sigma^2 = 2$



Linear Filters

Comparison: Box filter - Binomial filter

Box filter — 3x3 filter

5x5 filter

9x9 filter

Binomial filter — $\sigma^2 = 0.5$

$\sigma^2 = 1$

$\sigma^2 = 2$

Linear Filter

Gaussian filter - Ideal Smoothing

Point spread function **G** of continuous Gaussian:

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

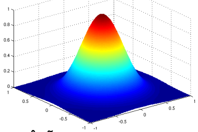
Transfer function $\hat{\mathbf{G}}$:

$$\hat{\mathbf{G}}(\tilde{\mathbf{k}}) = e^{-r\pi\sigma^2(\tilde{k}_x^2+\tilde{k}_y^2)} .$$

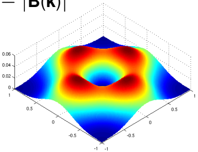
Advantages:

- purely isotropic
- strictly monotonically decreasing Gaussian transfer function
- separable
- arbitrary standard deviation selectable

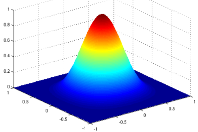
$|\hat{\mathbf{G}}(\tilde{\mathbf{k}})|$ $\sigma^2 = 1$



$|\hat{\mathbf{G}}(\tilde{\mathbf{k}})| - |\hat{\mathbf{B}}(\tilde{\mathbf{k}})|$



$|\hat{\mathbf{B}}(\tilde{\mathbf{k}})|$ $\sigma^2 = 1$



Derivatives - Finding Contours

The goal of derivative filters is to find the

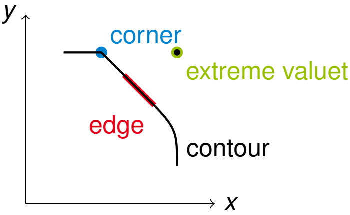
- edges, corners and local extreme values

in the image space. This is done by an evaluation of the

- gradient vector and the Hessian matrix

Not only the contour shape is of interest but also the

- strength of the local contour.

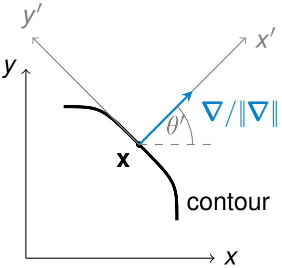


Linear Filters

Differential Operators - 1. Order

The partial derivatives of the image function along both dimensions at an image location $\mathbf{x}$ define the gradient vector $\boldsymbol{\nabla}$

$$\boldsymbol{\nabla}(\mathbf{x}) = \left[\frac{\partial G(\mathbf{x})}{\partial x}, \frac{\partial G(\mathbf{x})}{\partial y}\right]^\top , \quad \boldsymbol{\nabla}'(\mathbf{x}') = \mathbf{R}(\theta')\boldsymbol{\nabla}(\mathbf{x}) ,$$



or the gradient vector $\boldsymbol{\nabla}'$ in a rotated coordinate system $\mathbf{x}'$ with the absolute value of the gradient vector

$$\|\boldsymbol{\nabla}\| = \sqrt{\boldsymbol{\nabla}^\top\boldsymbol{\nabla}} = \sqrt{\left(\frac{\partial G(\mathbf{x})}{\partial x}\right)^2 + \left(\frac{\partial G(\mathbf{x})}{\partial y}\right)^2} , \quad \|\boldsymbol{\nabla}\| = \|\boldsymbol{\nabla}'\| ,$$

being invariant to rotations $\mathbf{R}(\theta')$ of the coordinate system.

Linear Filters

Differential Operators - 1. Order

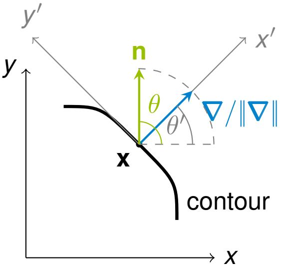
The directional derivative along the direction θ

$$\nabla^\top \mathbf{n} = \cos(\theta)\frac{\partial G(\mathbf{x})}{\partial x} + \sin(\theta)\frac{\partial G(\mathbf{x})}{\partial y}$$

indicates the strength of the change along this direction and corresponds to the scalar product between direction vector $\mathbf{n}$ and gradient ∇

$$\langle \nabla, \mathbf{n} \rangle = \|\nabla\| \cos(\theta - \theta'),$$

whereas $\mathbf{n} = [\cos(\theta), \sin(\theta)]^\top$, with $\|\mathbf{n}\| = 1$.



Technical University of Applied Sciences Würzburg-Schweinfurt

Differential Operators - 2. Order

Second order differential operators describe the curvature of a multidimensional signal. All possible combinations of the partial derivatives of second order form the so-called Hessian matrix **H**. For 2D signals holds:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 G(\mathbf{x})}{\partial x^2} & \frac{\partial^2 G(\mathbf{x})}{\partial x \partial y} \\ \frac{\partial^2 G(\mathbf{x})}{\partial x \partial y} & \frac{\partial^2 G(\mathbf{x})}{\partial y^2} \end{bmatrix}, \text{ and } \quad \mathbf{H}' = \mathbf{R}(\theta')\mathbf{H}\mathbf{R}^\top(\theta'),$$

in a coordinate system $\mathbf{x}'$ rotated by the angle θ' with diagonal Hessian matrix

$$\mathbf{H}' = \begin{bmatrix} \frac{\partial^2 G(\mathbf{x}')}{\partial x'^2} & 0 \\ 0 & \frac{\partial^2 G(\mathbf{x}')}{\partial y'^2} \end{bmatrix}, \quad \Delta = trace(\mathbf{H}) = trace(\mathbf{H}').$$

The trace Δ (also called Laplace operator) is invariant to rotations $\mathbf{R}(\theta')$ of the coordinate system.

Linear Filters

Edge Detection - Possibilities

1. Using Directional Derivatives

Edges correspond to extreme values in 1st order derivatives. Thus edges are obtained by a search for the largest changes of the gradient vector, i.e. the maxima in the magnitude of $\nabla(\mathbf{x})$.

2. Using Hessian Matrix

Edges are zero crossings in 2nd order derivative. Thus the Laplace operator Δ can be used to search for edges. The signal peaks next to the zero crossings must be significantly higher than the noise level, so that the zero-crossing corresponds to an edge.

Comparison: The Laplace operator is much more susceptible to noise than the magnitude of the gradient vector.

Linear Filters

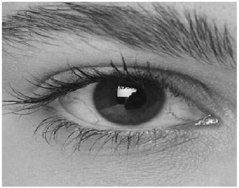
Derivatives - Ideal Properties

For an ideal derivative, the filter must have the following properties:

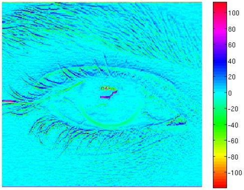
- No displacement: No change of object positions $\rightarrow$ antisymmetric filter masks for 1st derivatives, symmetric filter masks for 2nd derivatives.
- mean suppression: No response to spatially constant brightness $\rightarrow$ sum of all coefficients of the mask equal to zero.
- isotropy: Directional independence of the gradient $\rightarrow$ isotropic transfer function.

Objective: Design discrete filters that give the most accurate approximation of the derivatives.

Original



1. derivate in y-direction



Linear Filters

Derivatives - Finite First Order Differences

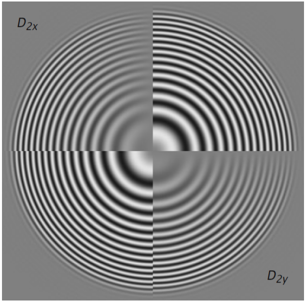
The first partial derivative in continuous space can be approximated in discrete space by differences. There are three different realizations for the derivative in x-direction $\partial G(x, y)/\partial x$ (analogously in y-direction $\partial G(x, y)/\partial y$), where only the symmetric difference is displacement-free:

Differences:

- Backward difference $\frac{(G(x,y)-G(x-\Delta x,y))}{\Delta x}$,
- Forward difference $\frac{(G(x+\Delta x,y)-G(x,y))}{\Delta x}$,
- Central difference $\frac{(G(x+\Delta x,y)-G(x-\Delta x,y))}{2\Delta x}$,

Filter masks:

- Backward difference ${}^{-}\mathbf{D}_x = \left[-1 \ \dot{1}\right]$.
- Forward difference ${}^{+}\mathbf{D}_x = \left[-\dot{1} \ 1\right]$.
- Central difference $\mathbf{D}_{2x} = \left[-\frac{1}{2} \ 0 \ \frac{1}{2}\right]$.



Linear Filters

Derivatives - Finite Second Order Differences

The approximation of the second derivative in x-direction $\partial^2 G(x,y)/\partial x^2$ (analogously in y-direction $\partial^2 G(x,y)/\partial y^2$) results from a twofold application of the first order differences:

$$\mathbf{D}_x^2 = {}^{-}\mathbf{D}_x \star {}^{+}\mathbf{D}_x = \begin{bmatrix} -1 & \dot{1} \end{bmatrix} \star \begin{bmatrix} 1 & -\dot{1} \end{bmatrix} = [1 \ \ -2 \ \ 1] \,.$$

Thus, for the discrete Laplacian operator for 2D images, the filter mask follows:

$$\mathbf{L} = \mathbf{D}_x^2 + \mathbf{D}_y^2 = [1 \ \ -2 \ \ 1] + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} .$$

disadvantages

- Poor approximation at large wavenumbers
- Not perfectly isotropic

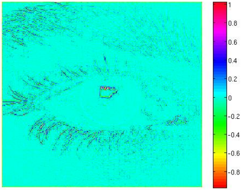
Linear Filters

Technical University
of Applied Sciences
Würzburg-Schweinfurt

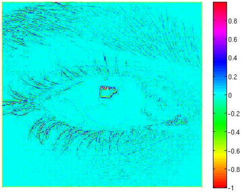
Derivatives - Finite Second Order Differences

An optimized Laplace operator $\mathbf{L}'$ with lower anisotropy is obtained via a linear combination of Binomial filter $\mathbf{B}$ and impulse filter $\mathbf{P}$:

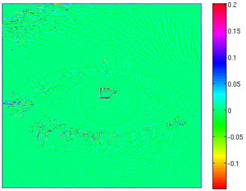
$$\mathbf{L}' = 4 * (\mathbf{B} - \mathbf{P}) = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$



$\mathbf{L}$



$\mathbf{L}'$



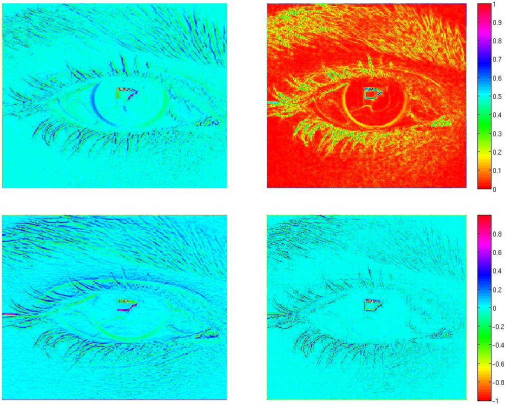
$\mathbf{L} - \mathbf{L}'$

Derivatives - Examples

Comparison of the edge strengths over the magnitude of the gradient vector $\|\nabla\|$ or the zero crossings of the Laplace operator Δ.

$\dfrac{\partial G}{\partial x}$

$\dfrac{\partial G}{\partial y}$



$$\|\nabla\| = \sqrt{\left(\frac{\partial G(\mathbf{x})}{\partial x}\right)^2 + \left(\frac{\partial G(\mathbf{x})}{\partial y}\right)^2}$$

$$\Delta = \frac{\partial^2 G(\mathbf{x})}{\partial x^2} + \frac{\partial^2 G(\mathbf{x})}{\partial y^2}$$

Derivatives - Regularized Edge Detection

There is a basic problem that occurs with derivative filters:

- derivatives of noisy signals amplify noise!

The solution or reduction of this problem is achieved by

- a preceding smoothing of the image, i.e. a suppression of high frequencies or noise.

Since smoothing and differentiation are both realized by a convolution, i.e. linear operators, they can be interchanged. That means the derivative of a smoothing filter can be used as a regularizing edge filter e.g.

$$\underbrace{\mathbf{G}}_{\text{image}} * \underbrace{\mathbf{b}^\top}_{\text{binomial filter}} * \underbrace{\mathbf{D}_{2x}}_{\text{x-difference}} = \mathbf{G} * \underbrace{\mathbf{D}_{2x} * \mathbf{b}^\top}_{\text{differentiated smoothing}} = \mathbf{G} * \underbrace{\mathbf{S}_x}_{\text{Sobel-x-Operator}}$$

Technical University of Applied Sciences Würzburg-Schweinfurt

Derivatives - Regularized 1st order derivative

There are several ways to design regularized derivative filters. They differ in the quality of the approximation properties, such as isotropy and transfer function.

Sobel Edge Detector

$$\mathbf{S}_x = \mathbf{D}_{2x} * \mathbf{b}^\top = \frac{1}{2}[1 \;\; 0 \; -1] * \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{S}_y = \mathbf{D}_{2y} * \mathbf{b} = \frac{1}{4}[1 \;\; 2 \;\; 1] * \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Disadvantage

Too strong smoothing perpendicular to the direction of derivation.

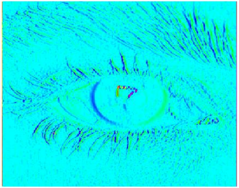
Linear Filters

Derivatives - Regularized 1st order derivative

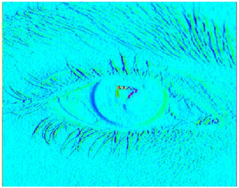
Optimized Sobel Edge Detector

$$\mathbf{S}'_x = \frac{1}{4}\mathbf{D}_{2x} * (3\mathbf{b}^\top + \mathbf{p}^\top) = \frac{1}{32}\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}, \quad \mathbf{S}'_y = \frac{1}{4}\mathbf{D}_{2y} * (3\mathbf{b} + \mathbf{p}) = \mathbf{S}'^{\top}_x .$$

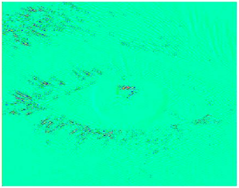
improvement: Optimal cross-smoothing $\rightarrow$ minimum anisotropy.



$\mathbf{S}_x$



$\mathbf{S}'_x$



$\mathbf{S}_x - \mathbf{S}'_x$

Derivatives - Regularized Laplace Operator

Since the transfer function of the Laplace operator is proportional to the square of the wavenumber, the noise susceptibility is very high. A regularization is especially recommended here.

Laplace of Gaussians: LoG-Filter

$$\mathbf{LoG} = \Delta \mathbf{G}(\mathbf{x}) \approx \mathbf{L} * \mathbf{B}^p$$

A good approximation of the LoG filter is provided by the DoG filter.

Difference of Gaussians: DoG-Filter

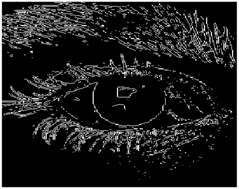
$$\mathbf{DoG} = \mathbf{G}(\mathbf{x}, \sigma_1^2) - \mathbf{G}(\mathbf{x}, \sigma_2^2) \approx 4 * (\mathbf{B}^q - \mathbf{P})\mathbf{B}^p = 4 * (\mathbf{B}^{q+p} - \mathbf{B}^p)\,,$$

where p and q denote the order of the binomial filters and thus approximate Gaussian smoothening with different variances σ^2.

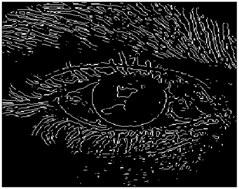
Edge Detectors - Nonlinearities and Examples

The last step is to evaluate the gradient strength or slope in the zero crossings. For this purpose two additional nonlinear operations are needed:

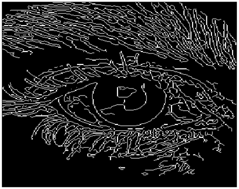
- threshold operation and
- possibly additional hysteresis (two thresholds, see upcoming lecture).



Sobel + Threshold



LoG + Threshold



Canny + Hysteresis

Nonlinear Filters

Ranking Filter - Principle

Ranking filters are an important class of nonlinear filters. Instead of generating a filter result by

- weighting and adding the gray values in a neighborhood, as is the case with linear filters,

the filter result here is given by

- compare and select the gray values in a neighborhood.

Since ranking filters do not perform arithmetic operations, but select gray values, no rounding problems arise, as in linear filter design. There is always a discrete set of gray values mapped to itself.
All ranking filters are based on an ascending or descending sorted list of gray values within the neighborhood under consideration. The neighborhood $\mathcal{N}_{\mathbf{p}}$ need not be rectangular, can take any local form and be location dependent.

Nonlinear Filters

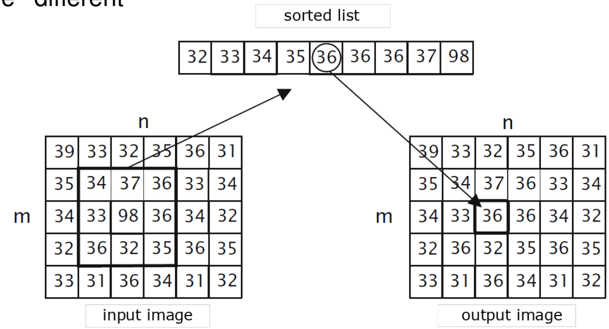
Ranking Filter - Types

Depending on the filter type, either the mean, the maximum or the minimum of the sorted list is selected as the filter result:

$$G(\mathbf{p}) = \mathcal{O}\{G(\mathbf{p} - \mathbf{p}')|\mathbf{p}' \in \mathcal{N}_{\mathbf{p}}\}\,, \quad \text{with} \quad \mathcal{O} = \{median,\, min,\, max\}\,.$$

Thus, there are three different types of ranking filters:

- Median-Filter ,
- Minimum-Filter,
- Maximum-Filter.



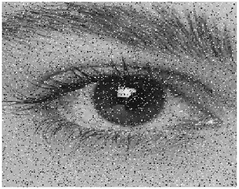
Nonlinear Filters

Median Filter - Properties

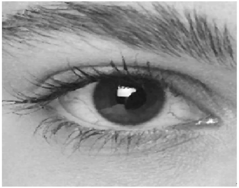
The median filter has the following special property. It

- suppresses salt & pepper distributed noise,
- without smoothing prominent edges too much.

The larger the neighborhood is chosen, the more fine structures are lost without losing high-contrast edges.



Salt & Pepper Noise



3x3 Median- Filter

19x19 Median- Filter

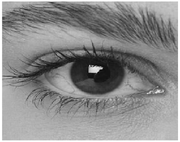
Nonlinear Filters

Minimum Filter - Properties

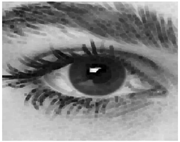
Local Minimum Operator: $\quad G(\mathbf{p}) = min\{G(\mathbf{p}-\mathbf{p}')|\mathbf{p}' \in \mathcal{N}_{\mathbf{p}}\}$.

Effects of the size and shape of the filter mask $\mathcal{N}_{\mathbf{p}}$ on

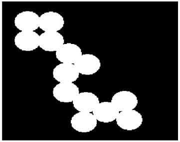
- greyscale image: local minima are extended to the shape of the filter mask $\mathcal{N}_{\mathbf{p}}$, dark contours become fattened.
- binary image: The size of objects is reduced. Objects smaller than the size of the filter mask disappear, object touches are separated.



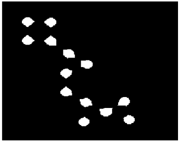
Original



Min-Filter



Binary image



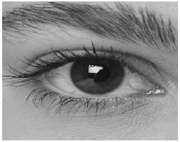
Min-Filter

Maximum Filter - Properties

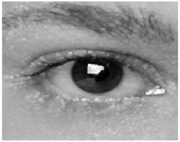
Local Maximum Operator: $G(\mathbf{p}) = max\left\{G(\mathbf{p}-\mathbf{p}')|\mathbf{p}' \in \mathcal{N}_\mathbf{p}\right\}$.

Effects of the size and shape of the filter mask $\mathcal{N}_\mathbf{p}$ on

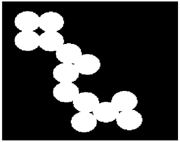
- greyscale image: local maxima are extended to the shape of the filter mask $\mathcal{N}_\mathbf{p}$, dark contours become thinned out.
- binary image: The size of objects is expanded, small holes or cracks are filled.



Original



Max-Filter



Binary image

Max-Filter

Morphological Operators - Definition

Morphological operators influence the shape of objects via reference forms **M**. The minimum and maximum operators form the two basic operations from which different composite operators can be generated. For binary images the minimum and maximum operators correspond to the following set operations:

- Erosion: $\mathbf{G} \ominus \mathbf{M}$ (Set of all pixels, for which **M** is completely contained in **G**.)
- Dilation: $\mathbf{G} \oplus \mathbf{M}$ (set of all pixels for which the intersection of **M** and **G** is not the empty set).

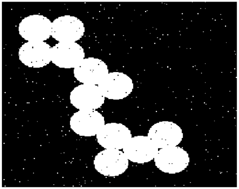
Common set operations are:

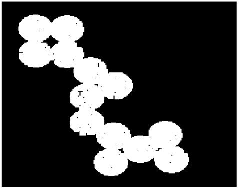
- Open: $(\mathbf{G} \ominus \mathbf{M}) \oplus \mathbf{M}$,
- Close: $(\mathbf{G} \oplus \mathbf{M}) \ominus \mathbf{M}$,
- Extraction of Edges: $\mathbf{G} \cap (\overline{\mathbf{G}} \oplus \mathbf{M})$.

Nonlinear Filters

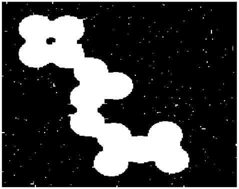
Morphological Operators - Examples

Original

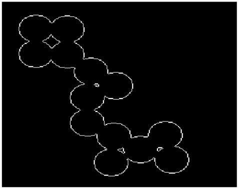




Open



Close



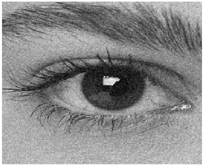
Extraction of Edges

Nonlinear Filters

Normalized Convolution - Principle and Example

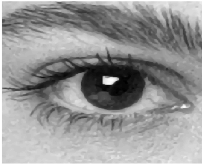
With a normalized convolution, the influence of the convolution operation can be varied locally. In the special case of a smoothing mask this corresponds to a locally varying weighted averaging. For each gray value $G(m, n)$ a corresponding weighting factor $W(m, n)$, which varies the influence of the filter **H** on the gray value is needed. The nonlinear calculation rule is:

$$\mathbf{G}' = \frac{\mathbf{H} * (\mathbf{W} \cdot \mathbf{G})}{\mathbf{H} * \mathbf{W}} .$$



Noisy Original

Weighting



Normalized Convolution

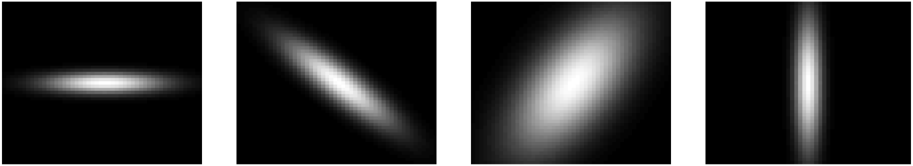
Averaging

Motivation - Example: Anisotropic Gaussian

Idea: The convolution mask $\mathbf{H_p}$ shall change depending on the local image structure in order to optimally adapt the filter properties to certain properties of the local image structure.

Problem: The filter masks $\mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})}$ must all be calculated and saved! It is advantageous to calculate the filter coefficients using functions with freely adjustable parameters $\boldsymbol{\alpha}(\mathbf{p})$ depending on the location.

Example: $\mathcal{N}(\mathbf{x}|\mathbf{0}, \Sigma(\boldsymbol{\alpha})) \propto e^{\mathbf{x}^T \mathbf{R}^T(\theta)\boldsymbol{\Lambda}(\sigma_i)\mathbf{R}(\theta)\mathbf{x}}$.



$\theta = 0^\circ, \sigma_1^2 = 100, \sigma_2^2 = 10.$ $\theta = 45^\circ, \sigma_1^2 = 150, \sigma_2^2 = 10.$ $\theta = 120^\circ, \sigma_1^2 = 250, \sigma_2^2 = 50.$ $\theta = 90^\circ, \sigma_1^2 = 200, \sigma_2^2 = 5.$

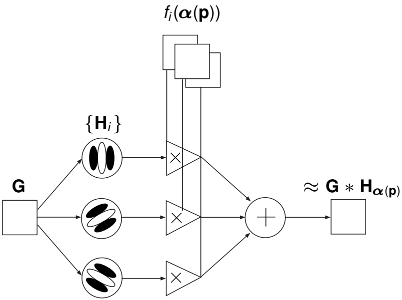
Steerable Filters

Motivation

Idee: The convolution mask $\mathbf{H}_\mathbf{p}$ shall depend on the local image structure in a certain neighborhood $\mathcal{N}_\mathbf{p}$ around the image point $\mathbf{p}$, in order to adapt the filter properties optimally to certain properties of the local image structure.

Definition: Steerable filters $\mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})}$ have some freely adjustable parameters $\boldsymbol{\alpha}(\mathbf{p})$ that control the filtering depending on the location.

Goal: Construct a set of a few basic filters $\mathbf{H}_i$, which can be used to obtain any response of a steerable filter via a $\boldsymbol{\alpha}$-dependent interpolation of the filter responses of the basic filters.



Steerable Filters

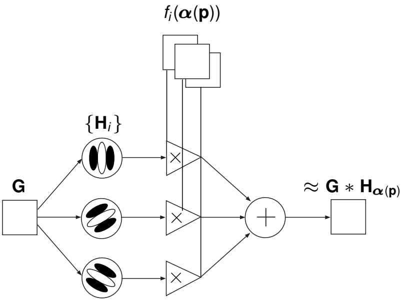
Design

Thus, the goal is to design a set of basic filters $\mathbf{H}_i$ and interpolation functions $f_i(\boldsymbol{\alpha}(\mathbf{p}))$, so that the controllable filter $\mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})}$ is approximated as well as possible:

$$\mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})} \approx \sum_{i=1}^{N} \overbrace{f_i\big(\boldsymbol{\alpha}(\mathbf{p})\big)}^{?} \overbrace{\mathbf{H}_i}^{?} \,.$$

An adaptive filter can be implemented very efficiently with a steerable filter $\mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})}$:

$$\mathbf{G} * \mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})} \approx \sum_{i=1}^{N} f_i\big(\boldsymbol{\alpha}(\mathbf{p})\big)\Big(\mathbf{G} * \mathbf{H}_i\Big) \,.$$



Steerable Filters

Examples - Adaptive Anisotropic Smoothing

Because of the linearity property of the Fourier transform, for any steerable filter holds:

$$\mathbf{H}_{\boldsymbol{\alpha}(\mathbf{p})} \approx \underbrace{\sum_{i=1}^{N} f_i\big(\boldsymbol{\alpha}(\mathbf{p})\big)\mathbf{H}_i}_{\text{image space}} \quad \circ\!\!-\!\!\bullet \quad \underbrace{\sum_{i=1}^{N} f_i\big(\boldsymbol{\alpha}(\mathbf{p})\big)\hat{\mathbf{H}}_i}_{\text{frequency space}} \approx \hat{\mathbf{H}}_{\boldsymbol{\alpha}(\mathbf{p})}$$

Therefore, the basic filters of a steerable filter can be designed using its transfer function.

Example of a transfer function of a directional smoothing filter in polar coordinates (k, θ) in Fourier space:

$$H(k, \theta) = 1 - f(k)\cos^2(\theta - \theta_0)\,.$$

The freely selectable parameter θ_0 determines the direction in which not to smooth. The radial function $f(k)$ determines the cutoff frequency.

Steerable Filters

Examples - Adaptive Anisotropic Smoothing

This results in the following decomposition into basic filters $\hat{H}_i$ and interpolation functions $f_i(\theta_0)$:

$$H_{\theta_0}(k,\theta) = \underbrace{1}_{f_1(\theta_0)} \cdot \underbrace{\left(1 - \frac{1}{2} f(k)\right)}_{\hat{H}_1} + \underbrace{\cos(2\theta_0)}_{f_2(\theta_0)} \cdot \underbrace{\left(-\frac{1}{2} f(k)\cos(2\theta)\right)}_{\hat{H}_2} + \underbrace{\sin(2\theta_0)}_{f_3(\theta_0)} \cdot \underbrace{\left(-\frac{1}{2} f(k)\sin(2\theta)\right)}_{\hat{H}_3} .$$

Through optimization, a basic set of 3×3 large separable filters can be found in image space:

$$\mathbf{H}_1 = \frac{1}{32}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 20 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \mathbf{H}_2 = \frac{1}{32}\begin{bmatrix} 0 & -4 & 0 \\ 4 & 0 & 4 \\ 0 & -4 & 0 \end{bmatrix}, \mathbf{H}_3 = \frac{1}{32}\begin{bmatrix} -2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & -2 \end{bmatrix} .$$

With this filter set, directionality decreases at higher frequencies.

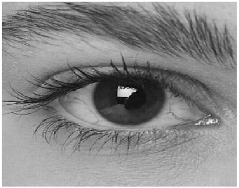
Steerable Filters

Examples - Adaptive Anisotropic Smoothing

In the simplest case, the angle θ_0 can be calculated by maximizing the directional derivative by direction:

$$\frac{\partial}{\partial\theta}\left(\cos(\theta)\frac{\partial G(\mathbf{x})}{\partial x} + \sin(\theta)\frac{\partial G(\mathbf{x})}{\partial y}\right) \stackrel{!}{=} 0\,.$$

$$\tan(\theta) = \frac{\partial G(\mathbf{x})}{\partial y}\left(\frac{\partial G(\mathbf{x})}{\partial x}\right)^{-1} \quad \leftrightarrow \quad \theta_0 = \tan^{-1}\left(\frac{\partial G(\mathbf{x})}{\partial y}\left(\frac{\partial G(\mathbf{x})}{\partial x}\right)^{-1}\right)\,.$$



Original

Steerable Smoothing

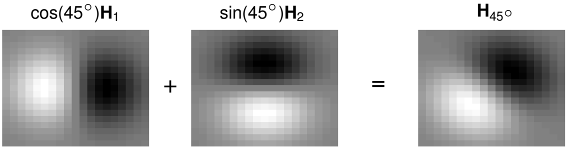
Isotropic Smoothing

Steerable Filters

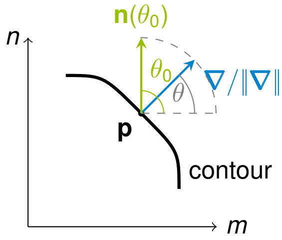
Examples - Adaptive Directional Derivative

The directional derivative along the direction θ_0 is in itself steerable!

$$\mathbf{H}_{\theta_0(\mathbf{p})} = \nabla^\top \mathbf{n}(\theta_0) = \sum_{i=1}^{2} f_i(\theta_0)\mathbf{H}_i = \underbrace{\cos(\theta_0)}_{f_1(\theta_0)} \underbrace{\frac{\partial G(\mathbf{x})}{\partial x}}_{\mathbf{H}_1} + \underbrace{\sin(\theta_0)}_{f_2(\theta_0)} \underbrace{\frac{\partial G(\mathbf{x})}{\partial y}}_{\mathbf{H}_2} .$$



Example of a filter of a Gaussian-smoothed directional derivative in direction $\theta_0 = 45^\circ$.



Steerable Filters

Examples - Adaptive Directional Derivative

Filter bank for seven different directional derivatives calculated from only two derivatives $\mathbf{G} * \mathbf{H}_{0^\circ}$ and $\mathbf{G} * \mathbf{H}_{90^\circ}$.