

Image
Derivative
Convolution

Leonhard
Applis

What
makes
an Edge?
Problems
Definition

Gradient
based
approach
Local
maxima
method

1D
approach
2D
Approach
Simple
Example
Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Comments

Edge Detection

Leonhard Applis

TH Nürnberg

05.11.2018

Table of Contents

1 What makes an Edge?

- Problems
- Definition

2 Basics of gradient-based edgedetection

- 1D approach
- 2D Approach
- Simple Example
- Filters

3 Advanced gradient-based edgedetection

- 3x3 Filters
- 3x3 Examples
- Evaluations

4 Compass Operators

5 Edge Sharpening

6 Summary

What makes an edge?

Image
Detection
101

Leonhard
Applis

What
makes
an Edge?

Problems
Definition

Convolution
Kernel
Based
Approach
Gradient
Magnitude

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters

3x3
Examples

Evaluations

Comments



Figure: Felix

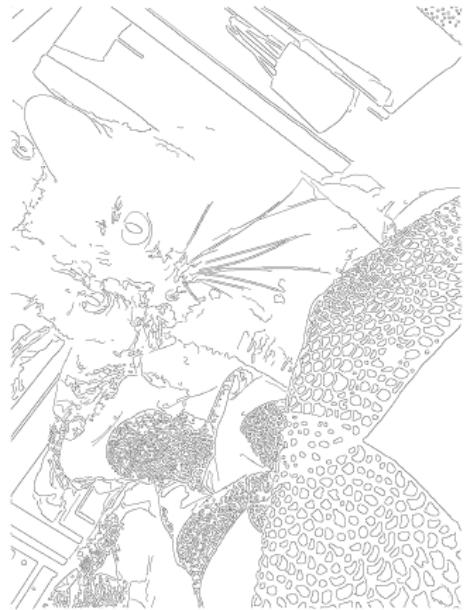


Figure: Felix's Edges

Problem I: Contrast



Figure: High Contrast Felix



Figure: Low Contrast Felix

Home
Contact
About

Leonhard
Applis

CV
Publications
Teaching

Problems

Definition

Convolutional
Neural Network
Based
Approach
Implementation

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradients
based
edge detec-
tion

3x3 Filters
3x3
Examples
Evaluations

Comments

Problem II: Smoothness



Figure: Smooth Felix

Leonhard
Applis

Convolution
and backprop
gradient

Problems
Definition

Convolution
and backprop
gradient
filter
blurring
edge detec

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Comments

Problem III: Noise



Figure: Salted Felix

Leonhard
Applis

Convolution
Cross-correlation
Gradients
Max pooling
Softmax

Problems
Definition

Classification
Regression
Classification
Regression
Classification
Regression

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradients
based
edge detec-
tion

3x3 Filters
3x3
Examples
Evaluations

Comments

Definition

In Image Processing, an edge can be defined as a set of contiguous pixel positions where an abrupt change of intensity, gray- or color-values occur. Edges represent boundaries between objects and background. Sometimes, the edge-pixel-sequence may be broken due to insufficient intensity difference.(Malay K. Pakhira)

Table of Contents

Image
Detection
Definition
**Leonhard
Applis**

What makes
an Edge
Problems
Definition

Basics of
gradient-
based
edgede-
tection

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient-
based
edgede-
tection

3x3 Filters
3x3
Examples
Evaluations

Compass

1 What makes an Edge?

- Problems
- Definition

2 Basics of gradient-based edgedetection

- 1D approach
- 2D Approach
- Simple Example
- Filters

3 Advanced gradient-based edgedetection

- 3x3 Filters
- 3x3 Examples
- Evaluations

4 Compass Operators

5 Edge Sharpening

6 Summary

One dimensional approach

Image
Derivatives
Convolution

Leonhard
Applis

What
makes
an edge
Problems

Definition

Convolution
Kernel
Filter
Blur
Gaussian

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradients
based
edge de-
tection

3x3 Filters

3x3
Examples

Evaluations

Comments

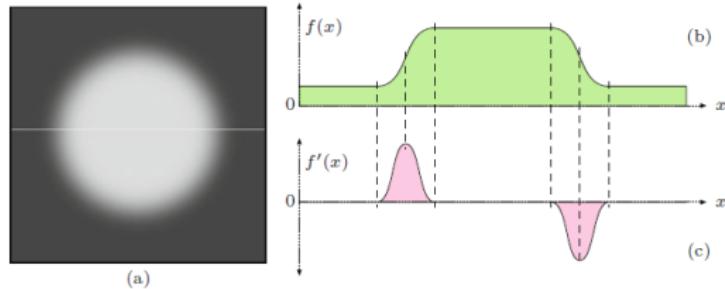


Figure: One dimensional image function and derivation

Only applicable with known, steady functions

Approximating discrete derivation

Leonhard
Applis

What
makes
an image
Problems
Definition

Convolution
Kernel
Filter
Blur
Sharpen

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Problem: the image function is discrete, therefore we need to approximate the derivation

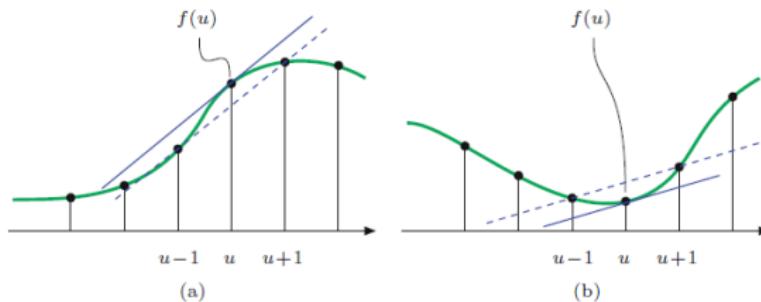


Figure: Approximation of the derivation for discrete imagefunctions

$$\frac{df}{dx}(u) \approx \frac{f(u+1) - f(u-1)}{(u+1) - (u-1)} = \frac{f(u+1) - f(u-1)}{2}$$

Two dimensional approach

If working with full images, we got two dimensions and therefore two partial derivations:

$$I_x = \frac{\partial I}{\partial x}(u, v), I_y = \frac{\partial I}{\partial y}(u, v)$$

the **gradient** at the point (u, v) is

$$\nabla I(u, v) = \begin{pmatrix} I_x(u, v) \\ I_y(u, v) \end{pmatrix}$$

And the **magnitude** is

$$|\nabla I| = \sqrt{I_x^2 + I_y^2}$$

Example

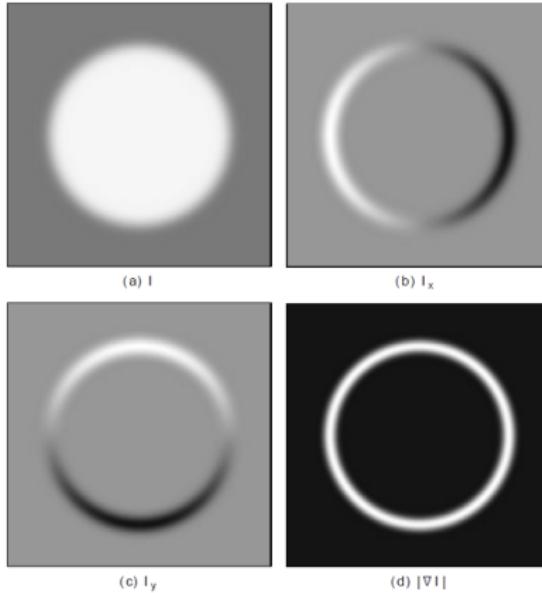


Figure: Visualisation of simple gradient-based edgedetection

Example with Felix



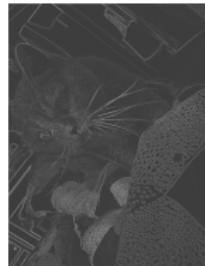
(a) I



(b) I_x



(c) I_y



(d) $|\nabla I|$

Figure: Simple Filters applied to Felix¹

¹ All images have a 50% increased brightness

Implementation with filters

Expressing the gradient as a *linear filter* is simple:

$$I_x = [-0.5 \quad 0 \quad 0.5]$$

$$I_y = \begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Leonhard
Applis

What
makes
an edge
Problem
Definition

Convolution
Kernel
Input
Output
Filter
Definition

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Comments

Table of Contents

Image
Detection
Definition
**Leonhard
Applis**

What
makes
an Edge
Problems
Definition

Convolution
Kernel
Filter
Blur
Dilation
Erosion
1D
approach

2D
Approach
Simple
Example
Filters

Advanced
gradient-
based
edgede-
tection

3x3 Filters
3x3
Examples
Evaluations

Compass

1 What makes an Edge?

- Problems
- Definition

2 Basics of gradient-based edgedetection

- 1D approach
- 2D Approach
- Simple Example
- Filters

3 Advanced gradient-based edgedetection

- 3x3 Filters
- 3x3 Examples
- Evaluations

4 Compass Operators

5 Edge Sharpening

6 Summary

Prewitt Operator

Idea: Include neighborhood

$$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\nabla I^P(u, v) \approx \frac{1}{6} \cdot \begin{pmatrix} (I * H_x^P)(u, v) \\ (I * H_y^P)(u, v) \end{pmatrix}$$

Sobel

Idea: Include neighbourhood but weight center more

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\nabla I^S(u, v) \approx \frac{1}{8} \cdot \begin{pmatrix} (I * H_x^S)(u, v) \\ (I * H_y^S)(u, v) \end{pmatrix}$$

Comparison with Felix

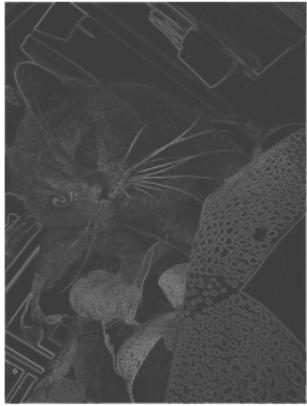


Figure: Simple edge filter

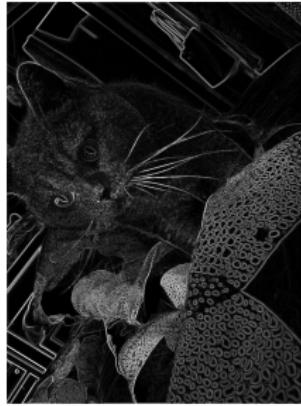


Figure: Prewitt Operator



Figure: Sobel Operator

Evaluations

general magnitude: $E(u, v) = \sqrt{I_x^2(u, v) + I_y^2(u, v)}$
holds for every Operator

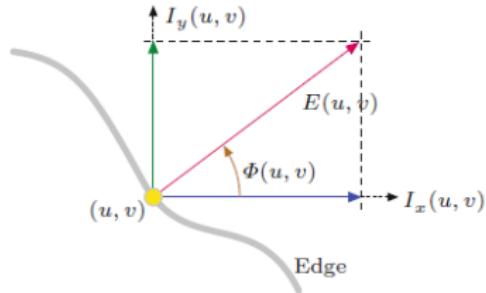


Figure: Visualisation of edge direction

$$\Phi(u, v) = \tan^{-1} \left(\frac{I_y(u, v)}{I_x(u, v)} \right) = \arctan(I_x(u, v), I_y(u, v))$$

Table of Contents

Image
Detection
Definition
**Leonhard
Applis**

What
makes
an Edge
Problems
Definition

Convolution
Kernel
Filter
Blur
Dilation
Erosion

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edgedetec
tion

3x3 Filters
3x3
Examples
Evaluations

Compass

1 What makes an Edge?

- Problems
- Definition

2 Basics of gradient-based edgedetection

- 1D approach
- 2D Approach
- Simple Example
- Filters

3 Advanced gradient-based edgedetection

- 3x3 Filters
- 3x3 Examples
- Evaluations

4 Compass Operators

5 Edge Sharpening

6 Summary

Compass Operators

Leonhard
Applis

What
makes
an image
Problem
Definition

Convolution
Kernel
Filter
Blur
Gaussian
Blur

1D
approach
2D
Approach
Simple
Example
Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

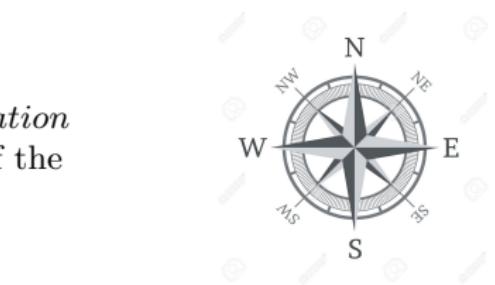
Compass

Problem: The stronger a filter responses to edge-like structures, the more sensitive it is to orientation

$\uparrow \text{Orientation} \approx \downarrow \text{Edges}$ $\uparrow \text{Edges} \approx \downarrow \text{Orientation}$

Solution: Apply 8 filters for each direction of the compass

Produce two pictures: One for the edge-strength taking the magnitude of the strongest activated filter,
one for the direction marking the strongest filter with colour



Extended Sobel Operator

$$H_0^{\text{ES}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad H_1^{\text{ES}} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix},$$
$$H_2^{\text{ES}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad H_3^{\text{ES}} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix},$$
$$H_4^{\text{ES}} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad H_5^{\text{ES}} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix},$$
$$H_6^{\text{ES}} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad H_7^{\text{ES}} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}.$$

Note: Its only required to apply 4 filters, as they are symmetric

Figure: Extended Sobel Operator

Extended Sobel Example

Image
Filters
Evaluations

Leonhard
Applis

What
makes
an image
interesting

Problems
Definition

Convolution
with filters
Input
Output
Forward
backward

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters

3x3
Examples

Evaluations

Compass



Figure: $\Phi^{ES}(Felix)$

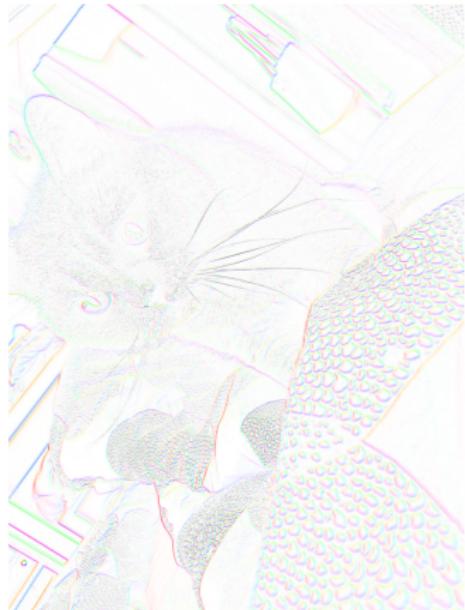


Figure: Colour encoding

Kirsch Operator

Image
Detection
Orientation
Tracking

Leonhard
Applis

What
makes
an edge
Problems

Definition

Convolution
Kernel
Implementation
Edge
Detection

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Compass

$$H_0^K = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}, \quad H_4^K = \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix},$$
$$H_1^K = \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}, \quad H_5^K = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{bmatrix},$$
$$H_2^K = \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}, \quad H_6^K = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix},$$
$$H_3^K = \begin{bmatrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix}, \quad H_7^K = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}.$$

Figure: Kirsch Operator



Figure: Felix with Kirsch

Canny-Edge Operator

- ① Pre-Processing
- ② Edge Localization
- ③ Edge Tracing and hysteresis thresholding

For complex applications the Canny-Edge-Algorithm is done with more than 3x3 Filters

Canny-Edge Pre-Processing

Image
Detection
Canny

Leonhard
Applis

What
makes
an Edge

Problems
Definition

Convolution
Sobel
Gaussian
Laplacian
Gaussian

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Compass

- ➊ Smooth Image with Gaussian-Filter
- ➋ Apply simple filters (See 14)
- ➌ Calculate magnitude

Also required: σ as Gauss-Radious and t_1, t_2 for thresholding edges

Canny-Edge Edge-Localization

Introduction
Definition
Implementation

Leonhard
Applis

What
makes
an Edge?

Problems
Definition

Gradient
Magnitude
Based
Edge
Localization

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge
detection

3x3 Filters
3x3
Examples
Evaluations

Compass

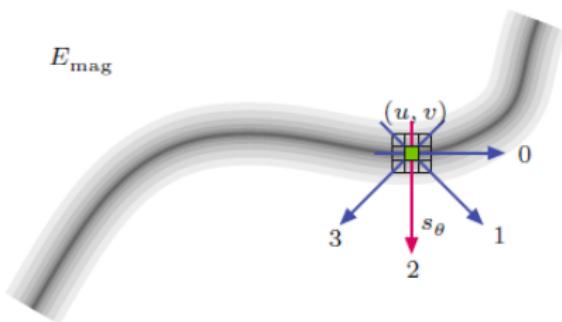


Figure: Edge-Localization

Operate over the magnitude-image:
Null every value that is not a local-maximum in any direction

Canny-Edge

Hysteresis Thresholding

Image
Detection
Hough

Leonhard
Applis

What
makes
an edge

Problems
Definition

Connected
Components
Contour
Detection
Depth
Information

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters

3x3
Examples

Evaluations

Compass

- ① Find points with $E \geq t_1$
- ② Add Every Neighboor with $E \geq t_2$
- ③ Repeat 2 until image is processed



Figure: Canny-Edge Felix

Table of Contents

Image
Processing
Tutorial

Leonhard
Applis

What
makes
an Edge?

Problems
Definition

Convolution
Kernel
Based
Approach
Definition

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Compass

1 What makes an Edge?

- Problems
- Definition

2 Basics of gradient-based edgedetection

- 1D approach
- 2D Approach
- Simple Example
- Filters

3 Advanced gradient-based edgedetection

- 3x3 Filters
- 3x3 Examples
- Evaluations

4 Compass Operators

5 Edge Sharpening

6 Summary

Laplacian-Filter

Concept

Leonhard
Applis

What
makes
an edge
Problems

Definition

Convolution
Kernel
Input
Output
Derivatives

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Idea: Alter *steep* imagepoints

$$\hat{f}(x) = f(x) + \omega \cdot f''(x)$$

Where ω is a weight for alternation,
 $f(x)$ is the image function

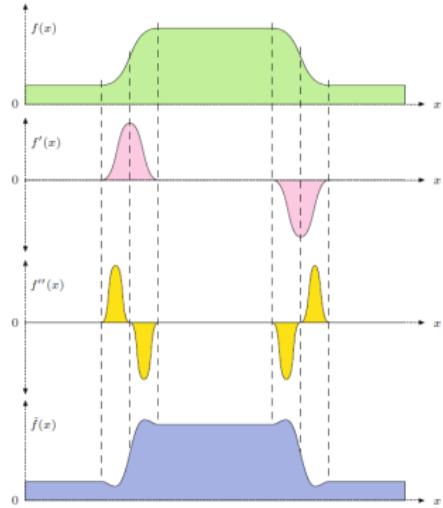


Figure: Applying the Laplacian-Filter

Laplacian-Filter Implementation

Laplacian Operator: $(\nabla^2 f)(x, y) = \frac{\partial^2 f}{\partial^2 x}(x, y) + \frac{\partial^2 f}{\partial^2 y}(x, y)$

Expressed as filters:

$$\begin{aligned}\frac{\partial^2 f}{\partial^2 x}(x, y) &\approx [1 \quad -2 \quad 1], \quad \frac{\partial^2 f}{\partial^2 y}(x, y) \approx \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \\ &\Rightarrow H^L \approx \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}\end{aligned}$$

Leonhard
Applis

What
makes
an image
Problem

Problems
Definition

Gradients
Edges
Corner
Dilate
Erode

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters
3x3
Examples
Evaluations

Comments

Laplacian-Filter Variants and Example

Image
Derivatives
Convolution
Blur

Leonhard
Applis

What
makes
an image
an edge?

Problems

Definition

Convolution
Kernel
Blur
Gaussian
Blur
Median
Blur

1D
approach

2D
Approach

Simple
Example

Filters

Advanced
gradient
based
edge de-
tection

3x3 Filters

3x3
Examples

Evaluations

Comments

common variants:

$$H^L \approx \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$H^L \approx \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



Figure: Laplacian Filter applied to Felix

Unsharp Masking (USM)

Concept

Leonhard
Applis

- ➊ produce a smoothed version of the image
- ➋ subtract the smoothed version from original
→ **mask** of *sharp* points
- ➌ add the mask with a weight α to original image

Any smoothing-filter can be used, but gaussian filters are common
Notable:

- ➊ highly adaptable through exchangeable filters and α
- ➋ Laplacian filters are a implementation of USM with very simple smoothing
- ➌ vulnerable to *boost* noise, therefore often combined with a threshold

Table of Contents

- 1 What makes an Edge?
 - Problems
 - Definition
- 2 Basics of gradient-based edgedetection
 - 1D approach
 - 2D Approach
 - Simple Example
 - Filters
- 3 Advanced gradient-based edgedetection
 - 3x3 Filters
 - 3x3 Examples
 - Evaluations
- 4 Compass Operators
- 5 Edge Sharpening
- 6 Summary

Summary

- Edges are found via gradient-based filters
- Most algorithms are simple
- Most algorithms require thresholds from user
- Best results are achieved when combined with smoothing
- Iteratively trying multiple variants can improve results
- Images with sharpened edges shouldn't be trusted 100%

Primary Sources

Digital Image Processing - An Algorithmic Introduction Using Java
2nd Edition

Wilhelm Burger, Mark J.Burge

Source for every non-felix picture.

User Guide and Documentation for HIPR

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/>

Sometimes longer explanations and additional variants of the filters