

Bildverarbeitung

Visual Computing

Winter Semester 2023-2024



Prof. Dr. A. Kuijper

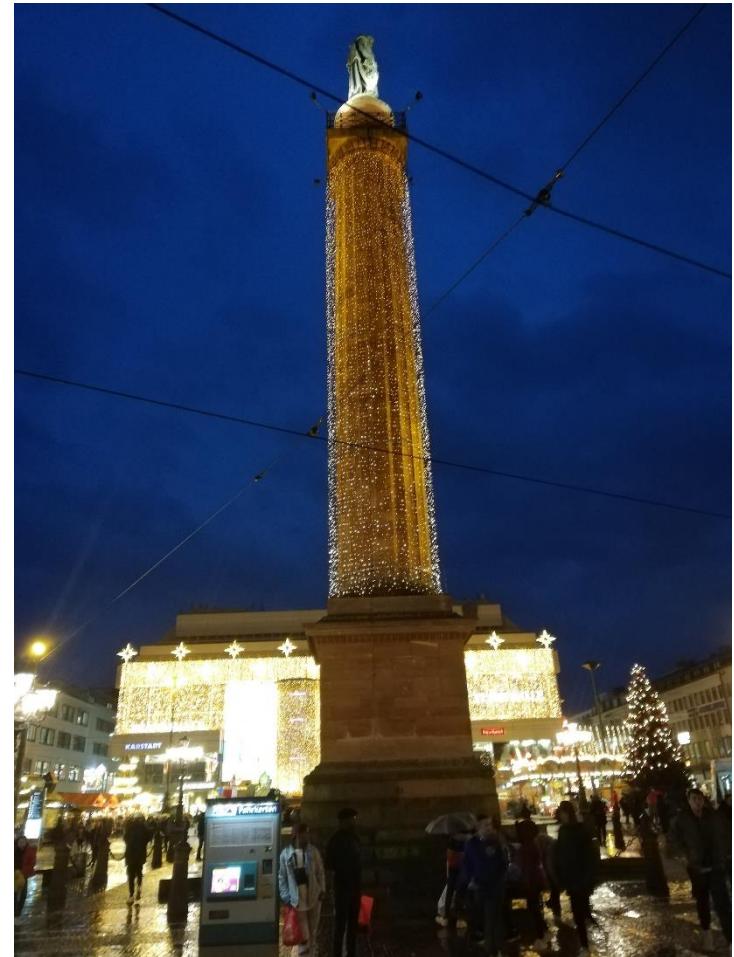
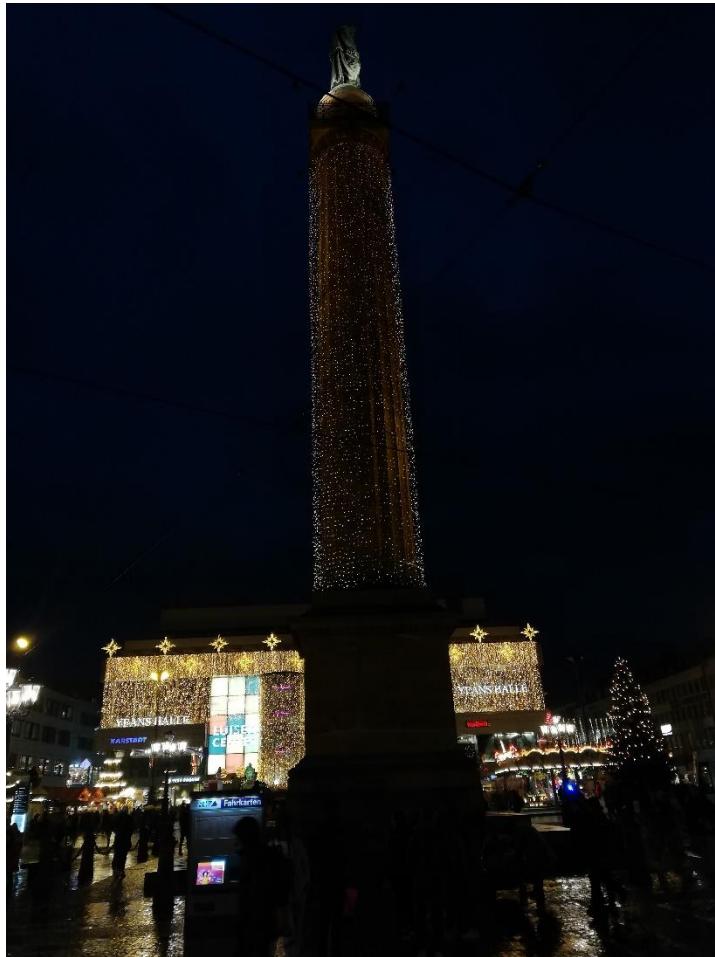
Mathematical and Applied Visual Computing (MAVC)
Graphisch-Interaktive Systeme (GRIS)
Fraunhofer IGD
Fraunhoferstrasse 5
D - 64283 Darmstadt

E-Mail: office@gris.tu-darmstadt.de
<http://www.gris.tu-darmstadt.de>
<https://www.mavc.tu-darmstadt.de>

HDR: High Dynamic Range „Automatische Wertfensterung“



TECHNISCHE
UNIVERSITÄT
DARMSTADT



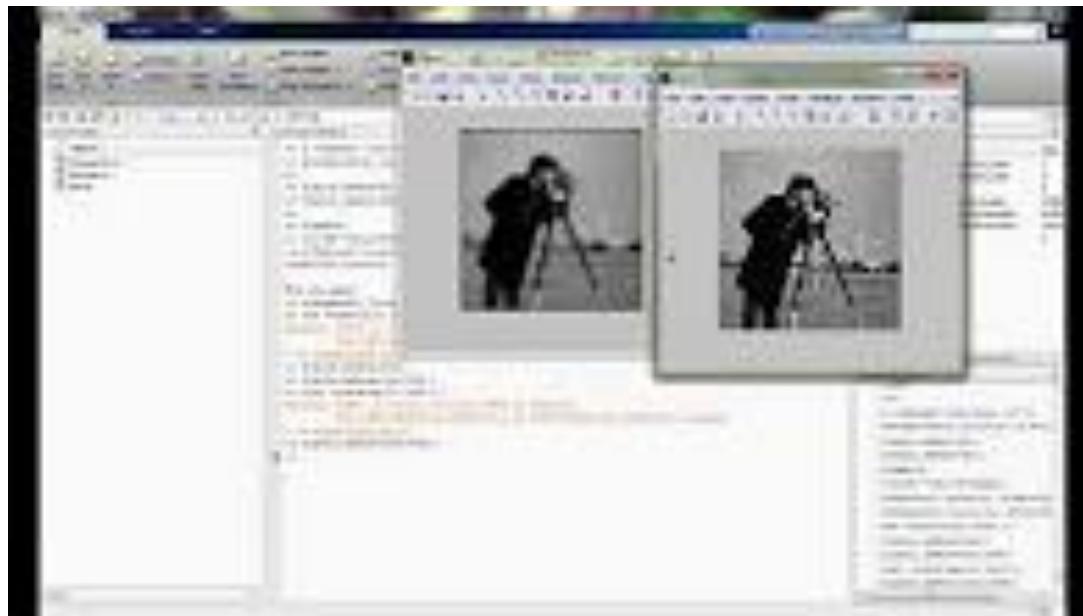
Gaussian Filter revisited



TECHNISCHE
UNIVERSITÄT
DARMSTADT

“We add a Gaussian noise and remove it using Gaussian filter and Wiener filter using Matlab”

<https://www.youtube.com/watch?v=q6fn-i16h20>



Wahrnehmung



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Laplacian Filter



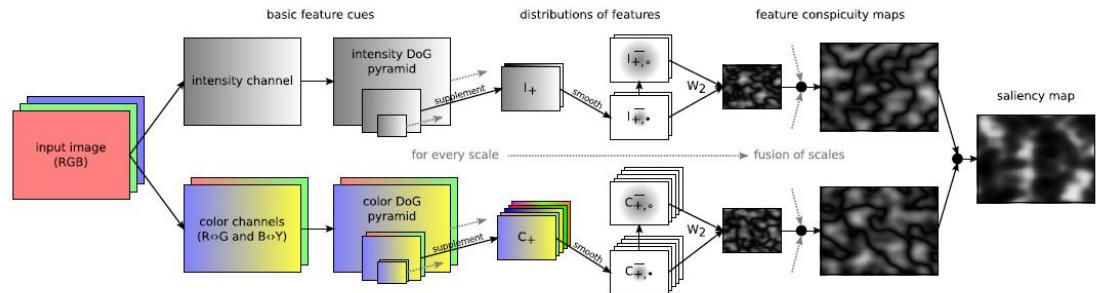
TECHNISCHE
UNIVERSITÄT
DARMSTADT





- Farbe / Strukturerkennung:

- Salient Pattern Detection Using W2 on Multivariate Normal Distributions
Dominik Alexander Klein, Simone Frintrop
DAGM, LNCS 7476,
2012, pp 246-255



- JPEG Kompression

- Automated Image Forgery Detection through Classification of JPEG Ghosts
Fabian Zach, Christian Riess,
Elli Angelopoulou
DAGM, LNCS 7476,
2012, pp 185-194



Saliency....



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Extracting salient region for pornographic image detection

ScienceDirect Journals | Books Shopping cart | Sign in | Help

PDF Purchase Export Search ScienceDirect Advanced search

Article outline Show full outline

Highlights
Abstract
Keywords
1. Introduction
2. The proposed ROI detection method
3. Our approach of pornographic image ...
4. Experiments and evaluation
5. Conclusions
Acknowledgements
References

Figures and tables

The figure of the proposed ROI detection strategy for pornographic images.

Input image → Image processing (color, contrast, histogram, intensity) → Detect posture → Detect gender → Detect face → Detect body → Detect clothing → Extract ROI → Saliency detection.

Original images → The results presented in the text and color mode.

Journal of Visual Communication and Image Representation
Volume 25, Issue 5, July 2014, Pages 1130–1135

Extracting salient region for pornographic image detection
Chenggang Clarence Yan^a, Yizhi Liu^b, Hongtao Xie^c, Zhuhua Liao^b, Jian Yin^d

Choose an option to locate/access this article:

Check if you have access through your login credentials or your institution

Show more <http://dx.doi.org/10.1016/j.jvcir.2014.03.005>

Highlights

- A novel approach for ROI detection in pornographic images is put forward.
- A ROI-based codebook algorithm is proposed.
- A hybrid approach of pornographic image detection is explored.

JPEG revisited: MH17

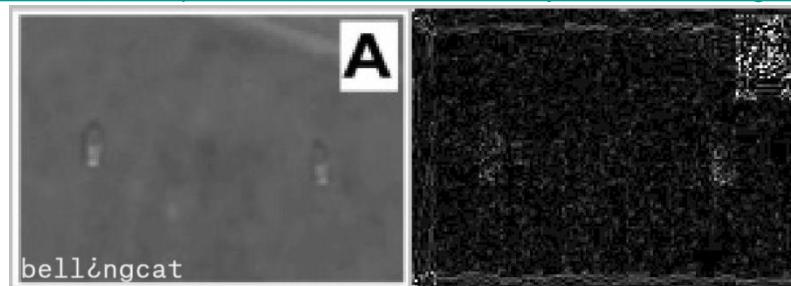


Das russische Verteidigungsministerium behauptete auf dieser Pressekonferenz, dass diese Satellitenfotos Aktivitäten der ukrainischen Luftabwehr am 17. Juli 2014, dem Tag des Abschusses von Flug MH17 in der Ostukraine beweisen.

Insbesondere sollte die Anwesenheit von zwei ukrainischen BUK- Raketenwerfern südlich des Dorfes Zaroschinskoe in Schussposition zu Flug MH17 durch das „Bild 5“ bzw. das hochauflösende „Bild 5-analytics“ und die Abwesenheit von einem Buk-Raketenwerfer auf der Militäreinheit A-1428 (nördlich von Donetsk) durch das „Bild 4“ belegt werden.

Die forensische Analyse aller drei Bilder hat eindeutig und zweifelsfrei ergeben, dass diese Bilder verändert wurden. In den Bildern wurden mit hoher Wahrscheinlichkeit signifikante Bildinhalte digital verändert. „Bild 4“ und „Bild 5“ wurden nachweislich durch die Software Adobe Photoshop CS5 digital modifiziert.

<https://www.bellingcat.com/news/uk-and-europe/2015/05/31/mh17-forensic-analysis-of-satellite-images-released-by-the-russian-ministry-of-defence/>



Belgien



TECHNISCHE
UNIVERSITÄT
DARMSTADT



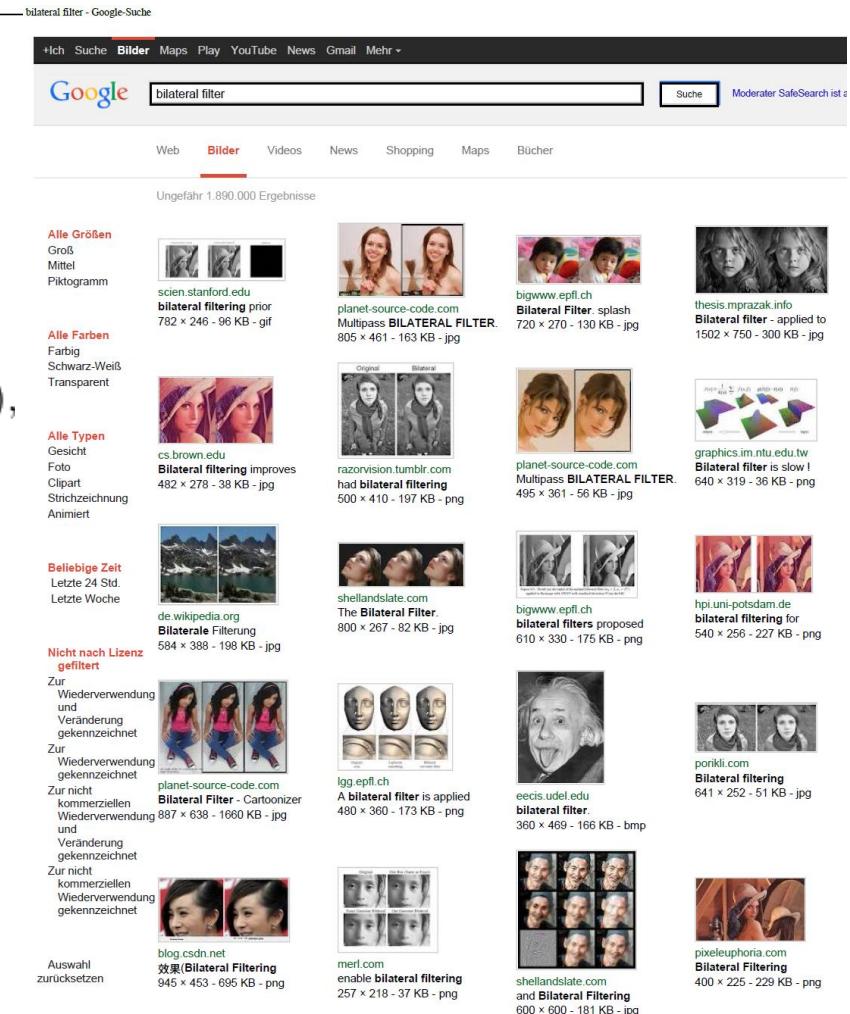
Bilateral Filter



- nichtlinearer Filter:
 - Bilder weichzeichnen
 - Objektkanten erhalten.

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|),$$

- Pixelfarben aus der Nachbarschaft eines Ausgabepixels fließen nicht nur in Abhängigkeit von deren *Entfernung*, sondern auch von deren *Farbabstand* in die Berechnung ein.



Links @Wolfram



TECHNISCHE
UNIVERSITÄT
DARMSTADT

<https://reference.wolfram.com/language/ref/LowpassFilter.html?view=all>

<https://reference.wolfram.com/language/ref/HighpassFilter.html?view=all>

<https://reference.wolfram.com/language/ref/BandpassFilter.html?view=all>

<https://reference.wolfram.com/language/ref/GaussianFilter.html?view=all>

<https://reference.wolfram.com/language/ref/LaplacianGaussianFilter.html?v ieview=all>

<https://reference.wolfram.com/language/ref/DerivativeFilter.html?view=all>

<https://reference.wolfram.com/language/ref/LaplacianFilter.html?view=all>

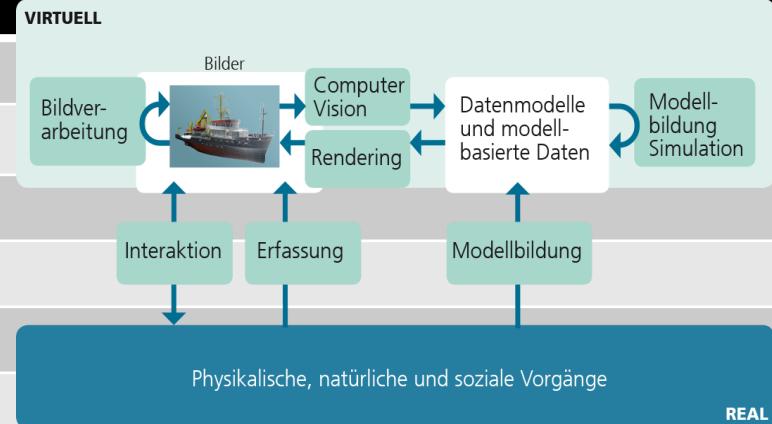
<https://reference.wolfram.com/language/ref/Blur.html?view=all>

<https://reference.wolfram.com/language/ref/GradientFilter.html?view=all>

Semesterplan



Datum	
20. Okt	Einführung + <u>Visual Computing</u>
27. Okt	<u>Wahrnehmung</u>
03. Nov	<u>Objekterkennung</u> und <u>Bayes</u>
10. Nov	<u>Fourier Theorie</u>
17. Nov	<u>Bilder</u>
24. Nov	<u>Bildverarbeitung</u>
01. Dez	Grafikpipeline & Eingabemodalitäten & VR+AR
08. Dez	Transformationen & 2D/3D Ausgabe
15. Dez	3D-Visualisierung
12. Jan	X3D – 3D in HTML
19. Jan	Informationsvisualisierung
26. Jan	Farbe
02. Feb	User Interfaces + Multimedia Retrieval
11. Feb	Biometrie (?)





- **Deblurring von Bildern**
- Einschrittverfahren
 - Wiener Filter
 - Ansatz mit mehreren Komponenten
- Intermezzo
- Mehrschrittverfahren / Iterative Methoden
 - Perona Malik
 - Eingeschränkte Evolution: Totale Variation
- Zusammenfassung

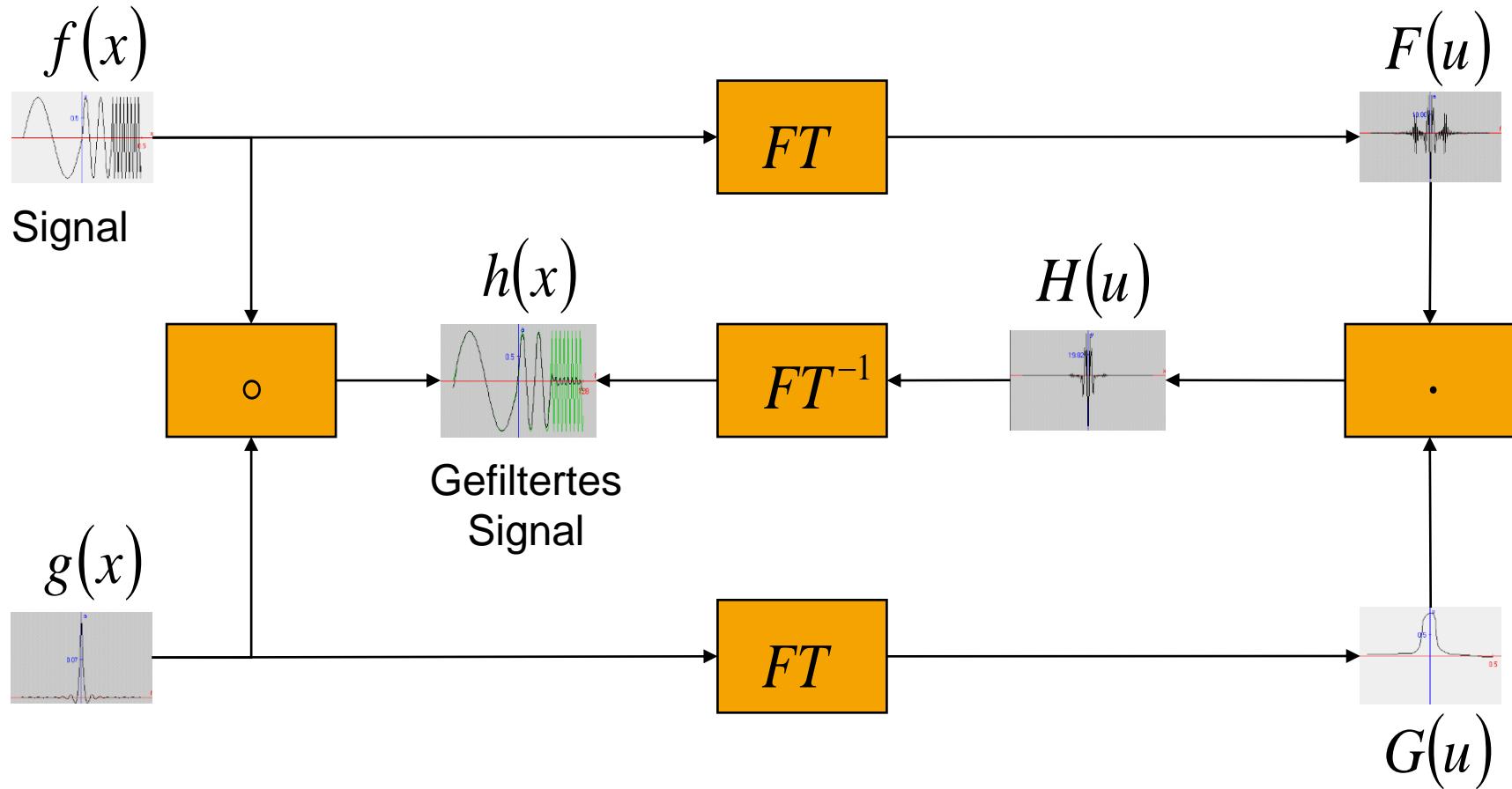
Bildentartung



TECHNISCHE
UNIVERSITÄT
DARMSTADT



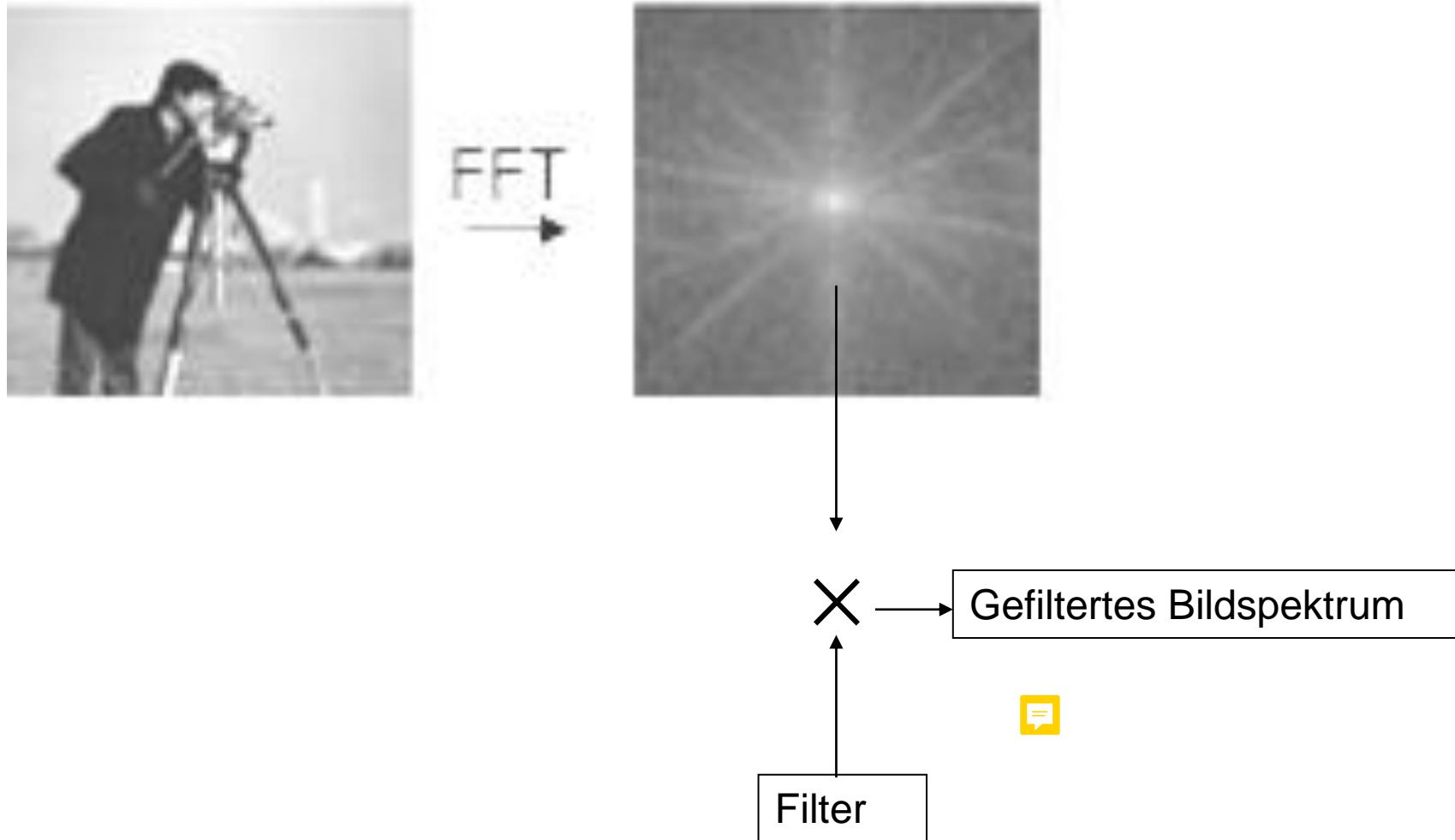
Erinnerung: Faltung und Fourierraum



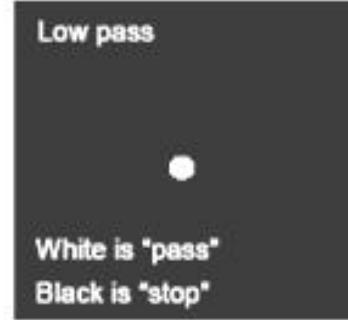
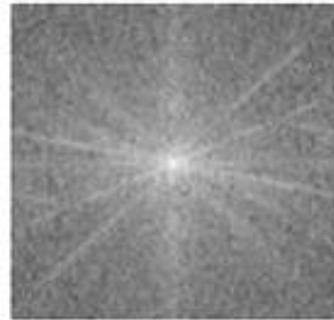
Filterung im Frequenzraum (VL05)



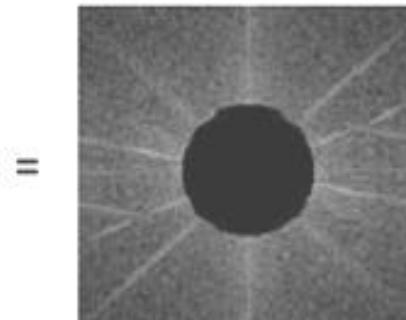
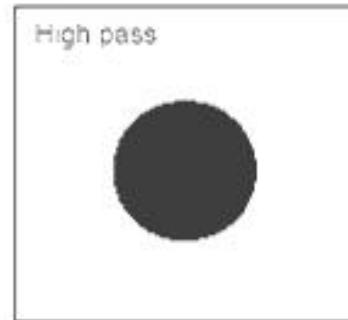
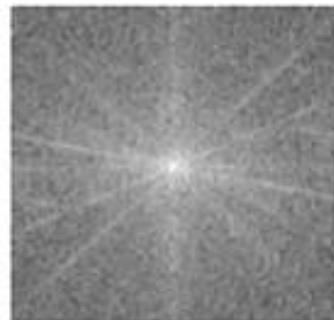
TECHNISCHE
UNIVERSITÄT
DARMSTADT



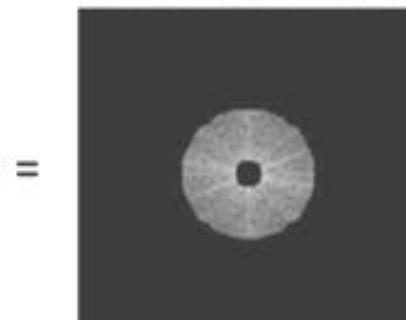
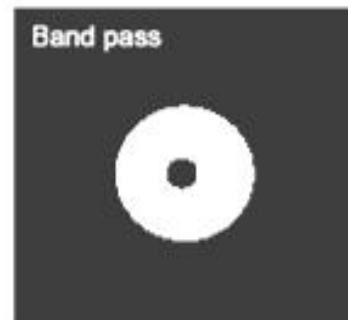
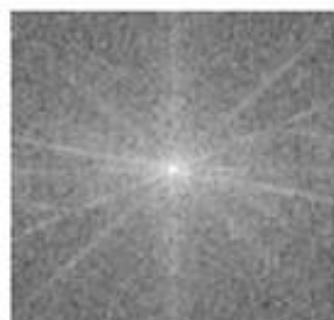
Beispiele (VL05)



IFFT



IFFT



IFFT



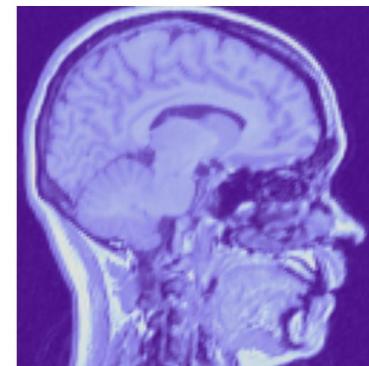


Angenommen, es sei bekannt, dass das erhaltene Bild g die mit einer Faltung mit einer Kernel a verwischte Version (\rightarrow Blurring) eines Bildes f ist: $g = a(f)$

Oft: a = Gaußglocke

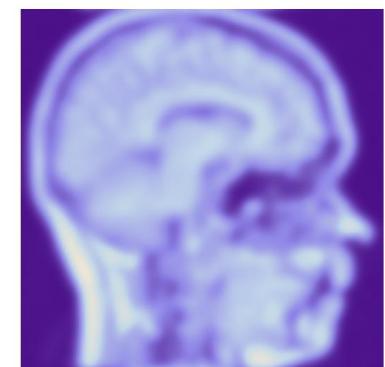
Im Fourierraum: $G = A * F$
 $(A$ = Gaußglocke!)

NB: $G_{ij} = A_{ij} F_{ij}$



f

$g=a(f)$



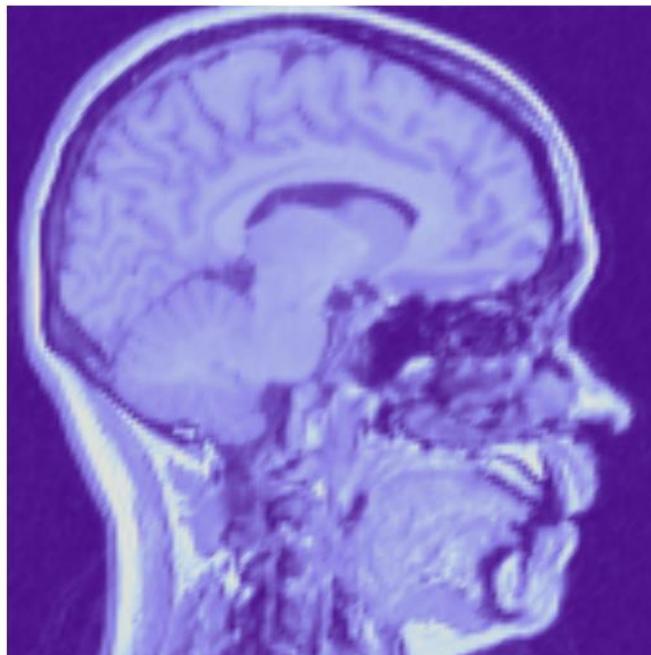
Auf den ersten Blick erscheint die Rekonstruktion
einfach: $F = G/A$

Vergleich: Ohne und mit Blurring

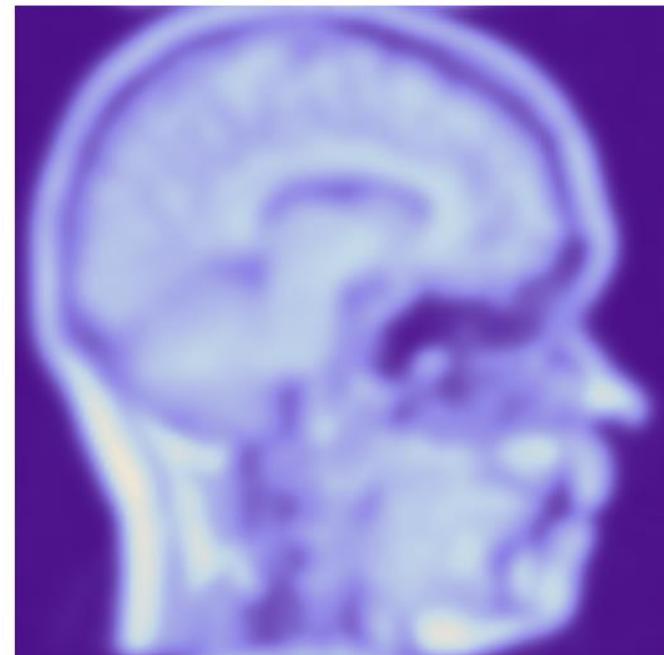


TECHNISCHE
UNIVERSITÄT
DARMSTADT

f

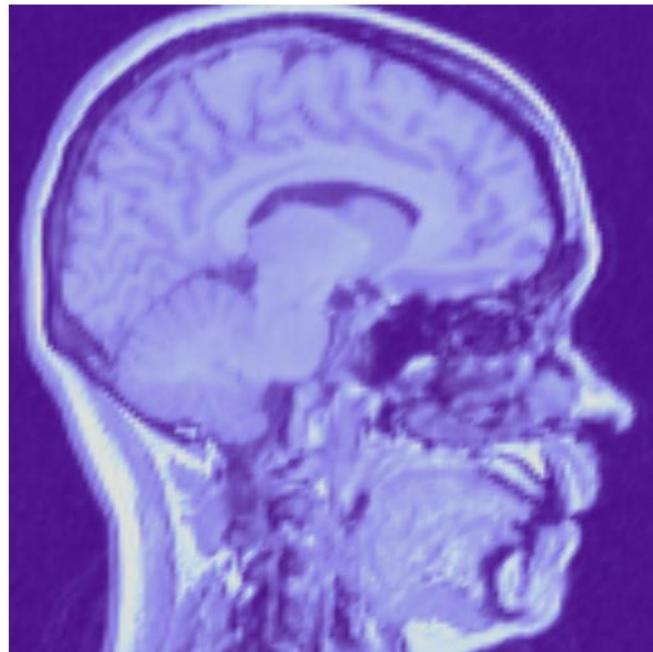


$g=a(f)$





Inverse Fouriertransformation (G/A):



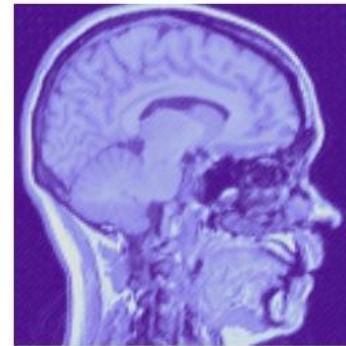
- Cheating!
- Im Gauß Filter beträgt das Minimum $7,6 \times 10^{-447}$, was numerisch unrealistisch ist („infinite precision“)

- Realistisch:
- Werte im Filter abschneiden (zB: 8-bit oder 16-bit Bild).
- Experiment:
 - Wenn Wert in $a < 10^{-10}$ dann Wert = 0

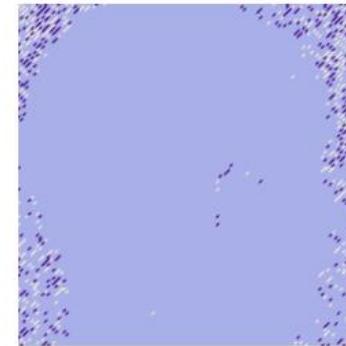


Das rekonstruierte Bild besteht nun, speziell im Hintergrund, aus komplexen Zahlen. Wie in Fourierraum dargestellt als:

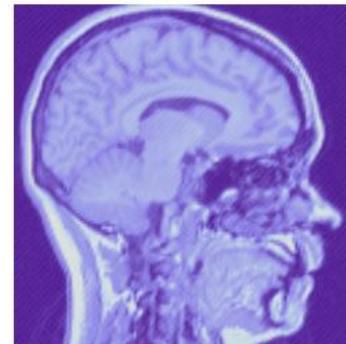
Amplitude



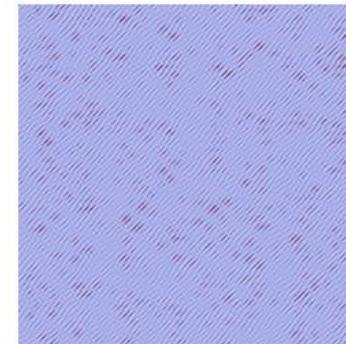
Phase



Realteil



Imaginärteil



(siehe „Fouriertheorie“)

(noch viel schlimmer bei IF Wert < 10^{-3} THEN Wert = 0)

$$F = G/A$$

Problem 1: Der Kernel a hat sehr kleine Werte, sodass es zu (kleine) numerische Fehler in A kommt.

[NB: für g genau so!]

- Beliebige Werte in G werden verstärkt!
- Es gibt komplexe Zahlen in der Rekonstruktion



Rekonstruktion



Problem 1: Der Kernel a hat sehr kleine Werte, sodass es zu (kleine) numerische Fehler in A kommt (komplexe Zahlen in der Rekonstruktion).

Lösung 1: Verwenden der *komplex konjugierten Matrix A^**

$$\begin{aligned} G &= A \cdot F \\ A^* \cdot G &= A^* \cdot A \cdot F \\ &= |A|^2 \cdot F \end{aligned}$$

Mit $|A|^2 \geq 0$ (Annahme: $>0!$) ist dann

$$F = A^{-1}G = \frac{A^*}{A^*A} G = \frac{A^*}{|A|^2} G$$

$$\begin{aligned} z &= a + i b \\ z^* &= a - i b \\ |z|^2 &= z^* z \\ z^* z &= (a + i b)(a - i b) \\ &= a^2 + b^2 \\ &\geq 0 \end{aligned}$$

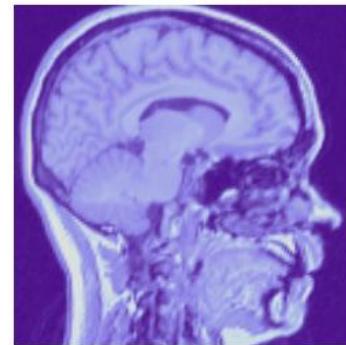
Komplex konjugierter Filter



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Das rekonstruierte Bild hat jetzt keine komplexen Zahlen mehr.

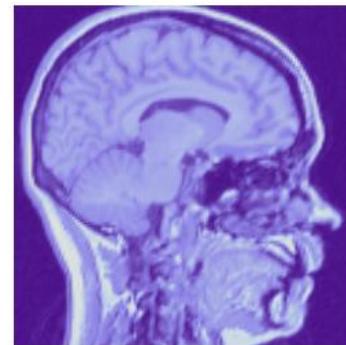
Amplitude



Phase



Realteil



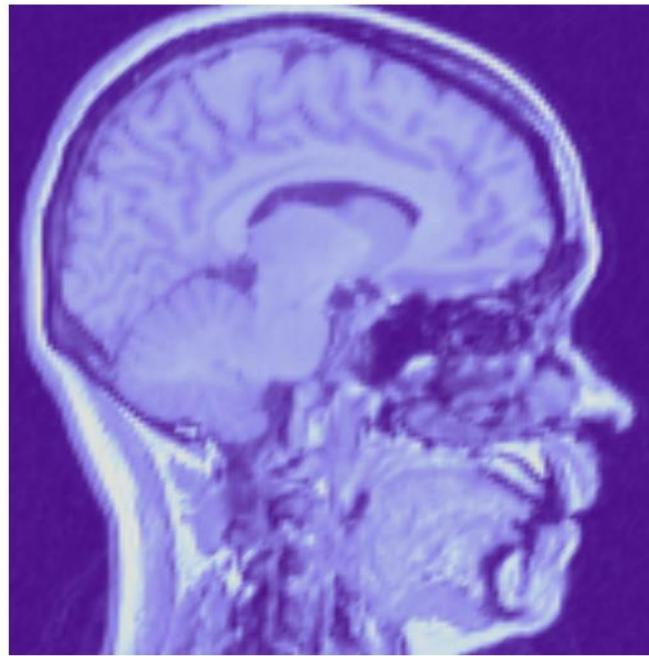
Imaginärteil



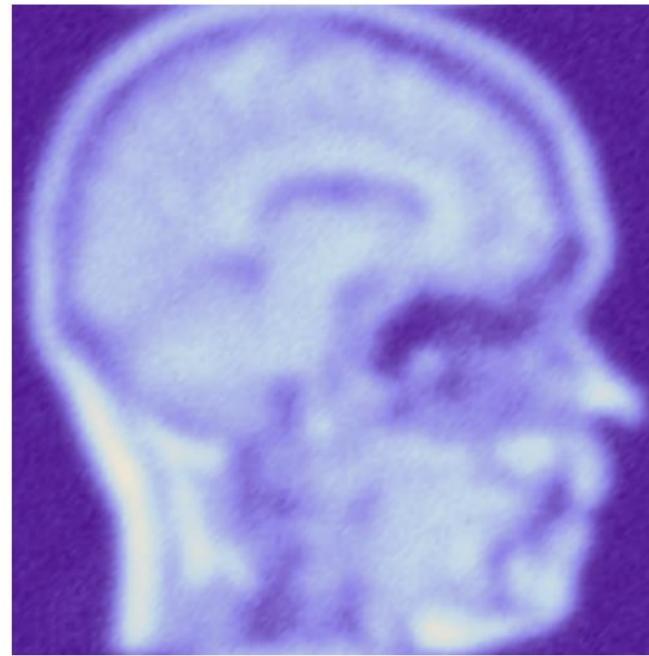


Problem 2: Es gibt immer Rauschen

- $g = a(f) + n$



Originalbild



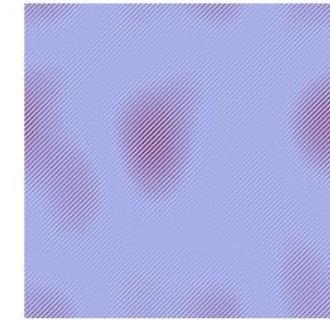
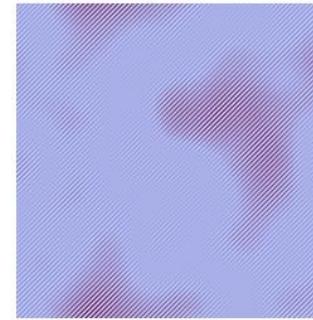
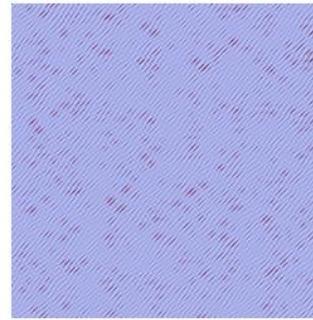
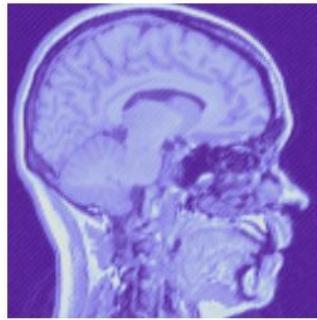
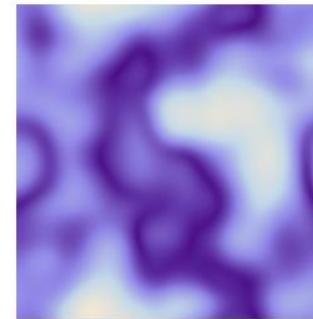
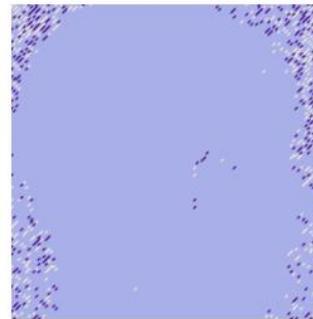
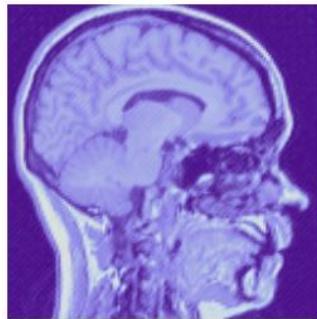
Verwischtes Bild mit Rauschen

Rekonstruktion ohne/mit Rauschen (invers)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Ergebnisse des inversen Filters



Ohne Rauschen

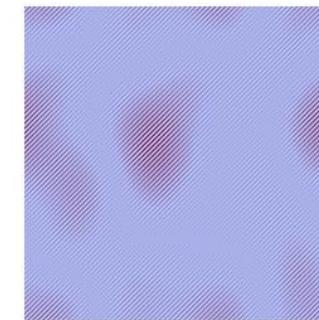
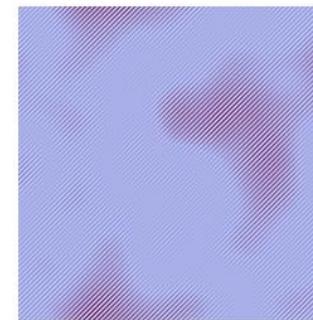
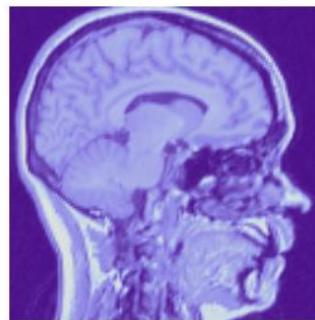
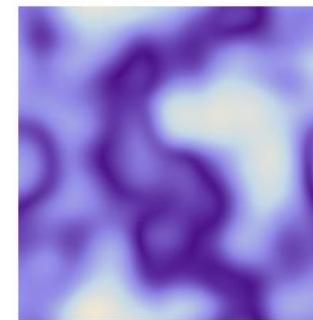
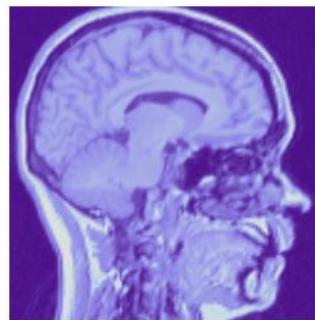
Mit Rauschen

Rekonstruktion ohne/mit Rauschen (komplex konjugiert)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Ergebnisse des komplex konjugierten Filters



Ohne Rauschen

Mit Rauschen

Fazit Deblurring



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Der deblurring Kernel hat sehr kleine Werte, sodass es zu numerische Fehler in der Rekonstruktion kommt.
2. Es gibt immer rauschen

Auswirkung:

Rauschen und numerische Fehler werden verstärkt

Rauschen ist problematisch...

- Ist es überhaupt möglich, Deblurring umzusetzen?

Jacques Hadamard: Korrekt gestellte Probleme



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Ein mathematisches Modell ist **korrekt gestellt**, wenn

- Eine Lösung existiert
- Die Lösung eindeutig ist
- Die Lösung in einer vernünftigen Topologie kontinuierlich von den Daten abhängt

Andernfalls ist das Problem **nicht korrekt gestellt**.

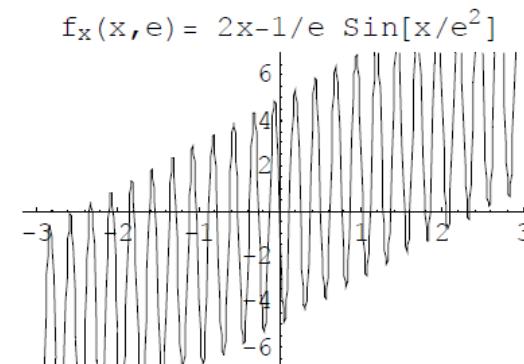
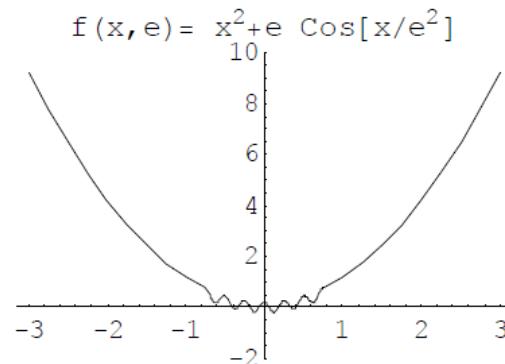
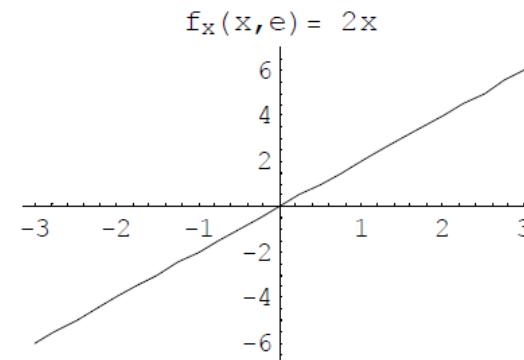
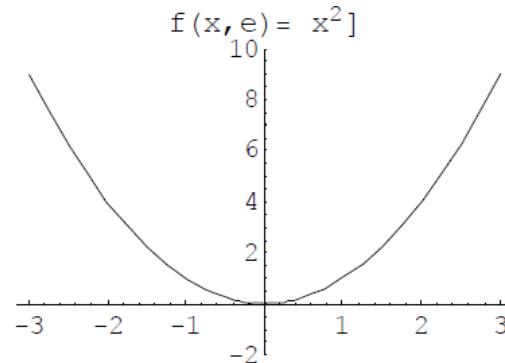


Beispiel: Ableitungen sind nicht korrekt gestellte



Für Ableitungen (nicht korrekt gestellt) gilt:

Das Hinzufügen eines sehr kleinen Anteils an hochfrequentem Rauschen beeinflusst das *Signal* nicht, hat aber großen Einfluss auf die *Ableitung*.



Jacques Hadamard: Korrekt gestellte Probleme



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Konsequenz

- Korrekt gestellt (Blurring):
 - Stabile Algorithmen
 - zB: Rauschen wird geglättet
- Nicht korrekt gestellt (Deblurring):
 - Instabile Algorithmen
 - zB: Rauschen wird verstärkt
- **Regularisierung** ist notwendig:
 - Zusätzliche Annahmen werden hinzugefügt
 - zB: Glätte, Informationen zum Rauschen





- Deblurring von Bildern
- **Einschrittverfahren**
 - **Wiener Filter**
 - Ansatz mit mehreren Komponenten
- Intermezzo
- Mehrschrittverfahren / Iterative Methoden
 - Perona Malik
 - Eingeschränkte Evolution: Totale Variation
- Zusammenfassung



Problem 2: Es gibt immer Rauschen:

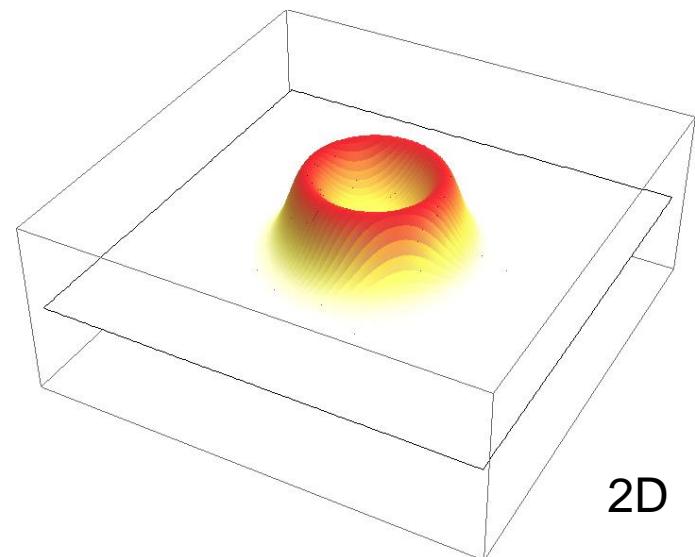
$$g = a(f) + n$$

Lösung 2: Regularisierung des komplex konjugierten Filters im Fourierraum mit \mathbb{R}^2 :

$$F = \frac{A^*}{|A|^2 + R^2} G$$

Das wird "Wiener Filter" genannt
„Verschmiertes Bandpassfilter“
R ist Verhältnis Signal zu Rauschen (SNR)

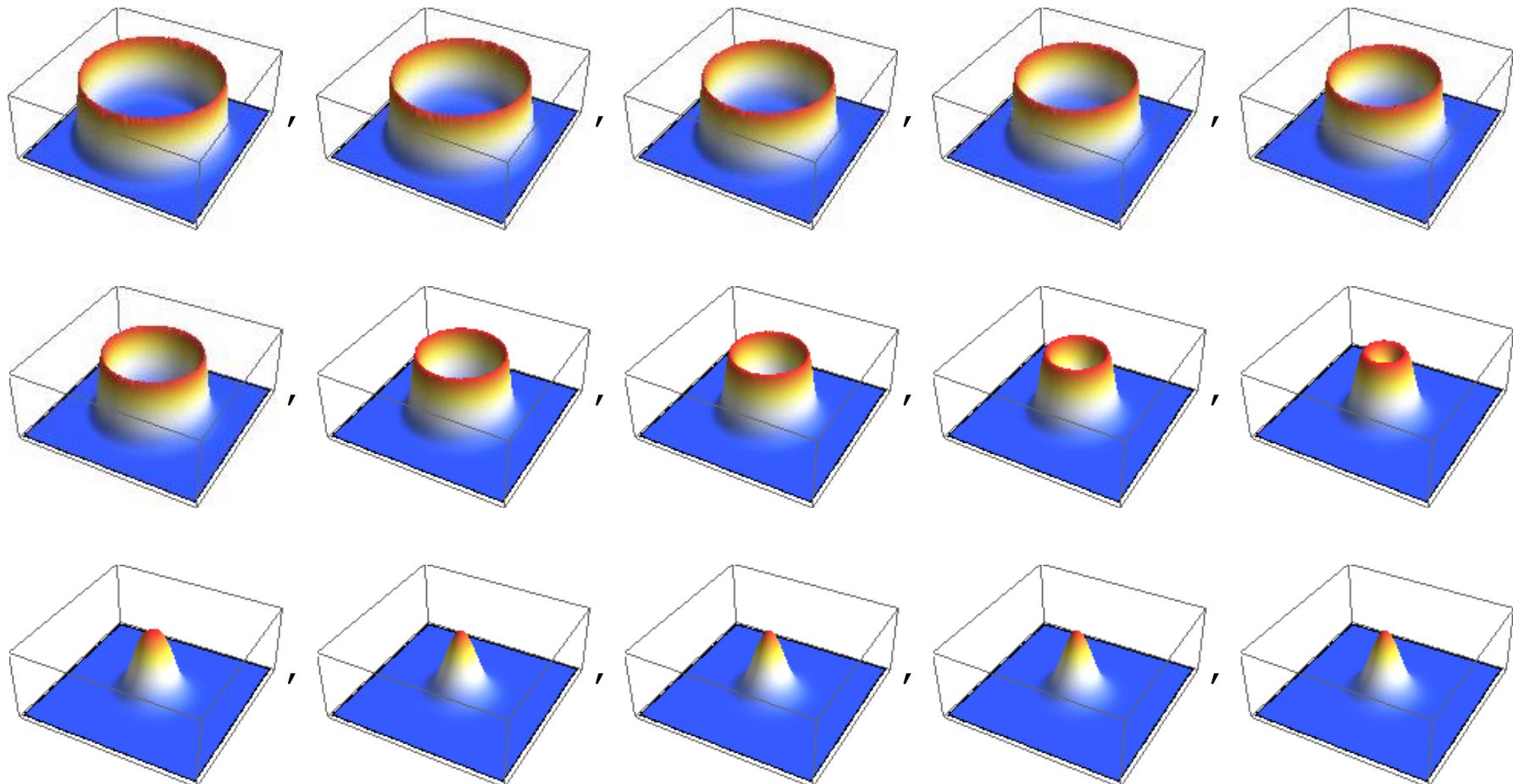
$$a = 1D \text{ Gaußglocke: } WF(u, R) = \frac{e^{-u^2}}{e^{-2u^2} + R^2}$$



2D Wiener Filter für $R^2 = 10^i$, $i = -14, \dots, 0$



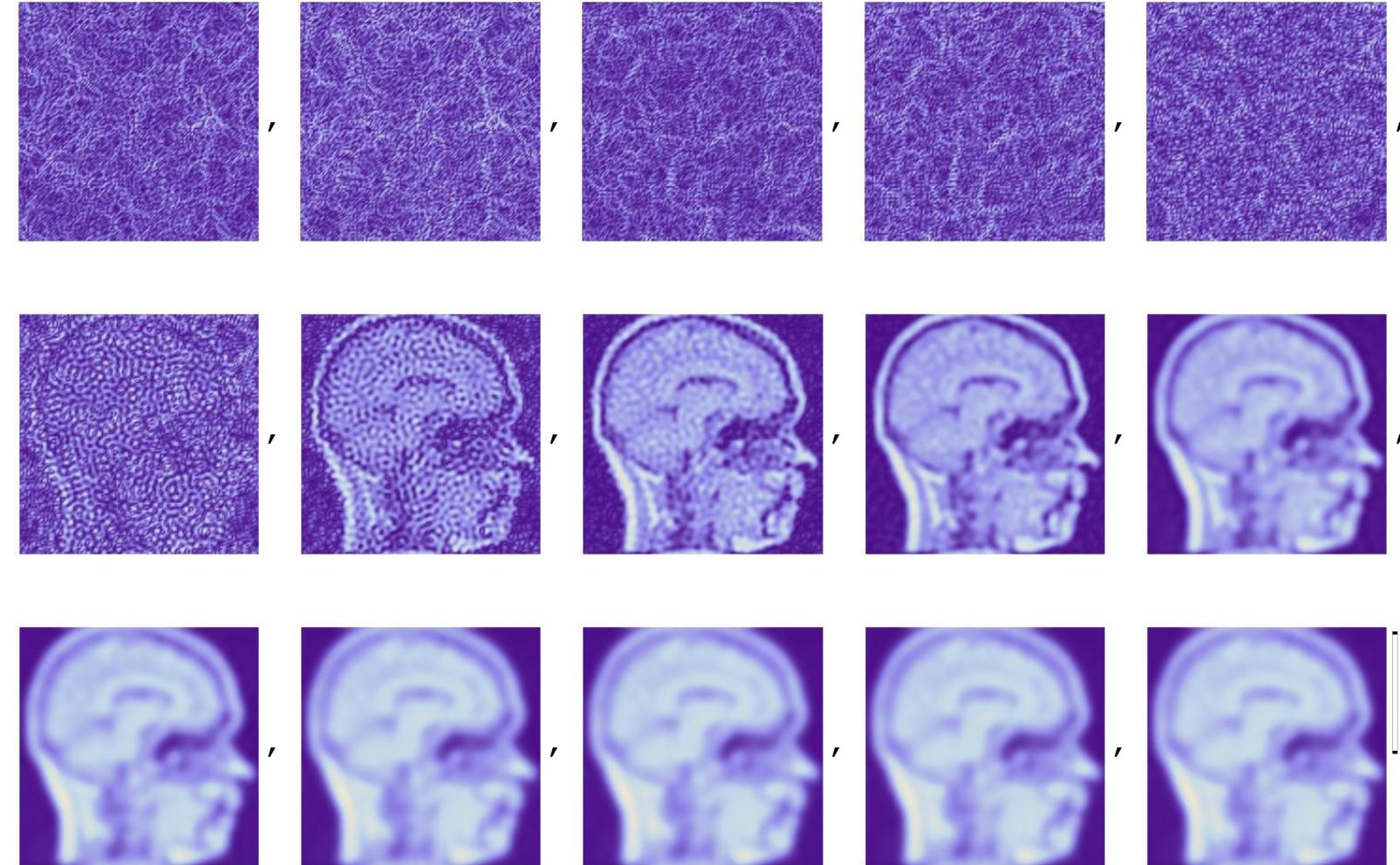
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Rekonstruktionen für $R^2 = 10^i$, $i = -14, \dots, 0$



TECHNISCHE
UNIVERSITÄT
DARMSTADT

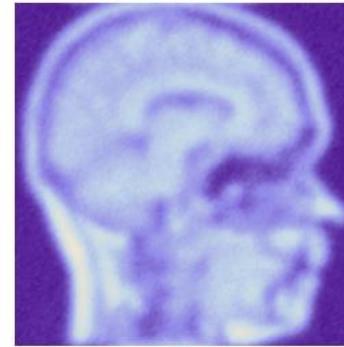
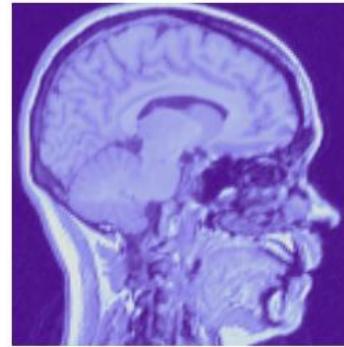


Wiener Filter

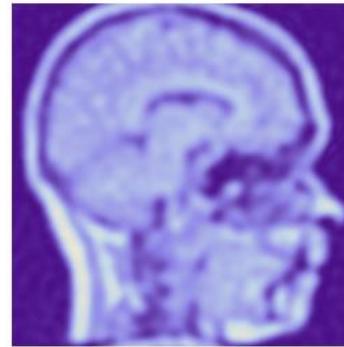


TECHNISCHE
UNIVERSITÄT
DARMSTADT

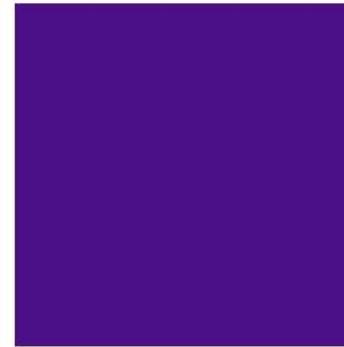
$$F = \frac{A^*}{|A|^2 + R^2} G$$



Rauschen entfernt,
aber immer noch
mit etwas Blurring



Keine komplexen Werte
in der Rekonstruktion



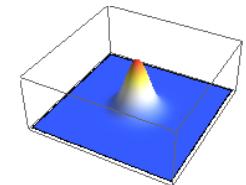
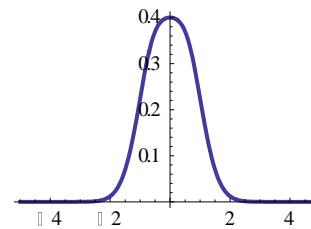
Wiener Filter



Der Parameter R entscheidet was verstärkt wird: muss klug gewählt werden:

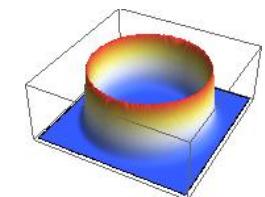
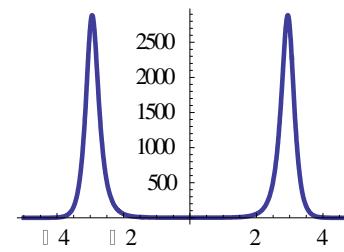
- Zu groß (-> Tiefpass Filter):

- Behaltet grobe Struktur
- Verwischt Kanten
- Entfernt Rauschen



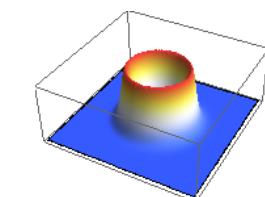
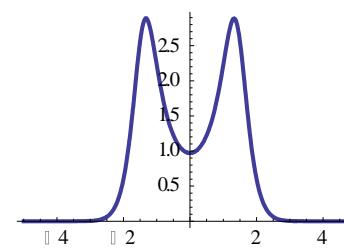
- Zu klein (-> Hochpass Filter):

- Entfernt grobe Struktur & Kanten
- Verstärkt das Rauschen



- Optimal (-> Bandpass Filter):

- Entfernt Rauschen
- Behaltet grobe Struktur
- Verstärkt Kantenstruktur leicht
(deblurring)



Beispiele



TECHNISCHE
UNIVERSITÄT
DARMSTADT

<https://de.mathworks.com/help/images/deblurring-images-using-a-wiener-filter.html>

<https://demonstrations.wolfram.com/ImageRestorationForDegradedImages/>

<https://reference.wolfram.com/language/ref/WienerFilter.html>

<https://docplayer.org/39650231-Bildverarbeitung-5-frequenzraum-und-bildrestauration.html>

Bewertung



Vorteile	Nachteile
<ul style="list-style-type: none">▪ Schnell▪ Häufig verwendet▪ Beliebt▪ Leicht zu implementieren	<ul style="list-style-type: none">▪ Nur ein Filter für das gesamte Bild▪ Keine lokalen, spezifischen Verbesserungen▪ Ein Wert für R

Kann es verbessert werden?

- lokale Verfeinerungen: Ansatz mit mehreren Komponenten
- iterative Verfeinerungen: Mehrschrittverfahren





- Deblurring von Bildern
- **Einschrittverfahren**
 - Wiener Filter
 - **Ansatz mit mehreren Komponenten**
- Intermezzo
- Mehrschrittverfahren / Iterative Methoden
 - Perona Malik
 - Eingeschränkte Evolution: Totale Variation
- Zusammenfassung

Deblurring mit einem Scale-Space-Ansatz



Subtrahieren des Laplace-Operators (siehe „Bilder“)
(multipliziert mit einer vom Blur abhängigen Konstanten t)
verschärft das Bild:

$$L_{Schärfen} = L_0 - t(L_{xx} + L_{yy}) = L_0 - t\Delta L$$

Deblurring mit einem Scale-Space-Ansatz



Subtrahieren des Laplace-Operators (siehe „Bilder“)
(multipliziert mit einer vom Blur abhängigen Konstanten t)
verschärft das Bild:



$$L_{Schärfen} = L_0 - t(L_{xx} + L_{yy}) = L_0 - t\Delta L$$

Das Hinzufügen von zusätzlichen Termen (mit Ableitungen höherer Ordnung) und Generieren einer Sequenz verfeinert das Ergebnis.

$$f(x) = \frac{4k}{\pi} \left(\sin(x) + \frac{1}{3} \sin(3x) + \dots + \frac{1}{25} \sin(25x) \right)$$

$$L[x, y, 0] - t(L_{xx} + L_{yy}) + \frac{1}{2}t^2(L_{xxxx} + 2L_{xxyy} + L_{yyyy})$$

$$- \frac{1}{6}t^3(L_{xxxxxx} + 3L_{xxxxyy} + 3L_{xxyyyy} + L_{yyyyyy})$$

Ergebnis Kein Rauschen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

order 4



order 8



order 16



order 32



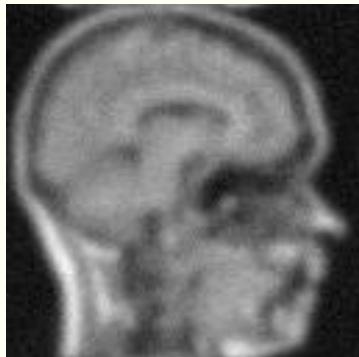
Ergebnis

Mit Gauß- und Schwarz-/Weiß-Rauschen

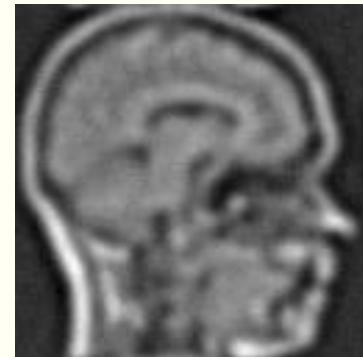


TECHNISCHE
UNIVERSITÄT
DARMSTADT

noisyblur



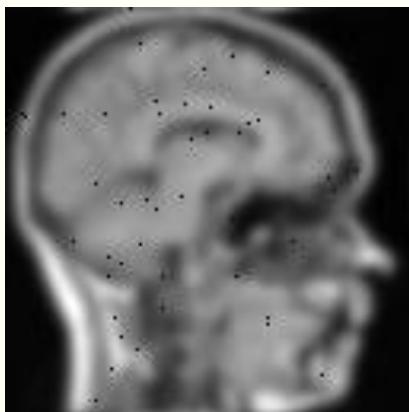
order $\square 4$



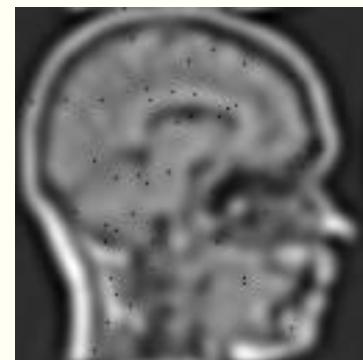
order $\square 8$



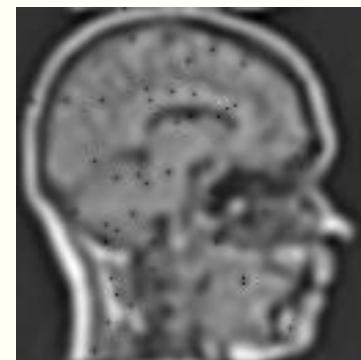
order $\square 16$



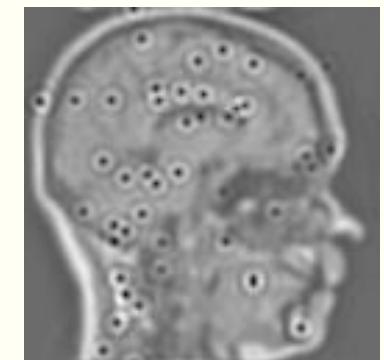
order $\square 4$



order $\square 8$



order $\square 16$





- Mit einem **Modell** des Blurrings (und des Rauschens) kann man versuchen, Bilder zu deblurren
- Deblurring ist **instabil** und kann nur analytisch durchgeführt werden, wenn keine Daten verloren gehen. Datenverlust tritt beispielsweise durch die endliche Repräsentation (8 Bit) von Intensitäten, Rauschen oder anderer Pixelfehler auf.
- Der **Wiener Filter** funktioniert im Allgemeinen gut, benötigt aber eine Abschätzung von Parametern. Er ist ein Ein-Schritt-Verfahren.
- Deblurring kann auch durchgeführt werden, indem der Laplace-Operator subtrahiert (‘PhotoShop’) wird und mehr Terme hinzugefügt werden, um das Ergebnis so lange zu verfeinern, bis ein optimales Bild vorliegt. Das Hinzufügen von zu vielen Termen wird das Rauschen allerdings wieder verstärken.



- Deblurring von Bildern
- Einschrittverfahren
 - Wiener Filter
 - Ansatz mit mehreren Komponenten
- **Intermezzo** 
- Mehrschrittverfahren / Iterative Methoden
 - Perona Malik
 - Eingeschränkte Evolution: Totale Variation
- Zusammenfassung

Deblurring mit einem Scale-Space-Ansatz



Die Sequenz

$$L[x, y, 0] - t(L_{xx} + L_{yy}) + \frac{1}{2}t^2(L_{xxxx} + 2L_{xxyy} + L_{yyyy}) \\ - \frac{1}{6}t^3(L_{xxxxxx} + 3L_{xxxxyy} + 3L_{xxyyyy} + L_{yyyyyy})$$

ist die Taylorreihe in $-t$ einer partiellen Differentialgleichung:



$$\frac{\partial L}{\partial t} = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$$

- Die Veränderung in einem Bild über eine gewisse Zeit ist durch die partiellen Ableitungen zweiter Ordnung definiert.



Gausssscher Scale-Space



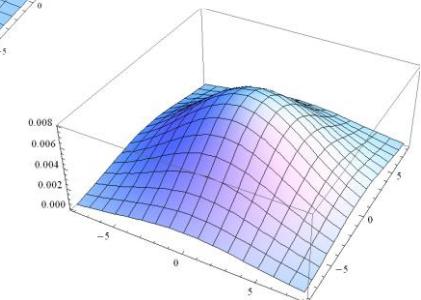
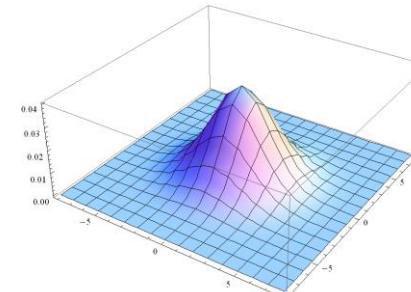
$$L_t(x, y; t) = \Delta L(x, y; t) = L_{xx} + L_{yy}$$

$$L_{t \rightarrow 0}(x, y; t) = L_0$$

Die allgemeine Lösung („Greensche Funktion“) dieser **Diffusionsgleichung** ist die Faltung des Originalbildes L_0 mit einer Gaußschen Funktion G :

$$G(x, y; t) = \frac{1}{4\pi t} e^{-\frac{(x^2+y^2)}{4t}}$$

$$L(x, y; t) = G(x, y; t) * L_0(x, y)$$



Beweis („in der Blackbox“)



Greensche Funktion = intrinsische Lösung

- $f_t - \Delta f$

Verwendung der Ableitungsregel im Fourierraum

- $F(f_x) = (i\omega)F(f) = (i\omega)\tilde{f}$

$$\frac{d\tilde{f}}{dt} = F(f_{xx} + f_{yy}) = (i\omega_x)^2 \tilde{f} + (i\omega_y)^2 \tilde{f} = -(\omega_x^2 + \omega_y^2) \tilde{f} \quad \text{NB } (x,y) \rightarrow (\omega_x, \omega_y)$$

$$\longrightarrow \frac{d\tilde{f}}{\tilde{f}} = -(\omega_x^2 + \omega_y^2) dt \longrightarrow \int \frac{d\tilde{f}}{\tilde{f}} = \int -(\omega_x^2 + \omega_y^2) dt$$

$$\ln \tilde{f} = -(\omega_x^2 + \omega_y^2)t + C_1 \longrightarrow \tilde{f} = e^{-(\omega_x^2 + \omega_y^2)t + C_1} = C_2 e^{-(\omega_x^2 + \omega_y^2)t}$$

$$f = \frac{C_3}{2t} e^{-\frac{(x^2 + y^2)}{4t}}$$

Das passiert, wenn t erhöht wird



TECHNISCHE
UNIVERSITÄT
DARMSTADT





- Deblurring von Bildern
- Einschrittverfahren
 - Wiener Filter
 - Ansatz mit mehreren Komponenten
- Intermezzo
- **Mehrschrittverfahren / Iterative Methoden**
 - Perona Malik
 - Eingeschränkte Evolution: Totale Variation
- Zusammenfassung

Energie – einfaches Modell:



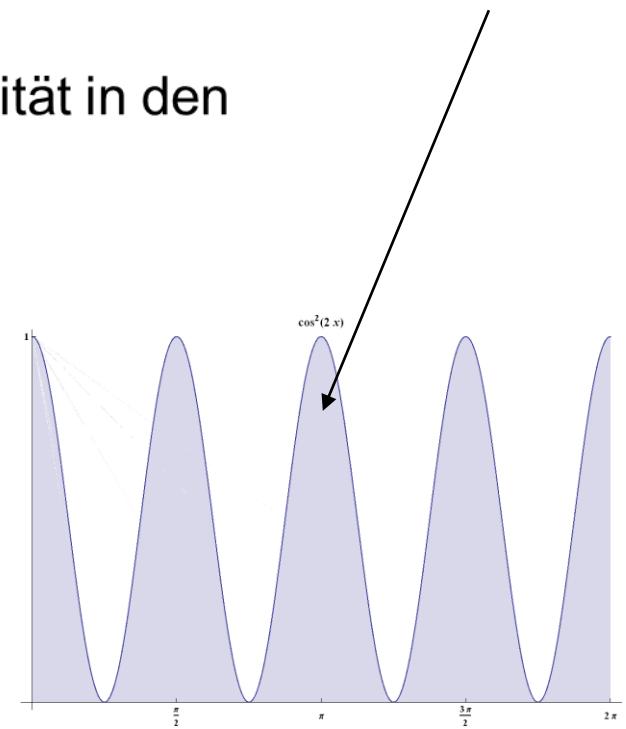
Sei E die Energie eines Bildes L :

$$E(L) = \frac{1}{2} \int_{x,y} L^2 dx dy$$

- Die Energie sagt im Prinzip aus, wieviel Intensität in den Pixeln vorhanden ist.

Minimale Energie:

Finde eine Funktion L
mit minimaler Fläche



Energie – einfaches Modell:

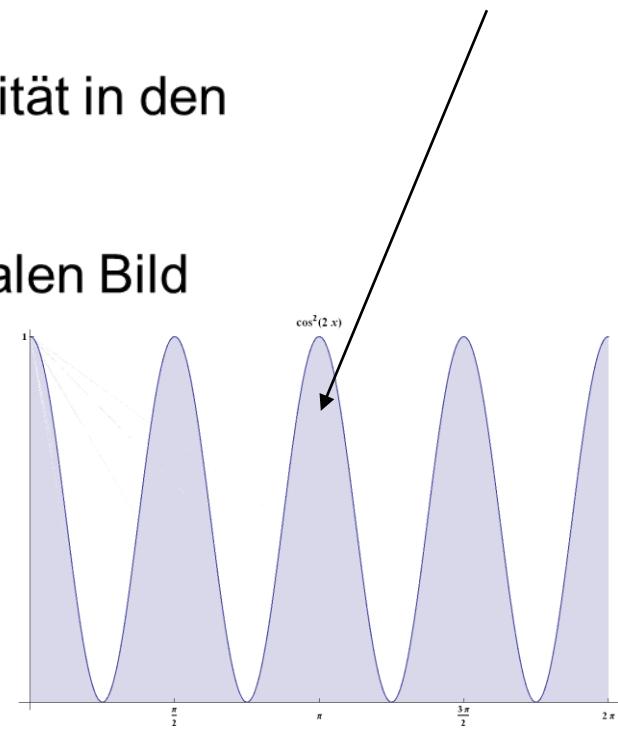
Sei E die Energie eines Bildes L :

$$E(L) = \frac{1}{2} \int_{x,y} L^2 dx dy$$

- Die Energie sagt im Prinzip aus, wieviel Intensität in den Pixeln vorhanden ist.
- Minimierung der Energie führt zu einem optimalen Bild
 - Bezuglich der definierten Energie
- Wie berechnet man das Minimum?
 - Variationsableitung
 - Iterativer Prozess

Minimale Energie:

Finde eine Funktion L
mit minimaler Fläche



Variationsableitung



Verallgemeinerung der normalen Ableitung

- Notation: $\delta E(L)$
- In dieser Veranstaltung als Blackbox betrachtet.

Für

- $E(L) = \frac{1}{2} \int_{x,y} L^2 dx dy$

erhält man

- $\delta E(L) = L$

Minimum: $\delta E(L) = 0$

- Trivial: $L = 0$ und $E(L) = 0$

Variationsableitung



Für

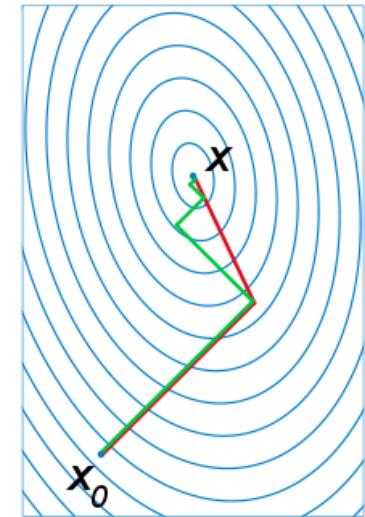
- $E(L) = \frac{1}{2} \int_{x,y} L_x^2 + L_y^2 \ dx dy$

erhält man

- $\delta E(L) = -(L_{xx} + L_{yy}) = -\Delta L$

Hier ist es weniger trivial, das Minimum zu finden

- Überführung in eine partielle Differentialgleichung
 - $L_t = -\delta E(L)$
- Gegen die Lösung iterieren („Heat Equation“)
 - $L_t = \Delta L$



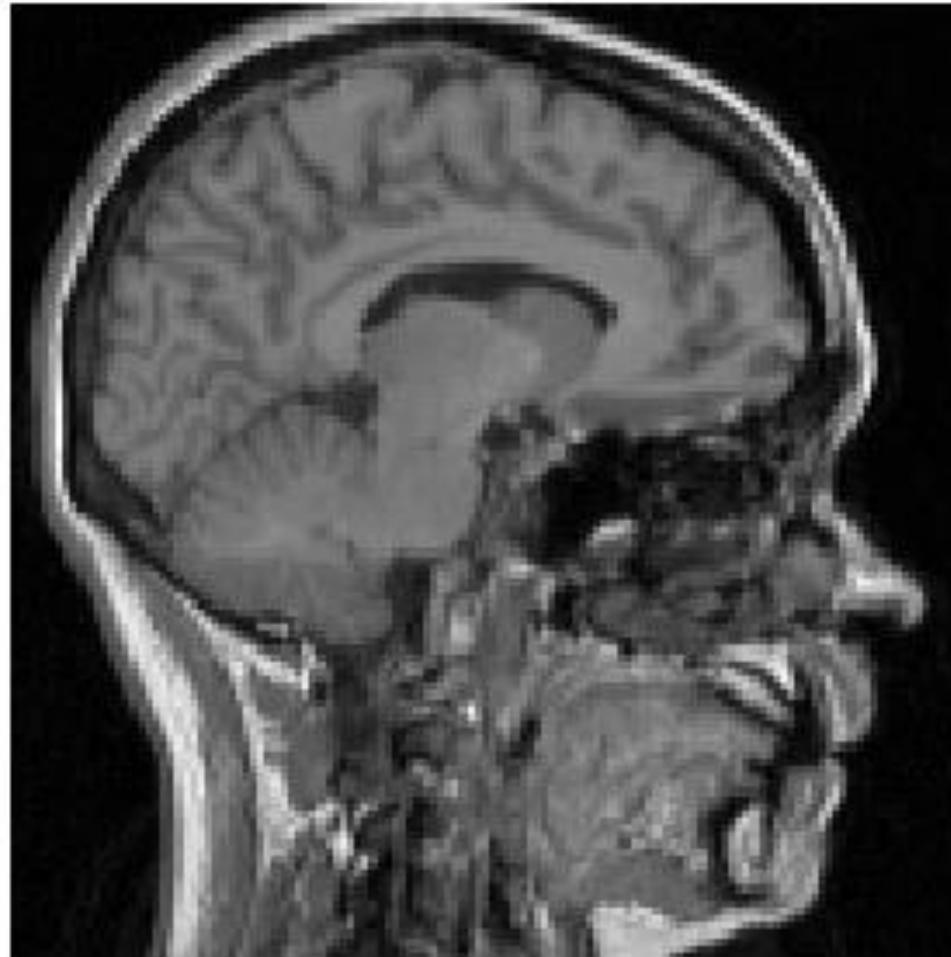
Lösung

Finden einer Funktion L
mit minimaler
Gesamtableitung

Ergebnis



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Beispiel



TECHNISCHE
UNIVERSITÄT
DARMSTADT



<https://www.youtube.com/watch?v=fjhVAEB1TS4>

Numerical Analysis: Heat equation $dt = 1/8$ - Iterations - 200



Alternativen

Da auf diese Weise nicht so gut *interessante* Bilder generiert werden können, wurden andere Energien vorgeschlagen, die nicht nur **Pixelwerte**, aber auch **Nachbarschaften** berücksichtigen:

- **Perona Malik** (& co)
 - Rauschen verwischen
 - Kanten verstärken
 - Smart Energy Term + **Stoppzeit**
- **Total Variation** (Rudin & co)
 - Rauschen verwischen
 - Kanten verstärken
 - Smart Energy Term + **Distance Penalty**
- Und sehr viele weitere...



- Deblurring von Bildern
- Einschrittverfahren
 - Wiener Filter
 - Ansatz mit mehreren Komponenten
- Intermezzo
- **Mehrschrittverfahren / Iterative Methoden**
 - **Perona Malik**
 - Eingeschränkte Evolution: Totale Variation
- Zusammenfassung

Nichtlineare Diffusionsgleichungen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Die Heat Equation ($L_t = L_{xx} + L_{yy} = \Delta L = \nabla \cdot \nabla L$) wird modifiziert zu:

$$\partial_t L = \nabla \cdot (c \nabla L)$$

c kann klug gewählt werden:



Die Heat Equation ($L_t = L_{xx} + L_{yy} = \Delta L = \nabla \cdot \nabla L$) wird modifiziert zu:

$$\partial_t L = \nabla \cdot (c \nabla L)$$

c kann klug gewählt werden:

- Es wird *Conductivity Coefficient* genannt
- Diffusion ist an lokale Bildstrukturen anpassbar:
 - $c = c(L, L_x, L_{xx}, \dots)$
 - $c=1$: Zurück zum Gaußschen Scale-Space
 - $\nabla L = (L_x, L_y)$
 - $\nabla \cdot (c \nabla L) = (\partial_x, \partial_y) \cdot (c(L_x, L_y)) = \partial_x(c(L_x)) + \partial_y(c(L_y))$

Die Perona-Malik-Gleichung



Perona und Malik (1990):

- $c(\cdot)$ ist eine Funktion der Gradientstärke
- Reduziert die Diffusion dort, wo Kanten sind (c nahe bei 0), aber verstärkt sie in flachen Bereichen ($c = 1$):

$$\partial_t L = \nabla \cdot (c(|\nabla L|^2) \nabla L)$$

Die Perona-Malik-Gleichung



Perona und Malik (1990):

- $c(\cdot)$ ist eine Funktion der Gradientstärke
- Reduziert die Diffusion dort, wo Kanten sind (c nahe bei 0), aber verstärkt sie in flachen Bereichen ($c = 1$):

$$\partial_t L = \nabla \cdot (c(|\nabla L|^2) \nabla L)$$

Zwei Möglichkeiten für c :

- $c_1 = e^{-\frac{|\nabla L|^2}{k^2}}$ $c_2 = 1 / \left(1 + \frac{|\nabla L|^2}{k^2}\right)$
- Mehr oder minder gleiches Verhalten
- k bestimmt den Einfluss der Kantenstärke



c_1 für verschiedene Belegungen von k

- Großes k: Nur größere Gradienten (stärkere Kanten) bleiben übrig.
- Kleines k: (fast) alle Gradienten (Kanten, rauschen) bleiben übrig.

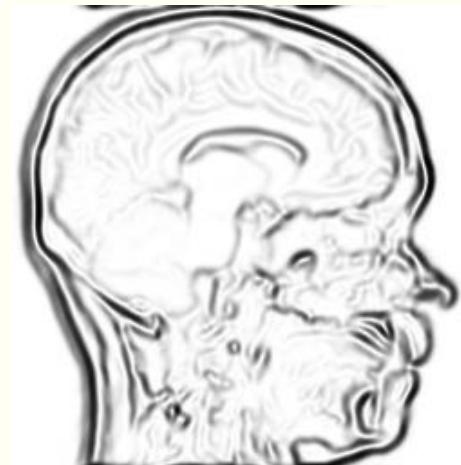
k □ 5



k □ 10



k □ 20



Formulierung als partielle Differentialgleichung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Für diejenigen, die es wirklich wissen wollen:

- c_1

$$L_t = \frac{e^{-\frac{L_x^2 + L_y^2}{k^2}} ((k^2 - 2 L_x^2) L_{xx} - 4 L_x L_{xy} L_y + (k^2 - 2 L_y^2) L_{yy})}{k^2}$$

- c_2

$$L_t = \frac{k^2 (-4 L_x L_{xy} L_y + L_{xx} (k^2 - L_x^2 + L_y^2) + (k^2 + L_x^2 - L_y^2) L_{yy})}{(k^2 + L_x^2 + L_y^2)^2}$$

Implementierung der Perona-Malik-Gleichung



Keine analytische Lösung für diese partielle Differentialgleichung:

Iterative Methoden:

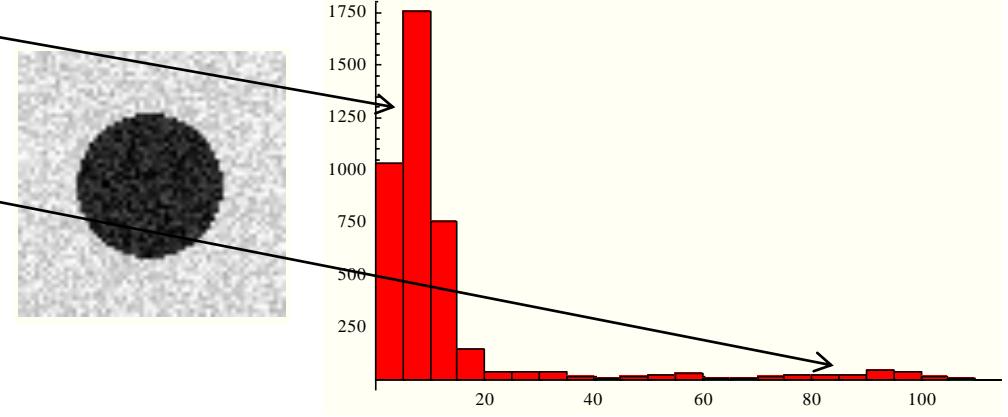
$$L^{t+1} = L^t + Dt \nabla \cdot (c(|\nabla L|^2) \nabla L)$$

- L^0 : original Bild
- Dt : Zeit Schritt (klein!)
- $\nabla \cdot (c(|\nabla L|^2) \nabla L)$: berechne bei jedem Schritt neu für Bild L^t
- „irgendwann“ aufhören:
 - Nach n Schritten
 - Wann L^{t+1} gut aussieht

Implementierung der Perona-Malik-Gleichung



- Der Wert von k hängt von der Wahl ab, welche Kanten verstärkt und welche entfernt werden sollen
- Das Histogramm der Gradientwerte: „Relative Kantenstärke“
 - Schwache Kanten:
Aufgrund von Rauschen
 - Starke Kanten:
Unterschiedliche Strukturen



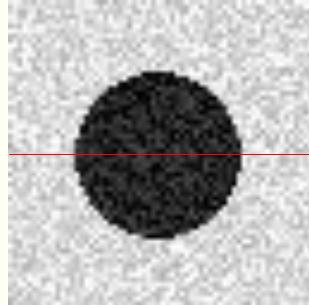
k muss angemessen gewählt werden



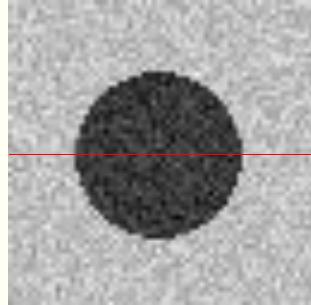
TECHNISCHE
UNIVERSITÄT
DARMSTADT

k ist ein Wendepunkt: Kantereduktion versus Verstärkung

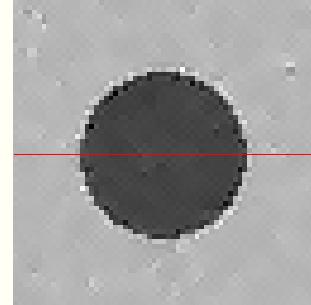
Original



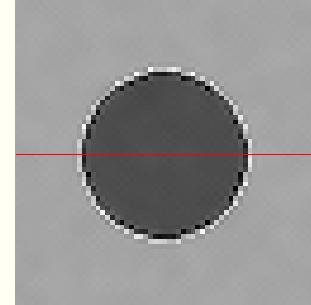
k=5



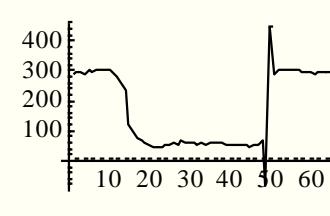
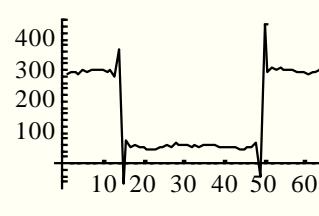
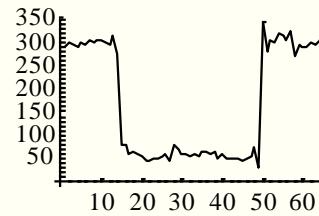
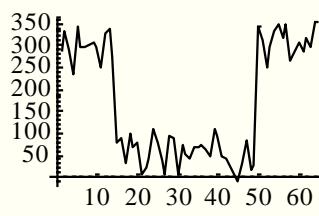
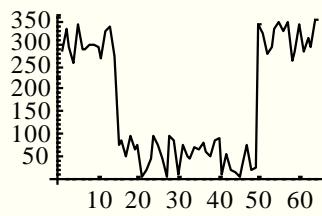
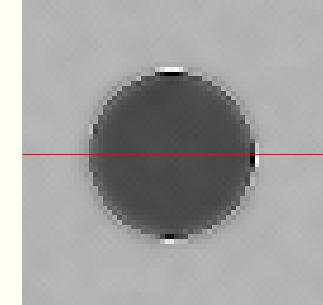
k=25



k=75



k=100



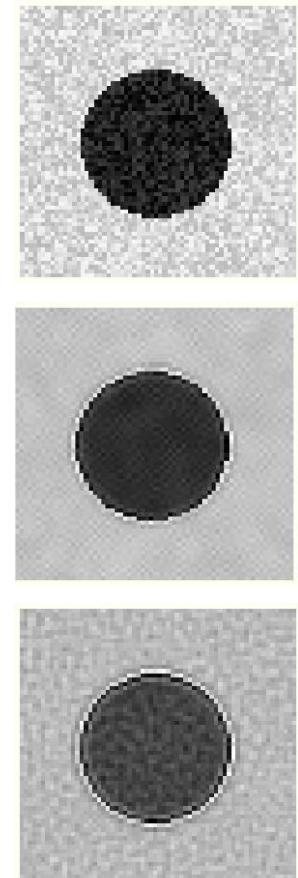
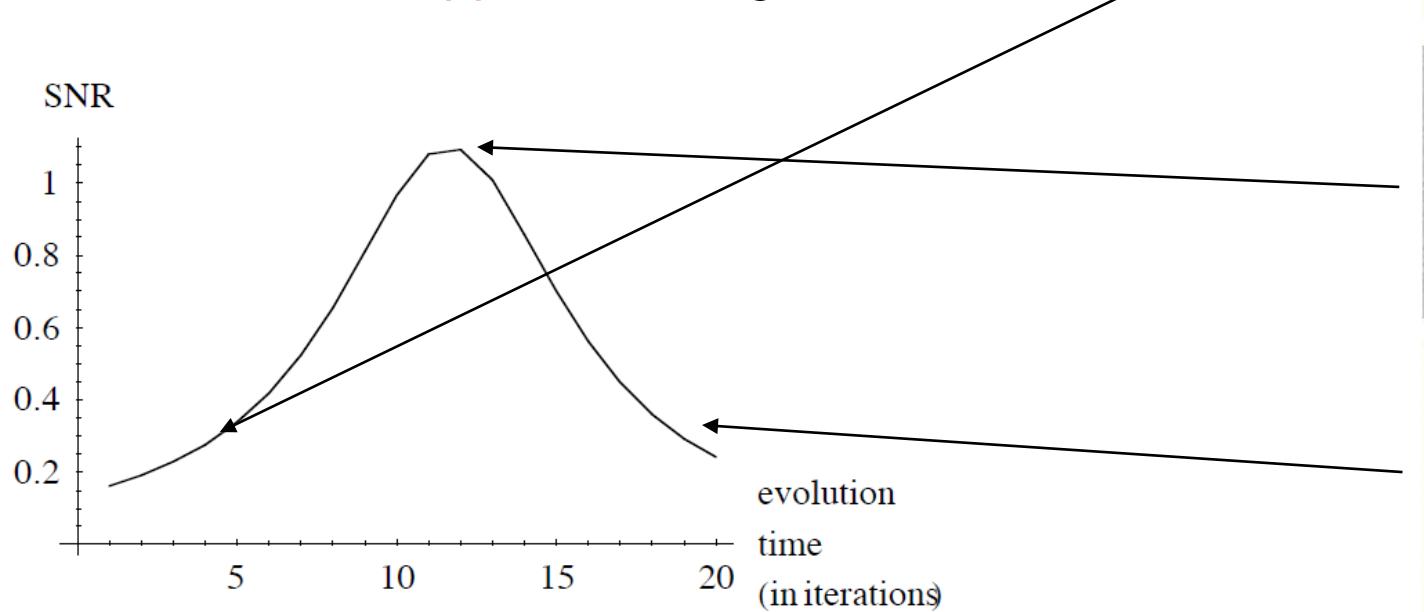
(Die Diagramme zeigen die Werte auf den roten Linien)

Nicht zu lange ausführen lassen!



Das Verhältnis Signal-zu-Rauschen steigt im Wesentlichen während der Evolution an, fällt danach aber wieder ab.

- Die Evolution stoppt **nicht** bei der optimalen Lösung
- Es wird eine **Stopzeit** benötigt



(De-)Blurring und Wendepunkt in 1D

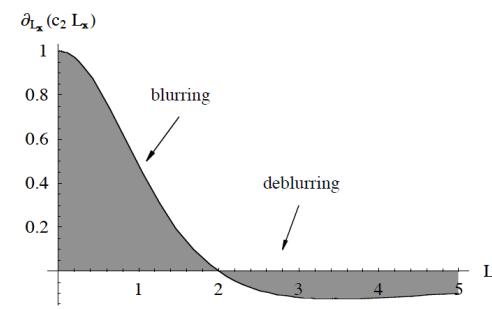
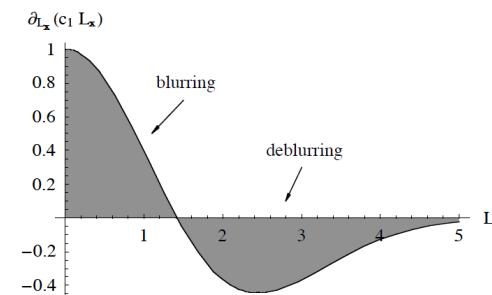
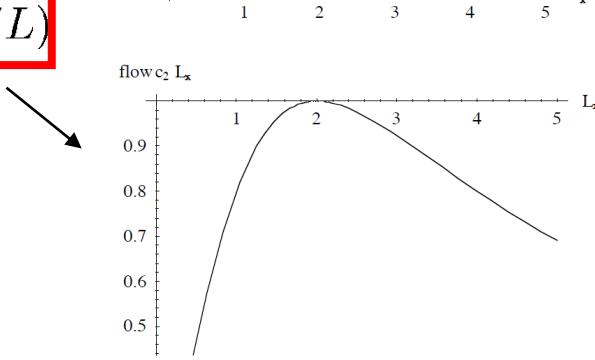
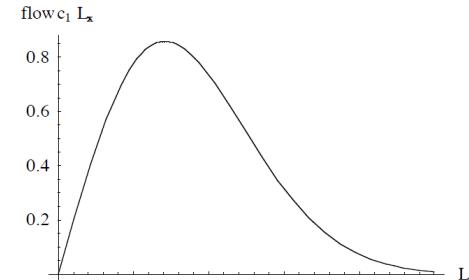


Das gleiche Blurring- und Deblurring-Verhalten wie das des Wiener Filters!

Blurring für kleine Gradienten, Deblurring für große Gradienten.

Die Graphen der Flüsse cL_x und $\frac{\partial(cL_x)}{\partial L_x}$ für beide c mit $k=2$:

$$\partial_t L = \nabla \cdot (c(|\nabla L|^2) \nabla L)$$



Stoppzeit



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Lösungen:

1. Stoppzeit
2. Regularisation
3. Numerisches Modell



Figure 3.9: Evolution of the image "Polonina" under the Perona-Malik filter for 0, 100, 500, 1000, 5000 and 10000 iterations with timestep 0.2 and $\lambda = 1$

Bekannt als Perona-Malik-Paradox:

Das Basismodell ist nicht korrekt gestellt, mit Diskretisierung kann es stabil gemacht werden.

Beispiel



<https://reference.wolfram.com/language/ref/PeronaMalikFilter.html?view=all>

PeronaMalikFilter

`PeronaMalikFilter[image]`

`PeronaMalikFilter[image]`

applies a Perona–Malik diffusion filter to *image*.

`PeronaMalikFilter[image, t]`

specifies the amount of diffusion time *t* to be applied.

`PeronaMalikFilter[image, t, k]`

uses a conductance parameter *k*.

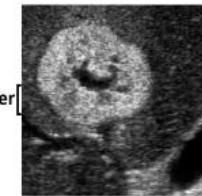
`PeronaMalikFilter[image, t, k, σ]`

applies a Gaussian regularization of width *σ* to the image gradient in the conductance function.

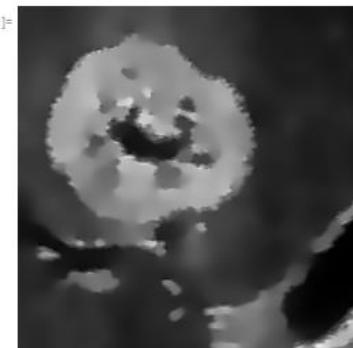
Specifying a minimum scale to neglect edges created by noise:

In[1]:=

`PeronaMalikFilter[`



Out[1]:=



▼ Details

- Perona–Malik filtering is an inhomogeneous diffusion method typically used for smoothing images while preserving edges.
- `PeronaMalikFilter` works on 2D grayscale or multichannel images, operating on each channel separately.
- `PeronaMalikFilter` applies the diffusion equation $\partial_t f = \nabla \cdot c_k(|\nabla_\sigma f|) \nabla f$ to every image channel *f*.
- The function $c_k(|\nabla_\sigma f|) = e^{-|\nabla_\sigma f|^2/k^2}$ of the σ -regularized gradient norm $|\nabla_\sigma f|$ defines the conductance of the diffusion current. At edges where the gradient norm is large in comparison to *k*, diffusion is suppressed, thereby preserving edges.
- In `PeronaMalikFilter[image, t]`, *t* parameterizes the evolution of the diffusion and thereby the spatial range of the filter.
- The conductance parameter *k* can take any positive value. The default value of *k* is `Automatic`, which assigns to *k* the 50% quantile of the gradient norm $|\nabla f|$ of *image*. If more than one channel is present, the gradient norm of the channel average is taken into account.
- The regularization parameter *σ* is the standard deviation of the Gaussian kernel K_σ , with which the image gradient $\nabla_\sigma f = \nabla(K_\sigma * f)$ is convolved. The σ -regularization makes the conductance term $c_k(|\nabla_\sigma f|)$ less susceptible to noise. If $\sigma < 1/6$, a finite difference scheme is used to determine the gradient.
- `PeronaMalikFilter[image]` is equivalent to `PeronaMalikFilter[image, 1, Automatic, 0]`.



Cosine-Driven Non-linear Denoising

Daniel Thuerck and Arjan Kuijper

ICIAR 2013, LNCS 7950, pp. 245–254, 2013

Image title	Variance	k_{PM}	t_S	SSIM _{PM}	k_{SM}	t_S	SSIM _{SM}	Difference
Barbara	0.001	19	3	0.9202	8	4	0.9156	-0.5%
	0.005	32	5	0.7968	7	12	0.7969	0.01%
	0.01	43	6	0.7144	7	18	0.7258	1.6%
	0.05	105	14	0.5784	12	32	0.5992	3.6%
	0.1	102	22	0.5449	12	44	0.5603	2.8%
Lena	0.001	21	3	0.8999	8	5	0.9037	0.4%
	0.005	38	6	0.8295	8	14	0.8423	1.5%
	0.01	50	8	0.7911	8	21	0.8116	2.6%
	0.05	102	18	0.7053	12	36	0.7261	2.9%
	0.1	120	26	0.6759	12	49	0.6876	1.7%
Cameraman	0.001	18	4	0.9306	7	6	0.9300	-0.01%
	0.005	35	7	0.8535	10	11	0.8643	1.3%
	0.01	46	10	0.8147	10	17	0.8317	2%
	0.05	78	25	0.6911	8	48	0.7523	8.9%
	0.1	90	55	0.6547	8	67	0.7122	8.8%
Boat	0.001	24	2	0.8765	7	4	0.8803	0.4%
	0.005	41	4	0.7833	11	8	0.7911	0.9%
	0.01	54	5	0.7269	11	13	0.7423	2.1%
	0.05	96	10	0.5893	12	29	0.6198	5.2%
	0.1	107	16	0.5404	8	58	0.5698	5.4%

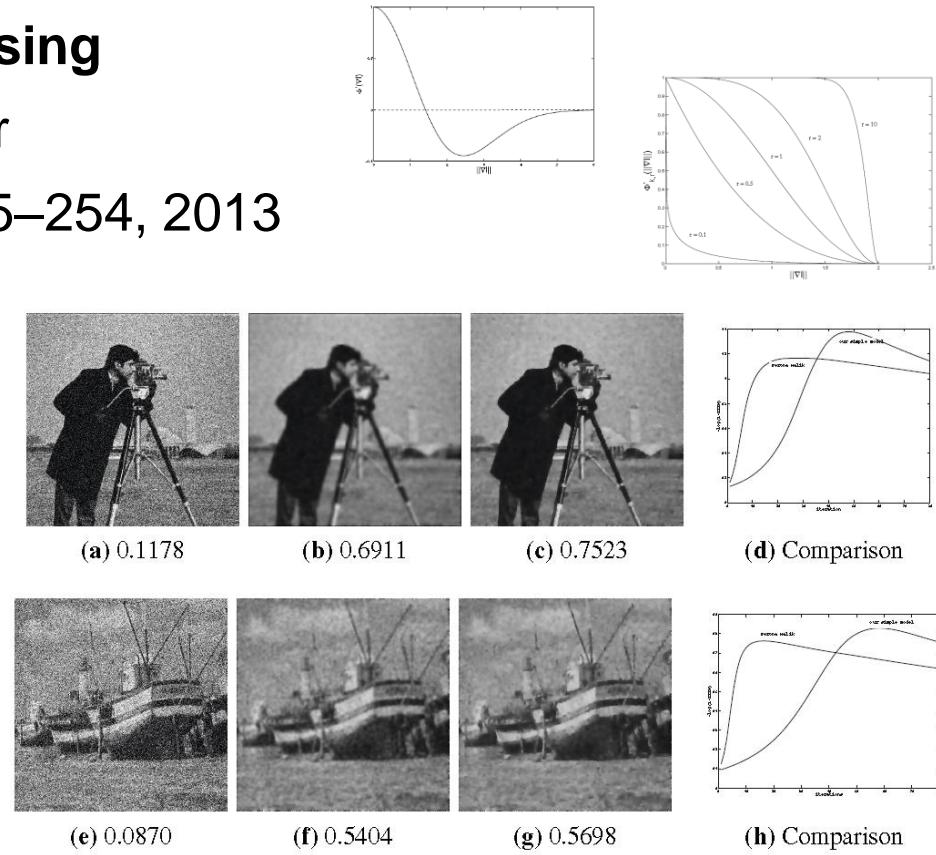


Fig. 6. Numerical results for Perona-Malik model (b,f) and our simple model (c,g) for *cameraman* with noise variance 0.05 (a) and *boat* with noise variance 0.1 (e), followed by the resulting SSIM values plotted as $-\log(1 - \text{SSIM})$ (d,h)

Rekapitulation



- Diffusion kann **lokal anpassbar** an die Bildstruktur gemacht werden.
- Hierfür ist eine geometrische Beweisführung vonnöten, um den Einfluss auf den Diffusionskoeffizienten c festzulegen.
- Die nichtlinearen partiellen Differentialgleichungen beinhalten lokale Bildableitungen und **können nicht** analytisch gelöst werden.
- Iterative Prozesse werden benötigt, da sie die graduellen Änderungen des Bildes in jedem Iterationsschritt zeigen.
- Die einfachste Gleichung ist die von **Perona und Malik** vorgeschlagene. Die Diffusion c ist hier eine Funktion der lokalen Kantenstärken.
- Hohe Gradientstärken verhindern Blurring lokal, der Effekt ist kantenerhaltendes Glätten.
- Das führt zu Deblurring (und verstärkten Kanten) bei Kanten die größer als der Wendepunkt k sind, und zu Blurring bei kleineren Kanten (Rauschen).
- Es wird eine Stoppzeit benötigt, da zu viele Iterationen zu schlechten Ergebnissen führen.



- Deblurring von Bildern
- Einschrittverfahren
 - Wiener Filter
 - Ansatz mit mehreren Komponenten
- Intermezzo
- **Mehrschrittverfahren / Iterative Methoden**
 - Perona Malik
 - **Eingeschränkte Evolution: Totale Variation**
- Zusammenfassung

Lösen der Perona-Malik-Probleme

Es muss sichergestellt werden, dass die Iterationen konvergieren.

- In einem bestimmten Moment ist das Bild optimal und wird nicht weiter verändert.
- Eine benutzerdefinierte **Stoppzeit** wird dann **nicht** benötigt.

Man versucht eine klug gewählte Energie zu minimieren und eine **Distance Penalty** hinzuzufügen.

- Ein Bild mit der geringsten Energie soll gefunden werden, allerdings soll es auch nicht allzu sehr vom Originalbild abweichen.



Hinzufügen der Distance Penalty



Verwenden von Statistiken zum Rauschen.

Modell zum Rauschen: $g = a(f) + n$

Es gibt dann zusätzliche Bedingungen für eine Lösung L , die nahe an f sein soll, wenn das Rauschen als Gaußsches Zufallsrauschen modelliert ist:

- $\int_{x,y} g - a(L) dx dy = 0$
- $\int_{x,y} (g - a(L))^2 dx dy = \sigma^2$

Das Verwischen von L und Hinzufügen gleichartigen Rauschens sollte Eigenschaften gleichartig zu g ergeben.

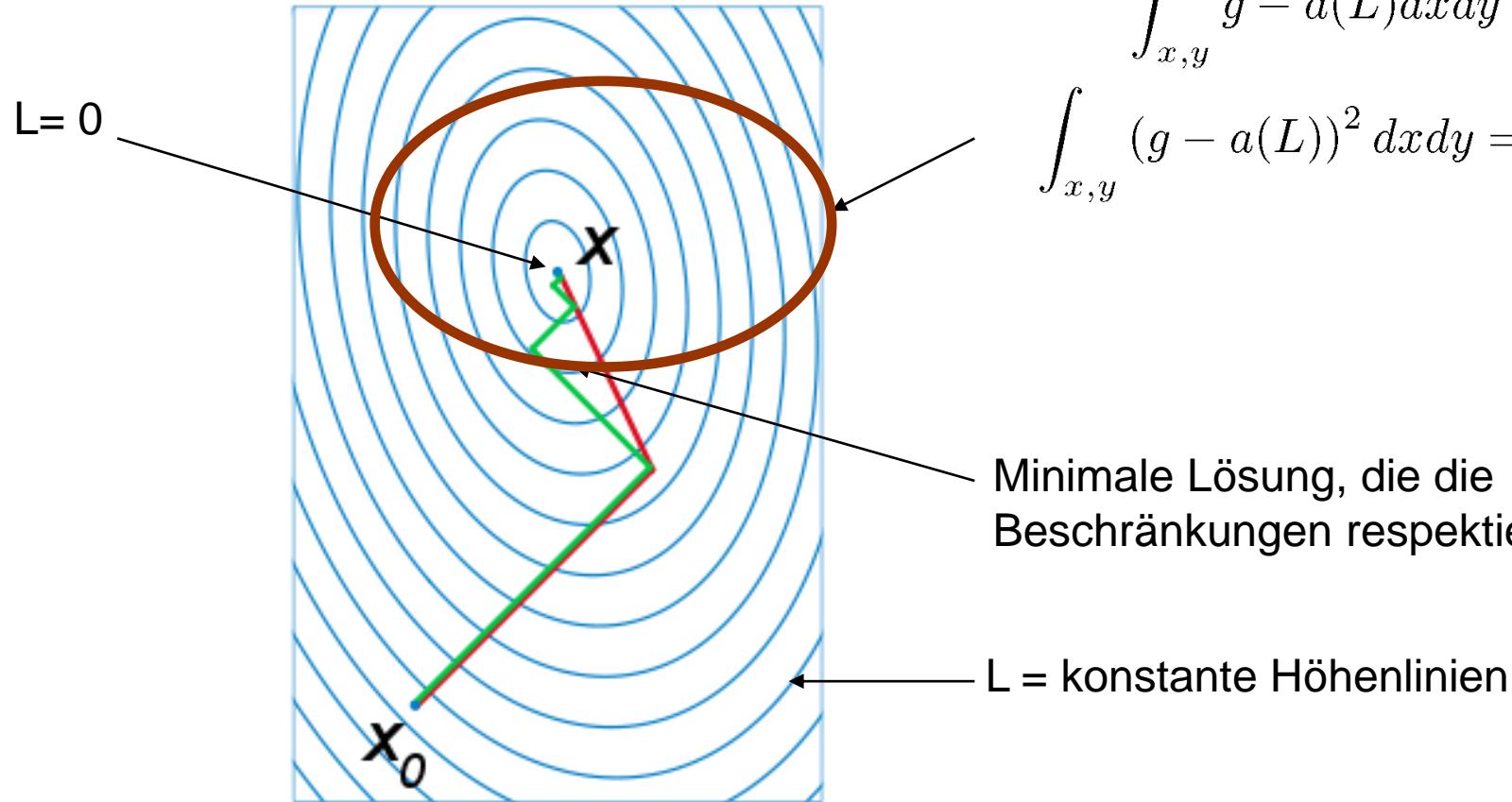


Diese Integrale (Beschränkungen) können zur Energie hinzugefügt werden (wobei das erste nichts beiträgt).

Partielle Differentialgleichungen: Gegen eine beschränkte Lösung konvergieren



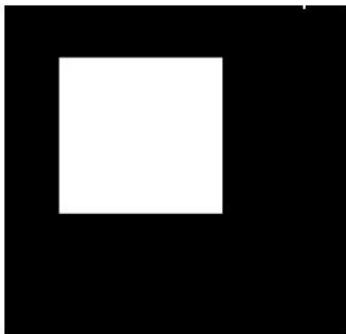
Grafische Interpretation:



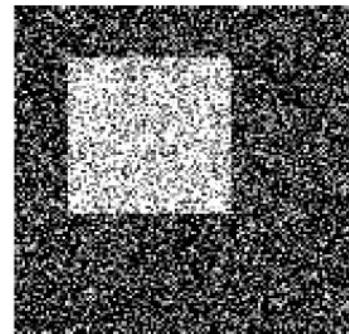
Noch einmal: Basisenergien

Gegeben sei ein Quadrat (a) mit Rauschen (b). Man bekommt mit diesen Beschränkungen

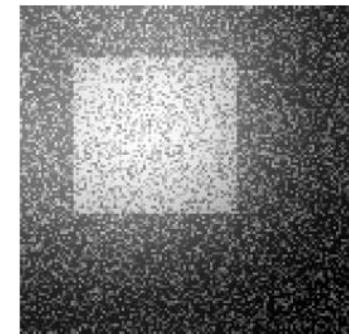
- Den Wiener Filter (c) für einen Energieterm
 - $E(L) = \frac{1}{2} \int_{x,y} L^2 dx dy$
- Den beschränkten Gaußschen Scale-Space (d) für den Energieterm
 - $E(L) = \frac{1}{2} \int_{x,y} L_x^2 + L_y^2 dx dy$



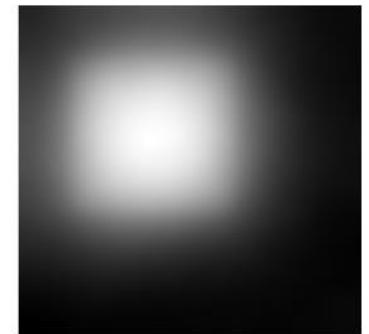
(a)



(b)



(c)



(d)

Totale Variation



Die Methode der totalen Variation:

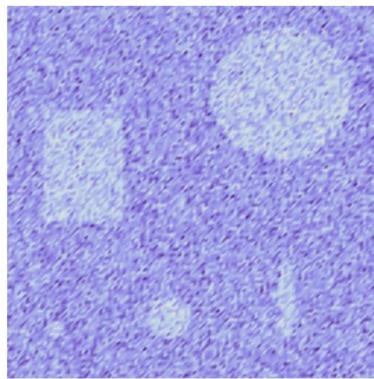
$$\int_{x,y} |\nabla L| + \lambda(g - a(L))^2 dx dy$$

- Kein Blurring, Stufenkanten werden bevorzugt
- Denoising
- Es wird keine Stoppzeit benötigt **-> TV konvergiert zu der optimalen Lösung**
- Findet ein stückweise konstantes Bild mit Sprüngen (Kanten), das die Rauschbeschränkungen einhalten.
- λ ist ein vom Rauschen abhängiger Parameter
- Eingeführt von Rudin et al., 1990
- Sehr kompliziert!
- Aber mathematisch sehr interessant!
- Öffnet ein völlig neues Feld der mathematischen Bildgebung

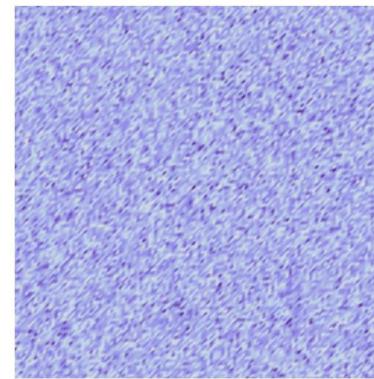
Beispiel



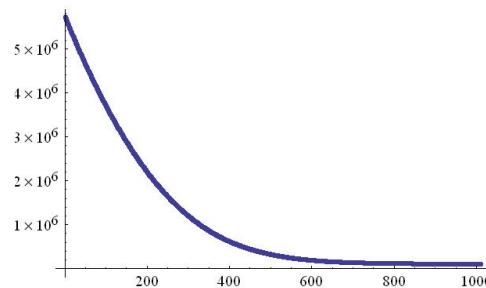
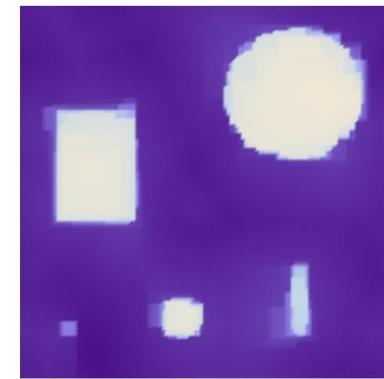
Bild mit Rauschen



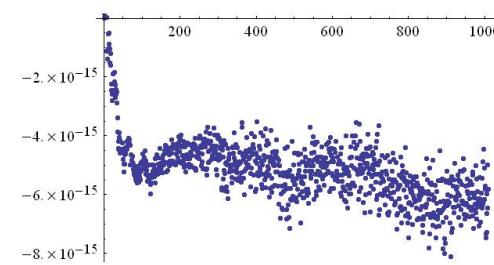
Entfernt



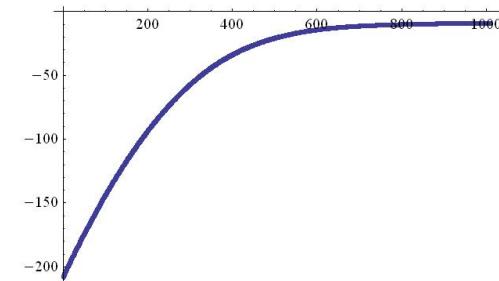
Wiederhergestellt



Totale Variation-Norm



Fehler im Durchschnitt



Fehler in Varianz

Beispiel



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Screenshot of a web browser showing the Wolfram Demonstrations Project page for "Total Variation Denoising".

The page title is "Total Variation Denoising".

Key features shown:

- An interactive interface with sliders for "type of noise" (Gaussian), "amount of noise", "parameter", and "iterations".
- A preview image showing a noisy spiral galaxy image being denoised.
- A large orange button: "Interact Now! Get free Wolfram CDF Player »".
- Share buttons for email, LinkedIn, Facebook, Twitter, and Google+.
- Links to "Download Demonstration as CDF" and "Download Author Code".
- A section titled "Related Demonstrations" listing other projects like "Image Deconvolution", "Image Inpainting", "Creating Posters from Photographic Images", and "Image Smoothing Using Stationary Wavelet Transform".
- A "SNAPSHOTS" section showing small versions of the demonstration interface.
- A "DETAILS" section with references:
 - [1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, 60, 1992 pp. 259–268.
 - [2] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, Philadelphia: SIAM, 2005.
 - [3] Wikipedia, "Total Variation Denoising," http://en.wikipedia.org/wiki/Total_variation_denoising.

TV @ Lena



TECHNISCHE
UNIVERSITÄT
DARMSTADT



TV @ Lena in Farbe



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Lena @ ICIP2015



Lena revisited



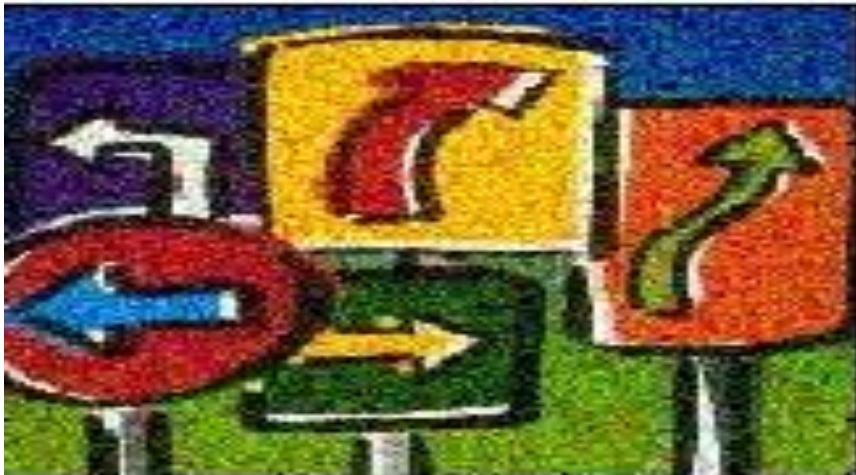
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Funktioniert gut mit cartoonähnlichen Bildern



TECHNISCHE
UNIVERSITÄT
DARMSTADT



TV funktioniert schlechter mit Barbara

Input Image



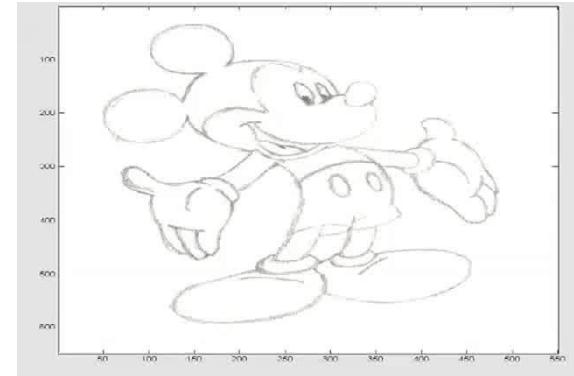
Denoised Image



Erweiterungen



- Schwierigere Funktionen/Energien
 - Statistik, Niveaumengen für Segmentierung, ...
- Andere Rauschstatistiken
 - Multiplikatives Rauschen
- Andere Steuermechanismen für den Conduktive Coefficient (c)
 - Kantenverstärkende Diffusion
 - Kohärenzverstärkende Diffusion
 - ...



Inpainting



TECHNISCHE
UNIVERSITÄT
DARMSTADT

The noisy image to be inpainted: SNR=20:1



The inpainted and denoised image



Russland, Iran,



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Nord Korea (In Gedenken an Kim Jong II),



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Russland / Krim / Ukraine [Photoshop (2)],



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Donetsk laut Russische Media

Nord Korea (Pt II),



TECHNISCHE
UNIVERSITÄT
DARMSTADT

■ 2013년 10월 7일 방영(장성택 삭제 전 영상) / 6분 05초



■ 2013년 10월 7일 방영(장성택 삭제 전 영상) / 9분 50초

■ 2013년 12월 7일 방영(장성택 삭제 후 영상) / 6분 05초



■ 2013년 12월 7일 방영(장성택 삭제 후 영상) / 6분 05초

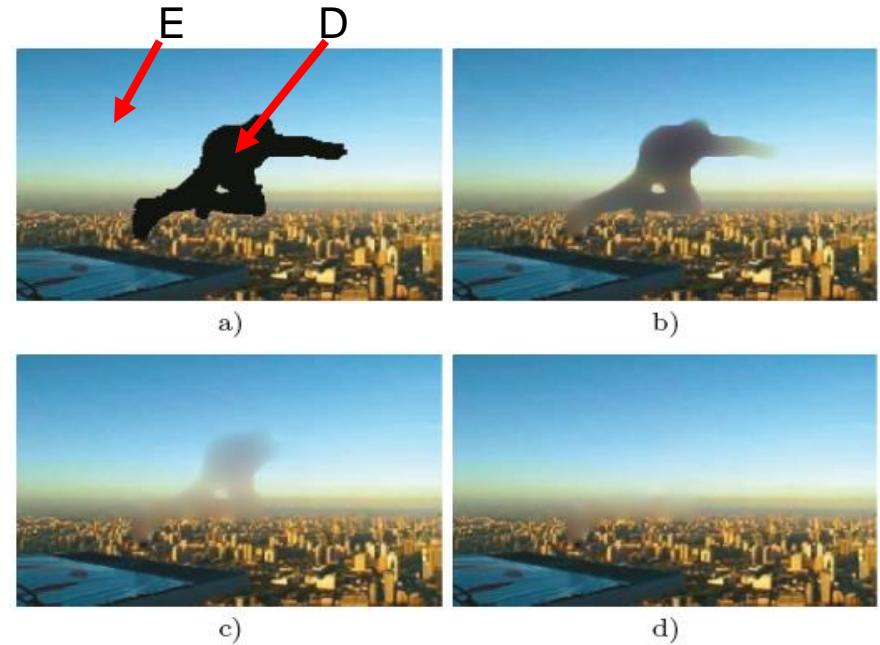
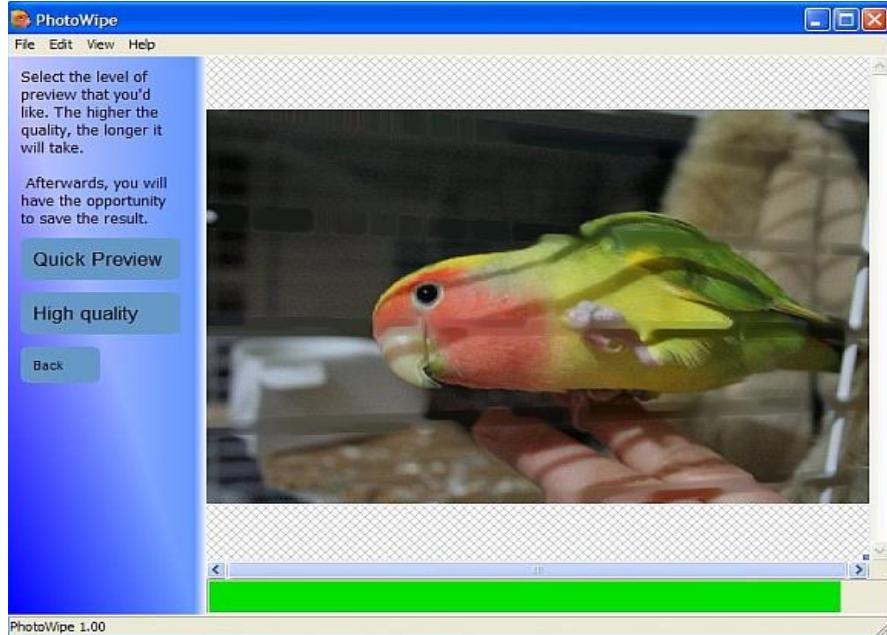
Katalonien Spanien,



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Keine „Clone Tools“ oder Photoshopping

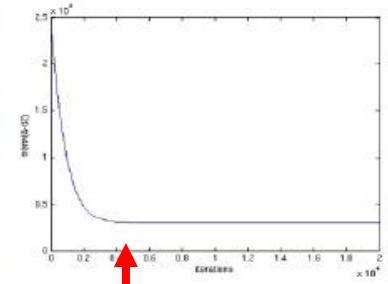


Wissenschaftliche Alternative
(stückweise konstantes Inpainting):

$$J[u] = \int_{E \cup D} |\nabla u| dx dy + \frac{\lambda}{2} \int_E |u - u^0|^2 dx dy,$$



e)
Konvergierte Energie





- Deblurring von Bildern
- Einschrittverfahren
 - Wiener Filter
 - Ansatz mit mehreren Komponenten
- Intermezzo
- Mehrschrittverfahren / Iterative Methoden
 - Perona Malik
 - Eingeschränkte Evolution: Totale Variation
- **Zusammenfassung**

Zusammenfassung



- Mit einem **Modell** des Blurrings (und des Rauschens) kann man versuchen, Bilder zu deblurren. Deblurring ist **instabil**.
- Der **Wiener Filter** funktioniert im Allgemeinen gut, benötigt aber eine Abschätzung von Parametern.
- Deblurring kann auch als **iterativer Prozess** durchgeführt werden, bei dem mehr Terme hinzugefügt werden, um das Ergebnis so lange zu verfeinern, bis ein optimales Bild vorliegt.
- **Energieminimierungsmethoden** benötigen eine korrekt definierte Energie für das Bild.
- Lösungen werden mithilfe von **partiellen Differentialgleichungen** gefunden. Diffusion kann **lokal anpassbar** an die Bildstruktur gemacht werden.
- Der Ansatz von **Perona und Malik** verwendet lokale Kantenstärken: **Deblurring** für starke Kanten, **Blurring** für Rauschen.
- Rauschen **beschränkt** klug gestaltete partielle Differentialgleichungen, die zu einem optimalen Bild führen.

Selbst spielen?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wiener Filter:

- <http://reference.wolfram.com/language/ref/WienerFilter.html>

Perona Malik:

- <http://reference.wolfram.com/language/ref/PeronaMalikFilter.html>

Total Variation:

- <http://reference.wolfram.com/language/ref/TotalVariationFilter.html>



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vielen Dank für die Aufmerksamkeit