

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

**Strategien zur adaptiven
Gewichtung von Daten- und
Glattheitstermen in
Variationsansätzen zur
Bildentrauschung**

Hannes Bonasch

Studiengang: Informatik

Prüfer/in: Prof. Dr.-Ing. Andrés Bruhn

Betreuer/in: Prof. Dr.-Ing. Andrés Bruhn

Beginn am: 15. Mai 2018

Beendet am: 15. November 2018

Kurzfassung

Eine Methode um Bildrauschen zu entfernen, ist der sogenannte Variationsansatz. Hierbei wird ein Energiefunktional aus einem Daten- und einem Glattheitsterm aufgestellt, dessen Minimierer dem entrauschten Bild entspricht. Der Datenterm stellt sicher, dass eine Ähnlichkeit zwischen Ein- und Ausgabebild besteht, während der Glattheitsterm für eine Weichzeichnung des Bildes sorgt. Die Arbeit von Hong et al. [HKDB17] stellte einen adaptiven Variationsansatz für die Bildsegmentierung und Bewegungsvorhersage vor. Dieser Ansatz verwendet eine adaptive Gewichtung zwischen Daten- und Glattheitsterm und führt zu einem robusteren Ergebnis bei variierenden Rauschpegeln. Das Ziel dieser Arbeit war es, zu untersuchen, ob der adaptive Ansatz auch für die Bildentrauschung nützlich ist. Als erstes wurde ein quadratischer Ansatz implementiert. Danach wurde der Glattheitsterm zu einer subquadratischen Bestrafungsfunktion erweitert, um dessen Kantenerhaltung zu verbessern. Sowohl der quadratische als auch der subquadratische Ansatz wurden dann mit der adaptiven Gewichtungsfunktion erweitert und die verschiedenen Verfahren miteinander verglichen. Während sich der quadratische adaptive Ansatz als ungeeignet für die Bildentrauschung hervorstellte, zeigte der subquadratische adaptive Ansatz einige Vorteile gegenüber den einfachen Variationsansätzen ohne adaptive Gewichtung.

Inhaltsverzeichnis

1 Einleitung	7
1.1 Ziel der Arbeit	8
1.2 Verwandte Arbeiten	8
1.3 Aufbau der Arbeit	9
2 Bildrauschen und Bildähnlichkeit	11
3 Quadratischer Ansatz	13
3.1 Diskreter eindimensionaler Fall	13
3.2 Diskreter zweidimensionaler Fall	14
3.3 Jacobi-Verfahren	15
3.4 Gauß-Seidel-Verfahren	18
3.5 SOR-Verfahren	19
4 Subquadratischer Ansatz	21
4.1 Diskreter eindimensionaler Fall	22
4.2 Diskreter zweidimensionaler Fall	23
4.3 Lagged-Nonlinearity-Verfahren	23
4.4 Untersuchung der Parameter	25
5 Adaptiver Subquadratischer Ansatz	27
5.1 Diskreter eindimensionaler Fall	27
5.2 Diskreter zweidimensionaler Fall	28
5.3 Probleme mit feiner Textur	31
5.4 Verbesserung der adaptiven Gewichtung mit verschiedenen Filtern	31
5.5 Untersuchung der Parameter	33
6 Adaptiver Quadratischer Ansatz	35
7 Evaluation	37
7.1 Ungleichmäßig verteiltes Rauschen	38
7.2 Salz-und-Pfeffer-Rauschen	40
7.3 Multiplikatives Rauschen	41
7.4 Verschiedene Bilder und Rauschpegel	42
8 Zusammenfassung und Ausblick	49
Literaturverzeichnis	51

1 Einleitung

Das Entrauschen von Bildern ist ein klassisches Problem der Bildverarbeitung. Das Ziel hierbei ist es ein ungewolltes Störsignal, wie zum Beispiel das Wärmerauschen eines Bildsensors, aus dem Bild zu entfernen. Es gibt viele verschiedene Verfahren zum Bildentrauschen, von einfachen Mittelungsfiltersn (siehe Abbildung 1.1) bis hin zu Deep-Learning-Ansätzen, die versuchen dieses Problem zu lösen.



Abbildung 1.1: Originalbild, Bild mit additivem Gaußschen Rauschen mit $\sigma = 20$, und entrausches Bild mit MittelungsfILTER mit Nachbarschaftsgröße 7x7.

Eines dieser Entrauschungsverfahren ist der Variationsansatz, hierbei entspricht das entrauschte Bild dem Minimierer eines geeigneten globalen Energiefunktionalen. Dieses Energiefunktional besteht zum Beispiel aus einem Daten- und einem Glattheitsterm. Der Datenterm stellt sicher, dass noch eine Ähnlichkeit zwischen dem Eingabe- und dem Ausgabebild besteht, während durch den Glattheitsterm das Bild weichgezeichnet wird. Ein Parameter α bestimmt das Gleichgewicht zwischen Ähnlichkeit und Glattheit des Ausgabebildes.

$$E(u) = \text{Datenterm} + \alpha \cdot \text{Glattheitsterm} \quad (1.1)$$

Da die benötigte Glättung stark von dem Eingabebild und dessen Rauschpegel abhängt, ist ein fester Parameter α für ein robustes Verfahren nicht ausreichend. Zudem kann die benötigte Glättung in verschiedenen Teilen eines verrauschten Bildes unterschiedlich groß sein, so sollten zum Beispiel Kanten und Texturen weniger geglättet werden als flache Regionen. Ungleichmäßig verteiltes oder multiplikatives Rauschen kann ebenfalls nicht mit einer über das ganze Bild konstanten Glättung effektiv entrauscht werden.

1.1 Ziel der Arbeit

In der Arbeit von Hong et al. [HKDB17] wurde ein adaptiver Variationsansatz für die Bildsegmentierung und Bewegungsvorhersage, welcher mit Hilfe einer Gewichtungsfunktion zwischen Daten- und Glattheitsterm zu robusteren Ergebnissen bei variierenden Rauschpegel führte, vorgestellt. Das Ziel dieser Arbeit war es zu untersuchen, ob diese Idee einer adaptiven Gewichtungsfunktion auch für die Bildentrauschung geeignet ist. Dafür wurde zunächst ein quadratischer Variationsansatz, bei dem sowohl im Daten- als auch im Glattheitsterm Abweichungen quadratisch bestraft werden, implementiert. Ein Problem, das jedoch durch diese starke Glattheitsannahme entsteht, ist das Weichzeichnen von semantisch wichtigen Kanten des Originalbildes. Um dem entgegen zu wirken, wurde der quadratische Glattheitsterm dann durch eine subquadratische Bestrafungsfunktion ersetzt. Durch die vergleichsweise geringe Bestrafungen bei starken lokalen Abweichungen verstärkt dieser subquadratische Ansatz die Kantenerhaltung. Danach wurden sowohl der quadratische als auch der subquadratische Ansatz mit der adaptiven Gewichtungsfunktion erweitert und evaluiert.

1.2 Verwandte Arbeiten

Die Arbeit von Buades et al. [BCM05b] gibt einen guten Überblick über mehrere Verfahren zur Bildentrauschung. Ein erster Variationsansatz basierend auf der Totalvariation wurde von Rudin et al. [ROF92] vorgestellt. Ein Überblick über weitere Entwicklungen dieses Variationsansatzes wird in der Arbeit von Chan [Cha04] gezeigt. Die Bachelorarbeit von Albrecht [Alb15] über Variationsansätze zur Struktur-Textur-Zerlegung war ebenfalls sehr hilfreich. Es existieren mehrere Arbeiten, welche eine verbesserte Texturerhaltung mit Hilfe einer adaptiven Gewichtung des Daten- oder Glattheitsterms erreichen. In der Arbeit von Gilboa et al. [GSZ06] wird diese Gewichtung anhand einer Textur-Struktur-Zerlegung bestimmt. Die Arbeit von Chen et al. [CMSH+10] bestimmt diese Gewichtung anhand eines Indikators, der zwischen Kanten, Rauschen und flachen Regionen unterscheiden kann. Eine adaptive Gewichtung zur Verbesserung des Variationsansatzes bei ungleichmäßig verteiltem oder multiplikativen Rauschen wird in der Arbeit von Guo et al. [GH09] und Dong et al. [DHR11] eingesetzt. In der Arbeit von Guo et al. [GH09] wird vor dem Einsatz des Variationsansatzes eine Gewichtung des Glattheitsterms für jeden Pixel anhand von Kantenerkennung und lokaler Varianz des Eingabebildes bestimmt. Die Gewichtung in der Arbeit von Dong et al. [DHR11] wird vor jeder Iteration anhand der lokalen Varianz des letzten Ausgabebildes bestimmt. Weitere interessante Bildentrauschungsverfahren, die nicht auf dem Variationsansatz basieren, sind das Block-Matching BM3D [DFKE07], das SVD basierte Verfahren [EA06], NL-Means [BCM05a] und Wavelet-Threshholding [CYV00].

1.3 Aufbau der Arbeit

Zu Beginn der Arbeit werden Grundlagen zu Bildrauschen und Bildähnlichkeit erklärt. In dem nächsten Kapitel wird dann der quadratische Variationsansatz vorgestellt und mit Hilfe verschiedener iterativer Löser implementiert. Das Kapitel 4 beschäftigt sich mit der Erweiterung des quadratischen Ansatzes mit Hilfe einer subquadratischen Bestrafungsfunktion. Danach wird in Kapitel 5 der subquadratische Ansatz mit der adaptiven Gewichtungsfunktion versehen und dessen Vor- und Nachteile untersucht. Die Probleme des adaptiven quadratischen Ansatz werden in Kapitel 6 angesprochen. In der Evaluation werden dann alle diese vier Verfahren bei verschiedenen Rauscharten miteinander verglichen. Zuletzt folgt dann eine Zusammenfassung und ein Ausblick auf mögliche Verbesserungen der adaptiven Gewichtungsfunktion.

2 Bildrauschen und Bildähnlichkeit

Bildrauschen

Rauschen ist in der Physik allgegenwärtig und bezeichnet grundsätzlich eine Überlagerung eines Signals mit einem ungewollten Störsignal. Da es sich bei Rauschen meist um stochastische Prozesse handelt, lassen sich die verschiedene Arten des Rauschens mit unterschiedlichen Wahrscheinlichkeitsverteilungen modellieren.

Für die digitale Photographie relevant ist vor allem das additive Gaußsche Rauschen, welches zum Beispiel das Wärmerauschen in elektrischen Schaltkreisen beschreibt. Die Wahrscheinlichkeitsverteilung des Gaußschen Rauschen mit Erwartungswert μ und Standardabweichung σ entspricht der Normalverteilung:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2.1)$$

Ein solches normalverteiltes Störsignal bildet sich als eine Art Körnung ab. Es gibt viele verschiedene Ansätze um Bilder zu entrauschen, eine einfache Methode ist das Weichzeichnen des gesamten Bildes mit einem Mittelungs-Filter. Hierzu wird jeder Pixelwert des Bildes durch den Mittelwert seiner Nachbarschaft ersetzt. Dies führt dazu, dass sich das Störsignal in den Nachbarschaften gegenseitig ausgleicht, lässt jedoch die Qualität des Bildes stark leiden (siehe Abbildung 1.1). Die Abbildung 2.1 zeigt verschiedene Arten des Rauschens. Für diese Arbeit ist neben dem gleichmäßig verteilten additiven Gaußschen Rauschen besonders das ungleichmäßig verteilte Rauschen relevant, da man an diesem besonders gut den Vorteil der adaptiven Gewichtungsfunktion erkennen kann.



Abbildung 2.1: Verschiedene Arten des Rauschens.

Bildähnlichkeit

Um neue, bessere Methoden zum Entrauschen zu finden, sollte man sich zunächst überlegen, wie man die Qualität dieser Methoden vergleichen kann. Da für diese Arbeit die verrauschten Bilder aus unverrauschten Originalbildern generiert werden, können wir nach dem Entrauschen das Originalbild g mit dem entrauschten Bild f vergleichen. Die Ähnlichkeit zweier Bilder ist jedoch sehr subjektiv, weshalb es viele verschiedene Fehlermaße gibt [WBSS04]. Für diese Arbeit wird das Spitzen-Signal-Rausch-Verhältnis (Peak Signal to Noise Ratio):

$$PSNR(f, g) = 10 \log_{10} \left(\frac{255^2}{MSE(f, g)} \right), \quad (2.2)$$

welches sich aus dem mittleren quadratischen Fehler (Mean Square Error):

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{i,j} - g_{i,j})^2 \quad (2.3)$$

ergibt, verwendet.

Mit Hilfe dieser Fehlermaße kann man die verschiedenen Ansätze dieser Arbeit miteinander vergleichen und deren Parameter optimieren. Sowohl ein zu starkes Weichzeichnen des Bildes als auch eine nicht ausreichende Entfernung des Rauschens beeinflusst die Bildähnlichkeit. In Abbildung 2.2 kann man sehen, wie der mittlere quadratische Fehler sinkt je stärker das Bild vom Originalbild abweicht, während der PSNR-Wert mit zunehmender Ähnlichkeit steigt.

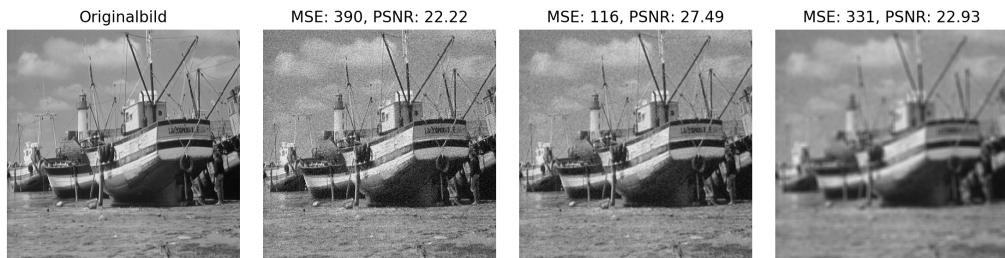


Abbildung 2.2: Verschiedene entrauschte Bilder und ihre Fehlermaße.

3 Quadratischer Ansatz

Der quadratische Variationsansatz ist eine globale Methode, es wird hierbei ein geeignetes Energiefunktional E aufgestellt und dann minimiert. Der Minimierer des Energiefunktionalen entspricht dann der Ausgabe, dem entrauschten Bild. Dieses Energiefunktional E besteht in unserem Fall aus einem Ähnlichkeitsterm, welcher die Abweichung des Ausgabebildes u vom Eingangsbild f quadratisch bestraft, und einem Glattheitsterm, welcher lokale Unterschiede der Pixelwerte quadratisch bestraft. Mit dem Parameter α kann man das gewollte Gleichgewicht zwischen Ähnlichkeit zum Eingangsbild und Glattheit des entrauschten Signals einstellen. Ein höherer Wert für α liefert ein glatteres Signal. Die folgende Formel zeigt den Fall für ein kontinuierliches zweidimensionales Eingabesignal Ω :

$$E(u) = \int_{\Omega} \underbrace{\frac{1}{2} (u - f)^2}_{\text{Ähnlichkeit}} + \underbrace{\frac{\alpha}{2} |\nabla u|^2}_{\text{Glattheit}} \, dx \, dy \quad (3.1)$$

mit

$$\nabla u = (u_x, u_y)^\top. \quad (3.2)$$

Zur Minimierung dieses Energiefunktionalen wird es zunächst diskretisiert, dann abgeleitet und schließlich das dadurch entstehende lineare Gleichungssystem gelöst.

3.1 Diskreter eindimensionaler Fall

Zur Veranschaulichung wird zunächst der diskrete eindimensionale Fall gezeigt. Anstelle von zweidimensionalen Bildern werden das verrauschte Eingangssignal $f = (f_1, f_2, \dots, f_N)$ und das Ausgangssignal $u = (u_1, u_2, \dots, u_N)$ verwendet. Der Ähnlichkeitsterm besteht aus der Summe der quadratischen Abweichung zwischen allen Punkten u_i und dem jeweils zugehörigen Punkt f_i . Der Glattheitsterm ist die Summe der Nachbarschaftsdifferenzen aller Punkte. Diese Nachbarschaftsdifferenzen bestehen im eindimensionalen Fall aus der quadratischen Differenz zwischen einem Punkt u_i und dessen linken Nachbarn u_{i-1} beziehungsweise dessen rechten Nachbarn u_{i+1} .

$$E(u) = \underbrace{\frac{1}{2} \sum_{i=1}^N (u_i - f_i)^2}_{\text{Ähnlichkeit}} + \underbrace{\frac{\alpha}{2} \sum_{i=1}^N \left(\frac{1}{2} (u_i - u_{i-1})^2 + \frac{1}{2} (u_i - u_{i+1})^2 \right)}_{\text{Glattheit}} \quad (3.3)$$

mit Randbedingungen $u_0 = u_1$ und $u_{N+1} = u_N$.

3 Quadratischer Ansatz

Das Energiefunktional $E(u)$ ist strikt konvex und besitzt somit maximal eine Extremstelle. Um diese zu finden wird nach u abgeleitet und das daraus resultierende lineare Gleichungssystem gelöst. Insgesamt erhält man je einen Fall für das linke und das rechte Ende des Signals, sowie einen allgemeinen Fall:

$$\begin{aligned} 0 &= \frac{\partial E(u)}{\partial u_1} = u_1 - f_1 + \alpha(u_1 - u_2) & (i = 1) \\ 0 &= \frac{\partial E(u)}{\partial u_i} = u_i - f_i + \alpha(u_{i-1} + 2u_i - u_{i+1}) & (1 < i < N) \\ 0 &= \frac{\partial E(u)}{\partial u_N} = u_N - f_N + \alpha(u_{N-1} + u_N) & (i = N). \end{aligned} \quad (3.4)$$

Das daraus resultierende lineare Gleichungssystem lässt sich wie folgt in der Matrixform $Ax = b$ darstellen:

$$\begin{pmatrix} 1 + \alpha & -\alpha & & & \\ -\alpha & 1 + 2\alpha & -\alpha & & \\ & \dots & \dots & \dots & \\ & & -\alpha & 1 + 2\alpha & -\alpha \\ & & & -\alpha & 1 + \alpha \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{pmatrix}. \quad (3.5)$$

Die Matrix A ist hierbei symmetrisch, positiv definit und diagonaldominant.

3.2 Diskreter zweidimensionaler Fall

Im zweidimensionalen Fall ist die Eingabe ein Bild f mit N mal M Pixeln. Der Ähnlichkeitsterm summiert nun die quadratische Differenz aller Pixel $u_{i,j}$ mit dem zugehörigen Pixel $f_{i,j}$. Der Glattheitsterm summiert die Nachbarschaftsdifferenzen aller Pixel. Die Nachbarschaftsdifferenz eines Pixels $u(i,j)$ umfasst nun die unmittelbare Nachbarschaft $\mathcal{N}(i,j)$, also die quadratische Differenz in beide Richtungen zwischen jeweils dem Pixel und den Pixeln links, rechts, oben und unten davon:

$$\begin{aligned} &\frac{1}{2} \sum_{(k,l) \in \mathcal{N}(i,j)} (u_{i,j} - u_{k,l})^2 \\ &= \frac{1}{2} ((u_{i,j} - u_{i-1,j})^2 + (u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i,j-1})^2 + (u_{i,j} - u_{i,j+1})^2 + \\ &\quad (u_{i-1,j} - u_{i,j})^2 + (u_{i+1,j} - u_{i,j})^2 + (u_{i,j-1} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2) \\ &= (u_{i,j} - u_{i-1,j})^2 + (u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i,j-1})^2 + (u_{i,j} - u_{i,j+1})^2 \end{aligned} \quad (3.6)$$

mit Randbedingungen $u_{0,j} = u_{1,j}$, $u_{N+1,j} = u_{N,j}$, $u_{i,0} = u_{i,1}$ und $u_{i,N+1} = u_{i,N}$.

Das Energiefunktional des diskreten zweidimensionalen quadratischen Ansatzes lautet somit:

$$E(u) = \frac{1}{2} \sum_{i=1, j=1}^{N, M} (u_{i,j} - f_{i,j})^2 + \frac{\alpha}{2} \sum_{i=1, j=1}^{N, M} \sum_{(k,l) \in N(i,j)} \frac{1}{2} (u_{i,j} - u_{k,l})^2. \quad (3.7)$$

Im zweidimensionalen Fall gibt es insgesamt neun verschiedene Fälle. Für alle Pixel, die nicht an den Ecken oder Kanten liegen ergibt sich folgende allgemeine Ableitung:

$$\begin{aligned} 0 &= \frac{\partial}{\partial u_{i,j}} \frac{1}{2} \sum_{i=1, j=1}^{N, M} (u_{i,j} - f_{i,j})^2 + \frac{\alpha}{2} \sum_{i=1, j=1}^{N, M} \sum_{(k,l) \in N(i,j)} \frac{1}{2} (u_{i,j} - u_{k,l})^2 \\ &= (u_{i,j} - f_{i,j}) + \alpha((u_{i,j} - u_{i-1,j}) + (u_{i,j} - u_{i,j-1}) + (u_{i,j} - u_{i,j+1}) + (u_{i,j} - u_{i+1,j})) \\ &= (u_{i,j} - f_{i,j}) + \alpha(-u_{i-1,j} - u_{i,j-1} + 4u_{i,j} - u_{i,j+1} - u_{i+1,j}). \end{aligned} \quad (3.8)$$

Zusammen mit allen acht Ausnahmen ergibt sich damit folgendes lineares Gleichungssystem:

$$\begin{aligned} 0 &= \frac{\partial E}{\partial u_{1,1}} = (u_{1,1} - f_{1,1}) + \alpha(2u_{1,1} - u_{1,2} - u_{2,1}) && (i = 1, j = 1) \\ 0 &= \frac{\partial E}{\partial u_{1,j}} = (u_{1,j} - f_{1,j}) + \alpha(-u_{1,j-1} + 3u_{1,j} - u_{1,j+1} - u_{2,j}) && (i = 1, 1 < j < M) \\ 0 &= \frac{\partial E}{\partial u_{1,M}} = (u_{1,M} - f_{1,M}) + \alpha(-u_{1,M-1} + 2u_{1,M} - u_{2,M}) && (i = 1, j = M) \\ 0 &= \frac{\partial E}{\partial u_{i,1}} = (u_{i,1} - f_{i,1}) + \alpha(-u_{i-1,1} + 3u_{i,1} - u_{i+1,2} - u_{i,2}) && (1 < i < N, j = 1) \\ 0 &= \frac{\partial E}{\partial u_{i,j}} = (u_{i,j} - f_{i,j}) + \alpha(-u_{i-1,j} - u_{i,j-1} + 4u_{i,j} - u_{i,j+1} - u_{i+1,j}) && (1 < i < N, 1 < j < M) \\ 0 &= \frac{\partial E}{\partial u_{i,M}} = (u_{i,M} - f_{i,M}) + \alpha(-u_{i-1,M} - u_{i,M-1} + 3u_{i,M} - u_{i+1,M}) && (1 < i < N, j = M) \\ 0 &= \frac{\partial E}{\partial u_{N,1}} = (u_{N,1} - f_{N,1}) + \alpha(-u_{N-1,1} + u_{N,1} - u_{N,2}) && (i = N, j = 1) \\ 0 &= \frac{\partial E}{\partial u_{N,j}} = (u_{N,j} - f_{N,j}) + \alpha(-u_{N-1,j} - u_{N,j-1} + 3u_{N,j} - u_{N,j+1}) && (i = N, 1 < j < M) \\ 0 &= \frac{\partial E}{\partial u_{N,M}} = (u_{N,M} - f_{N,M}) + \alpha(-u_{N-1,M} - u_{N,M-1} + 2u_{N,M}) && (i = N, j = M). \end{aligned} \quad (3.9)$$

3.3 Jacobi-Verfahren

Das Jacobi-Verfahren [Mei11] ist ein iterativer Algorithmus zur näherungsweisen Lösung von linearen Gleichungssystemen. Voraussetzung für die Anwendung ist ein lineares Gleichungssystem der Form $Ax = b$. Sei $A = D - N$, wobei D eine Diagonalmatrix und N der Rest ist, so kann $Dx = Nx + b$ iterativ mit folgender Gleichung gelöst werden:

$$x^{(k+1)} = D^{-1}(Nx^{(k)} + b). \quad (3.10)$$

Angewendet auf unser lineares Gleichungssystem 3.9 ergibt sich folgende Formel:

$$u_{i,j}^{(t+1)} = \frac{f_{i,j} + \alpha \sum_{k,l \in N(i,j)} u_{k,l}^{(t)}}{1 + \alpha \sum_{k,l \in N(i,j)} 1}. \quad (3.11)$$

Die Konvergenz des Jacobi-Verfahrens ist für eine symmetrische, positiv definite Matrix A, was diese in unserem Fall ist, bewiesen [Mei11].

Für die Implementierung des Jacobi-Verfahrens (siehe Algorithmus 3.1) können die Ausnahmen an den Ecken und Kanten des Bildes mit booleschen Ausdrücken geregelt werden. Für Pixel am Bildrand werden alle Nachbarn, die über das Bild hinausreichen, in Zähler und Nenner der Gleichung 3.11 ignoriert.

Algorithmus 3.1 Quadratischer Ansatz mit Jacobi-Verfahren

```

1: for t:=0 to Iterationen do
2:   for i:=0 to N do
3:     for j:=0 to M do
4:        $x_p := (i < N)$ 
5:        $x_m := (i > 0)$ 
6:        $y_p := (j < M)$ 
7:        $y_m := (j > 0)$ 
8:        $u_t[i, j] := \frac{f[i, j] + \alpha \cdot (x_m \cdot u[i-1, j] + y_m \cdot u[i, j-1] + y_p \cdot u[i, j+1] + x_p \cdot u[i+1, j])}{1 + \alpha \cdot (x_p + x_m + y_m + y_p)}$ 
9:     end for
10:   end for
11:    $u := u_t$ 
12: end for
```

Folgende Abbildung 3.1 zeigt das Originalbild, das Eingabebild, welches durch Hinzufügen von Gaußschem Rauschen mit Standardabweichung $\sigma = 20$ entstanden ist, sowie das Endergebnis nach 30 Jacobi-Iterationen mit Parameter $\alpha = 0.92$. Das Bild des entfernten Rauschens zeigt den Unterschied zwischen dem Original und dem entrauschten Bild. Im Idealfall sollte dieses Bild nur das Gaußsche Rauschen und keine Struktur des Originalbilds enthalten. Wie man jedoch direkt erkennen kann, enthält das Bild des entfernten Rauschens auch stark definierte Kanten des Originalbildes. Der PSNR-Wert des Ausgabebilds zum Originalbild ist mit 29.08 eine deutliche Verbesserung zu dem PSNR-Wert von 22.17 des entrauschten Bildes.

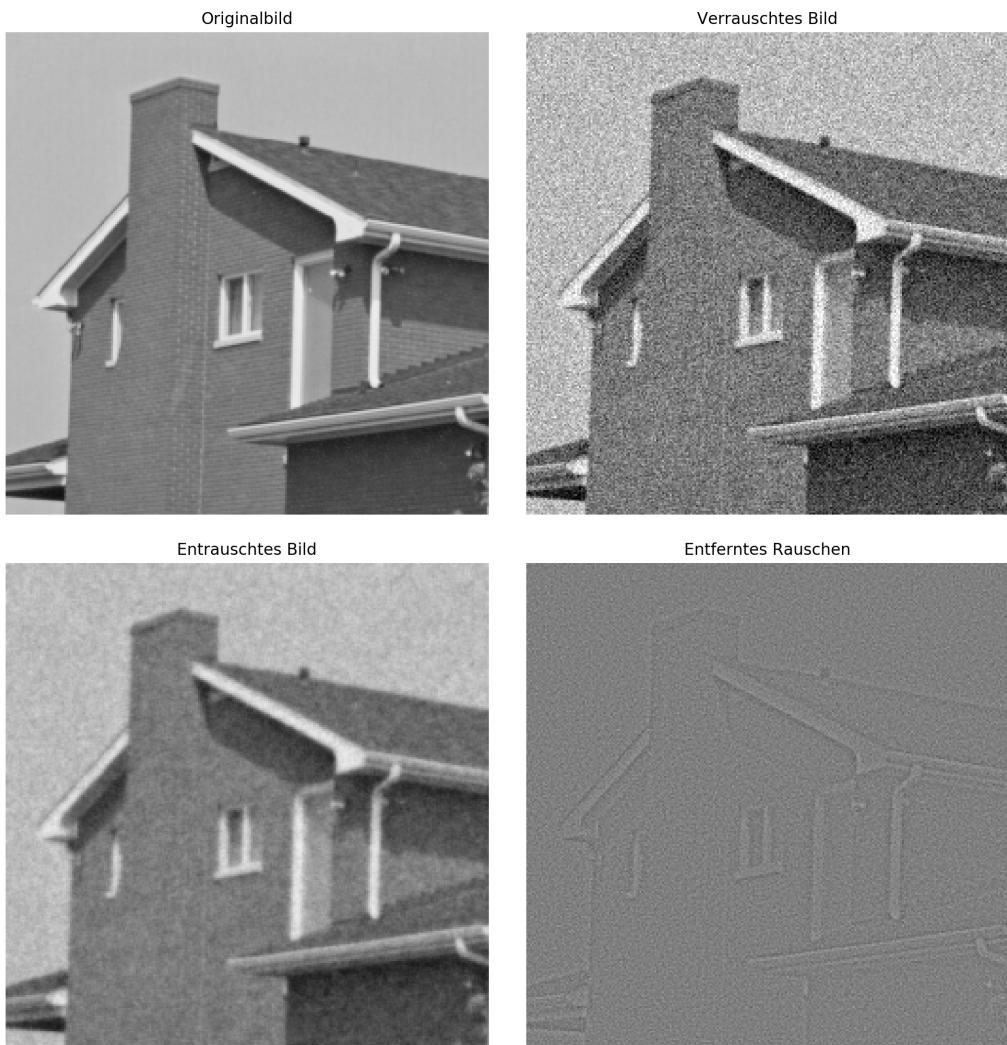


Abbildung 3.1: Jacobi-Verfahren nach 30 Iterationen, $\alpha = 0.92$.

Mit jeder Iteration nähert sich das Jacobi-Verfahren der Lösung des linearen Gleichungssystems an. Wie man jedoch in Abbildung 3.2 sieht, kann dies bei einem schlecht gewählten Parameter α auch zu einer Verschlechterung des PSNR-Wertes führen. Zudem kann man erkennen, dass je höher der Parameter α , desto mehr Iterationen werden benötigt um sich der Lösung des linearen Gleichungssystem anzunähern.

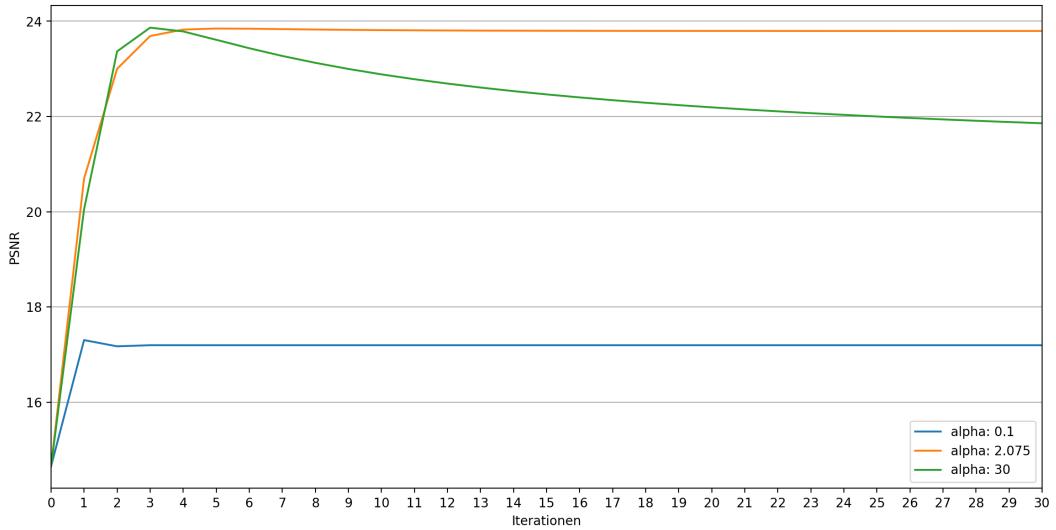


Abbildung 3.2: PSNR-Werte für verschiedene α nach unterschiedlich vielen Iterationen.

3.4 Gauß-Seidel-Verfahren

Das Gauß-Seidel-Verfahren [Mei11] unterscheidet sich vom Jacobi-Verfahren nur durch die sofortige Verwendung ausgerechneter Pixelwerte. Während beim Jacobi-Verfahren das Bild u nur am Ende jeder Iteration verändert wird, schreibt das Gauß-Seidel-Verfahren die neu ausgerechneten Werte direkt in das Bild und verwendet sie umgehend für die nächsten Berechnungen. Im Gegensatz zur Gleichung 3.11 des Jacobi-Verfahrens werden für das Gauß-Seidel-Verfahren zusätzlich die Mengen $N_0(i, j)$ und $N_1(i, j)$ benötigt. Die Menge $N_0(i, j)$ enthält die Nachbarn des Pixels an der Stelle (i, j) , die in dieser Iteration noch nicht verändert wurden, und $N_1(i, j)$ alle Nachbarn, die bereits neu berechnet wurden.

$$u_{i,j}^{(t+1)} = \frac{f_{i,j} + \alpha \sum_{k,l \in N_0(i,j)} u_{k,l}^{(t)} + \alpha \sum_{k,l \in N_1(i,j)} u_{k,l}^{(t+1)}}{1 + \alpha \sum_{k,l \in N(i,j)} 1} \quad (3.12)$$

Die Konvergenz ist für eine strikt diagonaldominante Matrix A, was diese in unserem Fall ist, bewiesen [Mei11].

Im Algorithmus 3.2 des Gauß-Seidel-Verfahrens wird lediglich das temporäre Bild u_t aus dem Jacobi-Algorithmus (siehe Algorithmus 3.1) entfernt und direkt in das Bild u geschrieben.

Algorithmus 3.2 Quadratischer Ansatz mit Gauß-Seidel-Verfahren

```

for t:=0 to Iterationen do
    for i:=0 to N do
        for j:=0 to M do
             $x_p := (i < N)$ 
             $x_m := (i > 0)$ 
             $y_p := (j < M)$ 
             $y_m := (j > 0)$ 
             $u[i, j] := \frac{f[i, j] + \alpha \cdot (x_m \cdot u[i-1, j] + y_m \cdot u[i, j-1] + y_p \cdot u[i, j+1] + x_p \cdot u[i+1, j])}{1 + \alpha \cdot (x_p + x_m + y_m + y_p)}$ 
        end for
    end for
end for

```

Im Vergleich zum Jacobi-Verfahren ist das Gauß-Seidel-Verfahren nicht gut parallelisierbar, da die Gleichungen voneinander abhängig sind. Dafür konvergiert das Gauß-Seidel-Verfahren bereits nach weniger Iterationen zur tatsächlichen Lösung. Die Abbildung 3.3 zeigt, wie sich der PSNR-Wert der beiden Verfahren über die Iterationen einpendelt.

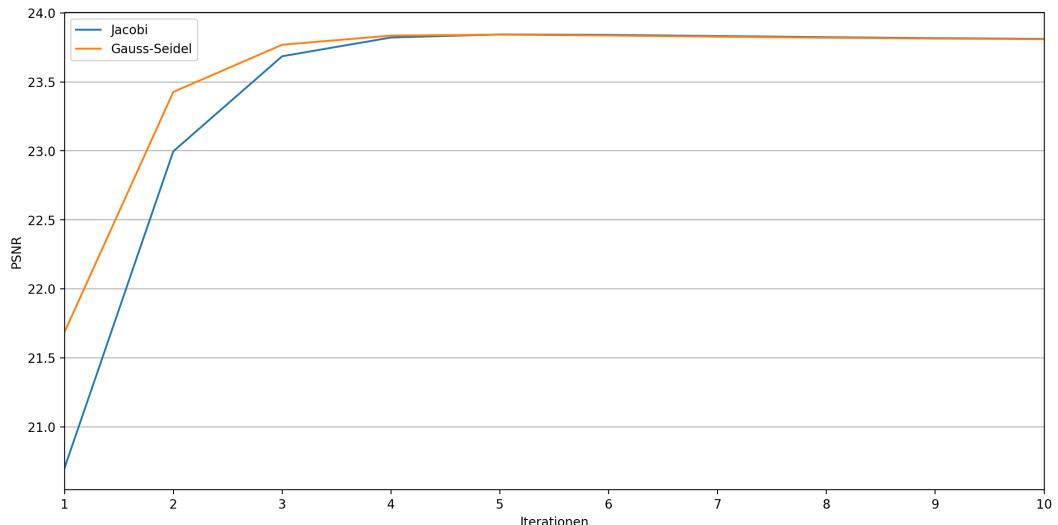


Abbildung 3.3: PSNR-Werte der Jacobi- und Gauss-Seidel-Verfahren im Vergleich, $\alpha = 2.06$.

3.5 SOR-Verfahren

Um das Gauß-Seidel-Verfahren noch schneller konvergieren zu lassen, wird bei dem Successive-Over-Relaxation-Verfahren [Mei11] ein gewichtetes Produkt aus der vorherigen Iteration und dem Wert, den das Gauß-Seidel-Verfahren für die nächste Iteration liefert, berechnet. Die dadurch entstehende Gleichung 3.13 benötigt einen weiteren Parameter $\omega \in [0, 2]$, der die Gewichtung bestimmt. Bei $\omega = 1$ entspricht das SOR-Verfahren dem Gauß-Seidel-Verfahren.

$$u_{i,j}^{(t+1)} = (1 - \omega)u_{i,j}^t + \omega \frac{f_{i,j} + \alpha \sum_{k,l \in N_0(i,j)} u_{k,l}^{(t)} + \alpha \sum_{k,l \in N_1(i,j)} u_{k,l}^{(t+1)}}{1 + \alpha \sum_{k,l \in N(i,j)} 1} \quad (3.13)$$

Für Parameter $\omega \in [0, 2)$ ist die Konvergenz des SOR-Verfahrens für eine symmetrische, positiv-definite Matrix A bewiesen [Mei11].

In den folgenden Abbildungen 3.4 und 3.5 werden verschiedene Werte für den Parameter ω anhand des PSNR-Wertes über die Anzahl der Iterationen verglichen. Es zeigt sich, dass das SOR-Verfahren mit einem gut gewählten Parameter ω wesentlich schneller konvergiert als das Gauß-Seidel-Verfahren.

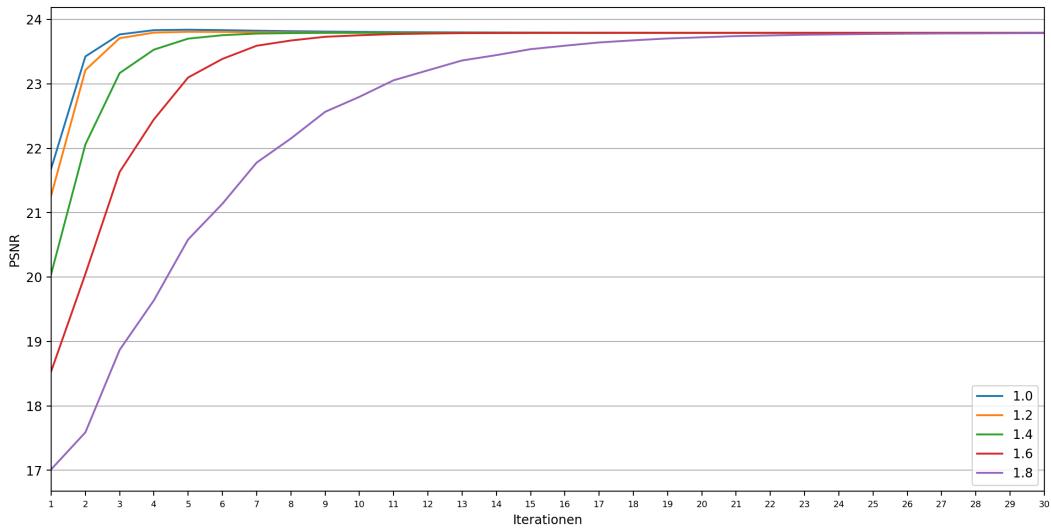


Abbildung 3.4: Verschiedene Werte für ω verglichen, $\alpha = 2.06$.

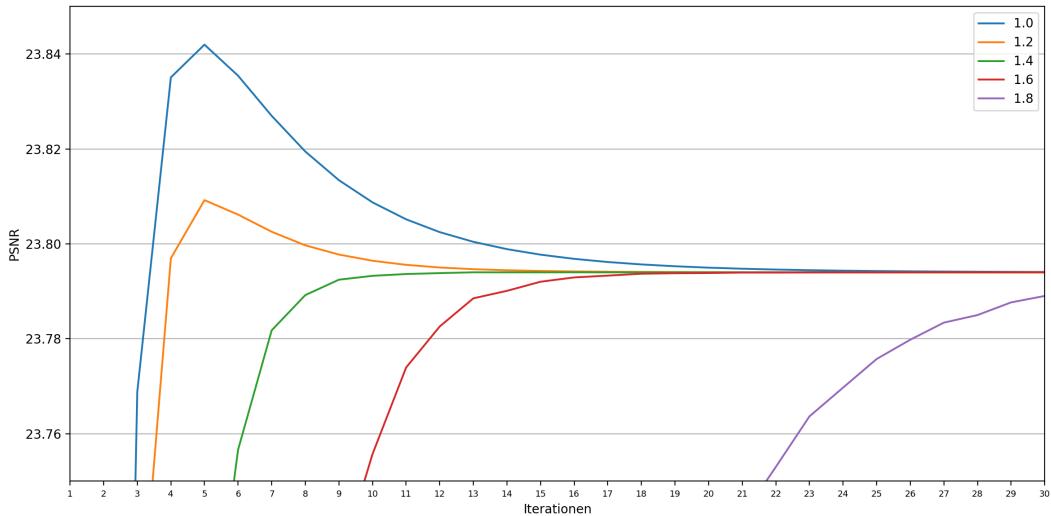


Abbildung 3.5: Vergrößerter Ausschnitt aus Abbildung 3.4.

4 Subquadratischer Ansatz

Im quadratischen Ansatz werden große Abweichungen der Glattheit wesentlich stärker bestraft. Da sich nicht nur durch Rauschen veränderte Pixel von ihren Nachbarn stark unterscheiden, sondern vor allem auch Kanten des Originalbildes, führt dies zu einer Weichzeichnung semantisch wichtiger Kanten. Um dieses Problem zu verringern kann man subquadratische Bestrafungsfunktionen, wie die folgende Charbonnier-Funktion ψ [CBAB97], verwenden.

$$\psi(s^2) = 2\lambda^2 \sqrt{1 + \frac{s^2}{\lambda^2}} - 2\lambda^2 \quad (4.1)$$

Die Abbildung 4.1 zeigt den Unterschied zwischen der Quadratischen und der Charbonnier-Bestrafungsfunktion. Wie man erkennt, gleicht sich mit einem höheren Wert für Parameter λ die Charbonnier-Funktion der quadratischen Funktion an. Ein niedriger Wert für Parameter λ bestraft Kanten weniger, sollte jedoch hoch genug gewählt werden um nicht auch das Rauschen zu erhalten.

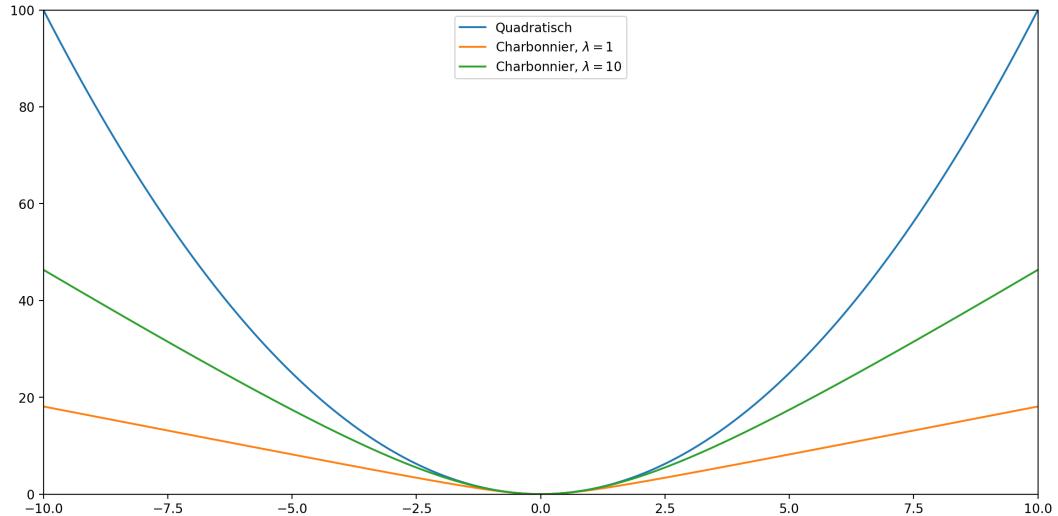


Abbildung 4.1: Quadratische- und Charbonnier-Bestrafungsfunktion im Vergleich.

Diese Funktion wird direkt auf den Glattheitsterm angewendet, wodurch dann im kontinuierlichen Fall folgendes Energiefunktional 4.2 entsteht:

$$E(u) = \int_{\Omega} \frac{1}{2}(u - f)^2 + \frac{\alpha}{2}\psi(|\nabla u|^2) dx dy. \quad (4.2)$$

4.1 Diskreter eindimensionaler Fall

Es wird wieder zunächst der diskrete eindimensionale Fall gezeigt. Mit dem Eingangssignal $f = (f_1, f_2, \dots, f_N)$ und dem Ausgangssignal $u = (u_1, u_2, \dots, u_N)$ ergibt sich folgendes Energiefunktional:

$$E(u) = \frac{1}{2} \sum_{i=1}^N (u_i - f_i)^2 + \frac{\alpha}{2} \sum_{i=1}^N \psi\left(\frac{1}{2}(u_i - u_{i-1})^2 + \frac{1}{2}(u_i - u_{i+1})^2\right) \quad (4.3)$$

mit Randbedingungen $u_0 = u_1$ und $u_N + 1 = u_N$.

Für den allgemeinen Fall in der Mitte des Signals ergibt sich folgende Ableitung:

$$\begin{aligned} 0 &= \frac{\partial}{\partial u_i} \frac{1}{2} \sum_{i=1}^N (u_i - f_i)^2 + \frac{\alpha}{2} \sum_{i=1}^N \psi\left(\frac{1}{2}(u_i - u_{i-1})^2 + \frac{1}{2}(u_i - u_{i+1})^2\right) \\ &= u_i - f_i + \alpha\left(\frac{\psi'_i + \psi'_{i-1}}{2}(u_i - u_{i-1}) + \frac{\psi'_i + \psi'_{i+1}}{2}(u_i - u_{i+1})\right) \end{aligned} \quad (4.4)$$

mit

$$\psi'_i = \psi'\left(\frac{1}{2}(u_i - u_{i-1})^2 + \frac{1}{2}(u_i - u_{i+1})^2\right) \quad (4.5)$$

und

$$\psi'(s^2) = \frac{1}{\sqrt{1 + \frac{s^2}{\lambda^2}}} \quad (4.6)$$

Mit den Ausnahmen am der linken und rechten Ende des Signals ergeben sich damit folgende drei Fälle:

$$\begin{aligned} 0 &= \frac{\partial E}{\partial u_1} = u_1 - f_1 + \alpha\left(\frac{\psi'_1 + \psi'_2}{2}(u_1 - u_2)\right) && (i = 1) \\ 0 &= \frac{\partial E}{\partial u_i} = u_i - f_i + \alpha\left(\frac{\psi'_i + \psi'_{i-1}}{2}(u_i - u_{i-1}) + \frac{\psi'_i + \psi'_{i+1}}{2}(u_i - u_{i+1})\right) && (1 < i < N) \\ 0 &= \frac{\partial E}{\partial u_N} = u_N - f_N + \alpha\left(\frac{\psi'_N + \psi'_{N-1}}{2}(u_N - u_{N-1})\right) && (i = N). \end{aligned} \quad (4.7)$$

4.2 Diskreter zweidimensionaler Fall

Der zweidimensionale Fall unterscheidet sich wieder durch eine vergrößerte Nachbarschaft. Es ergibt sich folgendes Energiefunktional:

$$E = \frac{1}{2} \sum_{i=1, j=1}^{N, M} (u_{i,j} - f_{i,j})^2 + \frac{\alpha}{2} \sum_{i=1, j=1}^{N, M} \psi \left(\sum_{(k,l) \in \mathcal{N}(i,j)} \frac{1}{2} (u_{i,j} - u_{k,l})^2 \right). \quad (4.8)$$

Wie auch im quadratischen Fall wird wieder die allgemeine Ableitung bestimmt um ein Gleichungssystem für die Minimierung des Energiefunktional zu finden:

$$\begin{aligned} & \frac{\partial}{\partial u_{i,j}} \frac{1}{2} \sum_{i=1, j=1}^{N, M} (u_{i,j} - f_{i,j})^2 + \frac{\alpha}{2} \sum_{i=1, j=1}^{N, M} \psi \left(\sum_{(k,l) \in \mathcal{N}(i,j)} \frac{1}{2} (u_{i,j} - u_{k,l})^2 \right) \\ &= (u_{i,j} - f_{i,j}) + \alpha \sum_{(k,l) \in \mathcal{N}(i,j)} \left(\frac{\psi'_{i,j} + \psi'_{k,l}}{2} \right) (u_{i,j} - u_{k,l}) \end{aligned} \quad (4.9)$$

mit

$$\psi'_{i,j} = \psi' \left(\sum_{(k,l) \in \mathcal{N}(i,j)} \frac{1}{2} (u_{i,j} - u_{k,l})^2 \right). \quad (4.10)$$

Das resultierende Gleichungssystem ist nicht mehr linear, da die Ableitung von ψ (siehe Formel 4.6) von u abhängig ist.

4.3 Lagged-Nonlinearity-Verfahren

Da sowohl das Jacobi-Verfahren, das Gauß-Seidel-Verfahren als auch das SOR-Verfahren zur Lösung von nichtlinearen Gleichungssystemen nicht geeignet sind, kann man diese Verfahren nicht für den subquadratischen Fall benutzen. Das Lagged-Nonlinearity-Verfahren [CM99] berechnet die nichtlinearen Teile des Gleichungssystems zuerst und ersetzt sie danach durch die berechneten Konstanten. Aus der die allgemeine Ableitung des zweidimensionalen Falles (siehe Gleichung 4.9) entsteht somit mit der Konstanten $\psi'^t_{i,j}$ aus dem vorherigen Zeitschritt t die folgende Gleichung:

$$0 = (u_{i,j}^{t+1} - f_{i,j}) + \alpha \sum_{(k,l) \in \mathcal{N}(i,j)} \left(\frac{\psi'^t_{i,j} + \psi'^t_{k,l}}{2} \right) (u_{i,j}^{t+1} - u_{k,l}^{t+1}) \quad (4.11)$$

mit

$$\psi'^t_{i,j} = \psi' \left(\sum_{(k,l) \in \mathcal{N}(i,j)} \frac{1}{2} (u_{i,j}^t - u_{k,l}^t)^2 \right). \quad (4.12)$$

4 Subquadratischer Ansatz

Das dadurch entstehende lineare Gleichungssystem kann dann mit dem SOR-Verfahren gelöst werden. Um die Konvergenz zu beschleunigen, können für jede Berechnung der ψ' -Werte mehrere Schritte des SOR-Verfahrens ausgeführt werden. Der Algorithmus 4.1 zeigt wie das Lagged-Nonlinearity-Verfahren für den subquadratischen Fall implementiert wurde.

Algorithmus 4.1 Subquadratischer Ansatz mit Lagged-Nonlinearity-Verfahren

```

1: for t:=0 to Iterationen do
2:   for i:=0 to N do
3:     for j:=0 to M do
4:        $s = \frac{1}{2}((u_{i,j} - u_{i-1,j})^2 + (u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i,j-1})^2 + (u_{i,j} - u_{i,j+1})^2)$ 
5:        $\psi'[i,j] = \frac{1}{\sqrt{1+s^2}}$ 
6:     end for
7:   end for
8:   for t:=0 to Schrittgröße do
9:     for i:=0 to N do
10:      for j:=0 to M do
11:         $x_p := (i < N) \cdot (\psi'[i,j] + \psi'[i+1,j])$ 
12:         $x_m := (i > 0) \cdot (\psi'[i,j] + \psi'[i-1,j])$ 
13:         $y_p := (j < M) \cdot (\psi'[i,j] + \psi'[i,j+1])$ 
14:         $y_m := (j > 0) \cdot (\psi'[i,j] + \psi'[i,j-1])$ 
15:         $u[i,j] := (1-\omega) \cdot u[i,j] + \omega \cdot \frac{f[i,j] + \alpha \cdot (x_m \cdot u[i-1,j] + y_m \cdot u[i,j-1] + y_p \cdot u[i,j+1] + x_p \cdot u[i+1,j])}{1 + \alpha \cdot (x_p + x_m + y_m + y_p)}$ 
16:      end for
17:    end for
18:  end for
19: end for

```

In der folgenden Abbildung 4.2 wurde das Lagged-Nonlinearity-Verfahren auf das gleiche Bild der Abbildung 3.1 mit den Parametern $\alpha = 9.06$ und $\lambda = 1.66$ angewendet. Nach 10 Iterationen mit Schrittgröße 10 und SOR-Parameter $\omega = 1.5$ wurde ein wesentlich besserer PSNR-Wert von 31.27 erreicht. Im Vergleich zur Abbildung 3.1 sind die Strukturen im Bild des entfernten Rauschens schwächer definiert.

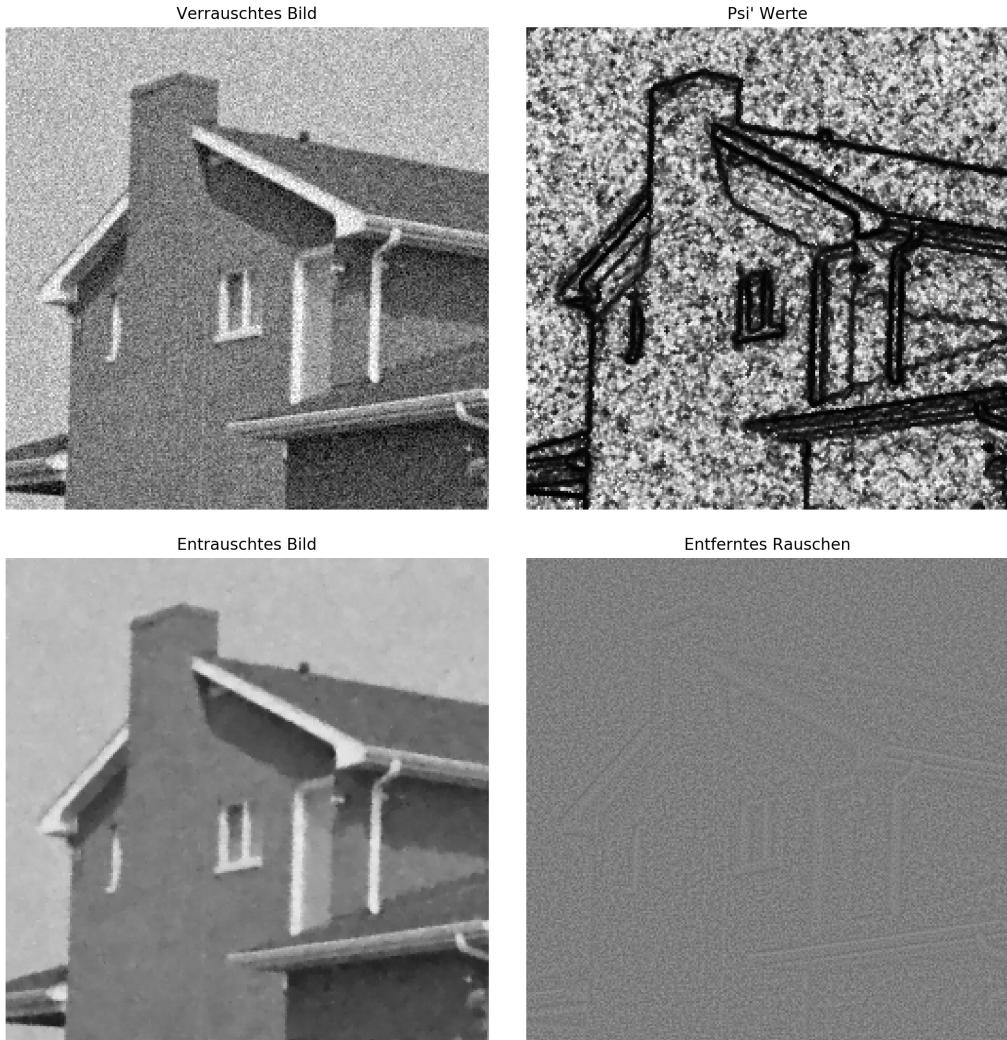


Abbildung 4.2: Lagged-Nonlinearity-Verfahren mit 10 Iterationen mit Schrittgröße 10, $\alpha = 9.06$, $\lambda = 1.66$, $\omega = 1.5$.

4.4 Untersuchung der Parameter

In den folgenden Bilderserien wird der Parameter λ und dessen Einfluss auf das entrauschte Bild untersucht. Als Eingabe wird wieder das verrauschte Bild aus Abbildung 3.1 verwendet. In Abbildung 4.3 wird für alle Bilder ein fester Wert $\alpha = 10$ bei zehn Iterationen mit Schrittgröße zehn und SOR-Parameter $\omega = 1.5$ verwendet. Wie man erkennen kann, erzeugt ein höherer Parameter λ vor allem an den Stellen ohne Kanten eine stärkere Glättung. Um diese erhöhte Glättung auszugleichen, wird in Abbildung 4.4 der Parameter α angeglichen. Anhand der ψ' -Werte kann man nun gut erkennen, wie mit jeder Erhöhung von λ die Kanten schwächer erhalten werden, dafür jedoch immer weniger Rauschen in der Kantenerhaltung auftritt.

4 Subquadratischer Ansatz

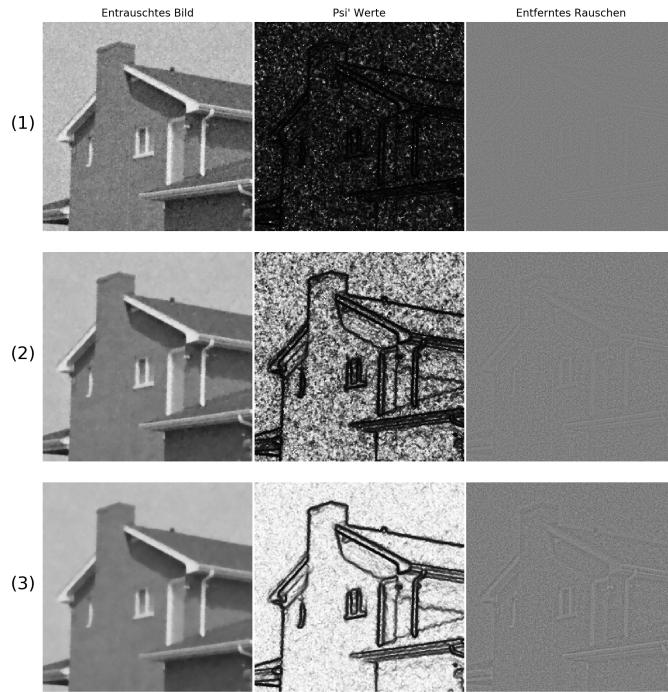


Abbildung 4.3: Einfluss des Parameters λ bei festem $\alpha = 10$. Bilderreihe (1): $\lambda = 0.75$, (2): $\lambda = 1.5$, (3): $\lambda = 3$.

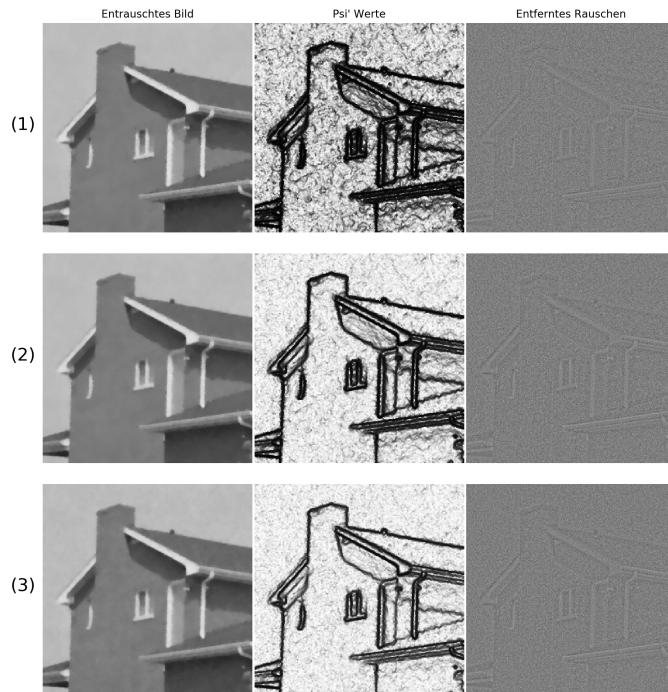


Abbildung 4.4: Bilderreihe (1): $\alpha = 30$, $\lambda = 0.75$, (2): $\alpha = 20$, $\lambda = 1.5$, (3): $\alpha = 10$, $\lambda = 3$.

5 Adaptiver Subquadratischer Ansatz

Der subquadratische Ansatz liefert gute Ergebnisse, wenn die Parameter für ein spezifisches, gleichmäßig verrauscht Eingabebild optimiert wurden. Normalerweise hat man jedoch keinen Zugriff auf die unverrauschten Originalbilder und kann somit die Parameter nicht für jedes Bild einzeln optimieren. Im Idealfall sollte deswegen ein Satz von Parametern für viele verschiedene Eingabebilder mit unterschiedlichem Rauschen ein gutes Ergebnis liefern.

Um das Verfahren robuster gegenüber Veränderungen im Rauschpegel zu machen, wird eine lokale Gewichtung zwischen dem Daten- und dem Glattheitsterm eingeführt. Diese Idee wurde von Hong et al. [HKDB17] für die Bewegungseinschätzung und zur Segmentierung vorgestellt. Die Gewichtung wird lokal anhand des Datenterms bestimmt. Wie stark die Gewichtung ausfällt wird durch den Parameter β geregelt. Der Parameter $\varepsilon > 0$ wird klein gewählt und dient dazu, eine Glättung zu garantieren. In dieser Arbeit wurde $\varepsilon = 0.01$ gewählt. Angewendet auf den subquadratischen Ansatz ergibt sich damit folgendes Energiefunktional:

$$E(u) = \int_{\Omega} c_{u,f} \frac{1}{2} (u - f)^2 + (1 - c_{u,f}) \frac{\alpha}{2} \psi(|\nabla u|^2) dx dy \quad (5.1)$$

mit der Gewichtungsfunktion

$$c_{u,f} = (1 - \varepsilon) e^{-\frac{(u-f)^2}{\beta^2}}. \quad (5.2)$$

Diese Veränderung führt dazu, dass an Stellen, an denen das Ausgabesignal u stark vom Eingabesignal f abweicht, der Glattheitsterm im Vergleich zum Datenterm höher gewichtet wird. Da mit höherem Rauschen ein größerer Unterschied zwischen dem verrauschten Signal und dem entrauschten Ausgabesignal entsteht, wird die Gewichtung des Glattheitsterm in diesem Fall ebenfalls erhöht.

5.1 Diskreter eindimensionaler Fall

Zunächst wird wieder der diskrete eindimensionale Fall gezeigt. Im Vergleich zum eindimensionalen subquadratischen Fall (siehe Gleichung 4.3) wird nur die Gewichtungsfunktion $c_{u,f}$ auf den Datenterm und $(1 - c_{u,f})$ auf den Glattheitsterm multipliziert. Damit ergibt sich folgendes Energiefunktional:

$$E(u) = \frac{1}{2} \sum_{k=1}^N c_{u,f}(u_k - f_k) + \frac{\alpha}{2} \sum_{k=1}^N (1 - c_{u,f})\psi\left(\frac{1}{2}(u_k - u_{k-1})^2 + \frac{1}{2}(u_k - u_{k+1})^2\right) \quad (5.3)$$

mit Randbedingungen $u_0 = u_1$ und $u_{N+1} = u_N$.

Im allgemeinen Fall ergibt sich folgende Ableitung:

$$\begin{aligned} 0 &= \frac{\partial}{\partial u_i} \frac{1}{2} \sum_{k=1}^N c_{u,f}(u_k - f_k) + \frac{\alpha}{2} \sum_{k=1}^N (1 - c_{u,f})\psi\left(\frac{1}{2}(u_k - u_{k-1})^2 + \frac{1}{2}(u_k - u_{k+1})^2\right) \\ &= c_{u_i, f_i}(u_i - f_i) + \alpha\left(\frac{\phi_i + \phi_{i-1}}{2}(u_i - u_{i-1}) + \frac{\phi_i + \phi_{i+1}}{2}(u_i - u_{i+1})\right) \end{aligned} \quad (5.4)$$

mit

$$\phi_i = (1 - c_{u_i, f_i})\psi'_i. \quad (5.5)$$

Mit den zwei Ausnahmefällen an den beiden Rändern des Signals ergibt sich folgendes Gleichungssystem:

$$\begin{aligned} 0 &= c_{u_1, f_1}(u_1 - f_1) + \alpha\left(\frac{\phi_1 + \phi_2}{2}(u_1 - u_2)\right) && (i = 1) \\ 0 &= c_{u_i, f_i}(u_i - f_i) + \alpha\left(\frac{\phi_i + \phi_{i-1}}{2}(u_i - u_{i-1}) + \frac{\phi_i + \phi_{i+1}}{2}(u_i - u_{i+1})\right) && (1 < i < N) \\ 0 &= c_{u_N, f_N}(u_N - f_N) + \alpha\left(\frac{\phi_N + \phi_{N-1}}{2}(u_N - u_{N-1})\right) && (i = N). \end{aligned} \quad (5.6)$$

5.2 Diskreter zweidimensionaler Fall

Im diskreten zweidimensionalen Fall ergibt sich folgendes Energiefunktional:

$$E(u) = \frac{1}{2} \sum_{i=1,j=1}^{N,M} c_{u,f}(u_{i,j} - f_{i,j})^2 + \frac{\alpha}{2} \sum_{i=1,j=1}^{N,M} (1 - c_{u,f})\psi\left(\sum_{(k,l) \in \mathcal{N}(i,j)} \frac{1}{2}(u_{i,j} - u_{k,l})^2\right). \quad (5.7)$$

Die allgemeine Ableitung dieses Energiefunktionalen lautet:

$$\begin{aligned} 0 &= \frac{\partial}{\partial u_{i,j}} \frac{1}{2} \sum_{i=1,j=1}^{N,M} c_{u,f}(u_{i,j} - f_{i,j})^2 + \frac{\alpha}{2} \sum_{i=1,j=1}^{N,M} (1 - c_{u,f})\psi\left(\sum_{(k,l) \in \mathcal{N}(i,j)} \frac{1}{2}(u_{i,j} - u_{k,l})^2\right) \\ &= c_{u_{i,j}, f_{i,j}}(u_{i,j} - f_{i,j}) + \alpha \sum_{(k,l) \in \mathcal{N}(i,j)} \left(\frac{\phi_{i,j} + \phi_{k,l}}{2}\right)(u_{i,j} - u_{k,l}) \end{aligned} \quad (5.8)$$

mit

$$\frac{\phi_{i,j} - \phi_{k,l}}{2} = \frac{\psi'_{i,j}(1 - c_{u_{i,j}, f_{i,j}}) + \psi'_{k,l}(1 - c_{u_{k,l}, f_{k,l}})}{2}. \quad (5.9)$$

Das dadurch entstehende nichtlineare Gleichungssystem kann nun wieder mit dem Lagged-Nonlinearity-Verfahren (siehe Algorithmus 5.1) minimiert werden. Wie im subquadratischen Fall (siehe Formel 4.11) wird die Berechnung so aufgeteilt, dass das entstehende Gleichungssystem linear ist. So werden in jedem Schritt zunächst ψ' und $c_{u,f}$ für alle Pixel berechnet und danach mit diesen festen Werten das Gleichungssystem gelöst.

$$0 = (u_{i,j}^{t+1} - f_{i,j}^{t+1}) + \alpha \sum_{(k,l) \in \mathcal{N}(i,j)} \left(\frac{\psi'_{i,j}^t(1 - c_{u_{i,j}, f_{i,j}}^t) + \psi'_{k,l}^t(1 - c_{u_{k,l}, f_{k,l}}^t)}{2} \right) (u_{i,j}^{t+1} - u_{k,l}^{t+1}) \quad (5.10)$$

Algorithmus 5.1 Adaptiver subquadratischer Ansatz mit Lagged-Non-Linearity-Verfahren

```

1: for t:=0 to Iterationen do
2:   for i:=0 to N do
3:     for j:=0 to M do
4:        $s = \frac{1}{2}((u_{i,j} - u_{i-1,j})^2 + (u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i,j-1})^2 + (u_{i,j} - u_{i,j+1})^2)$ 
5:        $\psi'[i,j] = \frac{1}{\sqrt{1+\frac{s}{\lambda^2}}}$ 
6:     end for
7:   end for
8:   for i:=0 to N do
9:     for j:=0 to M do
10:       $d[i,j] = (u[i,j] - f[i,j])^2$ 
11:       $c[i,j] = (1 - \varepsilon) \cdot \frac{e^{-d[i,j]}}{\beta^2}$ 
12:    end for
13:  end for
14:  for t:=0 to Schrittgröße do
15:    for i:=0 to N do
16:      for j:=0 to M do
17:         $x_p := (i < N) \cdot (\psi'[i,j] \cdot (1 - c[i,j]) + \psi'[i+1,j] \cdot (1 - c[i+1,j]))$ 
18:         $x_m := (i > 0) \cdot (\psi'[i,j] \cdot (1 - c[i,j]) + \psi'[i-1,j] \cdot (1 - c[i-1,j]))$ 
19:         $y_p := (j < M) \cdot (\psi'[i,j] \cdot (1 - c[i,j]) + \psi'[i,j+1] \cdot (1 - c[i,j+1]))$ 
20:         $y_m := (j > 0) \cdot (\psi'[i,j] \cdot (1 - c[i,j]) + \psi'[i,j-1] \cdot (1 - c[i,j-1]))$ 
21:         $u[i,j] := (1 - \omega) \cdot u[i,j] + \omega \cdot \frac{f[i,j] + \alpha \cdot (x_m \cdot u[i-1,j] + y_m \cdot u[i,j-1] + y_p \cdot u[i,j+1] + x_p \cdot u[i+1,j])}{1 + \alpha \cdot (x_p + x_m + y_m + y_p)}$ 
22:      end for
23:    end for
24:  end for
25: end for

```

5 Adaptiver Subquadratischer Ansatz

In der folgenden Abbildung 5.1 wurde ein Gaußsches Rauschen mit unterschiedlicher Standardabweichung auf jeden der vier Quadranten des Originalbildes addiert. Wie man erkennen kann, spiegeln sich diese verschiedenen Rauschpegel in den Werten der adaptiven Gewichtungsfunktion c wieder. In den stärker verrauschten Quadranten führt die Gewichtungsfunktion zu einer stärkeren Glättung.

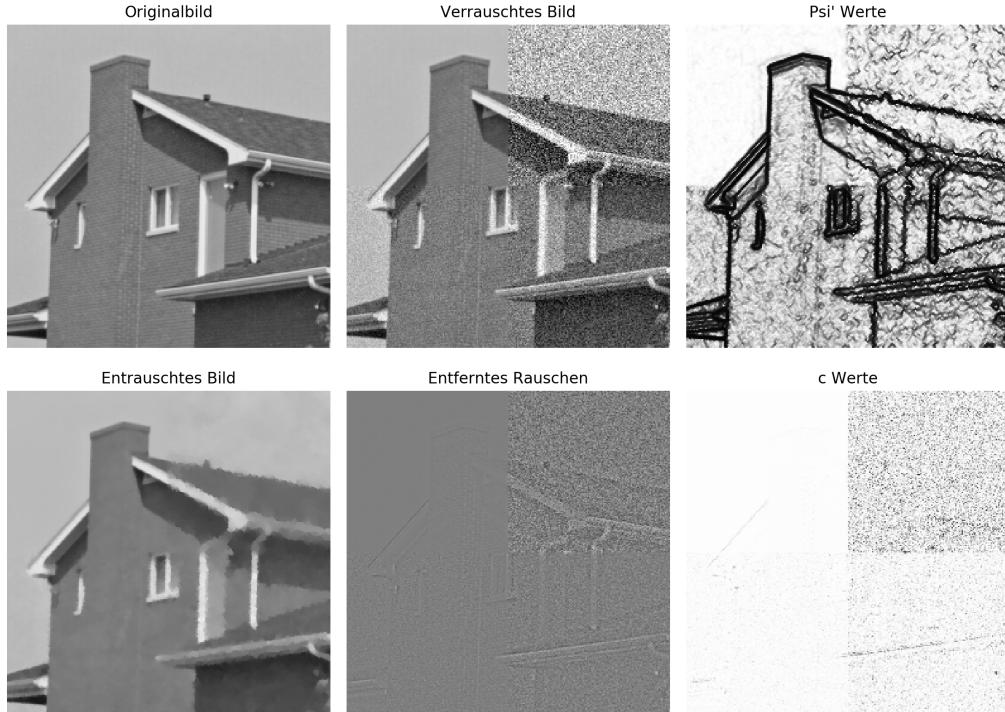


Abbildung 5.1: Adaptiver Subquadratischer Ansatz mit Lagged-Non-Linearity-Verfahren nach zehn Iterationen mit Schrittgröße 10, $\alpha = 300$, $\lambda = 1.5$, $\beta = 100$.

Die Abbildung 5.2 zeigt wie sich die Werte der Gewichtungsfunktion über die Iterationen verändern. So wird in der ersten Iteration wegen der Initialisierung mit $u = f$ nur aufgrund des Parameters $\epsilon > 0$ eine geringe Glättung garantiert. Danach steigen die Werte der Gewichtungsfunktion an Stellen die sich stark vom Eingabebild unterscheiden langsam an. Dies führt dann zu einer verstärkten Glättung an verrauschten Stellen.

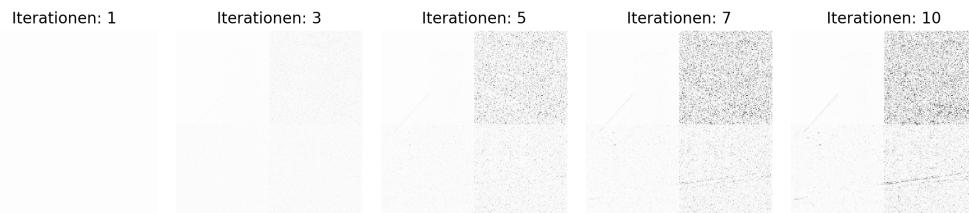


Abbildung 5.2: Werte der Gewichtungsfunktion nach verschiedenen Iterationen.

5.3 Probleme mit feiner Textur

Ein Problem des adaptiven subquadratischen Ansatzes ist, dass zwischen sehr feiner kontrastreicher Textur, wie bei den karierten und linierten Stoffen in Abbildung 5.3, und Rauschen nicht unterschieden werden kann. Dadurch werden diese Texturen genau wie Rauschen behandelt und zu stark geglättet. Dies könnte eventuell in der Struktur-Textur-Zerlegung eine Anwendung finden, ist jedoch für das Bildentrauschen ein Nachteil. Mit einer erhöhten Texturerhaltung, wie es in den Arbeiten von Gilboa et al. [GSZ06] und von Chen et al. [CMSH+10] vorgestellt wurde, könnte man dieses Problem verringern.

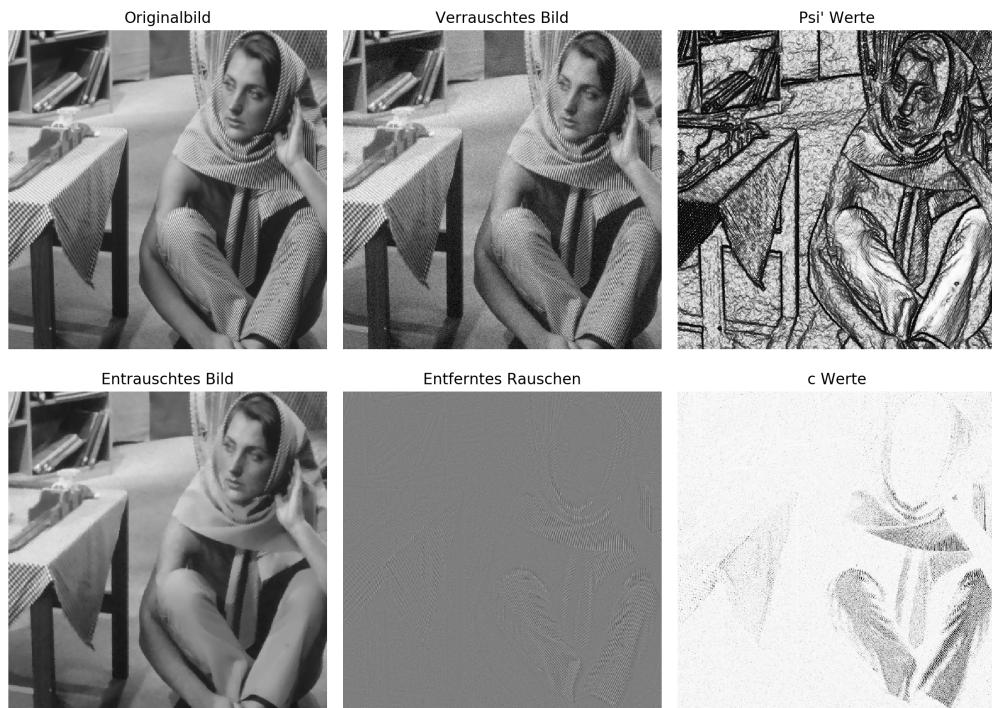


Abbildung 5.3: Probleme des adaptiven Ansatzes bei Bildern mit feiner Textur.

5.4 Verbesserung der adaptiven Gewichtung mit verschiedenen Filtern

Wie man in Abbildung 5.1 erkennen kann, ist das Ausgabebild besonders in den stärker entrauschten Quadranten sehr fleckig. Um dieses Problem zu verringern, kann man den Datenterm innerhalb der adaptiven Gewichtungsfunktion $c_{u,f}$ glätten. Die Glättung durch eine Faltung mit einem Gaußschen Filterkern K_σ (siehe Formel 5.11) führte auch in der Arbeit von Hong et al. [HKDB17] zu besseren Ergebnissen.

$$c_{u,f} = e^{-\frac{K\sigma*(u-f)^2}{\beta^2}}, \quad (5.11)$$

In der folgenden Abbildung 5.4 werden verschiedene Filter in der Gewichtungsfunktion angewendet und deren Einfluss auf das Ausgabebild untersucht. Das Eingabebild entspricht dem der Abbildung 5.1. Im Vergleich zur Bilderreihe (1) in der kein Filter angewendet

5 Adaptiver Subquadratischer Ansatz

wurde, zeigt Bilderreihe (2) die Anwendung eines Gaußschen Filterkerns der Größe 5×5 mit Standardabweichung $\sigma = 1$, was zu einem etwas weniger fleckigen Bild führt. In Bilderreihe (3) wurde der quadratische Variationsansatz (siehe Kapitel 3) mit $\alpha = 10$ verwendet. Durch die gleichmäßige Weichzeichnung der c -Werte entsteht an vielen Stellen ein deutlich besseres Ausgabebild, an den Übergängen zwischen den verschiedenen Rauschpegeln wird jedoch nicht ausreichend entrauscht. In Bilderreihe (4) wurden alle c -Werte gemittelt, wodurch sich die Gewichtungsfunktion nicht mehr dem lokalen Rauschpegel angleicht. Dies führt bei unregelmäßig verteiltem Rauschen zu schlechteren Ergebnissen, ist jedoch für eine gleichmäßige Rauschverteilung ausreichend. Die Wahl eines geeigneten Filters und dessen Parameter sind von der Verteilung des Rauschens abhängig. Welche Filter sich für die verschiedene Arten des Rauschens eignen wird in Kapitel 7 noch weiter untersucht. Für alle weiteren Anwendungen des adaptiven Ansatzes wird, falls nicht explizit erwähnt, die Faltung mit dem Gaußschen Filterkern angewendet.

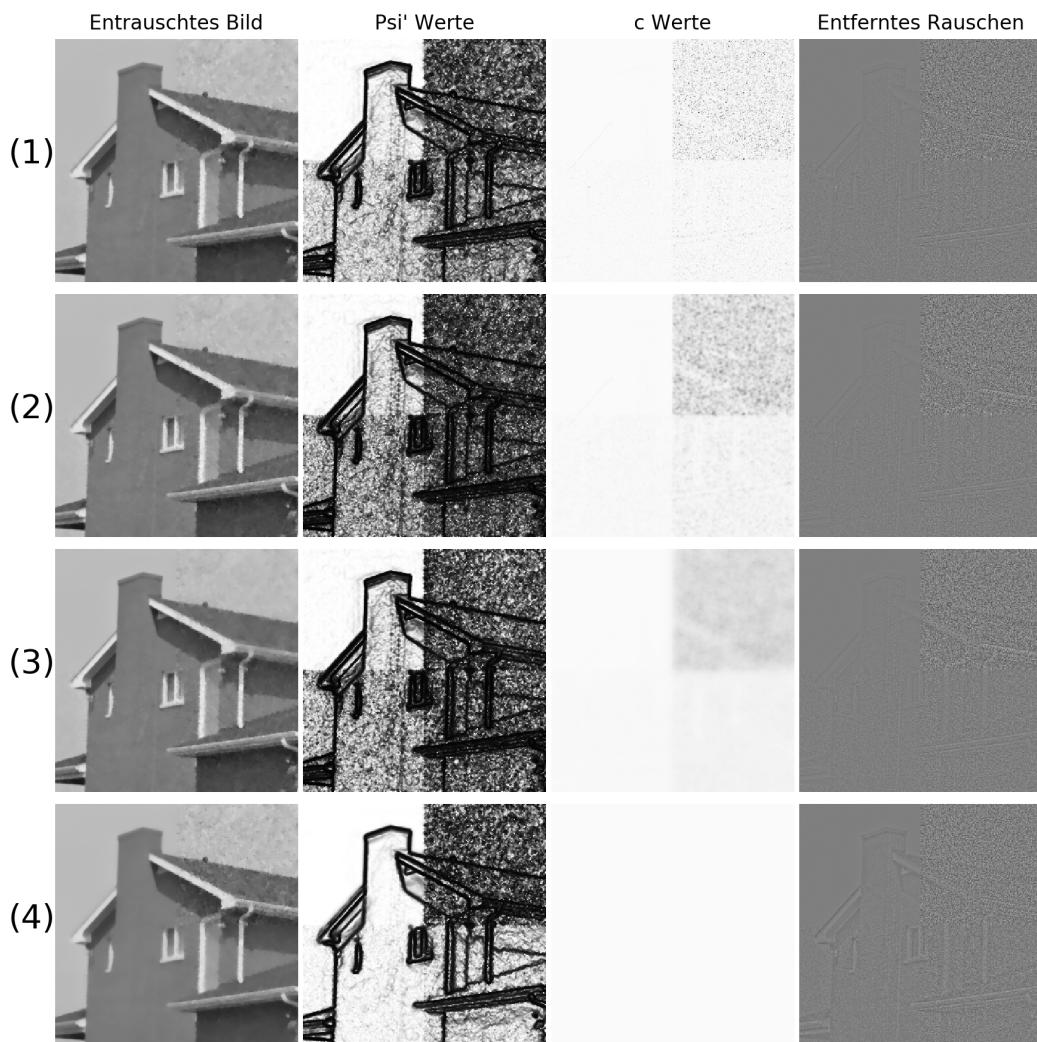


Abbildung 5.4: Verschiedene Filter in der Gewichtungsfunktion. (1): Kein Filter, (2): Gauß-Filter, (3): Quadratischer Variationsansatz, (4): Mittelung.

5.5 Untersuchung der Parameter

In der folgenden Abbildung 5.5 soll das Zusammenspiel des Parameters β der Gewichtungsfunktion mit den Parametern α und λ dargestellt werden. In den ersten beiden Bilderreihen wurden feste Werte für die Parameter α und λ verwendet. Der Parameter β wurde in Bilderreihe (1) absichtlich zu niedrig und in Bilderreihe (2) zu hoch gewählt, um zu zeigen, wie dieser nicht nur die Gewichtung zwischen den Rauschpegeln, sondern auch die allgemeine Glättung beeinflusst. In den Bilderreihen (3) und (4) wurden dann für die selben festen Gewichtungsparameter β die Parameter α und λ für einen maximalen PSNR-Wert optimiert. Wie man erkennen kann, führt die höhere Gewichtung zwischen den Rauschpegeln in Bilderreihe (3) zu einer Erhaltung von mehr Details in den weniger verrauschten Quadranten. In Bilderreihe (4) sind hingegen die Übergänge zwischen den unterschiedlich verrauschten Quadranten weniger auffällig.

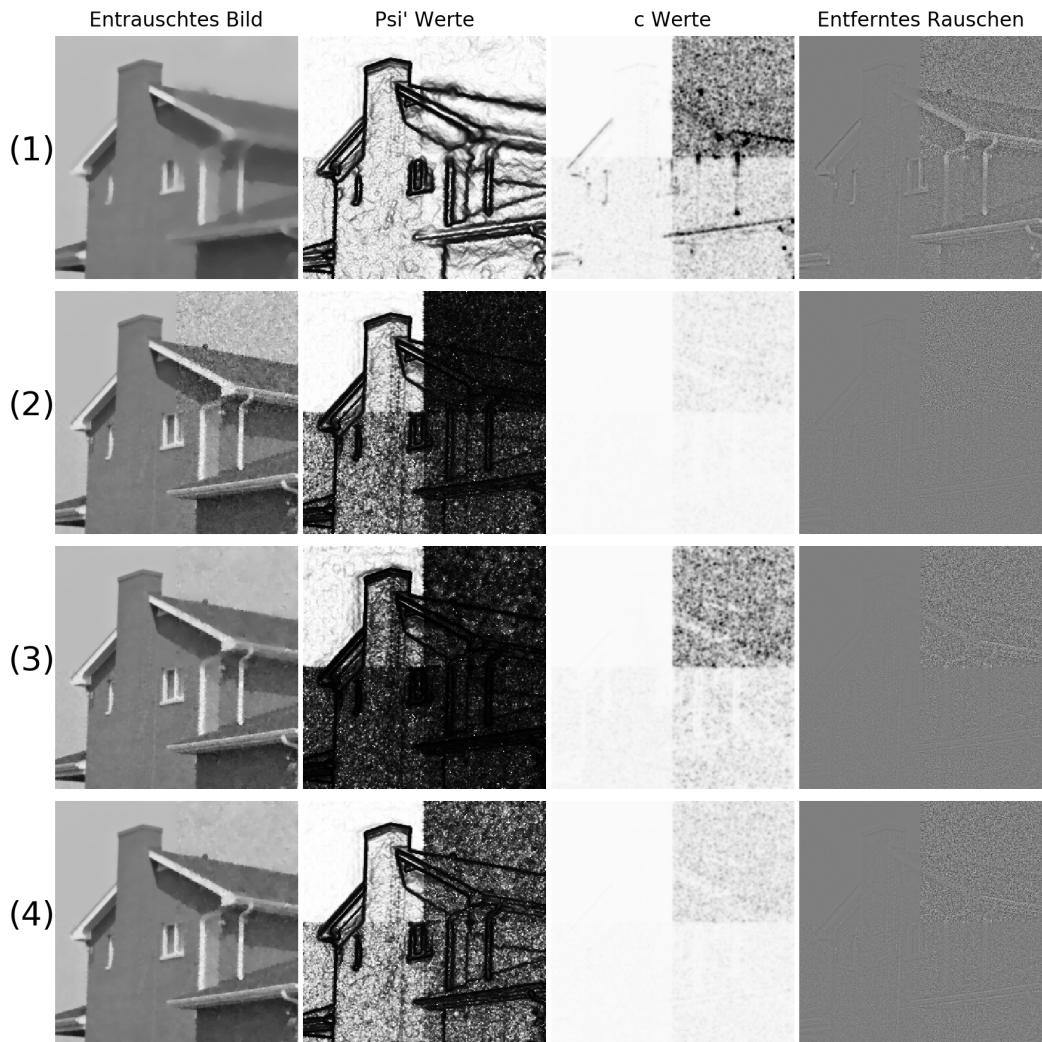


Abbildung 5.5: Bilderreihe (1): $\alpha = 500, \lambda = 1, \beta = 80$, (2): $\alpha = 500, \lambda = 1, \beta = 160$, (3): $\alpha = 724, \lambda = 0.38, \beta = 80$, (4): $\alpha = 557, \lambda = 1.1, \beta = 160$.

6 Adaptiver Quadratischer Ansatz

Eine Implementierung der adaptiven Gewichtung zwischen den Daten- und Glattheitstermen wurde auch für den quadratischen Ansatz getestet, liefert jedoch schlechte Ergebnisse für die Bildentrauschung. Da besonders an Kanten ein großer Unterschied zwischen Eingabebild und Ausgabebild entsteht, verstärkt die adaptive Gewichtung die Glättung an Kanten. Somit wird die Schwäche des quadratischen Ansatzes im Vergleich zum subquadratischen Ansatz, die Weichzeichnung an Kanten, noch weiter verstärkt. Im adaptiven subquadratischen Ansatz hingegen wirkt wie in Abbildung 5.5 gesehen der ψ -Term dieser Glättung der Kanten entgegen.

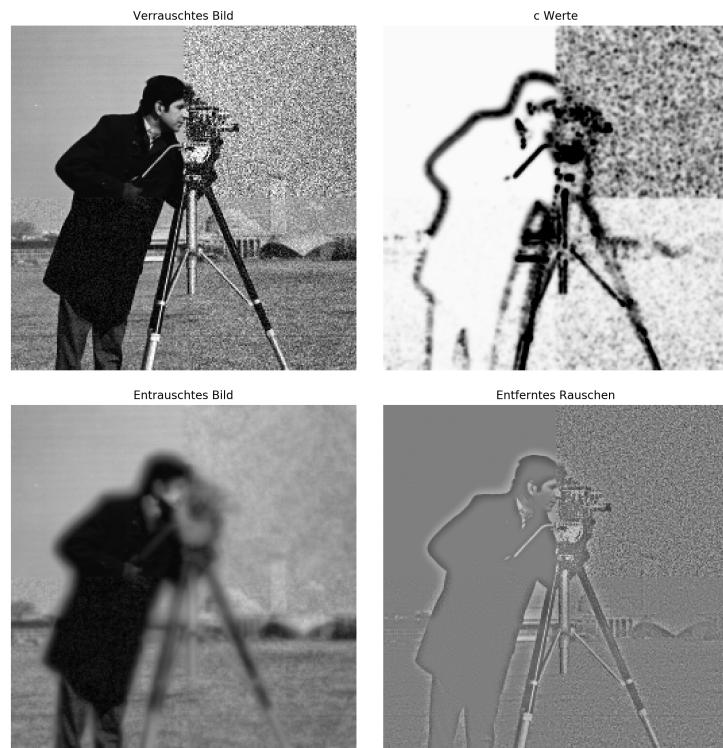


Abbildung 6.1: Adaptiver quadratischer Ansatz führt zu erhöhter Kantenglättung.

Die Abbildung 6.1 zeigt dieses kantenglättende Verhalten des adaptiven quadratischen Ansatzes. Auf das Originalbild wurde wieder unterschiedliches Gaußsches Rauschen auf die vier Quadranten addiert. Wie man besonders an den c -Werten erkennen kann, funktioniert zwar die erhöhte Glättung bei größerem Rauschen, jedoch wird an den tatsächlichen Kanten des Bildes die Gewichtung des Glattheitsterms zu stark.

7 Evaluation

Um die Nützlichkeit des adaptiven subquadratischen Ansatzes weiter zu untersuchen werden die verschiedenen Ansätze in den folgenden Absätzen bei unterschiedlichen Verwendungszwecken verglichen. Der adaptive quadratische Fall wurde aufgrund der schlechten Eignung für das Bildentrauschen nicht weiter evaluiert. Zuerst wird das Verhalten der verschiedenen Verfahren bei ungleichmäßig verteilem Gaußschen Rauschen untersucht. Danach wird der Vorteil der adaptiven Gewichtung bei Salz und Pfeffer Rauschen und multiplikativem Rauschen gezeigt. Im letzten Teil wird die Robustheit der verschiedenen Ansätze über mehrere Bilder und Rauschpegel verglichen.



Abbildung 7.1: Originalbilder.

Für die folgenden Bilderreihen wurde für den quadratischen Ansatz das SOR-Verfahrens mit 100 Iterationen verwendet. Für den subquadratischen und adaptiven subquadratischen Ansatz wurden je zehn Iterationen des Lagged-Nonlinearity-Verfahren mit jeweils zehn Schritten des SOR-Verfahrens pro Iteration verwendet. Für den SOR-Parameter ω wurde 1.5 gewählt. Die restlichen Parameter der verschiedenen Verfahren wurden für einen möglichst hohen PSNR-Wert zwischen Originalbild und Ausgabe optimiert.

7.1 Ungleichmäßig verteiltes Rauschen

In Abbildung 7.2 wurden der quadratische, der subquadratische und der adaptive subquadratische Ansatz auf ein Bild mit variierendem Rauschpegel angewendet. Es wurde ein Gaußsches Rauschen mit Standardabweichung σ gleich 0, 10, 20 und 40 auf die vier Quadranten des Bildes addiert. Die Bilderreihe (1) zeigt den quadratischen Ansatz mit Parameter $\alpha = 1.04$. Der subquadratische Ansatz wird in Bilderreihe (2) mit Parametern $\alpha = 16.87$ und $\lambda = 1.41$ dargestellt. Im Vergleich zu dem quadratischen Ansatz führt der subquadratische Ansatz zu einer stärkeren Entrauschtung, verliert dabei jedoch viele Details in den weniger verrauschten Quadranten. In Bilderreihe (3) wird der adaptive subquadratische Ansatz mit Parametern $\alpha = 495$, $\lambda = 1.03$ und $\beta = 137$ gezeigt. Dieser passt die Glättung dem Rauschpegel der jeweiligen Quadranten an. Dadurch werden in den weniger verrauschten Quadranten mehr Details erhalten und der rechte obere Quadrant wird dennoch ausreichend weichgezeichnet.

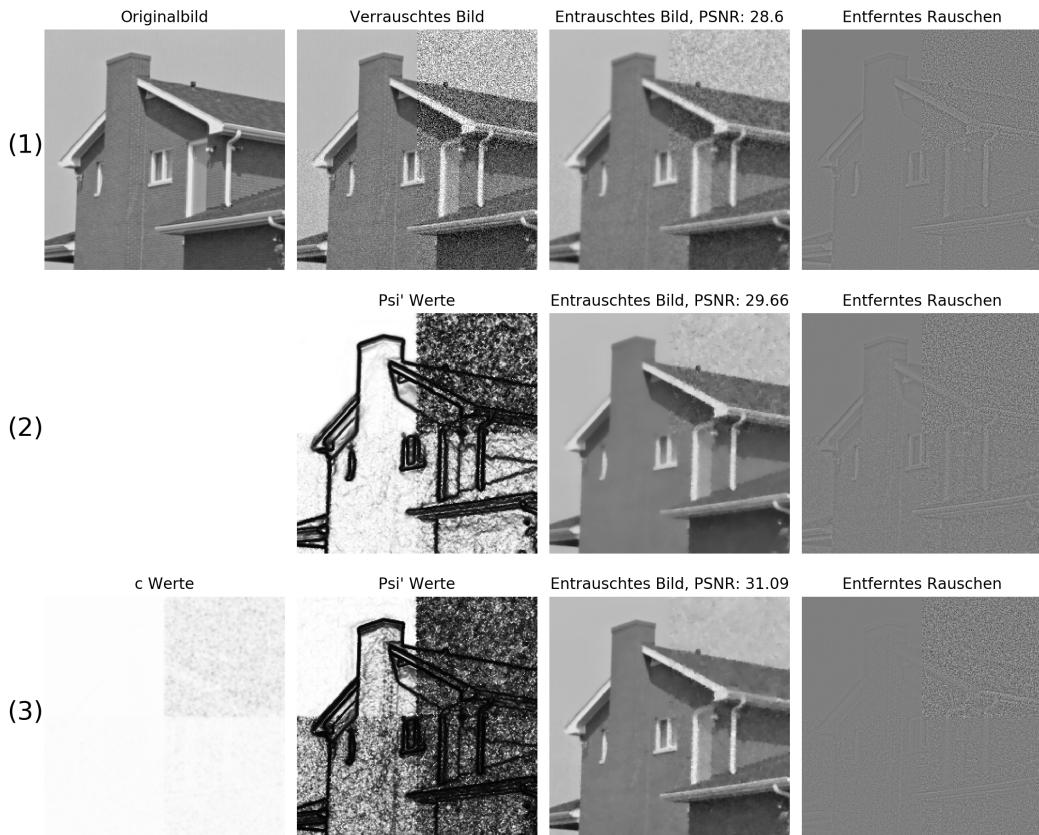


Abbildung 7.2: Bilderreihe (1): Quadratischer Ansatz, (2): Subquadratischer Ansatz, (3): Adaptiver Ansatz.

Für die Abbildung 7.3 wurde dem Originalbild ein Gaußsches Rauschen mit wellenförmig verteilter Standardabweichung σ hinzugefügt. Der quadratische Ansatz in Bilderreihe (1) mit Parameter $\alpha = 0.65$ entfernt nur wenig Rauschen. Der subquadratische Ansatz in Bilderreihe

(2) mit Parametern $\alpha = 12.83$ und $\lambda = 1.55$ entfernt wie auch in Abbildung 7.2 zwar mehr Rauschen als der quadratische Ansatz, verliert jedoch in den weniger verrauschten Regionen viele Details. Die nächsten beiden Bilderreihen verwenden den adaptiven subquadratischen Ansatz mit den Parametern $\alpha = 925$, $\lambda = 0.44$ und $\beta = 119$. Hier soll gezeigt werden, wie eine gleichmäßige Glättung innerhalb des Gewichtungsterms des adaptiven Ansatzes bei einer glatteren Rauschverteilung zu leichten Verbesserungen führt. In Bilderreihe (3) wurde eine Faltung mit einem Gaußschen Filterkern mit $\sigma = 1$ angewendet, während in Bilderreihe (4) eine Glättung mit dem quadratischen Variationsansatz mit $\alpha = 1.5$ verwendet wurde.

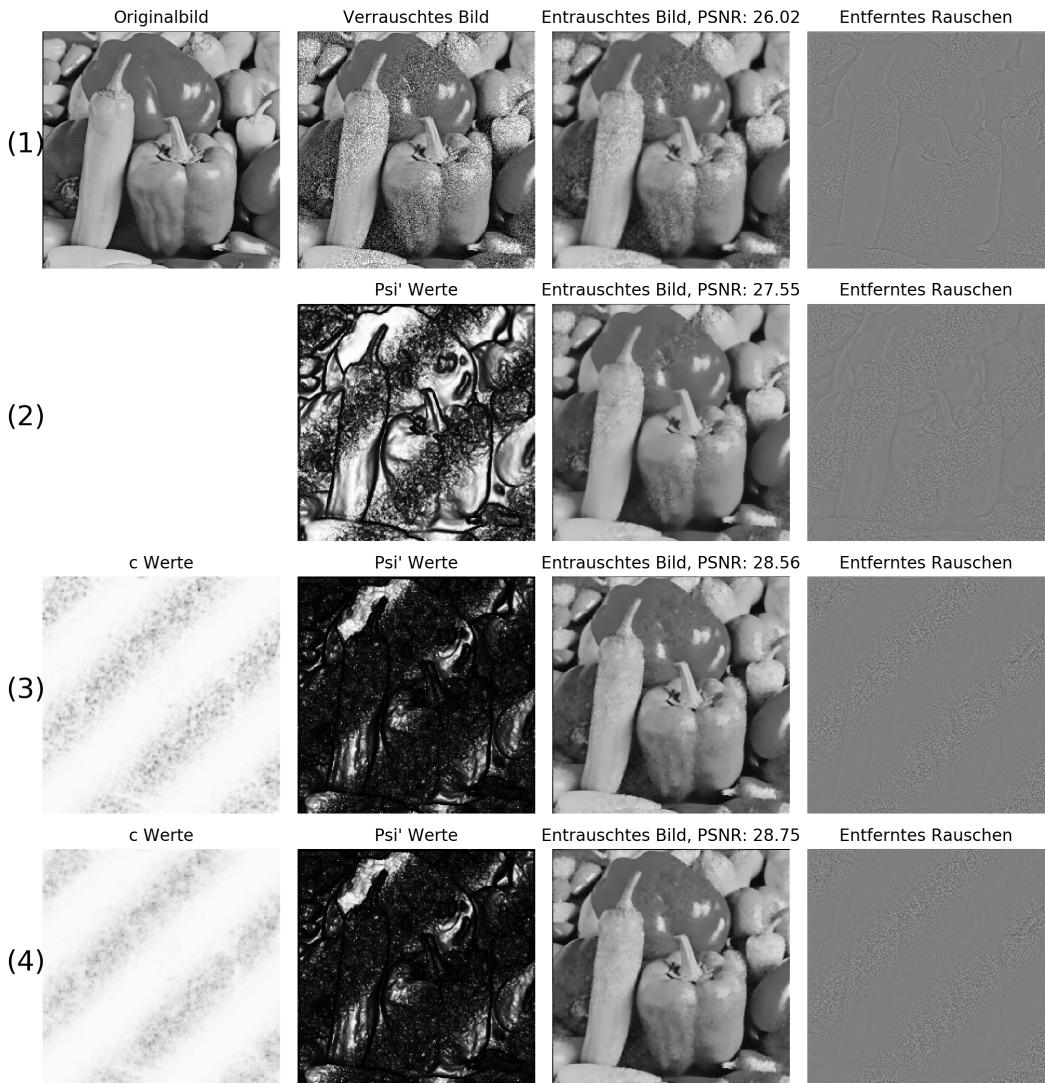


Abbildung 7.3: Bilderreihe (1): Quadratischer Ansatz, (2): Subquadratischer Ansatz, (3): Adaptiver subquadratischer Ansatz mit Gauß-Filter in der Gewichtungsfunktion, (4): Adaptiver subquadratischer Ansatz mit quadratischem Variationsansatz in der Gewichtungsfunktion.

7.2 Salz-und-Pfeffer-Rauschen

In der Abbildung 7.4 wurde auf das Originalbild zunächst ein Gaußsches Rauschen mit Standardabweichung $\sigma = 10$ addiert. Danach wurden zufällig gewählte Pixel auf Wert 0 oder 255 gesetzt, ein sogenanntes Salz-und-Pfeffer-Rauschen.

Die Bilderreihe (1) zeigt den quadratischen Ansatz mit Parameter $\alpha = 0.59$. In Bilderreihe (2) wurde der subquadratische Ansatz mit den Parametern $\alpha = 1.13$ und $\lambda = 31.87$ angewendet. Der adaptive subquadratische Ansatz in Bilderreihe (3) verwendet die Parameter $\alpha = 1151$, $\lambda = 0.25$ und $\beta = 103$. Es wurde keine Faltung des Datenterms (siehe Formel 5.11) in der adaptiven Funktion verwendet.

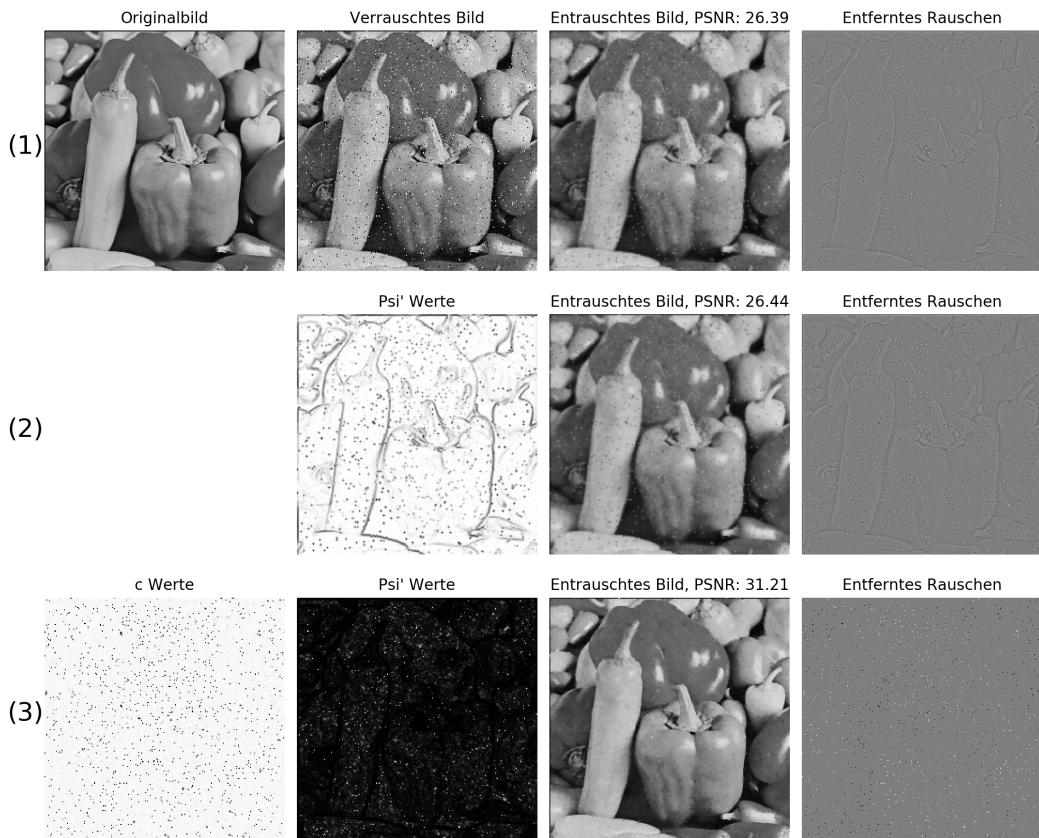


Abbildung 7.4: Bilderreihe (1): Quadratischer Ansatz, (2): Subquadratischer Ansatz, (3): Adaptiver Ansatz.

Wie man erkennen kann, werden die Salz- und Pfefferkörner bei dem quadratischen und subquadratischen Ansatz einfach nur verschmiert, während der adaptiven Ansatz fast alle diese Körner komplett entfernt. Dies führt zu einem wesentlich besseren PSNR-Wert.

7.3 Multiplikatives Rauschen

Für die Abbildung 7.5 wurde Gaußsches Rauschen mit Standardabweichung $\sigma = 0.1$ und Erwartungswert $\mu = 1$ mit dem Originalbild multipliziert. Der quadratische Ansatz in Bilderreihe (1) mit Parameter $\alpha = 0.17$ kann nur wenig Rauschen entfernen. Der subquadratische Ansatz in Bilderreihe (2) mit Parametern $\alpha = 22.92$ und $\lambda = 0.4$ erzielt schon wesentlich bessere Ergebnisse, besonders an dem Kreis verbleibt jedoch noch viel Rauschen. In Bilderreihe (3) wurde der adaptive subquadratische Ansatz mit Parametern $\alpha = 181$, $\lambda = 0.56$, $\beta = 31.24$ und Gaußscher Glättung mit $\sigma = 1$ im Gewichtungsterm verwendet. Hier kann man die Verbesserungen besonders in dem entfernten Rauschen erkennen, so wurde zum Beispiel der Schriftzug kaum weichgezeichnet. Der adaptive subquadratische Ansatz in Bilderreihe (4) verwendet die Parameter $\alpha = 478$, $\lambda = 0.23$ und $\beta = 33.5$ und eine Glättung innerhalb der Gewichtungsfunktion mit dem quadratischen Variationsansatz mit $\alpha = 1.5$. Diese führt wieder zu einer leichten Verbesserung des PSNR-Wertes.

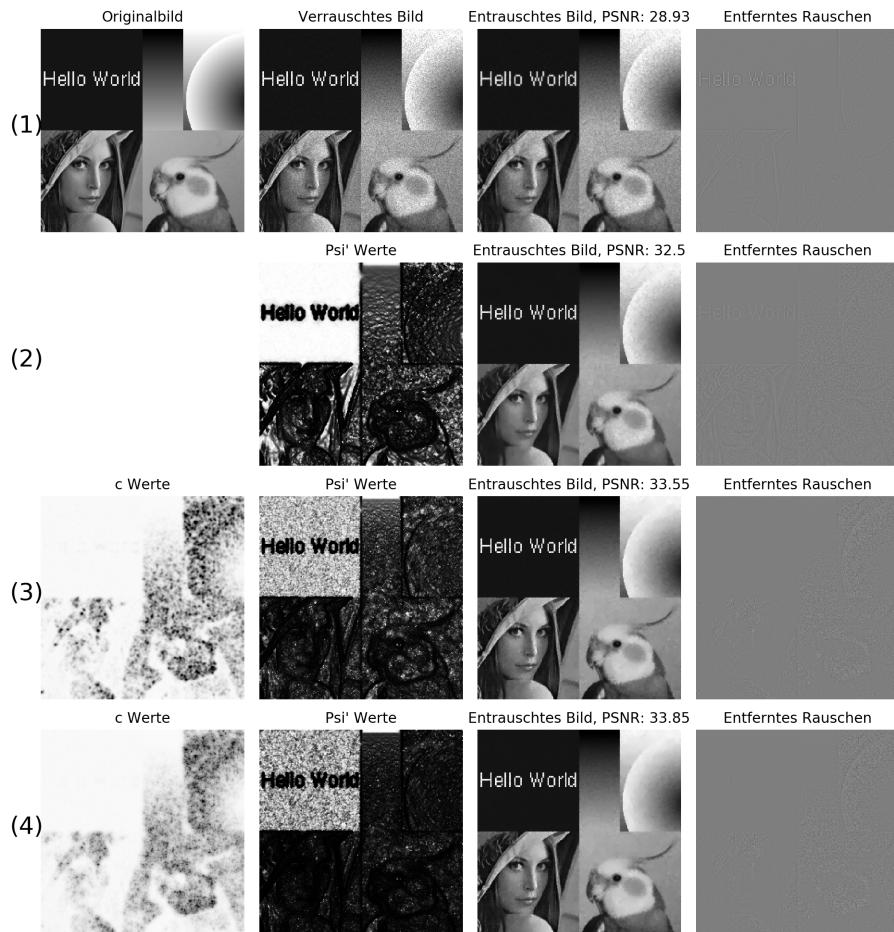


Abbildung 7.5: Bilderreihe (1): Quadratischer Ansatz, (2): Subquadratischer Ansatz, (3): Adaptiver Ansatz mit Gauß-Filter, (4): Adaptiver Ansatz mit quadratischem Variationsansatz in der Gewichtungsfunktion.

7.4 Verschiedene Bilder und Rauschpegel

In der folgender Abbildung 7.6 wurden die vier Originalbilder aus Abbildung 7.1 mit einem Gaußschen Rauschen mit Standardabweichung σ gleich 10, 20 und 40 verrauscht. Diese verrauschten Bilder wurden als Eingabe für die verschiedenen Ansätze verwendet. Die Parameter der Ansätze wurden für den größten addierten PSNR-Wert aller Ausgabebilder gewählt.

Wie man in Bilderserie 7.7 sehen kann, liefert der quadratische Ansatz keine zufriedenstellenden Ergebnisse. Das Rauschen wurde kaum entfernt, da ein höherer Wert für Parameter α die Kanten zu stark aufweichen lässt und damit den PSNR-Wert verringern würde. Insgesamt wird mit dem quadratischen Ansatz ein mittlerer PSNR-Wert von 26.22 erreicht, was im Vergleich zum mittleren PSNR-Werts der verrauschten Bilder von 22.45 dennoch eine große Verbesserung ist.

In Abbildung 7.8 zeigt der subquadratische Ansatz schon wesentlich bessere Ergebnisse. Durch die subquadratische Bestrafung kann das restliche Bild stärker entrauscht werden, ohne dass ein Aufweichen der Kanten auftritt. Diese erhöhte Kantenerhaltung sorgt, wie man gut anhand der unterschiedlich starken Verbesserung der linken Spalte erkennen kann, für eine verbesserte Robustheit gegenüber verschiedenen Eingabebildern. Wie auch bei dem quadratischen Ansatz, zeigt sich die größte Verbesserung bei den Bildern, die mit σ gleich 20 verrauscht wurden. Die geringer verrauschten Bilder werden zu stark weichgezeichnet, während die stärker verrauschten Bilder zu wenig entrauscht werden. Dieser Effekt entsteht, da diese beiden Verfahren keine Möglichkeit haben sich an den Rauschpegel der Bilder anzupassen und somit mit einem mittleren Rauschpegel den größten kombinierten PSNR-Wert erreichen. Der gemittelte PSNR-Wert für die Ausgabebilder des subquadratischen Ansatzes beträgt 27.96.

Der adaptive Ansatz in Bilderserie 7.9 löst genau dieses Problem und zeigt deswegen im Vergleich zu den vorherigen Ansätzen ähnlich gute Verbesserungen für alle Rauschpegel und nicht nur den mittleren. Bei den weniger stark verrauschten Bildern erhält der adaptive Ansatz wesentlich mehr Details. Dies ist besonders an dem Kamin des Hauses sichtbar, der bei dem subquadratischen Ansatz in Abbildung 7.8 komplett mit der Hausfassade verschmilzt. Zudem wird bei den stark verrauschten Bildern das Rauschen besser unterdrückt. Für den mittleren Rauschpegel hingegen erzielt der adaptive Ansatz im Vergleich zu dem subquadratischen Ansatz einen etwas geringeren PSNR-Wert. Insgesamt ergibt sich damit ein deutlich besserer mittlerer PSNR-Wert von 29.19.

Die Abbildung 7.10 zeigt den adaptiven subquadratischen Ansatz mit einer Mittelung der Gewichtungswerte. Diese Mittelung verhindert zwar eine lokale Anpassung an den Rauschpegel, da in diesem Fall jedoch ein gleichmäßig verteiltes Rauschen benutzt wurde, führt dies sogar zu einem leicht verbesserten PSNR-Wert von 29.39.

Insgesamt zeigt sich der adaptive Ansatz deutlich robuster gegenüber variierenden Rauschpegeln. Eine weitere Verbesserung der Robustheit gegenüber verschiedenen Eingabebildern scheint der adaptive Ansatz jedoch nicht zu bringen.



Abbildung 7.6: Testbilder mit hinzugefügtem Gaußschen Rauschen.

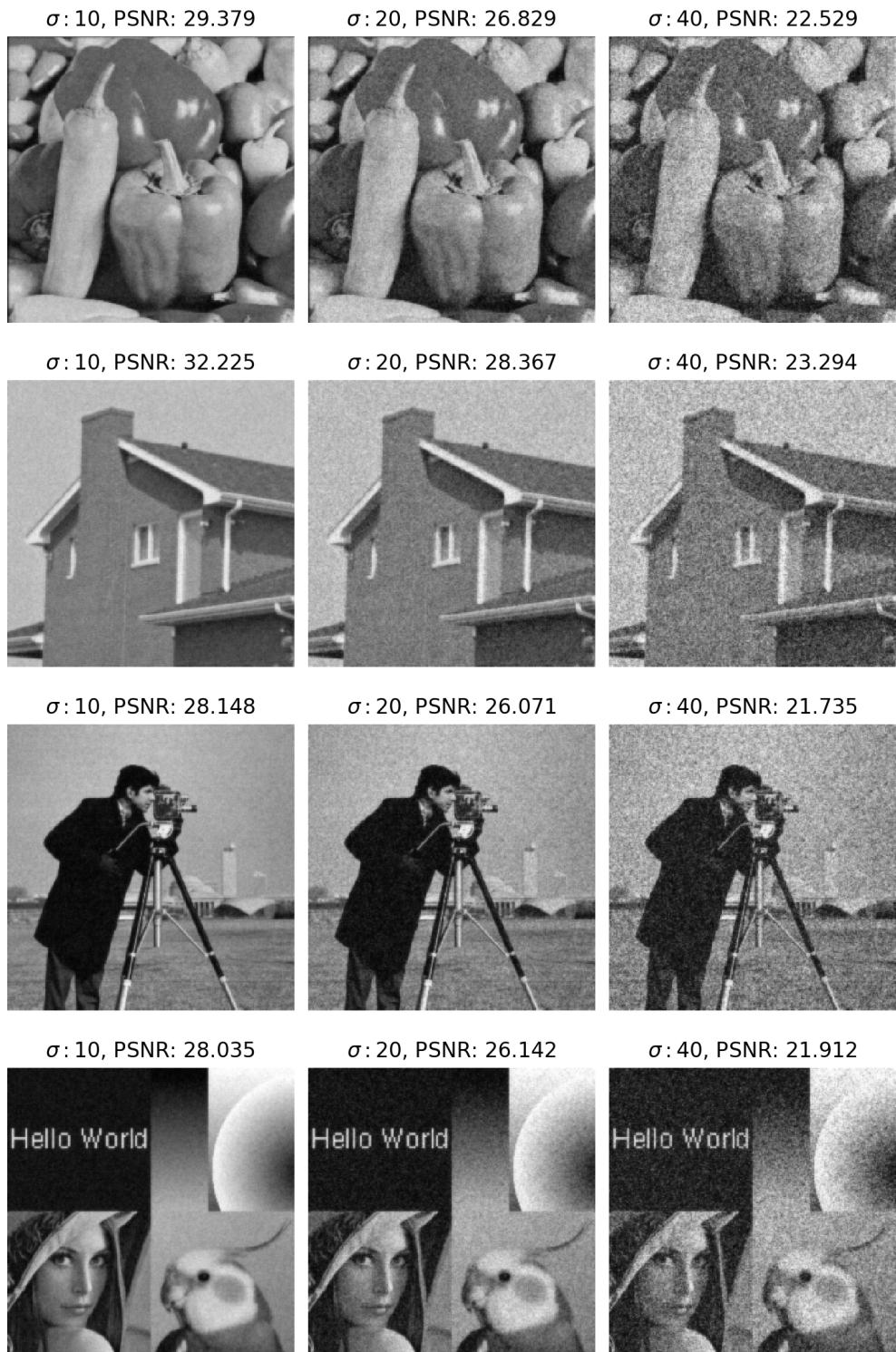


Abbildung 7.7: Quadratischer Ansatz, $\alpha = 0.47$.

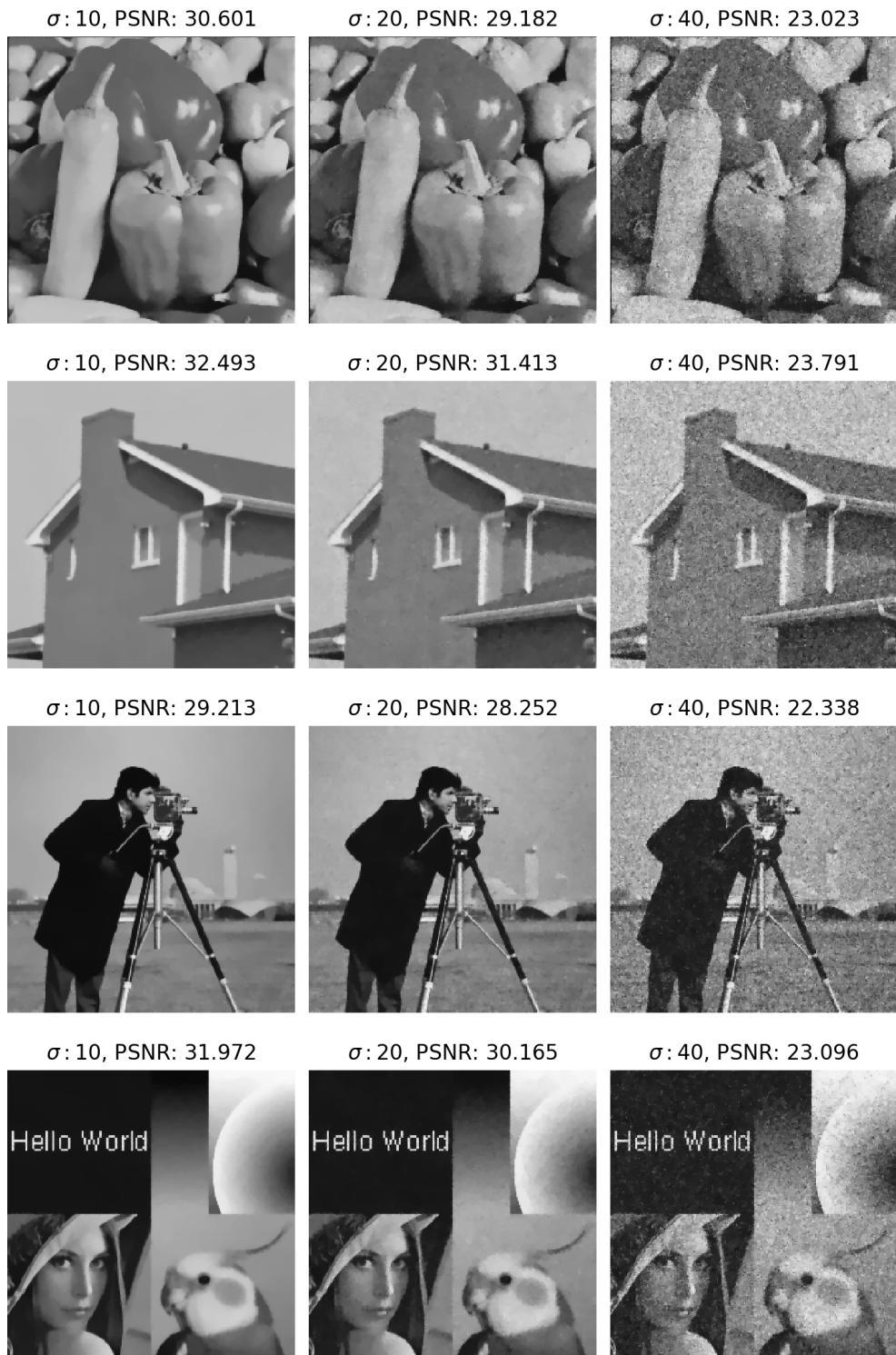


Abbildung 7.8: Subquadratischer Ansatz, $\alpha = 305.32$, $\lambda = 0.049$.



Abbildung 7.9: Adaptiver subquadratischer Ansatz mit Gauß-Filter in der Gewichtungsfunktion, $\alpha = 6671.39$, $\beta = 106.86$, $\lambda = 0.054$.



Abbildung 7.10: Adaptiver subquadratischer Ansatz mit Mittelung der Gewichtungsfunktion, $\alpha = 17670$, $\beta = 129.21$, $\lambda = 0.028$.

7 Evaluation

Um das Verhalten der verschiedenen Ansätze noch besser zu sehen, zeigt Abbildung 7.11 die PSNR-Werte der gleichen vier Bilder aus Abbildung 7.6 über variierende Rauschpegel. Die Parameter der einzelnen Verfahren wurden aus den vorherigen Abbildungen übernommen.

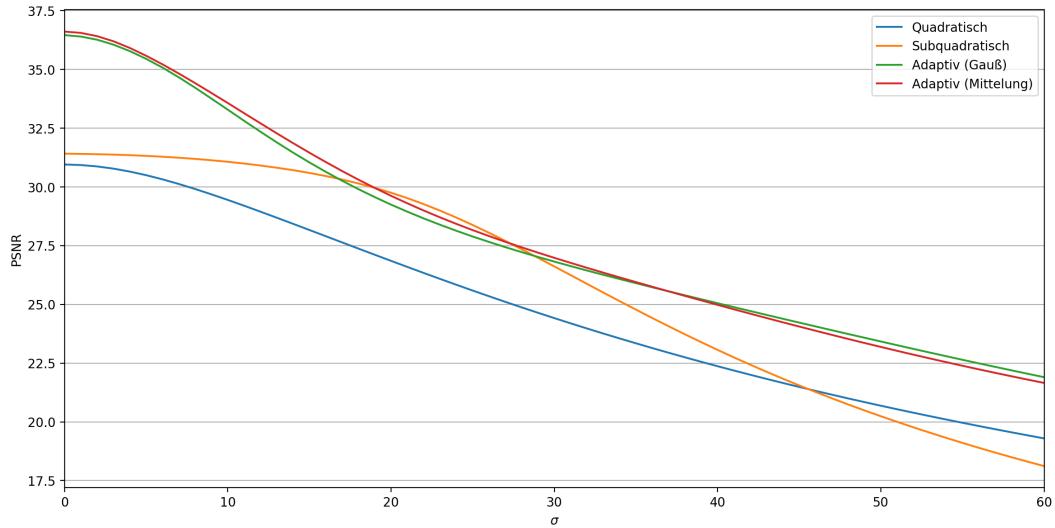


Abbildung 7.11: PSNR-Werte der verschiedenen Methoden für die Bilder aus Abbildung 7.6 mit unterschiedlichen Rauschpegeln.

8 Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war es, die Eignung der lokalen adaptiven Gewichtung zwischen Daten- und Glattheitsterm für die Bildentrauschung zu untersuchen. Es wurde zunächst ein quadratischer Variationsansatz aufgestellt und mit Hilfe des Jacobi-Verfahrens gelöst. Danach wurden zur schnelleren Konvergenz das Gauß-Seidel-Verfahren, sowie das SOR-Verfahren implementiert. Um die Kantenerhaltung zu verbessern wurde dann mit Hilfe der Charbonnier-Funktion der Datenterm zu einem subquadratischen Ansatz erweitert und das entstehende Gleichungssystem mit Hilfe des Lagged-Nonlinearity-Verfahrens gelöst. Die adaptive Gewichtung wurde dann sowohl für den quadratischen als auch den subquadratischen Variationsansatz untersucht, wobei sich bei dem quadratischen Ansatz schnell herausstellte, dass dieser für die Bildentrauschung nur wenig Nutzen hat. Der adaptive subquadratische Ansatz hingegen zeigte eine deutlich verbesserte Robustheit gegenüber variierenden Rauschpegeln.

Ausblick

Die in dieser Arbeit implementierte Gewichtung zwischen Daten- und Glattheitsterm zeigt das Potential eines adaptiven Variationsansatzes in der Bildentrauschung. Eine Untersuchung verschiedener Gewichtungsfunktionen könnte noch weitere Verbesserungen bringen. Die Glättung des Datenterms innerhalb der Gewichtungsfunktion könnte noch weiter untersucht werden. Zusätzlich könnte man auch die Exponentialfunktion mit anderen Funktionen vergleichen. Auch ganz andere Gewichtungsfunktionen, die nicht auf dem Datenterm beruhen, könnten interessante Ergebnisse liefern.

Literaturverzeichnis

- [Alb15] S. Albrecht. „Variationsansätze zur Struktur-Textur-Zerlegung“. B.S. Thesis. Universität Stuttgart, 2015 (zitiert auf S. 8).
- [BCM05a] A. Buades, B. Coll, J.-M. Morel. „A non-local algorithm for image denoising“. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Bd. 2. 2005, S. 60–65 (zitiert auf S. 8).
- [BCM05b] A. Buades, B. Coll, J.-M. Morel. „A review of image denoising algorithms, with a new one“. In: *Multiscale Modeling & Simulation* 4.2 (2005), S. 490–530 (zitiert auf S. 8).
- [CBAB97] P. Charbonnier, L. Blanc-Féraud, G. Aubert, M. Barlaud. „Deterministic edge-preserving regularization in computed imaging“. In: *IEEE Transactions on Image Processing* 6.2 (1997), S. 298–311 (zitiert auf S. 21).
- [Cha04] T. Chan. „Recent developments in total variation image restoration“. In: *Handbook of Mathematical Models in Computer Vision* (2004), S. 1–18 (zitiert auf S. 8).
- [CM99] T. F. Chan, P. Mulet. „On the convergence of the lagged diffusivity fixed point method in total variation image restoration“. In: *SIAM Journal on Numerical Analysis* 36.2 (1999), S. 354–367 (zitiert auf S. 23).
- [CMSH+10] Q. Chen, P. Montesinos, Q. S. Sun, P. A. Heng et al. „Adaptive total variation denoising based on difference curvature“. In: *Image and Vision Computing* 28.3 (2010), S. 298–306 (zitiert auf S. 8, 31).
- [CYV00] S. G. Chang, B. Yu, M. Vetterli. „Adaptive wavelet thresholding for image denoising and compression“. In: *IEEE Transactions on Image Processing* 9.9 (2000), S. 1532–1546 (zitiert auf S. 8).
- [DFKE07] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian. „Image denoising by sparse 3-D transform-domain collaborative filtering“. In: *IEEE Transactions on Image Processing* 16.8 (2007), S. 2080–2095 (zitiert auf S. 8).
- [DHR11] Y. Dong, M. Hintermüller, M. M. Rincon-Camacho. „Automated regularization parameter selection in multi-scale total variation models for image restoration“. In: *Journal of Mathematical Imaging and Vision* 40.1 (2011), S. 82–104 (zitiert auf S. 8).
- [EA06] M. Elad, M. Aharon. „Image denoising via sparse and redundant representations over learned dictionaries“. In: *IEEE Transactions on Image processing* 15.12 (2006), S. 3736–3745 (zitiert auf S. 8).

- [GH09] W. Guo, F. Huang. „Adaptive total variation based filtering for MRI images with spatially inhomogeneous noise and artifacts“. In: *Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI): From Nano to Macro*. 2009, S. 101–104 (zitiert auf S. 8).
- [GSZ06] G. Gilboa, N. Sochen, Y. Y. Zeevi. „Variational denoising of partly textured images by spatially varying constraints“. In: *IEEE Transactions on Image Processing* 15.8 (2006), S. 2281–2289 (zitiert auf S. 8, 31).
- [HKDB17] B.-W. Hong, J.-K. Koo, H. Dirks, M. Burger. „Adaptive Regularization in Convex Composite Optimization for Variational Imaging Problems“. In: *Proceedings of German Conference on Pattern Recognition (GCPR)*. 2017, S. 268–280 (zitiert auf S. 3, 8, 27, 31).
- [Mei11] A. Meister. *Numerik linearer Gleichungssysteme*. Bd. 5. Springer, 2011 (zitiert auf S. 15, 16, 18–20).
- [ROF92] L.I. Rudin, S. Osher, E. Fatemi. „Nonlinear total variation based noise removal algorithms“. In: *Physica D: Nonlinear Phenomena* 60.1-4 (1992), S. 259–268 (zitiert auf S. 8).
- [WBSS04] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli. „Image quality assessment: from error visibility to structural similarity“. In: *IEEE Transactions on Image Processing* 13.4 (2004), S. 600–612 (zitiert auf S. 12).

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift