

1 Implementierung

1.1 Theorie: Iterative Bildrekonstruktion

Die **Richardson-Lucy-Deconvolution (RLD)** ist ein iteratives Verfahren zur Rekonstruktion eines scharfen Bildes aus einem unscharfen Bild, wenn die Art der Unschärfe bekannt ist – diese wird durch die sogenannte *Point Spread Function (PSF)* beschrieben.

Der Algorithmus beginnt mit einer anfänglichen Schätzung des Originalbildes und verbessert diese in mehreren Iterationen. Dabei wird jeweils überprüft, wie gut die aktuelle Schätzung das unscharfe Bild erklären kann. Die Schätzung wird dann entsprechend angepasst.

Die zentrale Gleichung lautet:

$$\hat{u}_j^{(t+1)} = \hat{u}_j^{(t)} \sum_i \frac{d_i}{c_i} p_{ij}$$

Dabei bedeuten:

- $\hat{u}_j^{(t)}$: aktuelle Schätzung des scharfen Bildes im t -ten Iterationsschritt
- d_i : gemessene Pixelwerte des unscharfen Bildes
- p_{ij} : PSF-Wert, der beschreibt, wie stark das Pixel j zum unscharfen Pixel i beiträgt
- c_i : Zwischenwert, der angibt, wie gut das aktuelle geschätzte Bild das unscharfe Bild wiedergibt:

$$c_i = \sum_j p_{ij} \hat{u}_j^{(t)}$$

In jeder Iteration wird die Schätzung \hat{u}_j so angepasst, dass sie besser zu den beobachteten Daten d_i passt. Wenn der Algorithmus richtig konvergiert, ergibt sich am Ende ein rekonstruiertes Bild, das unter den gegebenen Bedingungen (Unschärfe und Rauschen) am wahrscheinlichsten das ursprüngliche, scharfe Bild darstellt.

1.2 Umsetzung der Hauptschleife in Python

Im Code wird die iterative RLD-Methode wie folgt implementiert:

```
1  # Iterative RLD algorithm
2  for i in range(iterations):
3      # Apply the PSF to the current estimate to simulate the blurred version of the
      # → estimate.
4      estimate_conv = cv.filter2D(estimate, -1, psf, borderType=cv.BORDER_REFLECT)
5
6      # Calculate the ratio of the blurred image to the current estimate's blurred version.
7      ratio = image_blurred / (estimate_conv + 1e-7) # Avoid division by zero
8
9      # Convolve the ratio with the flipped PSF to compute the correction factor.
10     correction = cv.filter2D(ratio, -1, psf_mirror, borderType=cv.BORDER_REFLECT)
11
12     # Update the estimate by multiplying it with the correction.
13     estimate *= correction
```

1.2.1 Erklärung der Code-Schritte

1. **Anwenden der PSF:** Zuerst wird die aktuelle Schätzung des Bildes mit der PSF kombiniert, um eine unscharfe Version der Schätzung zu erzeugen (Zeile 4). Dies entspricht dem ersten Schritt der Iteration, bei dem die Schätzung durch die PSF "unscharf gemacht" wird.
2. **Berechnung des Verhältnisses:** Es wird nun das Verhältnis zwischen dem unscharfen Bild und der erzeugten unscharfen Version der Schätzung berechnet (Zeile 7). Dieses Verhältnis zeigt, wie stark die Schätzung von der tatsächlichen Unschärfe abweicht.
3. **Berechnung des Korrekturfaktors:** Der berechnete Unterschied wird dann mit der gespiegelten PSF kombiniert, um einen Korrekturfaktor zu erhalten (Zeile 10). Dieser Schritt hilft, die Schätzung in die richtige Richtung zu korrigieren.
4. **Aktualisierung der Schätzung:** Die Schätzung des Bildes wird schließlich mit diesem Korrekturfaktor multipliziert, um die Schätzung zu verbessern (Zeile 13). Dies ist der eigentliche Iterationsschritt, bei dem die Schätzung Schritt für Schritt näher an die tatsächliche Lösung gebracht wird.

Diese Schritte wiederholen sich, wobei die Schätzung nach jeder Iteration immer genauer wird, was schließlich zu einer besseren Rekonstruktion des Bildes führt.

1.3 Implementierte Filterkernels

Zur Erzeugung der unscharfen Bilder wurden vier verschiedene Filterkernel implementiert, um unterschiedliche Arten von Unschärfe zu simulieren:

- **Mittelwertfilter:** Ein einfacher 5×5 -Kernel, bei dem alle Werte gleich sind. Dies führt zu einer gleichmäßigen Unschärfe über das Bild.
- **Gauß-Filter:** Ein 9×9 -Kernel mit einer Standardabweichung von 2. Die Werte basieren auf einer Gaußverteilung und wurden mit `cv.getGaussianKernel()` aus OpenCV generiert.
- **Horizontaler Bewegungsfiler:** Ein 1×9 -Kernel mit gleichmäßigen Werten in horizontaler Richtung. Dieser simuliert eine Bewegungsunschärfe von links nach rechts.
- **Diagonaler Bewegungsfiler:** Ein 9×9 -Kernel, der nur entlang der Hauptdiagonale Werte besitzt. Dies simuliert eine diagonale Bewegungsunschärfe.

2 Test mit verschiedenen Startbildern und Filterkernels

In diesem Kapitel werden die Ergebnisse der Richardson-Lucy-Dekonvolution mit 100 Iterationen unter verschiedenen Bedingungen dargestellt. Ziel ist es, die Auswirkungen der gewählten *Point Spread Function*, *PSF* sowie des verwendeten Startbildes (Initialisierung) auf die Qualität der restaurierten Bilder zu untersuchen.

- **Random Guess Estimation:** Ein zufällig erzeugtes Bild. Das Ergebnis zeigt starke Artefakte und Konvergenzprobleme.
- **Gray Guess Estimation:** Ein konstant graues Bild als Startwert. Das Ergebnis ist bereits relativ gut.
- **Blurred Image Guess Estimation:** Das verschwommene Bild selbst wurde als Initialisierung verwendet, es ist sehr ähnlich zum grauen Startbild.
- **Duck Image Guess Estimation:** Ein Bild von Donald Duck wurde zur Initialisierung verwendet.

2.1 Mittelwertfilter als PSF

2.1.1 Originalbilder

In Abbildung 1 und 2 sind die Ausgangsbilder dargestellt, welche als Basis für diesen Test dienten. Das erste Bild stellt das Originalbild dar. Dieses wurde anschließend mit einem Mittelwertfilter gefaltet, um eine Unschärfe zu erzeugen.



Abbildung 1: Originalbild



Abbildung 2: Mit Mittelwertfilter (PSF) gefiltert

2.1.2 Restaurierte Bilder ohne Rauschen

Die Abbildungen 3 bis 6 zeigen die Ergebnisse der Dekonvolution ohne Rauschen, wobei der Mittelwertfilter als PSF verwendet wurde.

Bewertung: Die Ergebnisse zeigen, dass die Wahl des Startbildes einen Einfluss auf die Qualität der Rekonstruktion hat. Die *Random Guess*-Initialisierung führt zu deutlich sichtbaren Artefakten im rekonstruierten Bild. Dagegen liefern sowohl die *Gray Guess*- als auch die *Blurred Guess*-Variante deutlich bessere Resultate mit wenigen Artefakten. Bei der *Duck Guess*-Initialisierung ist noch ein Teil des Donald-Duck-Bildes in der Rekonstruktion erkennbar, was die Qualität beeinträchtigt.

Die Initialisierung erfolgte mit verschiedenen Startbildern:



Abbildung 3: Random Guess Estimation



Abbildung 4: Gray Guess Estimation



Abbildung 5: Blurred Image Guess Estimation



Abbildung 6: Duck Image Guess Estimation

2.2 Gauß-Filter als PSF

2.2.1 Originalbilder

In Abbildung 7 und 8 sind die Ausgangsbilder dargestellt, welche als Basis für diesen Test dienten. Das erste Bild stellt das Originalbild dar. Dieses wurde anschließend mit einem Gauß-Filter gefaltet, um eine Unschärfe zu erzeugen.



Abbildung 7: Originalbild



Abbildung 8: Mit Gaußfilter (PSF) gefiltert

2.2.2 Restaurierte Bilder ohne Rauschen

Bei Verwendung eines Gauß-Kernels als PSF zeigen sich (Abbildungen 9 bis 12) ähnliche Tendenzen wie beim Mittelwertfilter.

Bewertung: Auch beim Gauß-Filter hängt die Qualität der Rekonstruktion stark von der Initialisierung ab. Die *Random Guess*-Variante zeigt erneut viele Artefakte. *Gray Guess* und *Blurred Guess* liefern brauchbare Ergebnisse, wirken jedoch etwas verschwommener als beim Mittelwertfilter. Bei der *Duck Guess*-Initialisierung bleibt das ursprüngliche Donald-Duck-Motiv deutlich sichtbar, was zu einer fehlerhaften Rekonstruktion führt.



Abbildung 9: Random Guess Estimation



Abbildung 10: Gray Guess Estimation



Abbildung 11: Blurred Image Guess Estimation Abbildung 12: Duck Image Guess Estimation

2.3 Horizontal-Filter als PSF

2.3.1 Originalbilder

Die Originalbilder wurden mithilfe eines horizontalen PSFs gefiltert. Dabei wird vor allem horizontale Information verschmiert, wodurch vertikale Kanten stärker betroffen sind.



Abbildung 13: Originalbild



Abbildung 14: Mit horizontalem Filter (PSF) gefiltert

2.3.2 Restaurierte Bilder ohne Rauschen

Bewertung: Die Ergebnisse zeigen, dass der horizontale PSF erwartungsgemäß zu vertikalen Artefakten in der Rekonstruktion führt. Die *Random Guess*-Initialisierung resultiert erneut in vielen Störungen, wobei ein vertikales Muster erkennbar ist. *Gray Guess* und *Blurred Guess* liefern vergleichbare, aber durch vertikale Streifen beeinträchtigte Resultate. Die *Duck Guess*-Initialisierung führt zu einem besonders unscharfen Ergebnis, bei dem die Struktur des Entenbildes noch erkennbar, aber deutlich verschmiert ist.



Abbildung 15: Random Guess Estimation



Abbildung 16: Gray Guess Estimation



Abbildung 17: Blurred Image Guess Estimation Abbildung 18: Duck Image Guess Estimation

2.4 Diagonal-Filter als PSF

2.4.1 Originalbilder

Durch Anwendung eines diagonalen PSFs wird Bildinformation entlang einer Diagonale verschmiert, was zu einer Richtungsunschärfe führt.



Abbildung 19: Originalbild

Abbildung 20: Mit Mittelwertfilter (PSF) gefiltert

2.4.2 Restaurierte Bilder ohne Rauschen

Bewertung: Die Ergebnisse ähneln jenen des horizontalen PSFs, jedoch treten die Artefakte entsprechend der diagonalen Verschmierungsrichtung auf. Bei *Random Guess* zeigen sich deutliche diagonale Muster und Artefakte. *Gray Guess* und *Blurred Guess* liefern insgesamt stabile, aber leicht diagonale Streifen aufweisende Resultate. Die *Duck Guess*-Initialisierung führt erneut zu einem stark verschmierten Bild, bei dem die ursprüngliche Struktur nur noch schwach zu erkennen ist.



Abbildung 21: Random Guess Estimation



Abbildung 22: Gray Guess Estimation



Abbildung 23: Blurred Image Guess Estimation Abbildung 24: Duck Image Guess Estimation

3 Test mit unterschiedlich vielen Iterationen

In diesem Test wurde untersucht, welchen Einfluss die Anzahl der Iterationen auf das Rekonstruktionsergebnis hat. Als Startbild diente jeweils das verschwommene Bild, und es wurden verschiedene *Point Spread Functions* (PSFs) verwendet. Die maximale Iterationsanzahl wurde dabei auf 10, 50 und 100 festgelegt.

3.1 Iterationstests: Mittelwertfilter als PSF

Bewertung: Mit zunehmender Iterationsanzahl verbessert sich die Bildschärfe deutlich. Bereits bei 50 Iterationen sind feine Details besser sichtbar. Allerdings steigt mit der Iterationszahl auch das Risiko für Artefaktbildung.



Abbildung 25: 10 Iterationen Abbildung 26: 50 Iterationen Abbildung 27: 100 Iterationen

3.2 Iterationstests: Gauß-Filter als PSF

Bewertung: Auch beim Gauß-Filter führt eine höhere Iterationsanzahl zu einer leichten Verbesserung der Bildschärfe. Im Vergleich zum Mittelwertfilter fällt die Verbesserung jedoch weniger deutlich aus. Die Rekonstruktion bleibt insgesamt etwas weicher.



Abbildung 28: 10 Iterationen Abbildung 29: 50 Iterationen Abbildung 30: 100 Iterationen

3.3 Iterationstests: Horizontal-Filter als PSF

Bewertung: Eine höhere Iterationsanzahl führt zu einer deutlich verbesserten Schärfe. Gleichzeitig treten jedoch verstärkt vertikale Artefakte auf, was auf die Richtungsabhängigkeit des horizontalen PSFs zurückzuführen ist.



Abbildung 31: 10 Iterationen Abbildung 32: 50 Iterationen Abbildung 33: 100 Iterationen

3.4 Iterationstests: Diagonal-Filter als PSF

Bewertung: Mit steigender Iterationsanzahl verbessert sich die Bildschärfe deutlich. Gleichzeitig werden jedoch auch diagonale Artefakte immer sichtbarer, was der diagonalen Struktur des verwendeten PSFs entspricht.



Abbildung 34: 10 Iterationen Abbildung 35: 50 Iterationen Abbildung 36: 100 Iterationen

4 Test mit zusätzlichem Rauschen

In diesem Test wurde untersucht, wie sich zusätzliches Rauschen auf die Qualität der Bildrekonstruktion auswirkt. Ausgangspunkt waren Bilder, die mit verschiedenen Unschärfefiltern (Mittelwert-, Gauß-, horizontaler und diagonalen Filter) verschwommen wurden. Anschließend wurde diesen Bildern künstlich Rauschen mit unterschiedlicher Intensität (Standardabweichung 2, 5 und 10) hinzugefügt. Die Ergebnisse der Rekonstruktion wurden anschließend bewertet und verglichen.

4.1 Mittelwertfilter als PSF

4.1.1 Bilder mit Rauschen



Abbildung 37: Verrauschtes Bild mit Mittelwertfilter (Noise $\sigma = 2$)
 Abbildung 38: Verrauschtes Bild mit Mittelwertfilter (Noise $\sigma = 5$)
 Abbildung 39: Verrauschtes Bild mit Mittelwertfilter (Noise $\sigma = 10$)

4.1.2 Rekonstruierte Bilder

Bewertung: Bei geringer Rauschintensität (Standardabweichung 2) gelingt die Rekonstruktion noch relativ gut. Ab einer Standardabweichung von 5 wird das Rauschen deutlich sichtbar, obwohl die Bildschärfe noch verbessert wird. Bei hoher Rauschintensität (Standardabweichung 10) dominiert das Rauschen das Ergebnis, wodurch die Rekonstruktion insgesamt ziemlich unbrauchbar wird.



Abbildung 40: Rekonstruiertes Bild bei Noise $\sigma = 2$
 Abbildung 41: Rekonstruiertes Bild bei Noise $\sigma = 5$
 Abbildung 42: Rekonstruiertes Bild bei Noise $\sigma = 10$

4.2 Gauß-Filter als PSF

4.2.1 Bilder mit Rauschen



Abbildung 43: Verrauschtes Bild mit Gaußfilter (Noise $\sigma = 2$) Abbildung 44: Verrauschtes Bild mit Gaußfilter (Noise $\sigma = 5$) Abbildung 45: Verrauschtes Bild mit Gaußfilter (Noise $\sigma = 10$)

4.2.2 Rekonstruierte Bilder

Bewertung: Ähnlich wie beim Mittelwertfilter zeigt sich bei geringer Rauschintensität (Standardabweichung 2) eine noch gute Rekonstruktion. Bei einer Standardabweichung von 5 treten ebenfalls deutlich sichtbare Rauschartefakte auf, jedoch sind diese etwas größer als beim Mittelwertfilter. Bei einer Standardabweichung von 10 wird das Rauschen noch dominanter.



Abbildung 46: Rekonstruiertes Bild bei Noise $\sigma = 2$ Abbildung 47: Rekonstruiertes Bild bei Noise $\sigma = 5$ Abbildung 48: Rekonstruiertes Bild bei Noise $\sigma = 10$

4.3 Horizontal-Filter als PSF

4.3.1 Bilder mit Rauschen



Abbildung 49: Verrauschtes Bild mit Horizontalfilter (Noise $\sigma = 2$) Abbildung 50: Verrauschtes Bild mit Horizontalfilter (Noise $\sigma = 5$) Abbildung 51: Verrauschtes Bild mit Horizontalfilter (Noise $\sigma = 10$)

4.3.2 Rekonstruierte Bilder

Bewertung: Bei einer Standardabweichung von 2 zeigt sich noch eine gute Rekonstruktion. Mit einer Standardabweichung von 5 bleibt das Ergebnis akzeptabel, obwohl Rauschartefakte zunehmen. Bei einer Standardabweichung von 10 überwiegen die Artefakte, ähnlich wie beim Mittelwertfilter, wobei vor allem vertikale Störungen sichtbar werden.



Abbildung 52: Rekonstruiertes Bild bei Noise $\sigma = 2$ Abbildung 53: Rekonstruiertes Bild bei Noise $\sigma = 5$ Abbildung 54: Rekonstruiertes Bild bei Noise $\sigma = 10$

4.4 Diagonal-Filter als PSF

4.4.1 Bilder mit Rauschen



Abbildung 55: Verrauschtes Bild mit Diagonalfilter (Noise $\sigma = 2$) Abbildung 56: Verrauschtes Bild mit Diagonalfilter (Noise $\sigma = 5$) Abbildung 57: Verrauschtes Bild mit Diagonalfilter (Noise $\sigma = 10$)

4.4.2 Rekonstruierte Bilder

Bewertung: Bei einer Standardabweichung von 2 ist die Rekonstruktion noch gut. Mit einer Standardabweichung von 5 bleibt das Ergebnis akzeptabel, besonders im Vergleich zum Ausgangsbild. Bei einer Standardabweichung von 10 überwiegen die Rauschartefakte und die Qualität der Rekonstruktion nimmt deutlich ab.



Abbildung 58: Rekonstruiertes Bild bei Noise $\sigma = 2$ Abbildung 59: Rekonstruiertes Bild bei Noise $\sigma = 5$ Abbildung 60: Rekonstruiertes Bild bei Noise $\sigma = 10$

5 Tests mit anderem Bild

Um die Robustheit des Algorithmus zu prüfen, wurde die Rekonstruktion auch mit einem anderen Bild getestet. Dabei kamen verschiedene Unschärfefilter (Mittelwert, Gauß, horizontal, diagonal) zum Einsatz. In den Abbildungen ist jeweils links das verschwommene und rechts das rekonstruierte Bild zu sehen. Alle Filter liefern insgesamt gute Ergebnisse, jedoch zeigt sich, dass der Gauß-Filter im Vergleich die schlechteste Rekonstruktionsqualität aufweist.



Abbildung 61: Mittelwertfilter



Abbildung 62: Mittelwertfilter: Rekonstruiertes Bild

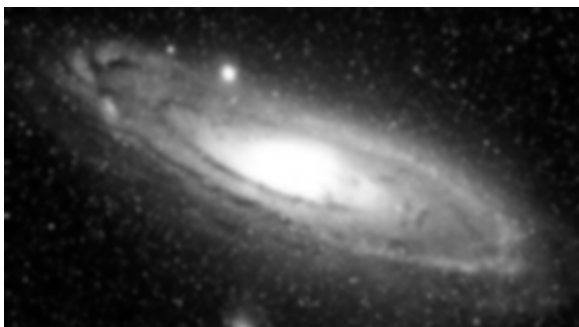


Abbildung 63: Gauß-Filter



Abbildung 64: Gauß-Filter: Rekonstruiertes Bild



Abbildung 65: Horizontalfilter



Abbildung 66: Horizontalfilter: Rekonstruiertes Bild

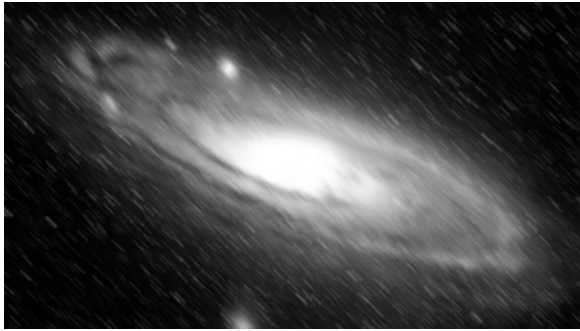


Abbildung 67: Diagonalfilter



Abbildung 68: Diagonalfilter: Rekonstruiertes Bild