

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 111

Fokus Fusion mit Variationsansätzen

Vladislav Ladyzhenskiy

Studiengang:	Informatik
Prüfer/in:	Prof. Dr.-Ing. Andrés Bruhn
Betreuer/in:	Prof. Dr.-Ing. Andrés Bruhn
Beginn am:	2014-02-03
Beendet am:	2014-08-05
CR-Nummer:	G.1.1 , G.1.2, G.1.6, G.1.8, I.4.3, I.4.6, I.4.8,

Kurzfassung

... Das Verfahren zur Bildfusion hat eine vielfältige Nutzungen in der digitalen Bildverarbeitung. In vielen Industriebereichen, aber auch Medizin und Weltraumforschung müssen mehreren Bilder von ein und derselben Szene mit verschiedenen Brennweiten aufgenommen werden. Die im Fokus liegende Bereiche sollen anschließend zu einem Bild verschmolzen werden, um so eine bessere Qualität zu erhalten. Ziel dieser Arbeit liegt also in der Modellierung eines Algorithmus für die Bildfusion. Als Grundlage werden die Verfahren von Boshtayeva *et al.* verwendet. Dafür sollen Kriterien für den Schärfegrad eines Pixels unter der Verwendung von Ableitung-basierten Funktionen formuliert werden und zwei Konfidenzmaße für die Gewichtung der einzelnen Pixeln vorgestellt werden. Der Variationsansatz wird dabei unter Ausnutzung verschiedener Bestrafungsfunktionen implementiert. Die eigentliche Bildfusion wird Pixel- als auch Gradienten-basierend implementiert. Anschließend werden die Ergebnisse in Abhängigkeit von Eingabeparametern evaluiert und optisch verglichen.

Inhaltsverzeichnis

1	Einleitung	7
2	Related Work	11
3	Grundlagen	13
3.1	Gradient	13
3.2	Hesse - Matrix	15
4	Algorithmus	17
4.1	Das Erstellen der Anfangstiefenkarte	18
4.2	Auswahl der Werte der initialen Tiefenkarte	21
4.3	Rekonstruktion der endgültigen Tiefenkarte	22
4.3.1	Allgemeine Minimierung des Energiefunktionals	24
4.3.2	Minimierung des Energiefunktionals	27
4.3.3	Diskretisierung für quadratischen Daten- und Glattheitsterm	28
4.4	Fusion	33
5	Experimentelle Ergebnisse	39
5.1	Vergleich und Evaluierung der Schärfekriterien	39
5.2	Vergleich und Evaluierung der Entscheidungsfunktionen	41
5.3	Evaluierung der Wirkung von sub-quadratischen Bestrafungstermen	41
5.4	Vergleich und Evaluierung der Fusionsverfahren	43
6	Zusammenfassung und Ausblick	51
	Literaturverzeichnis	53

1 Einleitung

Der Begriff der Fusion stammt aus dem Lateinischen und bedeutet Verschmelzung. Normalerweise assoziiert man ihn mit der Atomenergie und der Kernfusion. Dabei beschreibt er einfach den Prozess der Verbindung mehrerer Quellen zu einem neuen Objekt. Spricht man von der Fusion im Bezug auf die digitale Bildverarbeitung, so meint man damit das Verschmelzen von gewissen Features aus mehreren Quellen. Eine Feature ist eine Eigenschaft vom Interesse, wie der Wert des Gradienten, der Grauwert oder die Form. Ganze Objekte können ebenfalls unter dem Begriff „Feature“ zusammengefasst werden.

Verschmilzt man z.B. Objekte aus verschiedenen Bildern, entstehen Szenen, die vielleicht so in der realen Welt nicht vorkommen können (siehe Abbildung 1.1). Zum Beispiel kann man Features aus den Bildern extrahieren und zu einem neuen Bild zusammenfügen. Die Technik wird in zahlreichen Bildverarbeitungsanwendungen genutzt, um Fotomontagen anzufertigen. In der Erforschung der Erdoberfläche aus dem Weltall beeinträchtigt oft die Wolkenbildung die Aufnahmen. Verschmilzt man aber z.B. die Aufnahme mit einer Abbildung der Tiefe, können Wolken aus dem Bild entfernt werden [16].

Ein anderes Gebiet, in dem Fusion oft eingesetzt wird, ist die Makrofotografie. Man will dabei ein kleines Objekt sehr nah aufnehmen, um dadurch alle Details zu erfassen. Das Problem



Abbildung 1.1: Bildermontage. *Linke Spalte:* Eingabebilder. *Mittlere Spalte:* Einfaches Kombinieren der Bildbereiche. *Rechte Spalte:* Verfahren von Perez *et.al.*. Quelle: [16]

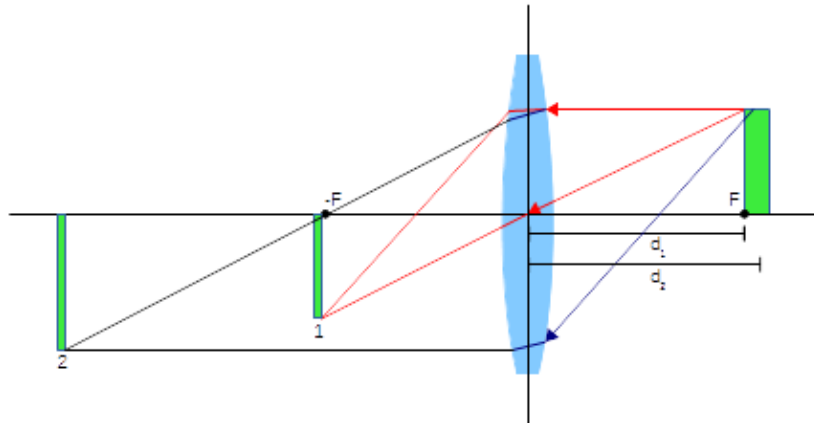


Abbildung 1.2: Beispiel für die Aufnahme eines Volumens mit einer Linse

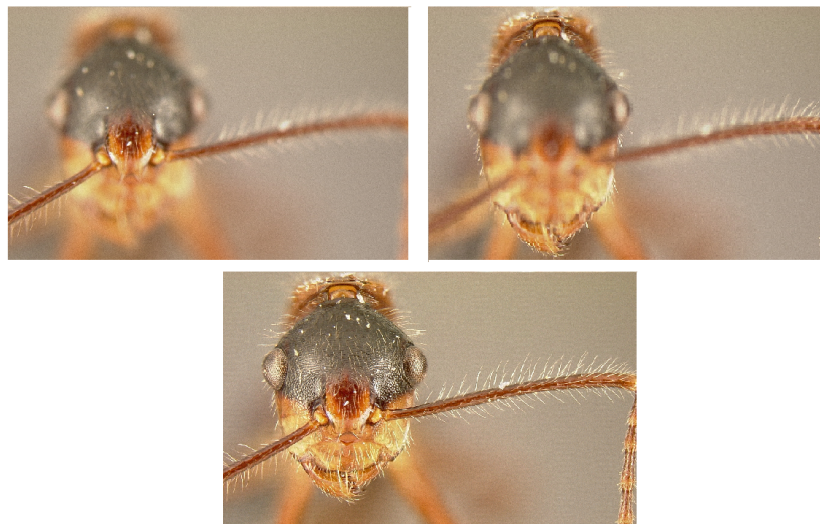


Abbildung 1.3: Beispiel für die Fusion in der Makrofotografie. *Oben:* Eingabebilder, aufgenommen mit unterschiedlichen Brennweiten. *Unten:* Das fusionierte Bild. Quelle: [2]

dabei liegt in der Tatsache, dass Objekte in der realen Welt selten flach sind. Fokussiert man das Objektiv der Kamera auf einen Teil des Objekts, verschwimmen dafür die anderen Bereiche. Die Abbildung 1.2 verdeutlicht das: Die linke Kante des grünen Rechtecks rechts von der Linse befindet sich im Fokus der Kamera und wird deswegen eins zu eins abgebildet (Rechteck 1 in der Abbildung 1.2). Weil das Objekt eine gewisse Dicke besitzt, muss das ebenfalls berücksichtigt werden. Nach den Regeln der Optik, falls die Entfernung des Objekts zu der Kamera im Bereich $F < d_2 < 2F$ liegt [6], vergrößern sich die Abmessungen seiner Abbildung (siehe Rechteck 2). Das bedeutet, die einzelnen Punkte der Abbildung haben eine viel größere Distanz zueinander und der entsprechende Bereich wird unscharf dargestellt. Man verliert damit Informationen.

Betrachtet man die Abbildung 1.3, stellt man fest, dass vor allem feine Strukturen wegen "falscher" Fokussierung des Bildes verloren gehen können. Das ist am besten an den Fühlern und größtenteils an den Härchen der Ameise deutlich zu sehen. Fusioniert man beide Bilder entsteht ein Bild, das überall scharf ist.

Die Fusion in der Abbildung 1.3 entstand mithilfe der sogenannten Methode der Variationsansätze. Dafür wird eine Schichtenaufnahme eines dreidimensionalen Objekts angefertigt, wobei jede Schicht eine Brennweite darstellt. Anschließend wird eine Tiefenkarte erstellt, die nur Tiefe des schärfsten Pixel auf der jeweiligen Position enthält. Damit wird erreicht, dass das Ergebnis überall scharf ist. Weil die Tiefenwerte der benachbarten Pixel ähnlich sein müssen, verwendet man oft einen globalen Optimierungsansatz, der geeignete Glattheitsforderungen modelliert. Dadurch erhält man ein glattes Bild, das jedoch gleichzeitig Kanten respektiert. Weil diese Methode der Bilderfusion oft in den Digitalkameras eingesetzt wird, werden wir in dieser Arbeit deswegen näher darauf eingehen.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 3 – Grundlagen: Der Kapitel befasst sich mit den mathematischen Grundlagen, die in dieser Arbeit verwendet werden.

Kapitel 4 – Algorithmus: Hier wird schrittweise dieser Arbeit zugrundeliegende Algorithmus erläutert.

Kapitel 5 – Experimentelle Ergebnisse: In diesem Teil der Arbeit werden die experimentelle Ergebnisse unter verschiedenen Bedingungen aufgezeigt .

Kapitel 6 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

2 Related Work

Auf dem Gebiet der Bildfusion wurden bereits mehrere Arbeiten verfasst. Hier muss man die Art der Fusion unterscheiden. Auf einer Seite hat man die allgemeine Fusion, d.h. aus mehreren Quellen werden bestimmte Informationen übernommen. Dazu zählt z.B. die Arbeit von Perez *et.al* [16]. Hier wird anhand eines Leitfeldes, das die Farbinformation approximiert, ein fusioniertes Bild erzeugt. Ziel ihrer Arbeit liegt in der nahtlosen Einfügen von ganzen Bildbereichen. Genauso konzentrieren sich Agarwala *et.al* [1] auf das Gebiet der Fotomontage. Sie verwenden ebenfalls das Gradient-Domain-Verfahren [16], nur erzeugen sie zusätzlich einen Graphen, dessen Knoten die Pixel aus dem jeweiligen Quellbild sind. Die optimale Fusion ist die Minimierung der Kosten des Rundgangs durch alle Knoten. Auf diese Weise entsteht Fusion, die Farbinformationen nahtlos einfügt.

Die zweite Variante der Bildfusion beschäftigt sich mit der Fusion von Bildern mit verschiedenen Brennweiten. Hier unterscheidet man zwischen Pixel-basierten und Region-basierten Verfahren. Für beide Varianten müssen zuerst Kriterien für die Schärfe eines Pixels im Quellbild aufgestellt werden. In ihrer Arbeit vergleichen Huang *et.al* [8] verschiedene Schärfekriterien in der räumlichen Domäne.

Auf dem Gebiet der Region-basierten Fusion in der räumlichen Domäne haben Shutao Li und Yang Bin [10] ein Verfahren aufgestellt, das eine Mittelwertbestimmung als Schärfekriterium verwendet. Maruthi *et.al* [13] nutzen Ortsfrequenz und die Sichtbarkeit als Schärfekriterium, indem sie den Betrag der Ortsfrequenz ausrechnen. Kong *et.al* [9] nutzen ebenfalls Betrag der Ortsfrequenz als Schärfekriterium. Die Blöcke mit dem höchsten Betrag werden zu einem neuen Bild fusioniert. Long *et.al* [11] stellen fest, dass eine dünn besiedelte Repräsentation der einzelnen Segmente zu einem robustem Ergebnis führt. Tian *et.al* [19] stellen dagegen eine Fusion in der Wavelet-Domäne vor. Die Verteilung von Koeffizienten der Wavelet-Transformation wird dabei als Kriterium für die Unschärfe des Bildes aufgefasst.

Im Vergleich zum Region-basierten Verfahren, wird bei der Pixel-basierten Vorgehensweise jeder Pixel einzeln betrachtet und mithilfe eines Schärfekriteriums evaluiert. Die Fusion erfolgt ebenfalls Pixel-weise. Rudnaya *et.al* [17] führen den Gradienten als ein mögliches Schärfekriterium ein. Das tun ebenfalls Boshtayeva *et.al* [4],[3]. Sie erweitern das Kriterium auf den Begriff der allgemeinen Ableitung und erstellen eine Tiefenkarte, die sie anschließend mit einem Energiefunktional optimieren. Saeedi *et.al* [18], verwenden dagegen die diskrete Wavelet-Transformation. Sie stellen eine Klassifizierung der hochfrequenten Wavelet-Koeffizienten als Schärfekriterium vor und verwenden als Eingabe eine Bildpyramide.

3 Grundlagen

Das Kapitel soll die grundlegenden Begriffe und mathematischen Abhängigkeiten erläutern, die in weiten Teilen dieser Ausarbeitung eingesetzt werden. Es werden insbesondere die Ableitungen von Funktionen mehrerer Veränderlichen diskutiert: der Gradient und die Hesse-Matrix. Sie legen die Beziehungen zwischen den benachbarten Punkten im Definitionsbereich einer kontinuierlichen Funktion fest und liefern die Richtungsänderung von Funktionswerten, teilweise auch die Geschwindigkeit dieser Änderung zurück. Anschließend werden finite Differenzen zur Diskretisierung vorgestellt und näher auf ihre Implementierung in Form von Filter für das gegebene Bild eingegangen. Ableitungen sind insbesondere deswegen für Fokus-Fusion relevant, weil sie sowohl die Bestimmung von scharfe Bildregionen ermöglichen als auch die Modellierung von Glattheitsverfahren.

3.1 Gradient

Ein Gradient beschreibt die partielle Ableitung einer mehrwertigen Funktion. Sei $f(x_1, x_2, \dots, x_n)$ eine solchen Funktion, dann ist der Gradient definiert als:

$$\nabla f(x_1, x_2, \dots, x_n) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T \quad (3.1)$$

Weil Bilder im einem zweidimensionalen Raum liegen, verkürzt sich die allgemeine Formulierung des Gradienten auf:

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^T \quad (3.2)$$

Im Bezug auf die Bildverarbeitung beschreibt der Gradient die Änderung der Grauwerte. D.h. aus dem obigen Vektor lässt sich die Richtung, die für die Änderungsrichtung steht, bestimmen. Zusätzlich kann man den Betrag, oder den Änderungsfaktor, als euklidische Norm berechnen. Sei $f(x, y)$ eine kontinuierliche und überall differenzierbare Funktion. Die Ableitung beliebiger Ordnung nach einem Parameter lässt sich folgendermaßen herleiten

3 Grundlagen

[6]. Man geht von dem Taylor - Polynom aus und berücksichtigt alle Glieder bis hin zu der gewünschten Ordnung. Der Rest wird als Fehlerglied aufgefasst. Somit erhalten wir:

$$f(x+h, y) = f(x, y) + \frac{f'(x, y)h}{1!} + \frac{f''(x, y)h^2}{2!} + \dots + \frac{f^{(n)}(x, y)h^n}{n!} + R_n(x, y) \quad (3.3)$$

Die Approximation der ersten Ableitung um den Pixel u_i lässt sich dann auf folgende Art approximieren. Man berechnet das Taylor - Polynom für die Pixel u_{i-1} , u_i und u_{i+1} .

$$u_{i-1} = u_i - u'_i h + \frac{u''_i h^2}{2!} + O(h^3) \quad (3.4)$$

$$u_i = u_i \quad (3.5)$$

$$u_{i+1} = u_i + u'_i h + \frac{u''_i h^2}{2!} + O(h^3) \quad (3.6)$$

Wollen wir die erste Ableitung approximieren, liefert der Vergleich der Koeffizienten:

$$\begin{aligned} 0 * u_i + 1 * u'_i + 0 * u''_i &= \alpha_{-1} \cdot u_{i-1} + \alpha_0 \cdot u_i + \alpha_1 \cdot u_{i+1} \\ &= (\alpha_{-1} + \alpha_0 + \alpha_1) \cdot u_i \\ &\quad + h(\alpha_{-1} - \alpha_1) \cdot u'_i \\ &\quad + \frac{h^2}{2}(\alpha_{-1} + \alpha_1) \cdot u''_i + O(h^3) \end{aligned} \quad (3.7)$$

Dies führt zu einem linearen Gleichungssystem:

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{h} \\ 0 \end{pmatrix} \quad (3.8)$$

Die Lösung des obigen Gleichungssystems legt die Koeffizienten der Maske fest:

$$\alpha_{-1} = -\frac{1}{2h} \quad \alpha_0 = 0 \quad \alpha_1 = \frac{1}{2h} \quad (3.9)$$

Berücksichtigt man lediglich die Koeffizienten der einzelnen Einträge, entstehen die Masken, mit denen man das aktuelle Bild falten und so die entsprechenden Gradienten bestimmen kann:

$$\partial_x u \approx \frac{1}{2h} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \quad \partial_y u \approx \frac{1}{2h} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (3.10)$$

Diese Masken (im Englischen stencils genannt) glätten das Bild zusätzlich in die Richtung der Ableitung, nicht aber senkrecht dazu. Um eine gleichmäßige Glättung zu erzielen, faltet man die Stencil 3.10 mit der Maske, die eine Approximation für den Glättungsfilter (Tiefpass) darstellt. Die Gesamtmaske wird auch Sobel-Operator genannt [6]

$$\partial_x u \approx \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{2h} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \frac{1}{8h} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$\partial_y u \approx \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \frac{1}{2h} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{8h} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.12)$$

3.2 Hesse - Matrix

Allgemein beschreibt die Hesse - Matrix die zweite Ableitung einer mehrwertigen Funktion und im Bezug auf die Bildebene hat diese die Form:

$$H(u) = \begin{bmatrix} \partial_{xx}u & \partial_{xy}u \\ \partial_{yx}u & \partial_{yy}u \end{bmatrix} \quad (3.13)$$

Die gemischten partiellen Ableitungen sind gleich, d.h:

$$\partial_{xy}u = \partial_{yx}u \quad (3.14)$$

3 Grundlagen

Zur Approximation der zweiten Ableitung verwenden wir finite Differenzen, die folgende Faltungsmasken bilden:

$$\partial_{xx} \approx \frac{1}{h^2} \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array}, \quad \partial_{yy} \approx \frac{1}{h^2} \begin{array}{|c|} \hline 1 \\ \hline -2 \\ \hline 1 \\ \hline \end{array}, \quad \partial_{xy} \approx \frac{1}{4h^2} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \quad (3.15)$$

4 Algorithmus

Auf dem Gebiet der digitalen Bildverarbeitung werden oft Verfahren benötigt, die ein überall scharfes Bild als Ausgabe liefern. Vor allem in der Makrofotografie müssen kleine dreidimensionale Objekte aufgenommen werden, um ihre Eigenschaften zu erforschen. Weil eine Kamera nicht gleichzeitig die gesamte Szene erfassen kann, bleiben Bereiche außerhalb des Fokus der Kamera verschwommen. Bezogen auf ein 3D-Objekt erstellt man deswegen eine Serie von Bildern, die das Objekt schichtweise darstellen. Weil die Anzahl solcher Aufnahmen recht hoch sein kann, möchte man die scharfen Pixeln aus jedem dieser Bilder übernehmen und in ein neues Bild übertragen.

Da jedes Bild mit verschiedener Brennweite aufgenommen wurde, kann diese Tiefeninformation für die Zuordnung eines Pixels verwendet werden. Man überträgt dabei also nicht die Farbwerte eines Pixels, sondern die Tiefe des Quellbildes. Deswegen kann der Algorithmus für die Bildfusion folgendermaßen zusammengefasst werden:

- *Aufbau des Bildstapels*: Hier müssen die Bilder mit verschiedenen Tiefen nach ihrer Tiefe einsortiert werden. Damit kann die Tiefe als Reihenfolgennummer verwendet werden.
- *Bestimmung des Schärfegrades eines Pixels* : Es werden sogenannte Schärfekriterien formuliert, mit deren Hilfe die Auswahl eines scharfen Pixels erfolgt.
- *Einordnen der ausgewählten Pixel auf einer Tiefenkarte*: Man übernimmt nicht die Farbwerte der ausgewählten Pixel, sondern die Positionsnummer des Bildes im Bildstapel, dem das Pixel gehört.
- *Aufstellen einer Konfidenzfunktion*: Nicht alle Tiefen werden korrekt ausgewählt. Deswegen muss entschieden werden, wie man die restlichen Pixel behandelt.
- *Erzeugung der endgültigen Tiefenkarte*: Die neu erstellte Tiefenkarte soll die „gut“ gewählte Tiefen behalten und die aussortierten Bereiche füllen.
- *Fusion*: Dies ist der letzte Schritt des Algorithmus. Er soll die entstandene Tiefenkarte interpretieren und daraus mit Hilfe des Bildstapels ein überall scharfes Bild erzeugen.

Auf nachfolgenden Seiten wird schrittweise auf alle Punkte des Verfahrens eingegangen und bereits vorgestellten Methoden erläutert.

4.1 Das Erstellen der Anfangstiefenkarte

Im ersten Schritt wird eine vorläufige Tiefenkarte erstellt. Da sie Tiefen der Pixel im Fokus beinhaltet, müssen die Kriterien formuliert werden, die den „Schärfegrad“ der Pixel bestimmen. Boshayeva *et. al.* [4], [3] haben festgestellt, dass die Schärfe anhand der Grauwertdifferenzen zwischen dem betrachteten Pixel und seinen Nachbarn ermitteln kann. Denn unterscheiden sich die Werte nur schwach, gehören die Pixeln entweder zu einer großen einfarbigen Fläche, oder sie erscheinen auf dem Bild verschwommen. D.h. ein Schärfekriterium muss große Änderungen aufspüren und kleine nicht beachten.

In der Bildverarbeitung kennt man solche Kriterien als Kantenfilter. Meistens sind das Ableitungsfilter, die mit dem Quellbild gefaltet werden. Einige Beispiele dafür stellen Gradient, Laplace - Operator und die Hesse -Matrix dar. Die Filtermasken, die sie abbilden und die im Kapitel 3 ausführlich behandelt wurden, beinhalten Koeffizienten, deren Summe Null ergibt. Sind also die Grauwerte auf dem Bild identisch, würde die Faltung ebenfalls eine Null liefern. Interpretiert man das Ergebnis aus der Sicht der Ableitung, stellen solche Bereiche keine Änderung dar. Im Gegensatz dazu verstärken die Ableitungsfilter die Grauwerte, falls sie sich stark voneinander unterscheiden.

Bevor die Filter angewandt werden können, muss das Bild mit dem Gauß-Kern K_σ geglättet werden. Dadurch wird das größte Rauschen entfernt und die Ergebnisse werden kaum verfälscht. Das entstandene Bild kann als eine Funktion $f_\sigma(x, y)$ interpretiert werden. (x, y) steht für eine Position im zweidimensionalen Bereich Ω . Wendet man darauf die Kantenfilter an, erhält man die ersten oder die zweiten partiellen Ableitungen.

Der Gradient berechnet einen Vektor, der aus den partiellen Ableitung erster Ordnung zusammengesetzt wird.

$$\nabla f_\sigma(x, y) = (f_{\sigma x}(x, y), f_{\sigma y}(x, y))^T \quad (4.1)$$

Weil nur die Größe der Änderung für die Bestimmung der Schärfe relevant ist, nicht aber die Richtung, wird das Ergebnis in einen skalaren Wert umgewandelt, der lediglich den Betrag der Änderung repräsentiert. Der Betrag eines Vektor kann mithilfe der euklidischen Norm dargestellt werden.

$$m(x, y) = |\nabla f_\sigma(x, y)| \quad (4.2)$$

Ein weiteres Kriterium kommt in Form der Hesse-Matrix. Sie besteht aus den partiellen Ableitungen zweiter Ordnung, sowie aus der gemischten partiellen Ableitung. Diese Zusammensetzung erleichtert das Bestimmen der Krümmung einer Funktion. Damit können die zweiten Ableitungen ermittelt werden. Da die Matrix ebenfalls richtungsabhängige Lösung liefert, muss von dem Ergebnis der Faltung eine Norm bestimmt werden. Für die

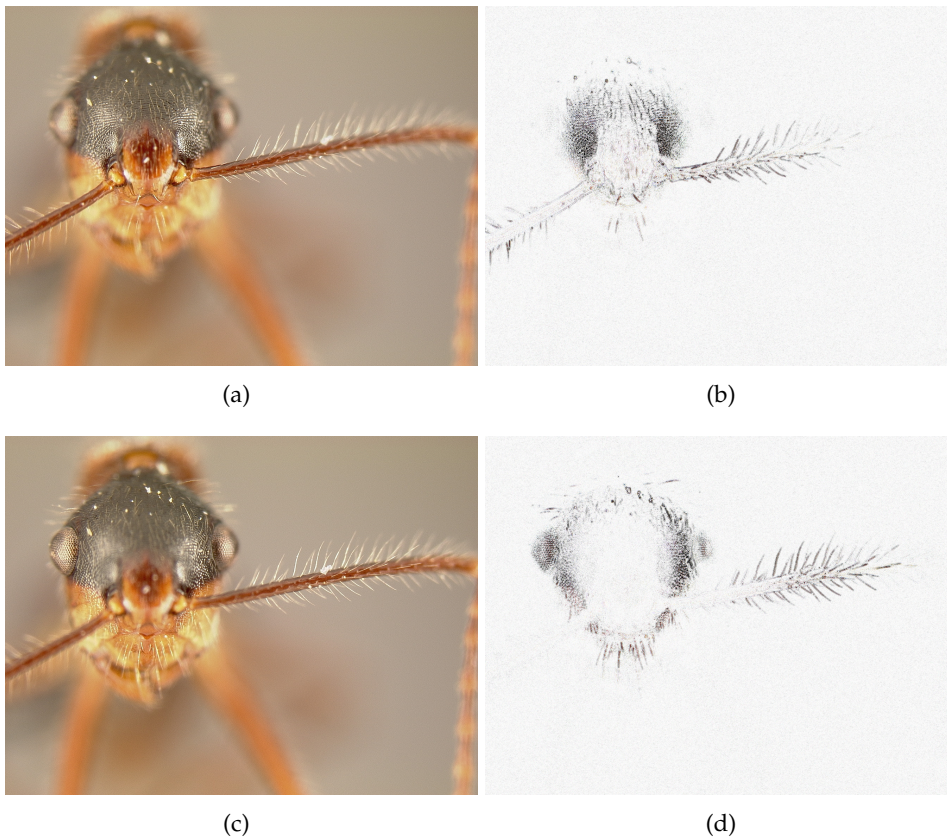


Abbildung 4.1: Beispiel der Wirkung von Schärfekriterien. (a) und (c) stellen die Eingaben dar, mit verschiedenen Schärfetiefen. (b) und (d) repräsentieren die Eingabebilder nach der Anwendung des Laplace-Betrags

Matrizen wird eine Variante der allgemeinen euklidischen Norm benutzt, die sogenannte Frobenius-Norm, die in der Gleichung 4.3 dargestellt ist.

$$m(x, y) = \|H(f_\sigma(x, y))\|_F = \sqrt{(\partial_{xx}f)^2 + 2(\partial_{xy}f)^2 + (\partial_{yy}f)^2} \quad (4.3)$$

Zusätzlich zu den oben genannten Schärfekriterien kann auch die Spur der Hesse-Matrix verwendet werden. Eine Spur bezeichnet in der Mathematik die Summe der Quadrate der Hauptdiagonalelementen einer Matrix. Im Bezug auf die Hesse-Matrix kennt man die Spur unter dem Namen *Laplace-Operator*

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.4)$$

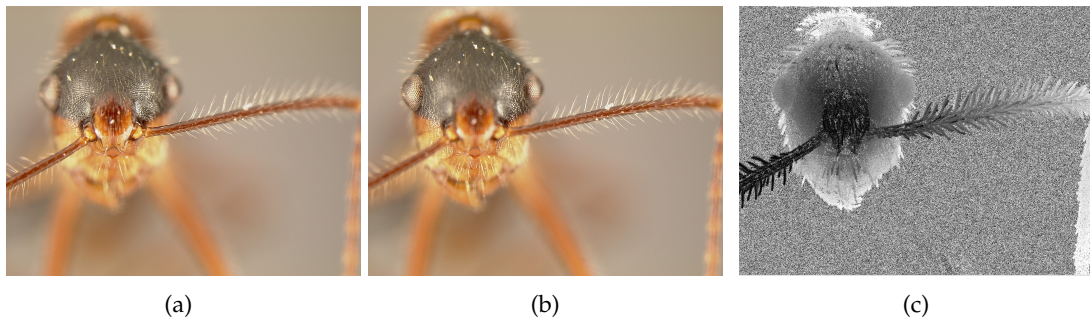


Abbildung 4.2: Die Erzeugung der Anfangstiefenkarte aus den Eingaben (a) und (b). Zwecks Übersicht wurden hier nur zwei Eingabebilder gewählt (aus ursprünglich 20). Je dunkler die Tiefenkarte, desto näher ist der entsprechende Bereich

Verwendet man die Masken aus der Gleichung 3.15, entsteht der folgende Stencil:

$$\partial f = \partial_{xx}f + \partial_{yy}f = \frac{1}{h^2} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \quad (4.5)$$

Der Laplace - Operator liefert die Summe der zweiten partiellen Ableitungen zurück als ein skalares Ergebnis, das negativ sein kann. Dementsprechend wird von dem Resultat ein Betrag bestimmt.

$$m(x, y) = |\Delta f_\sigma(x, y)| \quad (4.6)$$

Die Abbildung 4.1 (a, c) zeigt Bilder, die als Eingabe in dieser Arbeit verwendet werden. Die Pixel, die im Fokus liegen, werden von den Schärfekriterien, in diesem Fall Laplace - Operator, ermittelt und in den Abbildung (b, d) dargestellt. Pixel, die außerhalb des Fokus liegen, kommen in der Abbildung (b, d) nicht vor.

Weil der Eingabestapel mit Bildern gefüllt wurde, die mit verschiedenen Brennweiten aufgenommen wurden, kann man die Eingabe als die Schichtaufnahme eines dreidimensionalen Objekts $f_\sigma(x, y, z)$ auffassen, mit z die Tiefe des Pixels und damit die Reihenfolgennummer des Bildes auf dem Stapel. Man wendet deshalb die Schärfekriterien auf jedes Bild an. Betrachtet man die Bilder nacheinander, vergleicht die Schärfewerte für jedes (x, y) und wählt aus allen Bildern den größten Wert für den jeweiligen Pixel, erhält man die Anfangstiefenkarte d_{init} . Sie kann bereits für die Bildfusion verwendet werden, besitzt aber noch Rauschen. Das Ergebnis kann man der Abbildung 4.2 entnehmen.

4.2 Auswahl der Werte der initialen Tiefenkarte

Die Tiefenkarte auf der Abbildung 4.2 hat zwar die Tiefenwerte der schärfsten Pixel gespeichert, falls aber die Grauwertdifferenzen in einem Bereich sich nur marginal ändern, ist an dieser Stelle die Wahl der Tiefe eher zufällig. Solche „homogene“ Bereiche sind in der Abbildung 4.2 als „Rauschen“ erkennbar.

Deswegen liegt der nächste Schritt im Entfernen dieser Bereiche aus der Tiefenkarte. Üblicherweise wird zu diesem Zweck eine Entscheidungsfunktion aufgestellt, die die Werte gewichtet und sie anschließend mit einem Schwellwert vergleicht. Im einfachsten Fall nimmt man dafür eine Stufenfunktion c und legt einen Schwellwert θ fest[4].

$$c(m(x, y, d_{init}(x, y))) = \begin{cases} 1, & \text{falls } m(x, y, d_{init}(x, y)) > \theta, \\ 0, & \text{sonst.} \end{cases} \quad (4.7)$$

Das Ergebnis kann in der Abbildung 4.4 betrachtet werden.

Der Schwellwert wird manuell eingestellt. Die Funktion kann leicht implementiert werden, erzeugt aber keine optimale Lösung. Da der Schwellwert von dem Benutzer eingegeben wird, muss immer fallspezifisch entschieden werden, welchen Betrag für θ eingesetzt werden muss. Zusätzlich geht die Information verloren, wenn die Tiefenwerte aus der Karte entfernt werden.

Die Alternative zu der Stufenfunktion muss deswegen alle Werte auf der Karte berücksichtigen. Solche Funktionen sind in der Mathematik, und vor allem im Bereich der neuronalen Netzen, unter dem Begriff „Sigmoid-Funktion“ bekannt.

Eine Sigmoid-Funktion heißt auch Aktivierungsfunktion. Sie gewichtet alle ankommenden Werte in Abhängigkeit von dem Schwellenwert. Die Werte der Funktion sind auf dem Bereich $(0, 1)$ definiert. Sei c eine solche Sigmoid-Funktion mit der Gleichung:

$$c(m(x, y, d_{init}(x, y))) = \frac{1}{1 + e^{-s(m(x, y, d_{init}(x, y)) - \theta)}} \quad (4.8)$$

und der Funktionskurve in der Abbildung 4.3. Der Parameter θ stellt hier den Schwellwert dar. Zusätzlich verschiebt er die Funktion entlang der x-Achse, die hier alle möglichen Tiefenschärfen darstellt. Der Faktor s repräsentiert die Steigung der Kurve und steht maßgeblich für die Stärke der Gewichtung. Je größer die Steigung eingestellt ist, desto näher kommt die Sigmoid-Kurve an die Stufenfunktion heran.

Die entstandene Gewichte bewerten zwar die Tiefenwerte, entfernen sie nicht aus der Karte. Je rauschanfälliger der Tiefenwert ist, desto kleiner ist sein Gewicht, und umgekehrt. Die Gewichtung mit der Sigmoid-Funktion wird in der weiteren Implementierung des Algorithmus für ein ruhigeres Ergebnisbild sorgen.

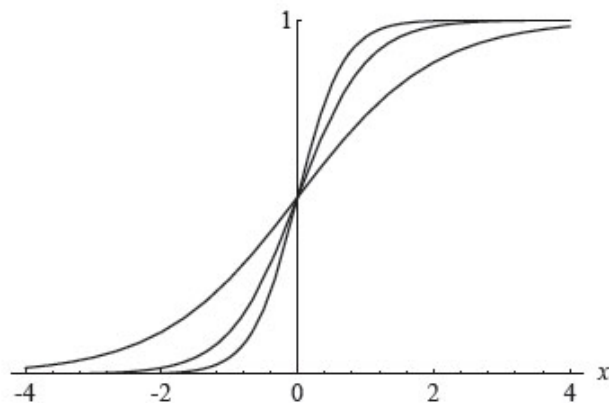


Abbildung 4.3: Beispiel der Änderung der Steilheit s in der Sigmoid - Funktion. Je größer der Wert, desto näher liegt Sigmoid-Kurve an der Stufenfunktion

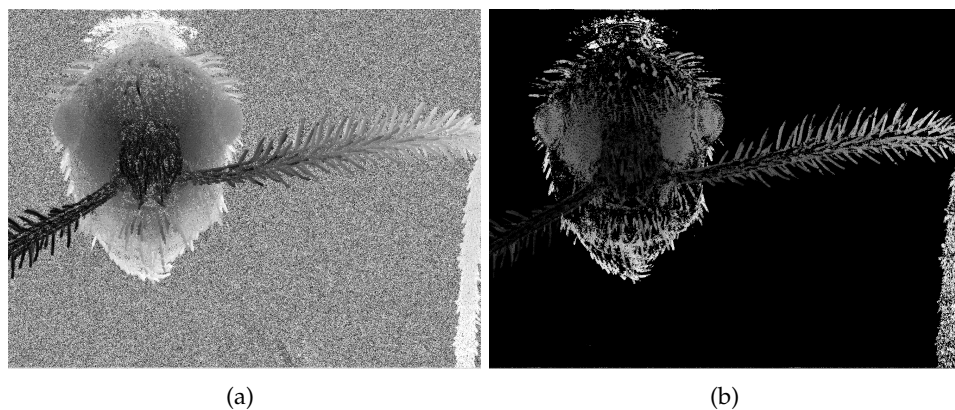


Abbildung 4.4: Anwendung der Entscheidungsfunktion auf die Anfangstiefenkarte. In diesem Beispiel wurde dafür die Stufenfunktion verwendet

4.3 Rekonstruktion der endgültigen Tiefenkarte

Die entstandene Tiefenkarte d_{init} hat nach der Gewichtung nur ein Teil ihrer Tiefenwerte. Je nach Wahl der Entscheidungsfunktion c wurden die Tiefenwerte entweder gänzlich aus der Lösungsmenge entfernt oder ihre Gewichte streben gegen Null. Weil jede Stelle eines Bildes in Fokus liegen muss, soll eine neue Tiefenkarte d erstellt werden, die alle Tiefenwerte aus d_{init} übernimmt und die restlichen Stellen „ausfüllt“. Das bedeutet, man sucht nach einer Funktion, die eine gegebene Punktemenge interpoliert. Dieser Abschnitt der Ausarbeitung beschäftigt sich deswegen mit der Aufstellung der allgemeinen Interpolationsgleichung für eine neue Tiefenkarte d .

Datenterm

Sei f eine gegebene Punktemenge, beschrieben durch eine Funktion. Man sucht nach einer kontinuierlichen Funktion u mit der Eigenschaft $(u - f)$ minimal, denn die Lösung u soll so nah wie möglich an der Ausgangsfunktion f liegen. Eine Funktion hat dann ein Minimum, wenn ihre erste Ableitung $u' = 0$ und ihre zweite $u'' > 0$ sind. Das Minimum soll zusätzlich global sein, denn so erhält man eine eindeutige Lösung für die obere Beziehung. Eine Klasse von Funktion, die ein globales Minimum haben, sind konvexe Funktion. Sie können aber mehrere Minima besitzen. Weil man eher an einer eindeutigen Lösung interessiert ist, greift man auf strikt-konvexe Funktionen zurück. Solche Funktion haben ein einziges globales Minimum. Zu den strikt-konvexen Funktionen gehören zum Beispiel quadratische Terme [12].

Formt man die lineare Differenz in eine quadratische um, erhält man $(u - f)^2$. Ihre erste Ableitung hat eine Lösung für den Fall $u = f$ gleich Null und die zweite Ableitung ist ein positiver Skalar, denn es wird angenommen dass u einen positiven Koeffizient hat. Die Beziehung kann zusätzlich mit einem Koeffizienten $\frac{1}{2}$ reguliert werden, um die Wirkung der Potenz auf die Ableitung zu verringern. $\frac{1}{2}(u - f)^2$ stellt sicher, dass die Interpolation der *Datenwerte* f mit einer gute Näherung erfolgt. Aus diesem Grund wird der Term auch *Datenerhaltungsterm*, oder kurz *Datenterm* genannt.

Weil man die Interpolation punktweise durchführt, muss für jedes $u(x, y) \in u$ die Differenz zu dem entsprechenden Wert $f(x, y)$ der Funktion f minimal sein. Die Gleichung $(u - f)$ wird damit zu einem Gleichungssystem [12]. Die Beziehung $(u - f)$ bestimmt eine Entfernung und kann deswegen in Form einer Summe der Quadrate einzelner Gleichungen dargestellt werden, mit Ω dem Definitionsbereich für (x, y) .

$$D(u) = \int_{\Omega} c \frac{1}{2} (u - f)^2 dx dy \quad (4.9)$$

Glattheitsterm

Die Gleichung 4.9 wird auch die *Methode der kleinsten Quadrate* genannt. Mit ihrer Hilfe lassen sich bereits existierenden Werte von $f(x, y)$ interpolieren. Nun gibt es auch noch Punkte, die zwar im Definitionsbereich enthalten sind, die aber keine Werte haben, d.h $c = 0$. Weil jedes $u(x, y)$ eine Unbekannte darstellt, hat das Gleichungssystem 4.9 mehr Unbekannte als Gleichungen. Solche *schlecht gestellt* Problemen haben unendliche viele Lösungen und eignen sich nicht als Interpolation [12], [15]. Deswegen erweitert man die Interpolation der Daten um eine *a priori* Annahme, die eine regularisierende Rolle für die Interpolation übernimmt.

Weil man keine Werte für $u(x, y)$ hat, wenn $c = 0$, nimmt man an, dass sie gleich den Werten in ihrer Umgebung sind. Auf diese Weise lassen sich starke Abweichungen in der Interpolation vermeiden. Die Tiefenkarte würde somit geglättet. Diese Bedingung korreliert mit der Feststellung von Boshayeva *et. al.*, dass benachbarten Werte der Tiefenkarte in etwa

gleich sein müssen, um einen kontinuierlichen Übergang der Tiefen zu gewährleisten [4]. Deswegen eignet sich die Gradientenberechnung als eine solche Annahme. Aus dem Kapitel 3 geht hervor, dass ein Gradient als eine Differenz der Bildwerte aufgefasst werden kann. Und weil die Funktion u möglichst glatt sein muss, also beliebig oft differenzierbar, darf sie keine Sprungstellen haben. Damit muss der Gradient für jede Stelle $u(x, y)$ ebenfalls minimal sein. Unter Verwendung der Methode der kleinsten Quadrate lässt sich die Minimierung aufschreiben als Glattheitsterm:

$$S(u) = \int_{\Omega} \frac{1}{2} |\nabla u|^2 dx dy \quad (4.10)$$

Energiefunktional

Nun hat man bereits alles um eine gute Interpolation aufzustellen. Fügt man die Gleichungen 4.10 und 4.9 zusammen, entsteht die Interpolationsgleichung 4.11.

$$E_f(u) = \frac{1}{2} \int_{\Omega} c(u - f)^2 + \alpha |\nabla u|^2 dx dy \quad (4.11)$$

Dieses Funktional ist unter dem *Tikhonov - Regularisierung* bekannt. Seine Minimierung liefert die Lösung für u .

4.3.1 Allgemeine Minimierung des Energiefunktionals

Die totale Energie in Form einer Kostenfunktion wird auch als Funktional bezeichnet. Dabei handelt es sich um eine Funktion vom Typ $\mathbb{R}^n \rightarrow \mathbb{R}$, d.h. man bildet die Funktionen auf einen Skalar ab.

1D-Fall

Betrachtet man einen eindimensionalen Fall, lässt sich der Energiefunktional auf folgende Weise darstellen:

$$E(u) = \int_a^b F(x, u, u') dx \quad (4.12)$$

Man nehme an, u hat die Form: $u(x, \epsilon) = v(x) + \epsilon h(x)$ [6]. Hier sei $v(x)$ ein möglichst oft differenzierbarer Minimierungsterm. $h(x)$ steht für eine beliebige Störfunktion und ϵ ist

ein Regulierungsparameter für die Störfunktion. D.h $v(x)$ minimiert $E(u)$ und damit den Funktional. Deswegen hat die skalare Funktion

$$g(\epsilon) = E(u(x, \epsilon)) = E(v(x) + \epsilon h(x)) \quad (4.13)$$

für den Fall $\epsilon = 0$ ein Minimum.

$$\frac{\delta E(u(x, \epsilon))}{\delta \epsilon} = \lim_{\epsilon \rightarrow 0} \frac{E(v(x) + \epsilon h(x)) - E(v(x))}{\epsilon} = \frac{dE(v(x) + \epsilon h(x))}{d\epsilon} = 0 \quad \forall h \quad (4.14)$$

Aus der Gleichung 4.14 folgt:

$$\begin{aligned} \frac{d}{d\epsilon} \int_a^b (F(x, u, u')) dx &= \int_a^b (F_u(x, u, u')h(x) + F_{u'}(x, u, u')h'(x)) dx \\ &= \int_a^b (F_u(x, v, v')h(x) + F_{u'}(x, v, v')h'(x)) dx = 0 \end{aligned} \quad (4.15)$$

Wendet man die partielle Integration an, erhält man:

$$0 = \int_a^b (F_u(x, v, v')h(x) - \frac{d}{dx} F_{u'}(x, v, v')h(x)) dx + F_{u'}(x, v, v')h(x) \Big|_{x=a}^{x=b} \quad (4.16)$$

Falls $v(x)$ eine Minimierung für eine Familie von ähnlichen Funktionen der Form $u(\epsilon, x) = v(x) + \epsilon h(x)$ ist, so minimiert sie auch eine kleinere Klasse von Funktionen, wobei die Störfunktion $h(x)$ die Bedingungen $h(a) = h(b) = 0$ erfüllt [6]. Damit gilt:

$$0 = \int_a^b (F_u(x, v, v')h(x) - \frac{d}{dx} F_{u'}(x, v, v')h(x)) dx \quad (4.17)$$

Das Integral 4.17 ist dann gleich 0, wenn das Innere des Integrals ebenfalls gleich 0 ist. Da h beliebig sein kann, muss somit gelten:

$$0 = F_u - \frac{d}{dx} F_{u'} \quad (4.18)$$

Diese Formulierung wird auch als Euler-Lagrange-Gleichung bezeichnet. Berücksichtigt man sie in der Gleichung 4.16, erhält man die natürliche Randbedingung:

$$F_{u'}(x, v, v')h(x) = F_{u'} = 0 \quad (4.19)$$

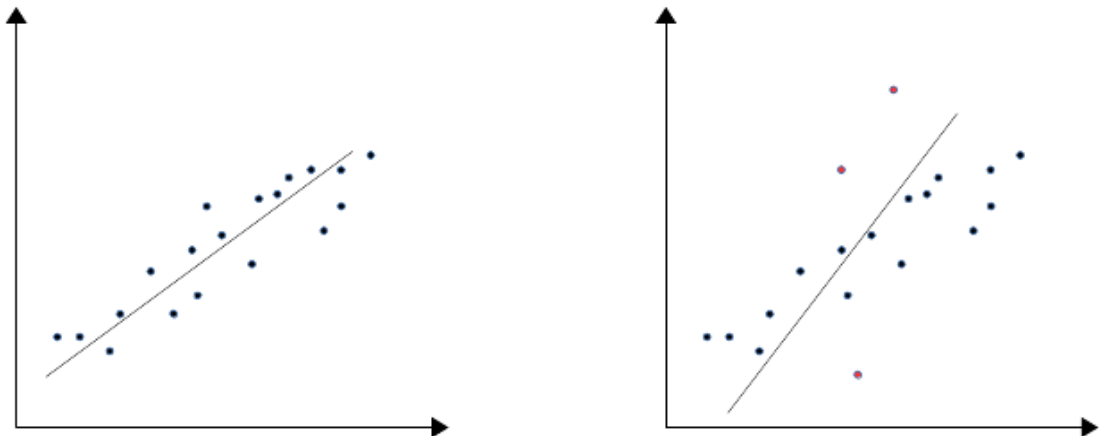


Abbildung 4.5: Einfluss von Ausreißern auf die lineare Regression. Die Verschiebung der roten Punkte führt zu einem neuen Regressionsmodell, dass die gegebene Punktemenge schlechter beschreibt

2D-Fall

Die obere Formulierung behandelt lediglich den eindimensionalen Fall. Erweitert man sie auf die zweite Dimension, erhält man die Euler-Lagrange-Gleichung, die in der Bildverarbeitung verwendet wird:

$$0 = F_u - \frac{d}{dx} F_{u_x} - \frac{d}{dy} F_{u_y} \quad (4.20)$$

Die natürliche Randbedingung wird in diesem Fall zu:

$$n^T \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = 0 \quad (4.21)$$

mit n als Normalenvektor über den Bildrand hinweg.

Die Randbedingung kann als Spiegelung der Randwerte um die Bildkante verstanden werden.

4.3.2 Minimierung des Energiefunktional

Wendet man die Minimierung des Energiefunktional aus dem letzten Kapitel auf die Gleichung 4.11 ein, müssen wir partielle Ableitungen erster Ordnung berechnen:

$$\begin{aligned} F_u &= c \cdot (u - f) \\ F_{u_x} &= \alpha u_x \\ F_{u_y} &= \alpha u_y \end{aligned} \tag{4.22}$$

Daraus lässt sich die untenstehende Euler-Lagrange-Gleichung formulieren.

$$0 = c \cdot (u - f) - \alpha \Delta u \quad \text{mit} \quad n^T \nabla u = 0 \tag{4.23}$$

Die Interpolation, die die Gleichung 4.23 liefert, wird auch als die lineare Interpolation bezeichnet. Ein Beispiel dafür ist in der Abbildung 4.5 dargestellt. Sie wird gerne als Modell verwendet, da ihre Implementierung leicht umgesetzt werden kann. Sie ist aber nicht besonders genau. Zudem ist das Funktional aufgrund der quadratischen Bestrafungsfunktion sehr anfällig gegenüber Ausreißern. Der Einfluss von Ausreißern auf das Ergebnis seiner Regression mit quadratischer Bestrafungsfunktion wird in der Abbildung 4.5 veranschaulicht. Um die Kosten möglichst gering zu halten, adaptiert sich die Lösung an die Ausreißer.

Energiefunktional mit robustem Datenterm

Sei $\Psi : [0, \infty] \rightarrow [0, \infty]$ eine monoton wachsende und hinreichend glatte Funktion der Form $\Psi((u - f)^2)$. Dann hat das Energiefunktional die Form:

$$E_f(u) = \frac{1}{2} \int_{\Omega} c \cdot \Psi((u - f)^2) + \alpha |\nabla u|^2 dx dy \tag{4.24}$$

mit der Euler-Lagrange-Gleichung:

$$0 = c(x, y) \Psi'((u - f)^2)(u - f) - \alpha \Delta u \quad \text{mit} \quad n^T \nabla u = 0 \tag{4.25}$$

Die Funktion Ψ soll die Abweichungen von der initialen Lösung reduzieren, deswegen wird sie auch als *Bestrafungsfunktion* bezeichnet. Einige dieser Funktionen sind in der Tabelle 4.1 aufgelistet. Der Fall $\Psi(s^2) = s^2$ wurde bereits im Abschnitt 4.3.2 verwendet.

Die Bestrafungsterme nach *Charbonnier* oder die *regularisierte Betragsfunktion* bieten eine deutlich robustere Interpolation für u , als eine quadratische Funktion. Ihre Abbildungen

haben einen subquadratischen Verlauf und die Ableitungsfunktionen streben deswegen für steigendes s^2 gegen Null (siehe Abbildung 4.6). Sie unterscheiden sich lediglich in der Geschwindigkeit, mit der ihre Werte gegen Null streben. Im Rahmen unseres Energiefunktionals mit robustem Datenterm kam die regularisierte Betragsfunktion zum Einsatz. Das liegt vor allem an der begrenzten Anzahl der Tiefenwerte. Bereits eine Differenz von $(u - f) = 1$ kann als große Abweichung angesehen werden. Mithilfe des Parameters ϵ können die Gewichte so eingestellt werden, dass keine Bestrafung für Wertdifferenzen unter 1 eintritt.

Energiefunktional mit kantenerhaltendem Glattheitsterm

Bisher wurde eine Tikhonov - Regulierung für den Glattheitsterm angenommen. Der Diffusionsterm Δu beschreibt deswegen eine sogenannte lineare isotrope Diffusion, also eine Diffusion, die alle Bereiche gleich stark diffundiert. Um die Struktur der Objekte zu erhalten, muss deswegen eine bessere Bestrafung eingeführt werden, die Kanten erkennt. Die Diffusion soll dann an diesen Kanten reduziert werden.

Kantenpunkte weisen große Ableitungen auf. Je größer also die Ableitung $|\nabla u|$, desto weniger stark soll die Bestrafungsfunktion ansteigen. In unserem Funktional haben wir deshalb die Funktion von *Charbonnier* dafür gewählt. Berücksichtigt man die bisherige Diskussion, entsteht eine neue Gleichung für das Energiefunktional:

$$E_f(u) = \frac{1}{2} \int_{\Omega} c \cdot \Psi_D((u - f)^2) + \alpha \Psi_S(|\nabla u|^2) dx dy \quad (4.26)$$

An dieser Stelle möchten wir auf die Herleitung der Euler-Lagrange-Gleichung verzichten und sie bereits in ihrer endgültigen Form aufschreiben:

$$0 = c \cdot \Psi'_D((u - f)^2)(u - f) + \alpha \cdot \text{div}(\Psi'_S(|\nabla u|^2)\nabla u) \quad \text{mit} \quad n^T \nabla u = 0 \quad (4.27)$$

4.3.3 Diskretisierung für quadratischen Daten- und Glattheitsterm

Bisher wurde nur ein kontinuierlichen Fall beschrieben. Bilder dagegen stellen diskrete Funktionen dar mit N und M als Anzahl der Zeilen und Spalten des Bildes und dem Pixelabstand h_x, h_y . Zusätzlich besteht die Eingabe aus endlich vielen Bildern. Damit kann die Tiefenkarte d nur Werte aus einer endlichen und abzählbaren Menge aufnehmen. Berücksichtigt man das, müssen die Positionsangaben (x, y) aus den vorherigen Diskussion durch die Pixelposition $(d_{i,j})$ ersetzt werden. Die Werte von $(i, j) \in \Omega$ sind auf einem Bildbereich $\Omega = [1..M, 1..N]$ definiert.

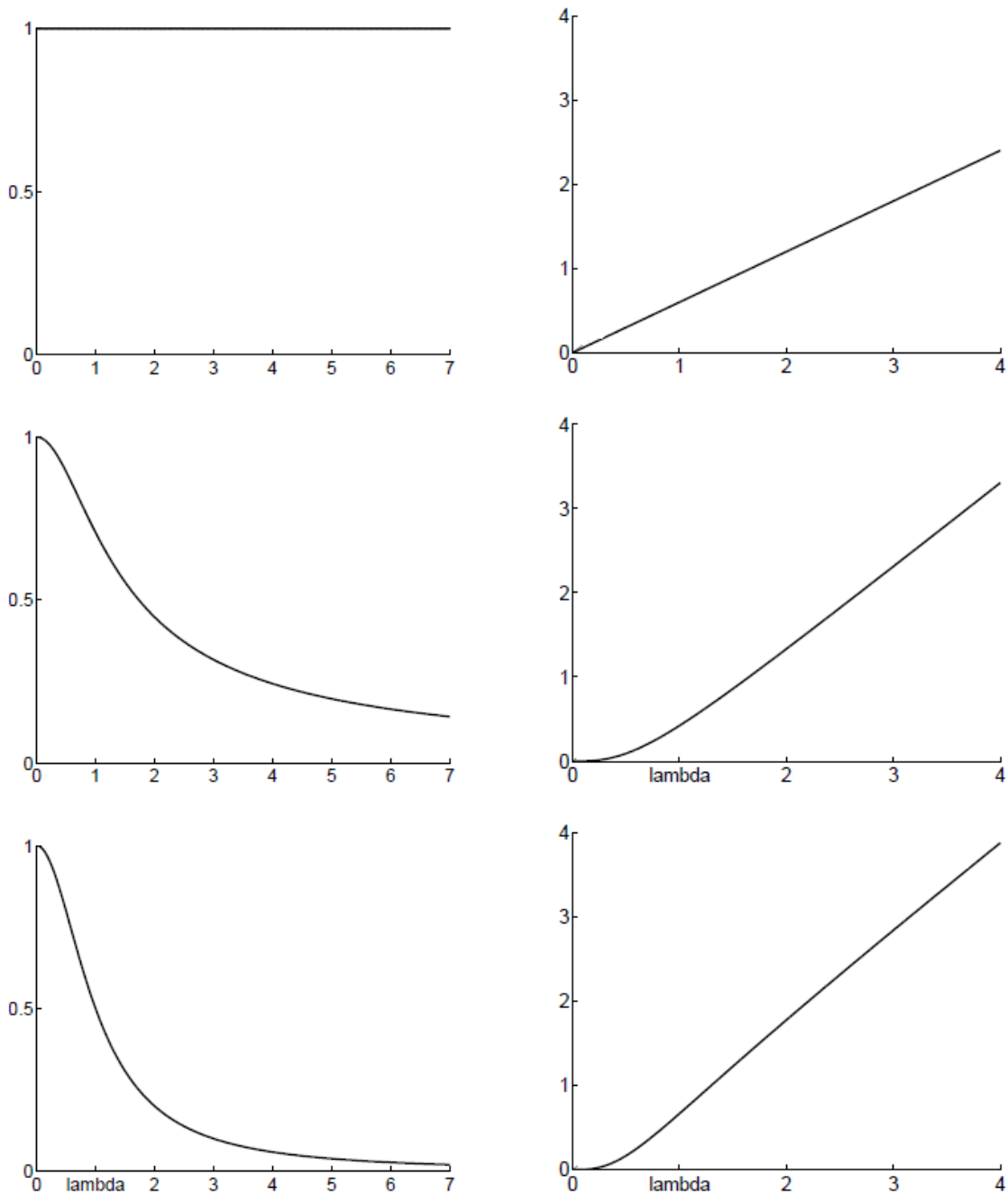


Abbildung 4.6: Zusammenfassung einiger nicht-linearen sowie linearer Bestrafungsfunktionen Ψ und ihre Ableitung. *Oben:* quadratisch, *Mitte:* Charbonnier, *Unten:* regularisierte Betragsfunktion. Rechte Spalte stellt Ψ' dar, die linke dagegen Ψ . Quelle: [14]

$\Psi(s^2)$	$\Psi'(s^2)$	Bestrafungsfunktion
s^2	1	quadratisch
$2\lambda^2(\sqrt{1 + \frac{s^2}{\lambda^2}} - 1)$	$(1 + \frac{s^2}{\lambda^2})^{-\frac{1}{2}}$	Charbonnier
$2(\sqrt{s^2 + \epsilon^2} - \epsilon)$	$(s^2 + \epsilon^2)^{-\frac{1}{2}}$	regularisierte Betragsfunktion
$\lambda^2 \log(1 + \frac{s^2}{\lambda^2})$	$(1 + \frac{s^2}{\lambda^2})^{-1}$	Perona - Malik

Tabelle 4.1: Varianten der Bestrafungsfunktion. Die oberen drei sind konvex, die letzte nicht.
Quelle: [7]

Man gehe von der Gleichung 4.11 aus und beachte, dass $u = d$ und $f = d_{init}$ sind. Zusätzlich wird angenommen, dass beide Bestrafungsterme quadratisch sind. Dadurch entsteht die Formulierung:

$$E_f(u) = \frac{1}{2} \int_{\Omega} c(d - d_{init})^2 + \alpha |\nabla d|^2 dydx \quad (4.28)$$

sowie die Euler-Lagrange-Gleichung:

$$0 = c \cdot (d - d_{init}) - \alpha \Delta d \quad \text{mit} \quad n^T \nabla d = 0 \quad (4.29)$$

Durch Diskretisierung dieser Gleichung entsteht ein lineares Gleichungssystem:

$$0 = c_{i,j}(d_{i,j} - d_{init,i,j}) - \alpha \left(\frac{d_{i-1,j} - 2d_{i,j} + d_{i+1,j}}{h_x^2} + \frac{d_{i,j-1} - 2d_{i,j} + d_{i,j+1}}{h_y^2} \right) \quad (4.30)$$

für $i = 1, \dots, N$ und $j = 1, \dots, M$.

Die partiellen Ableitungen zweiter Ordnung können unter der Voraussetzung $h_x = h_y = 1$ weiter vereinfacht werden.

Damit erhält die Gleichung 4.30 folgende Form:

$$0 = c_{i,j} \cdot d_{i,j} - c_{i,j} \cdot d_{init,i,j} - \alpha \cdot (d_{i-1,j} + d_{i+1,j} + d_{i,j-1} + d_{i,j+1} - 4d_{i,j}) \quad (4.31)$$

für $i = 1, \dots, N$ und $j = 1, \dots, M$.

Der Vektor $d = (d_{1,1}, \dots, d_{i,j})^T$ stellt die Lösungsmenge des Gleichungssystems dar.

Iterativer Löser

Weil das Gleichungssystem groß und dünn besetzt ist und darüber hinaus noch positiv definit ist, geht man iterativ vor und kann das Jacobi-Verfahren verwenden. Damit wird aus der Gleichung 4.31:

$$d_{i,j}^{n+1} = \frac{\alpha(d_{i-1,j}^n + d_{i+1,j}^n + d_{i,j-1}^n + d_{i,j+1}^n) + c_{i,j}d_{init,i,j}^n}{4 \cdot \alpha + c_{i,j}} \quad (4.32)$$

Das Jacobi-Verfahren ist für gewöhnlich sehr langsam. Denn zuerst werden die Lösungen für alle $d_{i,j}$ berechnet, bevor sie in der nächsten Iteration eingesetzt werden. Möchte man den Algorithmus beschleunigen, müssen die gerade bestimmten Werte $d_{i,j}^{n+1}$ im selben Iterationsschritt verwendet werden. Man legt fest, dass $d_{i-1,j}$ und $d_{i,j-1}$ die Ordnung $(n+1)$ haben, also gerade bestimmt wurden. Dies teilt den Bildbereich Ω in $(\mathcal{N}^- \cup \mathcal{N}^+) \in \Omega$ mit $(d_{i-1,j}, d_{i,j-1}) \in \mathcal{N}^-$ und $(d_{i+1,j}, d_{i,j+1}) \in \mathcal{N}^+$ auf.

Damit entsteht aus der Formulierung 4.32 die nachfolgende Gleichung:

$$d_{i,j}^{n+1} = \frac{\alpha(\sum_{\tilde{i},\tilde{j} \in \mathcal{N}^-} d_{\tilde{i},\tilde{j}}^{n+1} + \sum_{\tilde{i},\tilde{j} \in \mathcal{N}^+} d_{\tilde{i},\tilde{j}}^n) + c_{i,j}d_{init,i,j}^n}{4 \cdot \alpha + c_{i,j}} \quad (4.33)$$

die auch als Gauß-Seidel-Verfahren bezeichnet wird.

Die Gleichung 4.33 ist immer noch zu langsam im Bezug auf große Bilder. Daher hilft hier ein Kunstgriff: man vergrößert die Schrittweite der Iteration und geht dabei ein wenig weiter als vorgesehen. Damit extrapoliert man die Lösung auf einem Intervall zwischen dem aktuellen Wert $d_{i,j}^n$ und der Gauß-Seidel-Lösung.

$$d_{i,j}^{n+1} = \left(\frac{\alpha(\sum_{\tilde{i},\tilde{j} \in \mathcal{N}^-} d_{\tilde{i},\tilde{j}}^{n+1} + \sum_{\tilde{i},\tilde{j} \in \mathcal{N}^+} d_{\tilde{i},\tilde{j}}^n) + c_{i,j}d_{init,i,j}^n}{4 \cdot \alpha + c_{i,j}} \right) \cdot \omega + (1 - \omega) \cdot d_{i,j}^n \quad (4.34)$$

Diese Methode heißt auf Englisch *successive over relaxation*, kurz **SOR**. Der Gewichtungsp parameter steuert die Schrittweite. Ist $\omega < 1$ hat man eine sogenannte Unterrelaxation. Für $\omega = 1$ bekommt man das Gauß-Seidel-Verfahren und für $1 < \omega < 2$ erhält man eine Überrelaxation. Sei die Gleichung 4.33 durch eine Variable $\tilde{d}_{i,j}^{n+1}$ darstellbar. Dann kann das SOR umgeformt werden zu:

$$d_{i,j}^{n+1} = (1 - \omega) \cdot d_{i,j}^n + \omega \cdot \tilde{d}_{i,j}^{n+1} = d_{i,j}^n + \omega \cdot (\tilde{d}_{i,j}^{n+1} - d_{i,j}^n) \quad (4.35)$$

Diskretisierung für robusten Daten- und Glattheitsterm

Nimmt man eine nicht-lineare Bestrafungsfunktion für Daten- und Glattheitsterme, muss deren Diskretisierung ebenfalls durchgeführt werden. Aus der Euler-Lagrange-Gleichung 4.27 geht hervor, dass die Ableitung des Glattheitsterms gleich $\text{div}(\Psi'_S(|\nabla d|^2)\nabla d)$ ist. Die entsprechende Diskretisierung ist in der Gleichung 4.36 dargestellt [5].

$$\begin{aligned}
\text{div}(\Psi'_S \nabla d) &= (\Psi'_S d_x)_x + (\Psi'_S d_y)_y & (4.36) \\
&= \frac{(\Psi'_S d_x)_{i+\frac{1}{2},j} - (\Psi'_S d_x)_{i-\frac{1}{2},j}}{2(\frac{1}{2}h_x)} + \frac{(\Psi'_S d_y)_{i,j+\frac{1}{2}} - (\Psi'_S d_y)_{i,j-\frac{1}{2}}}{2(\frac{1}{2}h_y)} \\
&= \frac{\Psi'_{S,i+1,j} + \Psi'_{S,i,j}}{2} \left(\frac{d_{i+1,j} - d_{i,j}}{h_x^2} \right) - \frac{\Psi'_{S,i,j} + \Psi'_{S,i-1,j}}{2} \left(\frac{d_{i,j} - d_{i-1,j}}{h_x^2} \right) \\
&+ \frac{\Psi'_{S,i,j+1} + \Psi'_{S,i,j}}{2} \left(\frac{d_{i,j+1} - d_{i,j}}{h_y^2} \right) - \frac{\Psi'_{S,i,j} + \Psi'_{S,i,j-1}}{2} \left(\frac{d_{i,j} - d_{i,j-1}}{h_y^2} \right) \\
&= w_{i,j,i+1,j} \left(\frac{d_{i+1,j} - d_{i,j}}{h_x^2} \right) - w_{i,j,i-1,j} \left(\frac{d_{i,j} - d_{i-1,j}}{h_x^2} \right) \\
&+ w_{i,j,i,j+1} \left(\frac{d_{i,j+1} - d_{i,j}}{h_y^2} \right) - w_{i,j,i,j-1} \left(\frac{d_{i,j} - d_{i,j-1}}{h_y^2} \right)
\end{aligned}$$

mit $|\nabla d|^2 = \sqrt{d_x^2 + d_y^2}$. Die Werte von d_x und d_y werden diskretisiert durch:

$$\begin{aligned}
d_x &\approx \frac{d_{i+1,j} - d_{i-1,j}}{2h_x} & (4.37) \\
d_y &\approx \frac{d_{i,j+1} - d_{i,j-1}}{2h_y}
\end{aligned}$$

Im Bezug auf den Datenterm erhält man für die Ableitung

$$\Psi'_D((d_{i,j} - d_{\text{init},i,j})^2) \cdot (d_{i,j} - d_{\text{init},i,j}) \quad (4.38)$$

Weil $\Psi'(s^2)$ ein nicht-linearer Term sein kann, muss noch eine Anpassung am Algorithmus durchgeführt werden. Es wird eine sogenannte „lagged non-linearity“ Methode verwendet [3]. Dabei implementiert man eine nichtlineare Fixpunktiteration mit zwei Schleifen: In der äußeren Schleife bestimmt man die $\Psi'(s^2)$ und in der Inneren fixiert man sie und rechnet wie gewohnt im Fall von SOR weiter, mit den Gewichten als linearen Koeffizienten [3]. Wenn man das Gewicht von Nachbarpixel \tilde{i}, \tilde{j} bzgl. der Pixel i, j als $w_{i,j,\tilde{i},\tilde{j}}$ bezeichnet, ergibt sich

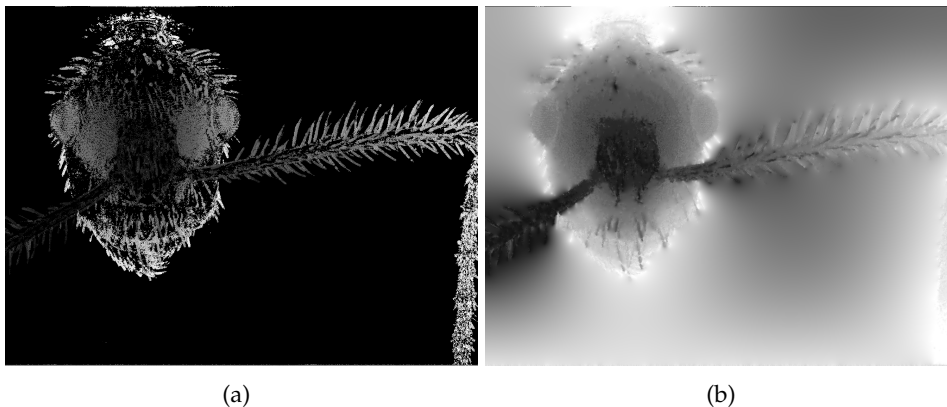


Abbildung 4.7: Übergang zwischen der Anfangstiefenkarte und der finalen Tiefenkarte. Das Bild entstand unter der Anwendung des Variationsansatzes mit quadratischen Daten- und Glattheitsterm. Der Schwellwert ist $\theta = 40$ der Glattheitsparameter ist $\alpha = 2$

die endgültige Form für das SOR Verfahren zu Lösung des linearen Gleichungssystems innerhalb der nichtlinearen Fixpunktiteration:

$$d_{i,j}^{n+1} = \left(\frac{\alpha \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}} d_{\tilde{i}, \tilde{j}}^n \cdot w_{i,j,\tilde{i},\tilde{j}} + c_{i,j} \Psi'_{D,i,j} d_{init,i,j}}{\alpha \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}} w_{i,j,\tilde{i},\tilde{j}} + c_{i,j} \Psi'_{D,i,j}} \right) \cdot \omega + (1 - \omega) \cdot d_{i,j}^n \quad (4.39)$$

Die endgültige Tiefenkarte kann man der Abbildung 4.7 entnehmen.

4.4 Fusion

Die interpolierte Tiefenkarte eignet sich noch nicht für die Fusion, denn ihre Werte sind auf dem kontinuierlichen Wertebereich $d \in \mathbb{R}$ definiert. Da jeder Tiefe ein separates Bild auf dem Stapel zugeordnet wird, kann d aber nur diskrete Werte besitzen, mit dem Minimalabstand von $|d_n - d_{n-1}| = 1$ zueinander.

Direkte Pixelfusion

Sei $d_{i,j}$ eine Tiefe mit $d_{i,j} \in \mathbb{R}$. Um eine reelle Zahl in eine Ganze zu überführen, werden die Abweichungen zwischen $d_{i,j}$ und der kleinsten oberen und der größten unteren ganzen Zahl bestimmt. Diese Abweichung übernimmt anschließend die Rolle eines Gewichts. Anschließend werden die Farbwerte der benachbarten Tiefen gemäß der Gewichte interpoliert [3].

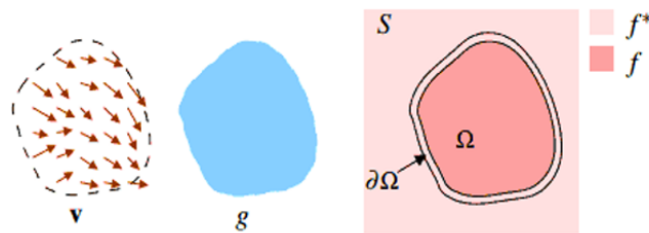


Abbildung 4.8: Poisson Image Editing Verfahren. g beschreibt den einzusetzenden Bereich, v ist sein Leitfeld und S - das Zielbild, wobei f eine unbekannte Funktion darstellt und f^* ist repräsentiert die Grauwerte des Zielbildes. Quelle: [16]

Sei hier die obere Schranke gleich $\lceil d \rceil$ und die untere Schranke gleich $\lfloor d \rfloor$. Man führt die Abweichung $\delta = d_{i,j} - \lfloor d \rfloor$ ein. Weil die Interpolation linear ist, beträgt die Abweichung zu der anderen Schranke $(1 - \delta)$. Die Interpolation für $s_{i,j}$ sieht dann wie folgt aus:

$$s_{i,j} = (1 - \delta) \cdot f_{\lfloor d_{i,j} \rfloor} + \delta \cdot f_{\lceil d_{i,j} \rceil} \quad (4.40)$$

Das Fusionsverfahren, das von Boshtayeva *et. al* implementiert wurde, nimmt auf diese Weise gewonnenen Werte der finalen Tiefenkarte und fügt die Pixel aus den entsprechenden Bildern zu einem Ausgabebild hinzu.

Gradient-basierte Fusion

Eine andere Technik, ausgearbeitet von Perez *et.al.* [16], fusioniert die Bilder im Gradientenraum. In ihrer Arbeit stellen sie fest, dass der Grauwert des einzusetzenden Pixels sich nur minimal von seiner Umgebung unterscheiden soll, damit die Einsetzung nahtlos erscheint. Das bedeutet, man greift an dieser Stelle auf die Minimierung der Abweichung zurück. Im vorigen Kapitel wurden bereits Methoden vorgestellt, die eine Lösung für das Minimierungsproblem liefern. Die Methode von Perez *et.al.* wird als **Poisson Image Editing** - Verfahren bezeichnet. Der Name stammt von der Minimierungsgleichung für das spezielle Kostenfunktional, die als Poisson - Gleichung bezeichnet wird.

Man betrachte die Abbildung 4.8. $S \in \mathbb{R}^2$ stellt das erwartete Ergebnis der Fusion dar. $\Omega \in S$ ist ein Bereich auf dem Bild, in das ein neues Objekt, im Fall dieser Arbeit ein Pixel, eingesetzt wird. $\partial\Omega$ beschreibt den Rand dieses Bereich [16]. f^* ist eine bekannte Funktion mit dem Definitionsbereich $f^* \in S \setminus \Omega$, $f \in \Omega$ soll berechnet werden, in diesem Fall der Grauwert des Pixels, und schließlich v ist ein Vektorfeld, definiert auf Ω . Die Lösung soll eine

gute Anpassung des eingesetzten Wertes an die Zielumgebung liefern. D.h. man erwarten eine optimale Lösung.

Wie schon im Kapitel 4.3.2 zu sehen, lassen sich Interpolationsmodelle als Minimierung eines Energiefunktionals aufstellen. Die Gleichung 4.11 kann an dieser Stelle ebenfalls verwendet werden. Weil man nur die Umgebungsanpassung der Grauwerte durchführen möchte, verzichtet man auf die Bestimmung des Datenterms. Damit wird das Energiefunktional zu:

$$E(s) = \frac{1}{2} \int_{\Omega} |\nabla s|^2 dx dy \quad (4.41)$$

Die Minimierung dieses Terms, also die Lösung der Euler-Lagrange-Identität, liefert ein stark geglättetes Interpolant [16]. Deswegen bedient man sich eines Leitfeldes, mit dem man richtungsabhängig interpolieren kann. Diese regulierende Rolle übernimmt das Feld v . Die Gleichung 4.41 muss dann so angepasst werden, dass die Änderung im eingesetzten Pixel und die in seiner Umgebung fast ähnlich sind. Das entsprechende Energiefunktional hat dann die Form:

$$E(s) = \frac{1}{2} \int_{\Omega} |\nabla s - v|^2 dx dy \quad (4.42)$$

Sei $s(x, y) = f^{d(x,y)}(x, y)$ eine Funktion im Raum S , die den Grauwert des Pixels im fusionierten Bild, mit der Tiefe $d(x, y)$ bestimmt. Sei $v(x, y) = (p(x, y), q(x, y))^T$ das Leitfeld und in dieser Arbeit durch ein Gradientenfeld repräsentiert. Die einzelnen Einträge des Feldes sind partielle Ableitungen erster Ordnung in x - und y -Richtung der Form:

$$\begin{aligned} p(x, y) &= [f_x]^{d(x,y)}(x, y) \\ q(x, y) &= [f_y]^{d(x,y)}(x, y) \end{aligned} \quad (4.43)$$

Eingesetzt in die Gleichung 4.42 ergibt das:

$$E(s) = \int_{\Omega} (s_x - p)^2 + (s_y - q)^2 dx dy \quad (4.44)$$

Dann liefert die anschließende Minimierung:

$$F_s - \frac{\partial}{\partial x} F_{s_x} - \frac{\partial}{\partial y} F_{s_y} = 0 \quad \text{mit} \quad n^T \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = 0 \quad (4.45)$$



Abbildung 4.9: Der Übergang zwischen der finalen Tiefenkarte und der daraus entstandenen Fusionsbild. Hier wurde die direkte Pixelfusion verwendet.

Formt man 4.45 mithilfe der Euler-Lagrange-Gleichung um, erhält man die Poisson-Gleichung [16]:

$$\Delta f = \operatorname{div} v \quad (4.46)$$

Lassen Sie uns nun das Funktional und die Euler-Lagrange-Gleichung diskutieren. Sei (i, j) die aktuelle Position. Dann hat $v_{i,j}$ folgende Einträge:

$$p_{i,j} = [f_x]_{i,j}^{d_{i,j}} = \frac{f_{i+1,j}^{d(i,j)} - f_{i,j}^{d(i,j)}}{h_x} \quad (4.47)$$

$$q_{i,j} = [f_y]_{i,j}^{d_{i,j}} = \frac{f_{i,j+1}^{d(i,j)} - f_{i,j}^{d(i,j)}}{h_y}$$

Die Gleichung 4.44 hat unter der Verwendung der diskreten Darstellung für die erste Ableitung die Form:

$$E(s) = \sum_{i,j} \left(\left(\frac{s_{i+1,j} - s_{i,j}}{h_x} \right) - p_{i,j} \right)^2 + \left(\left(\frac{s_{i,j+1} - s_{i,j}}{h_y} \right) - q_{i,j} \right)^2 \quad (4.48)$$

$$E(s) = \left(\left(\frac{s_{i,j} - s_{i-1,j}}{h_x} \right) - p_{i-1,j} \right)^2 + \left(\left(\frac{s_{i+1,j} - s_{i,j}}{h_x} \right) - q_{i,j} \right)^2 \quad (4.49)$$

$$+ \left(\left(\frac{s_{i,j} - s_{i,j-1}}{h_y} \right) - p_{i,j-1} \right)^2 + \left(\left(\frac{s_{i,j+1} - s_{i,j}}{h_y} \right) - q_{i,j} \right)^2$$

Die Euler-Lagrange-Gleichung kann dann folgendermaßen dargestellt werden:

$$\frac{s_{i-1,j} - 2s_{i,j} + s_{i+1,j}}{h_x^2} + \frac{s_{i,j-1} - 2s_{i,j} + s_{i,j+1}}{h_y^2} = \frac{p_{i,j} - p_{i-1,j}}{h_x} + \frac{q_{i,j} - q_{i,j-1}}{h_y} \quad (4.50)$$

für $i = 1, \dots, N, j = 1, \dots, M$. Man muss an dieser Stelle beachten, dass p und q partielle erste Ableitungen darstellen und ebenfalls diskretisiert werden müssen. Die Gleichung kann mit Hilfe des SOR-Verfahrens gelöst werden. Für $h_x = h_y = 1$ erhalten wir:

$$s_{i,j}^{k+1} = \left(\frac{\sum_{\tilde{i}, \tilde{j} \in \mathcal{N}^-} s_{\tilde{i}, \tilde{j}}^{n+1} + \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}^+} s_{\tilde{i}, \tilde{j}}^n - p_{i,j} - p_{i-1,j} - q_{i,j} - q_{i,j-1}}{4} \right) \cdot \omega + (1 - \omega) s_{i,j}^k \quad (4.51)$$

Das fertige, überall scharfe, Bild ist in der Abbildung 4.9 rechts zu sehen.

5 Experimentelle Ergebnisse

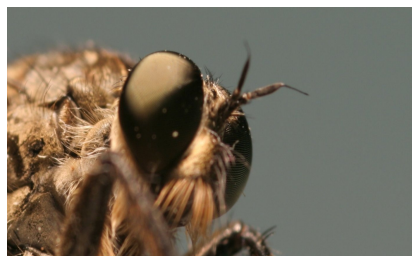
In diesem Kapitel werden wir die Ergebnisse nach der Anwendung der einzelnen Algorithmschritte präsentieren und die Wirkung der Eingabeparameter visuell vergleichen. Für den Test wurden drei Datensätze verwendet, die einen variierenden Anteil an feinen Strukturen und homogenen Flächen haben. Dadurch soll die Wirkung der Schärfekriterien, aber vor allem die Ergebnisse der Diffusion untersucht werden. Die Repräsentanten der verwendeten Bilderreihen sind in der Abbildung 5.1 dargestellt. Die Bilderreihen *flower*, *fly*, *fly2* bestehen aus jeweils 20 Bildern. Die Versuchsreihen wurden auf einem Windows PC durchgeführt, mit einem T2400 Prozessor (1,84) Ghz und 2 GB Arbeitsspeicher.

5.1 Vergleich und Evaluierung der Schärfekriterien

Für den Vergleich von Schärfekriterien wurden drei Verfahren aus dem Kapitel implementiert. Die Ergebnisse in Form der Anfangstiefenkarte kann man der Abbildung 5.2 entnehmen,



(a)



(b)



(c)

Abbildung 5.1: Die Bilderreihen (a)*flower*, (b)*fly*, (c)*fly2* wurden in dieser Arbeit als Testsets verwendet. Quelle: [2]

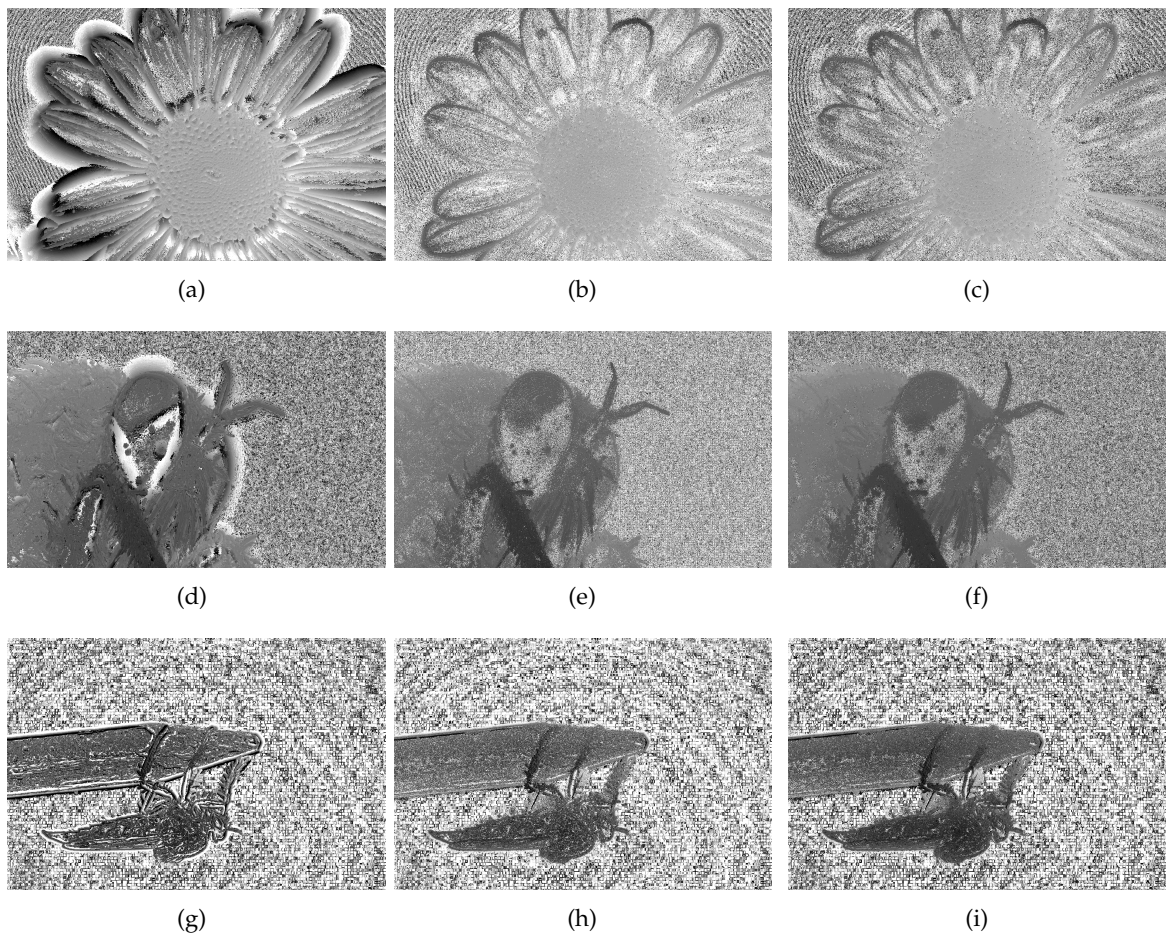


Abbildung 5.2: Vergleich von Schärfekriterien anhand der Anfangstiefenkarten. Von links nach rechts: Gradientensteigung, Betrag des Laplace-Operators, Frobenius - Norm der Hesse- Matrix

von links nach rechts: Gradientensteigung, Betrag des Laplace-Operators, Frobenius-Norm der Hesse-Matrix. Das Schärfekriterium soll die Wahl der Tiefe korrekt durchführen und nur für homogene Flächen Rauschen erzeugen.

Man kann der Abbildung 5.2 entnehmen, dass die Gradientensteigung in allen vier Bildsätzen am schlechtesten abschneidet. Die Kanteninformation geht verloren, die Tiefen verschmieren. Das liegt an dem niedrigen Informationsgehalt der ersten Ableitung, gegenüber der zweiten Ableitung des Laplace- sowie die Hesse-Schärfekriterium. Weil die Hesse-Matrix nicht nur Extremstellen berücksichtigt, sondern auch den Bereich dazwischen als lokale Krümmung bestimmt, enthält sie die meiste Information und stellt damit hier das beste Kriterium dar. Man erkennt, dass das Rauschen auf den Bildern (c), (f), (i) sich mehr auf die homogene Bereiche beschränkt. Zudem werden die Tiefen bereits klar voneinander separiert

und Struktur des Objekts bleibt damit erhalten. Das bedeutet vor allem, dass feine Strukturen korrekt erkannt werden und eine korrekte Tiefe zugewiesen bekommen.

5.2 Vergleich und Evaluierung der Entscheidungsfunktionen

In dieser Arbeit wurden beide Konfidenzfunktionen aus Kapitel implementiert. Die Abbildung 5.3 stellt die Ergebnisse für das jeweilige Datenset dar. Die schwarzen Pixeln in den Bildern der mittleren Spalte repräsentieren die Stellen, deren Tiefenwerte über dem Schwellwert liegen. Die rechte Spalte zeigt die initiale Tiefenkarte nach der Anwendung einer Stufen-Funktion. Beide Konfidenzfunktionen verwenden denselben Schwellwert θ für das jeweilige Datenset. Die jeweiligen Schwellwerten wurden so gewählt, dass die Form des Objekts deutlich erkennbar ist, und trotzdem wird das meiste Rauschen als solches erkannt und entweder aus der Tiefenkarte entfernt, oder mit dem entsprechend kleinem Gewicht versehen.

Man erkennt, dass nach der Anwendung der Stufenfunktion der große Teil der Tiefenkarte gleich Null gesetzt wird (hier als weißer Bereich erkennbar). Dagegen werden im Fall einer Sigmoid-Funktion alle Tiefenwerte mitberücksichtigt. Der weiße Bereich hier stellt Gewichte unterhalb $c(m(x, y, d(x, y))) < \theta$ dar. Ihre Wirkung ist zwar gering, nichtsdestotrotz geht ihre Information nicht verloren. Da die Sigmoid -Konfidenzfunktion die korrekt gewählten Tiefeninformationen nicht verliert und zusätzlich auch die restlichen Tiefen berücksichtigt, ist sie im Vergleich zu der Stufenfunktion besser als Auswahlfunktion geeignet. Sie wird nachfolgend als die Funktion der Wahl in dieser Arbeit verwendet.

5.3 Evaluierung der Wirkung von sub-quadratischen Bestrafungstermen

Die Abbildung 5.4 stellt die endgültige Tiefenkarte nach dem Einsatz der quadratischen Bestrafungsfunktion für den Daten- und Glattheitsterm. Rechts werden die Fusionsergebnisse gezeigt, die aus der direkten Pixelfusion entstanden sind. Die Tabelle 5.1 fasst alle Eingabeparameter für das Energiefunktional zusammen. Die Werte für $\lambda, \theta, \epsilon$ sind für die entsprechende Datensets in allen Versuchsreihen gleich.

Da hier die Sigmoid-Funktion zur Gewichtung der Tiefenwerte verwendet wurde, bewirkt Rauschen trotz des kleinen Gewichts die Anpassung des Modells. Dadurch entstehen dunkle Flecken in den homogenen Bereichen. Zusätzlich verschwimmen die Kanten, denn hier wird die lineare isotrope Diffusion eingesetzt. Die glättet in alle Richtungen gleich, ohne Berücksichtigung der Kantenverläufe.

Die Abbildung 5.5 nutzt eine subquadratische Bestrafungsfunktion für den Datenterm, nämlich die regularisierte Betragsfunktion. Sie bestraft die Abweichungen, die durch die Gewichte unterhalb des Schwellwerts entstehen. Dadurch werden in der Abbildung 5.5 homogene Bereiche stärker geglättet, denn sie fallen nicht mehr stark ins Gewicht. Die

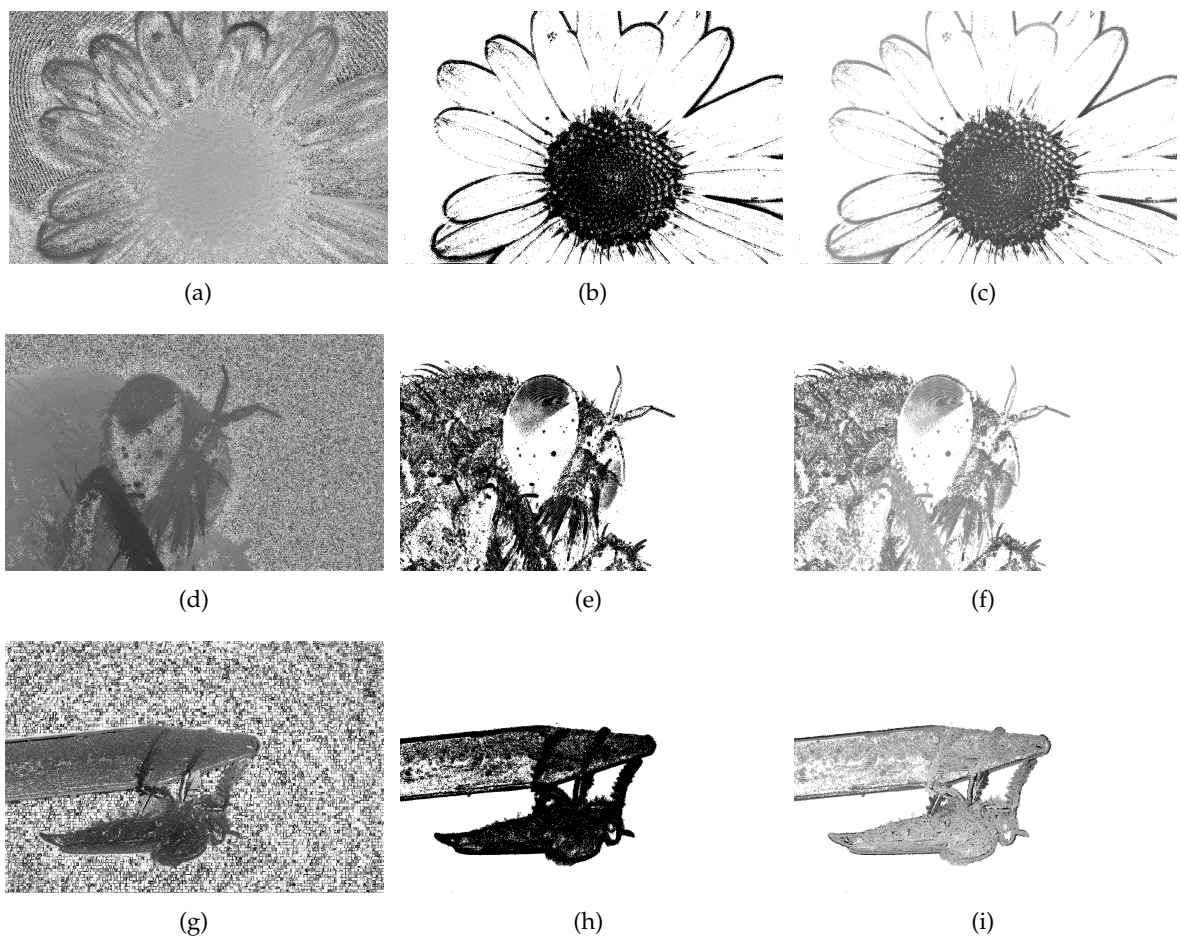


Abbildung 5.3: Vergleich von Konfidenzfunktionen. *Linke Spalte:* Die initiale Tiefenkarte, erzeugt mit der Frobenius-Norm der Hesse-Matrix. *Mittlere Spalte:* Schwarz-Weiß darstellung der beiden Konfidenzfunktionen. Schwarz repräsentiert Tiefenwerte größer als θ . *Rechte Spalte:* Gefilterte initiale Tiefenkarte. Hier mit einer Stufen-Funktion. Die Farben in der rechten Spalte wurden zur besseren Übersicht invertiert

Kantenglättung erfolgt hier ebenfalls linear. Das Tiefenkarte ist hier viel glatter als in der Abbildung 5.4.

Wird der Glattheitsterm ebenfalls subquadratisch Bestraft, entsteht die Abbildung 5.6. Hier wird die Charbonnier-Funktion eingesetzt, zusätzlich zu der regularisierte Betragsfunktion für den Datenterm. Man erkennt vor allem am Bild (e), dass es insgesamt nicht nur glatt ist, sondern die Kanten im Vergleich zu der Abbildung 5.5(e) klar sind. Die Abbildung 5.7 stellt die fusionierte Bilder des Datensets *fly2* unter Anwendung der oben beschriebenen Bestrafungsfunktionen. Der rote Rechteck ist das Gebiet von Interesse. Hier ist zu sehen, dass das robuste Energiefunktional (c) mehr Informationen liefert, als die beiden anderen (a) und

Datenset	Iterationen	θ	α	λ	ϵ
<i>fly</i>	10000	15	10	0.2	0.4
<i>fly2</i>	10000	10	10	0.3	0.4
<i>flower</i>	10000	10	10	0.65	0.3

Tabelle 5.1: Die Eingabeparameter für Schwellwert (θ), Glattheitskoeffizient (α), Datentermregularisierung (ϵ) und Glattheitstermregularisierung (λ)

(b). In (a) werden die Kanten geglättet und die Tiefen teilweise nicht korrekt bestimmt (siehe die Füße des Insekts). Dagegen hat (b) wegen der subquadratische Bestrafungsfunktion des Datenterms ausgeprägte Details im Bild, aber viel zu verwaschene Kanten.

5.4 Vergleich und Evaluierung der Fusionsverfahren

Die Ergebnisse der direkten Pixelfusion sowie der Fusion mithilfe des Poisson Image Editing können in der Abbildung 5.8 betrachtet werden. Hier haben wir die Ergebnisse als Graustufenbilder ausgegeben, um die Implementierung des Poisson Image Editing zu vereinfachen. Entsprechend, um den Vergleich einheitlich zu gestalten, haben wir die Ausgaben der direkten Pixelfusion als Graustufenbilder neu generiert. Man sieht, dass die Ergebnisse in etwa identisch sind. Wegen der wiederholten Glättung, bedingt durch die Diffusion, erscheinen die Ergebnisse in der rechten Spalte der Abbildung 5.8 leicht verwaschen. In der Abbildung 5.9 ist die Vergrößerung des Fusionsbildes des Datensets *fly2* dargestellt. Die roten Rechtecke markieren Stellen vom Interesse. Die Härchen, sowie Teile des Körpers des Insekts sind in der Abbildung (b) unscharf, dagegen erscheinen sie in der Abbildung (a) deutlicher.



Abbildung 5.4: Endgültige Tiefenkarte und das fusionierte Bild unter Verwendung von quadratischen Bestrafungsfunktionen für Daten- und Glattheitsterme. Die rechte Spalte entstand nach der direkten Pixelfusion.



Abbildung 5.5: Endgültige Tiefenkarte und das fusionierte Bild unter Verwendung von quadratischen Bestrafungsfunktionen für Glattheitsterme und regularisierten Bestrafungsfunktion für Datenterm. Die rechte Spalte entstand nach der direkten Pixelfusion.

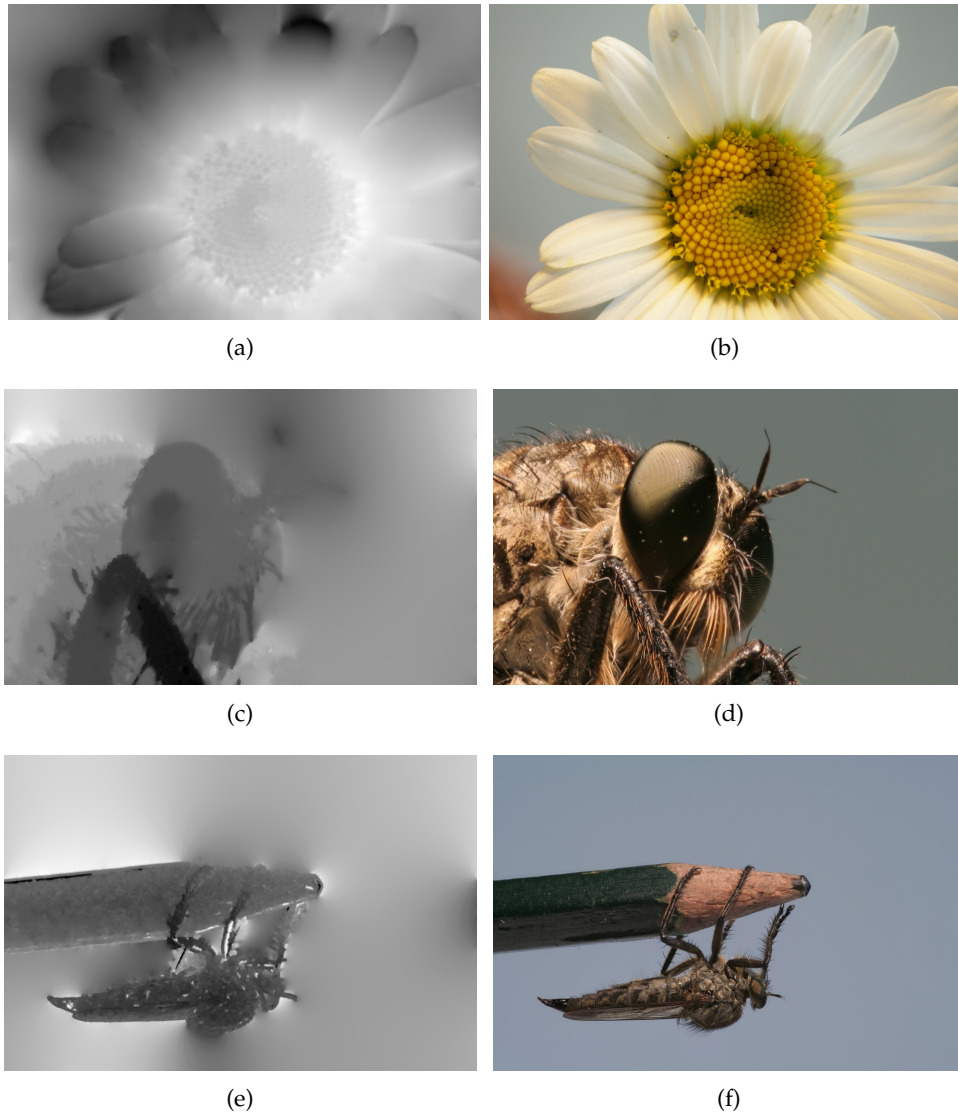
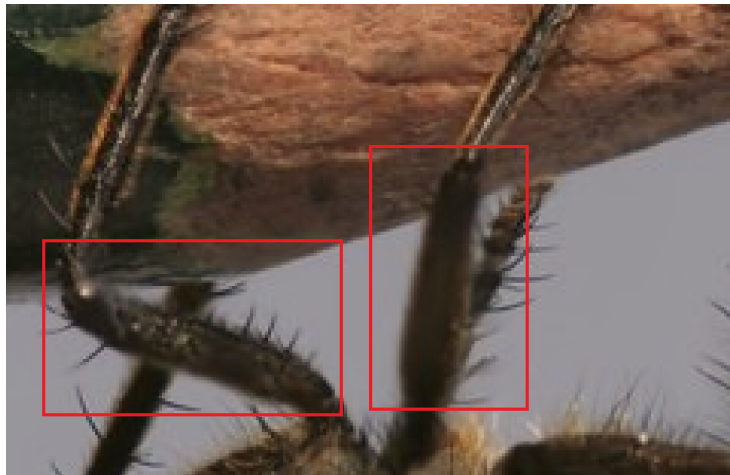


Abbildung 5.6: Endgültige Tiefenkarte und das fusionierte Bild unter Verwendung des Energiefunktionals, mit robustem Datenterm und kantenerhaltendem Glättigkeitsterm. Die rechte Spalte entstand nach der direkten Pixelfusion



(a)



(b)



(c)

Abbildung 5.7: Vergrößerung der Fusionierten Bilder aus den Abbildungen 5.4: (a), 5.5: (b) und 5.6: (c) für den Datenset *fly2*.

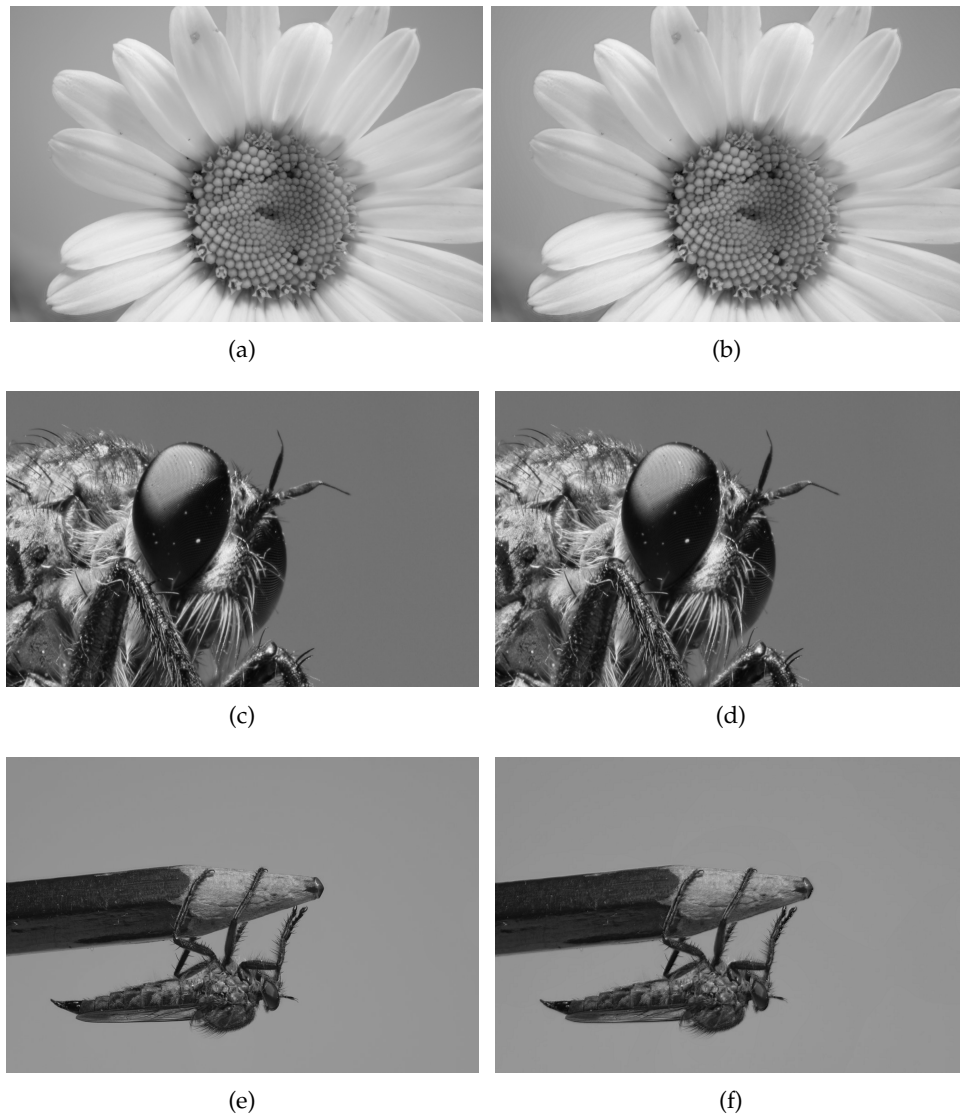
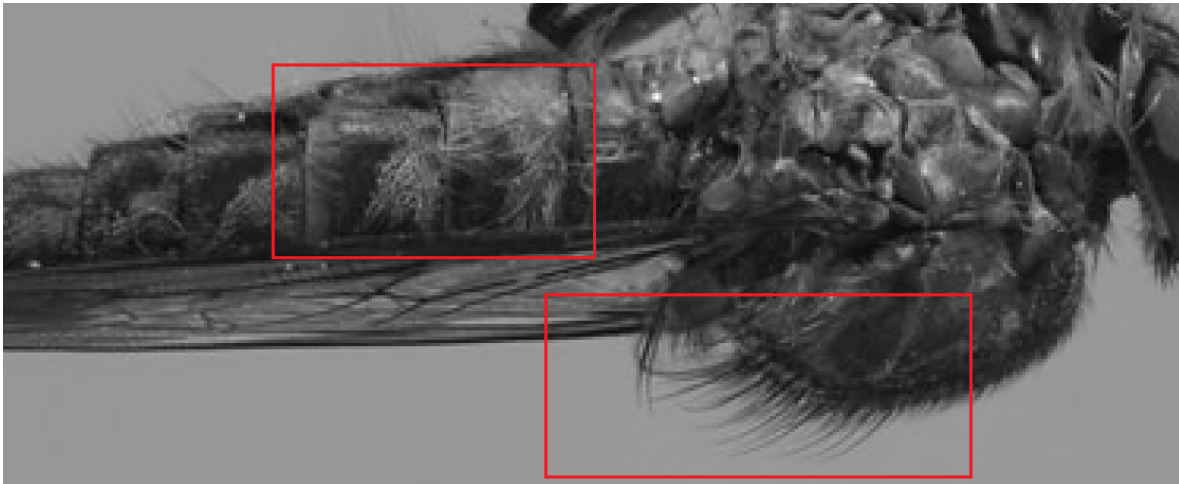


Abbildung 5.8: Vergleich der Fusionsverfahren. *Linke Spalte:* Direkte Pixelfusion. *Rechte Spalte:* Poisson Image Editing nach Perez *et.al* [16]. Die Ergebnisse wurden zwecks Vereinfachung der Implementierung als Graustufenbilder wiedergegeben.



(a)



(b)

Abbildung 5.9: Vergrößerung der fusionierten Bilder aus der Abbildung 5.7. (a): Direkte Pixelfusion (b): Poisson Image Editing.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde das Verfahren der Bildfusion von Boshtayeva *et.al* [4] nachimplementiert. Dabei haben wir drei Schärfekriterien auf Basis einer Ableitung verglichen und festgestellt, dass die Frobenius-Norm der Hesse-Matrix die besten Ergebnisse liefert. Die einzelnen Schärfewerte wurden einer Tiefe zugeordnet. Diese Tiefenkarte wurde anschließend anhand des Kriteriums evaluiert, dass benachbarte Werte gleiche Tiefen haben müssen. Als Konfidenzfunktion haben wir die Stufenfunktion aus [4] eingesetzt. Weil durch ihren Einsatz Tiefenwerte verloren gehen, haben wir eine Sigmoid-Konfidenzfunktion implementiert, um so den Informationsverlust zu minimieren. Die so entstandene Tiefenkarte haben wir mithilfe von sub-quadratischen Daten- und Glattheitsterme verbessert, indem wir ein Energiefunktional aufgestellt und es anschließend mithilfe der Euler-Lagrange-Gleichung minimiert haben. Die Bildfusion wurde als direkte Pixelfusion, sowie in Form des Poisson Image Editing implementiert. Beide Verfahren haben gleich gute Ergebnisse geliefert, mit der Ausnahme, dass Poisson Image Editing einige Bereiche leicht verwischt.

Ausblick

Der Einsatz einer Sigmoid-Konfidenzfunktion liefert bereits eine Verbesserung gegenüber der Vorgehensweise von [4]. Doch unsere Implementierung verwendet lediglich die isotrope Diffusion. Im Vergleich könnte die anisotrope Diffusion bessere Ergebnisse liefern. Zusätzlich haben wir uns in dieser Arbeit nur auf Pixel-basierte Verfahren konzentriert. Die Wirkung der Region-basierten Ansätze wurde hier nicht getestet und kann in den zukünftigen Arbeiten untersucht werden.

Literaturverzeichnis

- [1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, August 2004.
- [2] Oliver Barth and Markus Osthof. Project: Focal stack photography. Saarland University, 2010.
- [3] Madina Boshtayeva, David Hafner, and Joachim Weickert. A focus fusion framework with anisotropic depth map smoothing. University of Saarland, 2013.
- [4] Madina Boshtayeva, David Hafner, and Joachim Weickert. Focus fusion with anisotropic depth map smoothing. In Richard C. Wilson, Edwin R. Hancock, Adrian G. Bors, and William A. P. Smith, editors, *CAIP (2)*, volume 8048 of *Lecture Notes in Computer Science*, pages 67–74. Springer, 2013.
- [5] Andrés Bruhn. Correspondence problems in computer vision, summer term 2014. University of Stuttgart, 2014.
- [6] Andrés Bruhn, Martin Welk, and Joachim Weickert. Imaging science, summer term 2013. University of Stuttgart, 2013.
- [7] Jürgen Franke, Rainer Dahlhaus, Jörg Polzehl, Vladimir Spokoiny, Gabriele Steidl, Joachim Weickert, Anatoly Berdychevski, Stephan Didas, Siana Halim, Pavel Mrázek, SuhasiniSubba Rao, and Joseph Tadjuidje. Structural adaptive smoothing procedures. In Rainer Dahlhaus, Jürgen Kurths, Peter Maass, and Jens Timmer, editors, *Mathematical Methods in Signal Processing and Digital Image Analysis, Understanding Complex Systems*, pages 183–229. Springer Berlin Heidelberg, 2008.
- [8] Wei Huang and Zhongliang Jing. Evaluation of focus measures in multi-focus image fusion. *Pattern Recogn. Lett.*, 28(4):493–500, March 2007.
- [9] Jun Kong, Kaiyuan Zheng, and Jingbo Zhang. Multi-focus image fusion using spatial frequency and genetic algorithm. *International Journal of Computer Science and Network Security*, 8(2):220–224, February 2008.
- [10] Shutao Li and Bin Yang. Multifocus image fusion using region segmentation and spatial frequency. *Image Vision Comput.*, 26(7):971–979, July 2008.
- [11] Chen Long, Li Jinbo, and Chen C. L. Philip. Regional multifocus image fusion using sparse representation. *Opt. Express*, 21(4), February 2013.

- [12] Tibor Lukic. *Regularized Problems in Image Processing*. PhD thesis, University of Novi Sad, Novi Sad, 2011.
- [13] R Maruthi and K Sankarasubramanian. Multi focus image fusion based on the information level in the regions of the images. *Journal of Theoretical and Applied Information Technology*, 3(4):80–85, 2007.
- [14] Pavel Mrázek, Joachim Weickert, and Gabriele Steidl. Correspondences between wavelet shrinkage and nonlinear diffusion. In *Scale-Space*, pages 101–116, 2003.
- [15] Mila Nikolova. Energy minimization methods. In Otmar Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, pages 139–185. Springer New York, 2011.
- [16] Patrick Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003.
- [17] Maria Rudnaya and Robert Ochshorn. Sharpness functions for computational aesthetics and image sublimation. *IAENG International Journal of Computer Science*, 38(4):359–367, 2011.
- [18] Jamal Saeedi and Karim Faez. A classification and fuzzy-based approach for digital multi-focus image fusion. *Pattern Analysis and Applications*, 16(3):365–379, 2013.
- [19] Jing Tian and Li Chen. Multi-focus image fusion using wavelet-domain statistics. In *ICIP*, pages 1205–1208. IEEE, 2010.

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