Experimentelle Umsetzung eines mit Deep Learning unterstütztem Depth-from-Focus-Mikroskops

Masterarbeit
zur Erlangung des Grades Master of Science

Abgegeben von: Alina Malow
Matrikelnr.: 410591
Studiengang: Physikalische Ingenieurwissenschaft
E-Mail: alina.malow@campus.tu-berlin.de

Abgabedatum: 05.05.2025

Erster Prüfer: Prof. Dr.-Ing. Jörg Krüger

Zweite Prüferin: Iryna Shevchenko, M.Sc.

Betreuer: Conor Ryan, M.Sc.
Deutsches Zentrum für Luft- und Raumfahrt
Institut für Optische Sensorsysteme
Abteilung Weltrauminstrumente

Technische Universität Berlin
Fakultät V - Verkehrs- und Maschinensysteme
Institut für Werkzeugmaschinen und Fabrikbetrieb
Industrielle Automatisierungstechnik

Straße des 17. Juni 135, 10623 Berlin

Abstract

(English version below)

Im Rahmen dieser Arbeit wurde ein experimentelles 3D-Mikroskop um ein Deep-Learning-gestütztes Depth-from-Focus (DDFF)-Verfahren erweitert, um die Tiefenmessung geologischer Proben ohne zusätzliche optische Komponenten zu ermöglichen. Bei der Depth-from-Focus (DFF)-Methode wird die Schärfe mehrerer Aufnahmen mit variierendem Fokusabstand analysiert, um daraus Rückschlüsse auf die Objektentfernung zu ziehen. Zwei Convolutional Neural Networks wurden basierend auf einer systematischen Literaturrecherche ausgewählt, implementiert und mit synthetisch generierten unscharfen Bildern aus einem großen RGB-D-Datensatz neu trainiert. Die realitätsnahe Simulation der Unschärfe erfolgte unter Berücksichtigung der Kameraparameter des Mikroskops. Zur Validierung wurde der eigens erhobene GeoFocus3D-Datensatz herangezogen, bestehend aus realen Mikroskopaufnahmen und Ground-Truth-Tiefenkarten eines Streifenprojektionssystems. Die neu trainierten Modelle zeigten eine höhere Genauigkeit als die vortrainierten Varianten, konnten jedoch eine klassische DFF-Methode nicht übertreffen, die bei ausreichender Anzahl unscharfer Aufnahmen die präzisesten Ergebnisse erzielte. Die Analyse offenbarte zudem Grenzen neuronaler Verfahren hinsichtlich Generalisierbarkeit, Texturabhängigkeit und Störanfälligkeit gegenüber Rauschen. Ergänzend wurde ein multispektrales Beleuchtungssystem entwickelt und in den Aufbau integriert. Die Ergebnisse belegen, dass DDFF-Methoden – insbesondere in ressourcenbeschränkten Szenarien – eine effiziente Alternative darstellen, deren Genauigkeit jedoch maßgeblich von der Qualität der Trainingsdaten und der Modellierung der Bildschärfe abhängt.

In this thesis, an experimental 3D microscope was extended with a deep learning-based Depth-from-Focus (DDFF) approach to enable depth measurement of geological samples without requiring additional optical components. Depth-from-Focus (DFF) is a technique that estimates object depth by analyzing the sharpness of a series of images captured at varying focus settings. Two convolutional neural network models were selected based on a systematic literature review, implemented, and retrained using synthetically generated focal stacks from a large RGB-D dataset. The defocus effect was realistically simulated using the actual camera parameters of the microscope. For validation, a custom dataset (GeoFocus3D) was created, consisting of real microscope images and ground-truth depth maps generated by a structured-light microscope. The retrained models achieved higher accuracy than their pretrained counterparts, but a classical DFF method still outperformed them when dense focal stacks were available. The evaluation also revealed limitations of neural approaches regarding generalization, dependence on texture, and sensitivity to noise. Additionally, a multispectral illumination system was developed and integrated into the microscope setup. The results demonstrate that DDFF methods offer an efficient alternative for use in resource-constrained environments, although their performance strongly depends on the quality of training data and the realism of the defocus simulation.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

AiF All-in-Focus X, 5, 14, 15, 16, 19, 21, 22, 37, 40, 41, 43, 67

CAD Computer Aided Design 14, 15, 21, 22, 64, 67

CNN Convolutional Neural Network 1, 2, 3, 4, 8, 9, 10, 12, 13, 14, 16, 17, 18, 22, 48, 49, 50, 62, 67, 70

CoC Circle of Confusion 5, 26, 41, 42, 43, 64

DDFF Deep-Depth-from-Focus 8, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 40, 41, 42, 48, 52, 60, 63, 65, 66, 67, 68, 70

DFD Depth-from-Defocus 8, 15

DFF Depth-from-Focus II, III, 1, 2, 3, 4, 8, 9, 13, 14, 21, 22, 23, 28, 31, 34, 38, 39, 41, 44, 48, 55, 56, 59, 60, 61, 62, 64, 65, 66, 67, 68, 69

DFV Differential Focus Volume 50

DoF Depth-of-Field 4, 5

EFD Effective Downsampling 52, 53

FDM Fused Deposition Modelling 36

FOV Field-of-View 6, 36, 52

GPU Graphics Processing Unit 43, 51, 54

MAE Mean Absolute Error 11, 58, 64

MLS Moving-Least-Squares 47, 59

MSE Mean Squared Error 11, 48, 58

PLA Polylactid 36

PSF Point Spread Function 5, 14, 26, 41, 43, 69

RDF Ring Difference Filter 18, 44, 45

RMSE Root Mean Squared Error 58

SIFT Scale Invariant Feature Transform 55

SRD Sharp Region Detection 52, 53

VDFF Variational Depth-from-Focus 14, 15, 16, 18

WLLSR Windowed Linear Least Squares Regression 56, 59, 60, 61, 62, 64

Formelzeichen

$b_{f,0}$	Abstand zwischen der idealisierten Linse und dem Detektor bei Aktuatorposition $z = 0$
(O_c, x_c, y_c, z_c)	Kamerakoordinatensystem
(O_w, x_w, y_w, z_w)	Weltkoordinatensystem
(p_x, p_y)	Bildhauptpunkt
α	Strahlwinkel
$\bar{f}$	idealisierter Fokuswertverlauf
β	Beleuchtungswinkel
δ	Verzeichnung
ϵ	bivariantes Polynom
γ	freier Parameter der Smooth-L1-Verlustfunktion
κ	detektierte Keypoints in berechneter Tiefenkarte
κ^*	detektierte Keypoints in Ground-Truth-Tiefenkarte
$\mathbf{R}$	Rotationsmatrix
$\mathbf{S}$	Skalierungsmatrix
$\mathbf{T}$	Translationsvektor
$\mathbf{u_c}$	Objektpunkte im Kamerakoordinatensystem
$\mathbf{u}$	Objektpunkte im Weltkoordinatensystem
$\mathbf{y}$	Bildpunkte im Bildkoordinatensystem
$\mathbf{y}_{\text{ideal}}$	verzeichnungsfreie Bildpunkte
$\mathbf{y}_{\text{verzerrt}}$	verzeichnete Bildpunkte
ssi_trim	skalierungs- und translationsinvarianter Fehler
μ	Erwartungswert
ρ	Umgebung eines Pixels
σ	Standardabweichung

Θ	Gewichtsfunktion
θ	Rotationswinkel
$\tilde{D}$	skalierungs- und translationsfreie berechnete Tiefenkarte
$\tilde{D}^*$	skalierungs- und translationsfreie Ground-Truth-Tiefenkarte
$\tilde{s}$	geschätzter Skalierungsfaktor
$\tilde{t}$	geschätzte Translation
A	Durchmesser Blendenöffnung
a	mittlere absolute Abweichung
B	Bildpunkt
b	Bildweite
c	Steigung der Dreiecksfunktion
D	berechnete Tiefenkarte
d	Objekttiefe
D^*	Ground-Truth-Tiefenkarte
E	Energieterm
E_0	Abstand zwischen den Objektiven
f	Brennweite
f_I	Brennweite der imaginären idealen Linse
f_k	Fokuswertverlauf
G	Objektpunkt
g	Objektweite
G_f	fokussierter Objektpunkt
g_f	Fokusabstand
h	Filterradius des MLS-Filters
I'	unscharfes Bild
$I(x, y)$	All-in-Focus (AiF)-Bild

K	Anzahl der Biler im Fokalstapel
k	Nummerierung der Bilder im Fokalstapel
k_1, k_2, k_3	radiale Verzeichnungskoeffizienten
L	Laplace-Verteilung
l	Anzahl detektierter Keypoints
L_1	L1-Verlustfunktion
L_2	L2-Verlustfunktion
$L_{1,smooth}$	Smooth-L1-Verlustfunktion
M	Vergrößerung
m	Anzahl der Gitterpunkte des Kalibrierungsmusters
M_T	Transformationsmodell
N	Anzahl gültiger Pixel
n	Nummerierung der Kalibrierungsaufnahme
n_w	Fenstergröße
P	Wahrscheinlichkeitsvolumen
p, q	Schnittpunkte der Dreiecksfunktion mit der Ordinate
R	Abstand zwischen LEDs und optischer Achse
r	Abstand Bildpunkt zum Bildhauptpunkt
r_1	Radius der Scheibe des RDF-Filters
r_2	innerer Radius des Rings des RDF-Filters
r_3	äußerer Radius des Rings des RDF-Filters
R_B	Radius der von den LEDs beleuchteten Fläche
S	Anzahl der Skalen
s	Skalierungsfaktor
$sc\text{-}inv$	skalierungsunabhängiger Fehler
T	Dreiecksfunktion

TE	Transformationsfehler
U	Unsicherheitskarte
v_x	Translation in vertikale Richtung
v_y	Translation in horizontale Richtung
w	Loss-Gewichte
W_1, W_2	Teilintervalle
X	Parametervektor
x	Vertikale Pixelposition
x_w, y_w, z_w	Koordinaten des Objektpunkts im Weltkoordinatensystem
y	Horizontale Pixelposition
z	Aktuatorposition
f_o	Brennweite der Objektive
CoC	Durchmesser des Circle of Confusion
FOV	Größe des Bildfelds
PSF	Point Spread Function
WD	Arbeitsabstand

1 Einleitung

Für die Erforschung extraterrestrischer Himmelskörper kommen häufig Rover zum Einsatz. Am Institut für Optische Sensorsysteme des Deutschen Zentrums für Luft- und Raumfahrt (DLR) wird derzeit ein kombiniertes 3D-Mikroskop mit Raman-Spektrometer als Proof-of-Concept für zukünftige Rovermissionen entwickelt. Eine schematische Darstellung des Aufbaus ist in Abbildung 1 dargestellt. Das Raman-Spektrometer ermöglicht die Analyse der chemischen Zusammensetzung von Gesteinen, während das 3D-Mikroskop ergänzende Kontextinformationen sowie Daten zur Morphologie der Proben liefert. Dadurch lassen sich Rückschlüsse auf die Entstehungsprozesse, geologische Geschichte und Zusammensetzung extraterrestrischer Oberflächen ziehen (Ryan u. a. 2024).

Die Tiefeninformation wird mit dem 3D-Mikroskop durch Streifenprojektion erfasst. Dabei wird ein definiertes Muster, meist in Form von sinusförmigen oder rechteckigen Streifen, von einem Projektor auf das zu vermessende Objekt projiziert. Eine Kamera erfasst anschließend die Verzerrung dieses Musters auf der Objektoberfläche. Durch die Auswertung dieser Verzerrungen mithilfe triangulationsbasierter Berechnungen kann die exakte dreidimensionale Form des Objekts rekonstruiert werden.

Diese Technik ist jedoch nicht in allen Anwendungsszenarien zuverlässig. Scharfwinklige Vertiefungen oder Kanten können Schatten erzeugen, wodurch in den verdeckten Bereichen keine vollständige Tiefenmessung möglich ist.

In solchen Fällen bietet die Depth-from-Focus (DFF)-Methode eine vielversprechende Alternative. Bei diesem Verfahren wird die Tiefeninformation durch Analyse der Bildschärfe ermittelt. Durch sukzessive Änderung des Fokusabstands werden unterschiedliche Ebenen des Objekts fokussiert. Der Abstand des jeweils am besten fokussierten Pixels erlaubt Rückschlüsse auf die Tiefe. Da hierfür keine zusätzliche Hardware notwendig ist, lässt sich ein bestehendes Kamerasystem durch geeignete Softwarelösungen zur DFF-Methode erweitern. Besonders für weltraumtaugliche Systeme, bei denen Masse und Volumen begrenzt sind, stellt DFF eine attraktive Lösung dar.

Ziel dieser Arbeit ist die Erweiterung eines bestehenden experimentellen 3D-Mikroskops um eine DFF-Funktion. Eine Herausforderung besteht dabei in der zuverlässigen Bestimmung der Bildschärfe, insbesondere bei schwach texturierten Oberflächen. Um die Robustheit des Verfahrens zu erhöhen, wird ein Convolutional Neural Network (CNN) eingesetzt. CNNs ermöglichen durch ihre Faltungsoperationen die Extraktion relevanter Bildmerkmale, einschließlich Schärfeinformationen. Durch das Training auf umfangreichen Datensätzen können CNNs lernen, auch in texturarmen Regionen zuverlässige Schärfeabschätzungen vorzunehmen (Hazirbas u. a. 2019).

Der Einsatz von CNNs im Rahmen der DFF-Methode bietet darüber hinaus weitere Vorteile: So kann die notwendige Anzahl an Aufnahmen reduziert werden, wodurch sowohl die Aufnahmezeit als auch der Speicherbedarf verringert werden. Zudem sinkt der Energieaufwand für die Durchführung der Messungen, was insbesondere für weltraumbasierte Systeme von großer Bedeutung ist. Erfolgt

die Auswertung der Rover-Daten auf der Erde, so verringert sich durch die reduzierte Bildanzahl auch das zu übertragende Datenvolumen. Darüber hinaus ist die Berechnung der Tiefenkarten mit einem trainierten CNN weniger zeit- und rechenintensiv (Fujimura u. a. 2024; Hazirbas u. a. 2019).

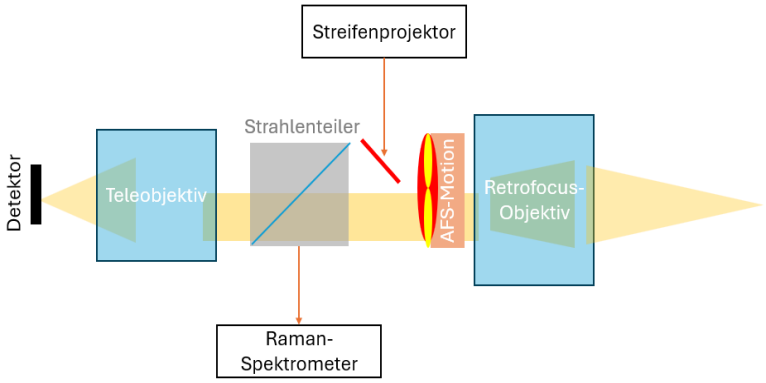


Abbildung 1: Schematische Darstellung des kombinierten 3D-Mikroskops mir Raman-Spektrometer. AFS - Auto-Focusing-System

Durch die Integration von DFF in das 3D-Mikroskop wird eine zuverlässige Tiefenmessung verschiedenster Proben ermöglicht. Neben der Anwendung in der Weltraumforschung erscheint der Einsatz auch in der industriellen Qualitätskontrolle denkbar (Matsubara u. a. 2022). Die Erfassung von Tiefeninformationen ist mit handelsüblichen Kamerasystemen möglich, ohne dass zusätzliche optische Komponenten erforderlich sind. Gleichzeitig werden durch den Einsatz von CNNs sowohl Rechenzeit als auch Speicherressourcen eingespart.

Im Rahmen dieser Arbeit erfolgte zunächst eine Recherche bestehender CNN-basierte DFF-Verfahren. Basierend auf der Literaturrecherche wurden zwei geeignete Verfahren ausgewählt und für das 3D-Mikroskop implementiert. Zur Bewertung der Verfahren wurde außerdem eine konventionelle DFF-Methode ohne Deep Learning hinzugezogen.

Für die Validierung der Methoden wurden zwei Datensätze herangezogen. Mit dem experimentellen 3D-Mikroskop wurden Gesteinsproben vermessen, deren Tiefenkarten mit einem kommerziellen Streifenprojektionsmikroskop erstellt wurden. Dieser Datensatz wird im Folgenden als GeoFocus3D bezeichnet. Zusätzlich kam der öffentlich verfügbare Micro-Topo-Datensatz (Siemens, Kästner und Reithmeier 2023) zum Einsatz. Da dieser lediglich Tiefenkarten und vollständig fokussierte Bilder enthält, wurden mittels Computersimulation synthetisch unscharfe Bilder generiert. Der Micro-Topo-Datensatz diente darüber hinaus auch als Trainingsgrundlage für die CNN-basierten Modelle.

Zur Erweiterung des experimentellen Aufbaus wurde außerdem ein multispektraler Beleuchtungsring integriert, um spektrale Informationen zu erfassen und somit den Informationsgehalt der

Messungen zu erhöhen. Zudem erfolgte die systematische Charakterisierung relevanter Komponenten des Mikroskops.

Der Aufbau dieser Arbeit gestaltet sich wie folgt: Kapitel 2 behandelt die optischen Grundlagen, die DFF-Methode sowie den Einsatz von CNNs in der Bildverarbeitung. Kapitel 3 gibt einen Überblick über den Stand der Technik im Bereich CNN-basierter DFF-Verfahren. In Kapitel 4 wird das konzeptionelle Vorgehen zur Integration der Deep-Learning-gestützten DFF-Methoden vorgestellt. Kapitel 5 beschreibt die konkrete Umsetzung der Verfahren, einschließlich des experimentellen Aufbaus, dessen Charakterisierung sowie der Integration der multispektralen Beleuchtung. Die experimentelle Evaluation und der Vergleich der Verfahren werden in Kapitel 6 präsentiert. Kapitel 7 schließt die Arbeit mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf zukünftige Entwicklungen ab.

2 Grundlagen

Im folgenden Kapitel werden die theoretischen Grundlagen dargestellt, die für das Verständnis der Umsetzung des Deep-Learning-gestützten DFF-Mikroskops von Bedeutung sind. Da viele DFF-Verfahren auf dem Dünnlinsenmodell basieren, werden zunächst die optischen Grundlagen des Dünnlinsenmodells erläutert. Außerdem wird die Modellierung der Bildschärfe erklärt, welche die Grundlage für die synthetische Erzeugung unscharfer Bilder im Rahmen der Datensatzerstellung bildet.

Anschließend wird die grundlegende Vorgehensweise einer geometrischen Kamerakalibrierung beschrieben. Diese ist insbesondere erforderlich, um mit dem experimentellen Mikroskop möglichst verzeichnungsfreie Aufnahmen zu erhalten. Im Rahmen der Kalibrierung werden daher bestehende Abbildungsfehler identifiziert.

Daraufhin werden zentrale Begriffe im Zusammenhang mit der DFF-Methode definiert sowie die Funktionsweise konventioneller, nicht auf Deep Learning basierender DFF-Verfahren beschrieben. Zudem erfolgt ein kurzer Vergleich der Vor- und Nachteile von DFF im Verhältnis zu anderen optischen Verfahren zur Tiefenmessung.

Zum Abschluss werden die fundamentalen Konzepte von CNNs präsentiert, mit besonderem Fokus auf deren Architektur und Training. Zusätzlich zu den üblichen Verlustfunktionen werden die potenziellen Vorteile der Anwendung von CNNs im Kontext der Bildverarbeitung erörtert, ebenso wie die Herausforderungen, die deren Implementierung mit sich bringen kann.

2.1 Modellierung der Tiefenschärfe auf Basis des Dünnlinsenmodells

Viele DFF-Verfahren basieren auf dem sogenannten Dünnlinsenmodell (Gur und Wolf 2019). Dabei wird die komplexe Optik eines Kamerasystems auf eine einzelne, idealisierte dünne Linse reduziert. Die Dicke der Linse wird dabei vernachlässigt und die Lichtbrechung als auf eine Ebene beschränkt angenommen.

Das Prinzip der Bildaufnahme eines fokussierten Objektpunkts G_f und einem nicht-fokussierten Objektpunkt G basierend auf diesem Modell ist in der Abbildung 2 dargestellt. Der Zusammenhang zwischen der Brennweite f der Linse, dem Objektabstand g und dem Bildabstand b ergibt sich gemäß der vereinfachten Linsengleichung:

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b} \tag{1}$$

Liegt der Bildpunkt B exakt auf der Detektorebene, wird der Objektpunkt scharf abgebildet. Der entsprechende Abstand des Objekts wird dann als Fokusabstand g_f bezeichnet. Weicht b vom Abstand zwischen Linse und Detektor ab, so entsteht ein unscharfes Bild des Punktes.

In Kamerasystemen wird die Schärfentiefe Depth-of-Field (DoF), definiert als jener Bereich im Raum, in dem Objektpunkte für das menschliche Auge als scharf wahrgenommen werden. Wird ein

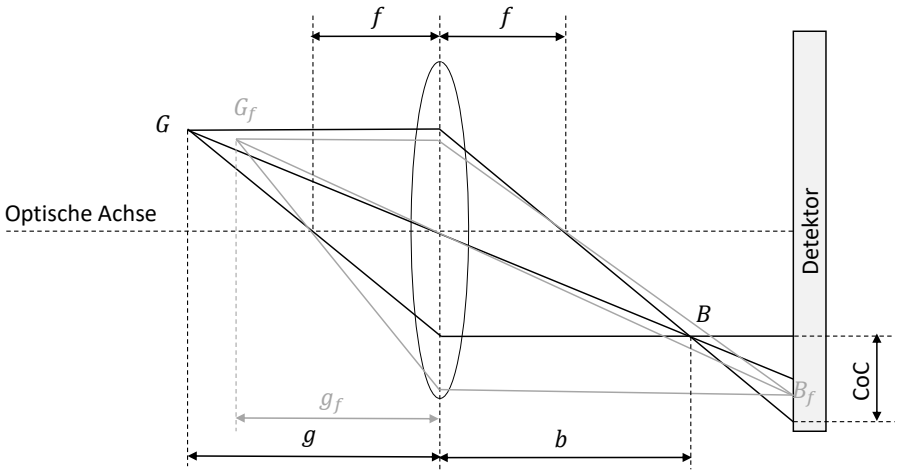


Abbildung 2: Prinzip des Dünnlinsenmodells in Anlehnung an Pertuz, Puig und Garcia (2013)

Bild mit einem Kamerasystem aufgenommen, das über eine ausreichend große Schärfentiefe verfügt – etwa durch die Wahl einer kleinen Blendenöffnung –, so entsteht ein nahezu vollständig scharfes Bild, ein sogenanntes All-in-Focus (AiF)-Bild, bei dem sämtliche Bildbereiche im Fokus liegen. Alternativ kann ein AiF-Bild auch rechnerisch erzeugt werden, indem die jeweils scharfen Bereiche mehrerer Aufnahmen mit unterschiedlichen Fokusabständen zu einem einzigen, durchgehend scharfen Bild zusammengesetzt werden.

Punkte außerhalb des DoF unterliegen dem sogenannten Defokus-Effekt, welcher mithilfe der Point Spread Function (PSF) beschrieben werden kann. Die PSF charakterisiert, wie ein optisches System eine punktförmige Lichtquelle abbildet.

Eine ideale, scheibenförmige PSF kann durch eine Gauß-Funktion angenähert werden. Das resultierende unscharfe Bild $I'(x, y)$ ergibt sich durch die Faltung des AiF-Bilds I mit der PSF. Das unscharfe Bild berechnet sich zu:

$$I'(x, y) = I(x, y) \circ \mathrm{PSF}(x, y) \tag{2}$$

$$\mathrm{PSF}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{3}$$

Eine vereinfachte Simulation der Unschärfe kann durch eine Disk-Funktion erfolgen. Der Durchmesser dieser Scheibe wird als Circle of Confusion (CoC) bezeichnet und berechnet sich mit dem Durchmesser der Blendenöffnung A wie folgt:

$$\mathrm{CoC} = A \cdot \left|\frac{g - g_f}{g}\right| \cdot \frac{f}{g_f - f} \tag{4}$$

Der abgebildete Objektbereich auf dem Sensor wird als Field-of-View (FOV) bezeichnet. Dieser ergibt sich aus der Sensorgröße und der Vergrößerung M gemäß:

$$\text{FOV} = \frac{\text{Sensorgröße}}{M} \tag{5}$$

$$M = \frac{b}{g} \tag{6}$$

2.2 Geometrische Kamerakalibrierung

Für quantitative Messungen, wie etwa der Tiefenbestimmung, ist eine präzise geometrische Kamerakalibrierung unerlässlich (Weng, Cohen und Herniou 1992). Ziel dieser Kalibrierung ist die Bestimmung der intrinsischen Kameraparameter, welche die Abbildung von Objektpunkten auf dem Kamerasensor mathematisch beschreiben. Zusätzlich werden extrinsische Kameraparameter bestimmt, die den geometrischen Zusammenhang zwischen Kamera und Objekt definieren.

In realen Kameras treten diverse Abbildungsfehler auf, von denen geometrische Verzeichnungen für geometrische Messungen besonders relevant sind. Zur Ermittlung der Kameraparameter und zur Modellierung der Verzeichnung wird häufig ein nichtlineares Optimierungsverfahren verwendet. Hierbei werden Gleichungen formuliert, die die Abbildung von Objektpunkten auf Bildpunkte beschreiben. Ein iterativer Lösungsansatz minimiert die Abweichungen zwischen den gemessenen und berechneten Bildpunktpositionen.

Für eine ideale, verzeichnungsfreie Kamera kann die Abbildung mit dem Lochkameramodell beschrieben werden. Ein Objektpunkt $\mathbf{u}$ im Weltkoordinatensystem (O_w, x_w, y_w, z_w) wird durch die extrinsischen Parameter, bestehend aus Rotationsmatrix $\mathbf{R}$ und Translationsvektor $\mathbf{T}$, in das Kamerakoordinatensystem (O_c, x_c, y_c, z_c) transformiert:

$$\mathbf{u_c} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \mathbf{R} \cdot \mathbf{u} + \mathbf{T} \tag{7}$$

Der zugehörige Bildpunkt $\mathbf{y}$ in Bildkoordinaten (x, y) ergibt sich anschließend unter Berücksichtigung der intrinsischen Kameraparameter Brennweite f und Bildhauptpunkt (p_x, p_y) durch:

$$\mathbf{y} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \cdot \frac{x_c}{z_c} + p_x \\ f \cdot \frac{x_c}{z_c} + p_y \end{bmatrix} \tag{8}$$

Um die Koordinaten des Bildpunkts im digitalen Bildraster anzugeben, erfolgt eine zusätzliche Umrechnung unter Berücksichtigung des Pixelgröße. Eine Übersicht zu den genannten Koordinatensystemen ist in der Abbildung 3 dargestellt.

In der Praxis weichen reale Kameras aufgrund von Verzeichnungen von diesem vereinfachten Modell ab. Solche Verzeichnungen können durch Fertigungstoleranzen und Unvollkommenheiten in der Linsengeometrie entstehen. Wenn $\mathbf{y}_{\text{ideal}}$ die verzerrungsfreie Position des Bildpunkts ist,

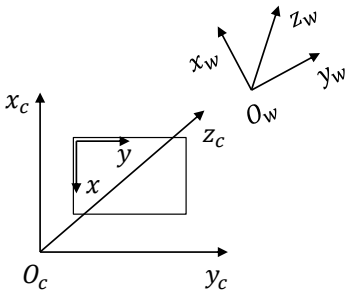


Abbildung 3: Koordinatensysteme bei der Bildaufnahme in Anlehnung an Weng, Cohen und Herniou (1992)

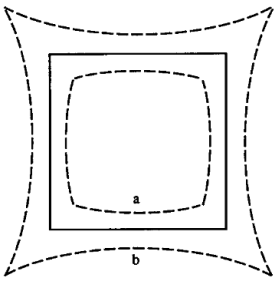


Abbildung 4: Effekt der radialen Verzeichnung. Durchgezogene Linie: keine Verzeichung; a: negative Verzeichung; b: positive Verzeichnung (Weng, Cohen und Herniou 1992)

berechnet sich der tatsächlich beobachtete Bildpunkt $\mathbf{y}_{\text{verzerrt}}$ unter Hinzunahme einer Verzeichnungskomponente δ wie folgt:

$$\mathbf{y}_{\text{verzerrt}} = \mathbf{y}_{\text{ideal}} + \delta \tag{9}$$

Eine häufige Form der Verzeichnung ist die radiale Verzeichnung, die durch eine ungleichmäßige Krümmung der Linsen verursacht wird. Diese führt dazu, dass Bildpunkte entweder zum Bildhauptpunkt hin (negativ, Barrel Distortion) oder von diesem weg (positiv, Pincushion Distortion) verschoben werden (siehe Abbildung 4). Mathematisch wird die radiale Verzeichnung durch:

$$\delta = (x - p_x,\, y - p_y) \cdot \left(k_1 r^2 + k_2 r^4 + k_3 r^6\right) \tag{10}$$

beschrieben, wobei r der Abstand des Punktes zum Bildhauptpunkt ist und k_1, k_2, k_3 die radialen Verzeichnungskoeffizienten sind.

Zur Kalibrierung der Kamera, das heißt zur Bestimmung der intrinsischen und extrinsischen Parameter sowie der Verzeichnungsparameter, wird eine hinreichende Anzahl an Objektpunkten und zugehörigen Bildpunkten benötigt. Ist es möglich, eigene Aufnahmen durchzuführen, so kann ein Kalibrierungsmuster verwendet werden. Die Kalibrierung erfolgt dabei in den folgenden Schritten:

1. **Bilderfassung:** Das Kalibrierungsmuster wird aus verschiedenen Perspektiven aufgenommen. Dabei werden sowohl unterschiedliche Abstände als auch verkippte Ausrichtungen zwischen Muster und Kamera verwendet. Das Muster enthält geometrisch bekannte Punkte, die Musterkennpunkte genannt werden.

2. **Detektion der Musterkennpunkte:** Mithilfe bildverarbeitender Algorithmen werden die Positionen der Musterkennpunkte in den aufgenommenen Bildern erkannt. Diese Positionen entsprechen den Bildpunkten.

3. **Berechnung der Transformation:** Die Transformation zwischen den bekannten Objektpunkten und den erkannten Bildpunkten wird modelliert. Dabei entsteht ein überbestimmtes Gleichungssystem, das durch Optimierungsverfahren gelöst wird. Die extrinsischen Parameter geben dabei die Position und Orientierung des Musters in jeder Aufnahme an.

Die ermittelten Transformationsparameter können schließlich dazu genutzt werden die verzerrten Bilder in ein verzeichnungsfreies Koordinatensystem zu übertragen.

2.3 Grundlagen der DFF-Methode

Um die Funktionsweise moderner, auf Deep Learning basierender -Verfahren nachvollziehen zu können, ist ein grundlegendes Verständnis der klassischen DFF-Methodik erforderlich. In diesem Abschnitt werden daher zentrale Begriffe eingeordnet, das Funktionsprinzip konventioneller DFF-Verfahren erläutert sowie deren Stärken und Schwächen dargestellt.

2.3.1 Begriffseinordnung

DFF ist eine bildbasierte Methode zur Tiefenschätzung, bei der eine Serie von Aufnahmen desselben Objekts mit variierenden Fokuseinstellungen verwendet wird. Diese Bildserie wird im Folgenden als Fokalstapel bezeichnet. Ziel ist es, für jeden Bildpunkt diejenige Aufnahme zu identifizieren, in der dieser Punkt am schärfsten erscheint, um daraus die Tiefe zu rekonstruieren.

In der Literatur wird DFF teilweise synonym mit Shape-from-Focus verwendet. Ein verwandtes Verfahren ist außerdem Depth-from-Defocus (DFD), welches auf der Auswertung eines einzelnen unscharfen Bildes basiert. Letzteres wird in dieser Arbeit jedoch nicht weiter behandelt.

In den letzten Jahren haben CNNs zunehmend Einzug in die DFF-Verfahren gehalten. Sie ermöglichen durch hierarchische Faltungsschichten die automatische Extraktion relevanter Bildmerkmale, so auch von Schärfeinformationen aus dem Fokalstapel. DFF-Methoden, die auf CNNs basieren, werden im Folgenden als Deep-Depth-from-Focus (DDFF) bezeichnet. Der Begriff wurde vermutlich erstmals von Hazirbas u. a. (2019) verwendet.

2.3.2 Funktionsprinzip konventioneller DFF-Methoden

Konventionelle DFF-Methoden bestimmen für jeden Pixel eines Fokalstapels einen Fokuswert, der die lokale Bildschärfe quantifiziert. Anschließend wird das Bild ermittelt, in dem der Fokuswert für den jeweiligen Pixel maximal ist. Aufgrund der bekannten Fokusabstände jedes Bildes kann daraus über die vereinfachte Linsengleichung die Entfernung des Objektpunkts berechnet werden.

Eine Übersicht zu verschiedenen Fokuswert-Operatoren wurde von Pertuz, Puig und Garcia (2013) zusammengestellt. Fokuswerte können beispielsweise mit Gradienten- oder Laplace-Filtern ermittelt werden. Diese Operatoren basieren auf der Annahme, dass in fokussierten Bildbereichen stärkere Kanten vorhanden sind. Andere Fokuswert-Operatoren verwenden aber zum Beispiel auch Wavelet-Transformationen oder statistische Kenngrößen wie Varianz oder Entropie.

Anstelle des globalen Maximums des Fokuswertverlaufs wird oft eine kontinuierliche Interpolation verwendet, etwa durch Anpassung einer Gaußverteilung (Pertuz, Puig und Garcia 2013), um eine präzisere Tiefenschätzung zu ermöglichen. Alternativ wurden auch Laplace-Verteilungen (Cou und Guennebaud 2024) oder Polynome (Subbarao und T. Choi 1995) zur Modellierung des Verlaufs untersucht.

2.3.3 Vorteile und Einschränkungen konventioneller DFF-Methoden

Konventionelle DFF-Methoden bieten den Vorteil, dass sie lediglich ein einzelnes Kamerasystem erfordern und somit ohne zusätzliche Komponenten auskommen. Sie eignen sich besonders für Anwendungen im Nahbereich, in denen andere Verfahren, zum Beispiel Stereosysteme, nicht zuverlässig arbeiten.

Gleichzeitig sind mit dem Einsatz klassischer DFF-Verfahren auch mehrere Einschränkungen verbunden. Um einen ausreichend klaren Fokuswertverlauf zu erhalten, sind in der Regel viele Aufnahmen mit unterschiedlichen Fokusabständen erforderlich. Dies führt zu einem hohen Speicherbedarf und macht die anschließende Tiefenberechnung rechenintensiv, da für jeden Bildpunkt Fokuswerte über alle Aufnahmen hinweg bestimmt werden müssen. Die Genauigkeit der Tiefenschätzung kann zudem durch geringe Bildtextur oder schwachen Kontrast erheblich beeinträchtigt werden. Ein weiteres Problem tritt an Kanten zwischen Bildbereichen mit stark unterschiedlichen Helligkeiten auf: Hier können unscharfe Pixel durch hohe Gradienten irrtümlich als fokussiert erkannt werden, was zu lokalen Fehlmessungen in der Tiefenkarte führt.

2.4 Convolutional Neural Networks (CNNs) in der Bildverarbeitung

CNNs zeichnen sich in vielen Bildverarbeitungsanwendungen durch ihre hohe Leistungsfähigkeit aus. Sie gehören zur Klasse der Deep-Learning-Modelle. Deep Learning wiederum ist ein Teilbereich des maschinellen Lernens, der auf tiefen neuronalen Netzen basiert, um komplexe Muster und Zusammenhänge in großen Datenmengen zu erkennen.

2.4.1 Architektur von CNNs

In den Convolutional Layers werden durch Faltungen charakteristische Bildmerkmale extrahiert und in sogenannten Feature-Maps gespeichert. Je nach Struktur der Eingabedaten können dabei zwei- oder dreidimensionale Faltungen zum Einsatz kommen. Während 2D-Faltungen typischerweise zur Analyse von Bilddaten verwendet werden – etwa zur Objekterkennung oder Klassifikation –, eignen sich 3D-Faltungen insbesondere für Aufgaben, bei denen eine zusätzliche Dimension (z.B. Zeit oder Tiefe) berücksichtigt werden muss, wie etwa in der Videoanalyse oder 3D-Objekterkennung.

Im Anschluss an die Convolutional Layers werden häufig nichtlineare Aktivierungsfunktionen wie die Rectified Linear Unit verwendet, um Nichtlinearitäten zu modellieren und somit die Lernfähigkeit des Netzwerks zu verbessern. Pooling-Schichten reduzieren anschließend die Dimensionalität

der Feature-Maps, was zu einer Komprimierung der Informationen führt und gleichzeitig die Rechenkosten verringert.

Für Klassifikationsaufgaben werden CNNs typischerweise um Fully Connected Layers ergänzt, die die extrahierten Merkmale zu einer abschließenden Entscheidung zusammenführen (Mascarenhas und Agarwal 2021; Simonyan und Zisserman 2015). Bei Regressionsaufgaben wie der Tiefenschätzung kommen hingegen Regressionsschichten zum Einsatz (Hazirbas u. a. 2019).

Eine häufig verwendete Netzwerkarchitektur stellt der Convolutional Autoencoder dar. Dieser besteht aus einem Encoder und einem Decoder: Der Encoder reduziert mithilfe von Convolutional Layers die Dimension der Eingangsdaten, während der Decoder – meist als gespiegelte Architektur des Encoders aufgebaut – versucht, die ursprünglichen Eingangsdaten zu rekonstruieren.

2.4.2 Trainingsstrategien

Datenverfügbarkeit. Die Modellgüte ist stark abhängig von der Qualität der Trainingsdaten. Das Erstellen von großen, hochwertigen Datensätzen kann jedoch mit einem hohen methodischen und zeitlichem Aufwand verbunden sein. Die Bereitstellung großer, annotierter Datensätze – wie beispielsweise ImageNet (Deng u. a. 2009)– war für den Fortschritt in diesem Bereich von zentraler Bedeutung. Mittels Transfer Learning können bereits auf großen Datensätzen vortrainierte Modelle für neue Aufgaben weiterverwendet werden. Dabei wird das bestehende Modell durch Fine-Tuning an spezifische Anforderungen angepasst. Dieses Vorgehen hat sich unter anderem in der Objekterkennung (Girshick 2015) sowie der Bildsegmentierung (Zhao u. a. 2017) bewährt.

Eine Strategie zur Generierung großer Datenmengen stellt die Nutzung synthetischer Datensätze dar, die mithilfe von Simulationssoftware erzeugt werden. Dabei ist es essenziell, eine möglichst realitätsnahe Darstellung zu erreichen, wenngleich gewisse Unterschiede zu realen Daten meist nicht vollständig vermeidbar sind. Es konnte jedoch gezeigt werden, dass bei ausreichender Qualität auch synthetische Daten zur erfolgreichen Modellierung beitragen können (Ceruso u. a. 2021; Maximov, Galim und Leal-Taixe 2020).

Trainingsprozess. Beim Training von CNNs ist die Strukturierung des Lernprozesses von zentraler Bedeutung. Anstelle der Verarbeitung des gesamten Trainingsdatensatzes in einem Schritt erfolgt die Aufteilung in kleinere Teilmengen, sogenannte Batches. Ein Batch besteht aus einer definierten Anzahl an Trainingsbeispielen, die gleichzeitig durch das Netzwerk propagiert werden. Dies ermöglicht eine effizientere Ausnutzung der verfügbaren Hardware-Ressourcen.

Ein vollständiger Durchlauf des gesamten Trainingsdatensatzes wird als Epoche bezeichnet. In der Regel sind mehrere Epochen erforderlich, um das Modell sukzessive an die zugrunde liegenden Datenstrukturen anzupassen und eine hinreichende Konvergenz zu erzielen.

Weitere zentrale Begriffe im Trainingsprozess sind der Trainingsfehler, der Validierungsfehler und die Lernrate. Der Trainingsfehler beschreibt den Fehler, den das Modell auf den Daten

erzielt, mit denen es direkt trainiert wurde. Er dient als primäre Rückmeldung während des Lernprozesses und zeigt, wie gut das Modell die vorhandenen Trainingsdaten approximieren kann. Der Validierungsfehler hingegen wird auf einem separaten, zuvor nicht gesehenen Teil des Datensatzes berechnet, der lediglich zur Bewertung der Modellleistung dient. Er gibt Aufschluss darüber, wie gut das Modell in der Lage ist, auf neue, unbekannte Daten zu generalisieren. Die Lernrate wiederum bestimmt die Schrittweite, mit der die Gewichte des neuronalen Netzes bei jeder Iteration aktualisiert werden.

Ein häufiges Problem beim Training tiefer neuronaler Netze ist das sogenannte Overfitting. Dabei lernt das Modell sehr gute Vorhersagen für die Trainingsdaten zu liefern, lässt sich aber schlecht für zuvor ungesehene Daten generalisieren. Zur Reduktion von Overfitting kommen verschiedene Regularisierungsmethoden zum Einsatz, darunter die Datenaugmentation. Augmentation ist die gezielte künstliche Erweiterung des Trainingsdatensatzes durch Transformationen wie Rotationen, Spiegelungen, Skalierungen oder Farbveränderungen. Ziel ist es, die Varianz im Trainingsdatensatz zu erhöhen und die Robustheit des Modells gegenüber variierenden Eingaben zu verbessern.

Verlustfunktionen. Um die Genauigkeit eines Deep-Learning-Modells zu bestimmen, kommen Verlustfunktionen zum Einsatz. Zwei weit verbreitete Verlustfunktionen sind der L1-Loss und der L2-Loss. Der L1-Loss, auch als Mean Absolute Error (MAE) bezeichnet, berechnet den absoluten Unterschied zwischen Vorhersage und tatsächlichem Wert. Der L1-Loss ist besonders robust gegenüber Ausreißern, da jeder Fehler gleich gewichtet wird. Der L1-Loss wird mit den vorhergesagten Daten D, den Ground-Truth-Daten D^* und der Anzahl der Datenpunkte N berechnet mit:

$$L_1 = \text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |D_i - D_i^*| \tag{11}$$

Im Gegensatz dazu verwendet der L2-Loss, auch bekannt als Mean Squared Error (MSE), das Quadrat der Differenz zwischen Vorhersage und Zielwert. Das hat zur Folge, dass größere Fehler deutlich stärker bestraft werden als kleinere. Der L2-Loss führt dadurch oft zu glatteren und stabileren Vorhersagen, da das Modell versucht, größere Abweichungen möglichst zu vermeiden. Allerdings ist er empfindlicher gegenüber Ausreißern, weil diese durch das Quadrieren des Fehlers überproportional ins Gewicht fallen. Der L2-Loss wird berechnet zu:

$$L_2 = \text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (D_i - D_i^*)^2 \tag{12}$$

Um die Eigenschaften des L1- und L2-Loss zu kombinieren kann der Smooth-L1-Loss verwendet werden. Damit wird erreicht, dass der Fehler sowohl robust gegen Ausreißer ist als auch glatte Gradienten für kleine Fehler bildet. Somit wird eine stabile und robuste Optimierung ermöglicht. Der Smooth-L1-Loss mit dem wählbaren Parameter γ wird berechnet gemäß:

$$L_{1,smooth} = \begin{cases} 0.5(D - D^*)^2/\gamma, & \text{wenn } |D - D^*| < \gamma \\ |D - D^*| - 0.5\gamma, & \text{sonst} \end{cases} \tag{13}$$

2.4.3 Potenziale und Herausforderungen von CNNs in der Bildverarbeitung

CNN-basierte Verfahren bieten eine Reihe bedeutender Potenziale für Anwendungen in der Bildverarbeitung. Zu den zentralen Vorteilen zählt insbesondere die Fähigkeit, komplexe und nichtlineare Zusammenhänge in den Bilddaten modellieren zu können. Zudem ermöglichen CNNs eine effiziente und hierarchische Merkmalsextraktion, wobei bereits vortrainierte Modelle als Grundlage genutzt und auf spezifische Anwendungsbereiche angepasst werden können. Dadurch lassen sich in vielen Fällen eine hohe Genauigkeit und Robustheit erzielen, selbst bei variierenden Bildinhalten.

Den genannten Stärken stehen jedoch auch mehrere Herausforderungen gegenüber. So erfordert der Trainingsprozess in der Regel große Mengen qualitativ hochwertiger Daten, um eine ausreichende Generalisierungsfähigkeit der Modelle zu gewährleisten. Darüber hinaus ist das Training mit erheblichem rechnerischen Aufwand verbunden. Ein weiterer kritischer Punkt betrifft die Nachvollziehbarkeit der Entscheidungen neuronaler Netzwerke: Aufgrund ihrer komplexen Struktur gelten sie häufig als schwer interpretierbar und werden daher auch als Black-Box-Systeme bezeichnet.

3 Stand der Technik

In diesem Kapitel werden zunächst bestehende DDFF-Verfahren vorgestellt. Auf Basis dieser Übersicht erfolgt anschließend die Auswahl zweier geeigneter Methoden, die für das experimentelle 3D-Mikroskop implementiert werden. Für das Training und die Evaluation der vorgestellten Verfahren kommen unterschiedliche Datensätze zum Einsatz. Zur besseren Übersicht wird eine zusammenfassende Beschreibung dieser Datensätze in einem separaten Abschnitt gegeben. Am Ende des Kapitels erfolgt eine Zusammenfassung des aktuellen Forschungsstandes.

3.1 Deep Depth-from-Focus Modelle

Der Einsatz künstlicher neuronaler Netze im Zusammenhang mit der DFF-Methode wurde vermutlich erstmals von Muhammad und T.-S. Choi (1999) beschrieben. In ihrer Arbeit kam ein Feedforward-Neuronales Netz zum Einsatz, um das Tiefenprofil einfacher geometrischer Strukturen zu bestimmen. Mit der zunehmenden Komplexität neuronaler Netzarchitekturen und dem Aufkommen vielschichtiger Deep Neural Networks ist es inzwischen möglich, auch die Tiefeninformationen komplexerer Szenen mithilfe maschineller Lernverfahren zu extrahieren. Im Folgenden werden DDFF-Modelle in chronologisch aufsteigender Reihenfolge vorgestellt.

3.1.1 DDFFNet

Hazirbas u. a. (2019) verfolgten zur Tiefenschätzung einen End-to-End-Learning-Ansatz, bei dem aus gegebenen Fokalstapel direkt die zugehörigen Tiefenkarten generiert werden. Das hierfür entwickelte CNN basiert auf einer Autoencoder-Architektur. Jedes Bild des Fokalstapels wird durch 2D-Convolutional-Layers verarbeitet, wobei der Encoder zentrale Bildmerkmale extrahiert. Der Decoder rekonstruiert anschließend für jedes Bild eine Feature-Map, die Fokusinformationen enthält. Dieser Schritt ähnelt konventionellen DFF-Methoden, bei denen für jeden Pixel eines Bildes ein Fokuswert bestimmt wird. Auf die Gesamtheit der Feature-Maps wird schließlich eine Regressionsschicht angewendet, die die Objekttiefe berechnet. Als Grundlage für das Autoencoder-Netzwerk diente das VGG-16-Netzwerk (Simonyan und Zisserman 2015), das für die Aufgabe der Tiefenschätzung modifiziert wurde.

Für das Training und die Evaluierung des Modells wurde der Datensatz DDFF-12-Scene verwendet, der mithilfe einer Lichtfeldkamera und eines RGB-D-Sensors erstellt wurde. Der Datensatz besteht aus Fokalstapeln sowie den zugehörigen Ground-Truth-Tiefenkarten von zwölf Alltagsszenen in Innenräumen. Lichtfeldkameras – auch plenoptische Kameras genannt – erfassen sowohl die Lichtintensität als auch die Einfallsrichtung des Lichts, was die Gewinnung von Tiefeninformationen ermöglicht. Aus den Aufnahmen der Lichtfeldkamera kann ein Fokalstapel berechnet werden, sodass bereits eine einzige Aufnahme pro Szene zur Generierung des Stapels genügt. Dies reduziert den Aufwand bei der Datensatz-Erstellung erheblich. Die Ground-Truth-Tiefenkarten wurden separat mithilfe eines RGB-D-Sensors erfasst.

Im Vergleich zur konventionellen DFF-Methode Variational Depth-from-Focus (VDFF) nach Moeller u. a. (2015) konnte das DDFFNet die Fehler in der Tiefenberechnung deutlich reduzieren und gleichzeitig eine schnellere Berechnung der Tiefenkarten ermöglichen.

Trotz dieser Fortschritte weist DDFFNet einige Einschränkungen auf. So berücksichtigt das Modell keine optischen Eigenschaften der aufnehmenden Kamera, was zu Overfitting führen kann. Wird das Modell mit Daten einer anderen Kamera getestet als jener, mit der es trainiert wurde, kann die Genauigkeit der resultierenden Tiefenkarten deutlich abnehmen. Zudem ist das Netzwerk nur auf Fokalstapel anwendbar, die dieselbe Anzahl an Bildern enthalten wie jene im Trainingsdatensatz. In der Praxis variiert jedoch die benötigte Anzahl von Aufnahmen je nach Schärfentiefe der Kamera, um den gesamten Fokusbereich abzudecken. Diese Einschränkung erschwert eine flexible Generalisierung des Modells auf unterschiedliche Aufnahmesituationen.

3.1.2 DefocusNet

Statt eines End-to-End-Learning-Ansatzes verfolgten Maximov, Galim und Leal-Taixe (2020) einen zweigeteilten Aufbau ihres DDFF-Modells. Dabei generiert ein erstes Autoencoder-CNN namens DefocusNet, basierend auf 2D-Convolutional-Layers, für jedes Bild des Fokalstapels eine Fokuswert-Karte. Anschließend berechnet ein zweites Autoencoder-CNN, DepthNet, die zugehörige Tiefenkarte.

Der Hauptunterschied zum Modell von Hazirbas u. a. (2019) besteht darin, dass DefocusNet unabhängig von der Tiefenberechnung trainiert wird, um ausschließlich die Bildschärfe zu ermitteln. Zudem integriert das Modell bildübergreifende Pooling-Schichten, die den Vergleich von Merkmalen zwischen den verschiedenen Bildern des Fokalstapels ermöglichen. Dadurch ist das Modell flexibel hinsichtlich der Anzahl der Bilder im Stapel beim Training und kann somit auch auf unterschiedlich große Fokalstapel angewendet werden.

Für das Training wurde der synthetischer Datensatz FoD500 erstellt, der neben den Ground-Truth-Tiefenkarten auch Ground-Truth-Fokuswert-Karten enthält. Der Einsatz computergenerierter Bilder bietet mehrere Vorteile: Die Erstellung eines umfangreichen Datensatzes ist deutlich weniger zeitaufwendig, und die erzeugten Tiefenkarten sind frei von Messfehlern. Mithilfe der Software Blender (Blender-Foundation 2018) wurden dazu verschiedene Computer Aided Design (CAD)-Modelle zufällig arrangiert und aus unterschiedlichen Kameraperspektiven aufgenommen.

Das vollständig mit synthetischen Daten trainierte DDFF-Modell wurde anschließend mit den realen Datensätzen DDFF-12-Scene (Hazirbas u. a. 2019) und MobileDepth (Suwajanakorn, Hernandez und Seitz 2015) sowie mit den synthetisch-unscharfen RGB-D-Datensätzen NYU-Depth-V2 (Silberman u. a. 2012), 7-Scenes (Shotton u. a. 2013), Middlebury (Scharstein u. a. 2014) und SUN-RGB-D (B. Zhou u. a. 2014) evaluiert. Für Letztere wurden Fokalstapel aus AiF-Bildern und Tiefenkarten berechnet. Hierzu wurde der Ansatz von Gur und Wolf (2019) adaptiert, die ein PSF Network Layer zur realitätsnahen Erzeugung unscharfer Bilder entwickelt hatten.

Der Datensatz MobileDepth umfasst Fokalstapel von Alltagsgegenständen, die mithilfe eines Smartphones aufgenommen wurden. Passende Ground-Truth-Tiefenkarten sind nicht verfügbar, so dass mit diesem Datensatz nur eine qualitative Evaluation möglich ist. Die Datensätze NYU-Depth-V2, 7-Scenes und SUN-RGB-D enthalten AiF-Bilder, die mit einer Kamera aufgenommen wurden, sowie Tiefenkarten, die auf der Streifenprojektionsmethode basieren. Die Datensätze zeigen typische Alltagsszenen. Der Middlebury-Datensatz beinhaltet AiF-Bilder sowie Disparitäten von Alltagsszenen, die mithilfe der Stereo-Vision-Methode akquiriert wurden.

Die Tiefenschätzungen mit den realen Datensätzen wurden mit den Ergebnissen von DDFFNet (Hazirbas u. a. 2019) und der VDFF-Methode (Moeller u. a. 2015) verglichen. DefocusNet erzielte dabei den geringsten mittleren Fehler. Auch im Vergleich mit der DFD-Methode Virtual Normal Loss (VNL) von Yin u. a. (2019), welcher auf den synthetisch-unscharfen Datensätzen durchgeführt wurde, zeigte DefocusNet im Durchschnitt die besten Ergebnisse.

3.1.3 MultiscaleDDFFNet

Mit dem Ziel, ein robusteres und schnelleres DDFF-Verfahren zu entwickeln, stellten Ceruso u. a. (2021) ein eigenes Modell vor. Um eine Unabhängigkeit von der Anzahl der Bilder im Fokalstapel zu gewährleisten, kam ein Siamese-Encoder zum Einsatz, bei dem die Gewichte zwischen den einzelnen Strängen für die Bilder des Stapels geteilt werden. Der Decoder basiert auf 3D-Convolutional-Layers, die die vom Encoder extrahierten Merkmale der verschiedenen Fokusebenen miteinander vergleichen. Die Tiefenkarte wird anschließend durch eine Regressionsschicht berechnet.

Für das Training des Modells wurden sowohl der synthetische Datensatz FlyingThings (Mayer u. a. 2016) als auch die synthetisch-unscharfen RGB-D-Datensätze Middlebury (Scharstein u. a. 2014) und DIML (Kim u. a. 2018) verwendet. FlyingThings beinhaltet abstrakte CAD-Geometrien sowie Szenen aus Animationsfilmen. Mithilfe der Software Blender (Blender-Foundation 2018) wurden daraus Fokalstapel mit künstlich erzeugter Unschärfe generiert. DIML enthält reale Aufnahmen von Alltagsszenen, die mit einem RGB-D-Sensor erfasst wurden.

Zur Erzeugung unscharfer Bilder aus den RGB-D-Datensätzen wurde das Dünnlinsenmodell angenommen. Der Tiefenbereich wurde in eine diskrete Anzahl von Schichten unterteilt. Auf die Bildbereiche, die einer bestimmten Tiefenschicht zugeordnet sind, wurde jeweils ein entsprechender Gauß-Filter angewendet. Die Standardabweichung des Gauß-Filters hängt von der Objekttiefe g und dem Fokusabstand des Kamerasystems g_f ab und wurde empirisch bestimmt zu $\sigma = 0.3 \cdot \mathrm{CoC}(g, g_f)$. Anschließend wurden die verschiedenen defokussierten Schichten mithilfe linearer Interpolation wieder zu einem Bild zusammengesetzt.

Beim Test mit den synthetisch-unscharfen Datensätzen erzielte das Modell geringere Fehler bei der Tiefenschätzung als die Methoden VDFF (Moeller u. a. 2015) und DDFFNet (Hazirbas u. a. 2019). Darüber hinaus wurde das mit Middlebury trainierte Modell auf reale Fokalstapel angewendet, die mit einem Smartphone aufgenommen wurden. Auch hier lieferte das Verfahren qualitativ bessere Tiefenkarten als die beiden Vergleichsmethoden.

3.1.4 AiFDepthNet

Wang u. a. (2021) entwickelten eine DDFF-Methode, die sowohl im Rahmen eines überwachten Lernverfahrens (supervised learning) mit Fokalstapeln und zugehörigen Tiefenkarten als auch mittels unüberwachtem Lernen (unsupervised learning) mit Fokalstapeln und AiF-Bildern trainiert werden kann. Im Gegensatz zu den überwachten Ansätzen ist das unüberwachte Verfahren unabhängig von möglichen Fehlern oder Ungenauigkeiten in den Ground-Truth-Tiefenkarten. Zudem kann die Generierung eines AiF-Bildes unter Umständen mit geringerem Aufwand verbunden sein als die Erstellung einer präzisen Tiefenkarte.

Das zugrunde liegende Autoencoder-CNN basiert auf dem Inception3D-Modell von Alayrac, Carreira und Zisserman (2019), das 3D-Faltungen einsetzt. Diese ermöglichen es dem Netzwerk, Fokusmerkmale nicht nur innerhalb einzelner Bilder, sondern auch zwischen den Bildern des Fokalstapels zu erkennen. Die 3D-Faltungen bieten damit eine effektivere Möglichkeit zur Modellierung zwischenbildlicher Zusammenhänge als die Pooling-Schichten in früheren Modellen von Maximov, Galim und Leal-Taixe (2020) und Ceruso u. a. (2021) und erlauben ebenso die Verarbeitung von Fokalstapeln variabler Größe.

Die Ausgabe des vorgeschlagenen Netzwerks ist eine sogenannte Attention-Map, die für jeden Pixel jedes Bildes eine Wahrscheinlichkeit angibt, dass dieser Pixel die höchste Schärfe im gesamten Fokalstapel aufweist. Aus diesen Attention-Maps können sowohl eine Tiefenkarte als auch ein AiF-Bild berechnet werden. Dadurch ist es möglich, das Modell flexibel entweder mit Tiefenkarten oder mit AiF-Bildern zu trainieren.

Das Netzwerk wurde sowohl mit dem realen Datensatz DDFF-12-Scene (Hazirbas u. a. 2019) als auch mit den synthetischen Datensätzen 4D-Light-Field (Honauer u. a. 2017) und FoD500 (Maximov, Galim und Leal-Taixe 2020) trainiert und getestet. 4D-Light-Field besteht aus mit Blender erzeugten Innenraumszenen und abstrakten Geometrien. In allen drei Fällen führte das Modell – sowohl bei supervised als auch bei unsupervised Training – zu geringeren Fehlern in der Tiefenberechnung im Vergleich zu den Methoden DefocusNet (Maximov, Galim und Leal-Taixe 2020) und VDFF (Moeller u. a. 2015).

Darüber hinaus wurde das Modell auf dem synthetischen Datensatz FlyingThings (Mayer u. a. 2016) trainiert und anschließend mit dem realen Datensatz MobileDepth (Suwajanakorn, Hernandez und Seitz 2015) sowie dem synthetisch-unscharfen Datensatz Middlebury (Scharstein u. a. 2014) evaluiert. Für Middlebury wurden die unscharfen Bilder mithilfe der Methode von Barron u. a. (2015) erzeugt, bei der aus Stereo-Bildpaaren unter Verwendung des sogenannten Bilateral-Space realistische Unschärfe simuliert wird. Der Bilateral-Space ist ein fünfdimensionaler Raum, der zwei Dimensionen für die Pixelposition und drei für die Farbwerte umfasst. Auch in dieser Konfiguration lieferte das vorgeschlagene Modell bessere Ergebnisse als die Vergleichsmethode DefocusNet.

3.1.5 DDFFintheWild

Durch die Verschiebung der Fokusebene beim Aufnehmen eines Fokalstapels können sich die Bildvergrößerung sowie die Lage des Bildhauptpunkts des optischen Systems leicht verändern. Dieser Effekt wird als Focal Breathing bezeichnet. Frühere Ansätze haben diese Veränderungen entweder vernachlässigt oder die aufgenommenen Bilder mithilfe speziell für den jeweiligen Datensatz entwickelter Algorithmen korrigiert. Won und Jeon (2022) entwickelten stattdessen ein Alignment-Netzwerk, das die Bilder auf Grundlage von Bildmerkmalen automatisch ausrichtet.

Im Anschluss an die Bildausrichtung wird ein CNN verwendet, um Fokuswerte für die einzelnen Bilder zu berechnen, aus denen wiederum die Tiefeninformation abgeleitet wird. Das zur Tiefenschätzung eingesetzte Netzwerk ist modular aufgebaut und besteht aus einer iterativen Abfolge von Merkmalsextraktions- und Downsampling-Einheiten. Für die Merkmalsextraktion kommen sowohl 2D- als auch 3D-Faltungen zum Einsatz, wodurch der Informationsgehalt gegenüber früheren Methoden deutlich gesteigert werden konnte. Das Netzwerk ist zudem in der Lage, Fokalstapel mit variabler Bildanzahl zu verarbeiten. Nach einer zusätzlichen Verfeinerung der berechneten Fokuswertkarten wird schließlich die Tiefenkarte generiert.

Das Modell zur Tiefenbestimmung wurde mit den synthetischen Datensätzen FlyingThings (Mayer u. a. 2016) und 4D-Light-Field (Honauer u. a. 2017) sowie dem realen Datensatz Smartphone (Herrmann u. a. 2020) trainiert und evaluiert. Der Smartphone-Datensatz enthält Fokalstapel von Alltagsszenen, die mit einem Smartphone aufgenommen wurden, sowie dazugehörige Tiefenkarten, die mithilfe der Multi-View-Stereo-Methode (Garg u. a. 2019) generiert wurden. Im Vergleich zu DefocusNet (Maximov, Galim und Leal-Taixe 2020) und AiFDepthNet (Wang u. a. 2021) konnte das vorgeschlagene Modell die höchste Genauigkeit bei der Tiefenschätzung erzielen.

Darüber hinaus wurde das auf FlyingThings trainierte Modell auf den Datensätzen Middlebury (Scharstein u. a. 2014) und FoD500 (Maximov, Galim und Leal-Taixe 2020) getestet. Auch hier erzielte DDFFintheWild geringere Fehler in der Tiefenberechnung als die Vergleichsmodelle DefocusNet und AiFDepthNet.

3.1.6 DFVNet

Die von Yang, Huang und Z. Zhou (2022) vorgeschlagene DDFF-Methode verwendet ein sogenanntes 4D-Focus-Volume, um eine Wahrscheinlichkeitsverteilung für jeden Pixel im Fokalstapel zu berechnen, die angibt, mit welcher Wahrscheinlichkeit dieser Pixel der am besten fokussierte ist. Zunächst werden für jedes Bild des Fokalstapels Merkmale mithilfe eines 2D-CNNs extrahiert. Das 2D-CNN basiert auf dem mit ImageNet (Deng u. a. 2009) vortrainiertem Modell ResNet-18-FPN (Lin u. a. 2017). Aus den Bildmerkmalen wird anschließend das 4D-Focus-Volume konstruiert. Ein darauf folgendes 3D-CNN berechnet auf Basis dieses Volumens die Wahrscheinlichkeiten für jeden Pixel. Die Tiefe wird schließlich mithilfe einer Wahrscheinlichkeitsregression ermittelt.

Es konnte gezeigt werden, dass sich die Leistungsfähigkeit dieses Ansatzes weiter steigern lässt, wenn vor der Anwendung des 3D-CNNs zunächst der Gradient des 4D-Focus-Volumes berechnet

wird. Da Gradienten besonders empfindlich auf lokale Extremwerte reagieren, liefern sie wertvolle Hinweise auf stark fokussierte Bildbereiche.

Das Modell wurde mit dem synthetischen Datensatz FoD500 (Maximov, Galim und Leal-Taixe 2020) sowie dem realen Datensatz DDFF-12-Scene (Hazirbas u. a. 2019) trainiert und evaluiert. Im Vergleich zu den Verfahren DDFFNet (Hazirbas u. a. 2019), DefocusNet (Maximov, Galim und Leal-Taixe 2020), AiFDepthNet (Wang u. a. 2021) sowie den klassischen Methoden VDFF (Moeller u. a. 2015) und Ring Difference Filter (RDF) von Surh u. a. (2017) konnte die vorgeschlagene Methode bessere Ergebnisse in der Tiefenschätzung erzielen.

Zusätzlich wurde das trainierte Modell auf dem realen Datensatz MobileDepth (Suwajanakorn, Hernandez und Seitz 2015) getestet. Im Vergleich zu den Referenzmethoden DDFFNet, DefocusNet, AiFDepthNet sowie der konventionellen Methode MobileDFF (Suwajanakorn, Hernandez und Seitz 2015) zeigte sich, dass die neue Methode insbesondere an Objektkanten eine präzisere Tiefenerfassung ermöglicht. Darüber hinaus erwies sie sich als robuster gegenüber Texturen oder Mustern in den Bildern.

3.1.7 DDFSNet

Die Unschärfe eines Bildes hängt wesentlich von Kameraparametern wie Fokusabstand, Brennweite und Blendenzahl ab. Werden DDFF-Modelle mit Kameraparametern trainiert, die sich von denen im Testzeitpunkt unterscheiden, kann dies zu Fehlern in der Tiefenschätzung führen. Um die Generalisierbarkeit von DDFF-Modellen gegenüber unterschiedlichen Aufnahmebedingungen zu verbessern, entwickelten Fujimura u. a. (2024) ein Verfahren, das diese Kameraparameter explizit berücksichtigt.

Im ersten Schritt wird aus dem Fokalstapel und den zugehörigen Kameraparametern ein Kostenvolumen berechnet. Durch die Einbeziehung der Kameraparameter ist das Modell in der Lage, mit unterschiedlichen Einstellungen in Trainings- und Testdaten umzugehen. Anschließend wird aus dem Kostenvolumen mithilfe eines Autoencoder-CNNs die Tiefenkarte berechnet.

Das vorgeschlagene Modell wurde auf dem synthetischen Datensatz FoD500 (Maximov, Galim und Leal-Taixe 2020) trainiert und mit dem synthetisch-unscharfen Datensatz NYU-Depth-V2 (Silberman u. a. 2012) getestet. Die unscharfen Bilder wurden dabei mithilfe der Methode von Carvalho u. a. (2019) erzeugt. Diese basiert auf einer diskreten Einteilung des Tiefenbereichs in mehrere Schichten. Auf die einer Tiefenschicht zugehörigen Bildbereiche wird dann eine entsprechende Disk-Funktion angewendet. Anschließend werden die Bildbereiche unter Berücksichtigung von Unschärfe-Überlappungen zu einem finalen Bild zusammengesetzt.

Im Vergleich zu den Methoden DDFFNet (Hazirbas u. a. 2019), DefocusNet (Maximov, Galim und Leal-Taixe 2020), AiFDepthNet (Wang u. a. 2021) und DFVNet (Yang, Huang und Z. Zhou 2022) erzielte das vorgeschlagene Verfahren geringere mittlere Fehler bei der Tiefenschätzung. Darüber hinaus konnte gezeigt werden, dass auch bei Anwendung auf die realen Bilddaten MobileDepth (Suwajanakorn, Hernandez und Seitz 2015) plausible Tiefenkarten erzeugt werden.

3.2 Deep Depth-from-Focus Datensätze

Die für Training und Evaluation der beschriebenen DDFF-Modelle verwendeten Datensätze lassen sich in drei Kategorien unterteilen:

- **Reale Datensätze:** Fokalstapel und zugehörige Ground-Truth-Tiefenkarten werden mit realen Kamerasystemen und 3D-Messverfahren aufgezeichnet.
- **Synthetisch-unscharfe Datensätze:** Aufnahmen einer realen Kamera werden auf Basis vorhandener Tiefenkarten künstlich defokussiert. Zur Erzeugung eines Fokalstapels genügen also ein AiF-Bild sowie eine dazugehörige Tiefenkarte.
- **Synthetische Datensätze:** Sowohl Fokalstapel als auch Tiefenkarten werden vollständig per Computersimulation generiert.

Tabelle 1 bietet eine Übersicht über die eingesetzten Datensätze und ihre jeweilige Kategorisierung.

Tabelle 1: Übersicht der Datensätze, die für das Training und die Evaluation von DDFF-Modellen verwendet wurden.

Datensatz	Typ	RGB-Erfassung	Tiefenmessung	Motive	Datensatz-Größe
DDFF-12-Scene (Hazirbas u. a. 2019)	real	Lichtfeldkamera	Streifenprojektion	Innenraumszenen	720
mDFF (Hazirbas u. a. 2019)	real	Smartphone	Smartphone	Alltagsszenen	202
MobileDepth (Suwajanakorn, Hernandez und Seitz 2015)	real	Smartphone	-	Alltagsgegenstände	6
Smartphone (Herrmann u. a. 2020)	real	Smartphone	Multi-View-Stereo	Alltagsszenen	510
FoD500 (Maximov, Galim und Leal-Taixe 2020)	synthetisch	Blender	Blender	CAD-Geometrien	1000
FlyingThings (Mayer u. a. 2016)	synthetisch	Blender	Blender	CAD-Geometrien, Animationen	250000
4D-Light-Field (Honauer u. a. 2017)	synthetisch	Blender	Blender	Innenräume, abstrakte Geometrien	24
NYU-Depth-V2 (Silberman u. a. 2012)	synthetisch-unscharf	Kamera	Streifenprojektion	Innenraumszenen	1449
7-Scenes (Shotton u. a. 2013)	synthetisch-unscharf	Kamera	Streifenprojektion	Innenraumszenen	43000
SUN-RGB-D (B. Zhou u. a. 2014)	synthetisch-unscharf	Kamera	Streifenprojektion	Innenraumszenen	10000
Middlebury (Scharstein u. a. 2014)	synthetisch-unscharf	Kamera	Stereo-Vision	Innenraumszenen	26
DIML (Kim u. a. 2018)	synthetisch-unscharf	Kamera	Streifenprojektion	Innenraumszenen	220000

3.3 Zusammenfassung des Forschungsstands

In der aktuellen Forschung zu DDFF-Verfahren wurden verschiedene Ansätze entwickelt, um die Tiefenberechnung anhand von Fokalstapeln zu verbessern. DDFFNet (Hazirbas u. a. 2019) nutzt einen End-to-End-Learning-Ansatz, zeigt jedoch Einschränkungen hinsichtlich Generalisierbarkeit und Flexibilität bei variabler Stapelgröße. DefocusNet (Maximov, Galim und Leal-Taixe 2020) verfolgt einen zweistufigen Aufbau mit getrennten Fokuswert- und Tiefenberechnungsnetzwerken und ermöglicht durch die Verwendung von Pooling-Schichten mehr Flexibilität hinsichtlich der Stapelgröße. Es wurde zudem gezeigt, dass auch mit dem Training mit synthetischen Datensätzen eine gute Generalisierbarkeit auf reale Bilddaten erreicht werden kann.

MultiscaleDDFFNet (Ceruso u. a. 2021) kombiniert einen Siamese-Encoder mit 3D-Convolutional-Decodern, um die Stapelgröße variabel zu halten und die Effizienz zu steigern. AiFDepthNet (Wang u. a. 2021) erlaubt sowohl überwachtes als auch unüberwachtes Lernen und nutzt Attention-Maps, um Tiefenkarten oder AiF-Bilder auszugeben, wodurch eine größere Trainingsflexibilität entsteht. DDFFintheWild (Won und Jeon 2022) adressiert erstmals systematisch den Einfluss des Focal-Breathing-Effekts und integriert ein Alignment-Netzwerk zur automatischen Bildausrichtung vor der Tiefenberechnung.

DFVNet (Yang, Huang und Z. Zhou 2022) entwickelt ein sogenanntes 4D-Focus-Volume zur präziseren Schärfeschätzung und nutzt Gradienteninformationen, um die Tiefenbestimmung weiter zu verbessern. Schließlich berücksichtigt DDFSNet (Fujimura u. a. 2024) explizit Kameraparameter wie Brennweite und Blendenzahl, um die Robustheit gegenüber unterschiedlichen Aufnahmebedingungen zu erhöhen.

Für das Training und die Evaluation dieser Modelle wurden unterschiedliche Datensätze eingesetzt, die sich in real, synthetisch-unscharf und vollständig synthetisch einteilen lassen. Die Modelle wurden dabei sowohl mit Datensätzen trainiert und getestet, die Alltagsszenen als auch abstrakte CAD-Geometrien umfassen. Reale Datensätze wie DDFF-12-Scene (Hazirbas u. a. 2019) und Smartphone (Herrmann u. a. 2020) basieren auf tatsächlichen unscharfen Aufnahmen mit gemessenen Tiefenkarten, während synthetisch-unscharfe Datensätze wie NYU-Depth-V2 (Silberman u. a. 2012) künstlich erzeugte Unschärfe aus realen Bildern simulieren. Vollsynthetische Datensätze wie FoD500 (Maximov, Galim und Leal-Taixe 2020) und FlyingThings (Mayer u. a. 2016) bestehen aus Fokalstapeln und Tiefenkarten die vollständig mit Computersimulation erzeugt wurden. Sie stellen umfangreiche Datenmengen bereit, die für das Training genutzt werden können.

Die Ergebnisse zeigen, dass moderne DDFF-Modelle konventionelle DFF-Methoden in Bezug auf Genauigkeit, Robustheit und Flexibilität deutlich übertreffen. Dabei sind die Qualität der Trainingsdaten, die Berücksichtigung physikalischer Effekte und die anpassungsfähige Modellarchitektur entscheidend für den erfolgreichen Praxiseinsatz.

4 Konzept

Ziel dieser Arbeit ist die Entwicklung einer Methode zur möglichst genauen Berechnung von Tiefenkarten mithilfe eines experimentellen DFF-Mikroskops. Der Stand der Technik zeigt, dass DDFF-Modelle, die auf CNNs basieren, in der Regel deutlich genauere Ergebnisse liefern als konventionelle, rein algorithmische DFF-Verfahren.

Im Rahmen dieser Arbeit werden zwei geeignete DDFF-Modelle identifiziert und getestet. Die Auswahl basiert auf spezifischen Vergleichskriterien, darunter die Performanz auf etablierten Datensätzen, die Flexibilität hinsichtlich der Anzahl von Bildern im Fokalstapel sowie die Verfügbarkeit von Implementierungen. Zur Einordnung der erzielten Ergebnisse wird zusätzlich ein konventionelles DFF-Verfahren herangezogen.

Für die meisten DDFF-Methoden stehen bereits vortrainierte Modellversionen zur Verfügung. Das Training erfolgte dabei allerdings mit Datensätzen, die alltägliche Szenen sowie CAD-Geometrien enthalten und hinsichtlich Struktur, Textur und Oberflächeneigenschaften kaum Ähnlichkeit zu Gesteinsproben aufweisen. Inwiefern diese Modelle auf Gesteinsproben generalisieren, ist unklar. Darüber hinaus ist bekannt, dass die Unschärfecharakteristik stark vom verwendeten optischen System abhängt, wobei viele der Modelle kamerarelevante Parameter nicht explizit berücksichtigen. Dies kann die Generalisierbarkeit weiter einschränken.

Aus diesen Gründen sollen die Modelle mit einem neuen Trainingsdatensatz trainiert werden, der sowohl hinsichtlich der Bildinhalte als auch der Kameraparameter dem experimentellen Aufbau entspricht. Die Erstellung eines solchen Datensatzes ist jedoch mit erheblichem Aufwand verbunden, da eine Vielzahl an Proben erforderlich ist und der gesamte Prozess zeitintensiv ist.

Zur Reduktion des methodischen und zeitlichen Aufwands wird stattdessen ein bereits existierenden Datensatz für das Training der DDFF-Modelle verwendet. Bei der Recherche nach öffentlich verfügbaren Datensätzen zeigte sich jedoch, dass kein DFF-Datensatz mit Mikroskopaufnahmen von Gesteinsproben verfügbar ist. Allerdings existiert ein RGB-D-Datensatz mit Mikroskopaufnahmen von Metall- und Kunststoffoberflächen, die mittels verschiedener Fertigungsverfahren erzeugt wurden. Diese weisen eine strukturierte Topografie mit Kerben und Vertiefungen auf, die in ihrer Oberflächencharakteristik Gesteinsproben ähnlich sind.

Da in diesem Datensatz keine Fokalstapel enthalten sind, müssen synthetisch defokussierte Bildreihen aus den vorhandenen AiF-Bildern und Tiefenkarten generiert werden. Diese Bildreihen können anschließend für das Training der Modelle verwendet werden.

Für die abschließende Evaluation wird ein Testdatensatz mithilfe des experimentellen Mikroskopaufbaus erstellt. Hierfür sind zunächst die geometrische Kalibrierung des Kamerasystems, die Charakterisierung der Positioniergenauigkeit des Aktuators sowie die Implementierung einer geeigneten Beleuchtung erforderlich. Die Ground-Truth-Tiefenkarten zur Validierung der Ergebnisse werden mit einem kommerziellen 3D-Mikroskop auf Basis von Streifenprojektion aufgenommen.

5 Umsetzung

In diesem Kapitel wird die Umsetzung des Deep-Learning-gestützten DFF-Mikroskops im Detail beschrieben. Zunächst erfolgt die Darstellung des bestehenden experimentellen Versuchsaufbaus. Anschließend wird erläutert, wie ein vereinfachtes optisches Modell für diesen Aufbau entwickelt wurde. Dieses Modell dient sowohl der geometrischen Kamerakalibrierung als auch der Erzeugung synthetisch unscharfer Bilder zur Erstellung eines Trainingsdatensatzes. Dabei wird auch auf mögliche Fehlerquellen eingegangen, die sich aus den Modellvereinfachungen ergeben.

Im nächsten Abschnitt wird die Vorgehensweise bei der geometrischen Kamerakalibrierung erläutert sowie deren Ergebnisse vorgestellt. Ziel der Kalibrierung ist die Quantifizierung vorhandener optischer Verzeichnungen. Außerdem wird die Bestimmung der Positionsgenauigkeit des Fokusmechanismus erklärt, welche einen direkten Einfluss auf die erreichbare Genauigkeit der Tiefenmessung hat.

Des Weiteren wird die Entwicklung des multispektralen Beleuchtungssystems beschrieben, das eine optimale Ausleuchtung der Messproben gewährleistet. Im Anschluss wird auf die Erstellung des Testdatensatzes GeoFocus3D eingegangen, einschließlich der Aufnahme- und Vorverarbeitung der Fokalstapel sowie der Messprozedur zur Generierung der Ground-Truth-Tiefenkarten. Darüber hinaus wird der für das Training verwendete Datensatz vorgestellt und die Methode zur Generierung synthetisch unscharfer Bilder beschrieben.

Die implementierten DDFF-Methoden werden im Weiteren einer konventionellen DFF-Methode gegenübergestellt, deren Funktionsweise im Anschluss dargelegt wird. Abschließend erfolgt die systematische Auswahl zweier geeigneter DDFF-Modelle sowie die Beschreibung ihrer konkreten Implementierung.

5.1 Experimenteller Aufbau

Zunächst wird eine Übersicht über die Komponenten und deren Eigenschaften des bestehenden Mikroskopaufbaus gegeben. Anschließend folgt die Vorstellung ein vereinfachten optischen Modells, das den Versuchsaufbau abbildet. Daraufhin werden die Vorgehensweise sowie die Ergebnisse der geometrischen Kamerakalibrierung erläutert. Zudem erfolgt die Charakterisierung der beweglichen Komponente des Aufbaus, des Aktuators. Abschließend wird die Entwicklung des multispektralen Beleuchtungssystems beschrieben und dessen Beleuchtungseigenschaften werden untersucht.

5.1.1 Komponenten des DFF-Mikroskops

Der Aufbau des Kamerasystems ohne LED-Ring ist in der Abbildung 5 dargestellt. Die wichtigsten optischen Parameter sind in der Tabelle 2 zusammengefasst. Als Detektor kommt der Sensor *ASI 183 mm Mono* (ZWO, China) zum Einsatz. Auf der Seite des Detektors ist das Festbrennweitenobjektiv *67-717* (EDMUND OPTICS, Großbritannien) montiert, während sich auf der

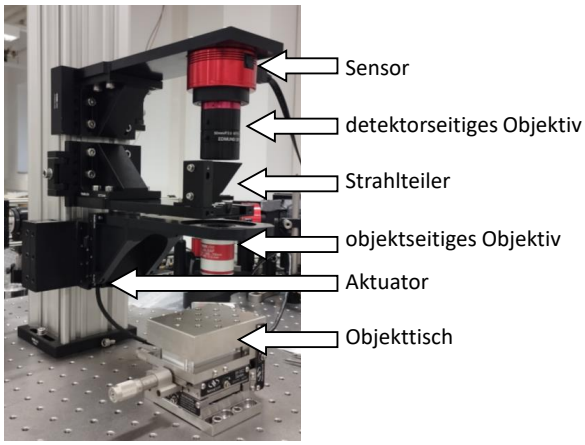


Abbildung 5: Experimenteller Versuchsaufbau

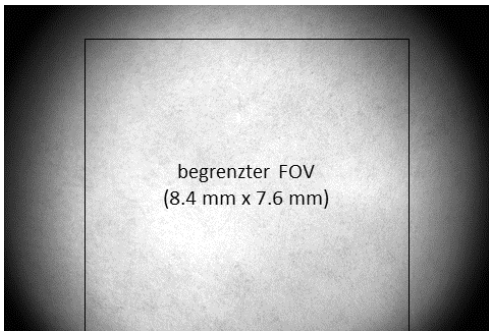


Abbildung 6: Gewählter begrenzter Bildbereich

Tabelle 2: Optische Parameter des Versuchsaufbaus

Optischer Parameter	Wert
Pixelgröße	2.4 µm × 2.4 µm
Sensorgröße	13.1 mm × 8.8 mm
Spektralbereich	650-1000 nm
Durchmesser Blende	20 mm
erwartete Vergrößerung	1
min. DoF	12 µm
räumliche Auflösung	10 µm

Objektseite das Mikroskopobjektiv *TL4x-SAP* (THORLABS, USA) befindet. Die Brennweiten f beider Objektive sind mit 50 mm angegeben.

Zwischen den Objektiven ist eine Blende positioniert, die jedoch konstruktionsbedingt nicht exakt auf der optischen Achse liegt. Dies führt insbesondere zu asymmetrischer Vignettierung im Bild. Zusätzlich ist ein Strahlteiler in das System integriert, der zur Unterdrückung von Umgebungslicht dient. Ein eigens entwickelter multispektraler LED-Ring ist unterhalb des objektseitigen Objektivs montiert.

Das objektseitige Objektiv ist auf einem linearen Aktuator des Typs *PLSZ* (THORLABS, USA) befestigt, der eine Bewegung entlang der z-Achse um bis zu 31 mm ermöglicht. Dadurch kann der Fokusabstand des Gesamtsystems gezielt verändert werden. Die Nullposition des Aktuators ist dabei als dessen oberster Anschlag definiert, welcher dem minimalen Fokusabstand entspricht. Die gewählte Anordnung der Objektive gewährleistet einen linearen Zusammenhang zwischen der Verschiebung des Aktuators z und der resultierenden Veränderung des Fokusabstands.

Die Randbereiche der aufgenommenen Bilder werden durch Vignettierung und optische Verzerrung beeinträchtigt. Da die Blende nicht exakt auf der optischen Achse zentriert ist, tritt eine asymmetrische Vignettierung auf, die in der Nullposition des Aktuators am stärksten ausgeprägt ist. Im Gegensatz dazu sind die optischen Verzerrungen in dieser Position minimal, nehmen jedoch

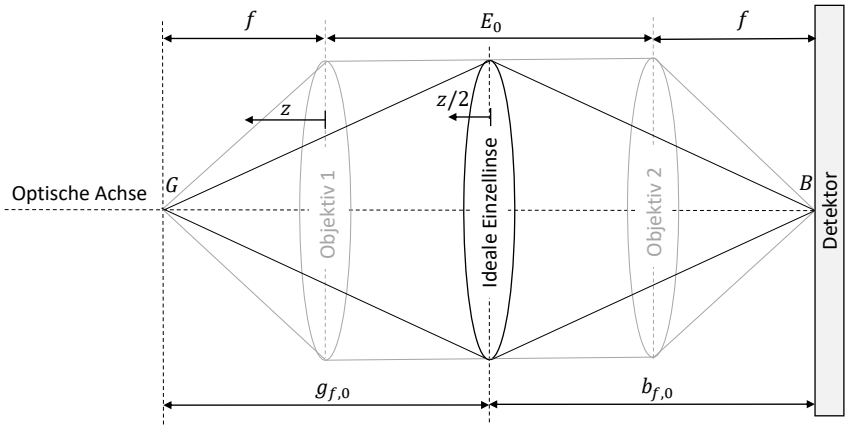


Abbildung 7: Vereinfachtes optisches Modell mit einer idealen Einzellinse

mit zunehmender Verschiebung des Aktuators zu.

Aufgrund dieser Einschränkungen wird für die weiteren Untersuchungen ein begrenzter Bildbereich von 8.4 mm × 7.6 mm herangezogen. Dieser Bereich ist aufgrund der asymmetrischen Vignettierung nicht mittig auf dem Sensor angeordnet. Die exakte Position des verwendeten Bildausschnitts ist in Abbildung 6 anhand einer Aufnahme eines weißen Blatt Papiers dargestellt.

Der Arbeitsabstand des objektseitigen Objektives beträgt im Ausgangszustand 17 mm. Durch die Montage des LED-Rings wird dieser jedoch um 2 mm reduziert, sodass sich ein effektiver Arbeitsabstand – zugleich der maximale Höhenmessbereich – von 15 mm ergibt.

Zur automatisierten Bildaufnahme sind der Detektor, der Aktuator und der LED-Ring mit einem Computer verbunden. Über entsprechende Python-Programmierschnittstellen lassen sich die Komponenten zentral steuern.

5.1.2 Vereinfachtes optisches Modell

Für die folgenden Berechnungen wurde das komplexe optische System des experimentellen Aufbaus durch eine ideale Einzellinse modelliert. Dadurch lässt sich die vereinfachte Linsengleichung auf das System anwenden. Das theoretische Kameramodell ist in Abbildung 7 dargestellt.

Effektive Brennweite. Die Veränderung des Fokusabstands durch das Verschieben des objektseitigen Objektivs kann in diesem Modell als Änderung des Abstands zwischen idealer Linse und Detektor interpretiert werden. Da die Vergrößerung auch bei Bewegung des Aktuators als konstant $M \equiv 1$ angenommen wird, entspricht die Gegenstandsweite eines fokussierten Objektpunkts der Bildweite:

$$g_f(z) = b_f(z) = b_{f,0} + \frac{z}{2} \tag{14}$$

Dabei bezeichnet $b_{f,0}$ den Abstand zwischen der idealisierten Linse und dem Detektor bei Aktuatorposition $z = 0$. Dieser berechnet sich aus dem initialen Abstand E_0 zwischen den Hauptebenen der realen Objektive und deren Brennweite f_o zu:

$$b_{f,0} = f_o + \frac{E_0}{2} \tag{15}$$

Der initiale Abstand zwischen den Objektiven wird mit $E_0 = 100$ mm angenommen. Auf Basis der vereinfachten Linsengleichung und unter Annahme konstanter Vergrößerung ergibt sich, dass sich die effektive Brennweite der idealisierten Linse mit der Aktuatorposition verändert. Die Brennweite f_I in Abhängigkeit von der Position des Aktuators lautet:

$$f_I(z) = \frac{b_f(z)}{2} = \frac{f_o}{2} + \frac{E_0 + z}{4} \tag{16}$$

Die Änderung der Brennweite über den gesamten Messbereich ist in Abbildung 8 dargestellt. Die maximale Änderung bei der Aktuatorposition $z = 15\,\text{mm}$ beträgt 3.57 mm.

Circle of Confusion. Unter Verwendung der Gleichungen 14 und 16 ergibt sich für den Fokusabstand:

$$g_f(z) = 2f_I(z) \tag{17}$$

Der CoC-Durchmesser aus Gleichung 4 vereinfacht sich damit zu:

$$CoC(z,d) = A \cdot \frac{|g(z,d) - g_f(z,d)|}{g(z,d)} \text{ mit } g(z,d) = g_f(z) + d$$

$$\rightarrow CoC(z,d) = A \cdot \frac{|d|}{f_o + E_0/2 + z/2 + d} \tag{18}$$

Dabei bezeichnet d die Tiefe, welche hier als der gerichteten Abstand eines Objektpunkts zur Fokusebene definiert ist. Objektpunkte in der Fokusebene werden scharf auf der Sensorebene abgebildet. In Abbildung 9 ist der CoC-Durchmesser in Abhängigkeit von der Objekttiefe d bei Aktuatorposition $z = 0$ dargestellt. Der Graph verdeutlicht, dass die Zuordnung der Unschärfe zur Objekttiefe nicht eindeutig ist. Einem bestimmten CoC-Durchmesser können zwei unterschiedliche Tiefen zugeordnet werden.

Aus der Gleichung 18 geht hervor, dass der CoC-Durchmesser nicht nur von der Objekttiefe, sondern auch von der Aktuatorposition abhängig ist. Die maximale Abweichung des CoC-Durchmessers zwischen $z = 0$ und $z = 15\,\text{mm}$ für unterschiedliche Objekttiefen ist in Abbildung 10 visualisiert.

Da die Blende nicht exakt auf der optischen Achse liegt, stimmt der Mittelpunkt des CoC im vereinfachten Modell nicht mit dem realen Mittelpunkt der PSF überein. Es lässt sich beobachten, dass sich der PSF-Mittelpunkt bei positiver Objekttiefe in Richtung des oberen Bildrands verschiebt, bei negativer Objekttiefe hingegen zum unteren Rand.

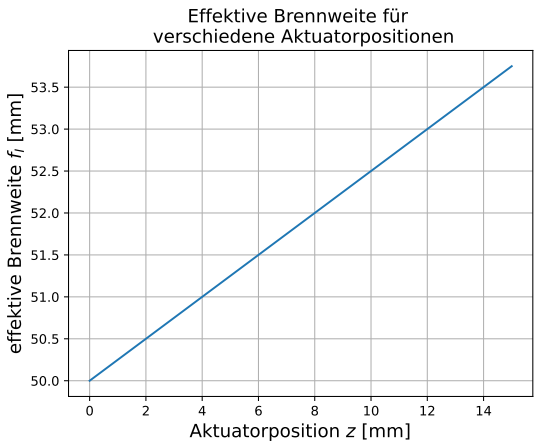


Abbildung 8: Veränderung der effektive Brennweite des Kamerasystems bei Bewegung des Aktuators

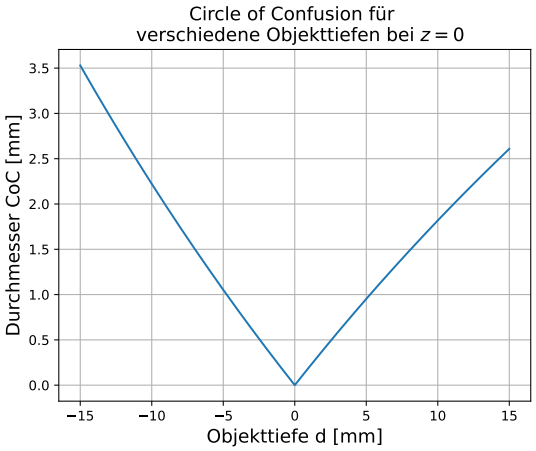


Abbildung 9: Durchmesser des CoC für verschiedene Objekttiefen bei der Nullposition des Aktuators

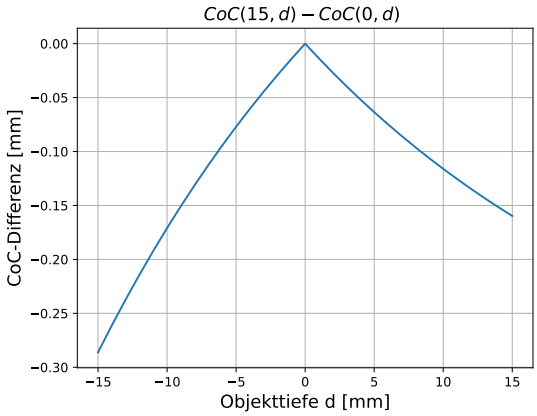


Abbildung 10: Maximale Abweichung des Durchmessers des CoC bei Aktuatorbewegung für verschiedene Objekttiefen

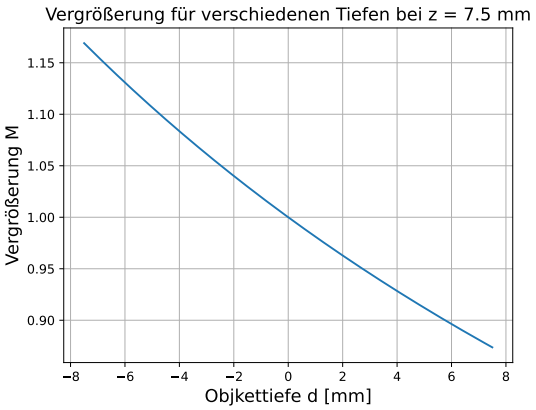


Abbildung 11: Veränderung der Vergrößerung unscharfer Bildpunkte bei verschiedenen Objekttiefen

Vergrößerung. Das optische System wurde so konzipiert, dass fokussierte Objektpunkte bei Bewegung des Aktuators mit konstanter Vergrößerung abgebildet werden. Unscharfe Punkte jedoch werden in Kameranähe vergrößert, in größerer Entfernung hingegen verkleinert. Die Vergrößerung M als Funktion von Aktuatorposition z und Objekttiefe d ergibt sich zu:

$$M(z,d) = \frac{b(z)}{g(z,d)} = \frac{f_I(z)}{g(z,d) - f_I(z)} = \frac{f_o/2 + E_0/4 + z/4}{f_0/2 + E_0/4 + z/4 + d} \tag{19}$$

Ein Beispiel für die Vergrößerung über den Messbereich bei einer Aktuatorposition von $z = 7.5\,\text{mm}$ ist in Abbildung 11 dargestellt.

5.1.3 Geometrische Kamerakalibrierung

Ziel der geometrischen Kamerakalibrierung ist die Bestimmung der intrinsischen Kameraparameter, der Bildverzeichnung sowie deren Veränderung bei variierendem Fokusabstand. Für das DFF-Verfahren ist es essenziell, dass Objektpunkte in allen Bildern eines Fokalstapels auf identische Pixelkoordinaten abgebildet werden. Nur so kann der maximale Fokuswert je Pixel zuverlässig ermittelt und daraus die korrekte Tiefe berechnet werden. Mithilfe der ermittelten Kalibrierparameter lassen sich nahezu verzeichnungsfreie Bilder erzeugen, was auch für den Vergleich mit Verifizierungsmessungen erforderlich ist. Für die Simulation von Unschärfeeffekten ist zudem die Kenntnis der Brennweite notwendig.

Die Kalibrierung erfolgte mittels eines Gittermusters, wobei die geringe Schärfentiefe des Kamerasystems die Varianz der möglichen Posen begrenzt. Für eine zuverlässige Separierung der Schätzung von Brennweite und Objektabstand ist jedoch eine signifikante Variation des Abstands zwischen Muster und Kamera erforderlich (Wohlfeil u. a. 2019). Andernfalls können die Ergebnisse sehr ungenau sein.

Zur Verbesserung der Stabilität wurde deshalb die Anzahl der zu schätzenden Parameter reduziert. Anstelle einer gleichzeitigen Bestimmung von Objektabstand und Brennweite wird ein vereinfachtes Modell verwendet: Es wird ein konstanter Abstand für alle Posen angenommen, und lediglich ein Skalierungsfaktor geschätzt. Dadurch wird eine robuste Kalibrierung auch bei eingeschränkter Schärfentiefe möglich.

Messungen. Für die Kalibrierung wurde das Gittermuster des *Combined Resolution and Distortion Test Targets* (THORLABS, USA) auf weißem Papier verwendet. Das Muster umfasst 21×21 Objektpunkte mit einem Abstand von 100 µm und ist damit deutlich kleiner als das Kamerabildfeld (siehe Abbildung 13). Um eine vollständige Abdeckung zu gewährleisten, wurde das Muster systematisch über das gewählte Bildfeld verschoben. Insgesamt wurden zwölf Aufnahmen erfasst. Um die Gitterstruktur hervorzuheben, wurde eine leichte Überbelichtung eingesetzt, um den weißen Hintergrund zu überstrahlen und Störmuster des Papiers zu minimieren.

Zusätzlich erfolgten Messungen bei vier verschiedenen Aktuatorpositionen (0.4 mm, 5 mm, 10 mm und 15 mm). Der Abstand des Kalibrierungsmusters wurde jeweils so gewählt, dass das Gitter fokussiert war.

Detektion der Gitterpunkte. Die auf dem Sensor projizierten Gitterpunkte wurden mithilfe automatisierter Bildverarbeitung detektiert. Eine Übersicht zu den Schritten des verwendeten Algorithmus ist in der Abbildung 12 dargestellt. Nach dem Zuschneiden der Aufnahmen auf das relevante Gitterfeld wurde ein adaptives Schwellwertverfahren angewendet, um ein binäres Bild zu erzeugen. Der Schwellwert wurde als gauß-gewichtete Summe der lokalen Umgebungspixel abzüglich einer Konstante berechnet. Der morphologische Operator Closing diente zur Reduktion von Rauschen und Artefakten. Die Eckpunkte wurden mithilfe des Harris-Corner-Detektors

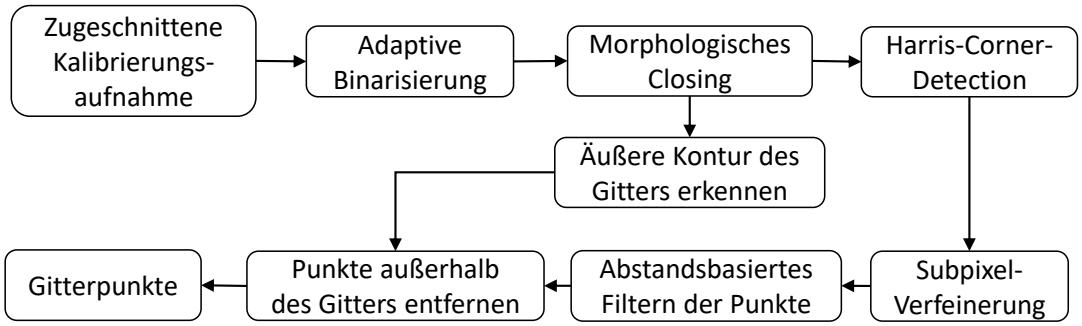


Abbildung 12: Algorithmus zur Detektion der Gitterpunkte

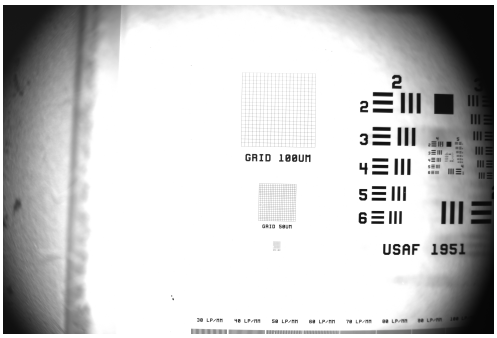


Abbildung 13: Kalibrierungsaufnahme des Gittermusters

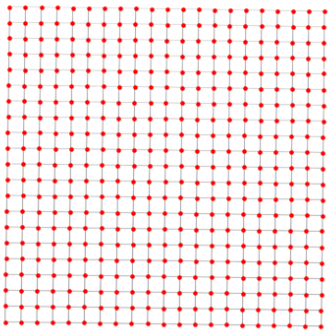


Abbildung 14: Detektierte Gitterpunkte im Kalibrierungsmuster

(Harris und Stephens 1988) und einer iterativen Subpixel-Optimierung nach Foerstner (1987) präzisiert. Detektierte Punkte mit geringerem Abstand zueinander als ein definierter Schwellwert wurden zusammengefasst bzw. entfernt. Punkte außerhalb des Gitters wurden ebenfalls gefiltert. Die gefilterten detektierten Punkte einer Beispielaufnahme sind in der Abbildung 14 dargestellt.

Least-Squares-Optimierung. Zur Schätzung der Kameraparameter wurde eine nichtlineare Least-Squares-Optimierung durchgeführt. Grundlage ist das Lochkameramodell mit einem radialem Verzeichnungsparameter. Für jede Aktuatorposition wurde die Optimierung separat durchgeführt.

Das Modell $M_{T,n}(X_n, \mathbf{u})$ beschreibt die Abbildung der Objektpunkte $\mathbf{u}$ auf die detektierten Bildpunkte $\mathbf{y_n}$ mittels einer Kombination aus affiner Transformation und radialer Verzeichnung. Der Index $n = 1...12$ bezeichnet dabei die Bildnummer des verschobenen Gittermusters. Die affine Transformation beinhaltet Rotation R mit dem Rotationswinkel θ, Skalierung S mit dem Skalierungsfaktor s und Translation T_n mit den Translationsdistanzen $v_{x,n}$ und $v_{y,n}$. Rotation und Skalierung wurden für alle Aufnahmen einer Aktuatorposition als konstant angenommen, während die Translationen aufgrund der Gitterverschiebung individuell bestimmt wurden. Die radiale Verzeichnung wird durch einen Parameter k_1 und den Bildhauptpunkt (p_x, p_y) beschrieben. Weitere

Verzeichnungsparameter führte zu Instabilitäten und wurde daher nicht berücksichtigt. Der Vektor $X_n = \begin{bmatrix}\theta & s & v_{x,n} & v_{y,n} & k_1 & p_x & p_y\end{bmatrix}$ beinhaltet die gesuchten Parameter. Das Abbildungsmodell wird beschrieben mit:

$$M_{T,n}(X_n, \mathbf{u}) = \underbrace{\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}}_{\mathbf{y_n}} = (1 + k_1 r^2) \begin{bmatrix} x_{c,n} - p_x \\ y_{c,n} - p_y \\ 0 \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \text{ mit } r^2 = x_{c,n}^2 + y_{c,n}^2 \tag{20}$$

$$\underbrace{\begin{bmatrix} x_{c,n} \\ y_{c,n} \\ 1 \end{bmatrix}}_{\mathbf{u_{c,n}}} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}} \cdot \underbrace{\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \underbrace{\begin{bmatrix} 1 & 0 & v_{x,n} \\ 0 & 1 & v_{y,n} \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}_n} \cdot \underbrace{\begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}}_{\mathbf{u}} \tag{21}$$

Die Zielfunktion ergibt sich aus der Minimierung der euklidischen Abweichung zwischen detektieretn und projizierten Bildpunkten über alle m Gitterpunkte:

$$E = \sum_{i=1}^{m} \|M_T(x, u_i) - y_i\|^2 \tag{22}$$

Die Optimierung erfolgte mit der Trust-Region-Reflective-Methode (The-SciPy-Community 2025) aus dem `SciPy`-Paket. Die numerische Berechnung der Jacobi-Matrix zeigte sich als flexibel und ausreichend genau im Vergleich zur analytischen Variante. Startwerte wurden manuell bestimmt. Für die Detektion der Gitterpunkte und die Least-Squares-Optimierung wurde das Programm `Camera-Calibration.py` verwendet.

Ergebnisse der geometrischen Kamerakalibrierung. Der Reprojektionsfehler bezeichnet die Differenz zwischen detektierten und durch das Modell berechneten Bildpunkten. Die mittleren Reprojektionsfehler der analysierten Aktuatorpositionen sind in Tabelle 3 dargestellt. Für alle überprüften Fokusdistanzen liegen die mittleren Fehler deutlich unter einem Pixel. Abweichungen lassen sich durch Modellvereinfachungen sowie potenzielle Abbildungsfehler erklären, die nicht durch einen einzelnen Verzeichnungsparameter beschrieben werden können. Der Fehler über den betrachteten Bildbereich bei der Ausgangsposition des Aktuators ist in Abbildung 15 visualisiert. Der verbleibende Fehler scheint annähernd zufällig über den Bildbereich verteilt zu sein.

Aktuatorpostion z in [mm]	0.4	5.0	10.0	15.0
Reprojektionsfehler in [Pixel]	0.26	0.25	0.25	0.26

Tabelle 3: Reprojektionsfehler der geometrischen Kamerakalibrierung

Die interpolierten Kameraparameter über die Aktuatorposition sind in der Abbildung 16 dargestellt. Es kann eine zunehmende radiale Verzeichnung mit wachsendem Fokusabstand beobachtet

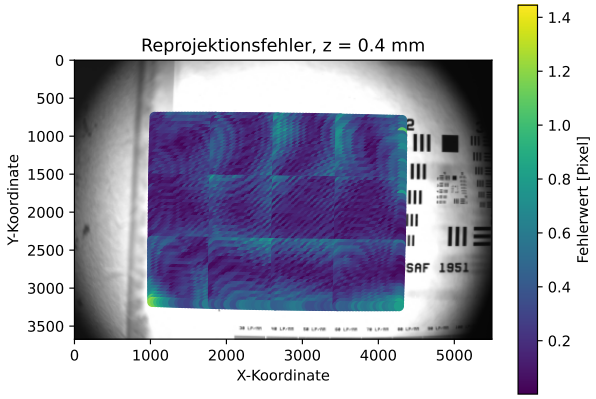


Abbildung 15: Reprojektionsfehler bei der Aktuatorposition z = 0.4 mm

werden. Der Bildhauptpunkt liegt nicht im geometrischen Zentrum des Sensors, sondern ist verschoben. In der Aktuatorposition $z = 0$ beträgt die Verschiebung $(\Delta p_x, \Delta p_y) = (0.28\ \text{mm}, 0.09\ \text{mm})$. Diese Verschiebung nimmt mit der Aktuatorposition zu, was auf eine leichte Verkippung oder Dezentrierung der Objektive hindeutet.

Die Skalierung bleibt über alle Positionen hinweg nahezu konstant. Die maximale Abweichung beträgt lediglich $3 \cdot 10^{-4}$, was eine Annahme konstanter Vergrößerung im Modell rechtfertigt. Der ermittelte mittlere Skalierungsfaktor liegt mit 0.99 nahe am erwarteten Wert von 1.

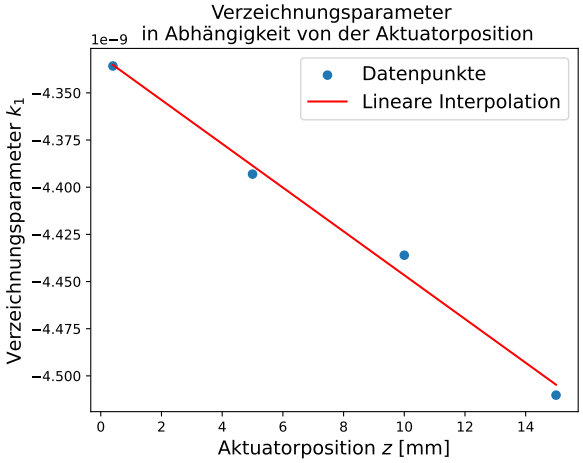
Die Schätzung der effektiven Brennweite konnte mit der gewählten Kalibrierungsstrategie nicht realisiert werden. Die in Abschnitt 5.1.2 getroffenen Annahmen zur Brennweite und zum Objektivabstand konnten daher nicht validiert werden. Dennoch ermöglichen die gewonnenen Parameter eine verzeichnungsfreie Abbildung und Skalierungskorrektur.

5.1.4 Charakterisierung des Aktuators

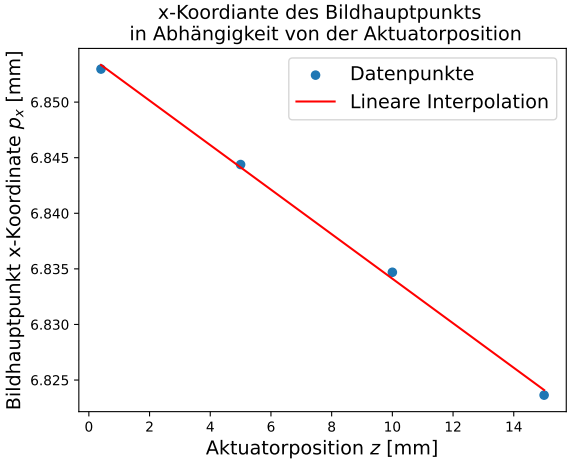
Das Ziel der Charakterisierung des Aktuators bestand darin, die Genauigkeit seiner Positionierung zu bestimmen. Die maximale Tiefenauflösung des DFF-Verfahrens wird wesentlich durch die Genauigkeit der bekannten Fokusabstände begrenzt.

Zur Charakterisierung wurde der konfokale Wegmesssensor *CL-L070* (KEYENCE, Japan) oberhalb der Aktuatorplattform montiert. Gemessen wurde die Veränderung der Distanz zwischen dem Sensor und der Aktuatoroberfläche während der Bewegung.

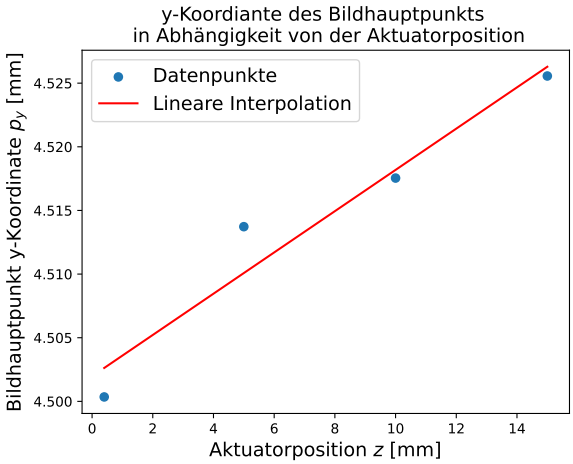
Der Sensor arbeitet nach dem Prinzip der chromatischen Konfokaltechnik: Ein Lichtstrahl mit einer bekannten spektralen Verteilung wird emittiert, wobei jede Wellenlänge eine spezifische Brennweite besitzt. Nur das Licht jener Wellenlänge, das exakt auf der Aktuatoroberfläche fokussiert ist, kann die Lochblende des Sensors passieren. Das reflektierte Licht wird über ein Spektrometer ausgewertet, wodurch die exakte Entfernung zur Oberfläche bestimmt wird. Der eingesetzte Wegsensor ist in der Lage, Abstände mit einer Auflösung von 0.25 µm zu messen.



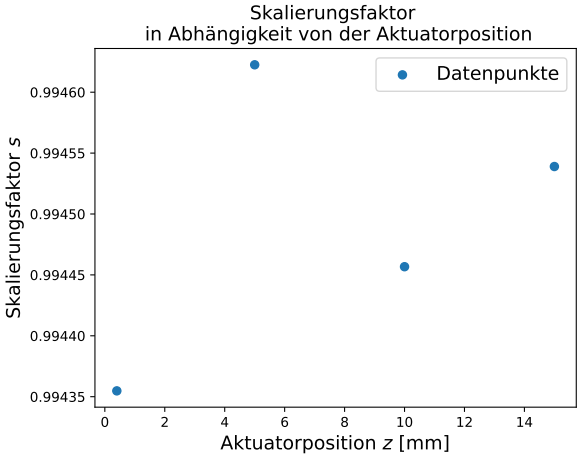
(a) Verzeichnungsparamater k_1



(b) X-Koordinate des Bildhauptpunkts p_x



(c) Y-Koordinate des Bildhauptpunkts p_y



(d) Skalierungsfaktor s

Abbildung 16: Kalibrierte Kameraparameter in Abhängigkeit von der Aktuatorposition

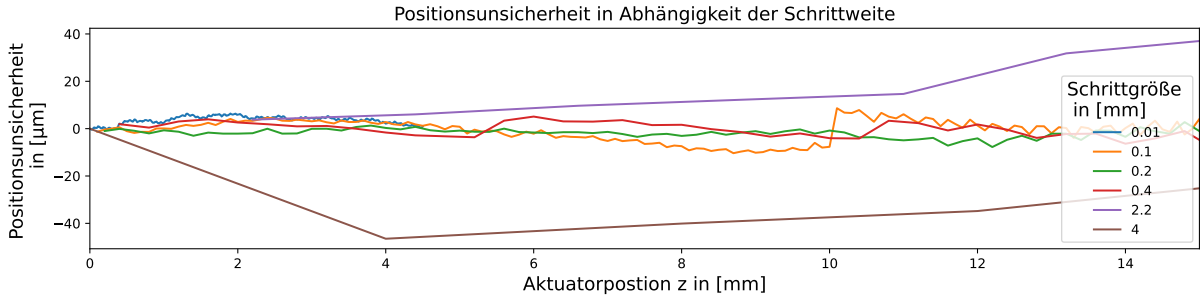


Abbildung 17: Positionsungenauigkeit des Aktuators für verschiedene Schrittweiten

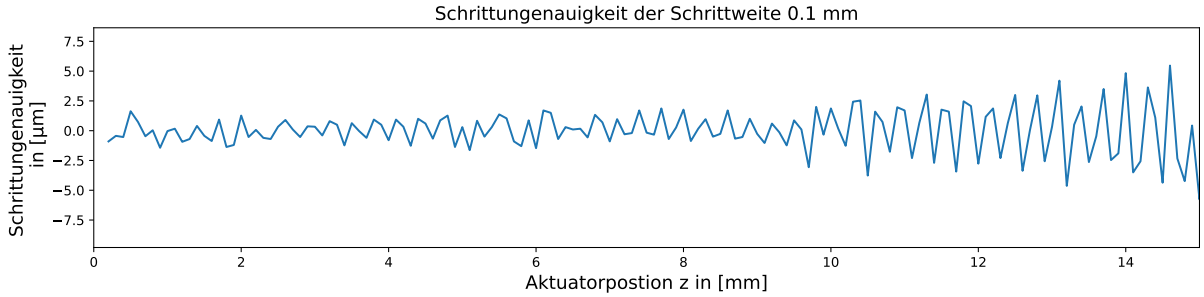


Abbildung 18: Schrittungenauigkeit des Aktuators bei einer Schrittweite von 0.1 mm

Zur Bestimmung der Positioniergenauigkeit wurde der Aktuator mit unterschiedlichen Schrittweiten verfahren, während die jeweilige Position mit dem Wegmesssensor erfasst wurde. Vor Beginn jeder Messreihe wurde der Sensor auf Null kalibriert. Die Differenz zwischen Soll- und Ist-Position ist in Abbildung 17 dargestellt. Es zeigte sich, dass die Positionsunsicherheit mit zunehmender Schrittweite zunimmt. Zudem nimmt die Positioniergenauigkeit mit wachsender Entfernung von der Nullposition des Aktuators ab.

Für die Aufnahme der Fokalstapel wurde eine Schrittweite von 0.1 mm gewählt. In dem Messbereich wurde für diese Schrittweite eine maximale Positionsunsicherheit von 8.6 µm festgestellt. Damit ist die Unsicherheit kleiner als die Schärfentiefe des Kamerasystems. Die erreichbare Tiefenauflösung des Versuchsaufbaus wird somit also von der Schärfentiefe begrenzt.

Zusätzlich wurde die Schrittgenauigkeit bei einer Schrittweite von 0.1 mm analysiert. Dabei wurde die vom Sensor gemessene Verschiebung nach jedem Schritt aufgezeichnet. Die Ergebnisse sind in Abbildung 18 dargestellt. Die Standarabweichung der Schrittgröße beträgt 3.1 µm, wobei eine maximale Abweichung von -9.0 µm beobachtet wurde.

Auch das mechanische Spiel des Aktuators wurde an drei verschiedenen Positionen untersucht. Dazu wurde der Aktuator zweimal mit einer Schrittweite von 0.1 mm in eine Richtung verfahren und anschließend ein Schritt zurück ausgeführt. Die Differenz zur erwarteten Position wurde erfasst. Die Ergebnisse sind in Tabelle 4 dargestellt. Die gemessenen Werte liegen innerhalb der Standardabweichung der Schrittgenauigkeit, sodass ein eventuell vorhandenes Spiel als vernachlässigbar eingestuft werden kann.

Tabelle 4: Mechanisches Spiel des Aktuators an drei Positionen

Aktuatorposition in [mm]	0.2	12.2	22.2
Spiel in [µm]	1.5	3.1	1.0

Darüber hinaus wurde die Wiederholgenauigkeit beim Anfahren der Nullposition geprüft. Dazu wurde die Nullposition aus drei unterschiedlichen Startpositionen mehrfach angefahren und die erreichten Endpositionen verglichen. Es wurde eine mittlere Abweichung der Position von 21.2 µm ermittelt. Zusätzlich wurde die Wiederholgenauigkeit beim Anfahren eines etwas von der Nullposition entfernten Punktes untersucht. Für eine Zielposition $z = 0.4$ mm konnte eine mittlere Wiederholgenauigkeit von 0.3 µm erzielt werden. Um eine möglichst hohe Genauigkeit der Fokusabstände in den ersten Bildern eines Fokalstapels sicherzustellen, wurde daher die Startposition auf 0.4 mm festgelegt.

5.1.5 Multispektrale LED-Beleuchtung

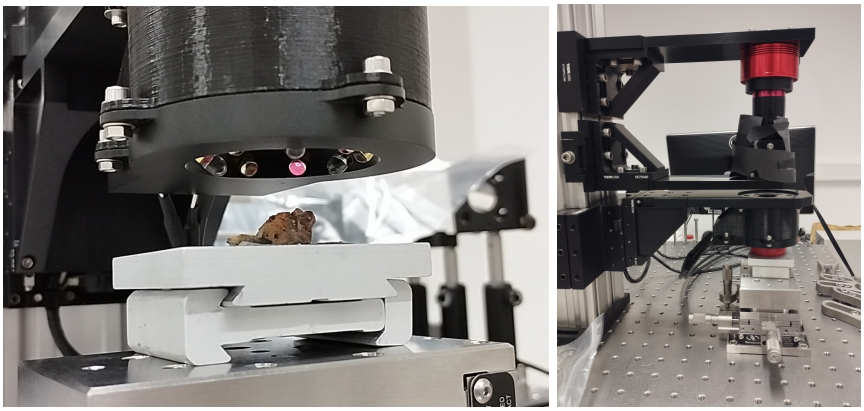
Die Verwendung einer multispektralen Beleuchtung ermöglicht es, mit einem monochromen Kamerasensor Informationen über die spektrale Reflektivität eines Objekts zu gewinnen. Diese beschreibt, welche Wellenlängen elektromagnetischer Strahlung von einem Material reflektiert werden. Unterschiedliche Materialien besitzen charakteristische spektrale Signaturen, anhand derer eine Klassifizierung möglich ist. Im Gegensatz zu Verfahren mit Farbfiltern, wie etwa dem Bayer-Filter zur Unterscheidung von Rot, Grün und Blau, bleibt bei der multispektralen Beleuchtung die volle laterale Auflösung des Sensors erhalten.

Durch Zuweisung der verschiedenen Spektralkanäle zu den RGB-Farbkanälen können aus den aufgenommenen Einzelbildern Falschfarbenbilder generiert werden. Falschfarbenbilder dienen zur besseren Veranschaulichung der Daten und können für eine Analyse genutzt werden. Weitere Erklärungen dazu finden sich im Anhang.

Multispektrale LED-Beleuchtungssysteme arbeiten in der Regel so, dass verschiedenfarbige LEDs das Objekt nacheinander belichten (Shrestha und Hardeberg 2013). LEDs bieten dabei Vorteile wie eine schnelle, computergestützte Ansteuerung, hohe Robustheit, breite Wellenlängenverfügbarkeit und ein gutes Kosten-Nutzen-Verhältnis.

Die Entwicklung der multispektralen Beleuchtung für den experimentellen Aufbau kann in drei wesentliche Schritte gegliedert werden: die optische Gestaltung einschließlich der Auswahl der LEDs, die Konstruktion sowie die Fertigung der mechanischen Struktur, und die Planung und Implementierung der elektrischen Steuerung. Der am DFF-Mikroskop installierte LED-Ring ist in Abbildung 19 veranschaulicht.

Optische Gestaltung. Die spektrale Sensitivität des verwendeten Detektors sowie die Spektralbereiche der gewählten LEDs sind in Abbildung 20 dargestellt. Der Einsatz des nahinfraroten Spektralbereichs reduziert den Einfluss von Umgebungslicht in der Laborumgebung. Es wurden



(a) Ansicht von schräg unten (b) Seitenansicht

Abbildung 19: Montierte LED-Beleuchtung

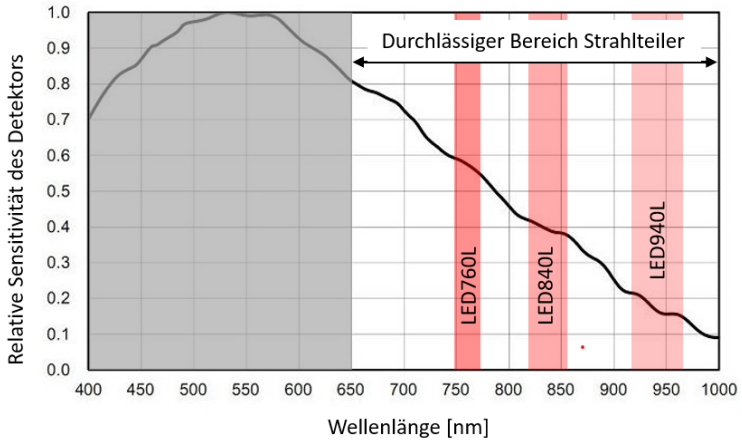


Abbildung 20: Spektrale Sensitivität des Versuchsaufbaus und Spektralkanäle des LED-Rings

drei LEDs ausgewählt, deren Spektren den empfindlichen Bereich des Kamerasystems möglichst gleichmäßig abdecken. Die Eigenschaften dieser LEDs von THORLABS, USA sind in Tabelle 5 zusammengefasst.

Die Auswahlkriterien für die LEDs lauteten:

- Hohe optische Leistung zur Reduzierung der Belichtungszeit,
- kleiner Abstrahlwinkel zur Maximierung der Beleuchtungsintensität im Bildbereich,
- möglichst gleiche Abstrahlwinkel, um eine vergleichbare Ausleuchtung der Szene zu gewährleisten,
- kurze Lieferzeit,
- niedrige bis mittlere Kosten.

Tabelle 5: Wellenlänge, Bandbreite und Strahlwinkel der LEDs des Beleuchtungsrings

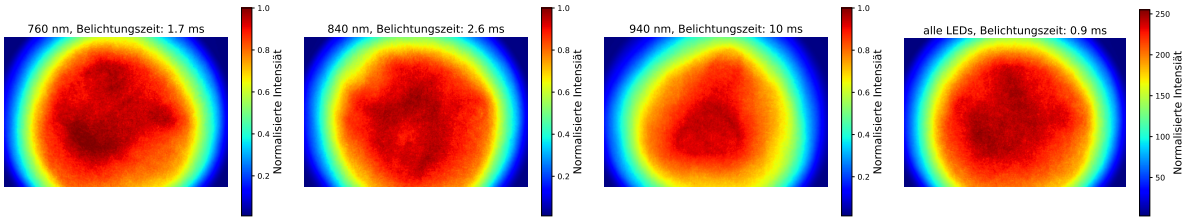
Bezeichnung	Wellenlänge	Bandbreite	Halber Strahlwinkel
LED760L	760 nm	24 nm	12°
LED840L	840 nm	35 nm	12°
LED940E	940 nm	50 nm	10°

Zur Erzeugung einer homogenen Ausleuchtung wurden für jede Wellenlänge drei LEDs radial angeordnet. Der Beleuchtungswinkel $\beta = 40°$ der LEDs wurde anhand der in Abbildung 22 dargestellten Geometrie berechnet mit:

$$\beta = \tan^{-1}\left(\frac{\mathrm{WD}}{R}\right) \tag{23}$$

Dabei bezeichnet WD den Arbeitsabstand und α den Strahlwinkel der LEDs. Für α wurde der Wert von 12° gewählt, der den Strahlwinkeln der 760-nm- und 840-nm-LEDs entspricht. Der radiale Abstand R zwischen den LEDs und der optischen Achse wurde auf 20.5 mm festgelegt, um ausreichend Bauraum zwischen Objektiv und LEDs zu gewährleisten.

Mit dem gewählten Beleuchtungswinkel und LED-Ring-Radius wird ein Bereich mit dem Radius $R_B = 13$ mm von allen LEDs abgedeckt. Damit wird der begrenzte FOV vollständig beleuchtet. Die Verteilung der Lichtintensität wurde mittels Bestrahlung eines weißen Papiers untersucht. Die normalisierten Intensitätsprofile der drei Spektralbereiche sowie bei simultaner Aktivierung aller LEDs über die gesamte Detektorfläche sind in Abbildung 21 präsentiert. Zur Ermittlung der Intensitäten wurden jeweils drei Aufnahmen gemittelt, während die Belichtungszeit so angepasst wurde, dass in jeder Konfiguration eine Sättigung der Intensität erreicht wird.



(a) 760 nm (b) 840 nm (c) 940 nm (d) alle LEDs

Abbildung 21: Intensitätsprofile der drei Spektralkanäle des LED-Rings sowie bei Aktivierung aller LEDs

Mechanischer Aufbau. Die Struktur des LED-Rings besteht aus einer Ober- und einer Unterseite. Die Unterseite enthält passgenaue Bohrungen zur Aufnahme der LEDs und wurde aus wärmeleitfähigem Aluminium gefertigt, um entstehende Wärme effizient abzuleiten. Eine schwarze Eloxierung minimiert reflektiertes Streulicht. Die Oberseite wurde zur Kostenreduktion im Fused Deposition Modelling (FDM) aus schwarzem Polylactid (PLA) additiv gefertigt. Eine

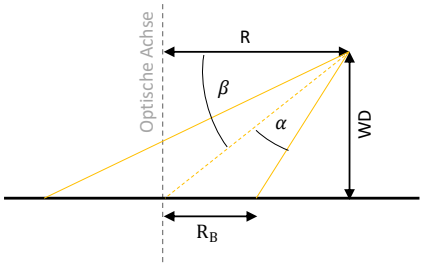


Abbildung 22: Schematische Darstellung des Beleuchtungswinkels des LED-Rings

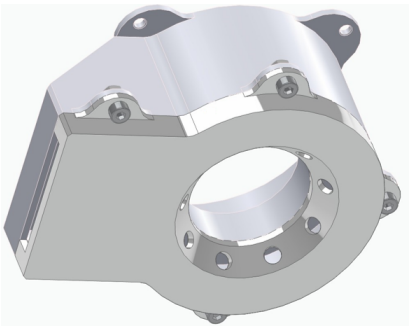


Abbildung 23: CAD-Darstellung des LED-Rings

CAD-Darstellung des LED-Rings ist in Abbildung 23 dargestellt.

Elektrische Ansteuerung Die elektrische Verbindung der LEDs erfolgte über gelötete Steckverbindungen auf einer Platine. Die Spannungsversorgung erfolgt über einen 5 V-Anschluss. Vor jede LED wurde ein 51 Ω-Widerstand geschaltet. Die relevanten elektrischen Parameter sowie die daraus resultierenden Strahlungsleistungen sind in Tabelle 6 angegeben. Diese Werte basieren auf Herstellerdaten und wurden durch lineare Interpolation aus Spezifikationsdiagrammen abgeschätzt.

Tabelle 6: Relevante elektrische Parameter der LEDs bei einem Vorwiderstand von jeweils 51 Ω

LED-Bezeichung	Durchlassspannung	Betriebsstrom	Optische Leistung
LED760L	1.9 V	60 mA	29 mW
LED840L	1.7 V	64 mA	28 mW
LED940L	1.3 V	72 mA	14 mW

Die Steuerung der LEDs erfolgt über den *YKUSH USB Switchable Hub* (YEPKIT, Portugal) mittels einer Python-Schnittstelle. Die drei Upward-Ports des Hubs können einzeln geschaltet werden. Jeder Port steuert drei LEDs derselben Wellenlänge, sodass jedem Spektralbereich ein Port zugeordnet ist.

5.2 GeoFocus3D-Datensatz

Zur Erstellung des realen GeoFocus3D-Datensatzes standen elf verschiedene Gesteinsproben zur Verfügung. Zwei dieser Proben wurden jeweils von zwei Seiten vermessen, sodass der Datensatz insgesamt 13 Sets umfasst. Jedes Set besteht aus einem Fokalstapel, einer Ground-Truth-Tiefenkarte sowie einem AiF-RGB-Bild.

5.2.1 Fokalstapel

Messprozedur. Die Gesteinsproben wurden nacheinander auf einem verfahrbaren Messtisch unter dem Mikroskopaufbau positioniert. Für den Aktuator wurde jeweils die Startposition $z = 0.4$ mm gewählt, um eine hohe Wiederholgenauigkeit sicherzustellen (vgl. Kapitel 5.1.4). Anschließend wurde die Höhe des Messtisches so eingestellt, dass sich die Fokusebene des Kamerasystems knapp oberhalb des höchsten Punkts der Probe befand. Der Aktuator wurde dann schrittweise nach unten bewegt, bis keine Bereiche der Probe mehr fokussiert abgebildet waren.

Vor jedem Schritt wurden vier Aufnahmen unter variierender Beleuchtung gemacht: mit den drei Spektralbereichen des LED-Rings sowie bei gleichzeitiger Aktivierung aller LEDs. Die Belichtungszeit wurde individuell angepasst, sodass die Intensität im fokussierten Bereich des Bildzentrums gerade die Sättigung erreichte. Eine Übersicht der verwendeten Belichtungszeiten und Größen der Fokalstapel ist im Anhang tabellarisch aufgeführt.

Für die anschließende Tiefenschätzung wurden ausschließlich die Aufnahmen mit 760 nm Beleuchtungswellenlänge verwendet. Es wird angenommen, dass die spektrale Zusammensetzung des Beleuchtungslichts bei der DFF-Methode keinen signifikanten Einfluss auf die Qualität der Tiefenrekonstruktion hat.

Wahl der Schrittweite. Der Fokusabstand zwischen aufeinanderfolgenden Bildern des Fokalstapels ist entscheidend für die Genauigkeit der Tiefenschätzung sowie für den Zeit- und Speicherbedarf der DFF-Verarbeitung (Ceruso u. a. 2021; Yang, Huang und Z. Zhou 2022). DFF-Methoden basieren auf der Annahme, dass für jeden Pixel exakt ein Bild im Stapel existiert, in dem dieser maximal fokussiert ist. Um dies zu gewährleisten, sollten sich die fokussierten Bereiche der Bilder möglichst nicht überlappen – gleichzeitig darf der Abstand nicht zu groß sein, um die Fokuskurve hinreichend aufzulösen.

Die theoretisch optimale Schrittweite entspräche der Schärfentiefe von ca. 12 µm. Um jedoch einen Messbereich von 15 mm vollständig abzudecken, wären damit 1250 Bilder notwendig – verbunden mit erheblichem Zeit-, Speicher- und Rechenaufwand. Als Kompromiss wurde eine Schrittweite von 100 µm gewählt, was zu einem Fokalstapel mit 150 Bildern führt.

5.2.2 Bildvorverarbeitung

Die Vorverarbeitung der Aufnahmen umfasst drei Schritte: Verzeichnungskorrektur, Zuschneiden und Größenanpassung. Beispielaufnahmen aus einem vorverarbeiteten Fokalstapel sind in der Abbildung 24 dargestellt.

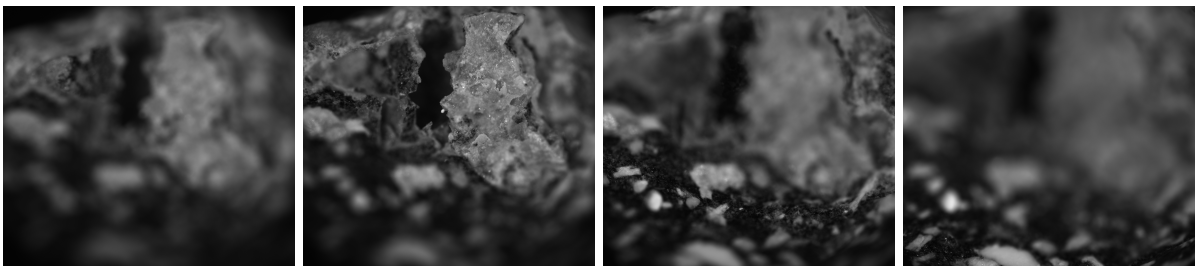
Verzeichnungskorrektur. Die Korrektur der Bildverzeichnung erfolgte mit der `undistort`-Funktion aus der OpenCV-Bibliothek. Auf Basis der kalibrierten Kameraparameter wurde die Kameramatrix `mtx` und die Verzeichnungsmatrix `dist` definiert:

$$\texttt{mtx} = \begin{bmatrix} s & 0 & p_x(z) \\ 0 & s & p_y(z) \\ 0 & 0 & 1 \end{bmatrix}, \quad \texttt{dist} = \begin{bmatrix} k_1(z) & 0 & 0 & 0 & 0 \end{bmatrix}$$

Zuschneiden. Die Verzeichnungskorrektur führt zu einer Wölbung der Bildränder. Um rechteckige, reguläre Bilder zu erhalten, wurden die Aufnahmen zugeschnitten. Da sich die Verzeichnung mit der Aktuatorposition verändert, wurde der maximale notwendige Zuschnitt ermittelt und einheitlich auf alle Bilder angewendet. Zusätzlich wurde die Bilder auf den beschränkten Bildbereich zugeschnitten, welcher in Kapitel 5.1 definiert wurde.

Größenanpassung. Aus der geometrischen Kamerakalibrierung ging hervor, dass der reale Abbildungsmaßstab des Systems leicht von 1 abweicht. Fokussierte Objekte erscheinen um den Faktor $s = 0.99$ verkleinert. Zur Korrektur wurden alle Bilder mit dem Faktor $1/s$ skaliert.

Die ursprüngliche Auflösung in x- und y-Richtung war zudem ca. dreimal höher als die der mit dem Streifenprojektionsmikroskop erzeugten Ground-Truth-Tiefenkarten. Um die mit DFF geschätzten Tiefenkarten direkt vergleichen zu können, wurden die Aufnahmen der Fokalstapel auf dieselbe Auflösung heruntergerechnet.



(a) $z = 0.4$ mm (b) $z = 2.0$ mm (c) $z = 4.0$ mm (d) $z = 6.0$ mm

Abbildung 24: Beispielbilder aus einem Fokalstapel des GeoFocus3D-Datensatzes

5.2.3 Ground-Truth-Tiefenkarten und All-in-Focus-Bilder

Die Erstellung der Ground-Truth-Tiefenkarten erfolgte mit dem Streifenprojektionsmikroskop *VR-5200* (KEYENCE, Großbritannien). Für die Aufnahmen wurde der Vergrößerungsmodus 40× verwendet. Dies ermöglicht eine Tiefenauflösung von 1 µm. Das Sichtfeld ist deutlich kleiner als beim experimentellen Aufbau. Um dennoch einen vergleichbaren Objektbereich zu erfassen, wurden jeweils sechs Einzelaufnahmen über die integrierte Stitching-Funktion zusammengesetzt. Diese Funktion beeinträchtigt jedoch die laterale Auflösung, die auf etwa 7.4 µm reduziert wurde.

Die Streifenprojektion stößt bei tiefen Mulden und steilen Kanten an ihre Grenzen, da bestimmte Bereiche im Schatten liegen und nicht vollständig ausgeleuchtet werden. Dies führt zu Lücken in den resultierenden Tiefenkarten. In diesen Bereichen ist eine Evaluation, der mit den DFF-Verfahren berechneten Tiefenkarten nicht möglich.

Zusätzlich zu den Tiefenkarten wurden mit dem Streifenprojektionsmikroskop auch All-in-Focus-RGB-Aufnahmen der Proben erstellt. Diese dienen vor allem als Referenz zur Einordnung der gemessenen Bildbereiche.

5.3 Synthetisch-unscharfer Micro-Topo-Datensatz

Um eine ausreichende Generalisierbarkeit der DDFF-Modelle zu gewährleisten, ist ein umfangreicher Trainingsdatensatz erforderlich. Die Erstellung eines solchen Datensatzes mit dem experimentellen Mikroskop hätte jedoch den Einsatz einer großen Anzahl unterschiedlicher geologischer Proben erfordert, die nicht in ausreichendem Umfang verfügbar waren. Zudem wäre der zeitliche Aufwand zur vollständigen Vermessung aller Proben sehr hoch gewesen. Aus diesem Grund wurde ein bereits existierender, umfangreicher Datensatz für das Training herangezogen, der im Folgenden näher beschrieben wird. Da dieser Datensatz ausschließlich aus AiF-Bildern und den zugehörigen Tiefenkarten besteht, wurden daraus synthetische Fokalstapel generiert. Die dabei verwendete Methode zur Simulation unscharfer Aufnahmen wird anschließend erläutert.

5.3.1 Zugrundeliegender RGB-D-Datensatz

Für das Training der DDFF-Modelle wurde der Datensatz Micro-Topo (Siemens, Kästner und Reithmeier 2023) verwendet. Micro-Topo umfasst AiF-RGB-Bilder sowie korrespondierende Tiefenkarten von insgesamt 11757 Messungen. Die Daten basieren auf 167 mikrotopografischen Proben aus acht verschiedenen Materialien, die mittels 23 unterschiedlicher Fertigungsverfahren hergestellt wurden. Beispiele für die in dem Datensatz enthaltenen Proben sind in Abbildung 25 sowie in Tabelle 7 dargestellt.

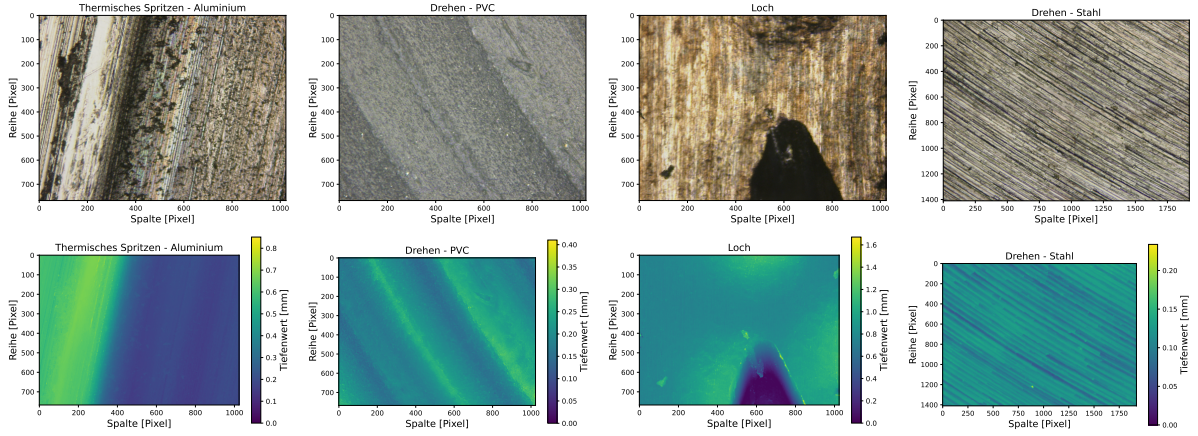


Abbildung 25: Beispiel-RGB-Bilder und -Tiefenkarten aus dem Micro-Topo-Datensatz

Die Erfassung erfolgte mithilfe eines konfokalen Laserscanning-Mikroskops. Bei diesem Verfahren tastet ein fokussierter Laserstrahl die Oberfläche der Probe punktweise ab. Ein Lochblendensystem sorgt dafür, dass Lichtanteile außerhalb der Fokusebene unterdrückt werden, wodurch ausschließlich Signale aus der exakt fokussierten Ebene detektiert werden. Durch das sequenzielle

Erfassen mehrerer Fokusebenen kann eine dreidimensionale Rekonstruktion der Probenoberfläche erzeugt werden.

Der Micro-Topo-Datensatz wurde in Trainings-, Validierungs- und Testdaten unterteilt, wobei die Aufteilung im Verhältnis 89%, 10% und 1% erfolgte. Die Trainingsdaten umfassen 10464 Messungen und dienen dem Training der DDFF-Modelle. Die Validierungsdaten bestehen aus 1175 Messungen und werden zur Feinabstimmung der Hyperparameter verwendet. Da die untersuchten DDFF-Modelle bereits vorab optimiert wurden, genügt in diesem Fall ein reduzierter Validierungsdatensatz. Für den Vergleich mit einer konventionellen DFF-Methode werden 118 Testmessungen herangezogen. Aufgrund des hohen zeitlichen Aufwands bei der Tiefenberechnung mittels der konventionellen Methode wurde hier ein kleiner Testdatensatz gewählt.

Tabelle 7: Beispiele für Proben des Micro-Topo-Datensatzes (Siemens, Kästner und Reithmeier 2023)

Bearbeitungsverfahren	**Material**	**Anzahl der Messungen**
Umfangsfräsen ($R_a = 0.7$µm)	Stahl	156
Umfangsfräsen ($R_a = 1.2$µm)	Stahl	129
Planfräsen ($R_a = 0.4$µm … $R_a = 12.5$µm)	vernickelter Stahl	226
Thermisches Spritzen	Al_2O_3	2053
Drehen	Aluminium	414
Drehen	PVC	547
Drehen	Stahl	412

5.3.2 Generierung synthetischer Fokalstapel

Methode. Zur Generierung synthetischer Fokalstapel wurde die von Gur und Wolf (2019) entwickelte PSF-Netzwerkschicht verwendet. Diese Schicht erhält als Eingabe ein AiF-Bild, eine zugehörige Tiefenkarte sowie die Kameraparameter Brennweite, Blendenöffnung und Fokusabstand. Auf Basis dieser Eingaben berechnet die PSF-Schicht ein unscharfes Bild für einen definierten Fokusabstand. Die Generierung der Fokalstapel erfolgt mittels des Programms `DefocusDataset.py`.

Die Unschärfe wird durch Anwendung eines Unschärfe-Kernels auf das Bild simuliert. Dieser Kernel ergibt sich aus der Überlagerung der Unschärfebeiträge benachbarter Pixel, die jeweils als Gauß-Kernel modelliert sind (vgl. Gleichung 3). Der Radius dieser Gauß-Funktion entspricht dem Radius des CoC in Pixeln.

Wie in Gleichung 18 dargestellt, hängt der Durchmesser des CoC neben der Objekttiefe auch von der Position des Aktuators im experimentellen Aufbau ab. Da in der Implementierung der PSF-Netzwerkschicht keine Anpassung des CoC vorgesehen ist, wird die resultierende Abweichung zunächst ignoriert. Es wird mit dem CoC der Nullposition des Aktuators gerechnet. Bei der Auswertung ist die potenzielle Fehlerhaftigkeit dieses Umstandes jedoch in Betracht zu ziehen.

Des Weiteren ermöglicht die Methode lediglich die Simulation der PSF als kreisförmige Funktion. Jedoch ergab die Analyse der optischen Eigenschaften des experimentellen 3D-Mikroskops, dass der

reale Defokus-Effekt unsymmetrisch ist (vgl. Kapitel 5.1.2). Dies stellt eine potenzielle zusätzliche Fehlerquelle dar.

Vorverarbeitung der Tiefenkarten. Die Messbereiche des Micro-Topo-Datensatzes variieren von wenigen Mikrometern bis hin zu mehreren Millimetern. Für das Training der DDFF-Modelle ist die exakte physikalische Größe der Proben in den Trainingsdaten jedoch zweitrangig. Um eine höhere Ähnlichkeit zu den realen Bedingungen des GeoFocus3D-Datensatzes zu erzielen, wurden die Tiefenkarten skaliert.

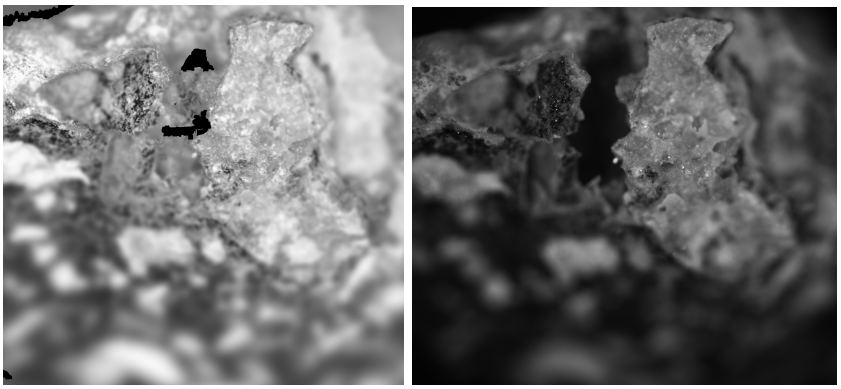
Dazu wurde zunächst der Tiefenbereich definiert, in dem der Großteil der Bildpunkte liegt. Dieser wurde durch Analyse des Tiefenwert-Histogramms ermittelt. Zunächst wurden das 1. und 99. Perzentil berechnet, um extreme Ausreißer zu eliminieren. Anschließend wurde der Interquartilsabstand (Interquartile Range, IQR) bestimmt, der die Streuung der mittleren 50% der Werte beschreibt. Der minimale Tiefenwert wurde ermittelt, indem vom 1. Perzentil sowohl der IQR als auch die Schrittweite subtrahiert wurden. Der maximale Tiefenwert ergibt sich analog durch Addition dieser Größen zum 99. Perzentil.

Die Tiefenkarten wurden anschließend skaliert, indem sie so oft mit dem Faktor 10 multipliziert wurden, bis der maximale Tiefenwert mindestens 1.5 mm erreichte. Zusätzlich wurde der minimale Tiefenwert subtrahiert, sodass der höchste Punkt bei 0 mm liegt. Dadurch entsprechen die Daten dem Messbereich des experimentellen Mikroskops von 0 bis 15 mm.

Reduktion der Rechenzeit. Die Berechnung synthetischer Fokalstapel aus dem umfangreichen Datensatz ist äußerst rechenintensiv. Um die Berechnungszeit zu verringern, wurden drei Optimierungsmaßnahmen umgesetzt: Begrenzung der Bildgröße, Reduzierung der Anzahl an Bildern pro Fokalstapel sowie Einschränkung der maximalen Kernelgröße.

Die Bilddaten des Micro-Topo-Datensatzes wiesen unterschiedliche Bildgrößen auf. Daher wurden die Bilder skaliert, sodass ihre Breite maximal 1100 Pixel beträgt. Zudem wurde der Fokalstapel-Messbereich auf 7 mm begrenzt, was den typischen Messbereichen der Proben im GeoFocus3D-Datensatz entspricht. Das heißt, auch bei größeren Objekten wurden nur so viele Aufnahmen generiert, bis Bildpunkte in einer Tiefe von 7 mm scharf abgebildet wurden. Während für den GeoFocus3D-Datensatz ein Fokusabstand mit einer Schrittweite von 0.1 mm gewählt wurde, musste für die synthetische Datengenerierung aufgrund der langen Rechenzeit eine Schrittweite von 0.5 mm verwendet werden.

Gur und Wolf (2019) definieren die Größe des Unschärfe-Kernels basierend auf dem maximalen CoC-Durchmesser, bei dem ein unscharfer Punkt noch als Punkt vom menschlichen Auge erkannt werden kann. Da im vorliegenden Anwendungsfall jedoch die mikroskopisch erfassbare Unschärfe zur Tiefenberechnung genutzt werden soll, ist diese visuelle Grenze ungeeignet. Um die maximal mögliche Genauigkeit zu erzielen, sollte stattdessen der CoC-Durchmesser für die maximal messbare Tiefe als Kernelgröße gewählt werden. Da jedoch die Rechenzeit mit zunehmender



(a) Synthetisch-unscharf (b) Real

Abbildung 26: Reales und synthetisch unscharfes Bild aus dem GeoFocus3D-Datensatz

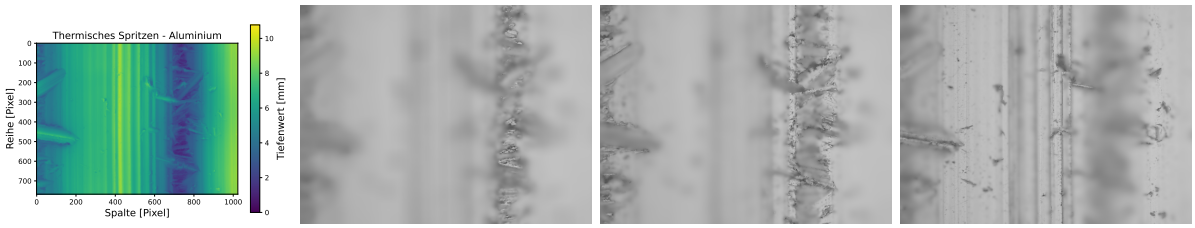


Abbildung 27: Tiefenkarte und dazugehörige RGB-Bilder aus dem synthetischen Fokalstapel des Micro-Topo-Datensatzes

Kernelgröße stark ansteigt, wurde der maximale Kernelradius auf 80 Pixel begrenzt. Dies entspricht dem CoC bei einer Tiefe von 7 mm.

Die Defokussierung der Bilder wurde auf einer *NVIDIA RTX 6000* Graphics Processing Unit (GPU) durchgeführt. Unter den beschriebenen Einschränkungen konnte die Rechenzeit pro Fokalstapel von ursprünglich etwa 20 Minuten auf rund 50 Sekunden reduziert werden. Insgesamt betrug die Dauer für die Erzeugung aller Fokalstapel des gesamten Datensatzes etwa sieben Tage.

Plausibilitätstest. Zur Validierung der synthetisch generierten, unscharfen Bilder wurde ein Plausibilitätstest durchgeführt. Hierzu wurde die PSF-Netzwerkschicht auf AiF-Bilder des GeoFocus3D-Datensatzes angewendet, um synthetisch defokussierte Bilder zu erzeugen. Diese wurden anschließend visuell mit den entsprechenden real aufgenommenen Bildern aus den Fokalstapeln verglichen.

Ein exemplarischer Vergleich ist in Abbildung 26 dargestellt. Die visuelle Ähnlichkeit der synthetischen und realen Unschärfen belegt die grundsätzliche Eignung der PSF-Schicht zur Generierung realistischer Defokus-Effekte. Die erkennbaren Lücken in dem synthetisch erzeugten Bild lassen sich auf fehlende Tiefeninformationen in der zugehörigen Tiefenkarte zurückführen. Beispielbilder aus einem generierten Fokalstapel aus dem Micro-Topo-Datensatz und die dazugehörige Tiefenkarte sind in der Abbildung 27 dargestellt.

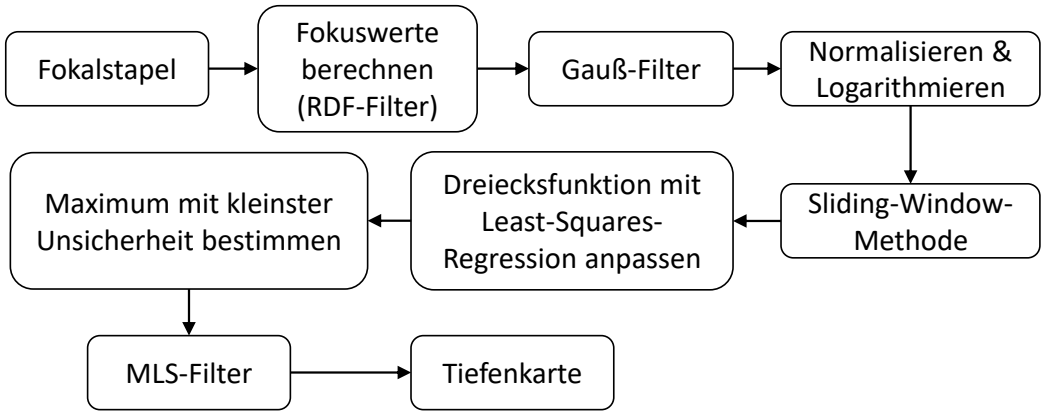


Abbildung 28: DFF-WLLSR-Algorithmus

5.4 Umsetzung des konventionellen DFF-Algorithmus

Zur Berechnung von Tiefenkarten mittels DFF, ohne auf Methoden des maschinellen Lernens zurückzugreifen, wurde der von Cou und Guennebaud (2024) entwickelte Algorithmus *Depth from Focus using Windowed Linear Least-Squares Regressions* (DFF-WLLSR) verwendet.

Zunächst werden für jede Aufnahme des Fokalstapels Fokuswerte berechnet. Anschließend wird innerhalb eines lokalen Fensters fester Größe für jedes Pixel das Maximum des Fokuswertverlaufs bestimmt. Dieses Maximum wird mithilfe einer linearisierten Least-Squares-Methode ermittelt. Der Punkt maximaler Schärfe wird aus den berechneten Regressionskoeffizienten abgeleitet.

Ein wesentlicher Vorteil dieser Methode liegt in ihrer Robustheit gegenüber Bildrauschen und geringer Texturierung. Durch eine gewichtete Analyse innerhalb lokaler Fenster wird der Einfluss von Ausreißern reduziert und die Stabilität der Schätzung verbessert. In experimentellen Vergleichen übertrifft der Algorithmus in Genauigkeit und Robustheit die klassischen DFF-Verfahren von Sakurikar und Narayanan (2017) und Billiot u. a. (2013), insbesondere in Bereichen mit schwacher Bildtextur.

Zur Implementierung wurde der von Cou und Guennebaud (2024) bereitgestellte MATLAB-Code in das Python-Programm `DFF-RDF.py` übertragen und an die spezifischen Datensätze angepasst. Eine Übersicht des Algorithmus ist in der Abbildung 28 dargestellt. Der Algorithmus lässt sich in drei Hauptschritte untergliedern: Fokuswertberechnung, Ermittlung des Maximums und Nachbearbeitung.

5.4.1 Berechnung der Fokuswerte

Die Berechnung basiert auf einem bereits ausgerichteten Fokalstapel mit konstanter Schrittgröße zwischen den Aufnahmen. Es wird angenommen, dass ein Objektpunkt in jeder Aufnahme auf den selben Pixel projiziert wird. Die Fokuswertberechnung erfolgt unter Verwendung des RDF nach Surh u. a. (2017), da dieser laut Cou und Guennebaud (2024) die besten Ergebnisse liefert.

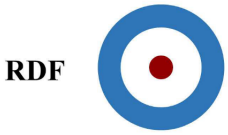
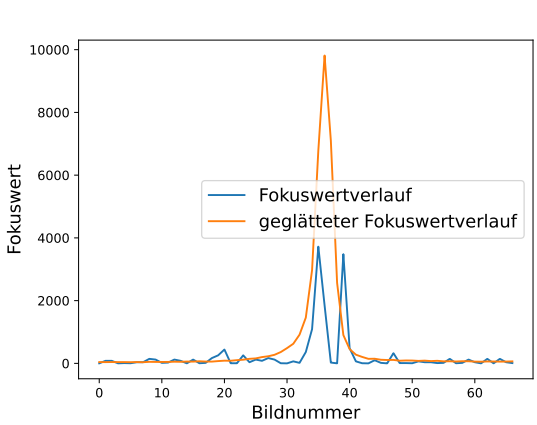
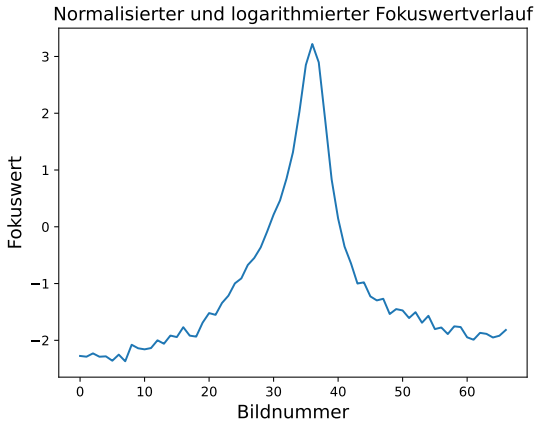


Abbildung 29: Struktur des RDF-Filters (Surh u. a. 2017)



(a) Originaler und gauß-geglätteter Fokuswertverlauf



(b) Normalisierter und logarithmierter Fokuswertverlauf

Abbildung 30: Fokuswertverläufe für einen Pixels bei der Schrittweite 0.1 mm

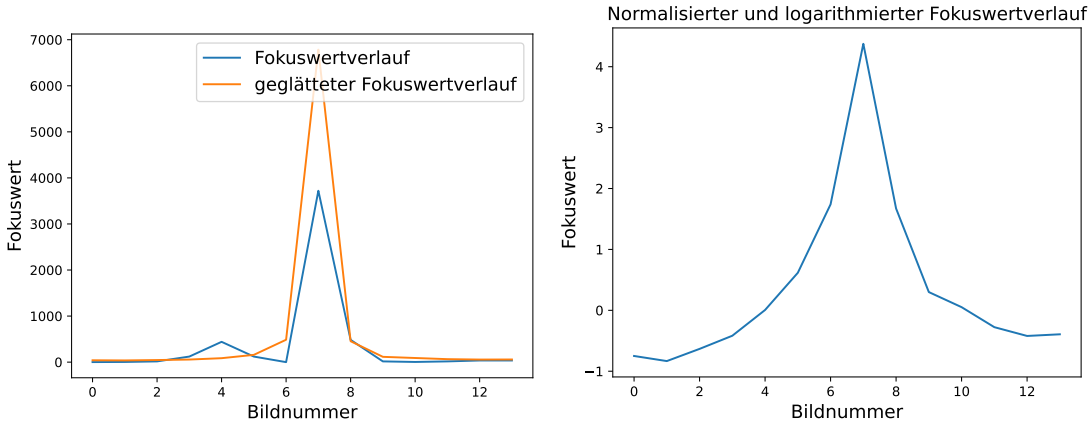
Die Struktur des RDF-Filters ist in Abbildung 29 visualisiert. Die Filterstruktur besteht aus einer inneren Scheibe (rot) und einem äußeren Ring (blau). Der Fokuswert ergibt sich aus der Differenz der Mittelwerte beider Bereiche. Die Zwischenbereiche werden nicht berücksichtigt. Die Fokuswertfunktion für ein Pixel an Position y_0 berechnet sich zu:

$$\text{RDF} = \begin{cases} -\frac{1}{\pi r_1^2} & |y_0 - y| \leq r_1 \\ \frac{1}{\pi(r_3^2 - r_2^2)} & r_2 < |y_0 - y| \leq r_3 \\ 0 & sonst \end{cases} \tag{24}$$

Dabei ist r_1 der Radius der Scheibe und r_2 und r_3 sind die Radii des Rings.

5.4.2 Ermittlung des Fokuswertmaximums

Zur Erhöhung der Robustheit gegenüber Rauschen wird zunächst ein 2D-Gaußfilter auf die Fokuswertbilder angewendet. Hierbei wurde eine Filtergröße von 20 Pixeln gewählt. Die Abbildungen 30a und 31a veranschaulichen den beispielhaften Verlauf der originalen und geglätteten Fokuswerte für einen einzelnen Pixel, dargelegt für Schrittweiten von 0.1 mm und 0.5 mm. Anschließend wird für jedes Pixel (i, j) die Tiefe z anhand des Fokuswertverlaufs f_k bestimmt. Dabei entspricht die Tiefe dem Maximum des idealisierten rekonstruierten Fokuswertverlaufs $\bar{f}$:



(a) Originaler und gauß-geglätteter Fokuswertverlauf

(b) Normalisierter und logarithmierter Fokuswertverlauf

Abbildung 31: Fokuswertverläufe für einen Pixels bei der Schrittweite 0.5 mm

$$d = \underset{y}{\operatorname{argmax}}\, \bar{f}(y) \tag{25}$$

Der rekonstruierte Fokuswertverlauf kann durch eine Laplace-Verteilung mit Erwartungswert μ und mittlerer absoluter Abweichung a angenähert werden:

$$L(\mu, a) = \frac{1}{2a} \exp\left(-\frac{|y-\mu|}{a}\right) \tag{26}$$

Zur robusten Schätzung wird eine Sliding-Window-Technik eingesetzt, wobei für jedes Fenster die Laplace-Verteilung mittels nichtlinearer Regression angepasst wird. Der zu minimierende Energie-Term lautet:

$$E = \sum_k \left(L_{\mu,a}(z_k) - f_k\right)^2 \tag{27}$$

Da die nichtlineare Regression rechenintensiv ist, wird E durch Logarithmierung linearisiert und der Laplace-Term $\frac{1}{2a}$ durch einen freien Parameter a ersetzt. Damit ergibt sich die logarithmierte Laplace-Verteilung zu:

$$\log L(\mu, a) = \log a - \frac{|y-\mu|}{y} \tag{28}$$

Gleichung 28 kann umformuliert werden zu einer stückweisen linearen Dreiecksfunktion:

$$T = \begin{cases} c \cdot y + p, & \text{für } y < \mu \\ -c \cdot y + q, & \text{für } y \geq \mu \end{cases} \quad \text{mit} \quad \mu = \frac{q-p}{2c}, \quad a = \frac{q+p}{2} \tag{29}$$

Die Fokuswertverläufe werden in zwei Teilintervalle W_1 und W_2 um die Mitte des lokalen Fensters mit Fenstergröße n_w geteilt:

$$W_1 = \left[k,\, k + \frac{n_w}{2} - 1\right], \quad W_2 = \left[k + \frac{n_w}{2},\, k + n_w - 1\right] \tag{30}$$

Dabei ist k die Nummer des Bildes im Fokalstapels. Durch Least-Squares-Optimierung wird das Minimum des folgenden Terms bestimmt:

$$E = \sum_{k \in W_1} (cz_k + p - \log(f_k))^2 + \sum_{k \in W_2} (-cz_k + p - \log(f_k))^2$$

Die Fenstergröße wurde anhand der logarithmieren Fokuswertverläufe (Abbildung 30b und 31b) festgelegt. Für eine Schrittweite von 0.5 mm wurde eine Fenstergröße von 7 Bildern gewählt, für 0.1 mm eine von 11 Bilder.

5.4.3 Nachbearbeitung der Tiefenkarte

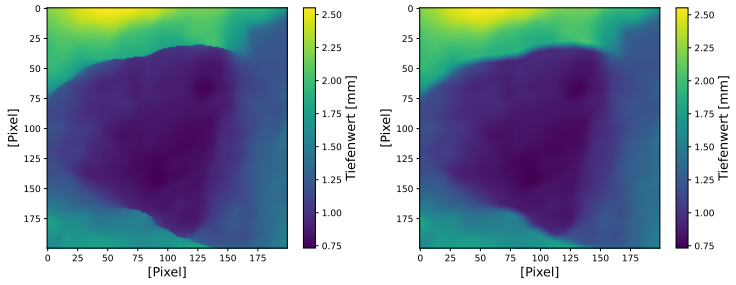
Als Resultat liefert der beschriebene Algorithmus eine Tiefenkarte D sowie eine zugehörige Unsicherheitskarte U. Zur Reduktion von Rauschen und zur Korrektur unsicherer Bereiche wird ein Moving-Least-Squares (MLS)-Filter verwendet. Für jeden Pixel (i, j) wird ein neuer Wert auf Basis der Nachbarschaft berechnet. Die Umgebung eines Pixels ρ_{ij} wird durch ein bivariates Polynom ϵ beschrieben, das durch Minimierung des folgenden Terms bestimmt wird:

$$\sum_{x,y \in N_{ij}} U_{x,y} \Theta(\|\rho - \rho_{x,y}\|) \cdot (\epsilon(\rho_{x,y}) - D_{x,y})^2 \tag{31}$$

Dabei ist Θ eine Gewichtsfunktion mit dem Filterradius h, gewählt als:

$$\Theta(t) = \left(1 - \frac{t^2}{h^2}\right)^2 \tag{32}$$

Der verwendete Filterradius für die Nachbearbeitung beträgt 5 Pixel. In Abbildung 32 wird ein Ausschnitt einer berechneten Tiefenkarte im Vergleich mit und ohne Anwendung des MLS-Filters dargestellt. Dabei lässt sich feststellen, dass vor allem scharfe Kanten durch den Einsatz des MLS-Filters geglättet werden.



(a) ohne MLS-Filter (b) mit MLS-Filter

Abbildung 32: Mit dem DFF-WLLSR-Verfahren berechnete Tiefenkarten mit und ohne MLS-Filter

5.5 Implementierung der DDFF-Modelle

In den nachfolgenden Abschnitten wird die Auswahl der implementierten DDFF-Modelle begründet. Anschließend erfolgt eine detaillierte Beschreibung und Erläuterung der Implementierung der ausgewählten Modelle. Für beide Modelle standen vortrainierte Versionen, bereitgestellt von den ursprünglichen Autoren, zur Verfügung, die auf die Testdatensätze angewendet wurden. Des Weiteren erfolgte ein erneutes Training der Modelle mithilfe des erstellten Trainingsdatensatzes.

5.5.1 Auswahl der Modelle

Die klassische DFF-Methode ohne Einsatz maschinellen Lernens wurde mit zwei DDFF-Methoden verglichen, die auf CNNs basieren. Die Auswahl der beiden DDFF-Modelle erfolgte auf Grundlage einer Bewertung aktueller Verfahren aus dem Stand der Technik anhand definierter Kriterien. Diese umfassten: die Performanz auf öffentlich verfügbaren Datensätzen, die Flexibilität hinsichtlich der Anzahl von Bildern im Fokalstapel sowie die Verfügbarkeit des Quellcodes.

Ausgewählt wurden Modelle, die in Vergleichsstudien eine möglichst geringe Fehlerquote bei der Tiefenschätzung aufwiesen. Die Unterstützung variabler Fokalstapellängen stellt einen Vorteil in Bezug auf die Anwendbarkeit in unterschiedlichen Szenarien dar. Um den Aufwand bei der Reimplementierung zu minimieren und mögliche Implementierungsfehler zu vermeiden, wurde zudem auf die Verfügbarkeit veröffentlichter Programmcodes geachtet.

Die betrachteten Methoden wurden in Tabelle 8 anhand den genannten Auswahlkriterien gegenübergestellt. Für die Bewertung der Performanz wurden fünf Datensätze identifiziert, auf denen mindestens drei der betrachteten Modelle evaluiert wurden. Basierend auf dem Fehlerwert MSE wurde für jeden Datensatz ein Ranking erstellt. Dabei steht eine Platzierung mit Rang 1 für das Modell mit dem geringsten Fehler auf dem jeweiligen Datensatz. Da nicht alle Modelle für alle Datensätze evaluiert wurden, sind fehlende Einträge in der Tabelle mit einem „–“ gekennzeichnet.

Da die Trainingsdaten und Vorverarbeitungsschritte zwischen verschiedenen Studien teils erheblich variieren, wurden die MSE-Werte nicht direkt miteinander verglichen. Stattdessen erfolgte die Einordnung relativ zueinander auf Grundlage der jeweils berichteten Ergebnisse. Es kann daher vorkommen, dass mehrere Methoden auf demselben Datensatz die gleiche Platzierung erhalten, wenn sie nicht direkt miteinander verglichen wurden.

Von den betrachteten Methoden ist lediglich bei DDFFNet (Hazirbas u. a. 2019) und DFVNet (Yang, Huang und Z. Zhou 2022) die Anzahl der Bilder im Fokalstapel nicht variabel. Der Quellcode liegt für alle Modelle mit Ausnahme des neuesten Ansatzes DDFSNet (Fujimura u. a. 2024) vollständig vor. Für DDFSNet ist jedoch dokumentiert, auf welchem öffentlich verfügbaren Code die eigene Implementierung basiert und welche Anpassungen vorgenommen wurden. Die Netzwerkarchitektur ist zudem in tabellarischer Form beschrieben.

Im Performanzvergleich zeigt sich, dass DDFFNet (Hazirbas u. a. 2019) und DefocusNet (Maximov, Galim und Leal-Taixe 2020) von nachfolgenden Modellen in sämtlichen Datensätzen übertroffen

Tabelle 8: Vergleich der DDFF-Modelle anhand relevanter Kriterien

Kriterium	DDFFNet (2019)	DefocusNet (2020)	MultiscaleDDFFNet (2021)	AiFDepthNet (2021)	DDFFintheWild (2022)	DFVNet (2022)	DDFSNet (2024)
Ranking auf DDFF 12-Scene	4	3	-	2	1	1	-
Ranking auf NYU-Depth-V2	3	4	-	5	-	2	1
Ranking auf FoD500	4	3	-	2	1	1	4
Ranking auf Smartphone	-	3	-	2	1	-	-
Ranking auf Middlebury	2	2	1	1	-	-	-
Fokalstapel variabler Größe möglich	nein	ja	ja	ja	ja	nein	ja
Programm verfügbar	ja	ja	ja	ja	ja	ja	teilweise

wurden. Für MultiscaleDDFF (Ceruso u. a. 2021) liegen keine direkten Vergleiche mit neueren Modellen vor, sodass keine abschließende Aussage zur Genauigkeit möglich ist. AiFDepthNet (Wang u. a. 2021) erzielte lediglich auf einem von fünf Datensätzen den besten Rang. Im Gegensatz dazu erreichte DDFFintheWild (Won und Jeon 2022) bei allen drei untersuchten Datensätzen den jeweils geringsten Tiefenfehler. DFVNet (Yang, Huang und Z. Zhou 2022) lieferte bei zwei von drei Datensätzen die genaueste Tiefenschätzung; im dritten Fall belegte es den zweiten Platz hinter DDFSNet (Fujimura u. a. 2024). DDFSNet erzielte auf NYU-Depth-V2 (Silberman u. a. 2012) die höchste Genauigkeit, belegte jedoch auf dem Datensatz FoD500 (Maximov, Galim und Leal-Taixe 2020) nur Platz vier.

Basierend auf den genannten Auswahlkriterien wurden die Modelle DDFFintheWild (Won und Jeon 2022) und DFVNet (Yang, Huang und Z. Zhou 2022) für den Vergleich ausgewählt. Beide verfügen über öffentlich zugänglichen Quellcode und konnten in der Mehrzahl der getesteten Datensätze den geringsten Fehler bei der Tiefenschätzung erzielen. Auch wenn DFVNet nicht auf Fokalstapel variabler Länge anwendbar ist, erscheint eine Evaluation des Modells aufgrund seiner hohen Performanz dennoch als sinnvoll.

5.5.2 Implementierung von DFVNet

Modell. Abbildung 33 zeigt den schematischen Aufbau des DFVNet-Modells (Yang, Huang und Z. Zhou 2022). Die Eingabe des Netzwerks besteht aus einem Fokalstapel sowie den Fokusabständen der einzelnen Bilder des Stapels. Es wird angenommen, dass der Fokalstapel vorab ausgerichtet wurde.

Zunächst werden mit einem 2D-CNN Merkmale aus jedem Bild des Fokalstapels extrahiert. Diese

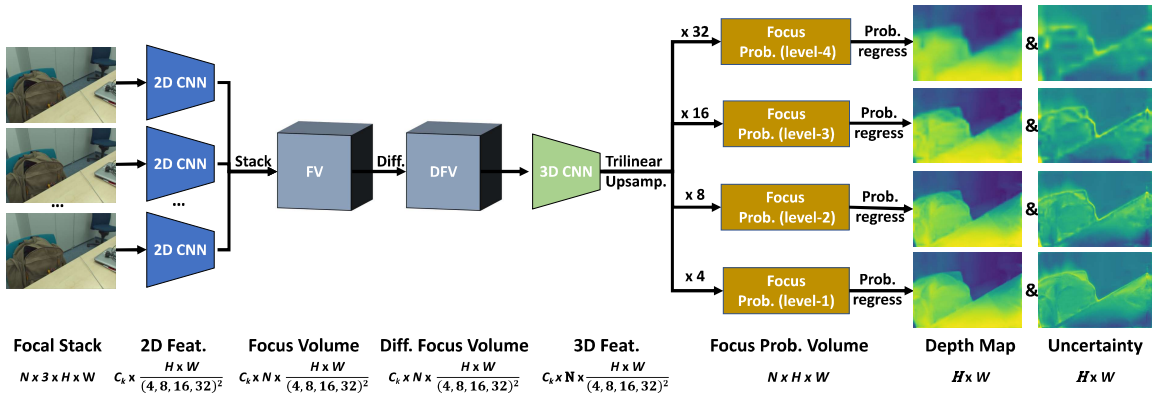


Abbildung 33: Aufbau von DVFNet (Yang, Huang und Z. Zhou 2022). Dabei ist N die Anzahl der Bilder im Fokalstapel, H und W die Größe des Bilder und C die Farbkanäle.

Merkmalsdarstellungen dienen als Grundlage für die Berechnung des Differential Focus Volume (DFV). Dabei werden die Merkmale benachbarter Bilder voneinander subtrahiert, um explizit Unterschiede in der Schärfe hervorzuheben. Der zugrunde liegende Gedanke ist, dass fokussierte Bildbereiche deutlichere Differenzen aufweisen, während unscharfe Regionen relativ konstant bleiben. Diese differenzielle Betrachtung verstärkt die Fokusinformation und erleichtert so die anschließende Tiefenschätzung.

Das resultierende DFV wird zusammen mit den ursprünglichen Bildmerkmalen einem dreidimensionalen CNN zugeführt. Dieses berechnet für jeden Pixel eine Wahrscheinlichkeitsverteilung darüber, in welchem Bild des Fokalstapels dieser Pixel am stärksten fokussiert ist. Das dabei entstehende Wahrscheinlichkeitsvolumen P ist so normiert, dass die Summe der Wahrscheinlichkeiten für jeden Pixel p_j gleich eins ist. Die wahrscheinlichste Fokusebene wird über folgende Gleichung berechnet:

$$\hat{k}_j = \sum_{k=1}^{K} i \cdot P_{k,j} \tag{33}$$

Dabei ist K die Anzahl der Bilder im Fokalstapel und $P_{i,j}$ die Wahrscheinlichkeit, dass Pixel j im Bild k am stärksten fokussiert ist. Der berechnete Index $\hat{i}_j$ kann dabei eine reelle Zahl zwischen zwei Bildern annehmen.

Die finale Tiefe d wird über eine Wahrscheinlichkeitsregression berechnet, wobei g_f der Fokusabständ ist:

$$d_j = \sum_{k=1}^{K} g_{f,k} \cdot P_{k,j} \tag{34}$$

Implementierung. Der Netzwerkaufbau von Yang, Huang und Z. Zhou (2022) und ist im Anhang detailliert dargestellt. Für das 2D-CNN wurde ein vortrainiertes ResNet18-FPN (Lin u. a. 2017) verwendet, das mit dem ImageNet-Datensatz (Deng u. a. 2009) vortrainiert wurde. Die DFV- und 3D-CNN-Komponenten basieren auf 3D-ResNet-Blöcken gemäß Hara, Kataoka und Satoh (2017). Die Implementierung erfolgte unter Verwendung von PyTorch.

Als Verlustfunktion wurde der multiskalare Smooth-L1-Loss zwischen der berechneten Tiefenkarte D und der Ground-Truth-Tiefenkarte D^* eingesetzt:

$$\text{Loss} = \sum_{i=1}^{4} w_i \left(\frac{1}{N_s} \sum_{j=1}^{N_s} L_{1,smooth} \left(D_j^{(i)} - D_j^{*(i)} \right) \right) \tag{35}$$

Hierbei ist N_s die Anzahl der gültigen Pixel auf Skala i und w_i sind die dazugehörigen Gewichte. Die Smooth-L1-Loss-Funktion wird gemäß Gleichung 13 berechnet.

Yang, Huang und Z. Zhou (2022) veröffentlichten ein Modell, das auf den Datensätzen FoD500 (Maximov, Galim und Leal-Taixe 2020) und DDFF-12-Scene (Hazirbas u. a. 2019) vortrainiert wurde. Das vortrainierte DFVNet wurde mit den eigenen Testdatensätzen evaluiert.

Das Training des vortrainierten Modells erfolgte mit Fokalstapeln, die jeweils fünf Bilder umfassen. Für die Auswertung der Testdatensätze wurden daher fünf Bilder pro Stapel ausgewählt, die gleichmäßig über den jeweiligen Messbereich verteilt sind. Da die Merkmalsextraktion auf einem ResNet-Modell basiert, das mit Bildausschnitten von 224 × 224 Pixeln trainiert wurde, ist auch die Eingabebildgröße festgelegt. Die Bilder aus den Testdatensätzen wurden daher so skaliert, dass ihre maximale Höhe 244 Pixel beträgt. Zur anschließenden Validierung wurden die berechneten Tiefenkarten wieder auf die ursprüngliche Bildgröße hochskaliert. Dadurch weisen die mit dem vortrainierten DFVNet erzeugten Tiefenkarten jedoch eine verringerte Detailgenauigkeit auf.

DFVNet wurde zusätzlich mit dem Micro-Topo-Datensatz neu trainiert. Yang, Huang und Z. Zhou (2022) stellten fest, dass eine Erhöhung der Anzahl an Bildern im Fokalstapel über fünf hinaus zu keiner signifikanten Verbesserung der Tiefengenauigkeit führt. Daher wurden auch in dieser Arbeit Fokalstapel mit jeweils fünf Aufnahmen für das Training verwendet. Die Datenaugmentation entsprach der in der Originalpublikation beschriebenen Vorgehensweise: Die Bilder der Fokalstapel sowie die zugehörigen Ground-Truth-Tiefenkarten wurden zufällig auf eine Größe von 224 × 224 Pixeln zugeschnitten und darüber hinaus zufällig horizontal oder vertikal gespiegelt.

Das Training erfolgte auf einer *NVIDIA RTX 6000* GPU. In Übereinstimmung mit der Arbeit von Yang, Huang und Z. Zhou (2022) wurde eine Batchgröße von 20 gewählt. Während in der Originalstudie 700 Epochen trainiert wurde, wurde das Training in dieser Arbeit nach 528 Epochen beendet, da sich keine Verbesserung des Validierungsfehlers gezeigt hat. Die Optimierung der Modellgewichte erfolgte mithilfe des Adam-Optimierers bei einer Lernrate von 0.0001. Der Trainingsprozess erstreckte sich über einen Zeitraum von ungefähr sechs Tagen.

Abbildung 34 illustriert den Verlauf des Trainings- und Validierungsfehlers über die Epochen. Der

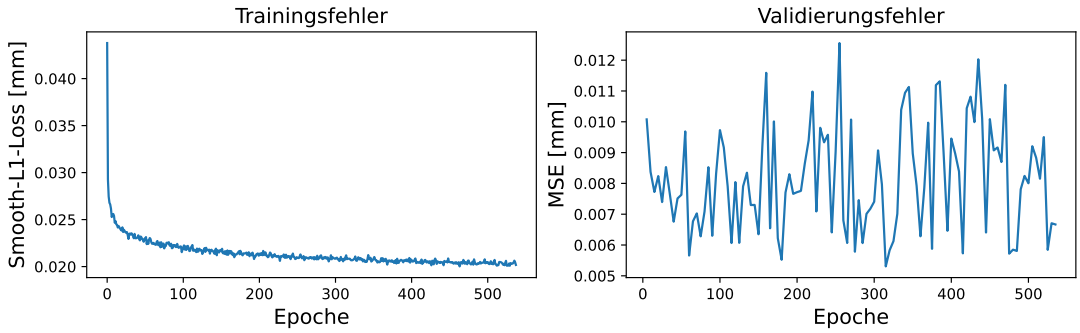


Abbildung 34: Trainings- und Validierungsfehler während des Trainings von DFVNet

Validierungsfehler zeigt bereits von Beginn an einen niedrigen Wert und erfährt keine signifikante Verbesserung, was auf Overfitting hindeutet.

5.5.3 Implementierung von DDFFintheWild

Modell Die von Won und Jeon (2022) vorgestellte DDFF-Methode verwendet als Eingabe einen Fokalstapel, die zugehörigen Fokusdistanzen sowie die Brennweite des optischen Systems. Die Berechnung der Tiefenkarte erfolgt in drei Hauptschritten: Zunächst richtet ein sogenanntes Alignment-Netzwerk die Bilder des Fokalstapels zueinander aus. Anschließend extrahiert ein Sharp Region Detection (SRD)-Modul fokussierte Bildmerkmale. Abschließend werden diese Merkmale durch das Effective Downsampling (EFD)-Modul auf die wesentlichen Informationen reduziert. Eine Übersicht über den Aufbau des DDFF-Netzwerks ist in der Abbildung 35 dargestellt.

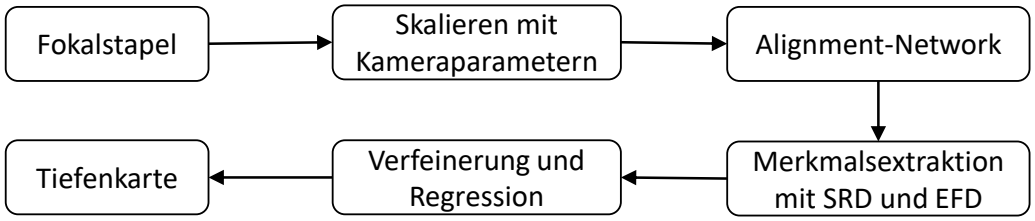


Abbildung 35: Aufbau von DDFFintheWild in Anlehnung an Won und Jeon (2022)

Das Alignment-Netzwerk kompensiert Veränderungen der Bildvergrößerung, die durch den sogenannten Focal-Breathing-Effekt entstehen, sowie geringfügige Verschiebungen des Bildhauptpunkts. Der Focal-Breathing-Effekt ist ein in herkömmlichen Kameras häufig auftretendes Phänomen, bei dem sich mit der Änderung des Fokusabstands ein Zoom-Effekt bemerkbar macht. Dieser Effekt resultiert aus der Bewegung der Linse, wodurch sich deren Abstand zum Sensor verändert. Das hat wiederum eine Änderung der Bildvergrößerung und des FOVs zur Folge (Herrmann u. a. 2020). Aufgrund der gewählten Anordnung der beiden Objektive im experimentellen Versuchsaufbau bleibt die Vergrößerung beim Variieren des Fokusabstands jedoch konstant, sodass

der Focal-Breathing-Effekt in diesem Fall nicht auftritt. Die geometrische Kalibrierung hat jedoch eine geringfügige Verschiebung des Bildhauptpunkts nachgewiesen (siehe Kapitel 5.1.3).

Für die Merkmalsextraktion im SRD-Modul werden sowohl 2D- als auch 3D-Faltungen verwendet, die auf ResNet-Blöcken basieren. Die 3D-Faltungen ermöglichen dabei den Informationsaustausch zwischen den einzelnen Bildern des Fokalstapels.

Das EFD-Modul verwendet eine 2D-Max-Pooling-Operation zur Reduktion der räumlichen Auflösung und führt anschließend eine 3D-Faltung auf das resultierende Merkmalsbild aus. Auf diese Weise ermöglicht das EFD-Modul dem Netzwerk, sowohl repräsentative Merkmale fokussierter Bereiche innerhalb eines lokalen Fensters zu extrahieren als auch den Informationsaustausch zwischen benachbarten fokalen Ebenen zu fördern.

Die resultierenden Merkmale, welche die Bildschärfe codieren, werden anschließend zu einem sogenannten Fokusvolumen aggregiert. Aus diesem Volumen wird für jeden Pixel die Wahrscheinlichkeit bestimmt, dass dieser die höchste Fokussierung innerhalb des Stapels aufweist. Die endgültige Tiefenschätzung erfolgt schließlich mittels Wahrscheinlichkeitsregression.

Implementierung. Die detaillierte Architektur des Netzwerks ist im Anhang dargestellt. Die Implementierung erfolgte mithilfe von PyTorch. Zur Optimierung während des Trainings wird ein multiskalarer, gewichteter L2-Loss verwendet, welcher den Vergleich zwischen der vorhergesagten Tiefenkarte D und der Ground-Truth-Tiefenkarte D^* erlaubt:

$$\text{Loss} = \sum_{i=1}^{4} w_i \cdot \|D_i - D_i^*\|^2 \tag{36}$$

Hierbei stehen i für die vier verschiedenen Skalen und w_i für die jeweils zugehörigen Gewichte.

Won und Jeon (2022) stellen ein vortrainiertes Modell zur Verfügung, das auf dem synthetisch-unscharfen Datensatz NYU-Depth-V2 (Silberman u. a. 2012) basiert. Die Fokalstapel dieses Datensatzes wurden mithilfe eines Simulators erzeugt, der sowohl Focal-Breathing-Effekte als auch Verschiebungen des Bildhauptpunkts berücksichtigt. Jeder Stapel umfasst genau zehn Bilder, was zur Folge hat, dass das Modell nicht auf Fokalstapel anderer Längen anwendbar ist. Daher werden aus den Testdatensätzen jeweils zehn gleichmäßig über den Messbereich verteilte Bilder ausgewählt. Im Gegensatz zu DFVNet ist DDFFintheWild flexibel gegenüber der Bildgröße, so dass diese nicht angepasst werden muss. Das vortrainierte Modell führt eine Korrektur des Focal-Breathing-Effekts durch, obwohl dieser in den Testdatensätzen nicht auftritt. Es bleibt zu untersuchen, inwieweit sich dies auf die resultierenden Tiefenkarten auswirkt.

Zusätzlich wurde das Modell DDFFintheWild mit dem Micro-Topo-Datensatz neu trainiert. Dabei kam ausschließlich der Netzwerkteil zur Tiefenbestimmung zum Einsatz, bestehend aus der Merkmalsextraktion, der Verfeinerung sowie der Regressionskomponente. Analog zum Vorgehen bei DFVNet wurden Fokalstapel mit jeweils fünf Aufnahmen verwendet.

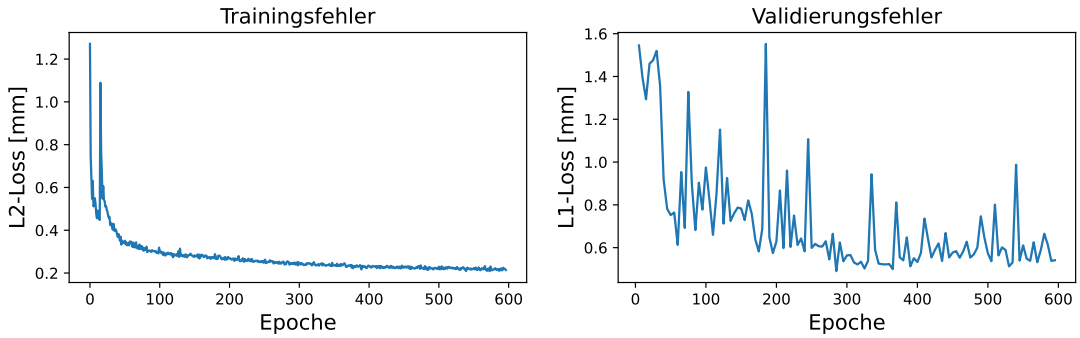


Abbildung 36: Trainings- und Validierungsfehler während des Trainings von DDFFin-theWild

Zur Beschleunigung des Trainingsprozesses wurden die unscharfen Bilder sowie die zugehörigen Ground-Truth-Tiefenkarten zufällig auf eine Auflösung von 256×256 Pixel zugeschnitten. Darüber hinaus kamen verschiedene Datenaugmentationen zum Einsatz, darunter zufällige horizontale und vertikale Spiegelungen sowie eine zufällige Bildkorrektur. Letztere umfasste Änderungen des Kontrasts und der Helligkeit sowie eine Gammakorrektur. Das augmentierte Bild I'_{aug} wurde dabei aus dem auf den Wertebereich [0,1] normierten unscharfen Bild I' wie folgt berechnet

$$I'_{aug} = (0.5 + \texttt{Kontrast} \cdot (I' - 0.5) + \texttt{Helligkeit})^{\texttt{Gamma}} \tag{37}$$

$$\text{mit } \texttt{Kontrast} \in [0.4, 1.6],\ \texttt{Helligkeit} \in [-0.1, 0.1],\ \texttt{Gamma} \in [0.5, 2.0]$$

Das Training erfolgte erneut auf einer *NVIDIA RTX 6000* GPU. Es wurde wieder eine Batchgröße von 20 gewählt. Es wurde für 600 Epochen trainiert bis keine weitere Verbesserung des Validierungsfehlers beobachtet werden konnte. Die Optimierung der Modellgewichte erfolgte mithilfe des Adam-Optimierers bei einer Lernrate von 0.0001. Der Trainingsprozess erstreckte sich über einen Zeitraum von ungefähr sieben Tagen.

In Abbildung 36 werden die Trainings- und Validierungsfehler im Verlauf der Epochen dargestellt. Der Validierungsfehler zeigt hier ebenfalls eine Stagnation, was darauf hinweist, dass das Modell möglicherweise überangepasst ist.

6 Validierung

Da die Fokalstapel und die Ground-Truth-Tiefenkarten des GeoFocus3D-Datensatzes mit unterschiedlichen Instrumenten aufgenommen wurden, ist zunächst eine Ausrichtung der berechneten Tiefenkarten erforderlich. Das dabei angewandte Vorgehen wird im Folgenden beschrieben. Anschließend werden die zur Bewertung der Tiefenkarten verwendeten Fehlerkennwerte vorgestellt. Daraufhin erfolgt die Präsentation der Ergebnisse, die sich aus der Anwendung der verschiedenen DFF-Methoden auf den Micro-Topo-Testdatensatz sowie auf den GeoFocus3D-Datensatz ergeben haben. Den Abschluss bildet eine kritische Diskussion der Resultate.

6.1 Ausrichten der Tiefenkarten

Für den GeoFocus3D-Datensatz ist eine Ausrichtung der berechneten Tiefenkarten an den Ground-Truth-Tiefenkarten notwendig. Da für Fokalstapel und Ground-Truth-Tiefenkarten verschiedene Messsysteme verwendet wurden, ist ein direkter Vergleich nicht möglich. Zudem ist das zusammengesetzte Sichtfeld der Ground-Truth-Tiefenkarten größer als das der aus dem Fokalstapel berechneten Tiefenkarten.

Zunächst wird manuell in den Ground-Truth-Tiefenkarten der Bildausschnitt identifiziert, der dem im Fokalstapel abgebildeten Bereich entspricht. Die Ground-Truth-Tiefenkarten werden entsprechend zugeschnitten. Diese manuelle Auswahl ermöglicht jedoch nur eine grobe Übereinstimmung. Zur weiteren Verbesserung wird eine geometrische Transformation zwischen den 3D-Koordinaten der berechneten Tiefenkarte D und der zugeschnittenen Ground-Truth-Tiefenkarte D^* durchgeführt. Diese Transformation berücksichtigt eine 3D-Translation $\mathbf{T}$, eine Rotation $\mathbf{R}$ um die z-Achse sowie eine 2D-Skalierung $\mathbf{S}$ und lässt sich zusammenfassen als:

$$D^* = \mathbf{S} \cdot \mathbf{R} \cdot D + \mathbf{T} \tag{38}$$

Während Ranftl u. a. (2020) die Transformation basierend auf allen 3D-Punkten mittels Least-Squares-Optimierung bestimmten, wird hier vorgeschlagen, lediglich korrespondierende Bildmerkmale (Keypoints) zu verwenden. Dies erhöht die Robustheit gegenüber lokalen Tiefenfehlern, beispielsweise durch unzureichende Beleuchtung, da solche Punkte nicht zuverlässig korrespondierende Keypoints aufweisen.

Zur Berechnung der Transformationen wurde das Programm `Depth-Map-Transformation.py` verwendet, dessen Ablauf in der Abbildung 37 visualisiert ist. Die Keypoints wurden mit dem Algorithmus Scale Invariant Feature Transform (SIFT) nach Lowe (2004) detektiert. Dieser ist robust gegenüber Skalierung, Rotation und Helligkeitsänderungen. Die Zuordnung der Keypoints zwischen den beiden Tiefenkarten erfolgte über einen Brute-Force-Matcher (OpenCV 2025), der korrespondierende Punkte auf Basis des geringsten euklidischen Abstands bestimmt. In der Abbildung 38a sind die detektierten Keypoint-Paare anhand der Tiefenkarten einer Beispielprobe dargestellt. Paare mit einem Abstand von mehr als 100 Pixeln wurden als fehlerhaft identifiziert und verworfen. Die verbleibenden Keypoint-Paare sind in Abbildung 38b veranschaulicht.

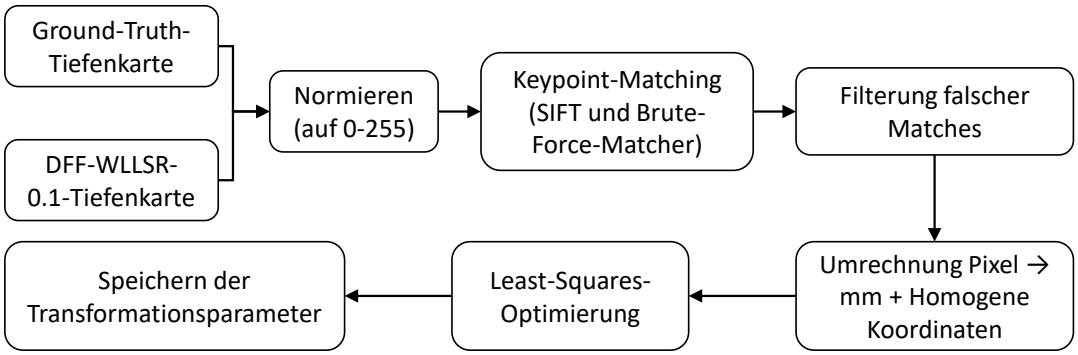


Abbildung 37: Arbeitsablauf bei der Berechnung der Transformation zwischen geschätzten Tiefenkarten und Ground-Truth-Tiefenkarten

Basierend auf den n korrespondierenden Keypoints κ_i und κ_i^* wird die Transformation durch Lösung eines Least-Squares-Problems berechnet:

$$\min_X \sum_{i=1}^{l} \|M_T(\kappa_i, X) - \kappa_i^*\|^2 \tag{39}$$

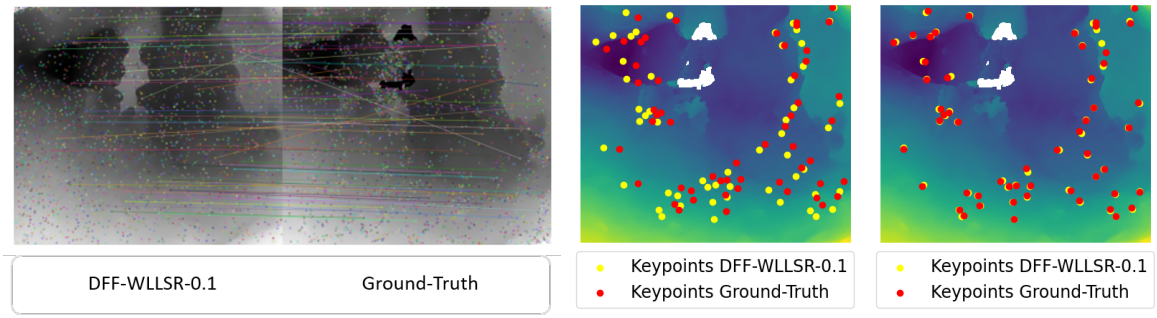
Dabei bezeichnet $M_T(\kappa_i, X)$ die Transformation der Koordinaten k_i durch den Parametervektor X, bestehend aus Translations-, Rotations- und Skalierungsparametern.

Zur Lösung des Optimierungsproblems wurde die Trust-Region-Reflective-Methode aus der SciPy-Bibliothek (The-SciPy-Community 2025) verwendet. Die Startwerte wurden auf Null gesetzt, und die Jacobi-Matrix numerisch berechnet. In der Abbildung 38c sind die transformierten Keypoints anhand der Beispielprobe dargestellt. Der verbleibende mittlere absolute Transformationsfehler (TE) wurde definiert als:

$$TE = \frac{1}{n} \sum_{i=1}^{l} \|M_T(\kappa_i, X) - \kappa_i^*\| \tag{40}$$

Die Transformationen werden auf Basis der Tiefenkarten berechnet, die mithilfe des DFF-Windowed Linear Least Squares Regression (WLLSR)-Verfahrens mit einer Schrittweite von 0.1 mm generiert wurden. Aufgrund des hohen Detailgrads dieser Tiefenkarten können möglichst viele Keypoints zuverlässig detektiert werden. Die daraus bestimmten Transformationen werden anschließend auf die Tiefenkarten aller untersuchten Methoden übertragen.

Die Tabelle 9 zeigt für jede Probe die Anzahl der gültig detektierten Keypoint-Paare, den resultierenden Transformationsfehler sowie die ermittelte Skalierung. Bei Probe 4 konnte lediglich ein einziges Keypoint-Paar identifiziert werden, wodurch eine Berechnung der Transformation nicht möglich war. Für Probe 6 wurden drei Keypoint-Paare gefunden, was grundsätzlich eine Transformation erlaubt, jedoch mit einer erhöhten Unsicherheit behaftet ist. Die Proben 4 und 6



(a) (b) (c)

Abbildung 38: Transformation der in den Tiefenkarten gefundenen Keypoints mit der Darstellung der korrespondierenden Keypoints (a), den gültigen Keypoints auf der Ground-Truth-Tiefenkarte (b) und den transformierten Keypoints (c)

Tabelle 9: Anzahl der Keypoint-Paare und Transformationsfehler je Probe bei der Ausrichtung der Tiefenkarten

Probe	Anzahl Keypoint-Paare	Transformationsfehler [µm]	Skalierung
1	59	27.60	1.06
2	35	39.76	1.05
3	67	16.77	1.06
4	1	-	-
5	35	18.88	1.06
6	3	7.71	1.04
7	54	25.13	1.06
8	62	25.27	1.06
9	98	16.34	1.06
10	23	37.92	1.05
11	97	34.51	1.05
12	63	22.24	1.05
13	82	22.21	1.05

werden daher von der quantitativen Auswertung ausgeschlossen.

Der mittlere Transformationsfehler beträgt 24.53 µm, was einer Abweichung von etwa drei Pixeln entspricht. Eine Erweiterung des Modells um zusätzliche Rotationsparameter führte zu keiner signifikanten Reduktion dieses Fehlers. Mögliche Ursachen für den verbleibenden Fehler sind nicht modellierte optische Verzeichnungen oder Ungenauigkeiten bei der Detektion und Zuordnung von Keypoints. Diese Faktoren müssen in zukünftigen Arbeiten systematisch untersucht werden. Der verbleibende Restfehler ist bei der quantitativen Analyse der resultierenden Tiefenkarten entsprechend zu berücksichtigen.

Für die untersuchten Tiefenkarten ergibt sich eine mittlere Skalierung von 1.05. Dies könnte auf eine unzureichende Kalibrierung oder eine ungenaue Angabe der lateralen Auflösung des zur Verifikation eingesetzten Streifenprojektionsmikroskops hinweisen. Die potenziellen Ursachen der

Skalierungsabweichung sollten in zukünftigen Arbeiten weitergehend untersucht werden.

6.2 Fehlergrößen

Zur Bewertung der Genauigkeit der berechneten Tiefenkarte D im Vergleich zur Ground-Truth-Tiefenkarte D^* wurden verschiedene Fehlermaße verwendet. Die Gesamtanzahl der Pixel in den Tiefenkarten wird mit N bezeichnet. Zum Einsatz kamen gebräuchliche Metriken für Vorhersagemodelle, darunter der MSE (siehe Gleichung 12), der Root Mean Squared Error (RMSE) sowie der MAE (siehe Gleichung 11). Darüber hinaus wurden zwei weiterführende Metriken berücksichtigt: der größenordnungsunabhängige Fehler (scale-invariant error, sc-inv) nach Eigen, Puhrsch und Fergus (2014) sowie der skalierungs- und translationsinvariante Fehler ssi_trim nach Ranftl u. a. (2020).

Der RMSE ist die Quadratwurzel des MSE und besitzt somit dieselbe Einheit wie die vorhergesagten Werte, was die Interpretation erleichtert:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(D_i - D_i^*)^2} \tag{41}$$

Eigen, Puhrsch und Fergus (2014) beobachteten, dass der Fehler bei der Tiefenschätzung stark von der Größenordnung der Szene abhängt. Aus diesem Grund führten sie den sc-inv ein, der auf logarithmierten Tiefenwerten basiert:

$$\text{sc-inv} = \frac{1}{N}\sum_{i=1}^{N}\left(\log D_i - \log D_i^* + \alpha\right)^2 \quad \text{mit} \quad \alpha = \frac{1}{N}\sum_{i=1}^{N}(\log D_i^* - \log D_i) \tag{42}$$

Trotz einer Transformation der mit dem GeoFocus3D-Datensatz berechneten Tiefenkarten verbleibt ein Ausrichtungsfehler gegenüber den Ground-Truth-Tiefenkarten. Ranftl u. a. (2020) schlugen daher die Fehlergröße ssi_trim vor, welche sowohl Skalierungs- als auch Translationsdifferenzen berücksichtigt. Zunächst werden die verbleibende Skalierung $\tilde{s}(D)$ und Translation $\tilde{t}(D)$ geschätzt:

$$\tilde{t}(D) = \text{median}(D),\ \tilde{s}(D) = \frac{1}{N}\sum_{i=1}^{N}\left|D_i - \tilde{t}(D)\right| \tag{43}$$

Anschließend erfolgt die Transformation der Tiefenkarten zur Entfernung von Skalierungs- und Translationskomponenten:

$$\tilde{D} = \frac{D - \tilde{t}(D)}{\tilde{s}(D)},\ \tilde{D}^* = \frac{D^* - \tilde{t}(D^*)}{\tilde{s}(D^*)} \tag{44}$$

Der absolute Fehler wird auf den transformierten Karten berechnet. Um die Robustheit gegenüber Ausreißern zu erhöhen, fließen lediglich die 80 % der kleinsten absoluten Fehler in die Berechnung des Erwartungswerts ein:

$$\text{ssi\_trim} = \frac{1}{2N} \sum_{i=1}^{0.8N} |\tilde{D}_i - \tilde{D}_i^*| \text{ mit } |\tilde{D}_i - \tilde{D}_i^*| \leq |\tilde{D}_{i+1} - \tilde{D}_{i+1}^*| \tag{45}$$

6.3 Vergleich der Methoden anhand des Micro-Topo-Datensatzes

Verwendete Fokalstapel. Für die Evaluation wurden fünf verschiedene Verfahren anhand des Testdatensatzes des Micro-Topo-Datensatzes untersucht. Aufgrund methodenspezifischer Anforderungen an die Eingabedaten mussten die Fokalstapel vor der Tiefenkartenberechnung jeweils angepasst werden.

Für die konventionelle Methode DFF-WLLSR besteht die einzige Einschränkung darin, dass die Fokusabstände zwischen den Aufnahmen konstant sein müssen. Daher konnte der gesamte synthetische Fokalstapel, der mit einer Schrittweite von 0.5 mm erzeugt wurde, direkt verwendet werden. Im Mittel umfassen die Fokalstapel 9.8 Bilder. Die Originalauflösung der Aufnahmen blieb erhalten. Da der von DFF-WLLSR verwendete MLS-Filter an den Bildrändern jeweils 20 Pixel abschneidet, werden diese Randbereiche in den resultierenden Tiefenkarten mit Not-a-Number-Werten aufgefüllt, sodass die Ausgabegröße mit der der Ground-Truth-Tiefenkarten übereinstimmt (in der Regel 763 × 1024 Pixel).

Das Modell DFVNet – sowohl in der vortrainierten als auch in der erneut trainierten Variante – kann ausschließlich mit Fokalstapeln bestehend aus fünf Bildern im Format 224 × 224 Pixel arbeiten. Zur Durchführung der Tests wurden jeweils fünf Bilder gleichmäßig über den Messbereich aus den Fokalstapeln ausgewählt und auf die erforderliche Eingabegröße herunterskaliert. Die resultierenden Tiefenkarten wurden anschließend zur Vergleichbarkeit mit den Ground-Truth-Tiefenkarten wieder auf die ursprüngliche Auflösung hochskaliert.

Für das vortrainierte Modell DDFFintheWild gilt die Einschränkung, dass ausschließlich Fokalstapel mit zehn Bildern verarbeitet werden können. Daher wurden jeweils zehn Bilder pro Stapel gleichmäßig über den Messbereich ausgewählt. Da jedoch nur 49 der Fokalstapel im Testdatensatz über mindestens zehn Aufnahmen verfügen, reduzierte sich die für diese Methode nutzbare Testmenge entsprechend. Die Bildauflösung wurde in diesem Fall beibehalten.

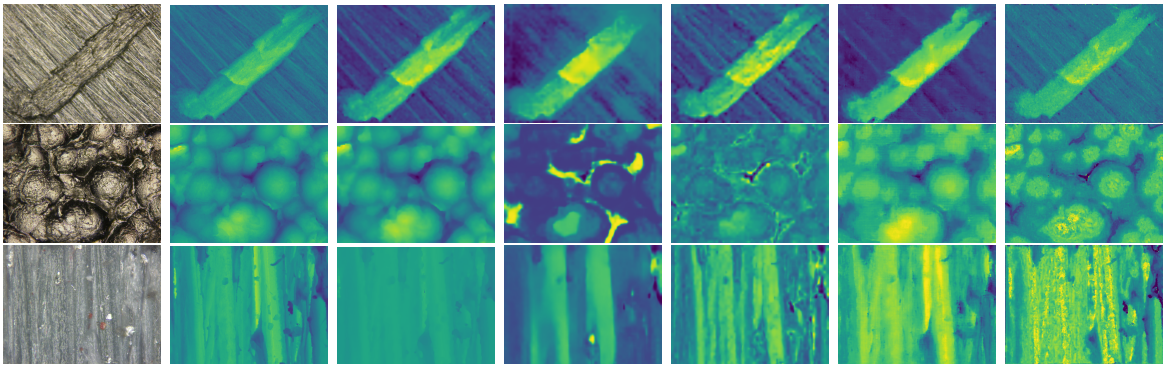
DDFFintheWild wurde unter Verwendung von Fokalstapeln aus fünf Bildern neu trainiert. Für die Evaluierung wurden ebenfalls fünf Bilder gewählt, die gleichmäßig über den gesamten Messbereich verteilt sind. Auch hierbei konnte die Auflösung beibehalten werden.

Rechenzeit. Die Berechnung einer Tiefenkarte mittels der DFF-WLLSR-Methode benötigte ungefähr 3 Minuten. Im Vergleich dazu verlief die Tiefenermittlung mit den Deep-Learning-Modellen wesentlich schneller. Mithilfe von DFVNet konnte eine Tiefenkarte innerhalb von etwa 70 ms erstellt werden, während DDFFintheWild diese in ungefähr 5 Sekunden bestimmte.

Evaluation der Genauigkeit. Zur Auswertung der Genauigkeit der Tiefenkarten wurde das Programm `Evaluation-Micro-Topo.py` eingesetzt. Da der maximale Messbereich bei der

Tabelle 10: Fehlergrößen auf dem Micro-Topo-Datensatz

Methode	MAE [mm]	MSE [mm^2]	RMSE [mm]	sc_inv	ssi_trim
DFF-WLLSR-0.5	0.17	0.11	0.26	0.05	0.19
DFVNet-vortrainiert	0.67	1.15	0.78	0.10	0.34
DFVNet	0.49	0.26	0.59	0.10	0.28
DDFFintheWild-vortrainiert	1.09	0.22	0.39	0.06	0.24
DDFFintheWild	0.28	0.22	0.39	0.06	0.24



(a) (b) (c) (d) (e) (f) (g)

Abbildung 39: Vergleich der ermittelten Tiefenkarten des Micro-Topo-Datensatzes mit dem RGB-Bild (a), der Ground-Truth-Tiefenkarte (b) und den mit der DFF-WLLSR-Methode (c), dem vortrainierten DFVNet (d), dem neu trainierten DFVNet (e), dem vortrainierten DDFFintheWild (f) und dem neu trainierten DDFFintheWild (g) berechneten Tiefenkarten

Generierung der synthetischen Fokalstapel auf 7 mm begrenzt wurde, erfolgte die Auswertung ausschließlich in jenen Bereichen, in denen die Ground-Truth-Daten ebenfalls innerhalb des Bereichs von 0 bis 7 mm liegen.

Die ermittelten Fehlergrößen sind in Tabelle 10 zusammengefasst. Die konventionelle DFF-Methode erzielte dabei eine höhere Genauigkeit als die untersuchten Deep-Learning-gestützten Modelle. Allerdings konnten die neu trainierten DDFF-Modelle gegenüber den vortrainierten Varianten eine verbesserte Tiefenschätzung erreichen. Zwischen den beiden neu trainierten Verfahren erzielte DDFFintheWild im Test auf dem Micro-Topo-Datensatz leicht bessere Ergebnisse als DFVNet.

In der Abbildung 39 sind für drei Szenen aus dem Micro-Topo-Datensatz das RGB-Bild (a), die Ground-Truth-Tiefenkarte (b) und die mit der DFF-WLLSR-Methode (c), dem vortrainierten DFVNet (d), dem neu trainierten DFVNet (e), dem vortrainierten DDFFintheWild (f) und dem neu trainierten DDFFintheWild (g) berechneten Tiefenkarten dargestellt.

6.4 Vergleich der Methoden anhand des GeoFocus3D-Datensatzes

Verwendete Fokalstapel. Auf Grundlage des GeoFocus3D-Datensatzes wurden insgesamt sechs verschiedene Konfigurationen evaluiert. Die konventionelle Methode DFF-WLLSR wurde

dabei mit Fokalstapeln getestet, deren Fokusabstände Schrittweiten von 0.1 mm bzw. 0.5 mm aufwiesen. Die Fokalstapel mit einer Schrittweite von 0.1 mm bestanden durchschnittlich aus 71 Aufnahmen, während die Stapel mit 0.5 mm Schrittweite im Mittel 14 Bilder enthielten. Die Verwendung der größeren Schrittweite diente der Untersuchung der Leistungsfähigkeit der DFF-WLLSR-Methode bei reduzierter Bildanzahl. Nach dem Auffüllen der Randbereiche zur Korrektur des durch die Filterung verursachten Zuschnitts weisen die mit der konventionellen Methode berechneten Tiefenkarten eine Auflösung von 984×1107 Pixeln auf, entsprechend der Größe der jeweiligen Ground-Truth-Tiefenkarten.

Die Modelle DFVNet und DDFFintheWild wurden in analoger Weise wie beim Micro-Topo-Datensatz getestet. Für das vortrainierte Modell DDFFintheWild konnten im Gegensatz zum vorherigen Datensatz jedoch alle Fokalstapel für die Evaluation verwendet werden, da sie jeweils über mehr als zehn Aufnahmen verfügen.

Evaluation der Genauigkeit. Die generierten Tiefenkarten wurden über den gesamten Tiefenbereich evaluiert. Die Proben 4 und 6 wurden jedoch von der quantitativen Evaluation ausschlossen, da das Ausrichten der Tiefenkarten mit einer zu großen Unsicherheit behaftet war. Zur Berechnung der Fehlergrößen wurde das Programm `Evaluation-GeoFocus3D.py` verwendet.

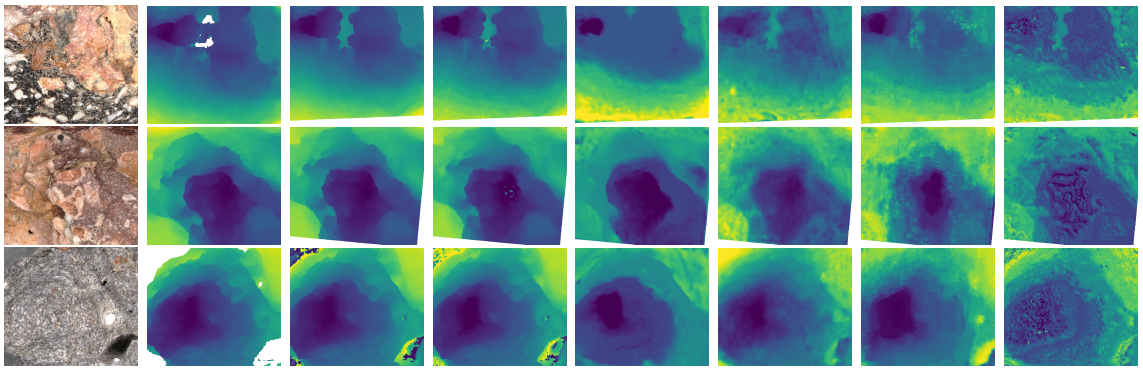
In der Tabelle 11 sind die ermittelten Fehlergrößen zusammengefasst. Auch für GeoFocus3D konnte die konventionelle Methode sowohl mit einer Schrittweite von 0.1 mm als auch mit 0.5 mm die höchste Genauigkeit in der Tiefenberechnung erzielen. Darüber hinaus weisen auch hier die neu trainierten Deep-Learning-Modelle eine überlegene Fähigkeit zur Generalisierung auf neue Daten auf, im Vergleich zu den von den Autoren publizierten vortrainierten Modellen. Hierbei konnte diesmal DFVNet bessere Ergebnisse erzielen.

In der Abbildung 40 sind für drei Proben die RGB-Bilder (a), die Ground-Truth-Tiefenkarten (b), und die mit dem DFF-WLLSR-Methode mit einer Schrittweite von 0.1 mm (c) und 0.5 mm (d), mit dem vortrainierten DFVNet (e), dem neu trainierten DFVNet (f), dem vortrainierten DDFFintheWild (g) und dem neu trainierten DDFFintheWild berechneten Tiefenkarten (h) dargestellt. Im Anhang befindet sich eine Übersicht zu den übrigen berechneten Tiefenkarten des GeoFocus3D-Datensatzes. Es lässt sich feststellen, dass die konventionelle Methode DFF-WLLSR Tiefenkarten erzeugt, die glatter sind und enger mit den mittels Streifenprojektion generierten Ground-Truth-Tiefenkarten übereinstimmen als jene, die durch Deep-Learning-Ansätze resultieren. Besonders auffällig ist, dass die durch DDFFintheWild produzierten Tiefenkarten erhebliches Rauschen aufweisen.

Für die Proben 4 und 6 wurde ausschließlich ein qualitativer Vergleich durchgeführt, da aufgrund einer unzureichenden Anzahl detektierter Keypoints keine zuverlässige Ausrichtung der berechneten Tiefenkarten auf die entsprechenden Ground-Truth-Daten möglich war. Die resultierenden Tiefenkarten sind in Abbildung 41 dargestellt. Beide Proben weisen keine ausgeprägten Tiefensprünge auf, sondern zeigen einen gleichmäßigen, graduellen Tiefenverlauf. Dieser Verlauf wird am

Tabelle 11: Fehlergrößen auf dem GeoFocus3D-Datensatz

Methode	MAE [mm]	MSE [mm^2]	RMSE [mm]	sc_inv	ssi_trim
DFF-WLLSR-0.1	0.47	0.63	0.76	0.26	0.13
DFF-WLLSE-0.5	0.51	0.71	0.80	0.38	0.14
DFVNet-vortrainiert	1.13	2.72	1.45	0.34	0.20
DFVNet	0.61	0.80	0.83	0.40	0.17
DDFFintheWild-vortrainiert	0.85	1.35	1.11	0.47	0.17
DDFFintheWild	0.75	1.13	1.01	0.49	0.23



(a) (b) (c) (d) (e) (f) (g) (h)

Abbildung 40: Vergleich der Tiefenkarten dreier Proben des GeoFocus3D-Datensatzes mit den RGB-Bildern (a), den Ground-Truth-Tiefenkarten (b), und den mit dem DFF-WLLSR-Methode mit einer Schrittweite von 0.1 mm (c) und 0.5 mm (d), mit dem vortrainierten DFVNet (e), dem neu trainierten DFVNet (f), dem vortrainierten DDF-FintheWild (g) und dem neu trainierten DDFFintheWild (h) berechneten Tiefenkarten

besten durch die konventionelle Methode DFF-WLLSR erfasst. Auch die mit dem neu trainierten DFVNet generierten Tiefenkarten zeigen einen weitgehend stufenlosen Tiefenverlauf, allerdings überlagert von texturähnlichen Mustern, die in den Ground-Truth-Karten nicht vorhanden sind. Bei den übrigen getesteten Deep-Learning-basierten Verfahren fällt der Tiefenverlauf hingegen deutlich stufenförmiger aus. Insbesondere bei dem neu trainierten DDFFintheWild-Modell ist zudem ein höheres Maß an Rauschen in den Tiefenkarten erkennbar.

6.5 Diskussion der Ergebnisse

Im Kapitel zum Stand der Technik wurde gezeigt, dass CNN-basierte DFF-Methoden in der Regel eine höhere Genauigkeit erzielen als konventionelle Verfahren. In dieser Arbeit hingegen konnten bei beiden untersuchten Datensätzen die höchsten Genauigkeiten mit der konventionellen Methode DFF-WLLSR erzielt werden. Allerdings wurden für das konventionelle Verfahren Fokalstapel mit einer höheren Bildanzahl verwendet, was zu einem größeren Informationsgehalt führte und die Genauigkeit positiv beeinflusst haben könnte. Zudem war für die Anwendung von DFVNet eine starke Reduktion der Bildgröße erforderlich, wodurch viele feine Bilddetails verloren gingen.

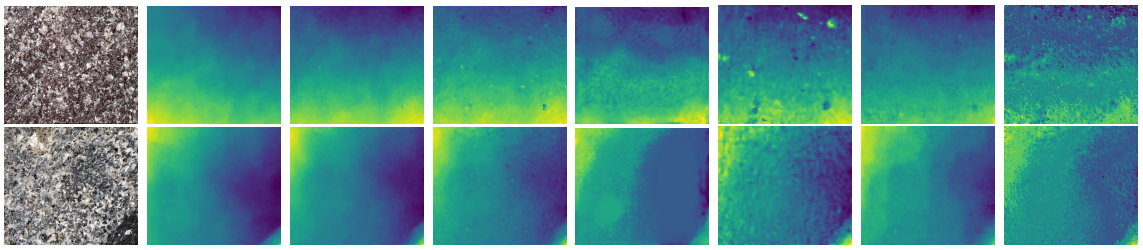


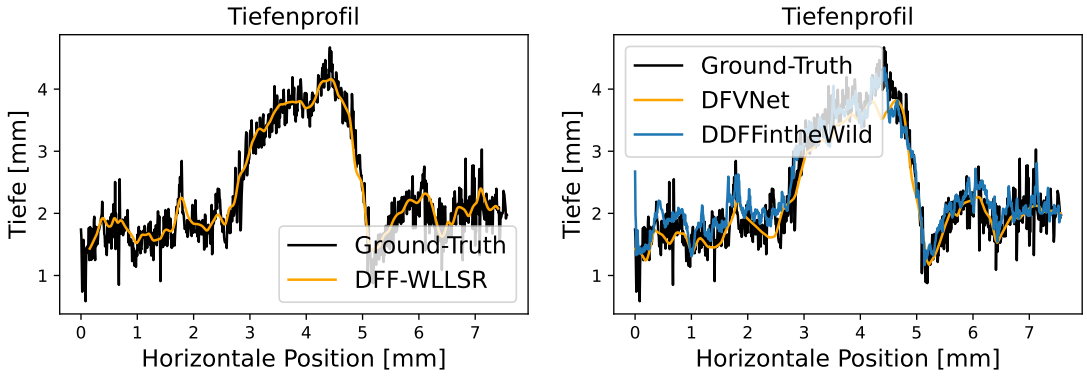
Abbildung 41: Vergleich der Tiefenkarten der Proben 4 und 6 des GeoFocus3D-Datensatzes mit den RGB-Bildern (a), den Ground-Truth-Tiefenkarten (b), und den mit dem DFF-WLLSR-Methode mit einer Schrittweite von 0.1 mm (c) und 0.5 mm (d), mit dem vortrainierten DFVNet (e), dem neu trainierten DFVNet (f), dem vortrainierten DDFFintheWild (g) und dem neu trainierten DDFFintheWild (h) berechneten Tiefenkarten

Darüber hinaus zeigten die neu-trainierten Deep-Learning-Modelle Anzeichen von Overfitting. Dies reduzierte ihre Übertragbarkeit auf neue Datensätze und führte zu geringerer Genauigkeit. Besonders deutlich wird dies im Vergleich der Modellleistung auf den beiden evaluierten Datensätzen: Die mit Micro-Topo trainierten Modelle lieferten auf dem GeoFocus3D-Datensatz deutlich schlechtere Ergebnisse. Dies weist außerdem darauf hin, dass die Trainingsdaten nicht ausreichend repräsentativ für die Zielanwendung waren.

Ein möglicher Grund dafür liegt in den Unterschieden der Bildinhalte beider Datensätze. Während Micro-Topo fein strukturierte und meist gleichmäßig verteilte Oberflächenmerkmale enthält, weisen die geologischen Proben des GeoFocus3D-Datensatzes gröbere Strukturen und stärkere Höhenkontraste auf. Hinzu kommt, dass die Ground-Truth-Tiefenkarten von Micro-Topo im Vergleich zu GeoFocus3D deutlich mehr Rauschen enthalten. Dieses Rauschen sowie die feinen Oberflächenmerkmale könnten von den Modellen mitgelernt worden sein und sich bei der Anwendung negativ auswirken. Die glatten Tiefenkarten von GeoFocus3D wurden mithilfe eines kommerziellen Streifenprojektionsmikroskops erzeugt, das als aktives Messverfahren weitgehend unempfindlich gegenüber Textur ist und interne Filter zur Rauschunterdrückung verwenden dürfte.

Abbildung 42 illustriert die Tiefenprofile der Ground-Truth-Daten sowie der berechneten Tiefenkarten einer Probe aus dem Micro-Topo-Datensatz. Insbesondere wird das ausgeprägte Rauschverhalten der Ground-Truth-Tiefenkarte deutlich sichtbar. Während das Modell DDFFintheWild dieses Rauschen rekonstruiert, erzeugen die Modelle DFF-WLLSR und DFVNet hingegen glattere Tiefenkarten.

Besonders auffällig ist, dass die DDFF-Modelle Schwierigkeiten haben, graduelle Tiefenverläufe – wie sie bei den Proben 4 und 6 auftreten – korrekt abzubilden. Dies deutet darauf hin, dass derartige Szenenverläufe in den Trainingsdaten nicht ausreichend repräsentiert waren. Eine unzureichende Abdeckung solcher Verläufe im Trainingsdatensatz führt dazu, dass die Modelle diese auch in neuen Daten schlechter generalisieren.



(a) (b)

Abbildung 42: Tiefenprofile der Ground-Truth-Tiefenkarte und der mit den verschiedenen Methoden berechneten Tiefenkarten einer Probe des MicroTopo-Datensatzes

Ein weiterer möglicher Einflussfaktor ist die vereinfachte Modellierung bei der Generierung der synthetisch unscharfen Bilder. Beispielsweise wurden Änderungen des CoC durch die Bewegung des Aktuator nicht berücksichtigt, ebenso wenig wie realitätsnahe, asymmetrische Defokussierungseffekte. Diese Vereinfachungen können dazu führen, dass die Modelle nicht in der Lage sind, Bildschärfe in realen Messungen zuverlässig zu bewerten.

Nichtsdestotrotz erzielten die mit Micro-Topo neu trainierten Modelle bessere Ergebnisse als die ursprünglich veröffentlichten, vortrainierten Varianten. Dies spricht dafür, dass der Datensatz insgesamt geeigneter für geologische Proben ist als die bisherigen Trainingsdatensätze, die überwiegend aus Alltagsszenen oder CAD-Geometrien bestanden. Durch die Generierung synthetisch unscharfer Bilder unter Berücksichtigung der tatsächlichen Kameraparameter konnten zudem realitätsnahe Trainingsdaten erzeugt werden, die die Modellergebnisse verbessern.

Trotz der besten erzielten Genauigkeit durch die Methode DFF-WLLSR-0.1 ist der mittlere absolute Fehler mit 0.47 mm noch immer relativ hoch. Ein Teil dieses Fehlers lässt sich auf verbleibende Ausrichtungsfehler zwischen berechneter und Ground-Truth-Tiefenkarte zurückführen. Für Probe 12 wurde eine manuelle Korrektur der Ausrichtung anhand des Tiefenprofils vorgenommen, wodurch der MAE auf 0.10 mm reduziert werden konnte. Dennoch liegt dieser Wert weiterhin fast um den Faktor 10 über der theoretischen Tiefenauflösung von 12 µm. Wie in der Fehlerkarte in Abbildung 43 erkennbar, treten die größten Abweichungen insbesondere an Objektkanten auf.

Eine Ursache hierfür könnte in der veränderten Bildvergrößerung unscharfer Bildbereiche liegen. Während fokussierte Punkte konstant abgebildet werden, können unscharfe Punkte abhängig von ihrer Tiefe entweder vergrößert oder verkleinert erscheinen. Das führt dazu, dass ein Objektpunkt in den verschiedenen Aufnahmen des Fokalstapels auf unterschiedliche Pixel projiziert wird – ein Umstand, der die Tiefenschätzung insbesondere an Kanten stark beeinträchtigen kann.

Hinzu kommt der sogenannte Verdeckungseffekt: Bei Kanten mit großen Höhenunterschieden kann die scharf gestellte untere Ebene durch den unscharfen Bereich der darüberliegenden Ebene

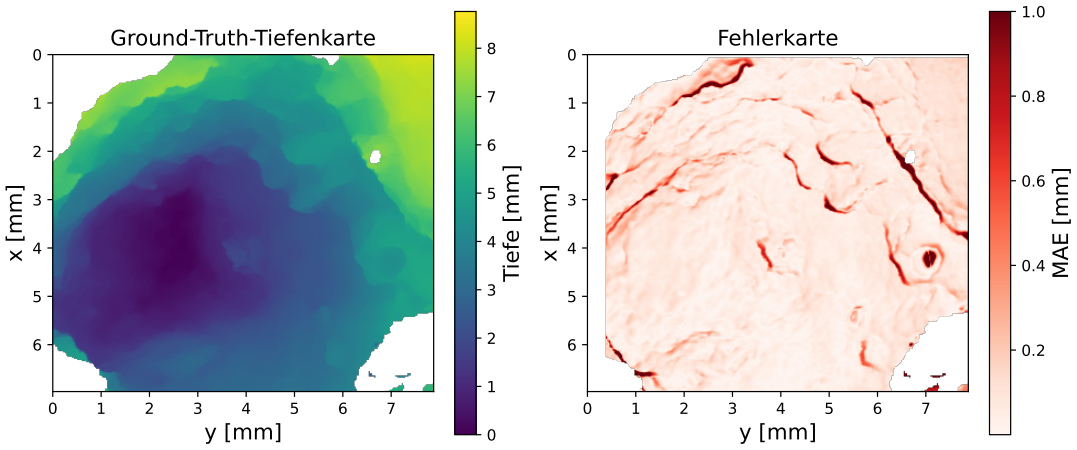


Abbildung 43: Ground-Truth-Tiefenkarte und mittlerer absoluter Fehler der mit DFF-WLLSR-0.1 berechneten und manuell ausgerichteten Tiefenkarte

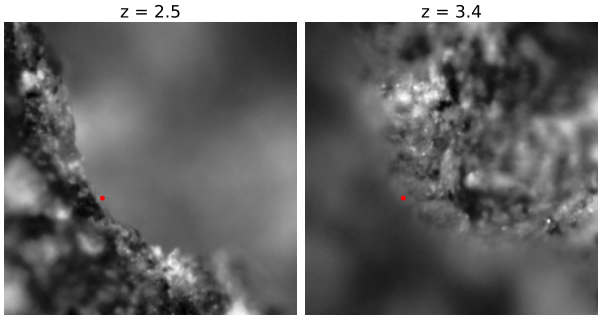
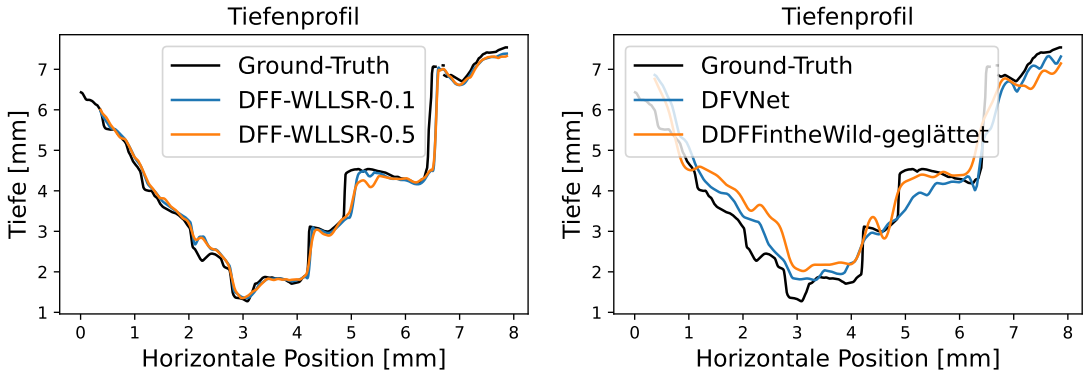


Abbildung 44: Darstellung der Verdeckung durch unscharfe Bereiche anhand von Bildausschnitten aus dem Fokalstapel der Probe 12. Rot: Markierung der Kante

teilweise verdeckt werden. In solchen Fällen ist eine korrekte Tiefenschätzung nicht mehr möglich. Abbildung 44 illustriert diesen Effekt, wobei die Kante mit einem roten Punkt markiert ist. Wird die untere Ebene fokussiert, überlagert der unscharfe Bereich der oberen Ebene den relevanten Bildinhalt.

Die manuelle Ausrichtungsanpassung wurde auf alle Tiefenkarten angewendet, die für Probe 12 berechnet wurden. In Abbildung 45a ist ein horizontaler Profilschnitt der Ground-Truth-Tiefenkarte im Vergleich zu den korrigierten Tiefenprofilen der Methode DFF-DDFF dargestellt. Abbildung 45b zeigt zudem die korrigierten Tiefenprofile der Modelle DFVNet und DDFFintheWild, wobei Letzteres zusätzlich mit einem Gauß-Filter zur Rauschreduktion nachbearbeitet wurde.

Auch im Profilschnitt lassen sich bei der konventionellen Methode deutliche Fehler an steilen Kanten erkennen. Im Gegensatz dazu zeigen die mit den DDFF-Methoden berechneten Tiefenprofile insgesamt einen glatteren Verlauf, wobei charakteristische Höhen und Tiefen des tatsächlichen Profils häufig nicht korrekt erfasst werden. Bei DFVNet lässt sich dies möglicherweise auf den Verlust lateraler Auflösung infolge der notwendigen Bildskalierung zurückführen. Bei DDFFintheWild hingegen könnte die Abweichung durch die nachträgliche Glättung der Tiefenkarte mittels



(a) (b)

Abbildung 45: Tiefenprofile der Ground-Truth-Tiefenkarte und der mit den verschiedenen Methoden berechneten Tiefenkarten einer Probe des GeoFocus3D-Datensatzes

Filterung bedingt sein, die zur Reduktion des ausgeprägten Rauschens erforderlich war.

Die Diskussion hat gezeigt, dass die konventionelle DFF-Methoden unter bestimmten Bedingungen – insbesondere bei höherem Informationsgehalt der Eingabedaten – eine höhere Genauigkeit erzielen konnte als Deep-Learning-basierte Verfahren. Gleichzeitig wurde deutlich, dass die Generalisierbarkeit der trainierten DDFF-Modelle stark von der Qualität und Repräsentativität der Trainingsdaten abhängt. Unterschiede in den Bildinhalten, vereinfachte Modellannahmen sowie architekturspezifische Schwächen führten insbesondere bei Anwendung auf reale geologische Proben zu merklichen Abweichungen. Zudem wirken sich physikalische Effekte wie defokusbedingte Vergrößerung und Verdeckung negativ auf die Genauigkeit aus. Insgesamt zeigt sich, dass sowohl die Wahl des Trainingsdatensatzes als auch die Modellarchitektur entscheidend für die Leistungsfähigkeit Deep-Learning-gestützter DFF-Verfahren sind.

7 Zusammenfassung

DDFF-Algorithmus zur Tiefenmessung geologischer Proben. Für ein experimentelles 3D-Mikroskop wurden zwei DDFF-Algorithmen implementiert, die mittels CNNs Bildschärfe-Merkmale extrahieren und daraus die Objekttiefe bestimmen. Grundlage war eine umfassende Literaturrecherche zu bestehenden DDFF-Verfahren, die eine bislang vermutlich einzigartige Übersicht über alle veröffentlichten Modelle lieferte. Auf Basis definierter Vergleichskriterien wurden die Methoden DFVNet und DDFFintheWild für die Implementierung ausgewählt. Während DFVNet Gradienteninformationen zur Schätzung der Schärfe verwendet, nutzt DDFFintheWild 3D-Faltungen, um Informationen zwischen benachbarten Bildern im Fokalstapel auszutauschen.

Für beide Modelle standen vortrainierte Versionen zur Verfügung. Da diese jedoch auf Datensätzen mit Innenraumszenen und abstrakten CAD-Geometrien basierten, erfolgte ein erneutes Training mit dem Datensatz Micro-Topo, dessen Bildinhalte eine höhere Ähnlichkeit zu geologischen Proben aufweisen. Micro-Topo enthält mikroskopische 3D-Messungen verschiedener Materialien mit unterschiedlichen Oberflächenstrukturen.

Da dieser Datensatz lediglich AiF-RGB-Bilder und zugehörige Tiefenkarten enthält, wurden synthetische Fokalstapel erzeugt. Die Defokussierung erfolgte dabei unter Einbeziehung der tatsächlichen Kameraparameter des Mikroskops, wodurch ein realistischer Defokus-Effekt simuliert werden konnte. Dies ermöglichte die Erstellung eines umfangreichen Trainingsdatensatzes bei vergleichsweise geringem methodischem Aufwand und bildet eine wichtige Grundlage für die Tiefenschätzung durch die DDFF-Modelle.

Zur Validierung der Modelle entstand der Datensatz GeoFocus3D mit realen geologischen Proben. Die Bildaufnahme erfolgte mit dem experimentellen Mikroskopaufbau. Eine angepasste geometrische Kamerakalibrierung war erforderlich, um verzeichnungsfreie Aufnahmen zu ermöglichen, da die geringe Schärfentiefe des Systems die möglichen Posen des Kalibriermusters stark einschränkte. Darüber hinaus wurde die Positionsgenauigkeit des Fokusmechanismus mithilfe eines konfokalen Wegmesssensors ermittelt. Die Ergebnisse zeigten, dass die erreichbare Tiefenauflösung in erster Linie durch die Schärfentiefe und nicht durch die Positioniergenauigkeit des Aktuators begrenzt wird. Die Ground-Truth-Tiefenkarten wurden mit einem kommerziellen Streifenprojektionsmikroskop aufgenommen.

Für den Vergleich mit den berechneten Tiefenkarten wurde eine Ausrichtungsstrategie entwickelt, bei der anhand von Keypoint-Paaren eine Transformation zwischen den Tiefenkarten bestimmt wurde.

Die mit Micro-Topo trainierten Modelle wurden anschließend auf GeoFocus3D angewendet und mit den vortrainierten Modellen der ursprünglichen Autoren verglichen. Dabei konnten beide neu trainierten Modelle eine höhere Genauigkeit erzielen, was die Bedeutung passender Bildinhalte im Trainingsdatensatz sowie der realitätsnahen Modellierung des Defokus-Effekts unterstreicht.

Zusätzlich wurden die DDFF-Modelle mit einer klassischen DFF-Methode ohne Deep Learning

verglichen. Dabei erreichte die konventionelle Methode die höchste Genauigkeit, was im Widerspruch zu bisherigen Ergebnissen aus der Literatur steht. Als eine der Ursachen ist der höhere Informationsgehalt der Fokalstapel bei der klassischen Methode zu nennen, da dort mit mehr Aufnahmen gearbeitet werden konnte.

Die Ergebnisse zeigten darüber hinaus, dass die Leistungsfähigkeit der DDFF-Modelle stark von der Qualität und Repräsentativität der Trainingsdaten abhängt. Während auf Micro-Topo akzeptable Ergebnisse erzielt wurden, zeigten sich deutliche Generalisierungsschwächen auf GeoFocus3D. Gründe hierfür liegen unter anderem in den Unterschieden der Bildinhalte, im stärkeren Rauschen der Trainingsdaten sowie in vereinfachenden Modellannahmen bei der synthetischen Defokussierung.

Gleichzeitig konnten die DDFF-Methoden im Hinblick auf Aufnahmezeit, Speicherbedarf und Rechenzeit klare Vorteile gegenüber der konventionellen Methode aufweisen: Es wurden lediglich fünf unscharfe Bilder benötigt und die Berechnung der Tiefenkarten erfolgte in wenigen Sekunden – gegenüber mehreren Minuten bei der klassischen Methode.

Genauigkeit der DFF-Methoden. Nach manueller Ausrichtung der berechneten Tiefenkarten konnte mit der konventionellen Methode ein minimaler mittlerer absoluter Fehler von 0.1 mm erreicht werden – etwa zehnmal höher als die theoretisch mögliche Tiefenauflösung von 12 µm. Als begrenzende Faktoren wurden insbesondere Ausrichtungsfehler, defokusbedingte Bildvergrößerung und Verdeckungseffekte identifiziert, die vor allem an Objektkanten zu signifikanten Fehlern führen.

Multispektrale Beleuchtung. Zur Erweiterung der Funktionalität des Mikroskops wurde ein multispektraler LED-Ring mit drei Wellenlängenbereichen im Infrarot konstruiert, montiert und elektronisch ansteuerbar gemacht. Damit können in zukünftigen Arbeiten Falschfarbenbilder erzeugt und spektrale Informationen der Proben analysiert werden.

Anwendung des Deep-Learning-gestützten DFF-Mikroskops. Sowohl die konventionelle Methode als auch das leistungsfähigere DFF-Modell DFVNet wurden in ein Python-Interface integriert, mit dem nun 3D-Messungen mit dem experimentellen Mikroskop durchgeführt werden können. Während die klassische Methode maximale Genauigkeit ermöglicht, bietet die Deep-Learning-basierte Variante eine erhebliche Zeitersparnis und kann zur schnellen Orientierung und Auswahl relevanter Probenbereiche eingesetzt werden.

In Anwendungen außerhalb der Laborumgebung ist eine Abwägung zwischen Genauigkeit und Geschwindigkeit erforderlich. Während bei der Vermessung geologischer Proben oder in der industriellen Qualitätskontrolle die Präzision möglicherweise von größerer Bedeutung ist, kann zum Beispiel in der Robotik bei der Tiefenmessung zur räumlichen Orientierung die Geschwindigkeit von höherer Wichtigkeit sein.

8 Ausblick

Verbesserung der Genauigkeit der DDFF-Methoden. Zur Generierung der synthetischen Fokalstapel im Trainingsdatensatz wurden mehrere Modellvereinfachungen vorgenommen. Die PSF wurde beispielsweise als rotationssymmetrische Gaußsche Scheibe modelliert. Beobachtungen am realen Mikroskopsystem zeigen jedoch, dass der Defokus-Effekt asymmetrisch verläuft – bedingt durch die verschobene Lage der Blende. Für eine realistischere Simulation wäre es daher notwendig, die tatsächliche PSF des Systems experimentell zu bestimmen. Dies könnte etwa durch die Verwendung einer Punktlichtquelle erfolgen, deren Unschärfeverlauf im Bild systematisch analysiert wird.

Darüber hinaus wäre eine zusätzliche Vorverarbeitung der Tiefenkarten des Trainingsdatensatzes sinnvoll, um deren Charakteristika besser an reale geologische Proben anzugleichen. Die Tiefenkarten von Micro-Topo enthalten feine Strukturen und ein vergleichsweise hohes Maß an Rauschen. Durch die Anwendung glättender Filter könnten Trainingsdaten erzeugt werden, die strukturell stärker den glatteren Ground-Truth-Tiefenkarten des GeoFocus3D-Datensatzes entsprechen.

Zudem besteht Potenzial zur Verbesserung der Modellgeneralität durch gezielte Anpassung der Hyperparameter (z.B. Batchgröße, Lernrate) sowie durch erweiterte Datenaugmentation. Zusätzliche Transformationen wie künstliche Skalierung oder Rotation könnten die Robustheit erhöhen und das Risiko von Overfitting verringern (Ying 2019).

Zur Vermeidung von Detailverlusten durch das notwendige Downscaling bei DFVNet wäre der Einsatz einer Stitching-Strategie denkbar: Dabei würden Tiefenkarten segmentweise auf unskalierte Bildausschnitte berechnet und anschließend zusammengesetzt. Es bleibt zu evaluieren, ob durch dieses Vorgehen die Genauigkeit gesteigert oder stattdessen Artefakte an den Segmentgrenzen erzeugt werden.

Ein weiterer offener Forschungsaspekt ist der Einfluss der Fokalstapelgröße auf die Genauigkeit. Während Yang, Huang und Z. Zhou (2022) zeigten, dass über fünf Bilder hinaus keine signifikante Verbesserung zu erwarten ist, könnte diese Aussage für Szenarien mit großer Tiefenerstreckung und geringer Schärfentiefe – wie beim GeoFocus3D-Datensatz – nicht zutreffen. In solchen Fällen könnten einzelne Objektbereiche unzureichend erfasst werden. Insbesondere für DDFFintheWild wäre es daher vielversprechend, ein Verfahren zu entwickeln, das flexibel auf unterschiedliche Fokalstapelgrößen reagieren kann – idealerweise angepasst an den zu messenden Tiefenbereich.

Verbesserung von Geschwindigkeit und Genauigkeit der konventionellen Methode. Zur Reduktion der Rechenzeit bei der konventionellen DFF-Methode bietet sich der Einsatz von Parallelverarbeitung an. Die Bestimmung des Fokuswertmaximums pro Pixel ist unabhängig voneinander möglich und lässt sich effizient auf mehrere CPU-Kerne verteilen, wie von Matsubara u. a. (2022) beschrieben.

Zur Erhöhung der Tiefengenauigkeit könnte eine kleinere Schrittweite beim Erfassen des Fokalstapels gewählt werden. Die theoretisch maximale Auflösung wäre mit einer Schrittweite entsprechend

der Schärfentiefe erreichbar. Dies würde jedoch mit deutlich erhöhtem Speicher- und Zeitaufwand einhergehen.

Darüber hinaus ist die Korrektur von systematischen Fehlern an Objektkanten durch den defokusbedingten Verdeckungseffekt ein zentrales Thema. Eine konventionelle Lösung könnte auf Kantendetektion und anschließender Interpolation basieren. Ikoma u. a. (2021) schlägt alternativ den Einsatz eines CNNs vor, das mit Daten trainiert wurde, die die Verdeckung einbeziehen. Da die Defokussierung in dieser Arbeit bereits Verdeckungen einbezieht, könnte eine zusätzliche Optimierung der DDFF-Methoden das Verdeckungsproblem ebenfalls beheben.

Literatur

Alayrac, Jean-Baptiste, João Carreira und Andrew Zisserman (2019). „The Visual Centrifuge: Model-Free Layered Video Representations“. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Barron, Jonathan T. u. a. (2015). „Fast Bilateral-Space Stereo for Synthetic Defocus“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, S. 4466–4474.

Billiot, Bastien u. a. (2013). „3D Image Acquisition System Based on Shape from Focus Technique“. In: *Sensors* 13.4, S. 5040–5053.

Blender-Foundation (2018). *Blender - a 3D modelling and rendering package*.

Carvalho, Marcela u. a. (2019). „Deep Depth from Defocus: How Can Defocus Blur Improve 3D Estimation Using Dense Neural Networks?“ In: *Computer Vision – ECCV 2018 Workshops*. Bd. 11129, S. 307–323.

Ceruso, Sabato u. a. (2021). „Relative multiscale deep depth from focus“. In: *Signal Processing: Image Communication* 99, S. 116417.

Cou, Corentin und Gaël Guennebaud (2024). „Depth from Focus using Windowed Linear Least Squares Regressions“. In: *The Visual Computer* 40.2, S. 1205–1214.

Deng, Jia u. a. (2009). „ImageNet: a Large-Scale Hierarchical Image Database“. In: *IEEE Conference on Computer Vision and Pattern Recognition*, S. 248–255.

Eigen, David, Christian Puhrsch und Rob Fergus (2014). *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network*. URL: http://arxiv.org/abs/1406.2283.

Foerstner, W. (1987). „A fast operator for detection and precise location of distincs points, corners and center of circular features“. In: *Proc. of the Intercommission Conference on Fast Processing of Photogrammetric Data*, S. 281–305.

Fujimura, Yuki u. a. (2024). „Deep Depth from Focal Stack with Defocus Model for Camera-Setting Invariance“. In: *International Journal of Computer Vision* 132.6, S. 1970–1985.

Garg, Rahul u. a. (2019). „Learning Single Camera Depth Estimation Using Dual-Pixels“. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, S. 7627–7636.

Girshick, Ross (2015). „Fast R-CNN“. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, S. 1440–1448.

Gur, Shir und Lior Wolf (2019). „Single Image Depth Estimation Trained via Depth From Defocus Cues“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, S. 7683–7692. URL: https://github.com/shirgur/UnsupervisedDepthFromFocus.

Hara, Kensho, Hirokatsu Kataoka und Yutaka Satoh (2017). „Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition“. In: *ICCV 2017 Workshop*.

Harris, C. und M. Stephens (1988). „A Combined Corner and Edge Detector“. In: *Procedings of the Alvey Vision Conference 1988*, S. 23.1–23.6.

Hazirbas, Caner u. a. (2019). „Deep Depth from Focus“. In: *Computer Vision – ACCV 2018*, S. 525–541.

Herrmann, Charles u. a. (2020). „Learning to Autofocus“. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 2227–2236.

Honauer, Katrin u. a. (2017). „A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields“. In: *Computer Vision – ACCV 2016*, S. 19–34.

Ikoma, Hayato u. a. (2021). „Depth from Defocus with Learned Optics for Imaging and Occlusion-aware Depth Estimation“. In: *2021 IEEE International Conference on Computational Photography (ICCP)*, S. 1–12.

Kim, Youngjung u. a. (2018). „Deep Monocular Depth Estimation via Integration of Global and Local Predictions“. In: *IEEE Transactions on Image Processing* 27.8, S. 4131–4144.

Lin, Tsung-Yi u. a. (2017). *Feature Pyramid Networks for Object Detection*. URL: http://arxiv.org/abs/1612.03144.

Lowe, David G. (1. Nov. 2004). „Distinctive Image Features from Scale-Invariant Keypoints“. In: *International Journal of Computer Vision* 60.2, S. 91–110.

Mascarenhas, Sheldon und Mukul Agarwal (2021). „A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification“. In: *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*. Bd. 1, S. 96–99.

Matsubara, Yoichi u. a. (2022). „Pixel-wise parallel calculation for depth from focus with adaptive focus measure“. In: *Multidimensional Systems and Signal Processing* 33.1, S. 121–142.

Maximov, Maxim, Kevin Galim und Laura Leal-Taixe (2020). „Focus on Defocus: Bridging the Synthetic to Real Domain Gap for Depth Estimation“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, S. 1071–1080.

Mayer, Nikolaus u. a. (2016). „A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation“. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Moeller, Michael u. a. (2015). „Variational Depth from Focus Reconstruction“. In: *IEEE Transactions on Image Processing* 24.12, S. 5369–5378.

Muhammad, Asif und Tae-Sun Choi (1999). „Learning shape from focus using multilayer neural networks“. In: *Vision Geometry VIII*. Bd. 3811. SPIE, S. 366–375.

OpenCV (2025). *Feature Matching*. URL: https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html (besucht am 01. 04. 2025).

Pertuz, Said, Domenec Puig und Miguel Angel Garcia (2013). „Analysis of focus measure operators for shape-from-focus“. In: *Pattern Recognition* 46.5, S. 1415–1432.

Ranftl, René u. a. (2020). *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer*. URL: http://arxiv.org/abs/1907.01341.

Ryan, Conor u. a. (2024). „Technology Selection for Inline Topography Measurement with Rover-Borne Laser Spectrometers“. In: *Sensors* 24, S. 2872.

Sakurikar, Parikshit und P. J. Narayanan (2017). „Composite Focus Measure for High Quality Depth Maps“. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, S. 1623–1631.

Scharstein, Daniel u. a. (2014). „High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth“. In: *Pattern Recognition*. Cham, S. 31–42.

Shotton, Jamie u. a. (2013). „Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images“. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, S. 2930–2937.

Shrestha, Raju und Jon Yngve Hardeberg (2013). „Multispectral imaging using LED illumination and an RGB camera“. In: *Color and Imaging Conference* 21.

Siemens, Stefan, Markus Kästner und Eduard Reithmeier (2023). „RGB-D microtopography: A comprehensive dataset for surface analysis and characterization techniques“. In: *Data in Brief* 48, S. 109094.

Silberman, Nathan u. a. (2012). „Indoor Segmentation and Support Inference from RGBD Images“. In: *Computer Vision – ECCV 2012*, S. 746–760.

Simonyan, Karen und Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. URL: http://arxiv.org/abs/1409.1556.

Subbarao, M. und Tao Choi (1995). „Accurate recovery of three-dimensional shape from image focus“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.3, S. 266–274.

Surh, Jaeheung u. a. (2017). „Noise Robust Depth from Focus Using a Ring Difference Filter“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 2444–2453.

Suwajanakorn, Supasorn, Carlos Hernandez und Steven M. Seitz (2015). „Depth From Focus With Your Mobile Phone“. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 3497–3506.

The-SciPy-Community (2025). *least_ squares*. SciPy v1.15.2 Manual. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html#id12 (besucht am 17. 03. 2025).

Wang, Ning-Hsu u. a. (2021). „Bridging Unsupervised and Supervised Depth From Focus via All-in-Focus Supervision“. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, S. 12621–12631.

Weng, J., P. Cohen und M. Herniou (1992). „Camera calibration with distortion models and accuracy evaluation“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.10, S. 965–980.

Wohlfeil, Jürgen u. a. (2019). „Automatic Camera System Calibration with a Chessboard Enabling Full Image Coverage“. In: *Remote Sensing and Spatial Information Sciences* XLII-2/W13.

Won, Changyeon und Hae-Gon Jeon (2022). „Learning Depth from Focus in the Wild“. In: *Computer Vision – ECCV 2022*. Cham, S. 1–18. URL: https://github.com/wcy199705/DfFintheWild/tree/main.

Yang, Fengting, Xiaolei Huang und Zihan Zhou (2022). „Deep Depth From Focus With Differential Focus Volume“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, S. 12642–12651. URL: https://github.com/fuy34/DFV.

Yin, Wei u. a. (2019). „Enforcing geometric constraints of virtual normal for depth prediction“. In: *Proeedings of the International Conference of Computer Vision 2019*.

Ying, Xue (2019). „An Overview of Overfitting and its Solutions“. In: *Journal of Physics: Conference Series* 1168.2, S. 022022.

Zhao, Hengshuang u. a. (2017). „Pyramid Scene Parsing Network“. In: *CVPR 2017*.

Zhou, Bolei u. a. (2014). „Learning Deep Features for Scene Recognition using Places Database“. In: *Advances in Neural Information Processing Systems*. Bd. 27. Curran Associates, Inc.

Anhang

Erzeugung von Falschfarbenbildern

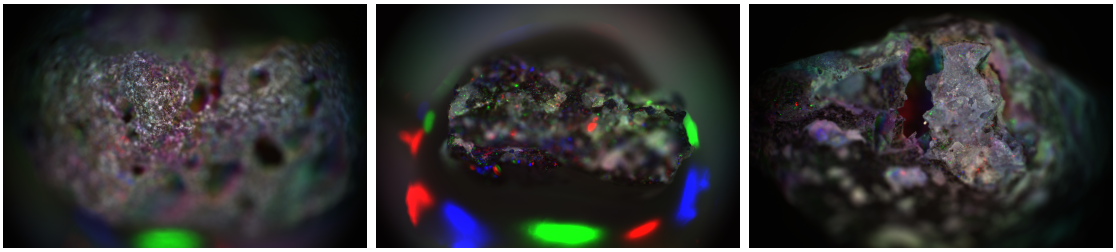
Aus den aufgenommenen Einzelbildern der verschiedenen Spektralkanäle des LED-Rings lassen sich Falschfarbenbilder erzeugen, indem die Kanäle den RGB-Farbkanälen zugewiesen werden. Die verwendete Zuordnung lautet:

- Rot: 940 nm
- Grün: 840 nm
- Blau: 760 nm

Zur Angleichung der Helligkeit der Spektralkanäle wurde ein Helligkeitsausgleich durchgeführt. Das resultierende Bild I_s für die Spektralbereiche $s = [940, 840, 760]$ wurde berechnet als:

$$I_s = \frac{\mu_s}{\max(\mu_s)} \cdot I_{\text{roh},s}$$

Dabei ist μ_s die mittlere Intensität des unbearbeiteten Bildes im jeweiligen Spektralbereich und $I_{\text{roh},s}$ die Rohaufnahme des jeweiligen Kanals. In der Abbildung 46 sind generierte Falschfarbenbilder dreier Proben dargestellt



(a) Probe 1 (b) Probe 5 (c) Probe 7

Abbildung 46: Falschfarbenbilder dreier Gesteinsproben

GeoFocus3D

Die Tabelle 12 führt die verwendeten Belichtungszeiten für die verschiedenen Beleuchtungskonfigurationen bei der Aufnahme des Fokalstapel auf. Außerdem sind die Größen der Fokalstapel aufgeführt.

DFVNet Netzwerkarchitektur

Die in der Abbildung 47 dargestellte Netzwerkarchitektur wurde dem Zusatzmaterial der Veröffentlichung von Yang, Huang und Z. Zhou (2022) entnommen. In der Darstellung steht B für die Batchgröße, N für die Anzahl der Bilder eines Fokalstapels sowie H und W für die Bildgröße. s bezeichnet das Skalenniveau.

Tabelle 12: Belichtungszeiten bei Aktivierung der einzelnen Spektralkanäle und aller LEDs sowie Fokalstapelgrößen für die Proben des GeoFocus3D-Datensatzes

Probe	Größe	940 nm [µs]	840 nm [µs]	760 nm [µs]	Alle LEDs [µs]
1	77	34000	7000	4200	2400
2	71	30000	5650	3300	1900
3	47	14000	3400	2050	1170
4	47	18000	4000	2000	1300
5	52	10000	4700	1800	1400
6	67	11000	1900	1000	900
7	67	17000	3100	1600	1200
8	67	17000	3100	1600	1200
9	67	16000	3600	2000	1200
10	61	18000	3750	2250	1300
11	81	18000	3900	2100	1250
12	91	33000	6600	4800	2600
13	61	33000	6600	4800	2600

DDFFintheWild Netzwerkarchitektur

Eine Darstellung des Netzwerkaufbaus von DDFFintheWild aus dem Zusatzmaterial von Won und Jeon (2022) ist in Abbildung 48 veranschaulicht.

Vergleich der Tiefenkarten des GeoFocus3D Datensatzes

In der Abbildung 49 sind die die berechneten Tiefenkarten der übrigen Proben des GeoFocus3D-Datensatzes dargestellt, welche nicht im Kapitel 6.4 visualisiert sind.

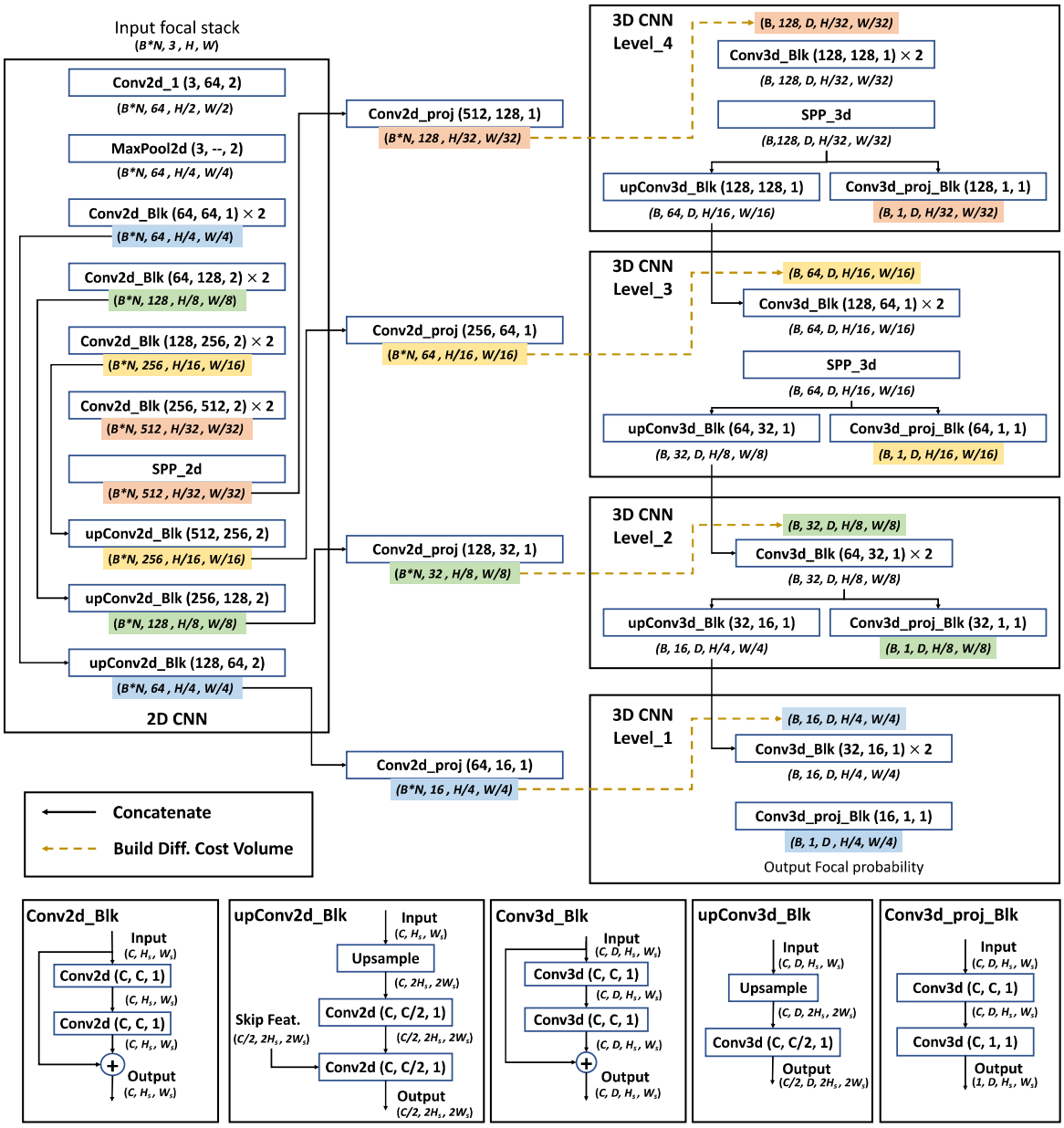


Abbildung 47: Netzwerkarchitektur von DFVNet (Yang, Huang und Z. Zhou 2022)

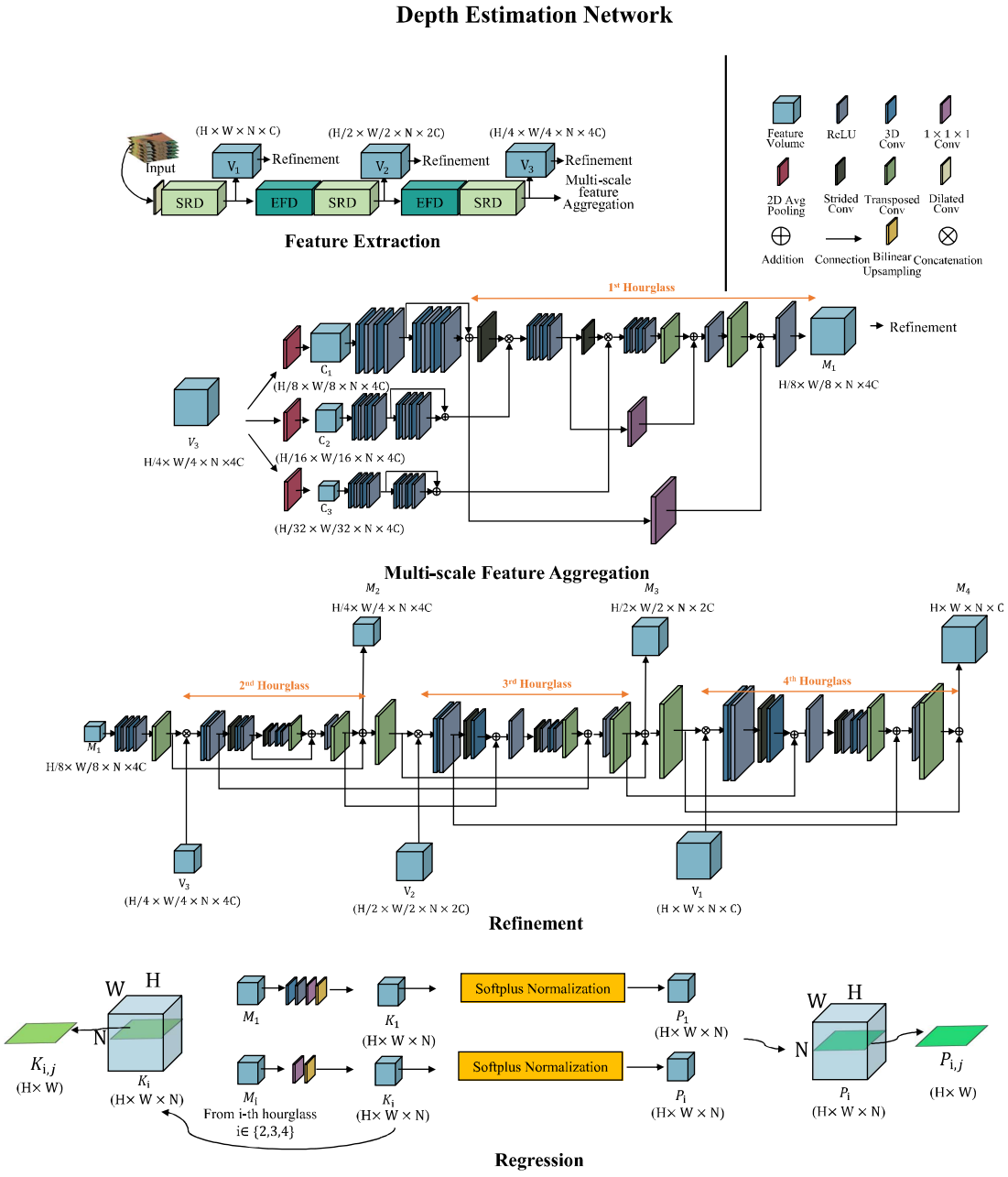
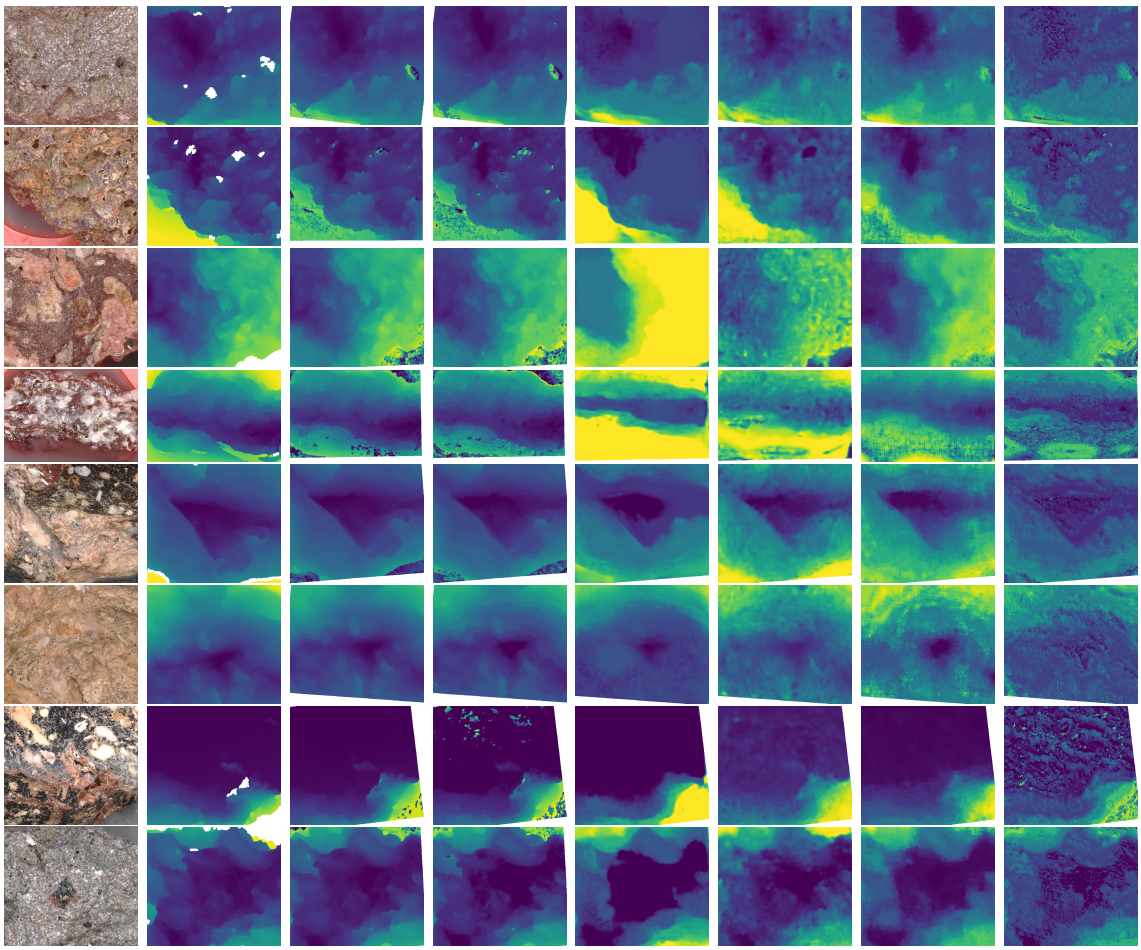


Abbildung 48: Netzwerkarchitektur von DDFFintheWild (Won und Jeon 2022)



(a) (b) (c) (d) (e) (f) (g) (h)

Abbildung 49: Vergleich der Tiefenkarten weiterer Proben des GeoFocus3D-Datensatzes mit den RGB-Bildern (a), den Ground-Truth-Tiefenkarten (b), und den mit dem DFF-WLLSR-Methode mit einer Schrittweite von 0.1 mm (c) und 0.5 mm (d), mit dem vortrainierten DFVNet (e), dem neu trainierten DFVNet (f), dem vortrainierten DDF-FintheWild (g) und dem neu trainierten DDFFintheWild (h) berechneten Tiefenkarten

Verwendung von KI-Hilfsmitteln

Zur sprachlichen Verbesserung sowie zur Unterstützung bei der LaTeX-Formatierung dieser Arbeit habe ich folgende KI-Tools verwendet:

- ChatGPT (Produkt von OpenAI, Version GPT-4)
- TeXGPT (Produkt von Writefull, Version GPT-3)

Die inhaltliche Erarbeitung sowie die wissenschaftliche Analyse und Argumentation erfolgten eigenständig durch mich.

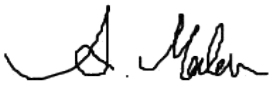
Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe. Alle Stellen, die den benutzten Quellen und Hilfsmitteln unverändert oder inhaltlich entnommen sind, habe ich als solche kenntlich gemacht.

Sofern generative KI-Tools verwendet wurden, habe ich Produktnamen, Hersteller, die jeweils verwendete Softwareversion und die Art der Nutzung (z.B. sprachliche Überprüfung und Verbesserung der Texte, systematische Recherche) benannt. Ich verantworte die Auswahl, die Übernahme und sämtliche Ergebnisse des von mir verwendeten KI-generierten Outputs vollumfänglich selbst.

Die Satzung zur Sicherung guter wissenschaftlicher Praxis an der TU Berlin vom 8. März 2017. https://www.static.tu.berlin/fileadmin/www/10000060/FSC/Promotion___Habilitation/Dokumente/Grundsaetze_gute_wissenschaftliche_Praxis_2017.pdf habe ich zur Kenntnis genommen.

Ich erkläre weiterhin, dass ich die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.



Berlin, 5. Mai 2025