



DGK Ausschuss Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 980

Julia Jasmin Aichinger

**B-Spline-Modellierung für TLS-Punktwolken
zur krümmungsbasierten Zielzeichen-freien
Deformationsanalyse**

München 2026

Verlag der Bayerischen Akademie der Wissenschaften, München

ISSN 0065-5325

ISBN 978-3-7696-5392-2



B-Spline-Modellierung für TLS-Punktwolken
zur krümmungsbasierten Zielzeichen-freien
Deformationsanalyse

Von der Fakultät für Luft- und Raumfahrttechnik und Geodäsie
der Universität Stuttgart zur Erlangung der Würde eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von

M.Sc. Julia Jasmin Aichinger

aus Stuttgart Bad-Cannstatt

München 2026

Verlag der Bayerischen Akademie der Wissenschaften, München

Adresse des Ausschusses Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften:



Ausschuss Geodäsie (DGK) der Bayerischen Akademie der Wissenschaften

Alfons-Goppel-Straße 11 • D – 80 539 München

Telefon +49 – 89 – 23 031 1113 • Telefax +49 – 89 – 23 031 - 1283 / - 1100

e-mail post@dgk.badw.de • <http://www.dgk.badw.de>

Hauptberichter: Prof. Dr.-Ing. habil. Dr. h.c. Volker Schwieger

Mitberichterin: Prof. Dr. Corinna Harmening

Tag der mündlichen Prüfung: 19. März 2026

Diese Dissertation ist auf dem Server des Ausschusses Geodäsie (DGK)
der Bayerischen Akademie der Wissenschaften unter <http://dgk.badw.de/>
sowie auf dem Publikationsserver der Universität Stuttgart unter
<https://elib.uni-stuttgart.de/> elektronisch publiziert

Inhaltsverzeichnis

Tabellen	V
Abbildungen	VII
Symbole	XIII
Kurzfassung	XV
Abstract	XVII
1 Einleitung	1
1.1 Methoden der TLS-basierten Deformationsanalyse	2
1.2 Zielsetzungen der Arbeit	4
1.3 Aufbau der Arbeit	5
2 Mathematische Grundlagen zu B-Splines	7
2.1 B-Spline-Basisfunktionen	8
2.1.1 Knotenvektoren	9
2.1.2 Eigenschaften von B-Spline-Basisfunktionen	13
2.2 B-Spline-Kurven	18
2.2.1 Periodische B-Spline-Kurven	18
2.2.2 Offene B-Spline-Kurven	20
2.2.3 Eigenschaften von B-Spline-Kurven	22
2.3 B-Spline-Flächen	29
2.3.1 Eigenschaften von B-Spline-Flächen	31
2.4 Interpolation und Approximation von Punktmengen mittels B-Splines	35
2.5 Zusammenfassung des Kapitels	37
3 Forschungsstand TLS bzgl. B-Spline Modellierung, Registrierung und Deformationsanalyse	39
3.1 Modellierung von TLS-Daten mit B-Splines	39
3.2 Registrierungsverfahren	41
3.2.1 Geometriebasierte Registrierungsverfahren	42
3.2.2 Merkmalsbasierte Registrierungsverfahren	42
3.3 Deformationsanalyse mittels B-Splines	43
4 Direkte B-Spline-Kontrollpunkt-Methode (DBKM)	45
4.1 Struktur der Methode	45
4.1.1 Vorbereitende Maßnahmen und Rahmenbedingungen	45
4.1.2 Prozessablauf	46
4.2 Modellierung von TLS-Daten mittels Tensorprodukt B-Spline-Flächen	48
4.2.1 Definition von Kontrollpolygonen	49
4.2.2 Knoteneinfügealgorithmen	53
4.2.2.1 Algorithmus von Böhm	54
4.2.2.2 Oslo-Algorithmus	59
4.2.2.3 Gegenüberstellung von Knoteneinfügealgorithmen	62
4.3 Mapcut-Algorithmus (Zielzeichen-freie Registrierung)	64
4.3.1 Mapcut-Algorithmus (B-Spline-Kurven)	64
4.3.2 Mapcut-Algorithmus (B-Spline-Flächen)	74

4.4 Zusammenfassung des Kapitels	81
5 Stochastische Modellierung	83
5.1 Varianz-Kovarianzfortpflanzung (B-Spline-Kurven)	83
5.2 Varianz-Kovarianzfortpflanzung (B-Spline-Flächen)	87
5.3 Simulation Varianz-Kovarianzfortpflanzung (B-Spline-Flächen)	89
5.4 Signifikanztest	93
5.5 Zusammenfassung des Kapitels	94
6 Evaluierungsdatensätze	95
7 Ergebnisanalyse	101
7.1 Anwendung der DBKM	101
7.1.1 Fassadenstück (2D-Punktwolke)	101
7.1.2 Gurkenblatt (3D-Punktwolke)	106
7.1.3 Tontorso (3D-Punktwolke)	112
7.1.4 Sandobjekt (3D-Punktwolke)	118
7.1.5 Diskussion der Ergebnisse	123
7.2 Anwendung der Deformationsanalyse	125
7.2.1 Fassadenstück (2D-Punktwolke)	125
7.2.2 Gurkenblatt (3D-Punktwolke)	129
7.2.3 Tontorso (3D-Punktwolke)	132
7.2.4 Sandobjekt (3D-Punktwolke)	135
7.2.5 Diskussion der Ergebnisse	137
8 Zusammenfassung und Ausblick	139
8.1 Zusammenfassung	139
8.2 Ausblick	141
Literatur	XIX
Abkürzungsverzeichnis	XXVII
A Anhang	XXIX
A.1 Algorithmus von Böhm - Stochastischer und numerischer Nachweis	XXIX
A.2 Varianz-Kovarianzfortpflanzung für B-Spline-Kurvenpunkte	XLIII
A.3 Varianz-Kovarianzfortpflanzung für B-Spline-Flächenpunkte	XLIV
A.4 Beispielrechnung zur Kreuzkovarianzfunktion (2D)	XLVI
Danksagung	XLVII
Lebenslauf	XLIX

Tabellen

2.1	Parameter der B-Spline-Fläche von Abbildung 2.28	30
4.1	Gegenüberstellung der Rechenzeiten vom Algorithmus von Böhm, Oslo-Algorithmus und Oslo-Algorithmus mit geänderter Implementierung.	63
4.2	Aufstellung der Verschiebungsparameter-Grenzen zur Bestimmung der Formel für die Kreuzkovarianzfunktion des Mapcut-Algorithmus für B-Spline-Flächen.	76
5.1	Vergleich des Einflusses der Schrittweite auf die Standardabweichungen von B-Spline-Kurvenpunkten für einen Teilkurvenbereich. Grafische Darstellung der Standardabweichungen in Abbildung 5.2.	86
5.2	Einfluss Schrittweite auf Standardabweichungen von B-Spline-Flächenpunkten . .	93
7.1	Maximale Deformation von Epoche 1 ohne Messrauschen (2D-Fassadenstück). . .	126
7.2	Maximale Deformation von Epoche 1 ohne Messrauschen betreffend x-, y-, z-Koordinate und 3D-Deformation (Gurkenblatt, Tontorso, Sandobjekt).	129
A.1	Differenz der X-Koordinaten der mittels Algorithmus von Böhm generierten B-Spline-Kurven in Relation zur ursprünglichen B-Spline-Kurve.	XL
A.2	Differenz der Y-Koordinaten der mittels Algorithmus von Böhm generierten B-Spline-Kurven in Relation zur ursprünglichen B-Spline-Kurve.	XL

Abbildungen

1.1	Struktur der Arbeit. Basiskapitel (grau), Kapitel der entwickelten Methode (magenta), Unterkapitel (cyan).	6
2.1	Kontrollpolygon bestehend aus den Kontrollpunkten $\mathbf{d}_0$ bis $\mathbf{d}_6$, links: Interpolierendes Polynom vom Grad 6, rechts: Bézier-Kurve vom Grad 6 mit konvexer Hülle. . . .	7
2.2	Periodische B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts). . .	10
2.3	Periodische B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts). . .	10
2.4	Nicht periodische uniforme B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts).	11
2.5	Nicht periodische uniforme B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts) (Aichinger und Schwieger 2022).	11
2.6	Nicht uniforme offene B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts).	12
2.7	Nicht uniforme offene B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts).	13
2.8	Offene B-Spline-Basisfunktionen vom Grad 2	14
2.9	Periodische B-Spline-Basisfunktionen vom Grad 1 (links) und Grad 2 (rechts). . .	15
2.10	Nicht uniforme B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts) mit einer Knotenvielfachheit von bis zu 2.	16
2.11	Nicht uniforme B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts) mit einer Knotenvielfachheit von bis zu 2.	16
2.12	Nicht uniforme B-Spline-Basisfunktionen vom Grad 4.	17
2.13	Erste Ableitung nicht uniformer B-Spline-Basisfunktionen vom Grad 4.	17
2.14	Periodische uniforme B-Spline-Kurven vom Grad 2 (links) und Grad 3 (rechts). . .	18
2.15	Geschlossene periodische uniforme B-Spline-Kurven vom Grad 3.	19
2.16	Offene uniforme B-Spline-Kurven vom Grad 2 (links) und Grad 3 (rechts) (Aichinger und Schwieger 2022).	20
2.17	Auswirkung der Veränderung eines Knotens eines Knotenvektors auf eine offene uniforme B-Spline-Kurve vom Grad 3.	21
2.18	Uniforme B-Spline-Kurven vom Grad 0 bis 5.	21
2.19	Auswirkung einer Kontrollpunktänderung auf die Kurvenform einer B-Spline-Kurve vom Grad 2 (oben) und Grad 3 (unten).	22
2.20	Auswirkung einer Kontrollpunktänderung auf die Kurvenform einer B-Spline-Kurve vom Grad 4 (oben) und Grad 5 (unten).	23
2.21	Nicht uniforme B-Spline-Kurven vom Grad 3 mit Vielfachheit 1 (links) und Vielfachheit 2 (rechts).	24
2.22	Nicht uniforme B-Spline-Kurven vom Grad 3 mit Vielfachheit 3 (links) und Vielfachheit 4 (rechts).	24
2.23	B-Spline-Basisfunktionen vom Grad 2.	25
2.24	Konvexe Hülle von B-Spline-Kurven vom Grad 1 (oben) und Grad 2 (unten). . .	26
2.25	Konvexe Hülle von B-Spline-Kurven vom Grad 3 (oben), Grad 4 (mitte) und Grad 5 (unten).	27
2.26	Erste und zweite Ableitung einer B-Spline Kurve dritten Grades in Form der Tangenten- und Krümmungsvektoren für zwölf Kurvenpunkte der B-Spline-Kurve.	28
2.27	Krümmung einer B-Spline Kurve vom Grad 3 (Aichinger und Schwieger 2022). . .	29
2.28	Kontrollgitter und B-Spline-Fläche vom Grad (3,3) (Aichinger und Schwieger 2022). .	31
2.29	Ebene B-Spline-Fläche vom Grad (3,3) mit zwei extrapolierten Kontrollpunkten. .	32

2.30	Einflussbereich der Kontrollpunktänderung auf eine ebene B-Spline-Fläche vom Grad (3,3).	32
2.31	Normalenvektor einer B-Spline-Fläche (Aichinger und Schwieger 2022).	35
4.1	Ablaufdiagramm der Direkten B-Spline-Kontrollpunkt-Methode (DBKM).	47
4.2	Vereinfachte Darstellung des Modellierungsprozesses von Tensorprodukt B-Spline-Flächen aus TLS-Daten, a) TLS-Daten, b) verbundene TLS-Daten entsprechend Scanreihenfolge, c) Definition der Kontrollpolygone, d) resultierende Kontrollpolygone, e) Neue Kontrollpunkte durch Knoteneinfügealgorithmus, f) Bidirektionale Kontrollgitterstruktur.	48
4.3	Punktwolke eines Fassadenstücks aus zwei Ansichten. Punkte verbunden entsprechend Matrixposition.	50
4.4	Definierte Kontrollpolygone aus zwei Ansichten.	50
4.5	Histogramm der definierten Kontrollpolygone eines Fassadenstücks.	51
4.6	Definierte Kontrollpolygone eines Laserscans einer Platte (Aichinger und Schwieger 2022).	51
4.7	Histogramm der definierten Kontrollpolygone (Platte).	52
4.8	Definierte Kontrollpolygone eines Gurkenblatt-Laserscans (Aichinger und Schwieger 2022).	52
4.9	Histogramm der definierten Kontrollpolygone (Gurkenblatt).	53
4.10	B-Spline-Kurve dritten Grades durch Anwendung des Algorithmus von Böhm.	55
4.11	Kontrollpolygone des Testdatensatzes. Magenta: Selektierte Kontrollpolygone, hellblau: nicht selektierte Kontrollpolygone.	56
4.12	Kontrollgitter vor Anwendung des Algorithmus von Böhm (Aichinger und Schwieger 2022).	57
4.13	Erzeugung von Kontrollpunkten mittels Algorithmus von Böhm.	58
4.14	Kontrollgitter nach Anwendung des Algorithmus von Böhm (Aichinger und Schwieger 2022).	58
4.15	Tensorprodukt B-Spline-Fläche dritten Grades und Kontrollgitter.	59
4.16	B-Spline-Kurve dritten Grades durch Anwendung des Oslo-Algorithmus.	61
4.17	Punktwolken eines Fassadenstücks (oben links), einer Platte (oben rechts), eines Gurkenblatts (unten links) und einer Ebene (unten rechts).	62
4.18	Ablauf des Mapcut-Algorithmus für B-Spline-Kurven im Kontext der DBKM.	65
4.19	Exemplarische Darstellung des Vorgangs der Kreuzkovarianzfunktion zweier Krümmungssignale gleicher Länge.	66
4.20	Aufteilung des Vorgangs der Kreuzkovarianzfunktion zweier Krümmungssignale unterschiedlicher Länge in drei Prozessschritte.	68
4.21	Ablauf von Stufe I und II des Mapcut-Algorithmus für B-Spline-Kurven (Grobmapping und Feinmapping).	69
4.22	Ablauf von Stufe III des Mapcut-Algorithmus für B-Spline-Kurven (Transformation).	70
4.23	B-Spline-Kurven eines Beispieldatensatzes von Epoche 0 und Epoche 1. Annähernd identische Punkte in 10 Kurventeilen und selektierte identische Punkte in 60 % der Kurventeile (Iteration 1).	71
4.24	B-Spline-Kurven eines Beispieldatensatzes von Epoche 0 und Epoche 1. Annähernd identische Punkte in 15 Kurventeilen und selektierte identische Punkte in 60 % der Kurventeile (Letzte Iteration).	72
4.25	Obere Grafik: Zwei Kontrollpolygone unterschiedlicher Punktzahl und deren B-Spline-Kurven. Untere Grafik: Punktdifferenzen zugehöriger B-Spline-Kurvenpunkte nach Stufe III des Mapcut-Prozesses.	73

4.26	Obere Grafik: Zwei Kontrollpolygone gleicher Punktzahl und deren B-Spline-Kurven. Untere Grafik: Punktdifferenzen zugehöriger B-Spline-Kurvenpunkte nach Stufe IV des Mapcut-Prozesses (Cutting).	74
4.27	Ablauf des Mapcut-Algorithmus für B-Spline-Flächen im Kontext der DBKM.	75
4.28	Schrittweise Verschiebung der Krümmungsgitter zweier Epochen im Grobmapping.	77
4.29	Grobmapping und Feinmapping für einen Beispieldatensatz (Iteration 1).	77
4.30	Grobmapping und Feinmapping für einen Beispieldatensatz (Letzte Iteration).	78
4.31	Ablauf von Stufe III des Mapcut-Algorithmus für B-Spline-Flächen (Transformation).	79
4.32	Selektion von annähernd identischen Punkten für einen Beispieldatensatz (links: erste Iteration, rechts: letzte Iteration)	79
4.33	Koordinatendifferenzen von Epoche 0 und Epoche 1 für einen Beispieldatensatz (links: erste Iteration, rechts: letzte Iteration).	80
5.1	Ablaufdiagramm der statistischen Auswertung der DBKM (B-Spline-Kurven).	83
5.2	Standardabweichung der Kurvenpunkte im Bereich des Anfangskurvenpunktes je nach Schrittweite. Numerische Darstellung der Werte der Standardabweichungen in Tabelle 5.1.	86
5.3	Ablaufdiagramm der statistischen Auswertung der DBKM (B-Spline-Flächen).	87
5.4	links: 3D-Punktwolke bestehend 500 Laserscanpunkten (Beispieldatensatz), rechts: Kontrollgitter.	90
5.5	Standardabweichungen der B-Spline-Flächenpunkte (Beispieldatensatz).	90
5.6	Schema zur Bildung von simulierten Standardabweichungen für B-Spline-Flächen	91
5.7	Schematische Darstellung der B-Spline-Flächengenerierung mit Bezugnahme auf die Eckpunkt-Eigenschaft.	91
5.8	Werte für simulierte Standardabweichungen von B-Spline-Flächen (Vergrößerter Ausschnitt).	92
6.1	2D-Punktwolke eines Fassadenstücks von Epoche 0 (links) und Epoche 1 (rechts).	95
6.2	links: Gurkenblatt-Testobjekt nach Harmening (2020), rechts: Laserscan-Punktwolke des Gurkenblatt-Testobjekts.	96
6.3	3D-Punktwolken des Gurkenblatt-Testobjekts von Epoche 0 (links) und Epoche 1 (rechts).	96
6.4	Nike von Samothrake nach Nguyen (2007) (magenta: Modellierter Ausschnitt für Testobjekt aus Ton).	97
6.5	Herstellungsprozess und fertig gestelltes Testobjekt aus Ton (Tontorso).	97
6.6	3D-Punktwolken des Tontorso-Testobjekts von Epoche 0 (links) und Epoche 1 (rechts).	98
6.7	Testobjekt aus Sand.	98
6.8	3D-Punktwolken des Sandobjekts von Epoche 0 (links) und Epoche 1 (rechts).	99
6.9	Instrumentenaufbau mit dem Tontorso-Testobjekt und Sand-Testobjekt.	99
7.1	2D-Kontrollpolygone des Fassadenstücks von Epoche 0 (links) und Epoche 1 (rechts).	101
7.2	Koordinatendifferenzen der Kontrollpunkte des Fassadenstücks von Epoche 0 und Epoche 1.	102
7.3	B-Spline-Kurven des Fassadenstücks von Epoche 0 (links) und Epoche 1 (rechts).	102
7.4	links: Korrelationskoeffizienten im Grobmapping, rechts: Korrelationskoeffizienten im Feinmapping (2D-Fassadenstück, Iteration 1).	103
7.5	links: Identische Punkte transformiert in Epoche 0, rechts: Euklidische Distanz der identische Punkte zwischen Epoche 0 und Epoche 1 (2D-Fassadenstück, Iteration 1).	103
7.6	Koordinatendifferenzen nach Stufe III des Mapcut-Algorithmus (2D-Fassadenstück, Iteration 1).	104

7.7	B-Spline-Kurven von Epoche 0 (links) und Epoche 1 (rechts) nach Durchführung der ersten Iteration des Mapcut-Algorithmus (2D-Fassadenstück, Iteration 2).	104
7.8	links: Korrelationskoeffizienten im Grobmapping, rechts: Korrelationskoeffizienten im Feinmapping (2D-Fassadenstück, Iteration 2).	105
7.9	links: Identische Punkte transformiert in Epoche 0, rechts: Euklidische Distanz der identische Punkte zwischen Epoche 0 und Epoche 1 (2D-Fassadenstück, Iteration 2).	105
7.10	Koordinatendifferenzen nach Stufe III des Mapcut-Algorithmus (2D-Fassadenstück, Iteration 2).	106
7.11	3D-Punktwolken des Gurkenblatts von Epoche 0 (links) und Epoche 1 (rechts).	106
7.12	Histogramme der definierten Kontrollpolygone des Gurkenblatts von Epoche 0 (links) und Epoche 1 (rechts).	107
7.13	Ausschnitt der Kontrollgitter von Gurkenblatt Epoche 0 (links) und Epoche 1 (rechts).	107
7.14	Ausschnitt der Kontrollgitter von Gurkenblatt Epoche 0 (links) und Epoche 1 (rechts).	108
7.15	B-Spline-Fläche von Gurkenblatt Epoche 0 (links) und Epoche 1 (rechts) (Rasterdarstellung).	108
7.16	Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 1).	108
7.17	Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 1).	109
7.18	links: Vorläufige annähernd identische Punkte, rechts: Koordinatendifferenzen von Epoche 0 und Epoche 1 (Gurkenblatt, Iteration 1).	109
7.19	Übereinandergelagerte B-Spline-Flächen von Epoche 0 und Epoche 1 (Gurkenblatt, Iteration 1).	110
7.20	Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 2).	111
7.21	Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 2).	111
7.22	links: Identische Punkte, rechts: Koordinatendifferenzen von Epoche 0 und Epoche 1 (Gurkenblatt, Iteration 2).	111
7.23	3D-Punktwolken von Tontorso von Epoche 0 (links) und Epoche 1 (rechts).	112
7.24	Histogramme der definierten Kontrollpolygone des Tontorsos von Epoche 0 und Epoche 1.	112
7.25	Ausschnitt der Kontrollgitter des Tontorso von Epoche 0 (links) und Epoche 1 (rechts).	113
7.26	B-Spline-Flächen des Tontorsos von Epoche 0 (links) und Epoche 1 (rechts) (Rasterdarstellung).	113
7.27	Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 1).	114
7.28	Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 1).	114
7.29	links: Vorläufige annähernd identische Punkte, rechts: Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Tontorso, Iteration 1).	115
7.30	Übereinandergelagerte B-Spline-Flächen von Epoche 0 und Epoche 1 (Tontorso, Iteration 1).	115
7.31	Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 3).	116
7.32	Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 3).	116
7.33	Identische Punkte (Tontorso, Iteration 3).	116
7.34	Koordinatendifferenzen von Epoche 0 und Epoche 1 (Tontorso, Iteration 3).	117

7.35	3D-Punktwolken des Sandobjekt von Epoche 0 (links) und Epoche 1 (rechts). . .	118
7.36	Histogramme der definierten Kontrollpolygone von Sandobjekt Epoche 0 (links) und Epoche 1 (rechts).	118
7.37	Ausschnitt der Kontrollgitter des Sandobjekts von Epoche 0 (links) und Epoche 1 (rechts).	119
7.38	B-Spline-Flächen des Sandobjekts von Epoche 0 (links) und Epoche 1 (rechts) (Rasterdarstellung).	119
7.39	Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 1).	119
7.40	Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 1).	120
7.41	links: Vorläufige annähernd identische Punkte, rechts: Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 1).	120
7.42	Übereinandergelagerte B-Spline-Flächen von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 1).	121
7.43	Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 3).	121
7.44	Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 3).	121
7.45	Identische Punkte von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 3).	122
7.46	Koordinatendifferenzen von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 3).	122
7.47	Ausschnitt der 3D-Punktwolke des Sandobjekts im Bereich einer Erhebung.	123
7.48	Ausschnitt der 3D-Punktwolke des Tontorso im Bereich einer Vertiefung und von Kanten.	123
7.49	Kontrollpolygone von Epoche 0 und Epoche 1 mit Deformation (2D-Fassadenstück).	125
7.50	Koordinatendifferenzen der Kontrollpunkte von Epoche 0 und Epoche 1 (2D-Fassadenstück).	126
7.51	Deformationen von Epoche 1 in x-Koordinate, y-Koordinate und im Kontrollpunkt ohne Messrauschen (Referenz-2D-Deformation, 2D-Fassadenstück).	126
7.52	Ergebnis des Signifikanztests (2D-Fassadenstück).	127
7.53	2D-Deformationen der B-Spline-Kurvenpunkte in signifikant deformierten Bereichen (2D-Fassadenstück).	127
7.54	Koordinatendifferenzen der B-Spline-Kurvenpunkte von Epoche 0 und Epoche 1 in signifikant deformierten Bereichen (2D-Fassadenstück).	128
7.55	Ausschnitte der signifikant deformierten Koordinatendifferenzen zwischen Epoche 0 und Epoche 1 in x-Koordinate, y-Koordinate und im B-Spline-Kurvenpunkt (2D-Fassadenstück).	128
7.56	3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 ohne Messrauschen (Referenz-3D-Deformation, Gurkenblatt).	130
7.57	Ergebnis des Signifikanztests (Gurkenblatt).	130
7.58	Oben: 3D-Deformation der unverrauschten Messdaten als 2D-Darstellung. Graue äußere Konturlinie: Begrenzung der Referenz-3D-Deformation, innere Konturlinie: Begrenzung der durch die DBKM signifikant festgestellten 3D-Deformation. Unten: Längsschnitt durch 3D-Deformation. Graue Linie: Referenz, Magentafarbene Linie: signifikant festgestellte 3D-Deformation (Gurkenblatt).	131
7.59	Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Gurkenblatt). Links: B-Spline-Flächenpunkte, rechts: x-Koordinate.	131

7.60	Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Gurkenblatt). Links: y-Koordinate, rechts: z-Koordinate.	132
7.61	3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 ohne Messrauschen (Referenz-3D-Deformation, Tontorso).	132
7.62	Ergebnis des Signifikanztests (Tontorso).	133
7.63	Links: 3D-Deformation der unverrauschten Messdaten als 2D-Darstellung. Graue äußere Konturlinie: Begrenzung der Referenz-3D-Deformation, innere Konturlinie: Begrenzung der durch die DBKM signifikant festgestellten 3D-Deformation. Rechts: Längsschnitt durch 3D-Deformation. Graue Linie: Referenz, Magentafarbene Linie: signifikant festgestellte 3D-Deformation (Tontorso)	133
7.64	Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Tontorso). Links oben: B-Spline-Flächenpunkte, rechts oben: x-Koordinate, links unten: y-Koordinate, rechts unten: z-Koordinate.	134
7.65	3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Sandobjekt).	135
7.66	Ergebnis des Signifikanztests (Sandobjekt).	135
7.67	Oben: 3D-Deformation der unverrauschten Messdaten als 2D-Darstellung. Graue äußere Konturlinie: Begrenzung der Referenz-3D-Deformation, innere Konturlinie: Begrenzung der durch die DBKM signifikant festgestellten 3D-Deformation. Unten: Längsschnitt durch 3D-Deformation. Graue Linie: Referenz, Magentafarbene Linie: signifikant festgestellte 3D-Deformation (Sandobjekt).	136
7.68	Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Sandobjekt). Links oben: B-Spline-Flächenpunkte, rechts oben: x-Koordinate, links unten: y-Koordinate, rechts unten: z-Koordinate.	137
A.1	B-Spline-Kurve dritten Grades.	XXXIII
A.2	B-Spline-Kurve dritten Grades nach erster Anwendung des Algorithmus von Böhm.	XXXV
A.3	B-Spline-Kurve dritten Grades nach zweiter Anwendung des Algorithmus von Böhm.	XXXVI
A.4	B-Spline-Kurve dritten Grades nach dritter Anwendung des Algorithmus von Böhm.	XXXVII
A.5	Standardabweichungen der Kontrollpunkte bei einmaliger Ausführung des Algorithmus von Böhm.	XXXVIII
A.6	Standardabweichungen der Kontrollpunkte bei zweimaliger Ausführung des Algorithmus von Böhm.	XXXIX
A.7	Standardabweichungen der Kontrollpunkte bei dreimaliger Ausführung des Algorithmus von Böhm.	XXXIX
A.8	Kontrollpolygone.	XLI
A.9	Standardabweichungen der Kontrollpunkte eines Kontrollgitters nach Anwendung des Algorithmus von Böhm.	XLII
A.10	Befüllung der Korrelationsmatrix mit Werten ungleich 0 eines Kontrollgitters nach Anwendung des Algorithmus von Böhm.	XLII

Symbole

$a + 1$	Kontrollpunktanzahl in t_2 -Richtung
b, b_1, b_2	Ableitungskoeffizienten der B-Spline-Basisfunktion, B-Spline-Kurve bzw. B-Spline-Fläche
$c + 1$	Anzahl an Laserscanpunkten
c_{t_1, t_2}	Erste Fundamentalform
C	B-Spline-Kurve
$Corr_{\kappa_1, \kappa_2}$	Kreuzkorrelationsfunktion der Krümmungssignale κ_1 und κ_2
Cov_{κ_1, κ_2}	Kreuzkovarianzfunktion der Krümmungssignale κ_1 und κ_2
d_i	Kontrollpunkt bzw. de Boor Punkt
D	Matrix bestehend aus Kontrollpolygonen
e	Grad der B-Spline-Basisfunktion bzw. B-Spline-Fläche in t_2 -Richtung
E, F, G	Koeffizienten der ersten Fundamentalform
$f + 1$	Neue Knotenanzahl bezüglich des Oslo Algorithmus
F	Jacobi-Matrix
$g + 1$	Anzahl der neuen Kontrollpunkte bezüglich des Oslo Algorithmus
h_{t_1, t_2}	Zweite Fundamentalform
$k + 1$	Knotenanzahl der Knoten in U
K	2D-Krümmungsparameter für B-Spline-Flächen
l	Vielfachheit der inneren Knoten
L, M, P	Koeffizienten der zweiten Fundamentalform
$m + 1$	Kontrollpunktanzahl in t - bzw. t_1 -Richtung
n	Grad der B-Spline-Basisfunktion, B-Spline-Kurve bzw. B-Spline-Fläche in t_1 -Richtung
n_1, n_2, n_3, n_4	Anzahl der Krümmungswerte innerhalb des Mapcut-Algorithmus
N	B-Spline-Basisfunktion
NO	Normaleneinheitsvektor
o	Ordnung der B-Spline-Basisfunktion
P_β	Punktmenge im Zusammenhang mit einer Interpolations- oder Approximationsmethode
q_i, q_j	Neuer Kontrollpunkt bzgl. des Algorithmus von Böhm bzw. Oslo Algorithmus
$r + 1$	Knotenanzahl der Knoten in V
R	Kreuzkorrelationsfunktion
s	Knotennummerierung bezüglich des Algorithmus von Böhm
S	Tensorprodukt B-Spline-Fläche
t	Intervall zur B-Spline-Basisfunktion- und B-Spline-Kurvenbestimmung
t_1, t_2	Intervalle zur B-Spline-Flächenbestimmung
u_i	Knoten bzw. Träger in U
$\bar{u}$	Einzufügender Knoten

$\mathbf{U}, \mathbf{V}$	Knoten- bzw. Trägervektor
$\overline{\mathbf{U}}$	Einzufügender Knotenvektor
v_i	Knoten bzw. Träger in $\mathbf{V}$
z	Anzahl an selektierten identischen Punkten
α	Koeffizient innerhalb des Oslo Algorithmus
γ	Koeffizient innerhalb des Algorithmus von Böhm
Θ	Poldistanzwinkel
κ	Krümmung einer B-Spline-Kurve
κ_1, κ_2	Krümmungssignale
Σ	Varianz-Kovarianzmatrix
λ_1, λ_2	Eigenwerte von $\mathbf{h}_{t_1, t_2}$
Φ	Azimutwinkel

Matrizen werden konventionsgemäß groß und fett, Vektoren klein und fett notiert. An einzelnen Stellen wurde davon abgewichen, um die Konsistenz mit der verwendeten Fachliteratur zu wahren. Bei folgenden Symbolen wurde beispielsweise von der Konvention abgewichen: $\mathbf{c}_{t_1, t_2}$, $\mathbf{d}_i$, E, F, G , $\mathbf{h}_{t_1, t_2}$, L, M, P , $\mathbf{NO}$, $\mathbf{q}_i, \mathbf{q}_j$, $\mathbf{U}, \mathbf{V}$, $\overline{\mathbf{U}}$.

Kurzfassung

Die fortschreitende Entwicklung des terrestrischen Laserscannings (TLS) in Bezug auf Abtastrate und Genauigkeit ermöglicht eine hochauflösende räumliche Diskretisierung von Objekten und ein breites Anwendungsfeld. Insbesondere für die Erfassung komplexer Objektoberflächen und die Erkennung unerwarteter Verformungen bietet das TLS starke Potentiale, um flächenhafte geodätische Überwachungsmessungen anzuwenden. Herkömmliche punktweise Messverfahren sind ohne Vorkenntnisse möglicher Verformungen oder erheblichen Messaufwand nicht in der Lage, komplexe unerwartete Deformationsprozesse von Oberflächen zu erfassen. Während punktweise Messverfahren identische Punkte über mehrere Epochen eindeutig erfassen und vergleichbar machen, werden beim TLS jedoch nicht automatisch vergleichbare Punktkorrespondenzen erzeugt.

Eine Zuordnung korrespondierender 3D-Punkte bedingt beim TLS somit immer einer Nachbearbeitung. Infolge der überaus vorteilhaften Eigenschaften und der stetigen Weiterentwicklung finden TLS-basierte Ansätze für Deformationsanalysen in der Forschung einen breiten Zuspruch. Einige Herausforderungen in der Prozesskette der TLS-basierten Deformationsanalyse wurden allerdings noch nicht zufriedenstellend gelöst. Diese Arbeit stellt einen neuartigen Ansatz, die Direkte B-Spline-Kontrollpunkt-Methode (DBKM), zur Deformationsanalyse dreidimensionaler TLS-Daten vor, um die Lücke an Innovationen in diesem Fachgebiet weiter zu schließen. Hierbei werden die TLS-Daten durch eine direkte funktionale Approximation zu Tensorprodukt B-Spline-Flächen modelliert. In Relation zu gängigen Verfahren wird kein aufwändiges Schätzverfahren zur Parameterbestimmung angewendet, sondern ein Ansatz der direkten Verwendung der Laserscandaten als B-Spline-Kontrollpunkte verfolgt. Die B-Spline-Flächen zweier Epochen werden daraufhin in einen iterativen Algorithmus zur lokalen Zielzeichen-freien Registrierung eingespeist. Der entwickelte Mapcut-Algorithmus bestimmt in einem vierstufigen Verfahren mit Hilfe eines Krümmungsparameters als Merkmalsdeskriptor in einer Korrelationsanalyse Punktkorrespondenzen in den B-Spline-Flächen der verschiedenen Epochen. Bei Annahme von kleinen Deformationen im Verhältnis zur Objektoberfläche in den Punktwolken beherrscht der Mapcut-Algorithmus eine Selektion identischer Punkte in ausschließlich stabilen Bereichen, die für die Transformation in das Referenzkoordinatensystem verwendet werden. In Folge der direkten Einbeziehung der TLS-Daten als Kontrollpunkte der B-Spline-Flächen, ist das enthaltene Messrauschen für die Analyse von Deformationen zu beachten. Diese Rahmenbedingung wird durch eine vollständige stochastische Modellierung der DBKM berücksichtigt. In einem abschließenden Signifikanztest werden signifikant deformierte B-Spline-Flächenpunkte ermittelt und für deformierte Punkte die Euklidische Distanz und entsprechende Koordinatenrichtung angegeben. Die entwickelte DBKM wurde erfolgreich an simulierten Daten evaluiert. Eine Übertragung auf reale Daten war erfolgreich, bedarf jedoch für die Deformationsanalyse zusätzlicher Forschungsarbeit.

Abstract

The progressive development of terrestrial laser scanning (TLS) in terms of sampling rate and accuracy enables high-resolution spatial discretization of objects and a wide range of applications. Especially for the measurement of complex object surfaces and the analysis of unexpected deformations, TLS offers strong potentials to apply areal geodetic monitoring measurements. Conventional point-wise measurement methods are not able to analyse complex unexpected deformation processes of surfaces without prior knowledge of possible deformations or considerable measurement effort. While point-wise measurement methods clearly determine identical points over several epochs and thus make them comparable, TLS does not automatically generate comparable point correspondences. An assignment of corresponding 3D points therefore always requires post-processing with TLS.

As a result of the beneficial properties and continuous development, TLS-based approaches for deformation analyses are widely used in research. However, some challenges in the TLS-based deformation analysis process have not been solved satisfactorily yet. This thesis presents a novel approach, the Direct B-Spline Control Point Method (DBKM), for deformation analysis of three-dimensional TLS data to further close the gap of innovation in this discipline. Here, the TLS data is modeled by a direct functional approximation to tensor product B-spline surfaces. In contrast to conventional methods, no complex estimation procedure is used to determine the parameters, instead an approach of directly using the laser scan data as B-spline control points is applied. The B-spline surfaces of two epochs are subsequently inserted into an iterative algorithm for local targetless registration. The developed Mapcut algorithm determines point correspondences in the B-spline surfaces in a four-step process using a curvature parameter as a feature descriptor in a correlation analysis. Assuming small deformations in relation to the object surface in the point clouds, the Mapcut algorithm handles a selection of identical points in stable areas, which are used for the transformation into the reference coordinate system. As a result of the direct inclusion of the TLS data as control points of the B-spline surfaces, the contained measurement noise must be taken into account for the deformation analysis. This framework condition is taken into account by a complete stochastic modeling of the DBKM. In a final significance test, significantly deformed B-spline surface points are determined and the Euclidean distance and corresponding coordinate direction are specified for the deformed points. The developed DBKM was successfully evaluated on simulated data. A transfer to real data was successful, but requires additional research work for the deformation analysis.

1 Einleitung

Seit einigen Jahren erlebt die Technik des terrestrischen Laserscannings (TLS) eine stetige Qualitäts- und Quantitätsverbesserung hinsichtlich der erzeugten 3D-Punktwolken. Im Vergleich zu traditionellen Messtechniken, wie der Tachymetrie, mit punktweiser Vermessung, liefert TLS eine schnelle, dichte und hochauflösende dreidimensionale Punktwolke des Zielobjekts (Lovas et al. 2008; Abbas et al. 2014). Durch den fortlaufenden hard- und softwarebasierten Fortschritt in der Ingenieurgeodäsie (Kuhlmann et al. 2014) werden mittels TLS auch im Bereich der Deformationsanalyse Potentiale zur Entwicklung neuer Methoden eröffnet (Holst und Kuhlmann 2016). So werden Methoden der Deformationsanalyse mittels TLS-Daten bei z.B. Industrieschloten (Ötsch und Neuner 2023), Staudämmen (Wiedemann und Holst 2022; Zaczek-Peplinska und Kowalska 2017), Brücken (Rashidi et al. 2020), Holzkonstruktionen (Steinbach und Harmening 2025) oder anderen Ingenieurbauwerken (Schill 2020) eingesetzt.

Grundsätzlich können vier Modelle der Deformationsanalyse nach Heunecke et al. (2013) definiert werden. Hierzu zählt das Dynamische Modell, das einen funktionalen zeitabhängigen Zusammenhang zwischen Einflusskräften und Deformationen herstellt. Es bildet ein reales Objektverhalten am besten ab, fällt allerdings äußerst komplex aus, da sämtliche äußere Einflüsse berücksichtigt werden müssen. Anwendungen sind z. B. Hangrutschungen unter Wind, Temperatur, Regenfällen und Tektonik. Das Statische Modell bildet einen funktionalen Zusammenhang zwischen Einflusskräften und Deformation ab, ohne eine zeitabhängige Betrachtung. Das Messobjekt sollte während einer Epoche unbewegt sein. Anwendungsbereiche sind z. B. ein Staudamm unter Einfluss des Wasserstands und der Temperatur oder ein Belastungsversuch auf einer Brücke. Kinematische Modelle sollen ein zeitabhängiges Verhalten von Deformationen modellieren. Eine Anwendung findet häufig statt, wenn die Ursachen unbekannt oder die Modellierung der Einflusskräfte nicht möglich oder zu aufwendig ist. Beispielanwendungen sind Setzungen oder Plattentektoniken. Das vierte und bekannteste Modell der Deformationsanalyse, das ebenfalls in dieser Arbeit verwendet wird, ist das Kongruenzmodell. Hierbei besteht der Fokus allein auf der Geometrie. Weder werden zeitabhängige Faktoren noch Einflusskräfte modelliert, da die Zielsetzung allein auf dem geometrischen Vergleich zweier Epochen liegt. Eine wesentliche Annahme des Kongruenzmodells ist, dass ein Teil des beobachteten Bereichs des Messobjekts geometrisch stabil bleibt (Neumann und Kutterer 2007).

Grundlage einer Deformationsanalyse bilden mindestens zwei Messungen des Zielobjekts, wobei die ausschließliche Untersuchung der Rohdaten der Punktwolken in den meisten Fällen keine verwendbaren Aussagen gestattet, sondern eine Nachverarbeitung verlangt (Walton et al. 2014). Für die Auseinandersetzung von TLS-Punktwolken innerhalb einer Deformationsanalyse beschreibt Mukupa et al. (2016) drei Methoden:

- Punkt-zu-Punkt-basierte Deformationsanalyse,
- Punkt-zu-Fläche-basierte Deformationsanalyse,
- Fläche-zu-Fläche-basierte Deformationsanalyse.

Ebenso kann das Ergebnis der Deformationsanalyse nach Lague et al. (2013) und Wujanz (2016) in drei mögliche Produkte kategorisiert werden:

- Verschiebungsvektorfeld auf Basis korrespondierender Punkte oder Flächen,
- Distanzmessungen zwischen zwei Epochen, die häufig als farbiger Plot dargestellt werden,
- Statistisch signifikante Differenzen von geometrischen Parametern.

1.1 Methoden der TLS-basierten Deformationsanalyse

Die Punkt-zu-Punkt-basierte Deformationsanalyse kann als Herleitung der klassischen punktweisen Messverfahren zur Deformationsanalyse mittels Tachymeter oder GNSS angesehen werden (Gordon et al. 2003). Allerdings liegen in Relation zu den klassischen Messverfahren beim TLS keine signalisierten Einzelpunkte in zwei Messepochen vor, die unmittelbar verglichen werden können. Die Punktverteilung auf dem zu scannenden Objekt ist zwar durch die Vorgabe eines Winkelrasters reproduzierbar, jedoch gilt dies nicht für die einzelnen Punktpositionen der Punktwolke (Holst et al. 2017). Für die Durchführbarkeit dieser Methode müssen daher korrespondierende Punkte entweder durch die Gestaltung der Studie an sich vorhanden sein oder nachträglich erzeugt werden. Mittels Festlegung korrespondierender Punkte kann die Euklidische Distanz als Ergebnis der Deformationsanalyse berechnet werden. In Little (2006) wird der Messaufbau derart festgelegt, dass ausschließlich Messpunkte in einem Raster von 5 m x 5 m zur Hangüberwachung erfasst und unmittelbar punktweise verglichen werden können. Tsakiri et al. (2006) führt eine punktweise TLS-basierte Deformationsanalyse durch, bei der ausschließlich das Zentrum von angebrachten Reflexzielmarken verglichen wird.

Der Punkt-zu-Punkt-basierten Methode können, abgesehen vom Vergleich einzelner korrespondierender Punkte, auch Techniken zum Vergleich zweier vollständiger Punktwolken zugeordnet werden. Für gewöhnlich wird hier vom Cloud-to-Cloud Ansatz gesprochen. Eine tiefere Untergliederung in ein Punkt-basiertes und ein Punktwolken-basiertes Modell ist ebenso möglich und wird von Ohlmann-Lauber und Schäfer (2011) definiert. Der Vergleich zweier Punktwolken kann durch Algorithmen mit Koordinatentransformationen, wie z.B. dem Iterative-Closest-Point-Algorithmus (ICP) realisiert werden (Neuner et al. 2016). Gojcic et al. (2021) wendet den ICP innerhalb einer Deformationsanalyse an, um mittels lokaler Merkmalsdeskriptoren 3D-Verschiebungsvektoren zur Überwachung von Erdrutschgebieten zu schätzen. Ebenfalls über einen merkmalsbasierten Ansatz weisen Wagner et al. (2017) mittels RGB-Informationen korrespondierende Punkte zu und können auf dieser Basis 3D-Verschiebungsvektoren bestimmen. Girardeau-Montaut et al. (2005) beschreibt eine baumartige Datenstruktur (Octree Struktur), bei der die Punktwolke in gleichmäßige Kacheln unterteilt und auf diese Weise verschiedene Methoden für Punktwolken-Vergleiche ermöglicht werden.

Bei der Punkt-zu-Fläche-basierten Deformationsanalyse wird aus einer Punktwolke eine Fläche, ein Modell oder Gitter erzeugt und die Punkte der zweiten Punktwolke mittels Bestimmung der Euklidischen Distanz in Relation gesetzt. Diese Methoden werden oftmals auch unter Cloud-to-Mesh (C2M) gehandelt. Eine der ersten Methoden der Punkt-zu-Fläche-basierten Deformationsanalyse wird von Cignoni et al. (1998) mit der sogenannten METRO-Methode (measuring error on simplified surfaces) etabliert. Diese generiert ein Dreiecksgitter und berechnet den minimalen Abstand eines Punktes zur Kante eines Dreiecks. Die Distanzen zum Referenzgitter werden als visuelles Ergebnis in einer Karte dargestellt. Ebenfalls mithilfe von Dreiecksvermaschung bestimmt Schäfer et al. (2004) für die Punktwolken eines Wasserkraftwerks eindimensionale Differenzen. Aspert et al. (2002) definiert einen Algorithmus zur Berechnung von Abständen zwischen Punktwolken auf Basis eines Dreiecksgitters unter Verwendung der Hausdorff-Distanz (Hausdorff 1935). Serantoni und Wieser (2016) erzeugen eine Dreiecksvermaschung aus einer ausgedünnten Punktwolke als Referenzfläche, auf die sich die berechneten Abweichungen von vier weiteren Punktwolken einer Schneestruktur in Form eines Iglus beziehen. Truong-Hong und Lindenbergh (2019) stellen als einen von drei Ansätzen eine Punkt-zu-Fläche Methode vor, in der die vertikalen Abweichungen einer Brücke in Bezug zu einer Referenzfläche gesetzt werden. Xiao et al. (2022) ermittelt die minimale Distanz von Punkten von sechs Epochen zu den Dreiecksseiten eines triangulierten Netzes zur Deformationsanalyse eines Steinschüttdamms während des Bauprozesses.

Die dritte Methode nach Mukupa et al. (2016) beschreibt die Fläche-zu-Fläche-basierte Deformationsanalyse. Diese vereint Methoden, die in Ohlmann-Lauber und Schäfer (2011) und Wunderlich et al. (2016) in drei Ansätze aufgespalten werden. Der erste Ansatz besteht aus einem flächenbasierten Ansatz, der auch als Mesh-to-Mesh (M2M) bekannt ist. Hierbei wird aus den Punktwolken oftmals durch

Dreiecksvermaschung eine Fläche erzeugt, die entweder direkt oder durch Interpolation aufgestellt wird. Der zweite Ansatz ist ein geometriebasierter Ansatz, bei dem aus den Punktwolken approximierende geometrische Primitive (z.B. Ebenen, Zylinder, Kegel) oder Freiformflächen (z.B. B-Splines) erzeugt werden. Der dritte Ansatz ist ein parameterbasierter Ansatz, der aus den Punktwolken Parameter extrahiert, die zum Vergleich der Messepochen herangezogen werden. Die Übergänge zwischen diesen Ansätzen sind allerdings fließend und oftmals ist eine strikte Abgrenzung nicht möglich. Eling (2009) erstellt im Rahmen einer Bauwerksüberwachung aus den Punktwolken ein Blockgittermodell. Für jeden Block wird ein repräsentativer Punkt je Messepoche abgeleitet, der zusammen mit einer mitgeführten Varianz-Kovarianzmatrix in einer Deformationsanalyse verglichen werden kann. Holst et al. (2012) parametrisieren einen Reflektor eines 100 m Radioteleskops als Rotationsellipsoid, um Verformungen auf Basis der Brennweite feststellen zu können. Lindenbergh und Pfeifer (2005) vergleichen TLS-Daten eines Schleusentors, in dem Punktwolkensegmente auf ein regelmäßiges ebenes Gitter interpoliert und für jedes ebene Modell Parameter geschätzt werden. Mittels Einbeziehung einer Varianz-Kovarianzmatrix in einem Signifikanztest können Deformationen analysiert werden. Lague et al. (2013) etabliert die sogenannte "Multiscale Model to Model Cloud Comparison" (M3C2) Methode. Dabei wird für jeden Referenzpunkt ein großer Zylinder gebildet, um einen Normalenvektor zu schätzen, der die lokale Oberflächenorientierung repräsentiert. Entlang dieses Vektors wird ein kleinerer Zylinder als Suchbereich in der zweiten Punktwolke projiziert. Der durchschnittliche Abstand der Punkte im kleinen Zylinder zum Referenzpunkt wird als Deformation quantifiziert. Durch Schätzung eines Konfidenzintervalls für jede Distanz können reale Verformungen von Messunsicherheiten unterschieden werden. Der inzwischen in Standardsoftware implementierte M3C2 Algorithmus wird von Paffenholz et al. (2017) zur Ableitung von Deformationen von Infrastrukturbauwerken am Beispiel einer historischen Gewölbebrücke genutzt. Yang (2024) entwickelt eine neuartige Variante des M3C2 mit der Deformationen mit geringerer Größe festgestellt werden können. Gordon et al. (2004) schätzt in einem Experiment mit der Belastung eines Holzbalkens 11 Parameter einer Polynomfunktion zur Bestimmung einer vertikalen Durchbiegung. Auf Basis dieser Parameter können für verschiedene Epochen z-Koordinaten berechnet werden, die unmittelbar einen Vergleich der Epochen gewährleisten. Harmening et al. (2021) modellieren epochenweise erfasste TLS-Punktwolken mittels approximierender B-Spline-Flächen. Auf dieser Grundlage werden korrespondierende Punkte definiert, die eine Schätzung von Starrkörperbewegungen ermöglichen sollen.

In der Fläche-zu-Fläche-basierten Deformationsanalyse wird jede Punktwolke modelliert. Eine Modellierung der Punktwolken zu einer Fläche kann für die Erkennung von geringfügigen Formveränderungen wirksam sein, da eine modellierte Fläche das Objekt genauer repräsentieren kann, als eine unmodellerte Beobachtung (Gordon et al. 2004). In einem nächsten Prozessschritt sind in allen Messepochen identische Punkte zu erzeugen oder die geschätzten Parameter statistisch zu vergleichen. Auf diese Weise können zwei wesentliche Herausforderungen der TLS-basierten Deformationsanalyse gelöst werden: Zum einen, dass eine vorhandene Deformation so gering ausfällt, dass sie in den Rohdaten nicht signifikant vom Messrauschen unterschieden werden kann und zum anderen, dass korrespondierende Punkte zwischen zwei Punktwolken in der Regel nicht vorliegen. Eine Behebung dieser Limitierung von TLS-Daten hat somit über eine adäquate Modellierung der Rohdaten und eine anschließende Zuweisung von Punktkorrespondenzen zu erfolgen.

Bei den bisher genannten Methoden betreffend Punkt-zu-Punkt, Punkt-zu-Fläche oder Fläche-zu-Fläche-basierten Deformationsanalyse werden überwiegend Abstände zwischen korrespondierenden Elementen angegeben. Eine Bestimmung der Koordinatenrichtungen der festgestellten Deformation findet nicht statt. Auf diese Weise kann keine Aussage darüber getätigt werden in welche Richtung sich ein Punkt durch die Deformation bewegt hat, da nur die Änderung des Punktabstands ausgegeben wird. Lediglich Gojcic et al. (2021) und Yang (2024) ermitteln 3D-Verschiebungsvektoren mittels lokaler Merkmalsdeskriptoren, wobei sich der Ansatz von Gojcic et al. (2021) auf eine Punkt-zu-Punkt-basierte Deformationsanalyse beschränkt. Im Bereich der Fläche-zu-Fläche-basierten Deformationsanalyse

besteht daher ein Entwicklungspotential Deformationen umfangreicher und präziser definieren zu können, indem zusätzlich zum Punktabstand das Maß der Punktänderung in Bezug auf die x-, y- und z-Koordinate angegeben würde.

Ein weiterer wichtiger Faktor für die Ergebnisbewertung einer entwickelten Methode zur Deformationsanalyse, ist die Möglichkeit Entscheidungen über statistisch signifikante Deformationen hervorbringen zu können. Ötsch et al. (2025) berücksichtigen eine stochastische Modellierung durch Anwendung einer Varianzkomponentenschätzung im Rahmen der geometrischen Punktwolkenapproximation mit Tensorprodukt B-Spline-Flächen und Sadiq und Harmening (2025) zeigen das Potenzial der stochastischen Modellierung für die Erfassung von Deformationsmustern. Ein Mitführen der Varianz-Kovarianzmatrix von TLS-Punktwolken während des Methodenprozesses ist zu empfehlen und soll mit der entwickelten Methode ebenfalls umgesetzt werden. Da sich die Modellierung von TLS-Daten mittels B-Spline-Flächen als ein äußerst vorteilhaftes Instrument herausgestellt hat, wird diese Technik in dieser Arbeit angewendet. Eine intensive Auseinandersetzung mit aktuellen Forschungsmethoden zur B-Spline-Modellierung von TLS-Daten findet in Kapitel 3.1 statt. Für die Zuweisung von Punktkorrespondenzen sind Registrierungsverfahren anzuwenden, die zwei Punktwolken bzw. Flächenmodelle mittels identischer Punkte in ein gemeinsames Koordinatensystem überführen. Eine ausgiebige Behandlung bestehender Registrierungsverfahren erfolgt in Kapitel 3.2, um das in dieser Arbeit entwickelte lokale Registrierungsverfahren besser einordnen zu können. Aktuelle Ansätze zur Deformationsanalyse, die auf B-Splines basieren, werden in Kapitel 3.3 diskutiert.

1.2 Zielsetzungen der Arbeit

TLS eröffnet durch seine flächenhafte hochauflösende Diskretisierung von Messobjekten großes Potential im Bereich der Deformationsanalyse. Eine zentrale Herausforderung bei der Messung und Arbeit mit TLS-Daten ist, dass bei der Messung zweier Epochen keine identischen Punkte, sondern entsprechend eines Winkelrasters folgend verteilte Einzelpunkte vorliegen. TLS-Rohdaten können somit im Rahmen einer klassischen Deformationsanalyse nicht unmittelbar verglichen werden und benötigen eine Nachbearbeitung. In dieser Nachbearbeitung müssen die Rohdaten modelliert und anschließend vergleichbar gemacht werden, um Deformationen feststellen zu können.

Als besonders vorteilhaft hat sich bei aktuellen Ansätzen der Fläche-zu-Fläche-basierte Ansatz herausgestellt. Dieser kann durch unterschiedliche approximierende Modelle umgesetzt werden. Eine große Entwicklungsmöglichkeit bietet in diesem Zusammenhang die B-Spline-Technik, da sie günstige Eigenschaften zur Modellierung von Daten enthält (Kapitel 2). Bisher existieren überwiegend Methoden, die eine B-Spline-Fläche durch die Schätzung von Parametern, wie z.B. der Knotenanzahl, erstellen (Unterkapitel 3.1). Ein konventioneller Ansatz im Rahmen einer vollständigen Deformationsanalyse, der ohne eine Parameterschätzung auskommt und unmittelbar auf TLS-Daten zurückgreift, um B-Spline-Flächen zu modellieren, wurde bisher noch nicht entwickelt. Ein limitierender Faktor besteht im Messrauschen, das durch eine stochastische Modellierung berücksichtigt werden kann.

In dieser Dissertation soll eine Fläche-zu-Fläche-basierte Deformationsanalyse entwickelt werden, die auf der Technik von B-Splines aufsetzt. Hierbei ist eine Methode zu entwerfen, die eine direkte Verwendung der TLS-Punktwolkendaten als Kontrollpunkte von B-Spline-Kurven bzw. -Flächen ermöglicht. Ein Ausgleichungsprozess zur Schätzung von B-Spline-Parametern soll nicht durchgeführt, sondern ein alternatives direktes Konzept ohne Ausgleichung erarbeitet werden. Für die lokale flächenhafte Registrierung zweier Messepochen ist ein Zielzeichen-freies Verfahren zu entwickeln, das einen Vergleich korrespondierender B-Spline-Kurven- bzw. Flächenpunkte gewährleistet. Der Prozess ist stochastisch zu modellieren, um mit Hilfe eines statistischen Tests signifikante Deformationen zwischen korrespondierenden B-Spline-Kurven- bzw. Flächenpunkten aufdecken zu können. Der Ansatz ist auf Basis von simulierten Daten zu entwickeln. Eine Evaluation der erarbeiteten Methode ist mittels

Realdaten auszuführen, um den aktuellen Stand für die Übertragung der Technik auf reale TLS-Daten abzufragen.

Die Zielsetzungen dieser Arbeit lauten demnach wie folgt:

- Entwicklung einer Methode zur Modellierung von TLS-Punktwolken auf Basis von B-Splines unter direkter Verwendung der TLS-Daten als B-Spline-Kontrollpunkte für 2D- und 3D-Daten (Modellierungsprozess)
- Entwicklung einer Zielzeichen-freien Registrierungsmethode zur Bestimmung und zum Vergleich korrespondierender B-Spline-Kurven- bzw. Flächenpunkte (lokale Zielzeichen-freie Registrierung)
- Stochastische Modellierung des Auswerteprozesses und Durchführung einer Deformationsanalyse mittels Signifikanztest (Deformationsanalyse)

1.3 Aufbau der Arbeit

Der Aufbau der Arbeit ist schematisch in Abbildung 1.1 dargestellt. Die grauen Rechtecke beziehen sich auf grundlegende Kapitel, wie Einleitung, Grundlagen, wissenschaftlicher Kontext und Zusammenfassung. Die magenta gefärbten Rechtecke bilden die essentiellen Kapitel der Arbeit betreffend der entwickelten Methode ab. Unterkapitel werden durch cyan gefärbte Rechtecke präsentiert. Um das Ablaufdiagramm übersichtlich zu halten, werden nicht alle Unterkapitel abgebildet.

Kapitel 2 befasst sich mit der Aufarbeitung mathematischer Grundlagen der B-Spline-Technik. In den Unterkapiteln werden Funktionen und wichtige Eigenschaften zu B-Spline-Basisfunktionen 2.1, B-Spline-Kurven 2.2 und B-Spline-Flächen 2.3 vermittelt. Eine Einführung in bestehende Modellierungsansätze mittels B-Splines durch Interpolation und Approximation erfolgt in Unterkapitel 2.4. Kapitel 3 ordnet die entwickelte Methode in den Rahmen aktueller Literatur ein und stellt in Unterkapitel 3.1 wissenschaftliche Veröffentlichungen zur Modellierung von TLS-Daten mittels B-Splines vor. Unterkapitel 3.2 widmet sich der Vorstellung und Einordnung aktueller Literatur in Sachen Registrierungsverfahren. In Kapitel 4 wird die in dieser Arbeit entwickelte Direkte B-Spline-Kontrollpunkt-Methode (DBKM) vorgestellt. Unterkapitel 4.1 beschäftigt sich mit Rahmenbedingungen zur Methode und gibt einen Überblick über den Prozessablauf. Unterkapitel 4.2 behandelt die notwendige Vorprozessierung, um aus sortierten TLS-Daten eine gleichmäßige bidirektionale Kontrollgitterstruktur für die Erstellung von Tensor-Produkt-B-Spline-Flächen zu generieren. Unterkapitel 4.3 demonstriert daraufhin ein entwickeltes lokales Zielzeichen-freies Registrierungsverfahren, den sogenannten Mapcut-Algorithmus, auf Basis eines Krümmungsparameters als Merkmalsdeskriptor und einer Korrelationsanalyse. Hierbei wird der Mapcut-Algorithmus sowohl für den 2D-Fall als auch den 3D-Fall erarbeitet. Kapitel 5 setzt sich mit der stochastischen Modellierung des Auswerteprozesses für B-Spline-Kurven (5.1) und B-Spline-Flächen (5.2, 5.3) auseinander, um in Unterkapitel 5.4 in einem Signifikanztest zu münden. Kapitel 6 stellt die in dieser Arbeit verwendeten bzw. selbst erstellten Messobjekte vor. In einer Ergebnisanalyse bearbeitet Kapitel 7 in Unterkapitel 7.1 die Ergebnisse der Anwendung der DBKM für die verwendeten simulierten und realen Datensätze und verweist in Unterkapitel 7.2 auf die Ergebnisse der Deformationsanalyse mittels Signifikanztest für simulierte Datensätze. Das letzte Kapitel 8 gibt eine Zusammenfassung und einen Ausblick für zukünftige Arbeiten.

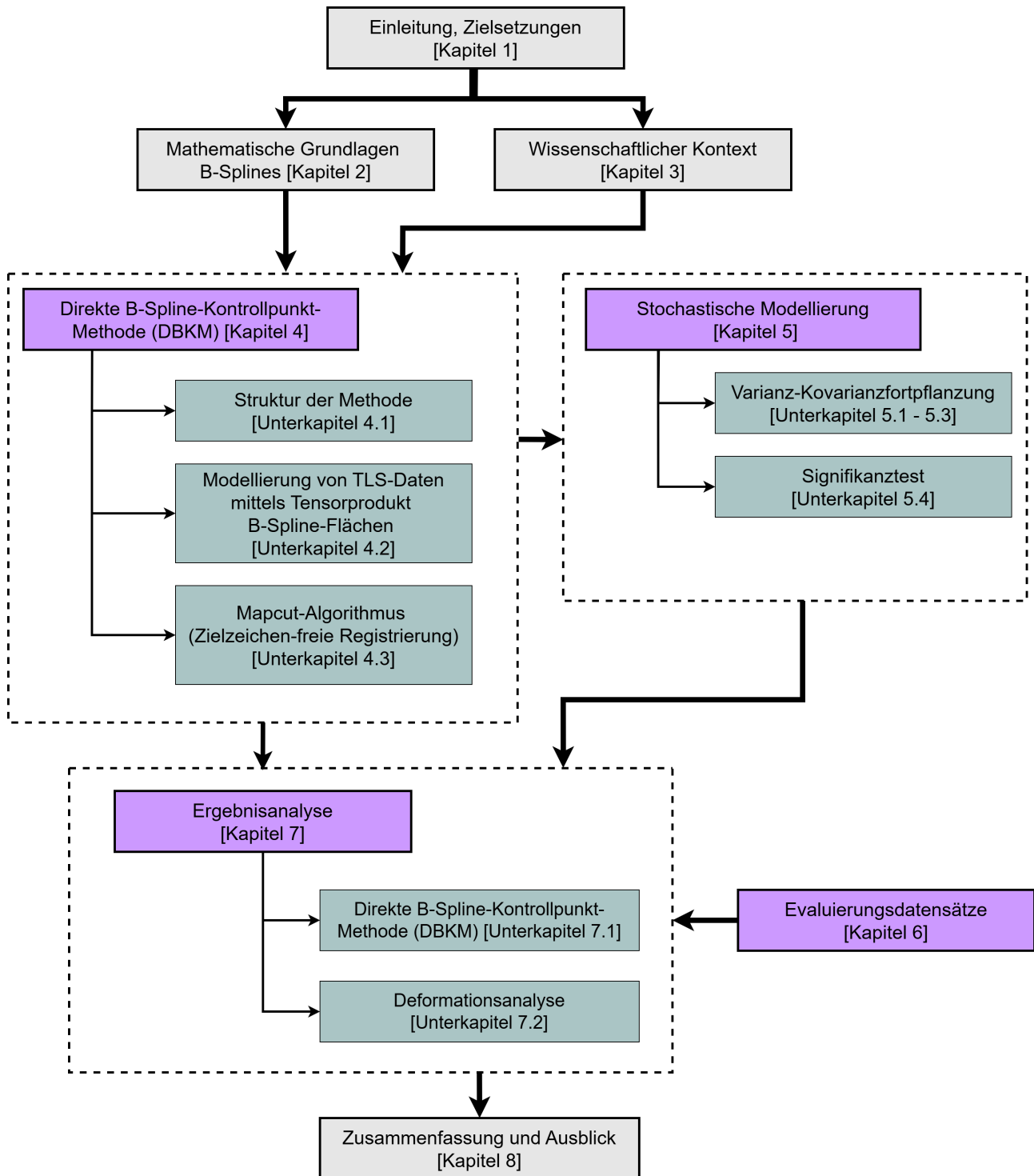


Abb. 1.1: Struktur der Arbeit. Basiskapitel (grau), Kapitel der entwickelten Methode (magenta), Unterkapitel (cyan).

2 Mathematische Grundlagen zu B-Splines

Für die geometrische Modellierung von Leitkurven, Profilen oder Flächen sind klassische Interpolationsfunktionen im Allgemeinen ungeeignet, da die resultierende Kurve starken Krümmungsunterschieden unterliegt und zur Oszillation neigt. Eine Erhöhung der Stützstellen erzeugt zwar eine einheitlichere Kurvenform, allerdings korreliert diese Maßnahme mit einem erhöhten Polynomgrad n und dem Auftreten von extremen Abweichungen an gewissen Stellen (Hoschek und Lasser 1992). Erkennbar ist dieses Verhalten in der linken Grafik von Abbildung 2.1, auf der ein Kontrollpolygon mit den Kontrollpunkten $\mathbf{d}_0$ bis $\mathbf{d}_6$ sowie ein Interpolationspolynom 6. Grades basierend auf der Lagrange-Darstellung aufgeführt ist. Das starke Oszillieren zwischen Kontrollpunkt $\mathbf{d}_0$ und $\mathbf{d}_1$ verdeutlicht die eklatanten Krümmungsunterschiede bei klassischen Interpolationsverfahren. Zur Konstruktion der Lagrange-Interpolation wird auf die Formeln in Richter und Wick (2017) verwiesen.

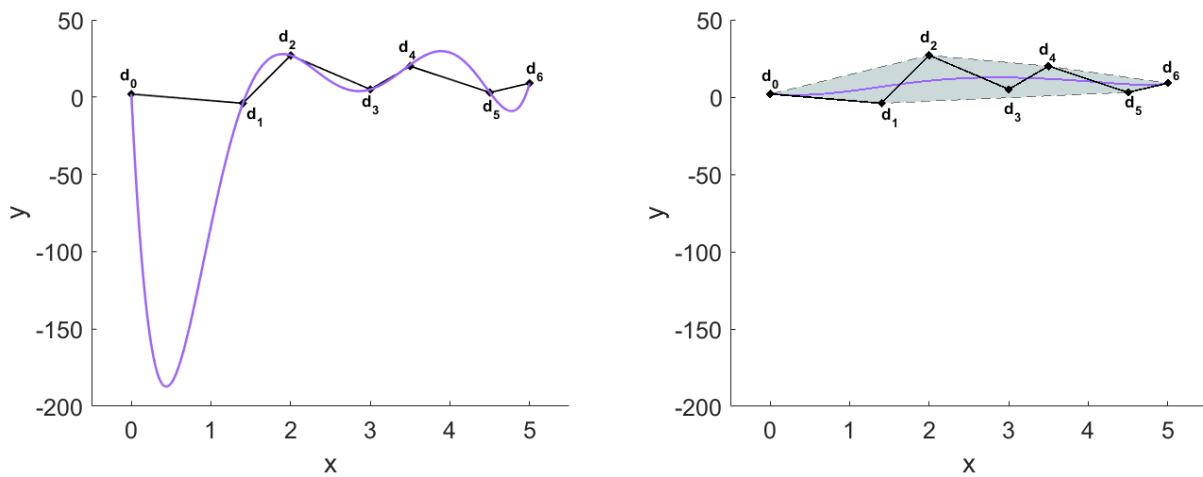


Abb. 2.1: Kontrollpolygon bestehend aus den Kontrollpunkten $\mathbf{d}_0$ bis $\mathbf{d}_6$, links: Interpolierendes Polynom vom Grad 6, rechts: Bézier-Kurve vom Grad 6 mit konvexer Hülle.

Als bessere Alternative zur Modellierung von Kurven kann die Bézier-Technik verwendet werden, deren Basisfunktion aus einem Bernsteinpolynom besteht. Hierbei sind Kurve und Kontrollpolygon derart eng miteinander verknüpft, dass das Kurvensegment in der konvexen Hülle des Polygons liegt und somit keine Ausreißer möglich sind. Eine Darstellung einer Bézier-Kurve vom Grad 6, bestehend aus den gleichen sieben Kontrollpunkten, ist exemplarisch in der rechten Grafik von Abbildung 2.1 präsentiert. Die graue Schraffur zeigt die zugehörige konvexe Hülle. Für weitere Informationen zur Bézier-Technik wird auf Farin (1994) und Prautzsch et al. (2002) verwiesen. Kurven, die lediglich aus einem Polynom oder rationalem Segment bestehen, enthalten jedoch diverse Defizite und Einschränkungen bezüglich der Flexibilität. So ist etwa bei Bézier-Kurven die Anzahl der Kontrollpolygone fest mit dem Grad der resultierenden Kurve verankert. Eine kubische Bézier-Kurve kann demgemäß nur mittels eines Polygons, aus vier Punkten bestehend, generiert werden. Um komplexe Formen modellieren zu können, ist bei Verwendung derartiger Basisfunktionen ein hoher Grad in Folge der erheblichen Anzahl an Kontrollpunkten notwendig. Allerdings steht ein hoher Kurvengrad für ineffizientes Prozessieren sowie für numerische Instabilität (Piegl und Tiller 1997). Zusätzlich sind Kurven, die ausschließlich aus einem Polynom bestehen, nicht in hohem Maße für interaktiv-grafische Modellierungsprozesse geeignet, da jegliche Kontrollpunktverschiebung eine globale Änderung der Kurve verursacht und somit zielgerichtete lokale Korrekturmaßnahmen an der Kurve faktisch ausschließen. Ursache hierfür ist die

globale Charakteristik der Basisfunktion, in Folge derer die Anwendung der Bézier-Technik für den Bereich der Deformationsanalyse als wenig geeignet hervorgeht.

Abhilfe bilden Splinefunktionen der Ordnung o , die sich als stückweise zusammengesetzte Polynome vom Grad $n = o - 1$ definieren, die an den Segmentübergängen $(n - 1)$ -mal stetig differenzierbar sind und somit die Stetigkeitsbedingung C^{m-1} erfüllen.

Wird eine Splinekurve $C(t)$ als Affinkombination von Kontrollpunkten $\mathbf{d}_i$ mit

$$C(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t), \quad t \in [u_0, u_k] \quad (2.1)$$

konstruiert, wobei $N_i^n(t)$ die von Schoenberg (1967) benannte B-Spline-Basisfunktion mit $i = 0, 1, \dots, m$ ($\mathbb{R}^2$ und $\mathbb{R}^3$) darstellt, handelt es sich um eine sogenannte B-Spline-Kurve, die besonders vorteilhafte Eigenschaften zur geometrischen Modellierung von Kurven zeigt (vgl. Prautzsch et al. (2002)). Eine dieser Charakteristiken beinhaltet die nicht globale Verhaltensweise bei Kontrollpunktänderungen. Um den maßgebenden Inhalt von B-Spline-Kurven verstehen zu können, sollen in Abschnitt 2.1 B-Spline-Basisfunktionen eingehend erläutert werden.

Die B-Spline-Technik wird erstmals von Schoenberg im Jahre 1946 eingeführt, der die Spline-Approximation anhand dividierter Differenzen zum Glätten statistischer Daten nutzt und so eine überwiegend mathematische Herangehensweise darlegt (Curry und Schoenberg (1947), Schoenberg (1946)). Mittels der Knoteneinfügealgorithmen nach Boehm (1980) oder dem Blossoming von Ramshaw (1987) kann dem Anwender der thematische Zugang zu B-Splines ebenso eröffnet werden. Eine Rekursionsformel zur numerischen Berechnung der Basisfunktionen entwickeln Cox (1972) und de Boor (1972, 1978) in den 70er Jahren unabhängig voneinander. Riesenfeld (1973) sowie Gordon und Riesenfeld (1974) wenden die B-Spline-Basisfunktion schließlich zur Kurvendefinition an. In Folge der Praktikabilität zur rechnergestützten Implementierung wird die Rekursionsformel zur Definition der B-Spline-Basisfunktionen in dieser Arbeit verwendet.

2.1 B-Spline-Basisfunktionen

Es sei

$$\mathbf{U} = \begin{pmatrix} u_0 & u_1 & \dots & u_{k-1} & u_k \end{pmatrix} \quad (2.2)$$

eine monoton steigende Folge reeller Zahlen, wie etwa $u_i \leq u_{i+1}$ mit $i = 0, 1, \dots, k - 1$. Die Variablen u_i werden Träger oder Knoten genannt und bilden $(k + 1)$ Elemente eines Träger- bzw. Knotenvektors $\mathbf{U}$, der in Unterabschnitt 2.1.1 näher behandelt wird. Als i -te B-Spline-Basisfunktion vom Grad n , der Ordnung $n + 1$, mit dem Parameterwert $t \in [u_0, u_k]$ werden folgende Funktionen deklariert:

$$N_i^0(t) = \begin{cases} 1 & \text{für } u_i \leq t < u_{i+1} \\ 0 & \text{sonst} \end{cases} \quad i = 0, \dots, k - 1, \quad (2.3)$$

$$N_i^n(t) = \frac{t - u_i}{u_{i+n} - u_i} \cdot N_i^{n-1}(t) + \frac{u_{i+n+1} - t}{u_{i+n+1} - u_{i+1}} \cdot N_{i+1}^{n-1}(t). \quad (2.4)$$

Anhand Gleichung (2.3) lässt sich erkennen, dass $N_i^0(t)$ eine Treppenfunktion darstellt, die für jeden außerhalb der Menge $t \in [u_i, u_{i+1})$ liegenden Parameterwert t den Wert 0 annimmt. Hieraus ist ebenso die Kausalität $N_i^0(t) = 0$ bei $u_i = u_{i+1}$ ableitbar. Gleichung (2.4) verdeutlicht die Abhängigkeit der

Generierung von $N_i^n(t)$ für einen Grad $n > 0$ als Linearkombination zweier B-Spline-Basisfunktionen vom Grad $n - 1$, woraus sich die Bedingung $N_i^n(t) = 0$ außerhalb der Menge $t \in [u_i, u_{i+n+1})$ ergibt. Zusätzlich ist zu beachten, dass der Quotient $\frac{0}{0}$ auftreten kann, der als 0 definiert wird. Vergleiche hierzu Piegl und Tiller (1997).

2.1.1 Knotenvektoren

Wie in Gleichung (2.3) und (2.4) nachvollzogen werden kann, hat die Wahl des Knotenvektors einen signifikanten Einfluss auf die Eigenschaften der B-Spline-Basisfunktion $N_i^n(t)$ und dementsprechend auf die resultierende B-Spline-Kurve. Als Knotenvektor kann jegliche Folge monoton steigender reeller Zahlen angesehen werden, die der Voraussetzung $u_i \leq u_{i+1}$ Rechnung trägt (siehe Gleichung (2.2)). Hierbei ist eine Differenzierung in uniforme und nicht uniforme Knotenvektoren zu treffen.

Uniforme Knotenvektoren enthalten äquidistante Knotenabstände, während nicht uniforme Knotenvektoren heterogene Distanzen zwischen den Knotenwerten vorweisen. Bei der Unterscheidung zwischen uniformen und nicht uniformen Knotenvektoren ist lediglich auf die Knotenabstände zu achten. Knoten innerhalb eines Knotenvektors können auch negative Werte annehmen, solange sie eine monoton steigende Folge reeller Zahlen abbilden. Zur Veranschaulichung werden in Gleichung (2.5) und (2.6) uniforme Knotenvektoren, in Gleichung (2.7) und (2.8) hingegen nicht uniforme Knotenvektoren dargestellt:

$$(0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6), \quad (2.5)$$

$$(-0,6 \quad -0,5 \quad -0,4 \quad -0,3 \quad -0,2), \quad (2.6)$$

$$(0 \quad 2 \quad 3 \quad 6 \quad 7), \quad (2.7)$$

$$(0,00 \quad 0,25 \quad 0,30 \quad 0,60 \quad 0,75 \quad 1,00). \quad (2.8)$$

Zusätzlich ist eine Normalisierung auf den Parameterbereich zwischen 0 und 1 möglich. Die Ableitung eines nicht genormten Knotenvektors in eine normalisierte Form erfolgt mittels:

$$\mathbf{U}_{\text{Norm}} = \frac{\mathbf{U}}{u_k}. \quad (2.9)$$

Ein normalisierter Knotenvektor wird mit Gleichung (2.8) exemplarisch präsentiert. Vergleiche diesbezüglich Rogers (2001).

Abgesehen von uniformer und nicht uniformer Ausprägung, ist eine Typisierung von Knotenvektoren anwendbar. Hierbei wird zwischen periodischen sowie nicht periodischen (bzw. offenen) Knotenvektoren unterschieden. Die in dieser Arbeit durchgeführten Auswertungen und Analysen werden ausschließlich mit offenen Knotenvektoren verwirklicht. Der Vollständigkeit halber sowie für ein umfassenderes Verständnis, findet ungeachtet dessen ein Umriss der Thematik von periodischen Knotenvektoren statt. Nach Rogers (2001) führt die Wahl eines periodischen uniformen Knotenvektors zur Erzeugung periodischer uniformer Basisfunktionen, für die gilt:

$$N_i^n(t) = N_{i-1}^n(t-1) = N_{i+1}^n(t+1). \quad (2.10)$$

In dieser Konstellation verkörpert jede B-Spline-Basisfunktion die Translation einer anderen. Eine Veranschaulichung der Parallelverschiebung erfolgt in Abbildung 2.2 und 2.3, in der periodische B-Spline-Basisfunktionen nullten bis dritten Grades auf Grundlage des periodischen uniformen Knotenvektors

$$\mathbf{U} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} \quad (2.11)$$

illustriert werden.

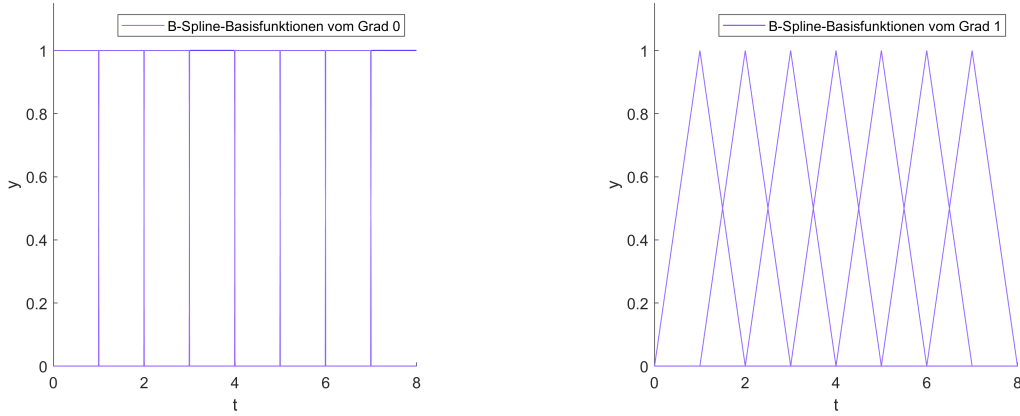


Abb. 2.2: Periodische B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts).

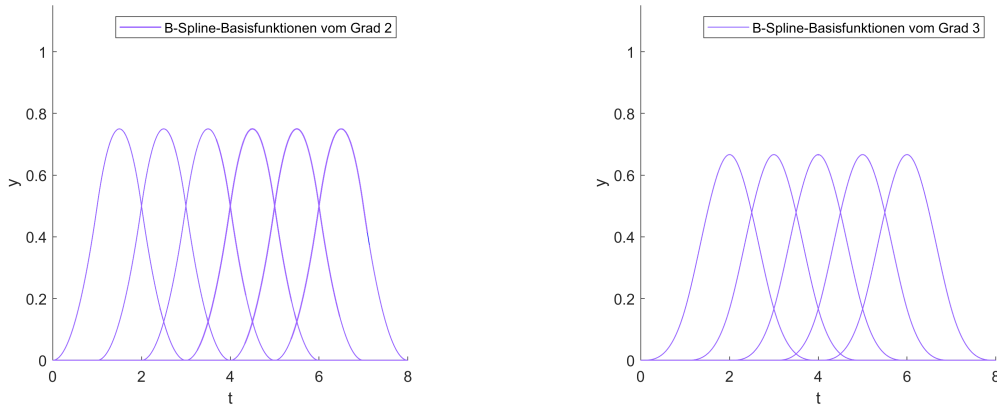


Abb. 2.3: Periodische B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts).

Bei eingehender Betrachtung des zugehörigen Knotenvektors $\mathbf{U}$ kann festgestellt werden, dass jeder Knoten ausschließlich einmal vorliegt. In diesem Fall wird von einer Vielfachheit von $l = 1$ gesprochen. Enthalten die Knotenwerte am Vektoranfang und -ende eine Vielfachheit von $(n + 1)$, spricht man von einem offenen bzw. nicht periodischen Knotenvektor mit der Randvielfachheit $(n + 1)$:

$$\mathbf{U} = \left(\underbrace{u_0 = u_1 = \dots = u_n}_{n+1 \text{ mal}} \ u_{n+1} \ \dots \ u_{k-n-1} \ \underbrace{u_{k-n} = u_{k-n+1} = \dots = u_k}_{n+1 \text{ mal}} \right). \quad (2.12)$$

Da offene Knotenvektoren eine Randvielfachheit von $(n + 1)$ vorweisen, gilt für diese ein Parameterintervall von $t \in [u_0, u_k] = [u_n, u_{k-n}]$. Zur besseren Verdeutlichung des Einflusses der Wahl des Knotenvektors auf die B-Spline-Basisfunktion werden nachfolgend mehrere Beispiele umgesetzt. Hierzu

wird ein nicht periodischer, uniformer Knotenvektor, bestehend aus 13 Knoten und der Randvielfachheit $l = 4$, gebildet, da eine Generierung von B-Spline-Basisfunktionen bis Grad 3 realisiert werden soll:

$$U = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 6 & 6 \end{pmatrix}. \quad (2.13)$$

Mittels der Gleichungen (2.3) und (2.4) kann für jeden Parameterwert $t \in [0, 6]$ eines Parametervektors $\mathbf{t}$ ein Punkt einer offenen uniformen B-Spline-Basisfunktion nullten bis dritten Grades konstruiert und in den Abbildungen (2.4) und (2.5) nachvollzogen werden.

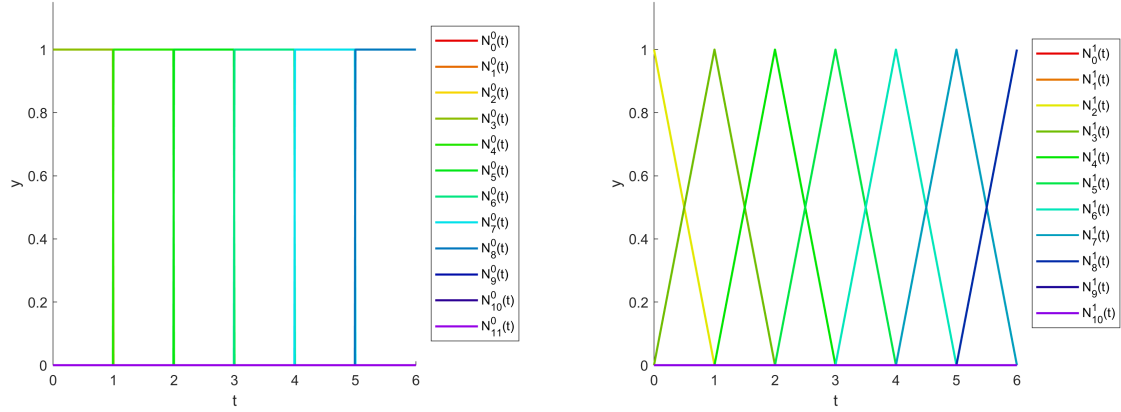


Abb. 2.4: Nicht periodische uniforme B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts).

Die farbige Hervorhebung der einzelnen B-Spline-Basisfunktionen soll zu einem leichteren Verständnis der erzeugten Basisfunktionen in Bezug auf ihre Belegung mit dem Wert 0 dienen. Die B-Spline-Basisfunktionen nullten Grades ergeben Rechteckfunktionen der Höhe 1 über dem Knotenintervall $[u_i, u_{i+1})$ mit $i = 0, \dots, k - 1$ und einer Knotenanzahl von $k + 1 = 13$. Außerhalb des Knotenintervalls sind die Funktionen Null. In diesem Beispiel betrifft dies die Basisfunktionen $N_0^0(t)$ bis $N_2^0(t)$ und $N_9^0(t)$ bis $N_{11}^0(t)$, da für diese $u_i = u_{i+1}$ und dementsprechend $N_i^0(t) = 0$ gilt. Zu beobachten ist, dass die B-Spline-Basisfunktionen der Ordnung 1 teilweise konstant und an den Segmenttrenngrenzen unstetig sind.

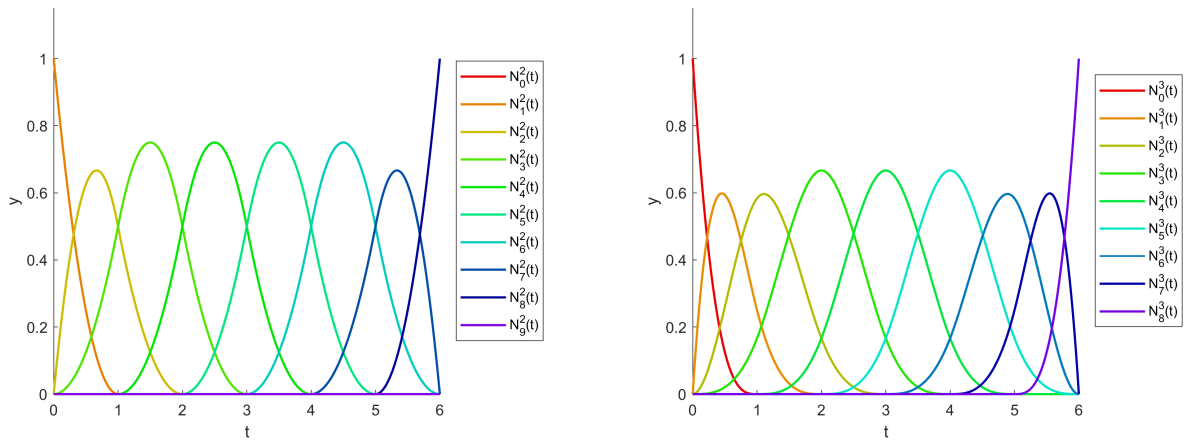


Abb. 2.5: Nicht periodische uniforme B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts) (Aichinger und Schwieger 2022).

Für B-Spline-Basisfunktionen der Ordnung 2 und dem Polynomgrad 1 ergeben sich Dreiecksfunktionen mit dem Knotenintervall $[u_i, u_{i+2})$. Das entsprechende Knotenintervall, innerhalb dessen die B-Spline-Basisfunktion nicht mit dem Wert Null belegt ist, kann mit Hilfe des Grades der Basisfunktion, konkret durch $[u_i, u_{i+n+1})$, ermittelt werden. Diese Eigenschaft wird auch Lokalität genannt und soll in Unterabschnitt 2.1.2 näher beleuchtet werden. $N_2^1(t)$ bis $N_8^1(t)$ sind stückweise linear und an den Segmenttrenngrenzen stetig. Durch Berechnung der B-Spline-Basisfunktionen zweiten Grades werden die Dreiecksfunktionen zu parabelähnlichen Funktionen mit dem Knotenintervall $[u_i, u_{i+n+1}) = [u_i, u_{i+3})$. $N_1^2(t)$ bis $N_8^2(t)$ sind stückweise quadratisch und können an den Segmenttrenngrenzen einmal abgeleitet werden. Auch die B-Spline-Basisfunktionen $N_i^3(t)$ der Ordnung 4 mit dem Polynomgrad 3 ergeben parabelähnliche Funktionen, jedoch mit dem Knotenintervall $[u_i, u_{i+4})$. Sie sind an den Segmenttrenngrenzen zweimal stetig differenzierbar und aufgrund des dritten Grades stückweise kubisch. Dieses Beispiel orientiert sich an Bender und Bill (2006).

Wird statt des offenen uniformen Knotenvektors ein offener nicht uniformer Knotenvektor wie in Gleichung (2.14) zur Generierung der B-Spline-Basisfunktionen vom Grad 0 bis 3 gewählt, entstehen die in den Abbildungen 2.6 und 2.7 dargestellten Basisfunktionen.

$$U = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 4 & 5 & 6 & 6,1 & 6,3 & 7 & 8 & 8 & 8 & 8 \\ u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} & u_{14} \end{pmatrix} \quad (2.14)$$

Mittels der farbigen Darstellung der B-Spline-Basisfunktionen können die Abhängigkeiten zwischen Knoten und B-Spline-Basisfunktionen, im Speziellen in Hinblick auf die im nachfolgenden Abschnitt behandelte Eigenschaft der Lokalität, besser nachvollzogen werden. In Kontrast zu den uniformen B-Spline-Basisfunktionen von Abbildung 2.4 und 2.5, zeichnen sich nicht uniforme B-Spline-Funktionen durch deutliche Dysbalancen aus, die in Folge unterschiedlicher Knotendistanzen verursacht werden.

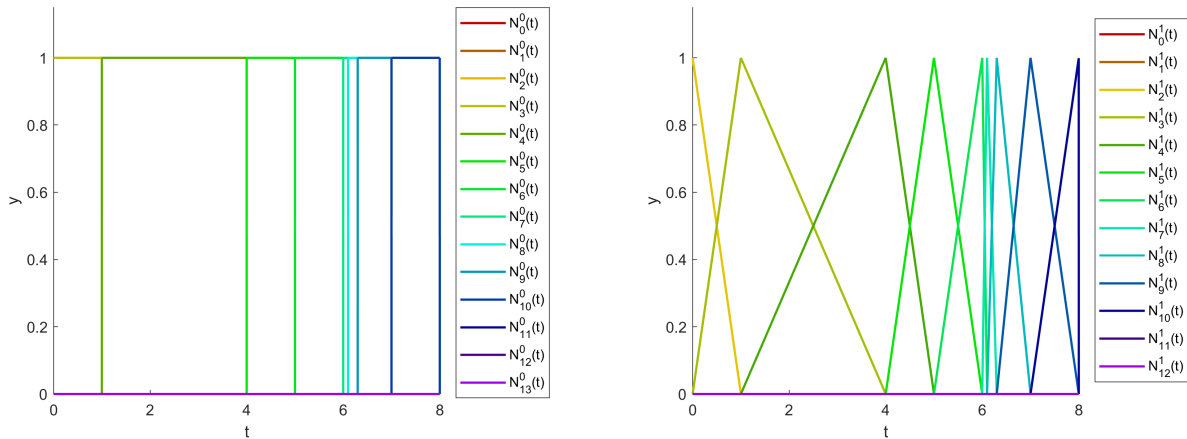


Abb. 2.6: Nicht uniforme offene B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts).

Besonders eindrucksvoll äußert sich der vergrößerte Abstand zwischen Knoten u_4 und u_5 in der gedehnten Ausprägung der B-Spline-Basisfunktion $N_4^0(t)$, sowie die reduzierte Distanz zwischen u_7 bis u_9 in einer Stauchung der Basisfunktionen $N_7^0(t)$ und $N_8^0(t)$. Mittels des erweiterten Knotenintervalls $[u_i, u_{i+2})$ der B-Spline-Basisfunktionen ersten Grades, ist eine Beeinflussung von jeweils zwei Basisfunktionen über einem Knotenintervall erkennbar. Dementsprechend vergrößert sich die Auswirkung nicht äquidistanter Knotenabstände je höher der Grad der B-Spline-Basisfunktion gewählt wird, da die Breite des Knotenintervalls ebenso zunimmt.

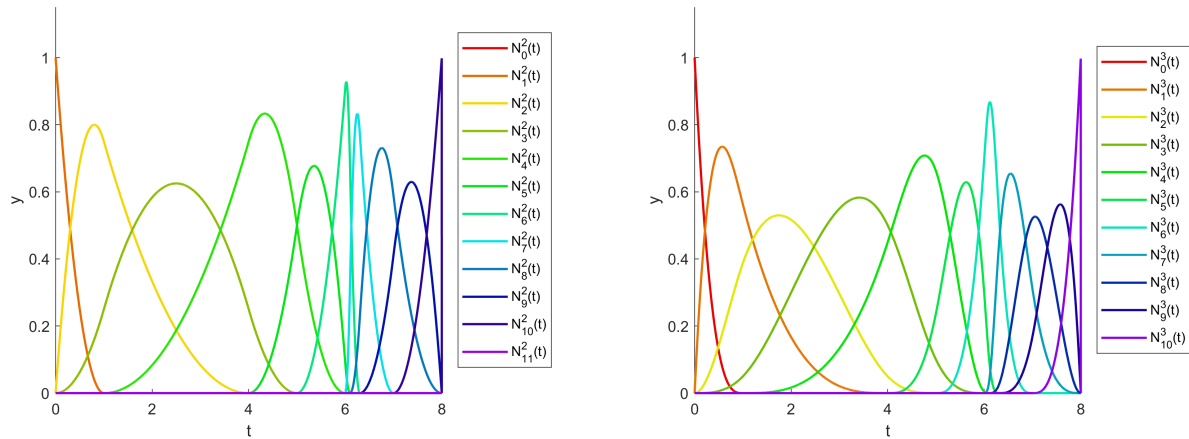


Abb. 2.7: Nicht uniforme offene B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts).

Eine eingehende Betrachtung dieser Verhaltensweise findet im folgenden Abschnitt bei Behandlung der Eigenschaften von B-Spline-Basisfunktionen statt.

2.1.2 Eigenschaften von B-Spline-Basisfunktionen

Lokalität

Eine der wichtigsten Eigenschaften von B-Spline-Basisfunktionen wird Lokalität genannt. Diese besagt, dass

$$N_i^n(t) = 0 \quad \text{falls} \quad t \notin [u_i, u_{i+n+1}), \quad (2.15)$$

wobei n dem Polynomgrad der i ten B-Spline-Basisfunktion mit $i = 0, 1, \dots, k-1$ und t einem Parameterwert auf dem Intervall $[u_0, u_k]$ entspricht. u_i sind die Knoten eines Knotenvektors $\mathbf{U}$. Siehe hierzu Piegel und Tiller (1997).

Aus Gleichung (2.4) ist ersichtlich, dass $N_i^n(t)$ eine Linearkombination von $N_i^{n-1}(t)$ und N_{i+1}^{n-1} darstellt. Diese Abhängigkeit wird im Dreiecksschema von Gleichung (2.16) verbildlicht.

$$\begin{array}{ccccccc}
 & & & & & & N_0^0 \\
 & & & & & & N_0^1 \\
 & & & & & N_0^2 & \\
 & & & N_1^1 & N_0^3 & & \\
 N_2^0 & & N_1^2 & N_1^3 & N_0^4 & & \\
 & N_2^1 & N_1^4 & N_0^5 & & & \\
 N_3^0 & & N_2^2 & N_1^5 & & & \\
 & N_3^1 & N_2^3 & N_1^6 & & & \\
 N_4^0 & & N_3^2 & N_2^4 & & & \\
 & N_4^1 & N_3^3 & N_2^5 & & & \\
 N_5^0 & & N_4^2 & & & & \\
 & N_5^1 & & & & & \\
 N_6^0 & & & & & &
 \end{array} \quad (2.16)$$

So kann $N_2^3(t)$ als Linearkombinationen von $N_2^0(t)$, $N_3^0(t)$, $N_4^0(t)$ und $N_5^0(t)$ generiert werden und hat folglich lediglich im Intervall $t \in [u_2, u_6]$ Einfluss, da nur dort $N_2^3(t) \neq 0$ gilt.

Im Gegenzug ist eine Ermittlung betroffener B-Spline-Basisfunktionen eines definierten Intervalls $[u_j, u_{j+1})$ realisierbar, da für einen festgelegten Grad n die B-Spline-Funktionen $N_{j-(n-s)}^n(t)$ für $s = 0, 1, \dots, n$ nicht den Wert 0 annehmen. Verwendet man beispielhaft den Knotenvektor in Gleichung (2.14) über dem Intervall $[1, 4) = [u_4, u_5)$ und gibt einen Polynomgrad von $n = 2$ vor, lassen sich für $j = 4$ die B-Spline-Basisfunktionen $N_2^2(t)$, $N_3^2(t)$ und $N_4^2(t)$ bestimmen. Diese Verhaltensweise lässt sich in den farblichen Darstellungen von Abbildung 2.6 und 2.7 überprüfen.

Allgemein kann die Faustregel festgesetzt werden, dass zur Bildung einer B-Spline-Basisfunktion vom Grad n höchstens $n + 1$ Basisfunktionen vom Grad 0 beteiligt sind.

Teilung der Eins

Addiert man die Werte offener B-Spline-Basisfunktionen $N_i^n(t)$ mit $i = 0, \dots, k - n - 1$ innerhalb des Parameterintervalls $[u_0, u_k]$ für jeden Parameterwert t auf, ergibt sich jeweils die Summe Eins. Somit gilt:

$$\sum_{i=0}^{k-n-1} N_i^n(t) = 1, \quad t \in [u_0, u_k] = [u_n, u_{k-n}]. \quad (2.17)$$

Durch die Randvielfachheit von $(n+1)$ gilt $[u_0, u_k] = [u_n, u_{k-n}]$. Vergleiche hierzu Bender und Bill (2006). Abbildung 2.8 verdeutlicht die Aufsummierung nicht periodischer B-Spline-Basisfunktionen zu Eins anhand der grauen Linie, deren Position der Summe der B-Spline-Basisfunktionen im Parameterwert t entspricht. Zur Konstruktion dieser B-Spline-Basisfunktionen zweiten Grades wird der Knotenvektor $U = (0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 5 \ 5)$ herangezogen.

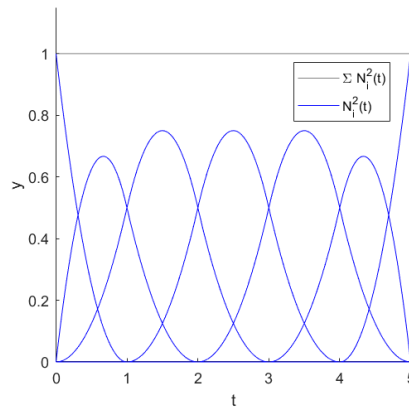


Abb. 2.8: Offene B-Spline-Basisfunktionen vom Grad 2

Soll diese Bedingung ebenso für periodische B-Spline-Basisfunktionen gelten, ist das Parameterintervall t ebenfalls auf $[u_n, u_{k-n}]$ zu beschränken, allerdings kann aufgrund der Randvielfachheit von $l = 1$ keine Gleichstellung zu $[u_0, u_k]$ angebracht werden. Es gilt:

$$\sum_{i=0}^{k-n-1} N_i^n(t) = 1, \quad t \in [u_n, u_{k-n}], \quad (2.18)$$

wobei $N_i^n(t)$ mit $i = 0, \dots, k - n - 1$ (Rogers 2001). Aus Gleichung (2.18) resultiert, dass im Polynomgrad $n = 0$ keine Einschränkung des Parameterintervalls ausgelöst wird.

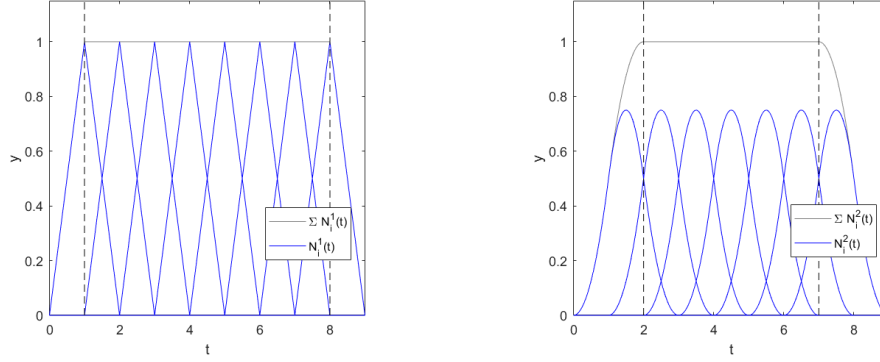


Abb. 2.9: Periodische B-Spline-Basisfunktionen vom Grad 1 (links) und Grad 2 (rechts).

Betrachtet man diese Eigenschaft am Beispiel von periodischen B-Spline-Basisfunktionen von Abbildung 2.9, die einen Knotenvektor von $\mathbf{U} = (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9)$ vorweisen, ist bei $n = 1$ der Parameterbereich bereits auf $(u_1 \leq t \leq u_8) = (1 \leq t \leq 8)$ reduziert, um eine Summierung zu Eins zu erhalten. Der eingeschränkte Bereich kann mittels der schwarz gestrichelten Linien in den zwei Grafiken eingesehen werden. Die graue Linie präsentiert hingegen die Summe aller B-Spline-Basisfunktionen im jeweiligen Parameterwert t . Bei $n = 2$ reduziert sich das Parameterintervall weiter auf $t \in [u_2, u_7] = [2, 7]$. Je höher der Polynomgrad demzufolge gewählt wird, desto stärker reduziert sich das Parameterintervall und desto weniger Knoten werden vom Randbereich des Trägervektors in die Definition des Parameterintervalls aufgenommen.

Nicht-Negativität

Für $n = 0$ ist $N_i^0(t)$ nach Gleichung (2.3) per Definition 0 oder 1 und somit nicht negativ. Für einen Polynomgrad $n > 0$ nach Gleichung (2.4) sollte beachtet werden, dass $N_i^n(t)$ eine Linearkombination von $N_i^{n-1}(t)$ und $N_{i+1}^{n-1}(t)$ darstellt, die lediglich Negativität zeigt, wenn einer der Terme von Gleichung (2.19), die mit den B-Spline-Basisfunktionen niedrigeren Grades multipliziert werden, negative Werte annehmen würde. Anhand der Lokaltäts-Eigenschaft ist $N_i^{n-1}(t) = 0$, falls $t \notin [u_i, u_{i+n})$. Demgemäß könnten die beiden Terme maximal innerhalb des Intervalls $[u_i, u_{i+n})$ Werte ungleich Null zeigen. Allerdings impliziert $t \in [u_i, u_{i+n})$, dass

$$\frac{t - u_i}{u_{i+n} - u_i} \quad \text{sowie} \quad \frac{u_{i+n+1} - t}{u_{i+n+1} - u_{i+1}} \quad (2.19)$$

keine Werte kleiner Null annehmen kann. Somit ist $N_i^n(t)$ nicht negativ. Expliziter kann dieser Nachweis in Piegel und Tiller (1997) eingesehen werden.

Differenzierbarkeit

Tritt ein Knoten u_i in einem Knotenvektor $\mathbf{U}$ mit der Vielfachheit l_i auf, verringert sich der Grad der Differenzierbarkeit der B-Spline-Basisfunktionen an der Stelle $t = u_i$ von C^{m-1} auf C^{m-l_i} . Dort ist N_i^n genau $n - l_i$ mal stetig differenzierbar. Siehe hierzu Abramowski und Müller (1991).

Mittels folgendem Beispiel soll eine Veranschaulichung dieses Sachverhalts erzeugt werden. Unter Verwendung des Knotenvektors $\mathbf{U} = (0 \ 1 \ 2 \ 3 \ 3 \ 4 \ 5)$ werden nicht uniforme B-Spline-Basisfunktionen nullten bis dritten Grades, in Abbildung 2.10 und 2.11 dargestellt, generiert.

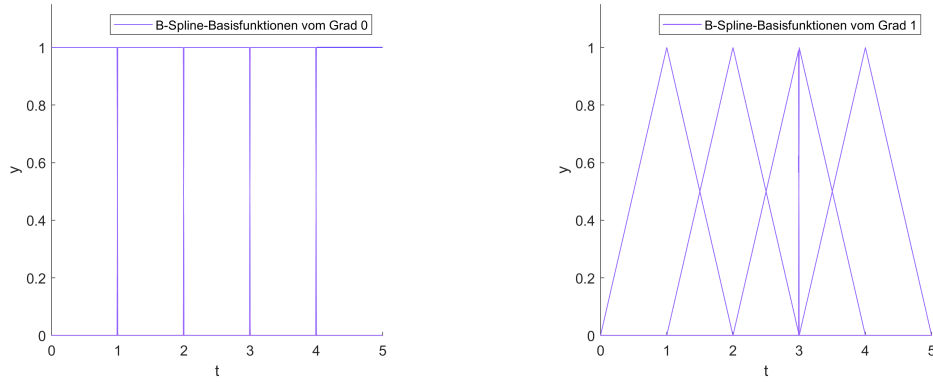


Abb. 2.10: Nicht uniforme B-Spline-Basisfunktionen vom Grad 0 (links) und Grad 1 (rechts) mit einer Knotenvielfachheit von bis zu 2.

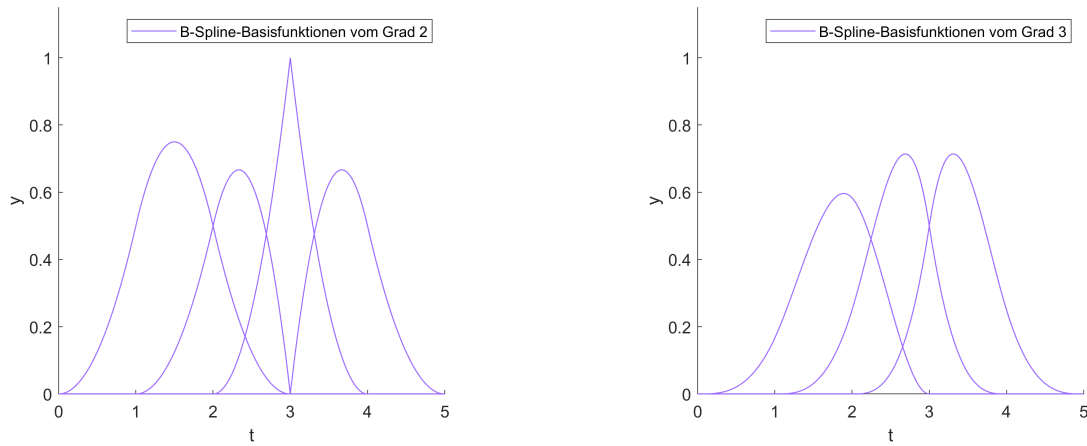


Abb. 2.11: Nicht uniforme B-Spline-Basisfunktionen vom Grad 2 (links) und Grad 3 (rechts) mit einer Knotenvielfachheit von bis zu 2.

Aufgrund des Knotens $u_3 = 3$ mit einer Vielfachheit von $l = 2$ reduziert sich die Differenzierbarkeit der B-Spline-Basisfunktionen ersten Grades $N_2^1(t)$ und $N_3^1(t)$ an der Stelle $t = 3$ auf C^{-1} . In Folge dessen sind diese B-Spline-Funktionen bei $t = 3$ unstetig. Die zweifache Knotenvielfachheit erzeugt bei $n = 2$ an der Stelle $t = 3$ einen C^0 -Übergang. B-Spline-Basisfunktionen dritten Grades weisen hingegen eine C^1 -Stetigkeit vor und sind somit stetig differenzierbar.

Ableitungen von B-Spline-Basisfunktionen

Nach Piegel und Tiller (1997) lautet die b -te Ableitung einer B-Spline-Basisfunktion $N_i^n(t)$ wie folgt:

$$N_i^{n(b)}(t) = \frac{n}{u_{i+n} - u_i} \cdot N_i^{n-1(b-1)}(t) - \frac{n}{u_{i+n+1} - u_{i+1}} \cdot N_{i+1}^{n-1(b-1)}(t). \quad (2.20)$$

Zur Berechnung der ersten Ableitung $N_i^{n(1)}(t)$, ist $N_i^{n-1(0)}(t) = N_i^{n-1}(t)$ zu verwenden.

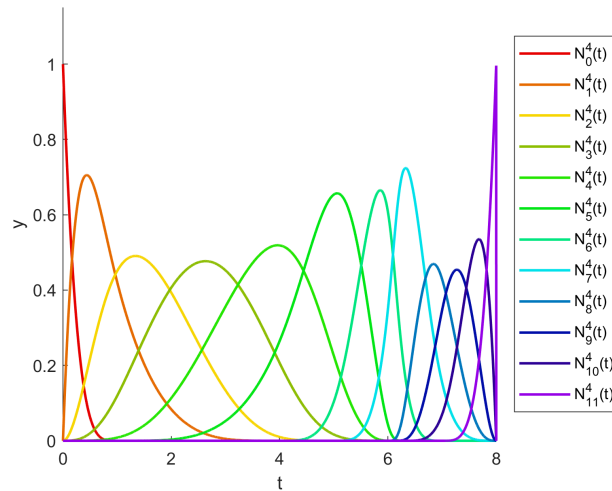


Abb. 2.12: Nicht uniforme B-Spline-Basisfunktionen vom Grad 4.

Abbildung 2.13 veranschaulicht, unter Ausnutzung eines nach Gleichung (2.14) aufgestellten nicht uniformen Knotenvektors mit Randvielfachheit $l = 5$, die erste Ableitung der in Abbildung 2.12 dargestellten B-Spline-Basisfunktionen vierten Grades. Mittels der farbigen Visualisierung ist eine Verknüpfung der jeweiligen i -ten B-Spline-Basisfunktionen mit der zugehörigen ersten Ableitung möglich.

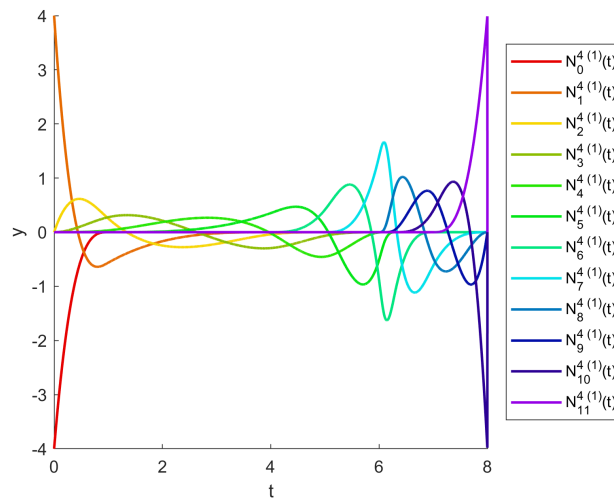


Abb. 2.13: Erste Ableitung nicht uniformer B-Spline-Basisfunktionen vom Grad 4.

Die Bestimmung der ersten und zweiten Ableitung der B-Spline-Basisfunktionen ist essentiell für die Generierung der Ableitungen von B-Spline-Kurven, insbesondere in Hinblick auf die Untersuchung der Krümmungsvektoren der Kurvenpunkte.

2.2 B-Spline-Kurven

In den folgenden Abschnitten werden zwei Arten von B-Spline-Kurven, sogenannte periodische und offene B-Spline-Kurven definiert. Eine Unterscheidung der zwei B-Spline-Kurventypen findet über deren B-Spline-Basisfunktionen, im Speziellen über deren Knotenvektoren statt, wobei hier entweder periodische Knotenvektoren für periodische B-Spline-Kurven oder nicht periodische bzw. offene Knotenvektoren für offene B-Spline-Kurven verwendet werden. Beide B-Spline-Kurventypen zeichnen sich durch eine Abhängigkeit der Knotenanzahl $(k + 1)$ vom Polynomgrad n sowie der Kontrollpunktanzahl $(m + 1)$ mittels der Formel

$$k = m + n + 1 \quad (2.21)$$

aus (Piegl und Tiller 1997).

2.2.1 Periodische B-Spline-Kurven

Eine periodische B-Spline-Kurve wird festgelegt durch

$$C(t) = \sum_{i=0}^m d_i \cdot N_i^n(t), \quad t \in [u_n, u_{k-n}], \quad (2.22)$$

wobei die vektorwertigen Koeffizienten d_i mit $i = 0, \dots, m$ (im $\mathbb{R}^2$ und $\mathbb{R}^3$) B-Spline-Kontrollpunkte bzw. de Boor-Punkte und $N_i^n(t)$ B-Spline-Basisfunktionen vom Grad n repräsentieren. $(m + 1)$ entspricht der Kontrollpunktanzahl. Die Verbindung der de Boor-Punkte zu einem Polygonzug wird de Boor-Polygon oder B-Spline-Kontrollpolygon genannt. $(k + 1)$ entspricht der Anzahl der Knoten im Knotenvektor $U = (u_0 \ u_1 \ \dots \ u_{k-1} \ u_k)$ (Rogers 2001).

Wird zur Generierung der B-Spline-Kurve ein periodischer Knotenvektor verwendet, bringt dies eine periodische B-Spline-Kurve hervor. Auf Basis der Eigenschaft von B-Spline-Basisfunktionen bezüglich der Teilung der Eins (siehe Gleichung (2.18)), resultiert eine Limitierung des Parameterintervalls zu $u_n \leq t \leq u_{k-n}$ je nach Polynomgrad n . Diese Einschränkung des Parameterbereichs äußert sich in einer fehlenden Interpolation der Endpunkte des Kontrollpolygons mit den Endpunkten der B-Spline-Kurve.

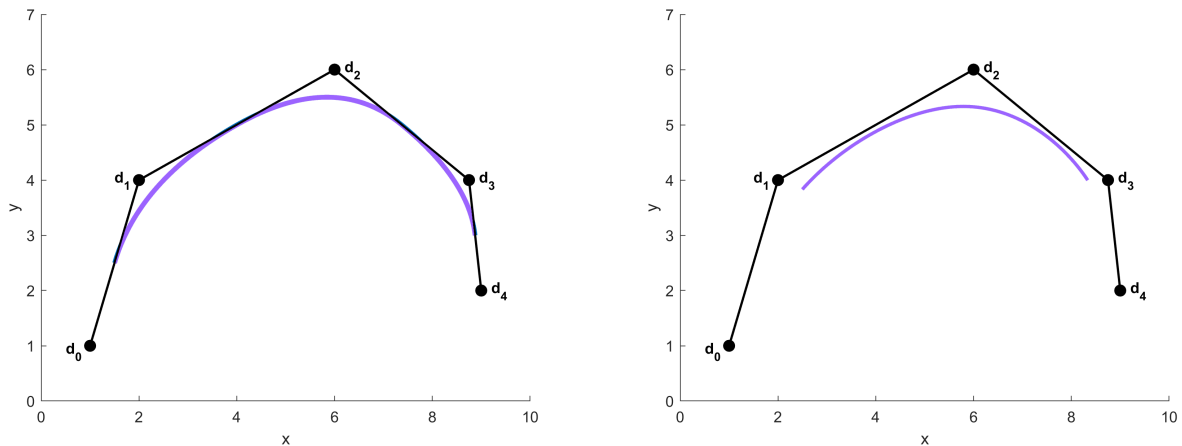


Abb. 2.14: Periodische uniforme B-Spline-Kurven vom Grad 2 (links) und Grad 3 (rechts).

Im Folgenden soll diese Verhaltensweise an einer B-Spline-Kurve zweiten und dritten Grades präsentiert werden. Hierfür wird auf Basis einer festgelegten Anzahl an Kontrollpunkten $(m + 1)$ und einem Kurvengrad n die notwendige Anzahl an Knotenwerten bestimmt. Bei $(m + 1) = 5$ Kontrollpunkten und einem Kurvengrad von $n = 2$, muss der Knotenvektor, auf Grund von $k = 4 + 2 + 1$, aus $k + 1 = 8$ Knoten aufgebaut werden. Für einen Knotenvektor

$$\mathbf{U} = \begin{pmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix} \quad (2.23)$$

wird das Parameterintervall dementsprechend durch $t \in [u_2, u_5] = [2, 5]$ definiert. Hieraus ergibt sich die in der linken Grafik von Abbildung 2.14 dargestellte periodische B-Spline-Kurve zweiten Grades.

Erhöht man den Kurvengrad auf $n = 3$, ist der Knotenvektor um einen Knoten zu vergrößern und der Parameterbereich auf $t \in [u_3, u_5] = [3, 5]$ weiter einzuschränken. Die resultierende periodische B-Spline-Kurve dritten Grades ist auf der rechten Seite von Abbildung 2.14 angeführt. Während periodische B-Spline-Kurven nullten und ersten Grades dem Kontrollpolygon entsprechen, werden die Ausmaße der B-Spline-Kurve mit zunehmendem Grad einer Reduzierung unterworfen, die von der Einschränkung des Parameterbereichs t herrührt. Diese Eigenschaft macht periodische B-Spline-Kurven für die Modellierung von Laserscanpunkten unattraktiv, da ein Verlauf der B-Spline-Kurve bis zu den Endpunkten des Kontrollpolygons von Vorteil ist.

Liegt die Zielsetzung auf einer Generierung geschlossener Formen, bietet sich die Anwendung periodischer B-Spline-Kurven hingegen in größerem Maße an, da mittels periodischer B-Spline-Basisfunktionen und dem Hinzufügen von n identischen Kontrollpunkten an das Ende des Kontrollpolygonvektors ein geschlossener Kurvenverlauf erwirkt werden kann. Abbildung 2.15 zeigt die identischen Kontrollpunkte der B-Spline-Kurven von Abbildung 2.14, deren Kontrollpolygon um drei Kontrollpunkte, in Form von $\mathbf{d} = (\mathbf{d}_0 \mathbf{d}_1 \mathbf{d}_2 \mathbf{d}_3 \mathbf{d}_4 \mathbf{d}_0 \mathbf{d}_1 \mathbf{d}_2)$, erweitert worden ist.

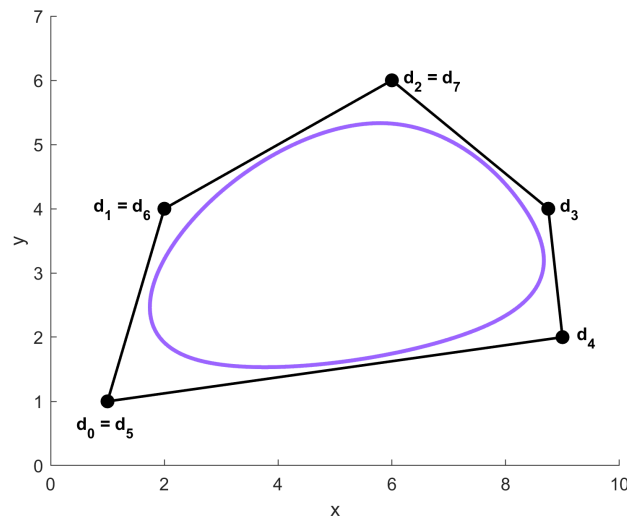


Abb. 2.15: Geschlossene periodische uniforme B-Spline-Kurven vom Grad 3.

Wie bereits erwähnt, eignen sich periodische B-Spline-Kurven nur in beschränktem Umfang für die Umsetzung der Modellierung von Laserscanpunktwolken zu Flächen. In Folge dessen soll eine weitere Vertiefung bezüglich offener B-Spline-Kurven auf Basis nicht periodischer B-Spline-Basisfunktionen durchgeführt werden.

2.2.2 Offene B-Spline-Kurven

Eine offene bzw. nicht periodische B-Spline-Kurve n -ten Grades wird definiert durch:

$$\mathbf{C}(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t), \quad t \in [u_0, u_k] = [u_n, u_{k-n}]. \quad (2.24)$$

Hierbei bezeichnet $N_i^n(t)$ nicht periodische B-Spline-Basisfunktionen und $\mathbf{d}_i$ mit $i = 0, \dots, m$ (im $\mathbb{R}^2$ und $\mathbb{R}^3$) Kontrollpunkte des Kontrollpolygons der B-Spline-Kurve. Vergleiche hierzu Abramowski und Müller (1991). Wie bereits in Abschnitt 2.1.1 behandelt, muss der Knotenvektor der offenen B-Spline-Basisfunktion die Randvielfachheit $(n + 1)$ erfüllen. Dies bewirkt eine Interpolation der Endpunkte der B-Spline-Kurve mit den Endpunkten des Kontrollpolygons. Aufgrund der benannten Randvielfachheit von $(n + 1)$ entspricht das Intervall $t \in [u_0, u_k]$ dem Intervall $t = [u_n, u_{k-n}]$. In Abbildung 2.16 werden offene B-Spline-Kurven zweiten und dritten Grades zu Vergleichszwecken mit dem identischen Kontrollpolygon der periodischen B-Spline-Kurven von Abbildung 2.14 dargestellt.

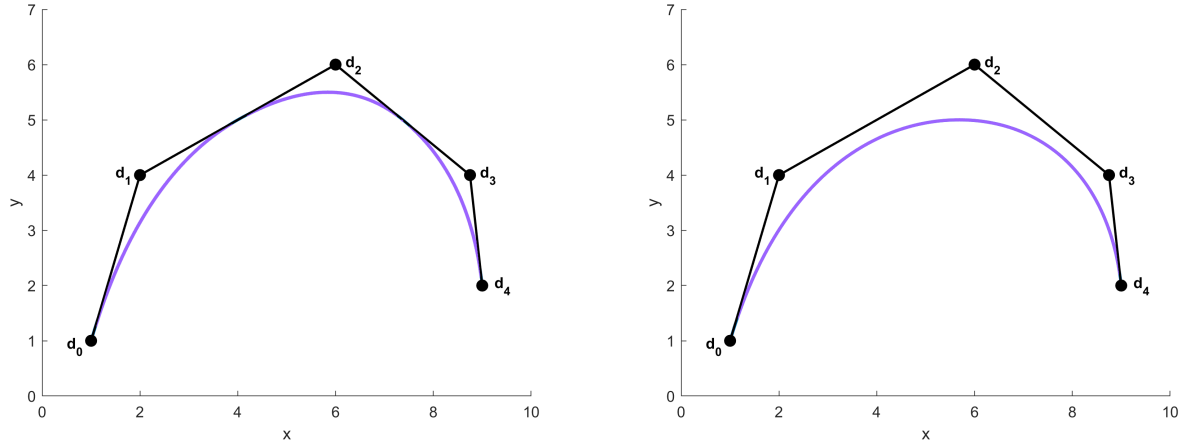


Abb. 2.16: Offene uniforme B-Spline-Kurven vom Grad 2 (links) und Grad 3 (rechts) (Aichinger und Schwieger 2022).

In Folge der fünf Kontrollpunkte und einem festgelegten Grad $n = 2$ muss der Knotenvektor, auf Basis der Formel $k = m + n + 1 = 4 + 2 + 1$, aus $k + 1 = 8$ Knoten und der Randvielfachheit $n + 1 = 3$ aufgebaut werden. Er wird in diesem Fall mittels $\mathbf{U} = (0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3)$ definiert. Für die nicht periodische B-Spline-Kurve dritten Grades ist ein zusätzlicher Knoten und die Randvielfachheit $l = 4$ notwendig. Hieraus resultiert der uniforme Knotenvektor $\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2)$.

Durch die Wahl eines nicht uniformen Knotenvektors, etwa $\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 0,5 \ 2 \ 2 \ 2 \ 2)$ oder $\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 1,5 \ 2 \ 2 \ 2 \ 2)$, kann die Form der B-Spline-Kurve gesteuert werden. Diese Auswirkungen sind in den Grafiken in Abbildung 2.17 einsehbar.

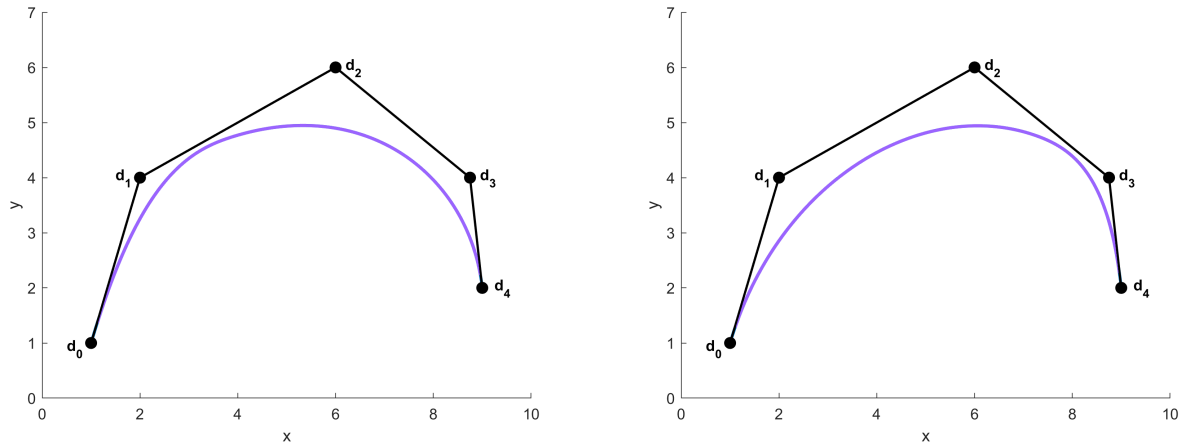


Abb. 2.17: Auswirkung der Veränderung eines Knotens eines Knotenvektors auf eine offene uniforme B-Spline-Kurve vom Grad 3.

Abgesehen von der Zusammensetzung des Knotenvektors, bestimmt der Grad n ebenfalls das Aussehen der Kurve. Er beeinflusst wie stark sich die B-Spline-Kurve an das Kontrollpolygon anschmiegt. In Abbildung 2.18 sind B-Spline-Kurven vom Grad 0 bis 5 mit einem Kontrollpolygon bestehend aus neun Kontrollpunkten dargestellt.

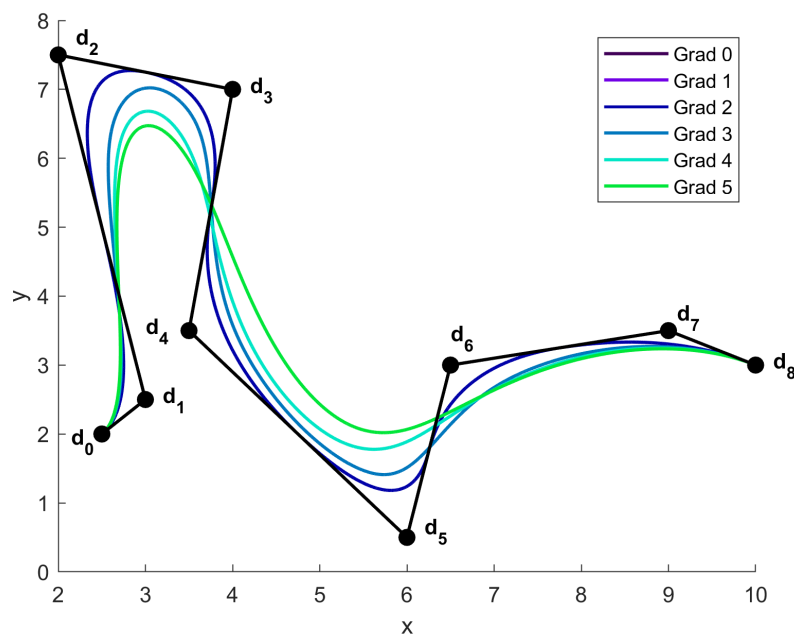


Abb. 2.18: Uniforme B-Spline-Kurven vom Grad 0 bis 5.

Je niedriger der Kurvengrad festgesetzt wird, desto stärker schmiegt sich die Kurve an das Kontrollpolygon. B-Splines vom Grad 0 und 1 entsprechen dem Kontrollpolygon. Nach Schumaker (2007) stellt der Polynomgrad 3 bei Spline Funktionen eine Art beste interpolierende Funktion dar. Die Ursache für das Verhalten des weniger starken Approximierens der Kurve in Verbindung mit einem zunehmenden Kurvengrad, hängt mit der konvexen Hülle-Eigenschaft zusammen, die im folgenden Unterkapitel 2.2.3 mit weiteren Eigenschaften der B-Spline Kurven vorgestellt wird.

2.2.3 Eigenschaften von B-Spline-Kurven

Lokalität

Um die Form einer B-Spline-Kurve lediglich an einer bestimmten Stelle anzupassen, ohne den gesamten Kurvenverlauf einer Änderung zu unterziehen, erweist sich die Lokaliäts-Eigenschaft der B-Spline-Technik als äußerst vorteilhaft. In Folge der Bedingung, dass die B-Spline-Basisfunktionen $N_i^n(t)$ mit dem Grad n und $i = 0, 1, \dots, k-1$ außerhalb des Knotenintervalls $t \in [u_i, u_{i+n+1})$ den Wert 0 annehmen, beeinflusst eine Kontrollpunktänderung ausschließlich die Kurvenpunkte $C(t)$, die innerhalb des Intervalls $t \in [u_i, u_{i+n+1})$ liegen (Abramowski und Müller 1991). Da das angegebene Intervall mit zunehmendem Polynomgrad größer ausfällt, vergrößert sich der Einflussbereich auf die Kurvenform, der durch Änderung eines Kontrollpunktes ausgelöst wird.

Abbildung 2.19 und 2.20 stellen die Auswirkungen einer Kontrollpunktänderung auf die Form einer B-Spline-Kurve im Speziellen dar. Die vier Grafiken zeigen das identische Kontrollpolygon, deren Kontrollpunkt d_3 in seiner Position verändert wird. Während die erste Grafik eine B-Spline-Kurve zweiten Grades veranschaulicht und der Einfluss der Punktänderung noch in geringem Maße ausfällt, kann bei der zweiten Grafik und einem Polynomgrad von 3 bereits eine Ausdehnung des Einflussbereichs festgestellt werden. Dieser lokale Wirkungsbereich nimmt bei den B-Spline-Kurven vierten und fünften Grades weiter zu bzw. die Lokaliät nimmt ab.

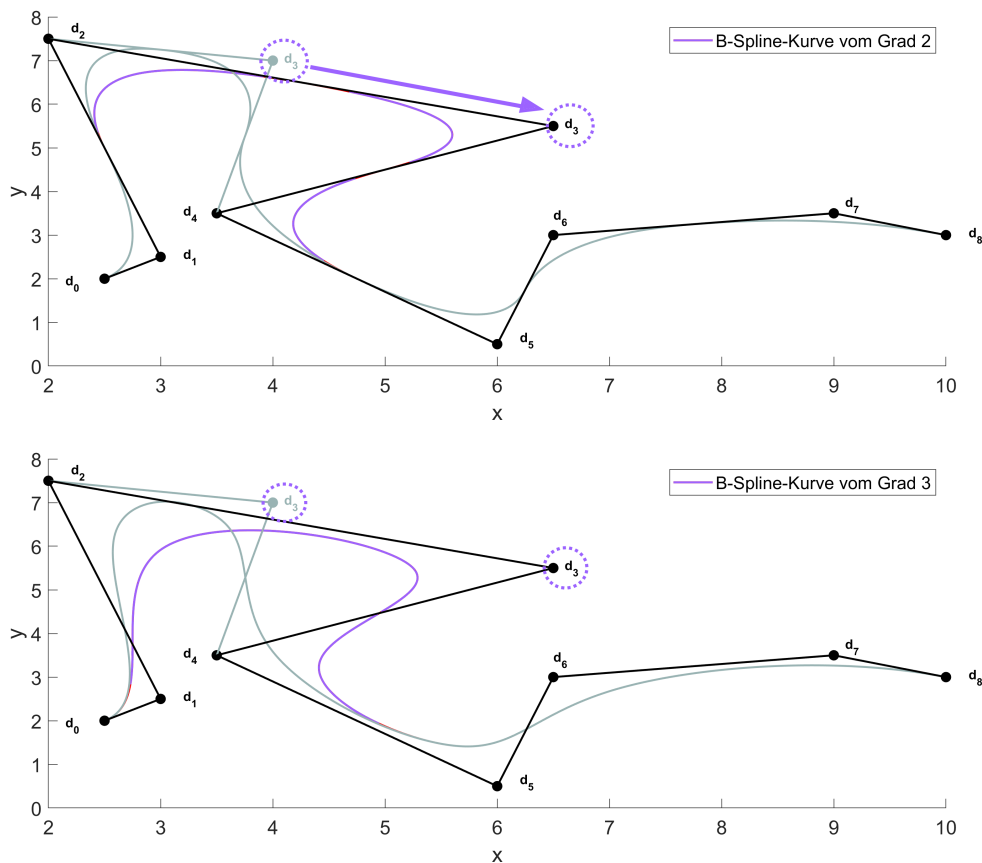


Abb. 2.19: Auswirkung einer Kontrollpunktänderung auf die Kurvenform einer B-Spline-Kurve vom Grad 2 (oben) und Grad 3 (unten).

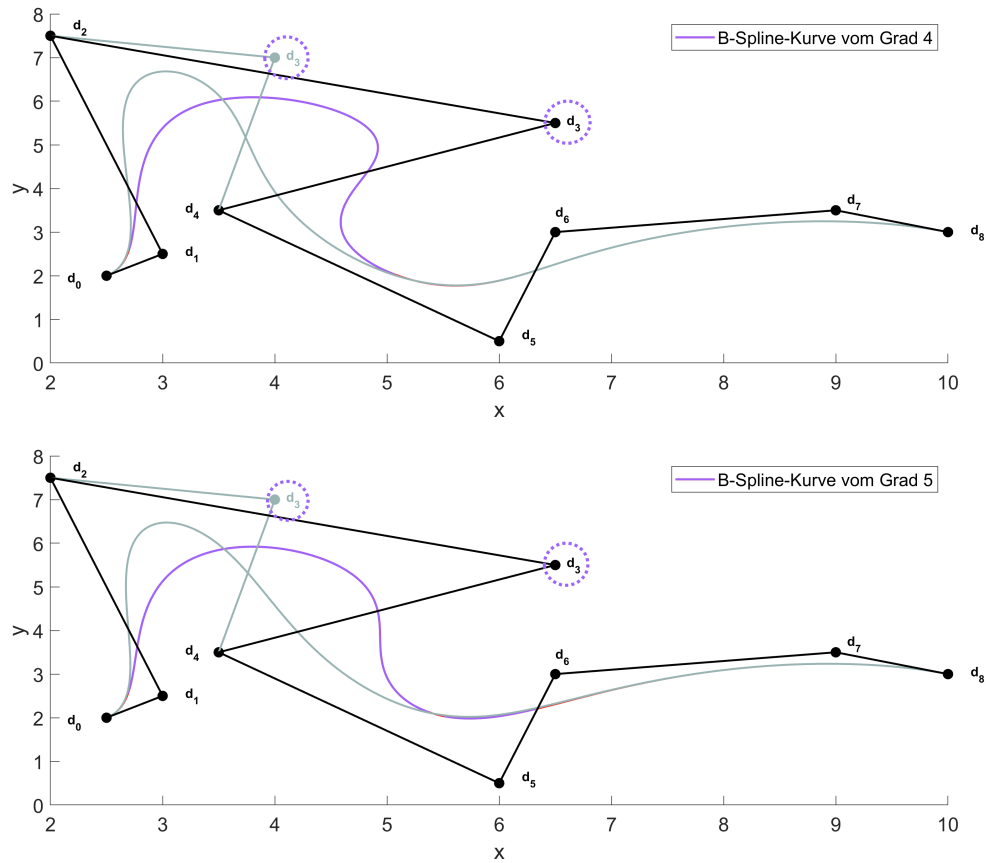


Abb. 2.20: Auswirkung einer Kontrollpunktänderung auf die Kurvenform einer B-Spline-Kurve vom Grad 4 (oben) und Grad 5 (unten).

Stetigkeitsbedingung

Mittels der Darstellung der B-Spline-Kurve aus den Kontrollpunkten und den B-Spline-Basisfunktionen erreicht man, dass die Differenzierbarkeits-Eigenschaft der B-Spline-Basisfunktionen auf die B-Spline-Kurve übertragen wird. Dementsprechend bedeutet dies bei einer B-Spline-Basisfunktion n -ten Grades über dem Knotenvektor $\mathbf{U}$ eine B-Spline-Kurve vom Grad n über $\mathbf{U}$. Die B-Spline-Kurve $\mathbf{C}(t)$ ist aufgrund dessen in den Knoten C^{n-1} stetig. Dies gilt für Knotenvektoren, deren innere Knoten die Vielfachheit $l = 1$ nicht übertreten. Somit haben beispielsweise kubische B-Spline-Kurven ($n = 3$) die Stetigkeit C^2 . Enthält der Knotenvektor einer B-Spline-Kurve einen inneren Knoten mit Vielfachheit l größer 1, lautet die Stetigkeit der Kurve in den Knoten C^{n-l} (Piegl und Tiller 1997).

Abbildung 2.21 und 2.22 zeigen eine B-Spline-Kurve vom Grad 3 mit steigender Vielfachheit eines inneren Knotens mit dem Wert 4. In der linken Grafik der Abbildung 2.21 weist die B-Spline-Kurve einen Knotenvektor $\mathbf{U} = (0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5)$ mit der Vielfachheit $l = 1$ sowie der notwendigen Randvielfachheit von $n + 1 = 4$ auf. Die Kurve ist somit C^2 -stetig. Die B-Spline-Kurve der rechten Grafik besitzt einen Knotenvektor von $\mathbf{U} = (0, 0, 0, 0, 1, 2, 4, 4, 5, 5, 5, 5)$ mit der Vielfachheit $l = 2$ und ist C^1 -stetig. Dementsprechend ist die erste Ableitung stetig, die zweite allerdings nicht.

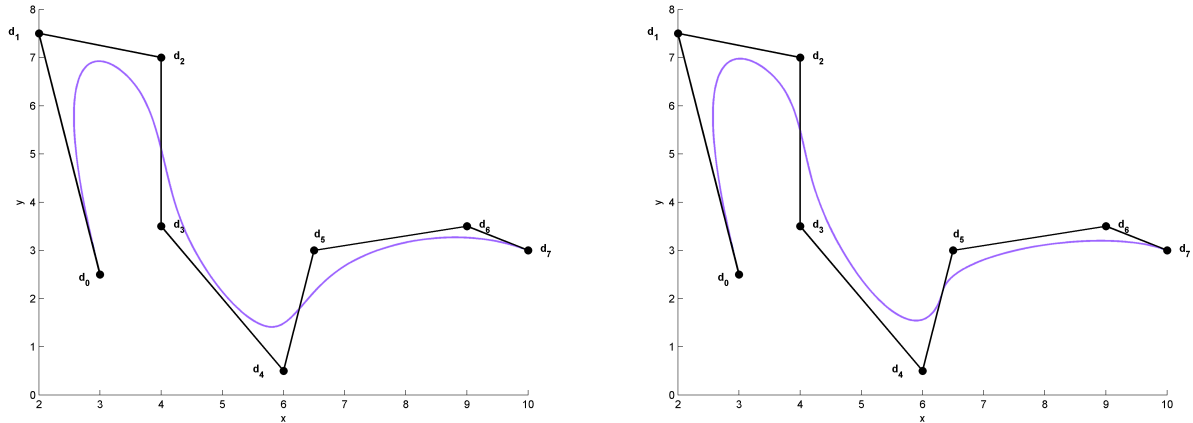


Abb. 2.21: Nicht uniforme B-Spline-Kurven vom Grad 3 mit Vielfachheit 1 (links) und Vielfachheit 2 (rechts).

Wird die linke B-Spline-Kurve von Abbildung 2.22 betrachtet, kann erkannt werden, dass die Kurve mit dem Kontrollpunkt d_4 zusammenfällt. Dies liegt an $U = (0, 0, 0, 0, 1, 4, 4, 4, 5, 5, 5, 5)$ mit der Vielfachheit $l = 3$ und der daraus resultierenden C^0 -Stetigkeit. Die Strecken des Kontrollpolygons zeigen die Tangentenrichtungen der Kurve in diesem Punkt an. Im rechten Teil der Abbildung 2.22 weist die B-Spline-Kurve einen Knotenvektor von $U = (0, 0, 0, 0, 4, 4, 4, 4, 5, 5, 5, 5)$ auf. Der innere Knoten mit dem Wert 4 hat eine Vielfachheit von $l = 4$. Die Unstetigkeit der Basisfunktionen des B-Splines äußert sich durch das Zerfallen der Kurve in zwei Teilkurven.

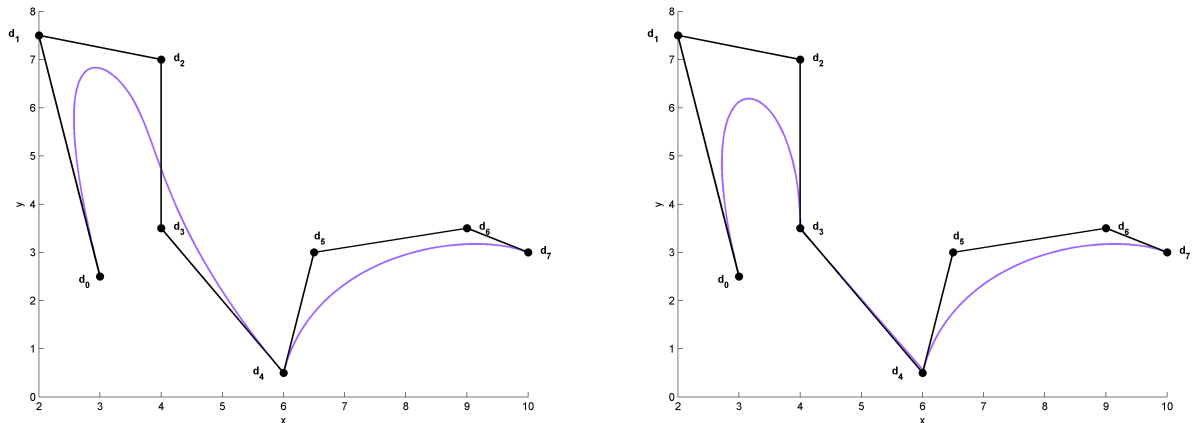


Abb. 2.22: Nicht uniforme B-Spline-Kurven vom Grad 3 mit Vielfachheit 3 (links) und Vielfachheit 4 (rechts).

Affine Invarianz

Wird eine affine Transformation auf eine B-Spline-Kurve angewandt, entspricht dies der Anwendung auf ihre Kontrollpunkte. Folglich ist die transformierte Kurve identisch mit der Kurve, die sich aus den transformierten Kontrollpunkten bestimmen lässt (Rogers 2001).

Gradunabhängigkeit

Im Gegensatz zu etwa Bézier-Kurven bieten B-Spline-Kurven den Vorteil, dass sich komplexe Kurvenformen mit einer Vielzahl an Kontrollpunkten nicht in einem hohen Kurvengrad n niederschlagen. Mittels der fundamentalen Formel von Gleichung (2.21) mit $k = m + n + 1$ kann eine erhöhte Kontrollpunktanzahl durch eine Vergrößerung des Knotenvektors kompensiert werden. Auf diese Weise gehen viele Kontrollpunkte nicht mit einer hohen Rechenleistung einher und es kann mit einem geringen Polynomgrad gearbeitet werden.

Die Beibehaltung eines niedrigen Kurvengrades erlaubt den Erhalt der besonders für Modellierungsarbeiten vorteilhaften Lokaltäts-Eigenschaft, die sich mit zunehmendem Polynomgrad auf eine immer größer werdende Anzahl an Kontrollpunkten ausdehnt. Denn je höher die Festlegung des Kurvengrades ausfällt, um so weitreichender wirkt sich der Einfluss einer Kontrollpunktänderung auf die Kurvenpunkte einer B-Spline-Kurve aus.

Anfangs- und Endpunkt-Eigenschaft

In Folge der Wahl eines offenen Knotenvektors mit einer Randvielfachheit von $n + 1$ werden offene B-Spline-Kurven gebildet, deren Anfangs- sowie Endpunkt mit dem Anfangs- und Endpunkt des Kontrollpolygons zusammenfallen. Es gilt:

$$\mathbf{C}(u_0) = \mathbf{d}_0 \quad \text{und} \quad \mathbf{C}(u_k) = \mathbf{d}_m \quad (2.25)$$

Dementsprechend sind der Anfangs- und Endpunkt des Kontrollpolygons mit dem Wert 1 gewichtet. Wie stark der Einfluss eines einzelnen Kontrollpunktes auf den Kurvenpunkt $\mathbf{C}(t)$ ausfällt, hängt von der Höhe seines Gewichts $N_i^n(t)$ zum Parameterwert t ab.

Abbildung 2.23 kann entnommen werden, dass im Parameterwert $t = 0$ lediglich die B-Spline-Basisfunktion $N_0^2(t)$ ungleich 0, im Speziellen dem Wert 1 entspricht. Somit dient ausschließlich der Kontrollpunkt $\mathbf{d}_0$ mit dem Gewicht 1 zur Berechnung des Kurvenpunktes bei $t = 0$, wodurch B-Spline-Kurve und Kontrollpolygon zusammenfallen.

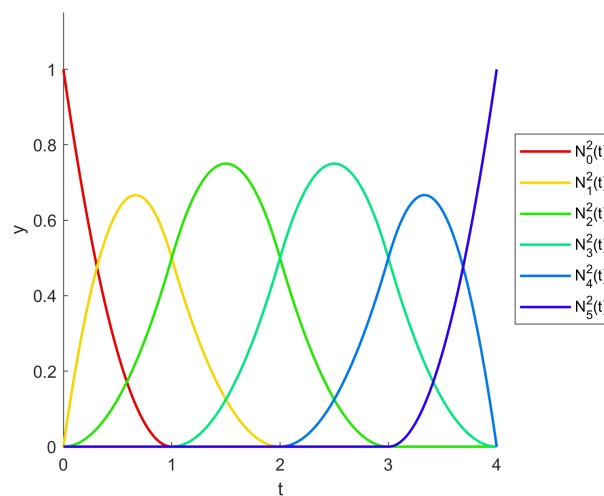


Abb. 2.23: B-Spline-Basisfunktionen vom Grad 2.

Der gleiche Sachverhalt ereignet sich im Parameterwert $t = u_k = 4$. $N_5^2(t)$ nimmt den Wert 1 an, wodurch lediglich $\mathbf{d}_m$ einen Beitrag zur Berechnung des letzten Kurvenpunktes der B-Spline-Kurve liefert. Vergleiche hierzu Bender und Bill (2006).

Konvexe Hülle

Zur Berechnung eines B-Spline-Kurvenpunktes $\mathbf{C}(t)$ bei einem bestimmten Parameterwert t setzt sich die Gleichung

$$\sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t) = \mathbf{d}_0 \cdot N_0^n(t) + \mathbf{d}_1 \cdot N_1^n(t) + \mathbf{d}_2 \cdot N_2^n(t) + \cdots + \mathbf{d}_m \cdot N_m^n(t) \quad (2.26)$$

aus den Vektoren der B-Spline-Basisfunktionen $N_i^n(t)$ sowie den Kontrollpunkten $\mathbf{d}_i$ im reellen Vektorraum mit $i = 0, \dots, m$ als Linearkombination zusammen, wobei $m + 1$ der Anzahl der Kontrollpunkte entspricht. Erfüllen die Vektoren einer Linearkombination sowohl die Bedingung der Nicht-Negativität sowie eine Aufsummierung zu 1 (Teilung der Eins), erhält man mit Gleichung (2.26) eine Konvexkombination. Da diese Nicht-Negativitäts-Eigenschaft und Teilung der Eins auf B-Spline-Basisfunktionen zutrifft (siehe Abschnitt 2.1.2), ergibt sich jeder Kurvenpunkt $\mathbf{C}(t)$ einer B-Spline-Kurve zu jedem Parameterwert t durch eine Konvexkombination aller Kontrollpunkte $\mathbf{d}_i$ und der B-Spline-Basisfunktionen $N_i^n(t)$.

In Folge der Lokali  ts-Eigenschaft von B-Spline-Basisfunktionen verfeinert sich diese Aussage dahingehend, dass die konvexe H  lle eines Kurvenpunktes nicht durch die Konvexkombination aller Kontrollpunkte resultiert, sondern sich f  r einen festen Parameterwert t , bei insgesamt $m + 1$ B-Spline-Basisfunktionen, h  chstens aus $n + 1$ benachbarten Kontrollpunkten zusammensetzt, da lediglich $n + 1$ B-Spline-Basisfunktionen einen Wert ungleich 0 erf  llen. Die Erzeugung der Vereinigungsmenge aller konvexen H  llen eines Kontrollpolygons bei einem Kurvengrad von n liefert die Menge, in der die B-Spline-Kurve mit Grad n liegen muss (Bender und Bill 2006).

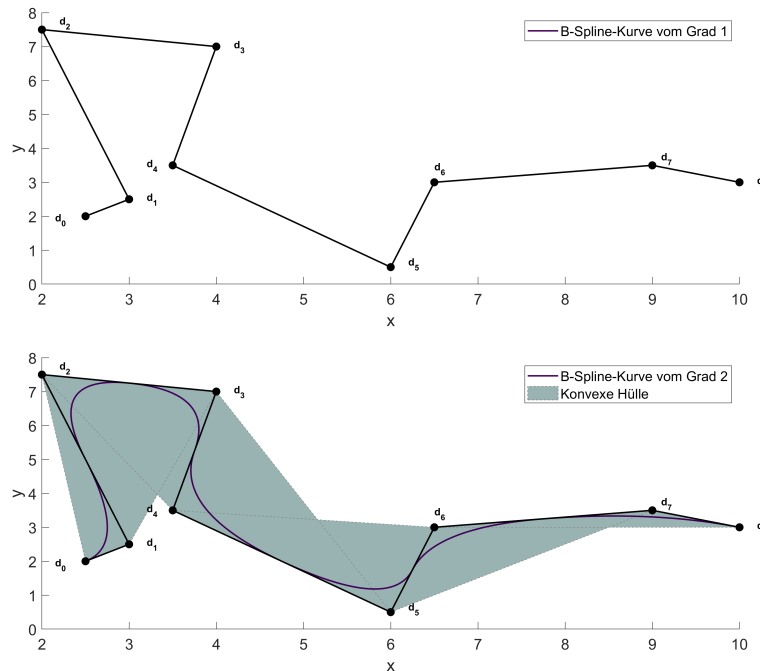


Abb. 2.24: Konvexe H  lle von B-Spline-Kurven vom Grad 1 (oben) und Grad 2 (unten).

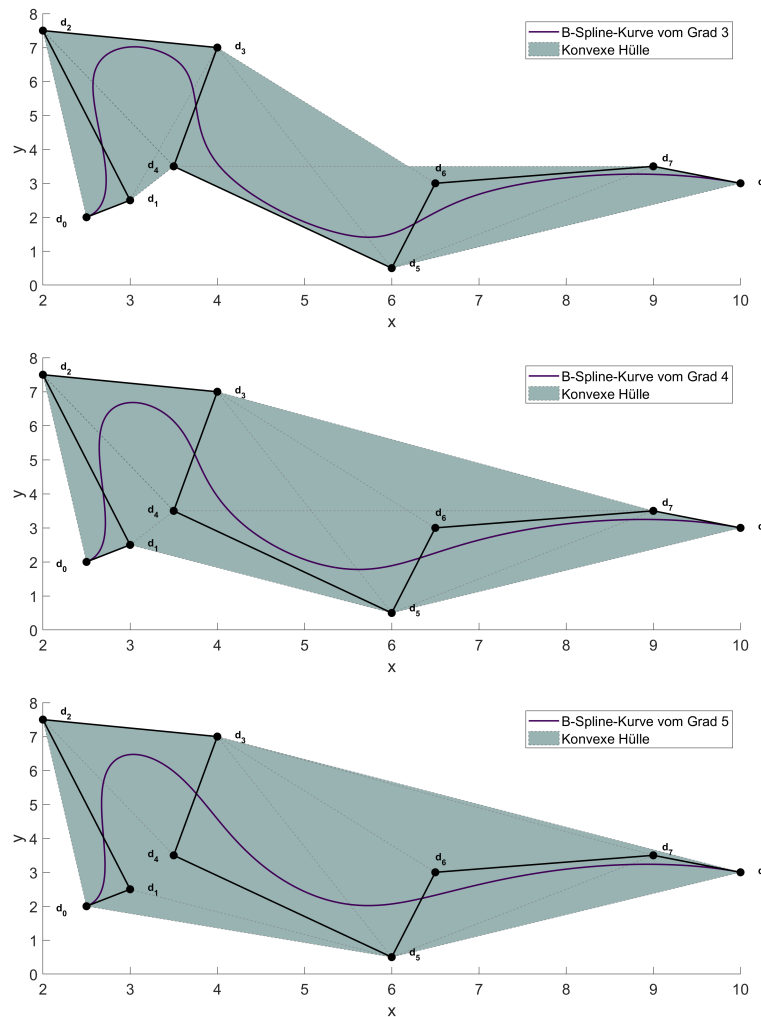


Abb. 2.25: Konvexe Hülle von B-Spline-Kurven vom Grad 3 (oben), Grad 4 (mitte) und Grad 5 (unten).

Die Abbildungen 2.24 und 2.25 stellen zusammen fünf Grafiken mit dem gleichen Kontrollpolygon dar. Die abgebildeten B-Spline-Kurven weisen einen unterschiedlichen Kurvengrad auf. Während die erste Grafik eine B-Spline-Kurve vom Grad 1 zeigt, steigt der Kurvengrad bis zur letzten Grafik bis zum Polynomgrad 5 an. Alle Abbildungen enthalten die jeweilige konvexe Hülle der B-Spline-Kurve, die sich aus den einzelnen konvexen Hüllen der Kurvenpunkte aufbaut. In der ersten Grafik entspricht die konvexe Hülle dem Kontrollpolygon, da die einzelnen konvexen Hüllen bei einem Grad $n = 1$ aus lediglich $n + 1 = 2$ Kontrollpunkten bestehen. In der zweiten Grafik mit einem Kurvengrad von 2 werden die konvexen Hüllen aus 3 benachbarten Kontrollpunkten erzeugt. Je höher der Kurvengrad gewählt wird, desto größer ist die Anzahl der Kontrollpunkte, die eine konvexe Hülle eines Kurvenpunktes aufspannen. So setzt sich eine konvexe Hülle bei der fünften Grafik und dem Polynomgrad $n = 5$ bereits aus 6 Kontrollpunkten zusammen. Die Vereinigungsmenge der konvexen Hüllen aller Kontrollpunkte zeigt somit eine deutlich größere Menge, als etwa im Falle des Kurvengrades 2. Diese Darstellung verdeutlicht zum einen die Lokaltäts-Eigenschaft der B-Spline-Basisfunktionen, die eine Erhöhung der Anzahl der beeinflussenden Kontrollpunkte auf die B-Spline-Kurve bei zunehmendem Kurvengrad bedeutet, zum anderen erklärt sie das Anschmiegen der Kurve an das Kontrollpolygon je niedriger der Polynomgrad gewählt wird.

Ableitungen von B-Spline-Kurven

Zur Bestimmung der Ableitung einer B-Spline-Kurve ist unmittelbar die Ableitung von B-Spline-Basisfunktionen notwendig, deren Berechnung in Gleichung (2.20) Abschnitt 2.1.2 aufgeführt wird. Demnach lautet die b -te Ableitung einer B-Spline-Kurve nach Piegel und Tiller (1997):

$$\mathbf{C}^{(b)}(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^{n(b)}(t), \quad (2.27)$$

wodurch die b -te Ableitung einer B-Spline-Kurve n -ten Grades mit Hilfe der b -ten Ableitung von $(m+1)$ B-Spline-Basisfunktionen vom Grad n sowie den zugehörigen $(m+1)$ Kontrollpunkten resultiert. Die Berechnung der ersten Ableitung einer B-Spline-Kurve ergibt den Tangentenvektor jedes einzelnen Kurvenpunktes:

$$\mathbf{C}'(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i'(t). \quad (2.28)$$

Die zweite Ableitung stellt hingegen den Krümmungsvektor für jeden B-Spline-Kurvenpunkt dar:

$$\mathbf{C}''(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i''(t). \quad (2.29)$$

Abbildung 2.26 präsentiert eine B-Spline-Kurve dritten Grades, sowie die Tangenten- und Krümmungsvektoren für zwölf Kurvenpunkte der B-Spline Kurve.

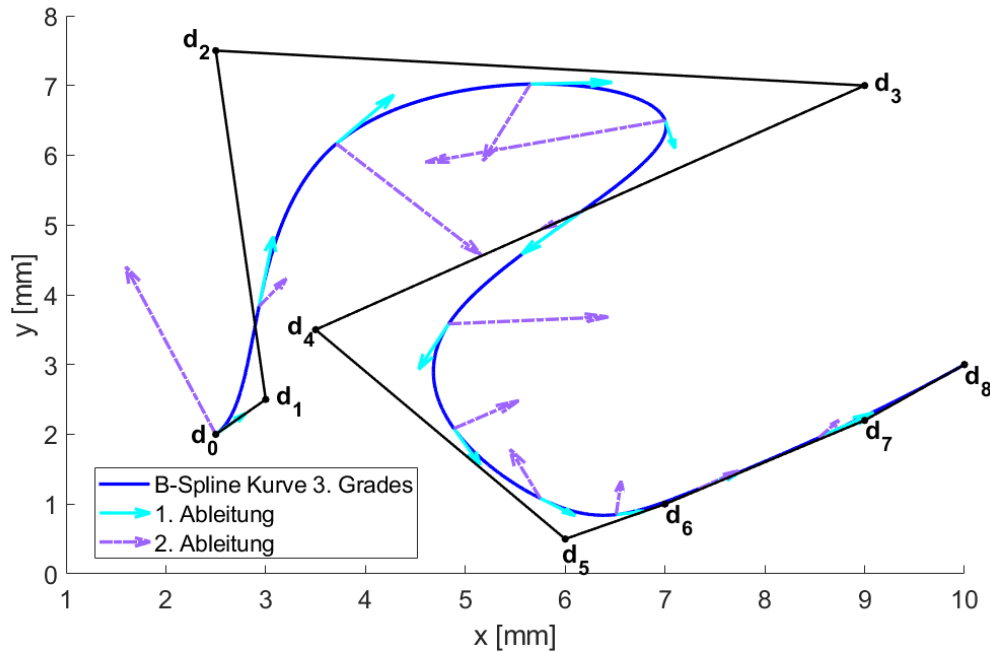


Abb. 2.26: Erste und zweite Ableitung einer B-Spline Kurve dritten Grades in Form der Tangenten- und Krümmungsvektoren für zwölf Kurvenpunkte der B-Spline-Kurve.

Durch die Berechnung des Betrags der zweiten Ableitung der B-Spline-Kurve, erhält man die Krümmung jedes Kurvenpunktes (Merziger und Wirth 2010):

$$|C''(t)| = \kappa(t) \quad (2.30)$$

In Abbildung 2.27 wird die Krümmung jedes Kurvenpunktes mittels einer magentafarbenen Schraffur dargestellt. Je geringer die Richtungsänderung der B-Spline-Kurve ausfällt, desto kleiner ist die Krümmung. Dies kann besonders am Kurvenverlauf entlang der Kontrollpunkte d_6 , d_7 und d_8 beobachtet werden, entlang derer die Kurve eine geringe Krümmung vorweist. Die starke Auslagerung des Kontrollpunktes d_3 in x-Richtung erzeugt eine B-Spline-Kurve, deren Form im lokalen Einflussbereich von d_3 deutlich enger verläuft und somit zu einem ausgeprägten Krümmungswert führt.

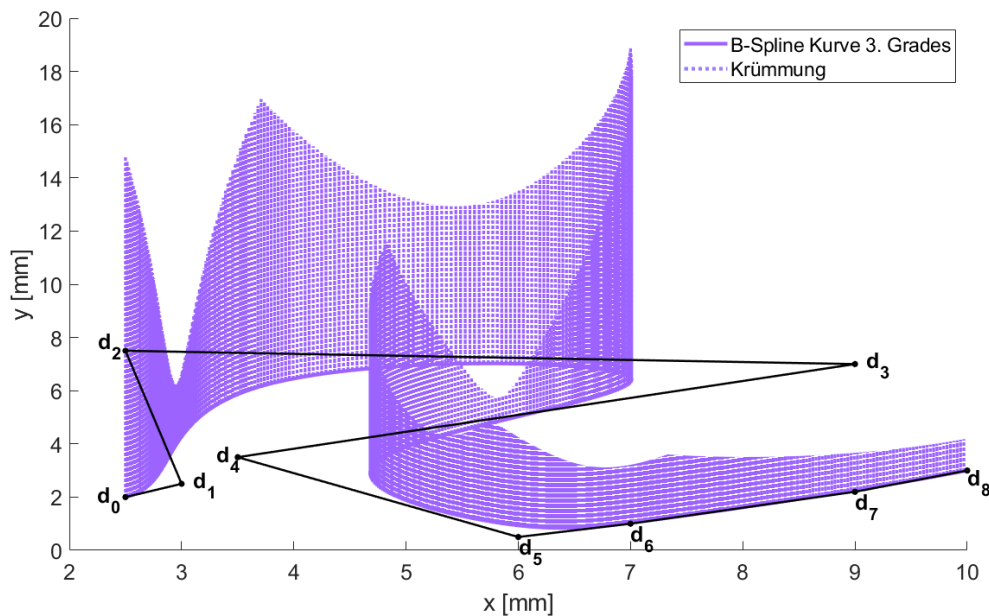


Abb. 2.27: Krümmung einer B-Spline Kurve vom Grad 3 (Aichinger und Schwieger 2022).

2.3 B-Spline-Flächen

Nachdem bisher B-Spline-Kurven in Verbindung mit ihren Eigenschaften behandelt wurden, soll sich diese Thematik auf Flächen ausweiten. Um Flächen mittels einer vektorbasierten Funktion $S(t_1, t_2)$ auf Basis zweier Parameter t_1 und t_2 abbilden zu können, bietet die Literatur diverse Schemata an (siehe hierzu Piegler (1989)). Hoschek und Lasser (1992) stellen etwa eine Bestimmung von Flächen mit dreieckigem Parametergebiet oder auch mit Hilfe einer transfiniten Methode vor. Eine der einfachsten und in geometrischen Modellierungsanwendungen überwiegend genutzte Methode wird Tensorprodukt-Schema genannt. Die Tensorprodukt-Methode ist ein bidirektionales Kurvenschema mit viereckigem Parametergebiet (Piegler und Tiller 1997).

Eine Tensorprodukt B-Spline-Fläche kann mittels einer B-Spline-Kurve $C(t_1)$ erzeugt werden, die entlang eines Parameters t_2 durch den Raum bewegt wird und hierbei Formänderungen ausgesetzt ist. Auf diese Weise entsteht ein bidirektionales Netz, das eine Rechteckstruktur vorweist und in jeder Reihe aus der gleichen Anzahl an Kontrollpunkten besteht.

Mathematisch lässt sich eine offene B-Spline-Fläche vom Grad (n, e) aus den Kontrollpunkten $\mathbf{d}_{i,j}$ eines polygonalen Kontrollnetzes, zweier B-Spline-Basisfunktionen $N_i^n(t_1)$ und $N_j^e(t_2)$ und zweier offener Knotenvektoren $\mathbf{U}$ und $\mathbf{V}$ wie folgt definieren (Abramowski und Müller 1991):

$$\mathbf{S}(t_1, t_2) = \sum_{i=0}^m \sum_{j=0}^a N_i^n(t_1) \cdot N_j^e(t_2) \cdot \mathbf{d}_{i,j}, \quad (t_1, t_2) \in [u_0, u_k] \times [v_0, v_r] \quad (2.31)$$

mit

$$\mathbf{U} = \left(\underbrace{u_0 = u_1 = \dots = u_n}_{n+1 \text{ mal}} \ u_{n+1} \ \dots \ u_{k-n-1} \ \underbrace{u_{k-n} = u_{k-n+1} = \dots = u_k}_{n+1 \text{ mal}} \right) \quad (2.32)$$

und der Bedingung

$$k = m + n + 1, \quad (2.33)$$

bei einer Knotenanzahl von $(k + 1)$, einer Kontrollpunktanzahl von $(m + 1)$ und einem Kurvengrad n sowie

$$\mathbf{V} = \left(\underbrace{v_0 = v_1 = \dots = v_e}_{e+1 \text{ mal}} \ v_{e+1} \ \dots \ v_{r-e-1} \ \underbrace{v_{r-e} = v_{r-e+1} = \dots = v_r}_{e+1 \text{ mal}} \right) \quad (2.34)$$

mit der Bedingung

$$r = a + e + 1, \quad (2.35)$$

bei einer Knotenanzahl von $(r + 1)$, einer Kontrollpunktanzahl von $(a + 1)$ und einem Kurvengrad e . Die Verwendung offener Knotenvektoren mit einer Randvielfachheit von $(n + 1)$ und $(e + 1)$ unterliegt ähnlicher Gründe wie im Kurvenfall und wird bei Aufführung der Eigenschaften von B-Spline-Kurven in Unterkapitel 2.3.1 näher dargelegt.

Abbildung 2.28 zeigt eine B-Spline-Fläche von Grad $(3, 3)$, die mittels eines bidirektionalen Kontrollnetzes aus 6×5 Kontrollpunkten aufgespannt wird. Die Parameter der Fläche sind zur besseren Nachvollziehbarkeit im Folgenden in Tabelle 2.1 aufgelistet:

	Parameterbereich t_1	Parameterbereich t_2
Kurvengrad	$n = 3$	$e = 3$
Kontrollpunktanzahl	$m + 1 = 6$	$a + 1 = 5$
Fundamentalgleichung	$k = m + n + 1 = 9$	$r = a + e + 1 = 8$
Knotenanzahl	$k + 1 = 10$	$r + 1 = 9$
Knotenvektor	$\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3 \ 3)$	$\mathbf{V} = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2)$

Tab. 2.1: Parameter der B-Spline-Fläche von Abbildung 2.28

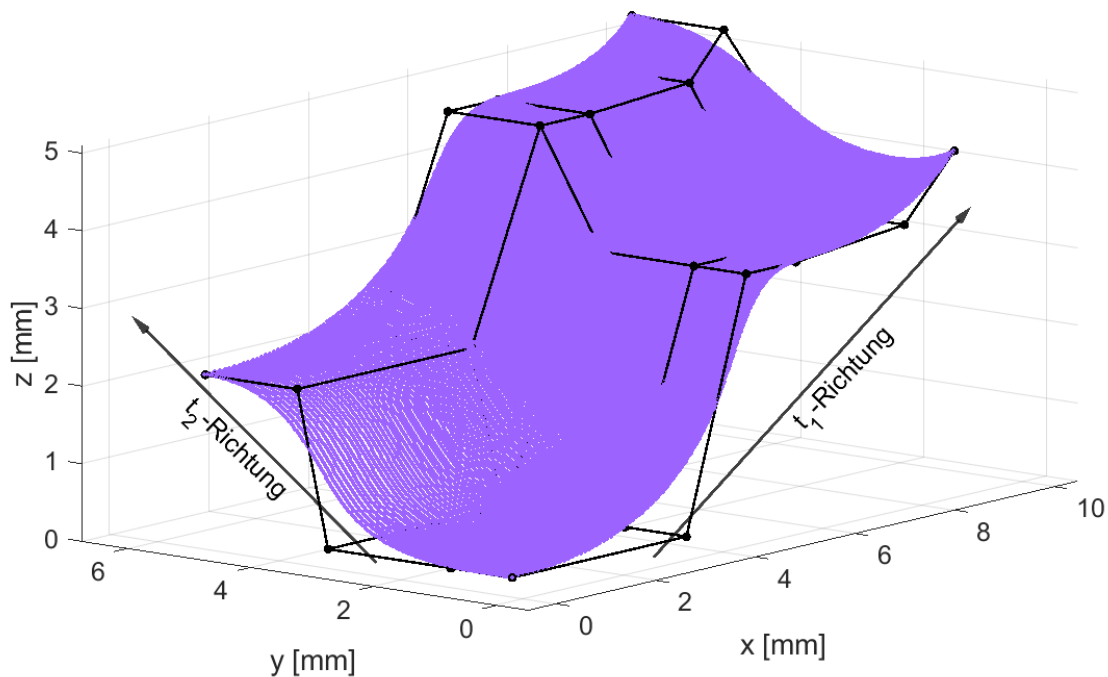


Abb. 2.28: Kontrollgitter und B-Spline-Fläche vom Grad (3,3) (Aichinger und Schwieger 2022).

2.3.1 Eigenschaften von B-Spline-Flächen

Für B-Spline-Flächen gelten diverse Eigenschaften, die durch eine Übertragung der B-Spline-Basisfunktionen oder B-Spline-Kurven vererbt werden. Hierzu zählen zum Beispiel die affine Invarianz oder die konvexe Hülle-Eigenschaft (Rogers 2001). Im weiteren Fortgang sollen weitere wichtige Eigenschaften vorgestellt werden.

Eckpunkt-Eigenschaft

Werden die Knotenvektoren $\mathbf{U}$ und $\mathbf{V}$ der B-Spline-Fläche vom Grad (n, e) derart aufgestellt, dass die Randvielfachheit von $(n + 1)$ sowie $(e + 1)$ eingehalten wird, kommt die Eigenschaft von offenen B-Spline-Kurven zum Tragen, die in diesem Fall gewährleistet, dass die B-Spline-Fläche mit ihren Eckkontrollpunkten zusammenfällt (Bender und Bill 2006).

Lokalität

Die Eigenschaft von B-Spline-Kurven durch Kontrollpunktänderung lediglich in einem bestimmten Bereich von Formänderungen beeinflusst zu werden, überträgt sich ebenso auf B-Spline-Flächen. Wird ein Kontrollpunkt $\mathbf{d}_{i,j}$ verschoben, beeinflusst dies die Fläche nur für den Parameterbereich $(t_1, t_2) \in [u_i, u_{i+n+1}) \times [v_j, v_{j+e+1})$ (Piegl und Tiller 1997).

Exemplarisch wird dies an Abbildung 2.29 gezeigt, in der eine B-Spline-Fläche vom Grad (3,3) auf Basis eines ebenen Kontrollgitters mit zwei aus der Ebene heraus verschobenen Punkten $\mathbf{d}_{2,2}$ und $\mathbf{d}_{7,3}$ dargestellt wird. Für eine bessere Visualisierung sind Kontrollnetz und B-Spline-Fläche um einen z-Wert von 3 versetzt aufgeführt.

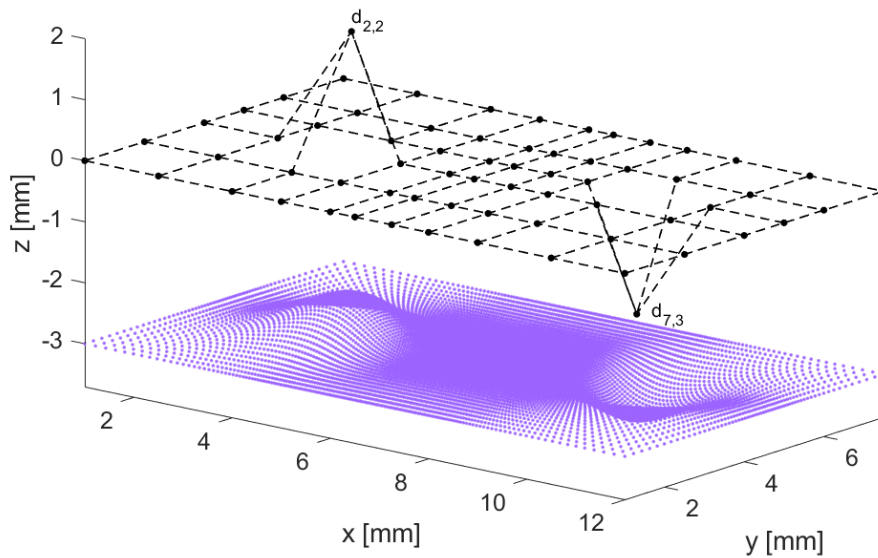


Abb. 2.29: Ebene B-Spline-Fläche vom Grad (3,3) mit zwei extrapolierten Kontrollpunkten.

In Abbildung 2.30 wird die B-Spline-Fläche von Abbildung 2.29 von oben präsentiert. Die hellblauen Farbareale verdeutlichen die Differenz der B-Spline-Fläche in Relation zur resultierenden B-Spline-Fläche eines planaren Kontrollgitters. Die magentafarbenen Bereiche entsprechen Flächenpunkten, die sowohl im planaren Kontrollgitter als auch im durch die verschobenen Punkte $d_{2,2}$ und $d_{7,3}$ geänderten Kontrollnetz eine identische Lage aufweisen.

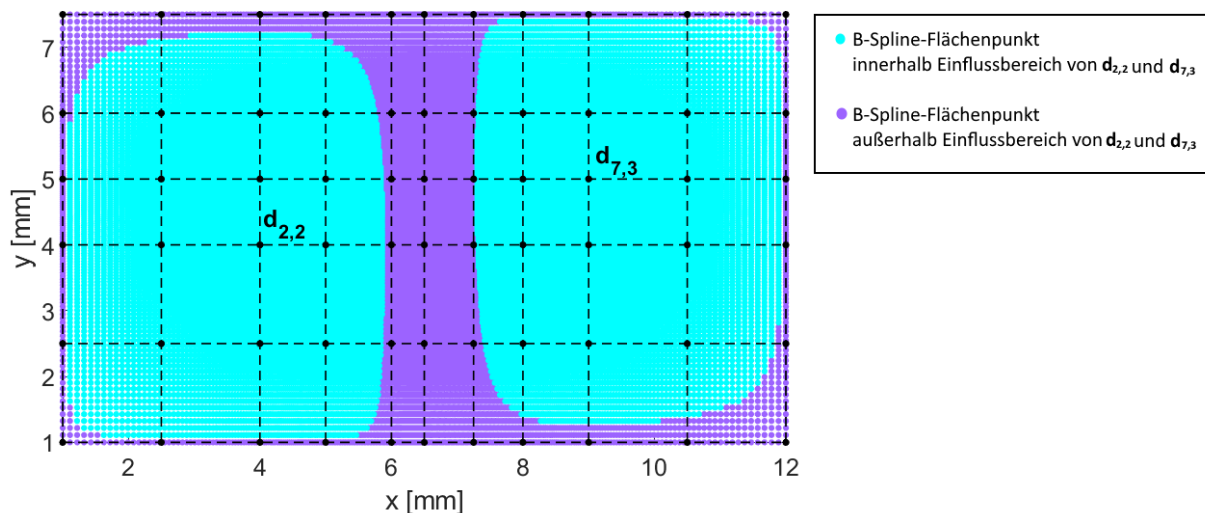


Abb. 2.30: Einflussbereich der Kontrollpunktänderung auf eine ebene B-Spline-Fläche vom Grad (3,3).

Ableitung von B-Spline-Flächen

Auch bei Flächen beschreibt die erste Ableitung die Tangente einer Fläche und legt somit die Steigung der Fläche in einem Punkt fest. Zur Generierung der Ableitung einer B-Spline-Fläche ist allerdings zu beachten, dass eine B-Spline-Fläche zwei Parameter t_1 und t_2 besitzt und in Folge dessen zwei partielle Ableitungen gebildet werden müssen. Der Vorgang der Bestimmung der partiellen Ableitung ähnelt dem der Ableitung von Kurven. Um eine B-Spline-Fläche differenzieren zu können, ist demnach auf die Ableitungen der B-Spline-Basisfunktionen zurückzugreifen.

Nach Piegl und Tiller (1997) lautet die allgemeine Definition für eine $(b_1 + b_2)$ -te partielle Ableitung der B-Spline-Fläche $\mathbf{S}(t_1, t_2)$:

$$\frac{\partial^{(b_1+b_2)}}{\partial^{(b_1)}t_1 \partial^{(b_2)}t_2} \mathbf{S}(t_1, t_2) = \sum_{i=0}^m \sum_{j=0}^a N_i^{n \ (b_1)}(t_1) \cdot N_j^{e \ (b_2)}(t_2) \cdot \mathbf{d}_{i,j}, \quad (t_1, t_2) \in [u_0, u_k] \times [v_0, v_r], \quad (2.36)$$

wobei die B-Spline-Fläche b_1 -mal nach t_1 und b_2 -mal nach t_2 differenziert ist.

Für die Berechnung der Ableitung der B-Spline-Basisfunktionen wird auf die Gleichung (2.20) von Abschnitt 2.1.2 verwiesen:

$$N_i^{n \ (b)}(t) = \frac{n}{u_{i+n} - u_i} \cdot N_i^{n-1 \ (b-1)}(t) - \frac{n}{u_{i+n+1} - u_{i+1}} \cdot N_{i+1}^{n-1 \ (b-1)}(t). \quad (2.20)$$

Um zum Beispiel die Tangentenvektoren einer B-Spline-Fläche in einem Punkt zu ermitteln, sind die erste partielle Ableitung nach t_1 sowie nach t_2 zu erzeugen:

$$\frac{\partial^{(1)}}{\partial^{(1)}t_1} \mathbf{S}(t_1, t_2) = \sum_{i=0}^m \sum_{j=0}^a N_i^{n \ (1)}(t_1) \cdot N_j^e(t_2) \cdot \mathbf{d}_{i,j}, \quad (2.37)$$

$$\frac{\partial^{(1)}}{\partial^{(1)}t_2} \mathbf{S}(t_1, t_2) = \sum_{i=0}^m \sum_{j=0}^a N_i^n(t_1) \cdot N_j^{e \ (1)}(t_2) \cdot \mathbf{d}_{i,j}. \quad (2.38)$$

Krümmung von B-Spline-Flächen

Wie bereits im vorherigen Abschnitt gezeigt, besitzt eine B-Spline-Fläche zwei partielle Ableitungen, wodurch sie zwei Tangentenvektoren aufweist, die orthogonal aufeinander stehen. Die zu einer Ebene aufgespannten Tangentenvektoren werden Tangentialebene genannt. Um die Fläche bezüglich ihrer inneren Geometrie überprüfen zu können, ist von Bedeutung, dass in jedem Punkt der Fläche eine Tangentialebene existiert. Dies ist der Fall, wenn es sich um eine reguläre Fläche handelt, die über lineare Unabhängigkeit der Tangentenvektoren definiert ist (Carmo 1993). Auf diese Weise können ebenso Selbstüberschneidungen oder Berührungen innerhalb der Fläche vermieden werden. Zur Bestimmung von geometrischen Eigenschaften, wie etwa der Bogenlänge, ist eine Aufstellung der ersten Fundamentalform c notwendig. Die erste Fundamentalform kann in Koordinaten $\mathbf{S}(t_1, t_2) = (x(t_1, t_2), y(t_1, t_2), z(t_1, t_2))$ als symmetrische und positiv definite Matrix $\mathbf{c}_{t_1, t_2}$ deklariert werden (Kühnel 2010):

$$\mathbf{c}_{t_1, t_2} = \begin{pmatrix} E(t_1, t_2) & F(t_1, t_2) \\ F(t_1, t_2) & G(t_1, t_2) \end{pmatrix} \quad (2.39)$$

Die Koeffizienten E , F und G definieren sich nach Weitz (2019) wie folgt:

$$E(t_1, t_2) = \left\langle \frac{\partial \mathbf{S}}{\partial t_1}, \frac{\partial \mathbf{S}}{\partial t_1} \right\rangle = \left| \frac{\partial \mathbf{S}}{\partial t_1} \right|^2 \quad (2.40)$$

$$F(t_1, t_2) = \left\langle \frac{\partial \mathbf{S}}{\partial t_1}, \frac{\partial \mathbf{S}}{\partial t_2} \right\rangle \quad (2.41)$$

$$G(t_1, t_2) = \left\langle \frac{\partial \mathbf{S}}{\partial t_2}, \frac{\partial \mathbf{S}}{\partial t_2} \right\rangle = \left| \frac{\partial \mathbf{S}}{\partial t_2} \right|^2 \quad (2.42)$$

Demnach kann die Matrix der ersten Fundamentalform $\mathbf{c}_{t_1, t_2}$, die auch als Maßtensor bezeichnet wird, mittels der partiellen Ableitungen der Flächenformel $\mathbf{S}(t_1, t_2)$ nach den Parametern t_1 und t_2 sowie deren Skalarprodukten generiert werden. Die Schreibweise $\langle \cdot, \cdot \rangle$ steht hierbei für das Skalarprodukt zweier Vektoren.

Während die erste Fundamentalform $\mathbf{c}_{t_1, t_2}$ Aufschluss über die innere Geometrie einer Fläche gibt, eröffnet die Aufstellung der zweiten Fundamentalform die Möglichkeit, Parameter der äußeren Geometrie wie etwa die Krümmung berechnen zu können. Die äußere Geometrie einer Fläche beschreibt die Lage der Tangentialebene sowie ihres orthogonalen Komplements, dem Normalenvektor, im umgebenden Raum (Eschenburg und Jost 2007).

Die zweite Fundamentalform $\mathbf{h}_{t_1, t_2}$ lautet:

$$\mathbf{h}_{t_1, t_2} = \begin{pmatrix} L(t_1, t_2) & M(t_1, t_2) \\ M(t_1, t_2) & P(t_1, t_2) \end{pmatrix} \quad (2.43)$$

und setzt sich aus den Koeffizienten

$$L(t_1, t_2) = \left\langle \frac{\partial^2 \mathbf{S}}{\partial t_1^2}, \mathbf{NO}(t_1, t_2) \right\rangle \quad (2.44)$$

$$M(t_1, t_2) = \left\langle \frac{\partial^2 \mathbf{S}}{\partial t_1 \partial t_2}, \mathbf{NO}(t_1, t_2) \right\rangle \quad (2.45)$$

$$P(t_1, t_2) = \left\langle \frac{\partial^2 \mathbf{S}}{\partial t_2^2}, \mathbf{NO}(t_1, t_2) \right\rangle \quad (2.46)$$

zusammen, die mittels des Skalarprodukts der zweiten partiellen Ableitungen sowie des Einheitsnormalenvektors der Fläche $\mathbf{S}$ berechnet werden können.

Der Einheitsnormalenvektor $\mathbf{NO}$ kann wie folgt generiert werden:

$$\mathbf{NO}(t_1, t_2) = \frac{\frac{\partial \mathbf{S}}{\partial t_1} \times \frac{\partial \mathbf{S}}{\partial t_2}}{\left\| \frac{\partial \mathbf{S}}{\partial t_1} \times \frac{\partial \mathbf{S}}{\partial t_2} \right\|}, \quad (2.47)$$

wobei im Zähler das Kreuzprodukt der ersten partiellen Ableitungen nach t_1 und t_2 und im Nenner die euklidische Norm des gleichen Kreuzproduktes zu ermitteln ist (Weitz 2019). Das Kreuzprodukt der zwei Tangentenvektoren im dreidimensionalen Raum spiegelt einen Vektor wieder, der orthogonal auf der von beiden Vektoren aufgespannten Ebene steht.

Abbildung 2.31 präsentiert den Normalenvektor eines Punktes auf einer B-Spline-Fläche, der mittels der beiden Vektoren der ersten partiellen Ableitungen nach t_1 und t_2 generiert wird.

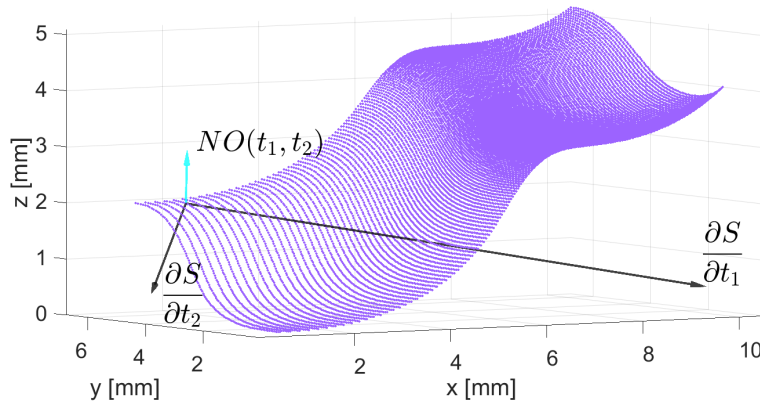


Abb. 2.31: Normalenvektor einer B-Spline-Fläche (Aichinger und Schwieger 2022).

Auf Basis der zweiten Fundamentalform kann durch Erzeugung der Eigenwerte λ_1 und λ_2 sowie der Berechnung des Produkts dieser beiden Werte ein Krümmungsparameter K ermittelt werden, der die äußere Geometrie der B-Spline-Fläche präsentiert:

$$K = \det(\mathbf{h}_{t_1, t_2}) = \lambda_1 \cdot \lambda_2. \quad (2.48)$$

Gleichung (2.48) zeigt die Berechnung des genannten Krümmungsparameters, der ebenso über die Determinante der Matrix der zweiten Fundamentalform hergeleitet werden kann. Analog zum Kurvenfall, bei dem über die zweite Ableitung der B-Spline-Kurvenpunkte in Gleichung (2.27) ein Krümmungsparameter bestimmt wird, kann somit gleichermaßen für den Flächenfall für B-Spline-Flächenpunkte ein Krümmungsparameter ermittelt werden (Aichinger und Schwieger 2022).

2.4 Interpolation und Approximation von Punktmengen mittels B-Splines

Die in den vorherigen Unterkapiteln vorgestellten Grundlagen zu B-Spline-Kurven und -Flächen liefern die Basis dieser Arbeit und gestatten ein besseres Verständnis für Funktionsweise und Eigenschaften der angewendeten B-Spline-Technik.

Mittels TLS ist eine hochauflösende räumliche Diskretisierung von Objekten möglich. Während die Aufnahme eines Objekts zu unterschiedlichen Zeitpunkten die datentechnische Grundlage für Deformationsanalysen schafft, gewährleistet sie jedoch keine unmittelbare Option zur Auswertung identischer Punkte über mehrere Epochen (Mukupu et al. 2016). Die Ursache liegt darin, dass TLS keine Messung eindeutig definierter 3D-Punkte vornimmt. Aus diesem Grund können TLS-Daten unterschiedlicher Messepochen nicht ohne Zwischenprozessierung verglichen werden (Holst et al. 2017). In Relation zur Tachymetrie ist die Punktgenauigkeit zwar in den meisten Fällen geringer, allerdings kann dieses Defizit durch eine geeignete Modellierung der Objektoberfläche unter Ausnutzung der hohen Datenredundanz kompensiert werden (Vežočník et al. 2009). Die Modellierung von TLS-Punktmengen mittels Kurven bzw. Flächen kann daher zum einen Grundlagen für weitere Methoden zur Vergleichbarkeit zwischen Messepochen schaffen und zum anderen die erfasste Objektoberfläche mit höherer Genauigkeit abbilden. Deshalb bietet es sich an, TLS-Punktmengen mittels B-Spline-Kurven bzw. -Flächen zu modellieren.

Um aus einer gegebenen großen Punktmenge $\mathbf{P}_\beta$ mit $\beta = 0, \dots, m$ eine B-Spline-Kurve oder -Fläche generieren zu können, bedarf es entsprechend aktueller Literatur einer Interpolations- oder Approximationsmethode. Während bei der Interpolation eine B-Spline-Kurve gesucht wird, die durch die aufgemessenen Laserscanning-Punkte verläuft, soll im anderen Fall eine approximierende B-Spline-Kurve die Punktmenge $\mathbf{P}_\beta$ unter Minimierung einer Fehlergrenze bestmöglich wiedergeben. Bei einer Approximation verläuft die Kurve in der Regel nicht durch die Punkte (Hoschek und Lasser 1992).

Ein essentieller Einflussfaktor auf die Qualität der erzielten Modellierung besteht in der Art der Parametrisierung, deren Wahl die Form der B-Spline-Kurve wesentlich steuert. So zeigt etwa Epstein (1976), dass die Parametrisierung die Existenz von Singularitäten oder die Möglichkeit der Bildung von Ecken beeinflussen kann.

Gleichung 2.49 verdeutlicht die Situation des Interpolationsvorgangs, der bei Bekanntsein eines Parameters t_β für jeden Interpolationspunkt $\mathbf{P}_\beta$ sowie eines Knotenvektors $\mathbf{U} = (u_0, u_1, \dots, u_k)$ als lineares Gleichungssystem zur Bestimmung der unbekannten Kontrollpunkte $\mathbf{d}_i$ gelöst werden kann (Piegl und Tiller 1997).

$$\mathbf{P}_\beta = \mathbf{C}(t_\beta) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t_\beta), \quad t \in [u_0, u_k]. \quad (2.49)$$

Um das Problem der Parametrisierung zu lösen, stehen diverse Methoden zur Verfügung. Die zwei bekanntesten stellen die chordale sowie die uniforme Parameterisierung dar. Bei der uniformen Parameterisierung werden die Parameterwerte derart gewählt, dass sie äquidistant entsprechend der Anzahl an Laserscanningpunkten über das Parameterintervall aufgeteilt werden. Schnell ersichtlich ist bei diesem Verfahren jedoch, dass das Ergebnis der erzielten B-Spline-Kurve um so schlechter ausfällt, je diverser die Punktabstände in der Punktmenge vorliegen. Befinden sich Punkte nahe beieinander, während andere große Abstände voneinander vorweisen, entstehen in der interpolierenden Kurve Knicke oder Schleifen (de Boor 1978). Die Ursache für die geringe Leistungsfähigkeit dieser Methode liegt in der vernachlässigten Beachtung der Geometrie der Datenpunkte. Dieses Defizit kann bei der chordalen Parameterisierung verringert werden, da die jeweiligen Knoten entsprechend der Euklidischen Distanz zweier benachbarter Datenpunkte und in Relation zur Gesamtlänge der aufsummierten Distanzen berechnet werden (Farin 1994). Die Berücksichtigung der unterschiedlichen Abstände liefert somit in der Regel bessere Ergebnisse als die uniforme Parameterisierung und verhindert zudem in den meisten Fällen das Auftreten von Ecken in der Kurve (Epstein 1976). Heuristische Überlegungen in physikalischer Analogie zur chordalen Methode führen zu einer dritten Parameterisierung, die sich Zentripedalmodell nennt. Diese soll von Vorteil sein, wenn die Anordnung der Laserscan-Punkte beispielsweise sehr spitzförmig zuläuft (Lee 1989a). Eine weitere Parameterisierungsmethode betrachtet, zusätzlich zur Euklidischen Distanz der Datenpunkte, die relative Form des Kontrollpolygons, wodurch Winkeländerungen berücksichtigt werden können (Foley und Nielson 1989).

Diverse Forschungsarbeiten beschäftigen sich zur Interpolation mit Weiterentwicklungen sowie Verbesserungen vorhandener Methoden. So stellen etwa Fang und Hung (2013) eine verfeinerte Zentripedalmethode vor, die weniger Schwankungen der Kurve in Relation zum Kontrollpolygon, auch in Fällen abrupter Formänderung, und zudem weniger Nachteile bezüglich Oszillationen oder Schleifen enthalten soll.

Die zweite Möglichkeit, neben der Interpolation, eine Punktmenge mittels B-Spline-Kurven oder -Flächen zu modellieren besteht in der Approximation, bei der abgesehen von den Datenpunkten ebenso eine Fehlergrenze Eingang findet. Da für gewöhnlich keine Information darüber gegeben ist, wie hoch die Anzahl an Kontrollpunkten ausfallen muss, um den vorgegebenen Grenzwert E zu erreichen, erfolgt die Durchführung von Approximationsmethoden in den meisten Fällen in einem iterativen Prozess und auf folgende der zwei Arten (Piegl und Tiller 1997):

- 1 a) Beginne mit dem Minimum oder einer kleinen Anzahl an Kontrollpunkten
 - b) Bilde eine approximierende Kurve oder Fläche zu den Datenpunkten unter Nutzung einer Approximationsmethode
 - c) Überprüfe die Abweichung der Kurve/ Fläche von den Daten
 - d) Wenn die Abweichung überall kleiner als E ist, beende den Vorgang, ansonsten erhöhe die Anzahl an Kontrollpunkten und gehe zu 1b)
- 2 a) Beginne mit dem Maximum oder so vielen Kontrollpunkten, dass die erste Approximation E genügt
 - b) Bilde eine approximierende Kurve oder Fläche zu den Datenpunkten unter Nutzung einer Approximationsmethode
 - c) Überprüfe, ob die Abweichung der Kurve/ Fläche von den Daten überall kleiner als E ist
 - d) Wenn die Abweichung nicht überall kleiner als E ist, stelle die vorherige Kurve/ Fläche wieder her, ansonsten reduziere die Anzahl an Kontrollpunkten und gehe zu 2b)

Für den speziellen Vorgang der Bestimmung einer approximierenden B-Spline-Kurve oder -Fläche muss auf eine Approximationsmethode zurückgegriffen werden. Eine dieser Methoden nennt sich Kleinst-Quadrate-Approximationsmethode und wird von de Boor (1978) unter gewissen Annahmen als beste Approximation betitelt. Wesentlicher Inhalt der Kleinsten-Quadrate-Methode ist eine Minimierung der Summe der quadrierten Abweichungen einer erstellten Funktion von den Datenpunkten (Lancaster und Salkauskas 1990). Bei $c+1$ Datenpunkten und der Annahme einer Anfangs- und Endpunkt-Interpolation der Kurve mit

$$\mathbf{P}_0 = \mathbf{C}(u_0) = \mathbf{d}_0 \quad \text{und} \quad \mathbf{P}_c = \mathbf{C}(u_k) = \mathbf{d}_m, \quad (2.50)$$

ist demnach eine B-Spline-Kurve gesucht, die die restlichen Laserscanpunkte $\mathbf{P}_\beta$ mit $\beta = 1, \dots, c-1$ im Sinne der Kleinsten-Quadrate-Methode in Bezug auf die unbekannten Kontrollpunkte $\mathbf{d}_0, \dots, \mathbf{d}_m$ approximiert (Piegl und Tiller 2000):

$$\min \sum_{\beta=1}^{c-1} |\mathbf{P}_\beta - \mathbf{C}(\bar{u}_\beta)|^2. \quad (2.51)$$

Im Gegensatz zur Interpolation verläuft die approximierende B-Spline-Kurve in der Regel nicht durch die Datenpunkte. Bei der Interpolation kann die Bestimmung der Kontrollpunkte nach Festlegung der Parametrisierung eindeutig ausgeführt werden. Im Vorgang der Approximation spielen dagegen sowohl die Selektion der Kontrollpunkte als auch die Parametrisierung eine bedeutende Rolle. Die am häufigsten zur Parameterbestimmung im Rahmen der Kleinsten-Quadrate-Methode verwendeten Methoden stellen die uniforme, chordale oder zentripedale Parametrisierung dar. Diese Methoden beruhen allerdings auf der Annahme, dass die Datenpunkte für Kurven in einer Reihe und für Flächen gitterförmig angeordnet sind (Ma und Kruth 1995).

2.5 Zusammenfassung des Kapitels

Die B-Spline-Technik bietet aufgrund ihrer positiven Eigenschaften günstige Bedingungen für eine Modellierung von Kurven- bzw. Flächenpunkten aus Laserscandaten. Die Eigenschaft der Lokalität beschränkt eine Auswirkung von Kontrollpunktpositionsänderungen auf einen lokalen Kurven- bzw. Flächenbereich. Die Anfangs- und Endpunkt-Eigenschaft sorgt dafür, dass die End- bzw. Eckkontrollpunkte mit den End- bzw. Eckflächenpunkten zusammenfallen und auf diese Weise eine eindeutig definierte Abgrenzung der B-Spline-Kurve bzw. -Fläche statt findet. Auf Basis der Ableitung von

B-Spline-Kurven bzw. -Flächen, kann ein Krümmungsparameter für jeden Kurven- bzw. Flächenpunkt bestimmt werden. Zur Modellierung von TLS-Punktmengen mit B-Splines sind bisher zwei übergeordnete Ansätze verbreitet. Beim Ansatz der Interpolation wird die B-Spline-Kurve bzw. -Fläche derart konstruiert, dass sie durch die TLS-Punkte verläuft. Der zweite Ansatz besteht aus der Approximation, bei der überwiegend mittels Kleinster-Quadrate-Methode die Kontrollpunkte und Parametrisierung für eine approximierende B-Spline-Kurve bzw. -Fläche geschätzt werden.

3 Forschungsstand TLS bzgl. B-Spline Modellierung, Registrierung und Deformationsanalyse

Die in dieser Arbeit entwickelte Methode zur Deformationsanalyse von TLS-Daten behandelt eine Modellierung von TLS-Punktwolken mittels direkter funktionaler Approximation von B-Splines, ein lokales Zielzeichen-freies merkmalsbasiertes Registrierungsverfahren zur Festlegung von Punktkorrespondenzen und eine abschließende Deformationsanalyse mittels B-Splines. In den folgenden drei Unterkapiteln sollen die drei Ansätze in einen aktuellen wissenschaftlichen Kontext eingeordnet werden. Das erste Unterkapitel stellt aktuelle Ansätze zur Modellierung von TLS-Daten mittels B-Splines mit besonderem Fokus auf Ansätze zur Approximation vor. Das zweite Unterkapitel widmet sich der Einführung in Registrierungsverfahren und geht dabei im Speziellen auf geometriebasierte und merkmalsbasierte Registrierungsverfahren ein. Das dritte Unterkapitel greift aktuelle Ansätze der Deformationsanalyse mit B-Spline-Kurven bzw. -Flächen auf.

3.1 Modellierung von TLS-Daten mit B-Splines

Für die Modellierung von Punktwolken kann die Verwendung geometrischer Primitive nützlich sein. Oftmals sind diese allerdings in ihrer Anwendung beschränkt, da komplexere Formen nicht ohne Verlust an Oberflächeninformationen des Objekts modelliert werden können. Deshalb geht die Entwicklung zunehmend hin zur Nutzung von Freiformflächen wie z.B. B-Splines, die in ihrer Anwendung flexibler sind. B-Splines werden in einer Vielzahl an unterschiedlichen geodätischen Fachgebieten genutzt. Mohammadiojdan et al. (2025) setzt B-Spline-Flächen zur Modellierung der Unterwassertopographie und Bereinigung von Daten eines bathymetrischen Multibeam-Echolot-Systems ein. Riel et al. (2018) untersucht Deformationen in Küstenbecken, die mittels Interferometric Synthetic Aperture Radar (InSAR) erfasst und deren saisonale Signale mittels B-Splines modelliert werden. Yuan et al. (2019) bringt eine B-Spline-Approximation in einem auf Luftbildern basierenden Methodenprozess zur Erstellung dichter 3D-Punktwolken zum Einsatz. Boergens et al. (2021) verwendet eine Erweiterung von B-Splines zur Modellierung von Satellitenaltimetriedaten nicht verzweigter Flussnetze.

Im Bereich der Ingenieurgeodäsie werden TLS-Daten zunehmend mittels B-Splines, häufig durch Anwendung von Approximationsverfahren, modelliert. In dieser Arbeit wird eine direkte funktionale Approximation von B-Spline-Kurven und -Flächen angewendet. Um einen Einblick in bisherige Arbeiten und Vorgehensweisen für die klassische Approximation von B-Splines zu erhalten, werden im Folgenden einige Ansätze näher erläutert.

Ansätze zur Approximation von B-Spline-Kurven liefern Schmitt und Neuner (2015). Hierbei wird ein iterativer Prozess konstruiert, der mit einer Minimalkonfiguration startet und auf Basis von Näherungswerten, die über die uniforme, chordale und zentripedale Parametrisierungsmethode generiert werden, Kontrollpunkte einer B-Spline-Kurve durch ein lineares Gauß-Markov-Modell schätzt. Erfüllen die Residuen, die aus der Kontrollpunktschätzung erhalten werden, ein festgelegtes Gütekriterium nicht, muss ein neuer Knoten sowie dessen Position innerhalb eines Schleifendurchgangs geschätzt werden. Die Näherungswerte für die Position des neuen Knotens werden auf Grundlage des Schnittpunktes zweier kumulierter Summen der quadrierten Residuen bestimmt. Für die B-Spline-Basisfunktionen wird ein Grad von $n = 2$ sowie ein offener Knotenvektor angenommen.

Ebenfalls im Bereich der B-Spline-Kurvenapproximation, rückt Bureick (2020) die Wichtigkeit der Modellwahl, Parametrisierung, Knotenvektwahl sowie Kontrollpunktschätzung in seiner Arbeit in den Fokus und stellt drei neue bzw. modifizierte Algorithmen zur Bestimmung des Knotenvektors

für die B-Spline-Approximation vor. Hierbei legt er die Residuenquadratsumme aus der Kontrollpunktschätzung als Fitnesswert fest, um die Güte der inneren Knoten des Knotenvektors, die er als Individuum bezeichnet, beurteilen zu können. Die angewandte Kontrollpunktschätzung erfolgt mittels der Methode der Kleinsten Quadrate. Für die zur Ermittlung des Fitnesswerts durchzuführende Kontrollpunktschätzung werden als Eingangsgrößen unter anderem der Grad der Basisfunktion, die Anzahl der Kontrollpunkte und der Knotenvektor als Eingangsgrößen festgesetzt. In Folge der größeren Datenmengen verhält sich eine Adaption zu B-Spline-Flächen als wesentlich rechenintensiver als für Kurven. Des weiteren führt eine Verschiebung der Bedeutung einzelner Teilschritte hierzu, dass der Parametrisierung bei der B-Spline-Flächenapproximation eine deutlich höhere Gewichtung zu Teil werden sollte.

Obwohl Flächenparameter mit einer starken Einflussnahme auf die B-Spline-Flächenschätzung einwirken, wird die Parametrisierung für gewöhnlich in geodätischen Anwendungen nicht eingehend untersucht. So zeigt etwa Harmening (2020), dass derart einfach gehaltene Verfahrensweisen zwar für ausgewählte Datenpunkte zufriedenstellende Ergebnisse liefern, jedoch keine allgemeine Einsatzmöglichkeit bieten. Zum Beispiel wird in Koch (2010) die Annahme einer gitterförmige Anordnung gemessener Punkte getroffen, um die chordale Parametrisierung nutzen zu können, die als Grundlage für die darauffolgende Schätzung der B-Spline-Fläche dienen soll. Für die Schätzung einer B-Spline-Fläche aus Laserscandaten mittels Approximation entwickeln Harmening und Neuner (2014) ein Verfahren, das selbst bei ungeordneten Punktwolken anwendbar sein und die Basis für Deformationsanalysen bilden soll. In Folge der thematischen Nähe der in dieser Arbeit vorzustellenden Methodik, soll die genannte Technik vorgestellt werden, um einen Eindruck der Dimension notwendiger Verfahrensschritte sowie des Gesamtaufwands zu gewährleisten.

Bei dem genannten Verfahren wird auf einer Methode nach Ma und Kruth (1995) aufgesetzt, bei der eine Basisfläche mit bekannter Parameterform erzeugt wird, auf die nachfolgend Punkte projiziert werden sollen. Eine Basisfläche kann durch vier Randkurven als sogenanntes Coons-Patch nach Farin (2002) definiert werden. Ausgehend von dieser Basisfläche findet anschließend die Durchführung der Ausgleichung statt. Eine Detektion der Randpunkte als Fundament der Randkurven erfolgt nach einem entwickelten Verfahren nach Harmening (2013). Das gesuchte Coons-Patch wird daraufhin mittels zweier Regelflächen, bestehend aus jeweils gegenüberliegenden Kurven mit linearer Interpolation (Farin 2002) sowie einem bilinearen Interpolant der vier Eckpunkte und demgemäß der Kombination dreier Flächen konstruiert (Piegl und Tiller 1997). Um die für die Gauß-Markov-Schätzung der approximierenden B-Spline-Fläche notwendigen Parameter t_1 und t_2 zu bestimmen, werden eine Projektion der Messpunkte mit Hilfe einer Hauptachsentransformation in das Koordinatensystem des nächstgelegenen Vierecks und daraufhin eine inverse bilineare Interpolation aus den Eckpunkten des Vierecks vorgenommen (Neuner et al. 2015). Wie bereits erwartet, approximiert die geschätzte B-Spline-Fläche die Punktwolke besser als die Coons-Patch Basisfläche. Ein iterativer Prozess mittels neuer Parametrisierung auf Basis von nach der chordalen Methode generierten Isolinien soll diese Ergebnisse weiter verbessern. Da bei einem der zwei getesteten Datensätzen jedoch Artefakte infolge geringer Punktdichten in den Randbereichen auftreten, ist die Einführung von Nebenbedingungen notwendig, die in Form von fiktiven Beobachtungen durch gewichtete Beschränkungen in den Ausgleichungsprozess inkludiert werden sollen (Harmening und Neuner 2015). Zusätzlich zu der entwickelten Parametrisierungsmethode definiert Harmening (2020) anhand der Lösung von Modellselektionsaufgaben die optimale Komplexität einer B-Spline-Fläche bezüglich Kontrollpunktanzahl und Grad der B-Spline-Fläche. Die erfolgreiche Punktwolkenapproximation dient daraufhin der Entwicklung einer Deformationsanalyse, die für ingenieurgeodätische Objekte mit kontinuierlicher Oberfläche und eindeutig definierten Rändern anwendbar ist. Außerdem können lediglich kontinuierliche Deformationsprozesse modelliert werden. Das Modell ist vor allem für Verformungen entwickelt, die klein in Relation zur Objektgröße ausfallen (Harmening 2020).

Zusammenfassend lässt sich feststellen, dass für die Modellierung von TLS-Daten überwiegend auf Approximationsverfahren mittels Kleinsten-Quadrate-Schätzung zurück gegriffen wird, bei der ein Teil

der Parameter a priori festgelegt werden muss. Der optimale Basisfunktionsgrad und die optimale Anzahl der Kontrollpunkte sind über eine Modellauswahl wie Informationskriterien entsprechend der Komplexität des erfassten Objekts zu bestimmen. Durch eine Parametrisierungsmethode ist für jeden Punkt der zu approximierenden Punktwolke ein Lageparameter zu definieren, sowie ein Knotenvektor durch ein deterministisches oder heuristisches Verfahren zu erzeugen. Die Güte dieser Parameterdefinitionen bestimmt, wie gut die geschätzte B-Spline-Kurve oder -Fläche die reale Objektform wiedergibt. Eine unsachgemäße Wahl der genannten Parameter kann daher einen erheblichen negativen Einfluss auf die Qualität der B-Spline-Approximation haben. Die in dieser Arbeit entwickelte Methode zur B-Spline Modellierung basiert auf einem Ansatz, der keine Parameterschätzung benötigt und eine sogenannte direkte funktionale Approximation einsetzt. Die TLS-Daten werden hierbei direkt als Kontrollpunkte genutzt. Die Methode soll einen neuartigen alternativen Ansatz aufzeigen, der den Vorteil der Modellierung mit B-Splines inne hat und gleichzeitig auf komplexe Parameterschätzungen verzichtet.

3.2 Registrierungsverfahren

Eine grundlegende Bedingung für die Durchführung einer Deformationsanalyse ist die Transformation zweier Messepochen in ein gemeinsames Koordinatensystem. Dieser Prozess wird Registrierung genannt und kann im Allgemeinen in zwei Ansätze differenziert werden: die absolute Registrierung bzw. Georeferenzierung, bei der die Punktwolken in ein globales Koordinatensystem transformiert werden, und die lokale Registrierung, bei der ein lokales Koordinatensystem einer Messepoche als Zielsystem dient (Yang 2024).

Für die Umsetzung der absoluten Registrierung werden entweder Zielzeichen benötigt, deren Position über Tachymetrie oder GNSS festgelegt wird, oder integrierte Sensoren, wie GNSS, inertielle Messeinheiten und andere unterstützende Sensoren, die eine direkte Bestimmung globaler Koordinaten gewährleisten (Wujanz 2016; Akca 2003). Die in dieser Arbeit entwickelte Methode kann der lokalen Zielzeichen-freien Registrierung zugeordnet werden und diese wird demzufolge intensiver behandelt. Für gewöhnlich verfolgen lokale Registrierungsverfahren zwei Ziele: eine Bestimmung von Korrespondenzen und eine Berechnung von Transformationsparametern. Die Ermittlung von Korrespondenzen bestimmt maßgebend die Güte der Registrierung und wird vor der Transformationsparameterberechnung durchgeführt (Stilla und Xu 2023). Ebenso wie bei der absoluten Registrierung können auch bei der lokalen Registrierung Zielzeichen zur Festlegung von Korrespondenzen verwendet werden (Frasaszek et al. 2009). Im weiteren Verlauf soll sich auf Zielzeichen-freie Verfahren konzentriert werden. Häufig werden bei Registrierungsverfahren Grob-zu-Fein-Prozesse eingesetzt, bei denen mittels einer Grobregistrierung eine grobe Zuordnung von Korrespondenzen erfolgt, um eine Transformation in ein ähnliches Koordinatensystem zu ermöglichen. In einem zweiten Schritt wird eine Feinregistrierung durchgeführt, die eine Koordinatentransformation zweier Epochen in ein gemeinsames System verbessern soll (Wujanz 2016). Bei etablierten Registrierungsverfahren bestehen oftmals Varianten und Weiterentwicklungen, die nicht mehr nur der Grob- oder Feinregistrierung zugeordnet werden können. Die grundlegende Idee, ob eine Methode z.B. auf Basis von Punktabständen der 3D-Rohdaten oder einem speziell berechneten Flächenmerkmal entwickelt wurde, bleibt in den meisten Fällen erhalten. Deshalb wird im Folgenden eine Unterscheidung der Registrierungsverfahren nach geometriebasierten und merkmalsbasierten Registrierungsverfahren durchgeführt, wobei im Detail bei jedem Ansatz auf die Anwendungsmöglichkeit für eine Grob- oder Feinregistrierung eingegangen wird.

Der Vollständigkeit halber soll erwähnt werden, dass es abgesehen von den zwei genannten Registrierungsverfahren ebenso Ansätze gibt, die Deep-Learning-basierten Registrierungsverfahren zugeordnet werden können. Diese beruhen auf neuronalen Netzwerken und finden in dieser Arbeit allerdings keine

Anwendung. Für weiterführende Informationen sei auf Stilla und Xu (2023), Dong et al. (2020) und Huang et al. (2021) verwiesen.

3.2.1 Geometriebasierte Registrierungsverfahren

Geometriebasierte Registrierungsverfahren arbeiten häufig mit Punkten oder geometrischen Primitiven, um die Euklidische Distanz zwischen Korrespondenzen zu ermitteln (Stilla und Xu 2023). Eine der bekanntesten Registrierungsmethoden ist der Iterative-Closest-Point (ICP) Algorithmus nach Besl und Mc Kay (1992) und Chen und Medioni (1992). Das Grundprinzip des ICP besteht in der Minimierung der Euklidischen Distanzen benachbarter Punkte. Durch einen iterativen Vorgang werden Punktkorrespondenzen gesucht, Transformationsparameter geschätzt und daraufhin eine Transformation auf alle Punkte angewendet. Dieser Vorgang wird solange wiederholt bis die Summe der Euklidischen Distanzen der betrachteten Punktkorrespondenzen unterhalb einer festgelegten Schwelle liegt. Für eine erfolgreiche Anwendung des ICP ist eine angenäherte Ausrichtung der beiden Punktwolken unabdingbar, da der ICP ansonsten nicht zuverlässig konvergiert und somit fehlerhafte Punktkorrespondenzen nicht korrigiert werden können. Bae und Lichti (2008) implementieren Weiterentwicklungen des ICP durch die Hinzunahme geometrischer Informationen in Form geschätzter Normalenvektoren und Censi (2008) entwickelt eine Variante des ICP, die eine Punkt-zu-Linie Methode nutzt. Allgemein erfordern ICP-basierte Methoden für die erfolgreiche Anwendung eine grobe Ausrichtung der Punktwolken und sind deshalb der Feinregistrierung zuzuordnen. Eine Grobregistrierung kann manuell durchgeführt (Vosselman und Maas 2010) oder durch ein automatisches Verfahren implementiert werden. Aiger et al. (2008) entwickelt das 4-Points Congruent Sets (4PCS) Registrierungsverfahren, das durch die Extraktion koplanarer 4-Punkte-Sätze eine robuste grobe Registrierung der 3D-Daten erzeugen kann. Da das 4PCS-Verfahren unter anderem mit dem Verhältnis der Punktdistanzen eines 4-Punkt-Satzes arbeitet und deshalb Beschränkungen bei stark variierender Punktdichte vorweist, wird dieser Ansatz durch den Keypoint-based 4PCS (K-4PCS) weiterentwickelt (Theiler et al. 2013). Das K-4PCS unterteilt die Punktwolke zunächst in ein regelmäßiges Gitter von Blöcken (z.B. Voxel), die durch den Schwerpunkt der zugehörigen 3D-Punkte repräsentiert und in denen Keypoints auf Basis eines 3D Difference-of-Gaussian Keypoint extrahiert werden. Die grob registrierten Punktwolken werden daraufhin mit dem ICP feinregistriert (Theiler et al. 2014).

3.2.2 Merkmalsbasierte Registrierungsverfahren

Bei merkmalsbasierten Registrierungsverfahren besteht die zentrale Aufgabe in der Erzeugung einer eindeutigen Merkmalsbeschreibung, die als Deskriptor bezeichnet wird. Viele dieser Ansätze haben den Vorteil, dass der extrahierte Deskriptor auf lokalen translations- und rotationsinvarianten Merkmalen beruht und sich auf diese Weise zur Herstellung lokaler Korrespondenzen eignet (Tombari et al. 2010). Zu den klassischen merkmalsbasierten Methoden gehört das Fast Point Feature Histogram (FPFH) von Rusu et al. (2009), das lokale Punktmerkmal-Histogramme berechnet. Die FPFH können mittels geometrischer Bedingungen wie der Oberflächennormalen von einem Punkt zu einer bestimmten Anzahl an Nachbarn berechnet werden. Ebenso auf Grundlage lokaler Histogramme agiert der Signature of Histograms of Orientations (SHOT) und seine Erweiterung mit RGB-Daten. Dieser Deskriptor beruht auf der Definition eines wiederholbaren lokalen Referenzrahmens, der auf der Eigenwertanalyse der Punktmatrix in der Umgebung eines Punktes basiert (Tombari et al. 2010; Tombari et al. 2011). Prinzipiell lassen sich zu merkmalsbasierten Registrierungsverfahren Ansätze zählen, die geometrische Primitive, wie Linien (Hebel und Stilla 2012; Yang et al. 2015), Kurven (Yang und Zang 2014), Ebenen (Chen et al. 2020) oder Oberflächen (Ge und Wunderlich 2016) zur Erzeugung von Merkmalen nutzen. Diese werden überwiegend zur Grobregistrierung eingesetzt und benötigen eine nachgeschaltete Feinregistrierung.

Die vorgestellten Registrierungsverfahren geben einen Einblick in vorhandene Ansätze zur Registrierung von 3D-Punktwolken. Ein Aspekt, der hierbei bisher nicht betrachtet wurde, ist die Verhaltensweise der Algorithmen bei deformierten Punktwolken oder uneinheitlichen überlappenden Bereichen zwischen den Epochen. Bei Verfahren zur Grobregistrierung könnte diese Beschränkung durch die Verwendung von Resampling-Algorithmen, wie z.B. RANSAC (Fischler und Bolles 1981), kompensiert werden. Methoden zur Feinregistrierung, wie der klassische ICP, sind nur dann in der Lage präzise Transformationsparameter zu bestimmen, wenn keine oder nur vernachlässigbare Verformungen auftreten (Wunderlich et al. 2016). Signifikante Deformationen können zu falschen Korrespondenzen, somit zu einer fehlerhaften Registrierung und letztlich zu einer falschen Deformationsanalyse führen. Dementsprechend sollten instabile Bereiche bei der Registrierung ausgeschlossen werden (Friedli 2020). Berücksichtigung finden diese Überlegungen bisher z.B. bei einer Weiterentwicklung des ICP, dem ICProx in Wujanz et al. (2014), der in Verbindung mit dem 4PCS ein zweistufiges Verfahren vorstellt. Dieses vergleicht Transformationsvektoren und schließt vorliegende Deformationen durch Ausschluss ganzer Octreezellen aus. Yang (2024) entwickelt einen Deformationsanalyseprozess, der eine Grobregistrierung mittels 4PCS und eine Feinregistrierung mit einem point-to-plane ICP (Park und Subbarao 2003) enthält. Besonderheit ist die Verwendung einer Patchbasierten Segmentierung, bei der die 3D-Punktwolke in kleine ebene Bereiche, sogenannte Supervoxel, unterteilt wird und lediglich stabile Supervoxel zur Feinregistrierung genutzt werden (Yang und Schwieger 2022).

Im Allgemeinen kann festgestellt werden, dass der ICP und seine Varianten ein etabliertes Werkzeug darstellt, um TLS-Daten präzise zu registrieren. Enthalten die Messungen Deformationen, kann dies zu fehlerhaften Korrespondenzen führen. Somit ist ein robusteres Verfahren auf Grundlage merkmalsbasierter Deskriptoren vorzuschalten, das instabile Bereiche ausschließt. Bisher gibt es in diesem Bereich nur wenige Ansätze, weshalb hier erhebliches Entwicklungspotential besteht. Die in dieser Arbeit entwickelte Registrierungsmethode kann den lokalen Zielzeichen-freien Registrierungsverfahren zugeordnet werden. Durch den Einsatz krümmungsbasierter Parameter könnte sie sowohl in geometriebasierte als auch merkmalsbasierte Registrierungsverfahren eingestuft werden. Die Krümmungsparameter werden allerdings nicht unmittelbar aus den Kontrollpunkten berechnet, sondern mittels direkter funktionaler Approximation aus den B-Spline-Kurven bzw. -Flächenpunkten durch verschiedene Prozessschritte gebildet. Für das Verfahren wird eine Krümmungsmatrix auf Basis von B-Spline-Kurven- bzw. Flächenpunkten bestimmt, die als Merkmal zur Korrespondenzsuche in einer Korrelationsanalyse herangezogen wird. Das entwickelte Verfahren wird deshalb mit größerer Gewichtung den merkmalsbasierten Registrierungsverfahren zugeschrieben. Der neue Ansatz ist bisher für eine Feinregistrierung konzipiert.

3.3 Deformationsanalyse mittels B-Splines

Die vorgestellten Methoden zur B-Spline Modellierung von TLS-Daten von Unterkapitel 3.1 zeigen, dass die B-Spline-Technik leistungsfähige Ansätze zur Flächenmodellierung liefert. Derartige Ansätze können als vielversprechende Grundlage herangezogen werden, um im Fachgebiet der Deformationsanalyse mittels B-Splines zur Bestimmung von Verformungen zwischen zwei Messepochen eingesetzt zu werden.

Wang et al. (2023) verwendet die B-Spline-Technik für die Deformationsanalyse von Tunneln. Aus den TLS-Daten der Tunnelquerschnitte werden B-Spline-Kurven approximiert und mittels ICP als Punkt-zu-Punkt-basierte Deformationsanalyse etabliert. Durch die B-Spline-Bestimmung können bei diesem Ansatz lokale Deformationen genauer festgestellt werden als bei vergleichbaren Ansätzen ohne Verwendung von B-Splines. Kermarrec et al. (2020) untersuchen die Auswirkungen auf die Ergebnisse einer Deformationsanalyse in Form von Entfernungsberechnungen durch Vergleich lokaler Abstände zwischen zwei rohen Punktwolken sowie zwischen zwei durch B-Spline-Flächen approximierten

Punktwolken. Als Entfernungsmaß werden die Hausdorff-Distanz sowie ihre gemittelte Ableitung berechnet und in Bezug auf lokale Verformungen gegenübergestellt. Die Ergebnisse der Simulationen werden auf reale Beobachtungen von einer belasteten Brücke erweitert. Harmening und Neuner (2020) stellen einen Ansatz zur Deformationsanalyse vor, in dem die Modellierung von TLS-Daten als räumlich-zeitliches Deformationsmodell, ähnlich einer Kollokation nach der Methode der Kleinsten-Quadrate, erfolgen soll. Hierbei wird das gemessene Objekt in drei Komponenten zerlegt: Einen deterministischen Trend, der das unverzerrte Objekt der ersten Epoche darstellt und mittels einer B-Spline-Fläche approximiert wird. Die B-Spline-Fläche dient in diesem Ansatz als Referenzfläche. Die zweite Komponente besteht aus einem stochastischen Signal, das den lokalen Deformationsprozess beschreibt. Die dritte Komponente berücksichtigt die bei dem Messvorgang verursachten Unsicherheiten und stellt das Messrauschen dar. Die Deformation wird bei diesem Ansatz ausschließlich durch ein stochastisches Signal in Form von abstandsabhängigen Variogrammen modelliert. Der Ansatz ermöglicht eine raum- und zeitkontinuierliche Beschreibung und Vorhersage der Deformation und umgeht die Herausforderung, identische Punkte in verschiedenen Messzeitpunkten definieren zu müssen.

In einem weiteren Ansatz stellen Harmening et al. (2021) eine Methode an simulierten Daten vor, um Starrkörperbewegungen mit Hilfe von auf B-Spline-Flächen konstruierten Punkten zu schätzen. Diese Strategie soll, abgesehen von der Erkennung von Starrkörperbewegungen zwischen zwei Epochen, ebenso Bereiche lokalisieren können, die deformiert sind. Genauer werden in dem entwickelten Ansatz zuerst B-Spline-Flächen von Punktwolken verschiedener Epochen geschätzt und Punktkorrespondenzen auf den B-Spline-Oberflächen festgelegt. Mittels eines RANSAC-Ansatzes werden die Parameter der Starrkörperbewegung robust geschätzt. Die so erhaltene Vergleichsmenge stellt zunächst die nicht verformten Bereiche des Objekts dar, die in einem zweiten Schritt statistisch untersucht werden.

Die Untersuchung der bisherigen Ansätze zur Deformationsanalyse unter Nutzung der B-Spline-Technik zeigt deren Potential hinsichtlich der Bestimmung von lokalen Verformungen. Feststellbar ist, dass Deformationen bislang ausschließlich mittels Angabe von Abständen oder stochastischen Signalen in Form abstandsabhängiger Variogramme ausgegeben werden. Eine präzisere Untersuchung, die über die Betrachtung von Abständen hinaus geht und 3D-Deformationen ermittelt, erfolgt bisher nicht. Durch Bestimmung von 3D-Deformationen könnten allerdings wichtige Daten erhoben werden, in welche Richtung sich eine Deformation bewegt. Aus diesem Grund liefert die in dieser Arbeit entwickelte Methode, abgesehen von Abständen zwischen korrespondierenden Punkten, zusätzlich 2D- bzw. 3D-Deformationen. Auf diese Weise kann differenziert eine Aussage über Verschiebungen von Punktkorrespondenzen bzgl. der x-, y- und z-Koordinatenrichtung gegeben werden.

4 Direkte B-Spline-Kontrollpunkt-Methode (DBKM)

Auf Basis der in den vorherigen Kapiteln dargelegten wesentlichen Grundlagen zu B-Spline-Kurven und Tensorprodukt B-Spline-Flächen und der Einordnung der entwickelten Methode in den aktuellen wissenschaftlichen Kontext, soll die entwickelte DBKM vorgestellt werden. Unterkapitel 4.1 erläutert die Struktur der Methode und geht im Detail auf vorbereitende Maßnahmen und Rahmenbedingungen (Abschnitt 4.1.1) sowie im Anschluss auf den Prozessablauf der DBKM (Abschnitt 4.1.2) ein. Unterkapitel 4.2 widmet sich den Einzelkomponenten der entwickelten Methode zur Modellierung von TLS-Daten mittels Tensorprodukt B-Spline-Flächen. Hierzu zählen die Definition von Kontrollpolygonen (Abschnitt 4.2.1) und die Anwendung eines Knoteneinfügealgorithmus (Abschnitt 4.2.2). An dieser Stelle findet eine kurze thematische Vorstellung und Gegenüberstellung zweier Knoteneinfügealgorithmen statt. Das nächste Unterkapitel 4.3 stellt den entwickelten Mapcut-Algorithmus als lokales Zielzeichen-freies Registrierungsverfahren auf Basis eines Krümmungsparameters als Merkmalsdeskriptor vor. Der Mapcut-Algorithmus wird in Abschnitt 4.3.1 für B-Spline-Kurven und in Abschnitt 4.3.2 für B-Spline-Flächen präsentiert.

4.1 Struktur der Methode

Im Folgenden sollen mit dem Ablaufdiagramm in Abbildung 4.1 die einzelnen Prozessschritte der neu entwickelten Methode, beginnend bei dem zu scannenden Objekt bis hin zur Deformationsanalyse, vorgestellt werden. Welche vorbereitenden Maßnahmen und Rahmenbedingungen an den TLS-Standpunkt bzw. die Punktwolke oder die Objektverformung für eine Funktionsfähigkeit der DBKM gestellt sind, soll zunächst behandelt werden.

4.1.1 Vorbereitende Maßnahmen und Rahmenbedingungen

Eingeleitet wird die Prozesskette durch eine Festlegung der Abtastrate für das TLS, die sich anhand der Eigenschaften des Objekts hinsichtlich Objektgröße und -entfernung, Detailierungsgrad des Objekts sowie erwarteter Deformationsausdehnung ausrichtet. Bei den Laserscanaufnahmen muss sichergestellt werden, dass stabile Objektbereiche gewährleistet sind. Im Zusammenhang mit der Vorbereitung der Erfassung des Objekts sollte berücksichtigt werden, dass mit der entwickelten Methode in Relation zur Objektgröße kleine Deformationen erfasst werden sollen.

Als Weiteres sollte bei den unterschiedlichen epochenweisen Laserscannmessungen beachtet werden, dass ähnliche Aufnahmekonfigurationen sowie die Einhaltung der gleichen Abtastrate die Vergleichbarkeit der unterschiedlichen Messepochen begünstigen. Bei Berücksichtigung der gleichen Parameter hinsichtlich Laserscanner, Abtastrate, Auftreffwinkel sowie Distanz zum Objekt sollten die Rahmenbedingungen gegeben sein, um annähernd die gleiche Anzahl an Kontrollpunkten pro Objektfläche pro Epochenmessung zu erhalten. Eine annähernd gleiche Anzahl an Kontrollpunkten pro Epoche beeinflusst den Aufbau des Knotenvektors und des Parameterbereichs sowie die Anzahl an B-Spline-Kurven bzw. -Flächenpunkten und ihre Distanzen zueinander. Sind diese Parameter in den unterschiedlichen Epochen gleich gewählt, resultieren annähernd gleiche Distanzen zwischen den B-Spline-Kurven bzw. -Flächenpunkten, die nach Korrespondenzzuordnung eine Vergleichbarkeit zwischen den Epochen unterstützen. Zu beachten ist ebenso, dass ausschließlich einzelne Scanaufnahmen verglichen werden, die in Scanreihenfolge abgespeichert vorliegen. Um keine Überschneidungen der einzelnen Scanreihen zu erhalten, sollte der mittlere Punktabstand der TLS-Daten deutlich größer als das Messrauschen sein. Die auszuwertenden Punktwolken sollten keine Datenlücken oder Versprünge aufweisen. Bisher ist

die DBKM an Punktwolken mit bis zu 150.000 Punkten mittels mehrerer Datensätze funktionsfähig überprüft worden. Eine Anwendung für größere Punktwolken ist möglich, solange die bereits genannten Bedingungen besonders im Hinblick auf die Scanreihenfolge eingehalten werden. Jede Punktwolke wird vor der Durchführung der DBKM einer manuellen groben Ausreißerelimination unterzogen.

Für die Generierung der B-Spline-Kurven oder -Flächen werden B-Spline Basisfunktionen dritten Grades verwendet. Die Wahl eines Polynomgrades $n = 3$ bietet eine gute Relation zwischen einer zu starken Approximation der Kurve an das Kontrollpolygon, aufgrund eines zu niedrigen Kurvengrades, sowie einer zu schwachen Approximation der Kurve an die Form des Kontrollpolygons, durch einen zu hohen Polynomgrad. In Bezug auf die Knotenvektoren werden auf Basis der Anzahl an Kontrollpunkten und dem Kurvengrad $n = 3$ vor Ausführung eines Knoteneinfügealgorithmus, uniforme offene Knotenvektoren gebildet. Diese sollen die Interpolation der Kurvenendpunkte mit den Endpunkten des Kontrollpolygons sowie einen gleichmäßigen Verlauf der Kurvenform gewährleisten. Auf diese Weise werden in diesem neuen Ansatz als Parameter lediglich der Kurvengrad sowie die Verwendung offener uniformer Knotenvektoren festgelegt (Aichinger und Schwieger 2022).

Zusammenfassen lassen sich die Rahmenbedingungen der DBKM wie folgt:

- Kongruenzmodell, kleine Deformation in Relation zu Objektgröße
- Verwendung desselben terrestrischen Laserscanners über mehrere Epochen
- näherungsweise gleicher Messaufbau (gleiche Standpunktposition)
- Verwendung von Messepochen eines Standpunktes
- Ausgabe der Punktwolke in Scanreihenfolge (sortierte Punktwolke)
- Mittlerer Punktabstand größer als Messrauschen
- Punktwolke sollte keine Lücken oder Versprünge enthalten und eine gleichmäßige Punktverteilung aufweisen
- Punktzahl der getesteten Punktwolken derzeit bis 150.000 Punkte, größere Punktzahl möglich
- B-Spline-Basisfunktionen vom Grad 3, offene uniforme Knotenvektoren

4.1.2 Prozessablauf

Abbildung 4.1 stellt die entwickelte DBKM in einem Ablaufdiagramm dar. Der linke Ablaufstrang verdeutlicht die funktionale Prozesskette und der rechte Ablaufstrang die stochastische Modellierung der Methode. Beide Komponentenabfolgen sind unabdingbar, um am Ende in einer Deformationsanalyse eine Aussage über signifikante Deformationen treffen zu können. Als Startprodukt für die entwickelte Methode stehen nach der laserscanbasierten Erfassung eines Objekts epochenweise 2D- bzw. 3D-Punktwolken zur Verfügung. Im 2D-Fall werden aus den 2D-Daten B-Spline-Kurven generiert und darauf aufbauend Krümmungsparameter für jeden B-Spline-Kurvenpunkt gebildet.

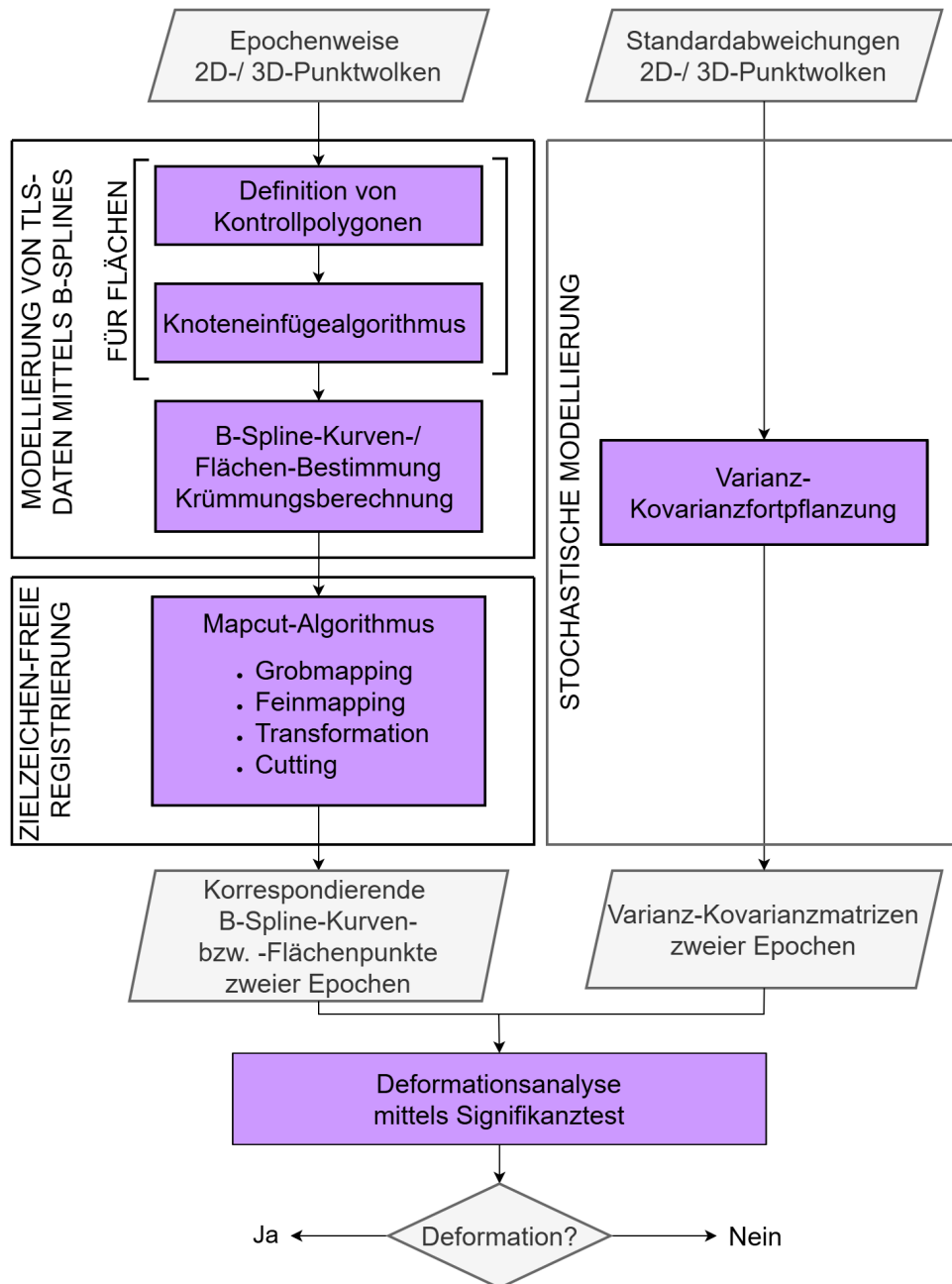


Abb. 4.1: Ablaufdiagramm der Direkten B-Spline-Kontrollpunkt-Methode (DBKM).

Im 3D-Fall sind die TLS-Daten für eine Modellierung zu Tensorprodukt B-Spline-Flächen vorzubereiten. Dies erfolgt durch Definition von Kontrollpolygonen und anschließendes Einfügen neuer Knoten bzw. Kontrollpunkte in die Kontrollpolygone bis eine bidirektionale Kontrollgitterstruktur vorliegt. Auf dieser Basis können Tensorprodukt B-Spline-Flächen und Krümmungsparameter bestimmt werden. Eine lokale Zielzeichen-freie Registrierung auf Grundlage der Krümmung als Merkmalsdeskriptor findet daraufhin mittels des entwickelten Mapcut-Algorithmus statt. Dieser ist iterativ aufgebaut und untergliedert sich in die vier Stufen: Grobmapping, Feinmapping, Transformation und Cutting. Im Grobmapping wird im 2D-Fall ein 1D-Krümmungsvektor zweier Epochen mittels Korrelationsanalyse verschoben, um eine annähernde Überlagerungsposition und Korrespondenzen zu erhalten. Diese Korrespondenzzuordnung wird im Feinmapping präzisiert. Der 3D-Fall ist analog aufgebaut, unterscheidet sich jedoch darin, dass eine 2D-Krümmungsmatrix aus den Tensorprodukt B-Spline-Flächen

extrahiert wird. Durch Auswahl einer Anzahl an annähernd identischen Punkten, die mittels Krümmungskorrespondenzsuche ermöglicht und durch die geringste Euklidische Distanz ausgewählt werden, kann eine Koordinatentransformation aller B-Spline-Kurven- bzw. -Flächenpunkte in das System von Epoche 0 stattfinden. Da unterschiedlich große Punktwolkenbereiche beider Epochen negativen Einfluss auf die Vergleichbarkeit korrespondierender B-Spline-Kurven- bzw. -Flächenpunkte haben, wird ein Cutting der größeren Punktwolke mittels Kontrollgitter der kleineren Punktwolke durchgeführt. Durch Iteration des Mapcut-Algorithmus resultieren korrespondierende B-Spline-Kurven- bzw. Flächenpunkte, deren Euklidische Distanz bestimmt werden kann. Der beschriebene funktionale Prozess wird in einer Varianz-Kovarianzfortpflanzung stochastisch modelliert. Als initiale Standardabweichung wird für die TLS-Daten Gaußsches Rauschen angenommen. Die stochastische Modellierung ergibt für jede Epoche eine Varianz-Kovarianzmatrix, die umgeformt zu einer Kofaktormatrix, in einem Signifikanztest zusammen mit den Koordinatendifferenzen der korrespondierenden B-Spline-Kurven- bzw. -Flächenpunkte auf signifikante Deformationen analysiert werden kann.

4.2 Modellierung von TLS-Daten mittels Tensorprodukt B-Spline-Flächen

Die in dieser Arbeit entwickelte Prozesskette zur Modellierung von TLS-Daten mittels Tensorprodukt B-Spline-Flächen ist in Abbildung 4.2 vereinfacht dargestellt. Grafik [a] zeigt die TLS-Rohdaten, die in Grafik [b] in Scanreihenfolge verbunden abgespeichert vorliegen.

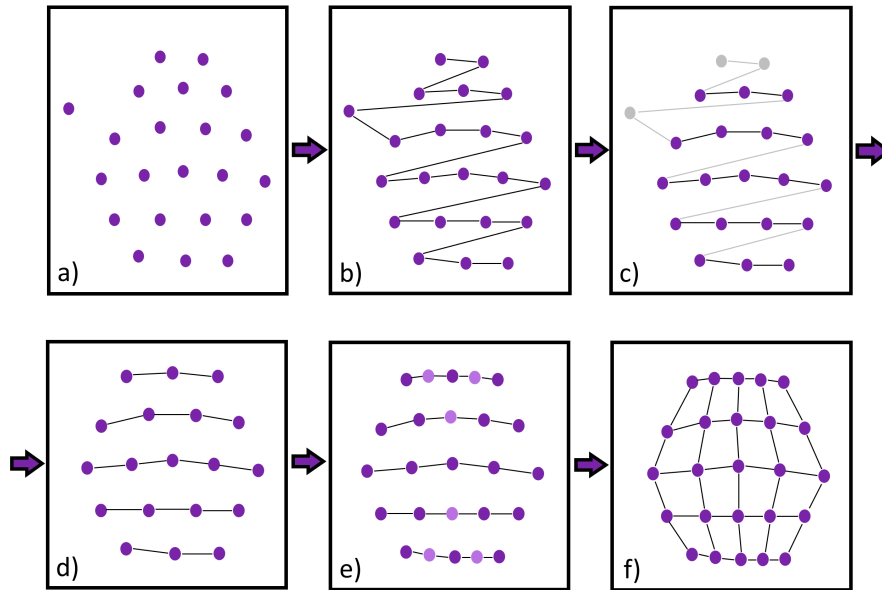


Abb. 4.2: Vereinfachte Darstellung des Modellierungsprozesses zur Tensorprodukt B-Spline-Flächengenerierung aus TLS-Daten, a) TLS-Daten, b) verbundene TLS-Daten entsprechend Scanreihenfolge, c) Definition der Kontrollpolygone, d) resultierende Kontrollpolygone, e) Neue Kontrollpunkte durch Knoteneinfügealgorithmus, f) Bidirektionale Kontrollgitterstruktur.

Im nächsten Schritt [c] werden Kontrollpolygone aus der Reihe an TLS-Punkten gebildet. Dies erfolgt durch eine Zuweisung der TLS-Punkte in einzelne Kontrollpolygone. Die Zuweisung ist an bestimmte Bedingungen geknüpft. Zwischenprodukt sind einzelne Kontrollpolygone, die unterschiedlich viele Kontrollpunkte enthalten [d]. Durch einen Knoteneinfügealgorithmus [e] können einzelne Knoten und somit Kontrollpunkte in die Kontrollpolygone eingefügt werden, ohne dass sich die Form einer hieraus

gebildeten B-Spline-Kurve vor oder nach dem Prozess ändert. Auf diese Weise kann eine bidirektionale Kontrollgitterstruktur [f] erstellt werden, die als Grundlage zur Bildung von Tensorprodukt B-Spline-Flächen dient. Unterkapitel 4.2.1 wird im Folgenden den entwickelten Vorgang der Definition von Kontrollpolygonen behandeln. Unterkapitel 4.2.2 setzt sich intensiv mit der Thematik von Knoteneinfügealgorithmen auseinander, um den angewendeten Vorgang sowie die Wahl des verwendeten Algorithmus offen zu legen und zu begründen.

4.2.1 Definition von Kontrollpolygonen

Bei der folgenden Darlegung des Prozessschritts zur Definition von Kontrollpolygonen sei darauf hinzuweisen, dass die wesentlichen Inhalte in der Publikation von Aichinger und Schwiager (2022) aufgeführt werden.

Um aus der originären Punktwolke Tensorprodukt B-Spline-Flächen bilden zu können, ist eine bidirektionale Kontrollgitterstruktur notwendig, die durch Anordnung von Kontrollpolygonen gleicher Punktzahl gewährleistet wird. Die Verwendung von Laserscanpunktwolken im ptx-Datenformat erzielt den Vorteil, dass die Laserscanpunkte bereits in Aufnahmereihenfolge der Laserscannmessung vorliegen und somit keine Sortierung der 3D-Punkte stattfinden muss. Dieser Zustand ist allerdings nur gegeben, wenn die Punktwolke von einem einzigen Laserscan-Standpunkt stammt. Sobald die zu verwendende Laserscanpunktwolke durch Registrierung mehrerer Standpunkte verknüpft vorliegt, bedingt das nachfolgende Verfahren eine Vorsortierung, in der die Punkte entsprechend ihrer Euklidischen Distanz zueinander sowie mittels Priorisierung entlang einer Achsrichtung geordnet und abgespeichert werden. Die TLS-Daten und gleichzeitig zukünftigen Kontrollpunkte für die B-Spline-Fläche liegen demgemäß in einer Matrix mit drei Spalten in aufeinanderfolgender Reihenfolge vor, wodurch eine Trennung der einzelnen Kontrollpunkte in Kontrollpolygone nicht gegeben ist. Auf dieser Basis ist demnach eine Tensorprodukt B-Spline-Flächengenerierung in Folge nicht vorhandener Kontrollgitterstrukturen unmöglich. Für eine Weiterverarbeitung der epochenweise erfassten 3D-Punktwolken zu Tensorprodukt B-Spline Flächen ist in der Konsequenz eine Definierung bzw. Zuweisung der Kontrollpunkte in Kontrollpolygone für jeden Laserscan separat durchzuführen.

Ein Ansatz zur Lösung dieser Thematik besteht in der im Folgenden beschriebenen Vorgehensweise. Aus den kartesischen Koordinaten X , Y , Z werden sphärische Koordinaten in Form des Poldistanzwinkels Θ sowie des Azimutwinkels Φ erzeugt. Die Bestimmung der Abweichungen dieser Winkel der aufeinanderfolgenden Punkte unter der Hinzunahme der Berechnung der Euklidischen Distanz zwischen nachfolgenden Punkten erzeugt drei Parameter, die zur Kontrollpolygonbildung herangezogen werden können. Punkte desselben Kontrollpolygons sollen demgemäß Parameter mit ähnlichen Abweichungen in den genannten drei Größen Diff_Θ , Diff_Φ und der Euklidischen Distanz vorweisen. Somit führt der Vergleich der Abweichungen jedes Punkts mit dem Mittelwert der vorherigen Punkte die Entscheidung herbei, ob ein neues Kontrollpolygon angelegt oder das bestehende um einen Punkt erweitert werden soll. Empirische Analysen haben bei den betrachteten Datensätzen gute Ergebnisse bei der Definition von Kontrollpolygonen ergeben, wenn bei der Entscheidung für oder gegen einen Kontrollpolygonwechsel die Parameter eines Kontrollpunktes mit dem Mittelwert der vier vorherigen Punkte des bestehenden Kontrollpolygons verglichen werden. Wenn zwei der drei Parameter Diff_Θ , Diff_Φ , Euklidische Distanz größer bzw. kleiner als eine festgelegte Ober- bzw. Untergrenze sind, erfolgt bei dieser Grenzwertüberschreitung ein Polygonwechsel zu einem neuen Kontrollpolygon. Die Festlegung der Größe der Ober- bzw. Untergrenze ist für jeden Datensatz empirisch zu ermitteln. Die Zugehörigkeit eines Laserscanpunktes zu einem bestimmten Kontrollpolygon kann mit Hilfe eines Spaltenvektors vorgenommen werden, der Zahlen, betitelt als Counter-Werte, enthält, die bei gleichem Wert auf das gleiche Kontrollpolygon zeigen.

Der Algorithmus zur Definition von Kontrollpolygonen kann in Kürze in drei Schritte zusammengefasst werden:

1. Umrechnung der kartesischen Koordinaten der TLS-Daten in sphärische Koordinaten und Weiterverwendung von Poldistanzwinkel und Azimutwinkel
2. Bestimmung der Differenzen aufeinanderfolgender Punkte bzgl. Poldistanzwinkel, Azimutwinkel und Euklidischer Distanz
3. Vergleich der Differenzen eines Punkts zum Mittelwert der vier vorhergehenden des gleichen Polygons unter Beachtung eines Grenzwerts führt zur Entscheidung, ob der Punkt zum bisherigen oder zu einem neuen Polygon definiert wird

Abbildung 4.3 zeigt eine exemplarische Punktwolke eines Fassadenstücks, das mit einer Leica HDS7000 erfasst wurde und Maße von 38 cm x 33 cm x 2 cm aufweist. Die Punktwolke besteht aus 1.850 Punkten und verfügt über einen durchschnittlichen Punktabstand von 8 mm. Die zwei Grafiken zeigen die 3D-Punktwolke von zwei Ansichten, wobei die einzelnen Punkte entsprechend ihrer Position innerhalb der abgespeicherten Matrix mittels Linien verbunden sind.

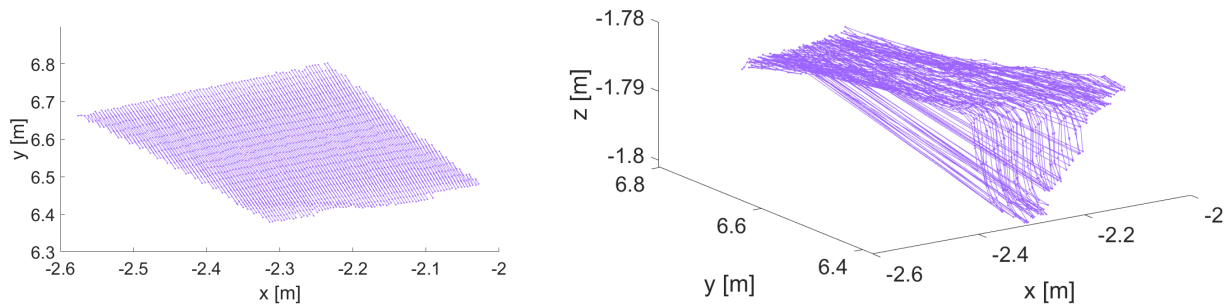


Abb. 4.3: Punktwolke eines Fassadenstücks aus zwei Ansichten. Punkte verbunden entsprechend Matrixposition.

Die linke Grafik bildet die Laserscanpunktwolke in der xy-Ebene ab, wodurch die parallel verlaufenden Verbindungslinien gut erkennbar sind. Wird diese Punktwolke allerdings aus seitlicher Perspektive betrachtet, ist mittels der Verbindungslinien erkennbar, dass auf einen Punkt am unteren Rand der Punktmenge ein Laserscanpunkt des oberen Rands der Punktwolke in der Punktmatrix folgt. Bei einer Selektion in Kontrollpolygone müssen diese Verbindungen demgemäß getrennt werden und ein neues Kontrollpolygon entsprechend der vorgestellten Methode muss erzeugt werden.

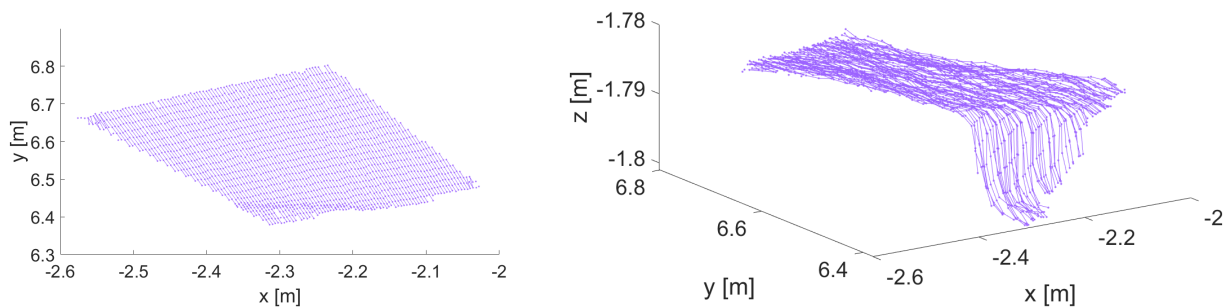


Abb. 4.4: Definierte Kontrollpolygone aus zwei Ansichten.

Abbildung 4.4 präsentiert die in Kontrollpolygone eingeteilten Kontrollpunkte der Laserscanpunktwolke der Abbildung 4.3. Deutlich erkennbar ist die strukturierte Selektion an den Rändern des Laserscans. Das Histogramm in Abbildung 4.5 stellt die Anzahl an gebildeten Kontrollpolygone der Punktzahl pro Polygon gegenüber.

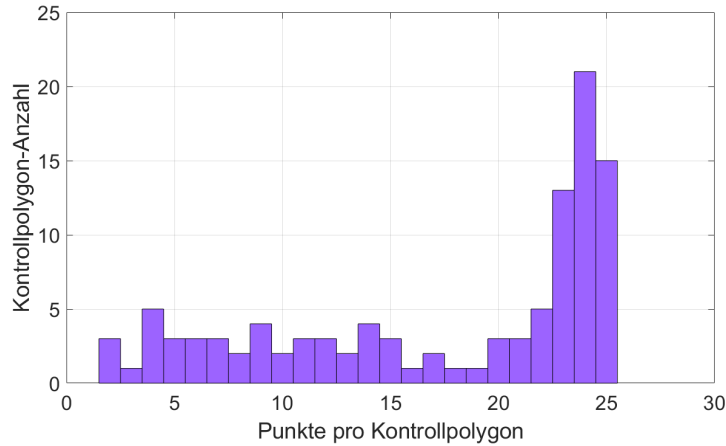


Abb. 4.5: Histogramm der definierten Kontrollpolygone eines Fassadenstücks.

Hieraus ist ablesbar, dass kein Polygon mit lediglich einem Kontrollpunkt selektiert wird, während etwa 12 Kontrollpolygone eine Punktzahl zwischen zwei bis fünf Kontrollpunkten beinhalten. 37 Kontrollpolygone bestehen aus 6 bis 20 Kontrollpunkten und 57 Polygone enthalten 21 bis 50 Kontrollpunkte.

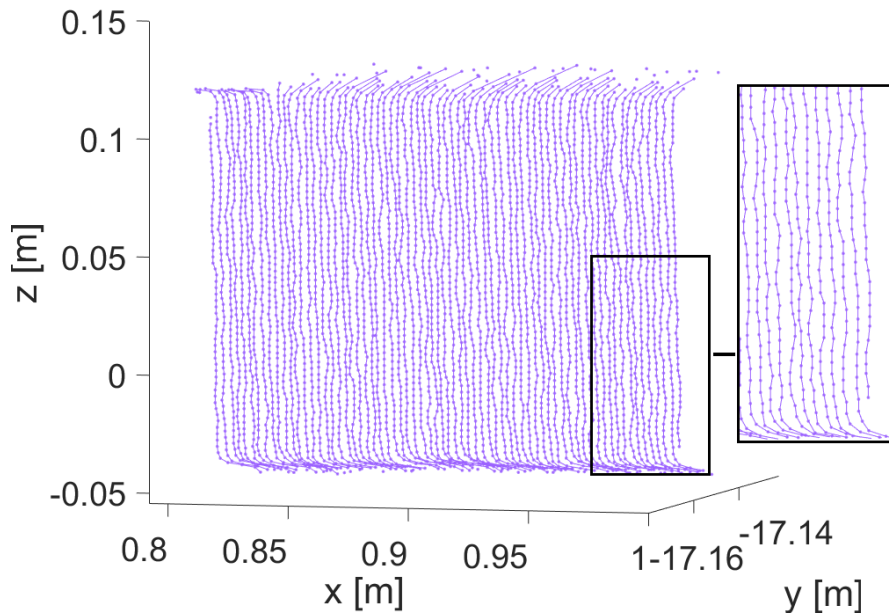


Abb. 4.6: Definierte Kontrollpolygone eines Laserscans einer Platte (Aichinger und Schwieger 2022).

Zur Überprüfung der Praktikabilität des erstellten Algorithmus, wird der Definitionsalgorithmus an Laserscans mit größeren Punktmengen ausgeführt. Zwei der getesteten Punktwolken werden im Folgenden näher beleuchtet. Der erste Laserscan stammt von einer terrestrischen Laserscanaufnahme

mit einem Leica HDS7000 einer Aluminium-Platte. Die 3D-Punktwolke weist Maße von 17 cm x 18 cm, einen durchschnittlichen Punktabstand von 3 mm und insgesamt 4.401 Punkte auf.

Die mittels des angewandten Algorithmus zugewiesenen Kontrollpolygone sind in Abbildung 4.6 dargestellt. Hierbei ist anzuführen, dass lediglich Polygone ab einer Punktzahl von größer vier Kontrollpunkten verbunden dargestellt werden, da für einen Kurvengrad von $n = 3$ mindestens vier Kontrollpunkte notwendig sind.

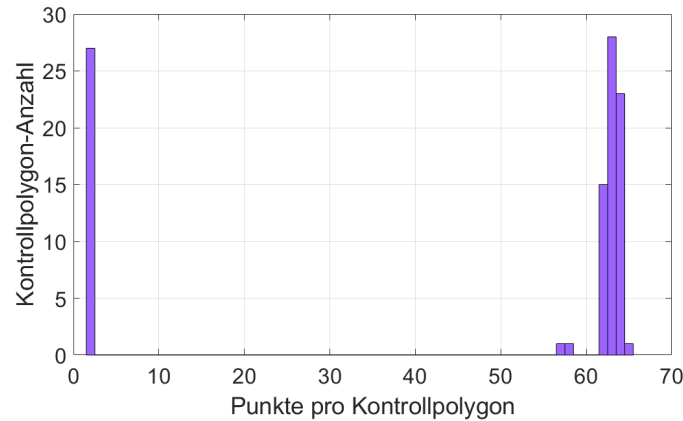


Abb. 4.7: Histogramm der definierten Kontrollpolygone (Platte).

Das zugehörigen Histogramm von Abbildung 4.7 enthält zwei Spitzen, die aufgrund von 27 Polygonen mit lediglich zwei Kontrollpunkten sowie 66 Polygonen mit 62 bis 64 Kontrollpunkten gebildet werden. Die klare Abtrennung der beiden Bereiche resultiert aus dem gleichmäßigen Verlauf der Objektform sowie der geraden Abtrennung der Punktwolke, wodurch eine Vielzahl an Kontrollpolygonen mit gleicher Punktzahl selektiert wird.

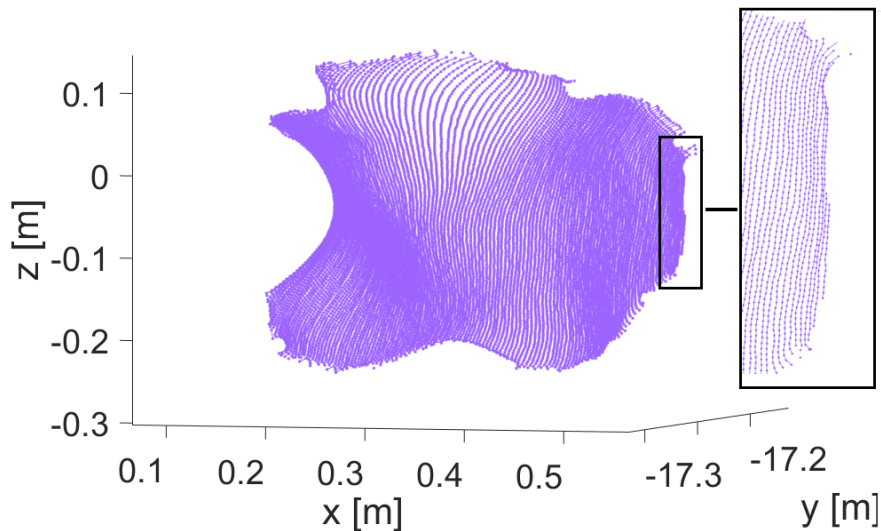


Abb. 4.8: Definierte Kontrollpolygone eines Gurkenblatt-Laserscans (Aichinger und Schwieger 2022).

Bei der Durchführung des Selektionsalgorithmus an den TLS-Daten eines Gurkenblatts, bestehend aus 15.627 Punkten, entstehen die in Abbildung 4.8 dargestellten Kontrollpolygone. Die präsentierte

Punktwolke ist wichtiger Bestandteil der Evaluationsdaten für die Entwicklung der DBKM und wird deshalb gesondert in Kapitel 6 beschrieben. Die TLS-Daten des Gurkenblatt-Testkörpers stammen aus Harmening und Neuner (2016) und Harmening (2020).

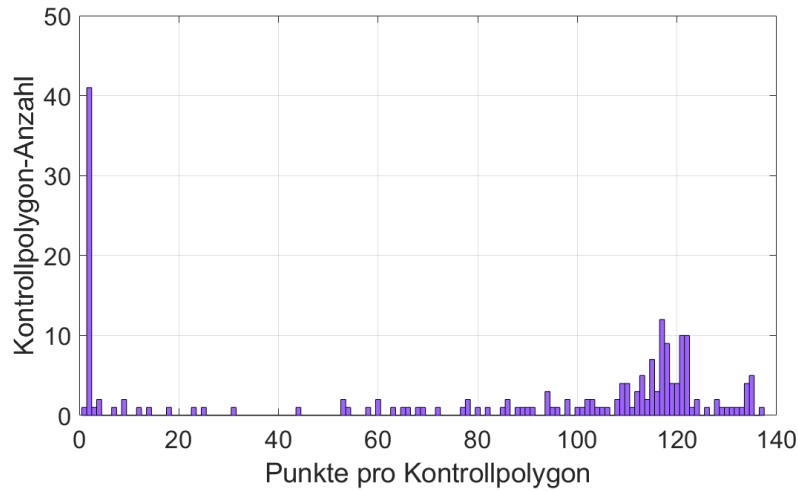


Abb. 4.9: Histogramm der definierten Kontrollpolygone (Gurkenblatt).

Das in Abbildung 4.9 veranschaulichte Histogramm zeigt, dass etwa 45 Polygone aus einem bis fünf Kontrollpunkten bestehen. Zehn Polygone enthalten zwischen 6 und 50 Punkte und 141 Polygone setzen sich aus 51 bis 150 Punkten zusammen. Die im Vergleich zur Aluminium-Platte breitere Streuung begründet sich durch die ungleichmäßige Form des Gurkenblatts-Laserscans.

Zusammenfassend ist anzuführen, dass mittels des erstellten Definitionssalgorithmus ein Werkzeug geschaffen wird, Kontrollpolygone aus einer Punktwolke zu definieren, das je nach Form der jeweiligen Punktwolke bei der Ausführung des Algorithmus eine unterschiedliche Ausprägung der zugewiesenen Kontrollpolygone im Histogramm zeigt. Je einheitlicher und geradliniger sich die Form der Laserscan-Punktwolke darstellt, desto klarer ist die Verteilung der Anzahl der Kontrollpunkte pro Kontrollpolygon. Die definierten Kontrollpolygone dienen als Basis für die Anwendbarkeit von Knoteneinfügealgorithmen, die eine Bildung von Tensorprodukt B-Spline-Flächen gewährleisten sollen.

4.2.2 Knoteneinfügealgorithmen

Um aus TLS-Daten Tensorprodukt B-Spline-Flächen generieren zu können, ist das Vorweisen eines bidirektionalen Netzes in Form einer rechteckförmigen Kontrollpunktgitterstruktur notwendig. Demgemäß müssen die Kontrollpolygone einer Richtung die gleiche Punktzahl aufzeigen. Da im Prozess der Zuweisung der Laserscanpunkte zu Kontrollpolygonen keine einheitliche Punktzahl resultiert, ist das Einfügen oder Entfernen von Kontrollpunkten notwendig. Nur so kann eine große Menge an Polygonen zur Kontrollgitterbildung genutzt und das abgetastete Objekt möglichst realitätsgetreu reproduziert werden. Hierbei soll allerdings die Form der aus den originären Laserscanpunkten generierten Kurve geometrisch unverändert bleiben.

Um dieser Anforderung Rechnung tragen zu können, wird ein Ansatz verfolgt, bei dem kein aktives Einsetzen von Kontrollpunkten erfolgt, sondern zusätzliche Knoten in den bestehenden Knotenvektor eingefügt werden. Auf diese Weise ändert sich die Anzahl der Knoten ($k + 1$) um einen Wert, wodurch, begründet durch die fundamentale Gleichung $k = m + n + 1$, entweder die Kontrollpunktanzahl ($m + 1$)

oder der Grad der Kurve n editiert werden müssen. Da eine Abwandlung des Polynomgrads zu einer Änderung des Kurvenverlaufs führt und somit nicht in Betracht kommt, erzeugt das Einfügen eines neuen Knotens einen zusätzlichen Kontrollpunkt. Mittels des Algorithmus von Boehm (1980) kann in jedem Iterationsschritt ein Knoten in den bestehenden Knotenvektor eingefügt werden. Der Oslo-Algorithmus nach Cohen et al. (1980) ermöglicht simultanes Einfügen mehrerer Knoten. Mit beiden Verfahren können somit zusätzliche Kontrollpunkte durch Einfügen neuer Knoten erzeugt werden. Um die Fragestellung zu klären, welcher der beiden Knoteneinfügealgorithmen sich für eine recheneffiziente Anwendung im Kontext der Tensorprodukt B-Spline-Flächen-Generierung anbietet, erfolgt in den nächsten Unterabschnitten eine Vorstellung des Algorithmus von Böhm und des Oslo-Algorithmus. Diese schließt ab mit einer Gegenüberstellung beider Methoden.

4.2.2.1 Algorithmus von Böhm

Basierend auf der B-Spline-Kurven-Definition von Gleichung 2.24

$$\mathbf{C}(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t), \quad t \in [u_0, u_k]$$

und einem Knotenvektor $\mathbf{U} = (u_0 \ u_1 \ \dots \ u_{k-1} \ u_k)$, wird ein neuer Knoten $\bar{u} \in [u_s, u_{s+1})$ in $\mathbf{U}$ eingesetzt, um einen Knotenvektor

$$\bar{\mathbf{U}} = \left(\bar{u}_0 = u_0 \quad \dots \quad \bar{u}_s = u_s \quad \bar{u}_{s+1} = \bar{u} \quad \bar{u}_{s+2} = u_{s+1} \quad \dots \quad \bar{u}_{k+1} = u_k \right) \quad (4.1)$$

zu generieren. Auf dieser Grundlage entsteht die Kurvendarstellung

$$\mathbf{C}(t) = \sum_{i=0}^{m+1} \mathbf{q}_i \cdot \bar{N}_i^n(t), \quad t \in [u_0, u_k] = [\bar{u}_0, \bar{u}_{k+1}]. \quad (4.2)$$

Mittels Lösen des linearen Gleichungssystems in Gleichung 4.3 nach $\mathbf{q}_i$, können die neuen Kontrollpunkte des Kontrollpolygons bestimmt werden

$$\sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t) = \sum_{i=0}^{m+1} \mathbf{q}_i \cdot \bar{N}_i^n(t). \quad (4.3)$$

Für die zur Lösung des linearen Gleichungssystems notwendige Substitution sei auf Piegler und Tiller (1997) verwiesen. Eine Überprüfung durch Induktion kann in Boehm (1980) und de Boor (2001) nachvollzogen werden.

Die Berechnung der neuen Kontrollpunkte $\mathbf{q}_i$ verläuft daraufhin nach

$$\mathbf{q}_i = (1 - \gamma_i) \cdot \mathbf{d}_{i-1} + \gamma_i \cdot \mathbf{d}_i \quad \text{mit} \quad (4.4)$$

$$\gamma_i = \begin{cases} 1 & i \leq s - n \\ \frac{\bar{u} - u_i}{u_{i+n} - u_i} & s - n < i \leq s \\ 0 & i > s + 1 \end{cases}, \quad (4.5)$$

wobei s die Knotennummerierung angibt, nach der der neue Knoten $\bar{u}$ eingefügt wird. Es gilt $\bar{u} \in [u_s, u_{s+1})$ (Vergleiche hierzu Piegler und Tiller (1997)). Nimmt γ_i den Wert 1 an, stimmt der Kontrollpunkt $\mathbf{d}_i$ mit dem neuen Kontrollpunkt $\mathbf{q}_i$ überein. Ist i größer als $(s - n)$ und kleiner gleich s , wird für γ_i ein

Parameterwert generiert, mittels dessen aus den Kontrollpunkten $\mathbf{d}_{i-1}$ und $\mathbf{d}_i$ ein neuer Kontrollpunkt $\mathbf{q}_i$ berechnet werden kann. Für i größer als $(s+1)$ entspricht der neue Kontrollpunkt $\mathbf{q}_i$ dem Kontrollpunkt $\mathbf{d}_{i-1}$ mit der vorherigen Nummerierung. Begründet wird dies hierdurch, dass sich aufgrund des einmaligen Knoteneinfügens die Kontrollpunktanzahl um einen Kontrollpunkt erhöht. Bei einem Grad n erfolgt eine Berechnung von n neuen Kontrollpunkten. Für eine recheneffiziente Umsetzung empfiehlt es sich, lediglich die Kontrollpunkte neu zu bestimmen, deren Koordinaten sich ändern. Dementsprechend werden nur die Kontrollpunkte $\mathbf{q}_i$ mit $i = s - n + 1, \dots, s$ nach Gleichung 4.4 erzeugt. Kontrollpunkte mit $i = 0, \dots, s - n$ behalten ihre Nummerierung und Position im Kontrollpunktvektor, während Kontrollpunkte mit $i = s + 1, \dots, m + 1$ um eine Stelle nach hinten verschoben werden.

Abbildung 4.10 zeigt ein Kontrollpolygon $\mathbf{d}$, bestehend aus neun Kontrollpunkten, sowie ein mittels Algorithmus von Böhm erzeugtes neues Kontrollpolygon $\mathbf{q}$, das aus zehn Kontrollpunkten besteht. Wird aus beiden Kontrollpolygonen eine B-Spline-Kurve dritten Grades erzeugt, kann festgestellt werden, dass die gleiche B-Spline-Kurve resultiert.

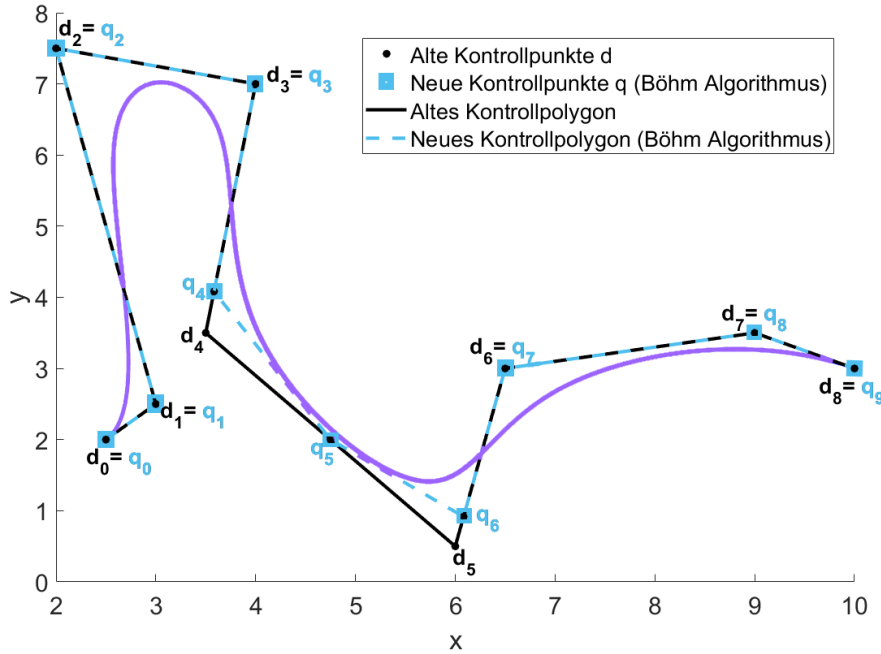


Abb. 4.10: B-Spline-Kurve dritten Grades durch Anwendung des Algorithmus von Böhm.

Um die im vorherigen Abschnitt erläuterten Vorgänge betreffend der Bildung neuer Kontrollpunkte vertieft zu erklären, sei auf den zugehörigen Knotenvektor

$$\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6) \quad (4.6)$$

sowie den einzufügenden neuen Knoten $\bar{u} = 3, 5$ zu verweisen, der den Parameterbereich $\bar{u} \in [u_6, u_7]$ und $s = 6$ angibt. Mittels $n = 3$ und $i = s - n + 1, \dots, s = 4, 5, 6$ kann berechnet werden, dass sich lediglich für die neuen Kontrollpunkte $\mathbf{q}_4, \mathbf{q}_5, \mathbf{q}_6$ Koordinatenänderungen ergeben, die über Gleichung 4.4 und 4.5 berechnet werden können. Für die restlichen Kontrollpunkte gilt $\mathbf{q}_0 = \mathbf{d}_0, \dots, \mathbf{q}_3 = \mathbf{d}_3$ und $\mathbf{q}_7 = \mathbf{d}_6, \dots, \mathbf{q}_9 = \mathbf{d}_8$. Auf diese Weise wird aus beiden Kontrollpolygonen eine identische Kurve generiert.

Übertragung des Algorithmus von Böhm zur Erstellung einer Kontrollgitterstruktur

Da nun das Prinzip des Algorithmus von Böhm ersichtlich ist, soll im Folgenden aufgeschlüsselt werden, wie die Methode zum Erzeugen von Kontrollpunkten für die Erstellung einer einheitlichen bidirektionalen Gitterstruktur zur Anwendung gebracht werden kann.

1. Aus den TLS-Rohdaten werden Kontrollpolygone durch Anwendung eines Algorithmus zur Definition von Kontrollpolygonen generiert. Die Realisierung der Definition erfolgt durch Erstellung eines Spaltenvektors mit Counter-Werten und wird in Unterkapitel 4.2.1 näher behandelt.
2. Auf Basis der Kontrollpolygone wird eine Matrix $\mathbf{D}$ spaltenweise aufgefüllt, deren Spaltenanzahl von der Anzahl der Kontrollpolygone abhängt. Die Zeilenanzahl der Matrix orientiert sich an der maximalen Anzahl der Kontrollpunkte in den vorhandenen Kontrollpolygonen.
3. Unter Einsatz des Algorithmus von Böhm findet ein polygonweises Einfügen so vieler Knoten statt, bis die maximale Anzahl an Kontrollpunkten in der Matrix erreicht ist. Der genaue Ablauf soll folgendermaßen durchgeführt werden:
 - a) Für die Kontrollpunkte $\mathbf{d}$ und einen festgelegten Kurvengrad n wird ein uniformer offener Knotenvektor $\mathbf{U}$ gebildet.
 - b) Ein neuer Knoten $\bar{u}$ wird erzeugt.
 - c) Mit Hilfe von $\bar{u}$ erfolgt eine Generierung eines neuen Knotenvektors $\bar{\mathbf{U}}$ und eines zusätzlichen Kontrollpunktes. Der bisherige Kontrollpunktvektor muss einer Änderung entsprechend den Formeln des Algorithmus von Böhm unterzogen werden.
 - d) Der in seinen Koordinaten und seiner Länge geänderte Kontrollpunktvektor ersetzt den entsprechenden ursprünglichen Vektor in der Matrix $\mathbf{D}$.
 - e) Der Vorgang muss so lange wiederholt werden bis der zu behandelnde Spaltenvektor der Kontrollpunkte vollständig aus Kontrollpunkten aufgebaut ist und somit die maximale Zeilenanzahl der Matrix $\mathbf{D}$ erreicht hat.
4. Anhand der mit Kontrollpunkten aufgefüllten Matrix $\mathbf{D}$ weist jeder Kontrollpunktvektor eine gleiche Anzahl an Kontrollpunkten auf. Auf dieser Basis kann eine Tensorprodukt B-Spline-Fläche mit dem Grad n generiert werden.

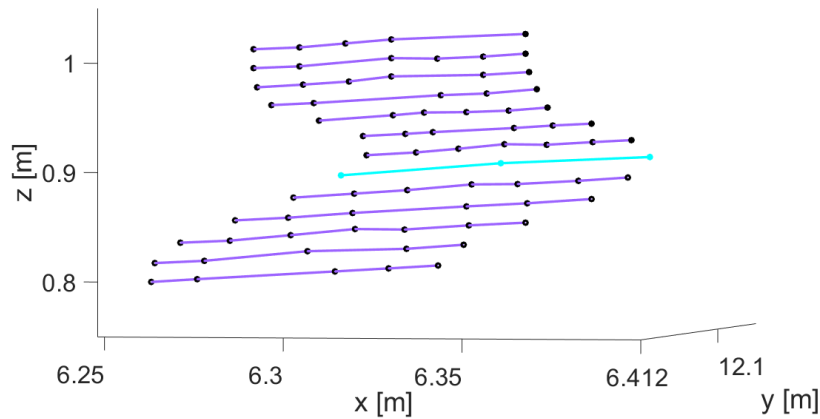


Abb. 4.11: Kontrollpolygone des Testdatensatzes. Magenta: Selektierte Kontrollpolygone, hellblau: nicht selektierte Kontrollpolygone.

Zur Verdeutlichung des beschriebenen Ablaufs soll der Vorgang mit Hilfe simulierter Laserscandaten skizziert werden. Hierbei handelt es sich um eine 3D-Punktwolke, die aus 74 Laserscanpunkten besteht und als ptx-Datei in Aufnahmereihfolge sortiert in einer Matrix der Dimension 74×3 vorliegt. Nach Durchführung des Algorithmus zur Definition von Kontrollpolygonen liegt eine Matrix der

Dimension 74×4 vor. Die vierte Spalte dieser Matrix enthält Counter-Werte, die eine Zugehörigkeit jedes Kontrollpunktes zu einem Kontrollpolygon signalisieren. Vor der Erstellung einer Matrix $\mathbf{D}$ sind die Kontrollpunkte jedes Kontrollpolygons auf eine Mindestanzahl an Punkten zu überprüfen, die notwendig ist, um die Bedingungen einer B-Spline-Kurve zu erfüllen, die mit ihrem Anfangs- und Endkontrollpunkt zusammenfällt. Hierzu ergibt sich zum Beispiel bei einem Kurvengrad von $n = 3$ eine essentielle Randvielfachheit von $n + 1 = 4$ und dementsprechend $k + 1 = 8$ Knoten in jedem Knotenvektor eines Kontrollpolygons. Im Zuge dessen resultiert bei $m = k - n - 1 = 3$ eine erforderliche Kontrollpunktanzahl von $k + 1 = 4$. In Abbildung 4.11 werden die aus den 74 Kontrollpunkten selektierten Kontrollpolygone dargestellt. Das hellblau gekennzeichnete Kontrollpolygon wird nicht in die Matrix übernommen, da die Punktanzahl des Kontrollpolygons zu gering ist. Somit erfolgt eine Aufstellung der Matrix $\mathbf{D}$ durch 12 Kontrollpolygone mit Längen zwischen 5 und 7 Kontrollpunkten, wodurch in Folge von X-, Y- und Z-Koordinaten eine Dimension von 7 Zeilen und 36 Spalten auftritt.

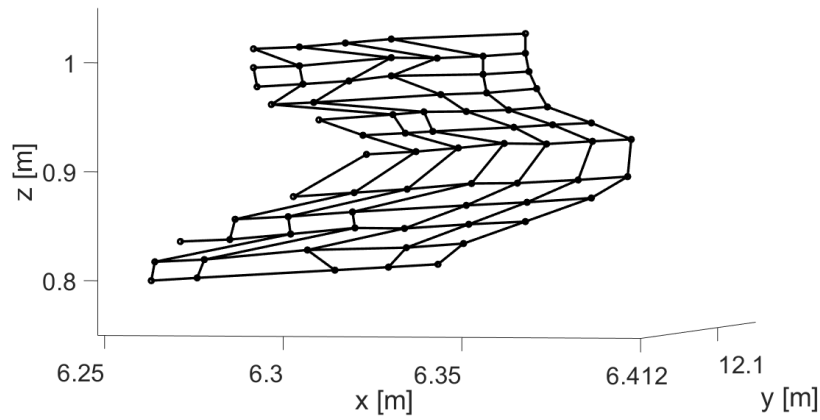


Abb. 4.12: Kontrollgitter vor Anwendung des Algorithmus von Böhm (Aichinger und Schwieger 2022).

Das hieraus entstehende Kontrollgitter kann in Abbildung 4.12 eingesehen werden. Auffällig ist die durch unterschiedliche Kontrollpunktanzahlen ausgelöste uneinheitliche Gitterstruktur.

Im weiteren Verlauf wird der Algorithmus von Böhm für jedes Kontrollpolygon so oft angewendet, bis jedes Polygon aus 7 Kontrollpunkten besteht. Hierbei ist zu beachten, dass bei der Generierung eines neuen einzusetzenden Knotens $\bar{u}$ ein Parameterwert in der Mitte des Knotenvektors gewählt wird. Eine Auseinandersetzung mit den Auswirkungen der Position des einzufügenden Knotens im Knotenvektor findet im Anhang in Unterkapitel A.1 statt.

Abbildung 4.13 veranschaulicht den iterativen Prozess der Matrixbefüllung für die aufgeführten Kontrollpunkte $\mathbf{d}_{i,j}$ des Testdatensatzes, wobei i den jeweiligen Kontrollpunkt im Kontrollpolygon j des Kontrollgitters darstellt. Die blau markierten Kontrollpunkte $\mathbf{q}_{i,j}$ sollen im Vorgang des Algorithmus von Böhm koordinatenmäßig geänderte Kontrollpunkte symbolisieren.

$$\begin{array}{c}
\begin{array}{c} \text{Algorithmus von Böhme:} \\ \text{Kontrollpolygon 1} \\ \text{Durchgang 2} \end{array} \\
\downarrow \\
D = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} & d_{1,6} & & d_{1,12} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} & d_{2,6} & & d_{2,12} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} & d_{3,6} & & d_{3,12} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} & d_{4,6} & \dots & d_{4,12} \\ d_{5,1} & d_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} & d_{5,6} & & d_{5,12} \\ - & - & d_{6,3} & d_{6,4} & d_{6,5} & d_{6,6} & & - \\ - & - & d_{7,3} & - & d_{7,5} & d_{7,6} & & - \end{bmatrix} \quad [7 \times 36]
\end{array}$$

$$\begin{array}{c}
\begin{array}{c} \text{Algorithmus von Böhme:} \\ \text{Kontrollpolygon 1} \\ \text{Durchgang 1} \end{array} \\
\downarrow \\
D = \begin{bmatrix} q_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} & d_{1,6} & & d_{1,12} \\ q_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} & d_{2,6} & & d_{2,12} \\ q_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} & d_{3,6} & & d_{3,12} \\ q_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} & d_{4,6} & \dots & d_{4,12} \\ q_{5,1} & d_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} & d_{5,6} & & d_{5,12} \\ q_{6,1} & - & d_{6,3} & d_{6,4} & d_{6,5} & d_{6,6} & & - \\ q_{7,1} & - & d_{7,3} & - & d_{7,5} & d_{7,6} & & - \end{bmatrix} \quad [7 \times 36]
\end{array}$$

$$\begin{array}{c}
\begin{array}{c} \text{Algorithmus von Böhme:} \\ \text{Kontrollpolygon 2} \\ \text{Durchgang 1} \end{array} \\
\downarrow \\
D = \begin{bmatrix} q_{1,1} & q_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} & d_{1,6} & & d_{1,12} \\ q_{2,1} & q_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} & d_{2,6} & & d_{2,12} \\ q_{3,1} & q_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} & d_{3,6} & & d_{3,12} \\ q_{4,1} & q_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} & d_{4,6} & \dots & d_{4,12} \\ q_{5,1} & q_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} & d_{5,6} & & d_{5,12} \\ q_{6,1} & q_{6,2} & d_{6,3} & d_{6,4} & d_{6,5} & d_{6,6} & & - \\ q_{7,1} & - & d_{7,3} & - & d_{7,5} & d_{7,6} & & - \end{bmatrix} \quad [7 \times 36]
\end{array}$$

$$\vdots$$

$$D = \begin{bmatrix} q_{1,1} & q_{1,2} & d_{1,3} & q_{1,4} & d_{1,5} & d_{1,6} & & q_{1,12} \\ q_{2,1} & q_{2,2} & d_{2,3} & q_{2,4} & d_{2,5} & d_{2,6} & & q_{2,12} \\ q_{3,1} & q_{3,2} & d_{3,3} & q_{3,4} & d_{3,5} & d_{3,6} & & q_{3,12} \\ q_{4,1} & q_{4,2} & d_{4,3} & q_{4,4} & d_{4,5} & d_{4,6} & \dots & q_{4,12} \\ q_{5,1} & q_{5,2} & d_{5,3} & q_{5,4} & d_{5,5} & d_{5,6} & & q_{5,12} \\ q_{6,1} & q_{6,2} & d_{6,3} & q_{6,4} & d_{6,5} & d_{6,6} & & q_{6,12} \\ q_{7,1} & q_{7,2} & d_{7,3} & q_{7,4} & d_{7,5} & d_{7,6} & & q_{7,12} \end{bmatrix} \quad [7 \times 36]$$

Abb. 4.13: Erzeugung von Kontrollpunkten mittels Algorithmus von Böhme.

Nach erfolgreicher Durchführung von Böhm's Algorithmus erhält man eine bidirektionale Gitterstruktur, mit deren Hilfe sich eine Tensorprodukt B-Spline-Fläche aus 7×12 Kontrollpunkten bestimmen lässt. Das genannte Kontrollgitter ist in Abbildung 4.14 ersichtlich.

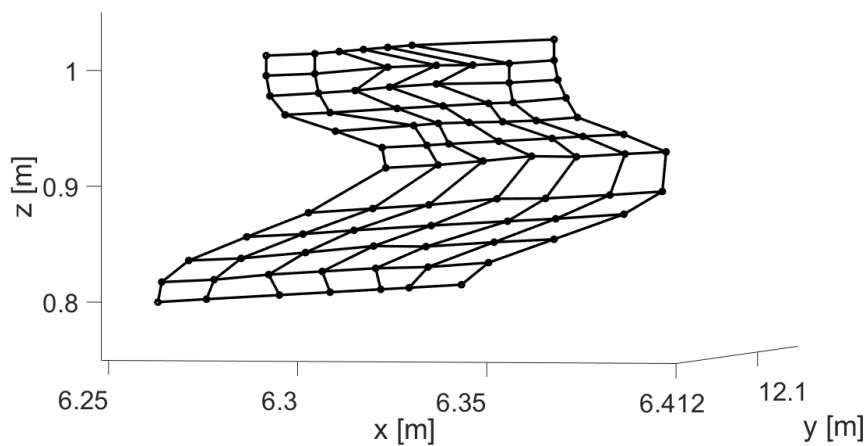


Abb. 4.14: Kontrollgitter nach Anwendung des Algorithmus von Böhme (Aichinger und Schwiager 2022).

Aus dem konstruierten Kontrollgitter wird als letzter Schritt eine B-Spline-Fläche dritten Grades, dargestellt in Abbildung 4.15, berechnet.

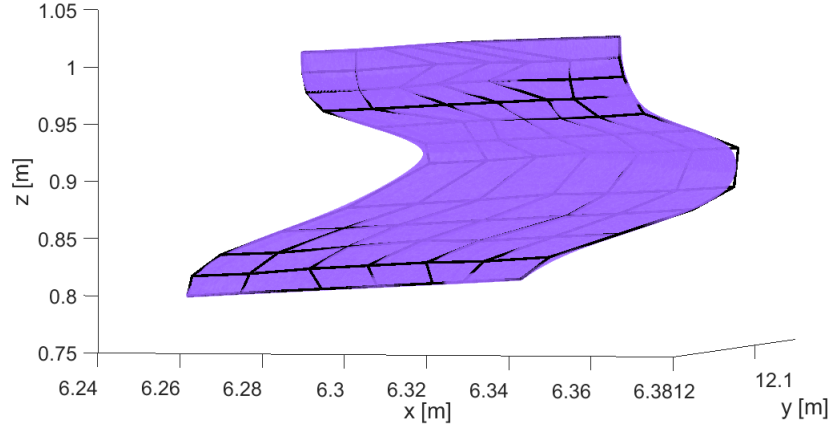


Abb. 4.15: Tensorprodukt B-Spline-Fläche dritten Grades und Kontrollgitter.

Wird mit großen Datenmengen gearbeitet, ist zu berücksichtigen, dass für jeden neuen Kontrollpunkt ein gesonderter Berechnungsdurchgang erforderlich ist, wobei jedes Mal ein neuer Knotenvektor $\bar{\mathbf{U}}$ sowie neue Kontrollpunkte $\mathbf{q}_{i,j}$ bestimmt werden müssen. Bei großen Matrizen kann dies mit ausgeprägten Rechenintensitäten korrelieren. In Folge dessen soll in Abschnitt 4.2.2.2 ein Verfahren vorgestellt werden, dass ein Einfügen mehrerer Knoten im selben Vorgang gestattet.

4.2.2.2 Oslo-Algorithmus

Der Oslo-Algorithmus wird von Cohen, Lyche und Riesenfeld (1980) in der norwegischen Hauptstadt Oslo etwa zeitgleich zum Algorithmus von Böhm entwickelt. Er ermöglicht unter Nutzung der alten Kontrollpunkte $\mathbf{d}_i$, des ursprünglichen Knotenvektors $\mathbf{U}$ sowie einem neuen verfeinerten Knotenvektor $\bar{\mathbf{U}}$ die Generierung neuer Kontrollpunkte $\mathbf{q}_j$, sodass der Formverlauf einer existierenden nicht uniformen B-Spline-Kurve erhalten bleibt.

In der Folgezeit der Veröffentlichung des Oslo-Algorithmus entstehen eine Fülle an darauf aufbauender Berechnungswege, Beweisen sowie Ansätzen, zu denen zum Beispiel folgende gehören: de Boor und Höllig (1985), Goldman (1990), Lee (1985) und (1989), Lyche (1988), Prautzsch (1984) und Seidel (1988). Eine ebenso verbreitete Methode nennt sich Blossoming und geht zurück auf Ramshaw (1987). Das Blossoming-Prinzip verfolgt im Wesentlichen die Idee, einer Konvertierung eines Polynoms vom Grad n in ein äquivalentes Polynom mit n Variablen vom Grad 1, das Blossom oder Polarform genannt wird. Die Art des in dieser Arbeit verwendeten Oslo-Algorithmus bezieht sich auf den von Cohen, Lyche und Riesenfeld (1980) ursprünglich entwickelten Ansatz. Dieser soll im Folgenden vorgestellt werden.

Gegeben sei die B-Spline-Kurven-Definition von Gleichung 2.24:

$$\mathbf{C}(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t), \quad t \in [u_0, u_k],$$

mit einem Knotenvektor $\mathbf{U} = (u_0 \ u_1 \ \dots \ u_{k-1} \ u_k)$, einem Kontrollpunktvektor $\mathbf{d}$ mit $(m+1)$ Kontrollpunkten und den B-Spline-Basisfunktionen $N_i^n(t)$ vom Grad n . Durch Einfügen von Knoten in

den vorhandenen Knotenvektor wird ein neuer Knotenvektor $\bar{U} = (\bar{u}_0 \ \bar{u}_1 \ \dots \ \bar{u}_{f-1} \ \bar{u}_f)$ mit $(f+1)$ Knoten erzeugt, mit dessen Hilfe die Kurvendarstellung

$$C(t) = \sum_{j=0}^g \mathbf{q}_j \cdot \bar{N}_j^n(t), \quad t \in [\bar{u}_0, \bar{u}_f], \quad (4.7)$$

bestehend aus $(g+1)$ neuen Kontrollpunkten $\mathbf{q}$, gebildet werden kann. Unter der Bedingung, dass Gleichung 4.7 Gleichung 2.24 entsprechen muss, lassen sich die neuen Kontrollpunkte wie folgt berechnen:

$$\mathbf{q}_j = \sum_{i=0}^m \alpha_{j,i}^n \cdot \mathbf{d}_i, \quad (4.8)$$

wobei $j = 0, \dots, g$ mit $g = f - n - 1$ und $i = 0, \dots, m$ mit $m = k - n - 1$ gilt. In Anlehnung an eine rekursive Darstellung für uniforme Knotenvektoren nach Lyche (1976) bestimmen Cohen et al. (1980) und Prautzsch (1984) die Parameterwerte für $\alpha_{j,i}^n$ mittels der Rekursionsformeln in Gleichung 4.9 und 4.10. Für $n = 0$ wird folgende Berechnung ausgeführt:

$$\alpha_{j,i}^0 = \begin{cases} 1 & u_i \leq \bar{u}_j < u_{i+1} \\ 0 & \text{sonst} \end{cases}, \quad (4.9)$$

während für $n \geq 1$ gilt

$$\alpha_{j,i}^n = \frac{\bar{u}_{j+n} - u_i}{u_{i+n} + u_i} \cdot \alpha_{j,i}^{n-1} + \frac{u_{i+n+1} - \bar{u}_{j+n}}{u_{i+n+1} - u_{i+1}} \cdot \alpha_{j,i+1}^{n-1}, \quad (4.10)$$

wobei für $j = 0, \dots, f - n - 1$ und $i = 0, \dots, k - n - 1$ angenommen wird. Löst man die angegebene Rekursionsformel mittels Matrizenberechnung, müssen bis zum Erhalt der Matrix α^n ($n+1$) Stufen durchlaufen werden. Stufe 0 besteht aus der Aufstellung der Matrix α^0 mittels Gleichung 4.9, die sich aus f Zeilen und k Spalten zusammensetzt. Gleichung 4.10 liefert die Grundlage, um anhand Stufe 1 die Matrix α^1 mit der Dimension $[(f-1) \times (k-1)]$ aufzustellen. In Stufe 2, mit Annahme von $n = 2$, wird in der Konsequenz die Matrix α^2 mit $(f-2)$ Zeilen und $(k-2)$ Spalten realisiert. Diese Abfolge wiederholt sich bis zur Stufe n , in der die Matrix α^n erzeugt wird, die zur Anwendung von Gleichung 4.8 ermöglicht. Mittels der im Zuge dessen gebildeten neuen Kontrollpunkten kann eine B-Spline-Kurve $\mathbf{K}(s)$ erzeugt werden, deren Darstellung in Formgleichheit mit derer von $C(t)$ resultiert.

Zur näheren Veranschaulichung, wird ein Beispiel unter Verwendung des identischen Kontrollpolygons und Polynomgrads wie in Abschnitt 4.2.2.1 (Algorithmus von Böhm) ausgeführt. Der ursprünglichen B-Spline-Kurve $C(t)$ vom Grad $n = 3$ liegt ein Knotenvektor

$$U = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 6 & 6 \\ u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} \end{pmatrix}, \quad (4.11)$$

sowie $(m+1) = 9$ Kontrollpunkte $\mathbf{d}_0$ bis $\mathbf{d}_8$ zu Grunde. Auf Basis der Einträge des Knotenvektors U erfolgt die Annahme von $k = 12$. In den vorhandenen Knotenvektor werden die Knotenwerte 3,25, 3,50 und 3,75 eingefügt, womit der folgende neue verfeinerte Knotenvektor $\bar{U}$ entsteht:

$$\bar{U} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 3,25 & 3,50 & 3,75 & 4 & 5 & 6 & 6 & 6 & 6 \\ \bar{u}_0 & \bar{u}_1 & \bar{u}_2 & \bar{u}_3 & \bar{u}_4 & \bar{u}_5 & \bar{u}_6 & \bar{u}_7 & \bar{u}_8 & \bar{u}_9 & \bar{u}_{10} & \bar{u}_{11} & \bar{u}_{12} & \bar{u}_{13} & \bar{u}_{14} & \bar{u}_{15} \end{pmatrix} \quad (4.12)$$

und demgemäß $f = 15$ vorgibt. Aus $(f + 1) = 16$ neuen Knoten und der Bedingung von Gleichung 2.21 mit $g = f - n - 1$ müssen sich $(g + 1) = 12$ neue Kontrollpunkte $\mathbf{q}$ ergeben. Zur Berechnung dieser Kontrollpunkte ist $\alpha_{j,i}^n$ in vier Stufen zu ermitteln. Stufe 0 erzeugt eine Matrix α^0 mit der Dimension $[15 \times 12]$. Durch Verwendung der in Stufe 0 ermittelten Parameter, kann α^1 , bestehend aus 14 Zeilen und 11 Spalten, berechnet werden. Nach Anwendung der Rekursionsformel für Stufe 2 und der Generierung der Matrix α^2 mit der Dimension $[13 \times 10]$, erfolgt die finale Berechnung von α^3 , die anhand der Dimension $[12 \times 9]$ mit der Matrix der Kontrollpunkte $\mathbf{d}$, bestehend aus 9 Zeilen und 2 Spalten, einer Matrixmultiplikation unterzogen werden kann. Das Resultat eröffnet die gesuchten 12 Kontrollpunkte $\mathbf{q}$, die in Abbildung 4.16 als blau umrandete Punkte das neue Kontrollpolygon konstruieren. Sowohl die B-Spline-Kurve aus den alten Kontrollpunkten wie auch aus den neu berechneten Kontrollpunkten entspricht derselben Kurvendarstellung.

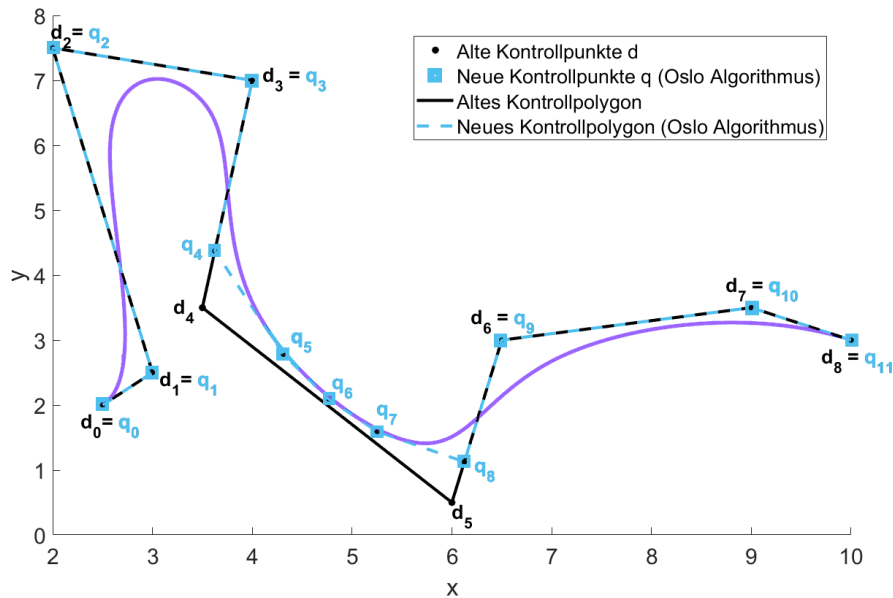


Abb. 4.16: B-Spline-Kurve dritten Grades durch Anwendung des Oslo-Algorithmus.

Im Vergleich zum Algorithmus von Böhm kann festgestellt werden, dass das Einfügen beliebig vieler zusätzlicher Knoten sowie die Erzeugung hiervon abhängiger neuer Kontrollpunkte in einem Berechnungskomplex simultan stattfinden kann, während im Algorithmus von Böhm für jeden neu eingefügten Knoten ein separater Berechnungsvorgang durchgeführt werden muss. Allerdings besteht hierbei lediglich die Notwendigkeit der Berechnung von n neuen Kontrollpunkten mittels n Affinkombinationen. Bei Ausführung des Oslo-Algorithmus in der zuvor aufgeführten Matrizenberechnung, werden, unabhängig der Anzahl der einzufügenden Knoten, $(n + 1)$ Matrizen mit maximalen Dimensionen von $[f \times k]$ erzeugt. Bei großen Datenmengen kann dies zu erhöhtem Rechenaufwand führen, der sich lediglich relativiert, wenn eine wesentlich große Menge an neuer Knoten eingefügt werden muss.

Die angesprochene Rechenzeit könnte beim Oslo-Algorithmus verringert werden, wenn nach jeder Stufenberechnung eine Selektion bezüglich der Matrixelemente ungleich Null erfolgt und ausschließlich diese Werte zur Weiterberechnung der nachfolgenden Stufen herangezogen werden. Somit würden je höher die Stufe, desto weniger Affinkombinationen durchgeführt werden müssen. Zu hinterfragen ist jedoch, ob der Rechenaufwand der Selektion in der jeweiligen Matrix die ersparte Rechenleistung derart abmildert, dass eine abgeschwächte Zeitersparnis resultiert.

4.2.2.3 Gegenüberstellung von Knoteneinfügealgorithmen

Dieser Fragestellung wird anhand der Auswertung mehrerer TLS-Punktwolken nachgegangen, wovon vier repräsentative Punktwolken in Abbildung 4.17 dargestellt sind. Zusätzlich zu den bereits vorgestellten Punktwolken des Fassadenstücks, der Platte sowie des Gurkenblatts werden die Laserscandaten einer Ebene mit den Maßen 49 cm x 72 cm hinzugezogen, die mit einer Leica HDS7000 erfasst wurde und über einen durchschnittlichen Punktabstand von 4 mm verfügt.

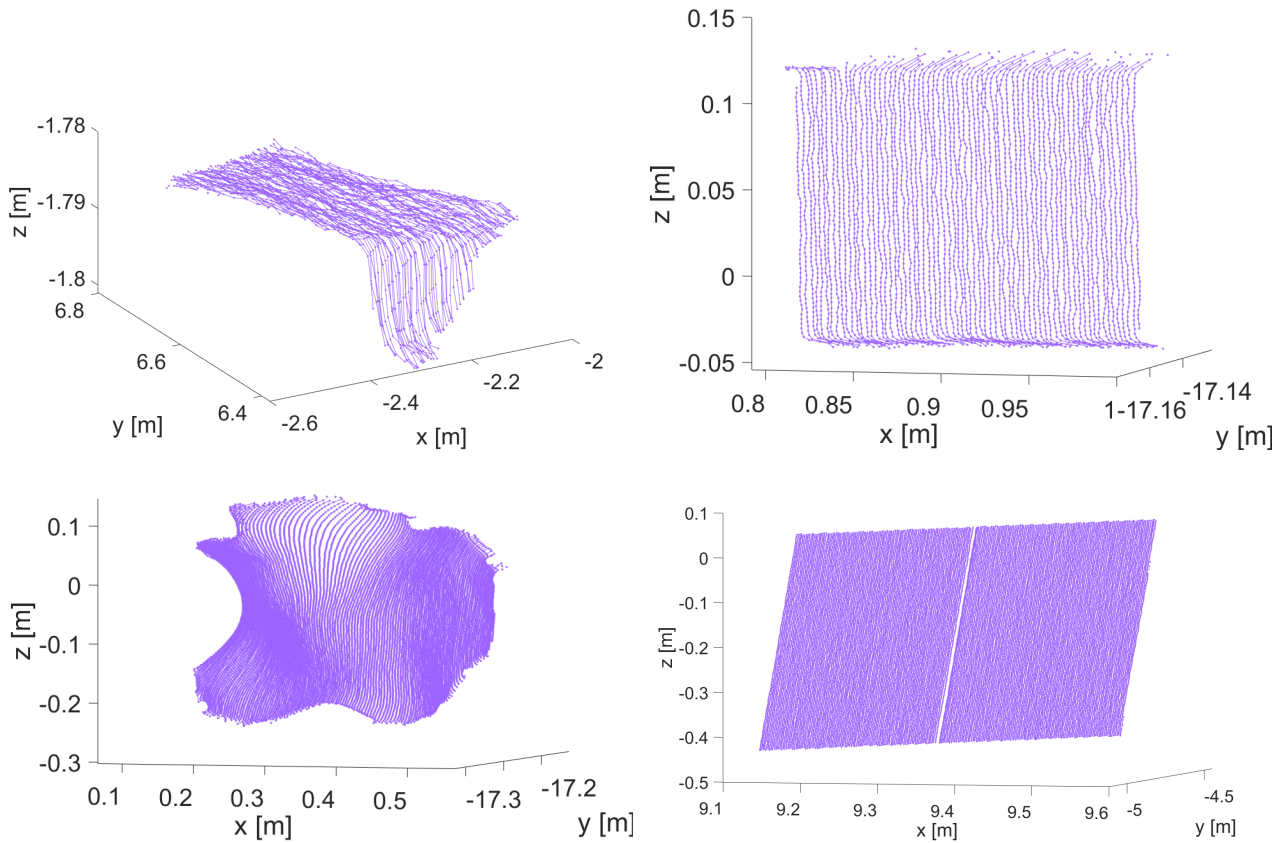


Abb. 4.17: Punktwolken eines Fassadenstücks (oben links), einer Platte (oben rechts), eines Gurkenblatts (unten links) und einer Ebene (unten rechts).

Die vier in Abbildung 4.17 präsentierten Punktwolken dienen ausschließlich der Gegenüberstellung von Knoteneinfügealgorithmen und der Modellierung von TLS-Daten mittels B-Splines und sollen, mit Ausnahme des Datensatzes des Gurkenblatts, nicht für die Anwendung der DBKM weiterverwendet werden. Als rechenzeitsparende Variante des Oslo-Algorithmus wird der ursprüngliche Oslo-Algorithmus derart modifiziert, dass statt der wiederholten Berechnung der ganzen Matrix $\mathbf{a}^n$, lediglich die Spalten identifiziert werden, die einen Parameter größer Null enthalten. Die Berechnung der Matrix $\mathbf{a}^n$ der nächsten Stufe erfolgt daraufhin nur mittels der erheblich reduzierten Anzahl an übergebenen Spalten und nicht wie beim Oslo-Algorithmus durch Benutzung der jeweils gesamten Matrix. Eine Variante bei der ausschließlich die Werte größer Null identifiziert und zur weiteren Berechnung übergeben werden, wird hierbei nicht gewählt, da die spalten- und zeilenbezogene Abspeicherung und Zuweisung im Verlauf von $(n + 1)$ Stufen in Testläufen größere Rechenzeiten in Anspruch genommen hat, als eine ausschließliche spaltenweise Vorgehensweise.

Zur Beurteilung der Effizienz sowie Rechenzeit des Algorithmus von Böhm, Oslo-Algorithmus und des in veränderter Form implementierten Oslo-Algorithmus wird die Matrix $\mathbf{D}$ für die Punktwolken zweier planarer Objekte sowie zweier ungleichmäßig verlaufender Objekte gebildet, die in Abbildung 4.17 dargestellt sind. Dies bedeutet, dass mittels Knoteneinfügealgorithmus eine Matrix $\mathbf{D}$, bestehend aus Kontrollpunkten gleicher Anzahl generiert wird, um eine rechteckförmige Gitterstruktur zu erhalten. Für die Durchführung des Vergleichs wird ein HP Omen 17-an016ng mit einem Intel Core i7-7700HQ Prozessor, 32GB RAM und einer NVIDIA GeForce GTX 1070 Grafikkarte verwendet. Die Ergebnisse dieses Vergleichs sind in Tabelle 4.1 aufgelistet und werden im Folgenden diskutiert.

Objektart	Fassadenstück		Platte		Gurkenblatt		Ebene	
Punktzahl	1.850		4.401		15.627		32.470	
Objektform	ungleichmäßig		gleichmäßig		ungleichmäßig		gleichmäßig	
	Zeit [s]	Differenz zu Böhm	Zeit [s]	Differenz zu Böhm	Zeit [s]	Differenz zu Böhm	Zeit [s]	Differenz zu Böhm
Algorithmus v. Böhm	0,016		0,005		0,129		0,012	
Oslo Algorithmus	0,040	+ 250 %	0,119	+ 2380 %	0,462	+ 358 %	0,294	+ 2450 %
Oslo Algorithmus (geänd. Implementierung)	0,041	+ 256 %	0,125	+ 2500 %	0,474	+ 367 %	0,304	+ 2533 %

Tab. 4.1: Gegenüberstellung der Rechenzeiten vom Algorithmus von Böhm, Oslo-Algorithmus und Oslo-Algorithmus mit geänderter Implementierung.

Als Objektart mit der geringsten Punktzahl von 1.850 wird eine Punktwolke eines Fassadenstücks mit ungleichmäßigem Formverlauf verwendet. Während die Laufzeit beim Algorithmus von Böhm lediglich 0,016 s beträgt, stellt sie sich beim Oslo-Algorithmus sowie der geänderten Implementierungsform mehr als doppelt so groß mit 0,040 s und 0,041 s dar. Dies entspricht einer Zunahme der Rechenzeit in Differenz zu Böhm von 250 % und 256 %. Beim Vergleich des Oslo-Algorithmus in Relation zum andersartig implementierten Oslo-Algorithmus, lässt sich, wie bereits vermutet, keine Rechenzeiterparnis feststellen, da der Prozess des Auffindens von Werten größer Null derart viel Laufzeit in Anspruch nimmt, dass sogar eine Erhöhung der Rechenzeit resultiert. In Vorwegnahme, beansprucht die geänderte Implementierungsform des Oslo-Algorithmus bei den vier betrachteten Punktwolken zwischen 2,4 % und 4,8 % mehr Zeit und bringt damit keinen Gewinn.

Bei Heranziehung eines gleichmäßig verlaufenden Objekts einer Aluminium-Platte mit 4.401 Laserscan-Punkten wird die Rechenzeit des Böhm Algorithmus von nur 0,005 s erreicht, während die zwei Arten des Oslo-Algorithmus 0,119 s und 0,125 s benötigen. Diese Erhöhung der Laufzeit um 2380 % und 2500 % kann damit begründet werden, dass die definierten Kontrollpolygone der Platte bereits vor der Ausführung eines Knoteneinfügealgorithmus überwiegend die gleiche Menge an Punkten beinhalten. Demnach sind innerhalb des Algorithmus von Böhm nur wenige Iterationsschritte notwendig, um eine gleichmäßige Kontrollgitterstruktur zu erzielen. Hierdurch erfolgt beinahe keine Einsparung der Rechenzeit durch den Oslo-Algorithmus, der die einzufügenden Knoten in einem Schritt generiert, allerdings für diese Berechnung eine Abfolge mehrere Bestimmungen von Matrizen verlangt. Bei Betrachtung der Punktwolke eines Gurkenblatts mit 15.626 Punkten zeigt sich der Algorithmus von Böhm um 358 % und 367 % schneller als die zwei Arten des Oslo-Algorithmus. Aufgrund der ungleichmäßigen Objektform liegen die Kontrollpolygone vor Durchführung der Algorithmen verstärkt mit unterschiedlicher Punktzahl vor, wodurch es beim Algorithmus von Böhm vieler Iterationsschritte bedarf, um Kontrollpolygone mit einer geringen Anzahl an Kontrollpunkten aufzufüllen, während dies innerhalb des Oslo-Algorithmus in einem Vorgang erfolgt. Als letztes Punktwolkenobjekt wird eine Ebene, bestehend aus 32.470 Punkten, herangezogen. Die gleichmäßige Objektform erzielt, wie im Falle der Platte, eine starke Bevorteilung des Algorithmus von Böhm, der sich mit 0,012 s in Relation zum Oslo-Algorithmus mit 0,294 s um 2450 % als schneller herausstellt.

Zusammenfassend lässt sich erörtern, dass sich der Algorithmus von Böhm als schneller und einfacher in der Implementierung und somit effizienter zeigt als die nach Cohen verwendete Methode des Oslo-Algorithmus. Dies kommt besonders bei gleichmäßig verlaufenden Objektformen zum Ausdruck. Ein Selektieren von Werten größer Null erbringt beim Oslo-Algorithmus zudem keinen Vorteil. Das resultierende Ergebnis zwischen Algorithmus von Böhm und dem Oslo-Algorithmus bestätigt sich in der Veröffentlichung von Boehm (1985), in der ein Vergleich des Oslo-Algorithmus nach Cohen et al. (1980) mit dem Knoteneinfügealgorithmus nach Boehm (1980) bezüglich der Effizienz stattfindet. Hierbei wird festgestellt, dass sich der Algorithmus von Böhm aufgrund der geringeren Menge an Linearkombinationen mindestens gleichwertig oder höherwertiger, auch im Falle von Flächen, darstellt.

Obwohl das Resultat an dieser Stelle den Böhm Algorithmus priorisierend hervorhebt, soll darauf hingewiesen werden, dass in Folge der starken Diversität an unterschiedlichen Methoden zum Oslo-Algorithmus keine absolut gültige Aussage erhoben wird. Für den weiteren Fortgang wird allerdings auf den Algorithmus von Böhm zurückgegriffen.

4.3 Mapcut-Algorithmus (Zielzeichen-freie Registrierung)

Um mittels B-Spline-Technik modellierte TLS-Daten vergleichen zu können, ist eine Registrierung der Epochen zueinander notwendig. Das in dieser Arbeit entwickelte lokale Zielzeichen-freie Registrierungsverfahren namens Mapcut-Algorithmus verwendet Krümmungsparameter als Merkmalsdeskriptor, die in einer Korrelationsanalyse der Korrespondenzsuche dienen. Derzeit ist Mapcut als Feinregistrierungsverfahren implementiert. Eine Erweiterung für eine Grob- und Feinregistrierung ist durch Anpassungen möglich und wird für zukünftige Arbeiten vermerkt. Der entwickelte Mapcut-Algorithmus wird durch ein iteratives vierstufiges Verfahren realisiert. In den folgenden Abschnitten soll der Mapcut-Algorithmus zuerst für B-Spline-Kurven und im Anschluss für B-Spline-Flächen vorgestellt werden.

4.3.1 Mapcut-Algorithmus (B-Spline-Kurven)

Der Ablauf des Mapcut-Algorithmus ist für B-Spline-Kurven in Abbildung 4.18 dargestellt und soll zunächst grob umrissen werden. In den nachfolgenden Abschnitten wird auf die Einzelkomponenten und deren Funktionsweise im Detail eingegangen. Ein Grob- und Feinmapping mittels einer Korrelationsanalyse der Krümmungen von B-Spline-Kurvenpunkten bildet Stufe I und II. Hierbei werden einander zugehörige B-Spline-Kurvenpunkte zweier Epochen identifiziert. Stufe III besteht aus einer Selektion annähernd identischer Punkte in beiden Systemen und der Koordinatentransformation der Laserscandaten einer Epoche 1 in das System von Epoche 0.

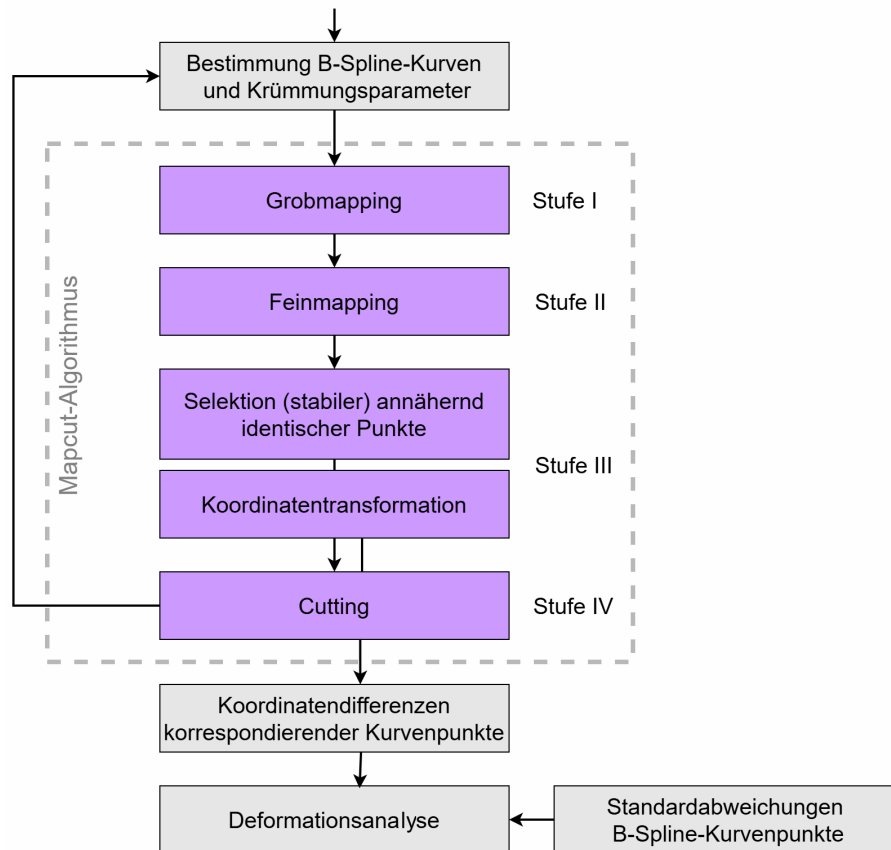


Abb. 4.18: Ablauf des Mapcut-Algorithmus für B-Spline-Kurven im Kontext der DBKM.

Stufe IV bildet ein sogenanntes Cutting, in dem die Ränder der Punktwolken zweier Epochen angeglichen bzw. ausgestanzt werden. Der enorme Vorteil des Cuttings wird in diesem Abschnitt ebenso erläutert. Durch eine iterative Vorgehensweise von Stufe I bis IV können die B-Spline-Kurven zweier Epochen derart ausgerichtet und in den Randbereichen angeglichen werden, dass ein direkter Koordinatenvergleich von zugehörigen B-Spline-Kurvenpunkten zweier Epochen hinsichtlich Deformationen ausgeführt werden kann.

Grobmapping und Feinmapping (Stufe I und II)

Gegeben sind die Kontrollpolygone zweier Epochen, die in einem gewissen Bereich dieselbe Objektform diskretisieren. Anhand der Kontrollpolygone, einer vorgegebenen Schrittweite und einem Grad n kann jeweils eine B-Spline-Kurve generiert werden. Daraufhin wird die 2. Ableitung sowie die Krümmung entsprechend Abschnitt 2.2.3 für jeden Kurvenpunkt erzeugt. Stufe I und II des entwickelten Mapcut-Algorithmus bestehen aus einem Vergleich der Krümmungen der B-Spline-Kurvenpunkte mit Hilfe einer Korrelationsanalyse. Die Korrelationsanalyse hat das Ziel zugehörige Kurvenpunkte in zwei Epochen zu identifizieren, um in Stufe III in ein gemeinsames Koordinatensystem zu transformieren.

Die Basisformeln der Korrelationsanalyse sollen im Folgenden erläutert werden, um darauf aufbauend Gleichungen für einen Korrelationskoeffizienten aufzustellen.

Das Korrelationsintegral für Zeitfunktionen, bei dem sich die Betrachtung der statistischen Abhängigkeiten auf zwei stationäre Funktionen $x(t)$ und $y(t)$ gründet, nennt sich Kreuzkorrelationsfunktion und lautet wie folgt:

$$R_{x,y}(\tau) = \int_{-\infty}^{\infty} x(t)y(t+\tau)dt, \quad (4.13)$$

wobei τ der Zeitverschiebung beider Signale entspricht (Möser 2018). In Analogie zu Gleichung 4.13 kann die Kreuzkorrelationsfunktion für zwei Krümmungssignale $\kappa_1(j)$ und $\kappa_2(j+k)$, die um eine räumliche Verschiebung k differieren mit:

$$R_{\kappa_1,\kappa_2}(k) = \int_{-\infty}^{\infty} \kappa_1(j)\kappa_2(j+k)dj \quad (4.14)$$

beschrieben werden. Bringt die Kreuzkorrelationsfunktion bei einem der variierenden Parameter k entsprechend der Krümmungssignalverschiebung einen Maximalwert hervor, resultiert die Annahme zweier identischer Signale. Selbst wenn sich die beiden Signale lediglich ähneln, kann auf Basis der Kreuzkorrelationsfunktion ein Maximum generiert werden, mittels derer eine Verschiebung detektiert werden kann (Handels 2009). Da die Krümmungssignale nicht kontinuierlich, sondern in diskreter Form als Summe vorliegen, hängen κ_1 und κ_2 von diskreten räumlichen Verschiebungspunkten ab. An dieser Stelle sei zu erwähnen, dass die hier temporär eingeführten Parameter n und k nichts mit den in der restlichen Arbeit gültigen Bezeichnungen für den Polynomgrad n oder die Knotenanzahl $k+1$ gemeinsam haben. Ausgehend von der Formel der diskreten Kreuzkovarianzfunktion von Zeitreihen nach Heunecke et al. (2013) und Metzner und Schwieger (2014) kann die diskrete Kreuzkovarianzfunktion von zwei stationären Krümmungssignalen gleicher Länge wie folgt beschrieben werden:

$$\begin{aligned} Cov_{\kappa_1,\kappa_2}(k) &= \frac{1}{n-k-1} \sum_{j=1}^{n-k} (\kappa_1(j) - \bar{\kappa}_1)(\kappa_2(j+k) - \bar{\kappa}_2) \\ &\text{mit } k = 0, 1, \dots, \frac{n}{10}, \end{aligned} \quad (4.15)$$

wobei n der Anzahl an Krümmungswerten entspricht. Für die zwei Krümmungssignale kann eine näherungsweise Stationarität und Äquidistanz angenommen werden, da zwei zu vergleichende Punktwolken von einem näherungsweise gleichen Standpunkt mit identischen Scanparametern aufgenommen werden. Aus Gründen der statistischen Sicherheit sollte die Kreuzkovarianzfunktion in Gleichung 4.15 nur bis zur Position $k = n/10$ berechnet werden. Für nähere Informationen hierzu sei auf Heunecke et al. (2013) verwiesen.

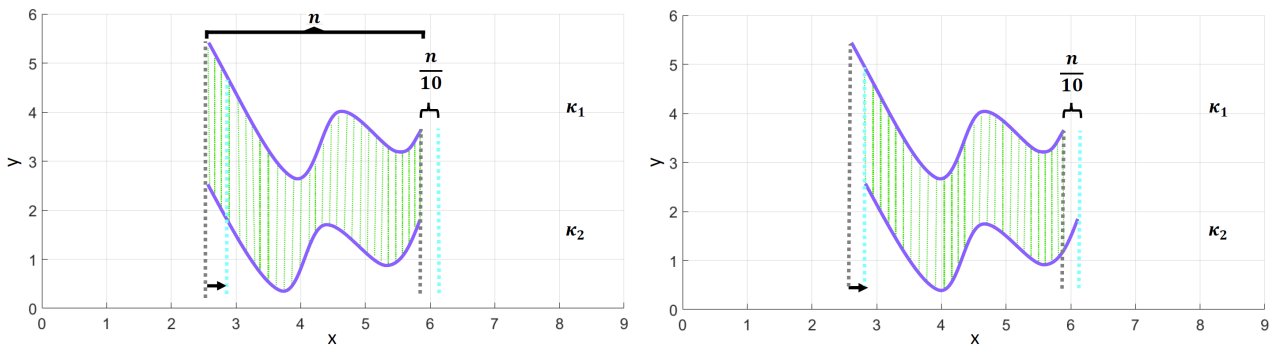


Abb. 4.19: Exemplarische Darstellung des Vorgangs der Kreuzkovarianzfunktion zweier Krümmungssignale gleicher Länge.

Abbildung 4.19 zeigt, zur Verdeutlichung der Funktionsweise der Kreuzkovarianzfunktion, zwei Krümmungssignale gleicher Länge, die allerdings nicht vollständig den gleichen Objektbereich diskretisieren. Mittels einer Verschiebung des Krümmungssignals κ_2 entlang κ_1 um maximal 10 % ($k = n/10$), soll der Parameter k ermittelt werden, der eine möglichst optimale Überlagerung beider Signale gewährleistet. Bei der Verwendung von Gleichung 4.15 sollte beachtet werden, dass zum einen eine Beschränkung auf die gleiche Länge beziehungsweise Anzahl an zu vergleichender Elemente beider Signale vorgegeben und zum anderen eine Limitierung auf eine einseitige 10 %-tige Signalverschiebung festgelegt ist.

Da die in dieser Arbeit zu vergleichenden B-Spline-Kurven eine unterschiedliche Anzahl an Kurvenpunkten und somit ebenso an Krümmungswerten aufweisen, muss Gleichung 4.15, die sich im ursprünglichen Sinne nicht auf Krümmungssignale, sondern Zeitreihen bezieht, derart angepasst werden, dass eine Verschiebung eines kürzeren Signals beziehungsweise eines Signals mit weniger Krümmungswerten entlang eines längeren Signals möglich ist. Als weitere Modifikation der Formel soll die bisher einseitige Verschiebung ausgehebelt und zu einer zweiseitigen Verschiebungsmöglichkeit ausgebaut werden. Gegeben seien die zwei Krümmungssignale κ_1 und κ_2 in diskreter Form und unterschiedlicher Anzahl an Krümmungswerten. κ_1 besteht aus n_1 Krümmungswerten und κ_2 ist aus n_2 Krümmungswerten aufgebaut, wobei $n_1 > n_2$ gilt. Um die Bestimmung einer diskreten Kreuzkovarianzfunktion für Signale unterschiedlicher Länge sowie mit einer zweiseitigen Verschiebung durchführen zu können, wird deren Berechnung in drei Prozessschritte aufgeteilt, die in den Gleichungen 4.17 bis 4.19 sowie mit der Abbildung 4.20 dargestellt werden. Im ersten Prozessschritt soll das kürzere Signal κ_2 beginnend bei einem Überstand von $n_2/10$ bis zu einer vollständigen Überlagerung von κ_1 verschoben werden. Dies generiert die Kovarianzfunktion $\mathbf{Cov}_{\kappa_1, \kappa_2}$ für $k \leq 0$ in Gleichung 4.17. Den zweiten Prozessschritt für $\mathbf{Cov}_{\kappa_1, \kappa_2}$ für $0 < k \leq n_1 - n_2$ in Gleichung 4.18 bildet die Verschiebung des Krümmungssignals κ_2 entlang κ_1 ab der vollständigen Überlagerung bis zum Ende des Signals κ_1 . Gleichung 4.19 schließt mit dem dritten Prozessschritt zur Bildung von $\mathbf{Cov}_{\kappa_1, \kappa_2}$ für $k > n_1 - n_2$ der Berechnung der diskreten Kreuzkovarianzfunktion ab, indem das kürzere Signal κ_2 bis zu einem Überstand von $n_2/10$ entlang κ_1 verschoben wird. Die Verschiebung der Krümmungssignale und die Verwendung der jeweiligen Formel wird über einen durchlaufenden Parameter k bestimmt:

$$k = -\frac{n_2}{10}, -\frac{n_2}{10} + 1, \dots, 0, \dots, n_1 - n_2 + \frac{n_2}{10}, \quad (4.16)$$

$$k \leq 0 : \quad \text{Cov}_{\kappa_1, \kappa_2}(k) = \frac{1}{n_2 - |k| - 2} \sum_{j=1}^{n_2+k} (\kappa_1(j) - \bar{\kappa}_1)(\kappa_2(j-k) - \bar{\kappa}_2), \quad (4.17)$$

$$0 < k \leq n_1 - n_2 : \quad \text{Cov}_{\kappa_1, \kappa_2}(k) = \frac{1}{n_2 - 2} \sum_{j=1+k}^{n_2+k} (\kappa_1(j) - \bar{\kappa}_1)(\kappa_2(j-k) - \bar{\kappa}_2), \quad (4.18)$$

$$k > n_1 - n_2 : \quad \text{Cov}_{\kappa_1, \kappa_2}(k) = \frac{1}{n_1 - |k| - 2} \sum_{j=1+k}^{n_1} (\kappa_1(j) - \bar{\kappa}_1)(\kappa_2(j-k) - \bar{\kappa}_2). \quad (4.19)$$

Zur besseren Nachvollziehbarkeit der Umsetzung kann eine Beispielrechnung im Anhang in Unterkapitel A.4 eingesehen werden. In den drei Grafiken von Abbildung 4.20 sind die drei Prozessschritte für zwei exemplarische Krümmungssignale abgebildet. Prozessschritt 1 zeigt die Verschiebung des kürzeren Krümmungssignals κ_2 in einer 1D-Verschiebung entlang des längeren Signals κ_1 ab dem 10 %-igen Überstand bis zur vertikalen blauen Linie. Die zweite Grafik zeigt Prozessschritt 2, bei dem das kürzere Krümmungssignal bis zum Ende des längeren Signals verschoben wird. Die dritte Grafik verbildlicht

die 1D-Verschiebung des kürzeren Signals bis zu einem 10 %-igen Überstand (Aichinger und Schwieger 2022).

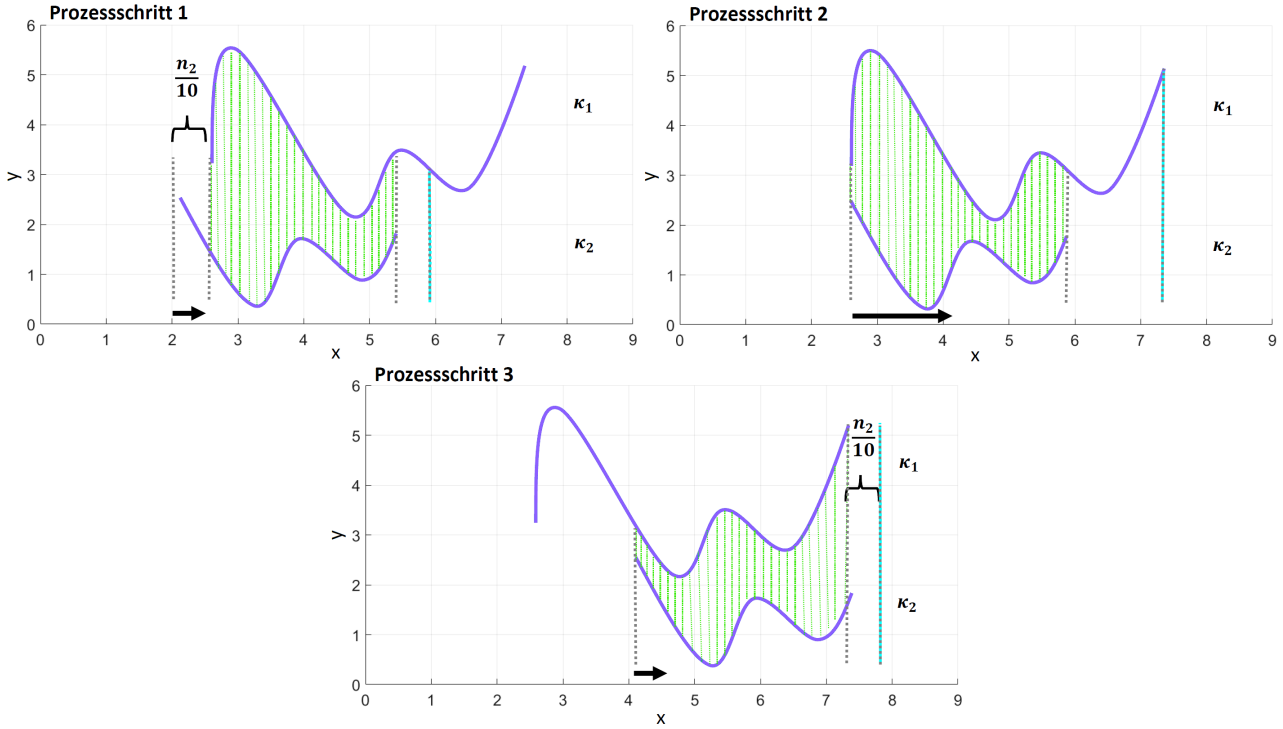


Abb. 4.20: Aufteilung des Vorgangs der Kreuzkovarianzfunktion zweier Krümmungssignale unterschiedlicher Länge in drei Prozessschritte.

Mittels Normierung der Kreuzkovarianzfunktion kann die Kreuzkorrelationsfunktion hergeleitet werden (Heunecke et al. 2013; Meyer 2017):

$$\text{Corr}_{\kappa_1, \kappa_2}(k) = \frac{\text{Cov}_{\kappa_1, \kappa_2}(k)}{\sqrt{\text{Cov}_{\kappa_1, \kappa_1}(0) \text{Cov}_{\kappa_2, \kappa_2}(0)}} = \frac{\text{Cov}_{\kappa_1, \kappa_2}(k)}{\sigma_{\kappa_1} \sigma_{\kappa_2}}, \quad (4.20)$$

die für jeden diskreten Wert k eine Berechnung des Korrelationskoeffizienten nach Bravais-Pearson zulässt. Der Pearson Korrelationskoeffizient kann Werte zwischen -1 und $+1$ annehmen und verdeutlicht negative bzw. positive lineare Zusammenhänge, die um so stärker ausgeprägt sind, je näher der Wert bei -1 bzw. $+1$ liegt (Hartung et al. 2002).

Die vorgestellte Korrelationsanalyse bildet Stufe I und II des Mapcut-Algorithmus. In Stufe I wird ein Grobmapping durchgeführt, das nur jeden zehnten Krümmungswert in der Korrelationsanalyse betrachtet. Im darauf folgenden Feinmapping (Stufe II) wird jeder Krümmungswert für einen ausgewählten Bereich beleuchtet. Die Durchführung eines Grob- und Feinmappings wird aus Gründen einer geringeren Rechenzeit angewendet. Detailliert können die Stufe I und II des Mapcut-Algorithmus für B-Spline-Kurven anhand von Abbildung 4.21 erklärt werden. Die beiden oberen Grafiken zeigen zwei exemplarische B-Spline-Kurven unterschiedlicher Länge der Epoche 0 und Epoche 1. Für jeden B-Spline-Kurvenpunkt wird ein Krümmungsparameter berechnet, wobei zur Datenreduzierung im Grobmapping (Stufe I) nur jeder zehnte Wert ($\Delta t = 10$) für die nachfolgende Korrelationsanalyse verwendet wird. Die Krümmungswerte stellen die zwei Grafiken der zweiten Zeile dar. Die Durchführung der Korrelationsanalyse erzielt einen Korrelationskoeffizienten für jeden Verschiebungswert k , deren höchster Parameter den Verschiebungswert deklariert, der die beste eindimensionale Verschiebung

und Überlagerung der Krümmungsparameter der Epochen 0 und 1 widerspiegelt. Die resultierenden Korrelationskoeffizienten für $\Delta t = 10$ sind in der dritten Zeile in Abbildung 4.21 abgebildet.

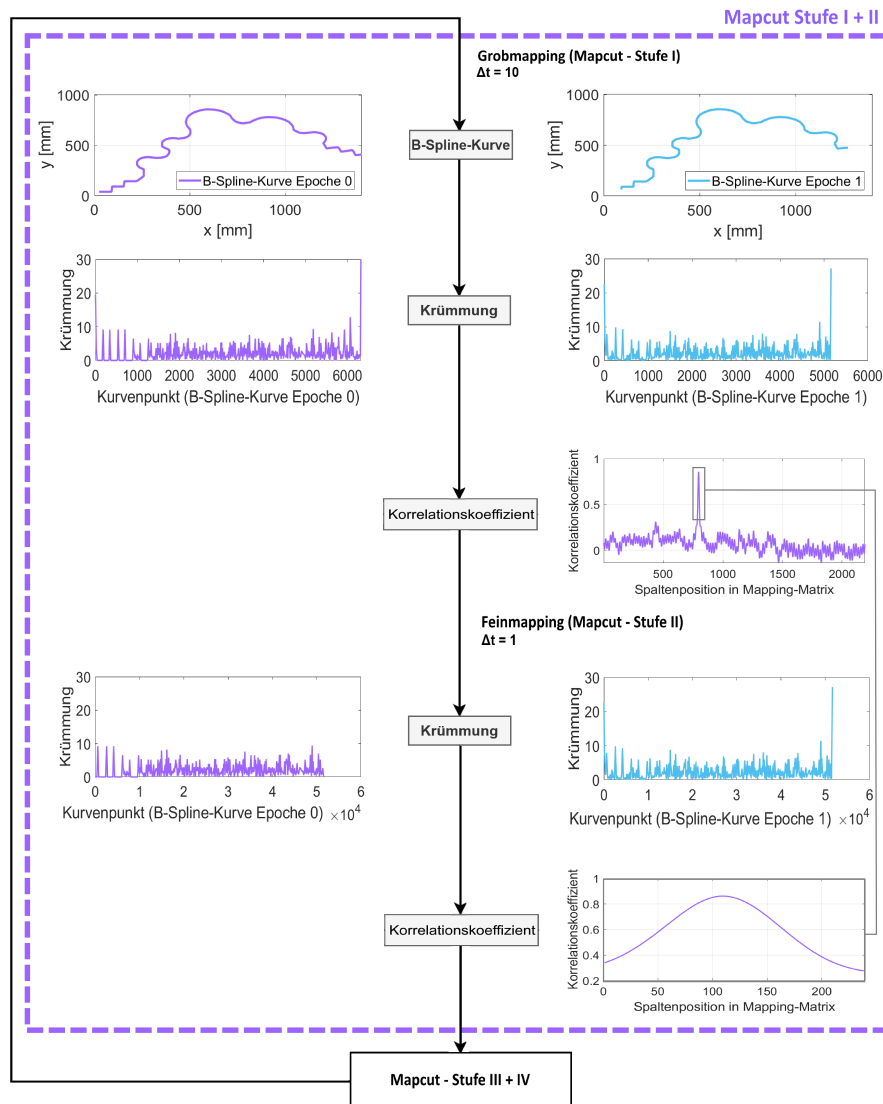


Abb. 4.21: Ablauf von Stufe I und II des Mapcut-Algorithmus für B-Spline-Kurven (Grobmapping und Feinmapping).

Um eine Präzisierung und Annäherung des gesuchten optimalen Verschiebungswerts zu erzielen, wird daraufhin das Feinmapping (Stufe II) durchgeführt. Hierbei findet eine Wiederholung des zuvor beschriebenen Vorgangs für jeden Wert ($\Delta t = 1$) statt. Es wird allerdings nicht mehr der gesamte Kurvenbereich der B-Spline-Kurven gegeneinander verschoben, sondern es werden lediglich die Krümmungswerte in einem Bereich von 120 Krümmungswerten vor und hinter der im Grobmapping definierten Verschiebungsposition verglichen. Die genannte Anzahl an Krümmungswerten für den Suchbereich hat sich empirisch als vorteilhaft gezeigt. Die zwei Grafiken in Zeile vier von Abbildung 4.21 stellen die Krümmungswerte von Epoche 0 und Epoche 1 für das Feinmapping dar. Zeile fünf zeigt die in der Korrelationsanalyse resultierenden Korrelationskoeffizienten. Der im Feinmapping betrachtete kleinere Verschiebungsbereich ist in der Grafik in Zeile drei durch ein schwarzes Rechteck gekennzeichnet. Nach dem Grob- und Feinmapping folgen Stufe III und IV des Mapcut-Algorithmus, die Teil eines iterativen Vorgangs sind.

Auf Grundlage des im Feinmapping berechneten Verschiebungswerts kann basierend auf den ermittelten korrespondierenden, jedoch nicht identischen Krümmungswerten beider Epochen auf die annähernd korrespondierenden Kurvenpunkte in beiden Messepochen zurück geschlossen werden. Durch den Mappingvorgang erhält man somit korrespondierende Kurvenpunkte von Epoche 0 und Epoche 1. Diese zugehörigen Kurvenpunkte unterscheiden sich allerdings in ihrem lokalen Koordinatensystem. Ebenso kann es sein, dass als korrespondierend deklarierte Kurvenpunkte in einem deformierten Kurvenbereich liegen. Eine Transformation in ein gemeinsames Koordinatensystem ist somit wichtig, um mögliche Starrkörperbewegungen zwischen den zwei Epochen ausfindig machen zu können. Zusätzlich kann es sein, dass die B-Spline-Kurven der beiden Epochen unterschiedlich viele Kontrollpunkte besitzen und deshalb unterschiedlich lang sind. Das bedeutet, dass der Kurvenanfang und das Kurvenende anders geformt sind, da die Anfangs- und Endpunkt-Eigenschaft (siehe Abschnitt 2.2.3) die B-Spline-Kurve dazu zwingt mit dem Anfangs- bzw. Endkontrollpunkt zusammenzufallen. Dieses Verhalten wirkt sich auf das Ergebnis des Mappings negativ aus. In Folge dessen sind weitere Maßnahmen notwendig, die in Stufe III und IV des Mapcut-Algorithmus näher erläutert werden.

Prinzipiell sollte bei der Methode beachtet werden, dass die Annahme möglicher korrespondierender Kurvenpunkte auch auf der Voraussetzung eines annähernd identischen Messaufbaus beruht. Denn nur bei einer Abtastung verschiedener Epochen mit gleichen Kenngrößen hinsichtlich Abtastrate, Auftreffwinkel, Entfernung zum Objekt sowie Schrittweite t in der B-Spline-Kurven-Generierung kann davon ausgegangen werden, dass die Abstände zwischen den einzelnen Kurvenpunkten bei beiden Epochen annähernd gleich sind. Eine hundertprozentige Übereinstimmung der Kurvenpunkte zweier Epochen kann mit dieser Methode allerdings nicht garantiert werden, da die Abstände zwischen den Kurvenpunkten nur annähernd gleich, aber nicht identisch sein können.

Transformation und Cutting (Stufe III und IV)

Im Folgenden wird Stufe III des Mapcut-Algorithmus, begleitet vom Ablaufdiagramm in 4.22, näher beschrieben. Nach der ersten Durchführung der Korrelationsanalyse im Grob- und Feinmapping (Stufe I und II) innerhalb des Mapcut-Algorithmus, liegen einander zugehörige korrespondierende Kurvenpunkte in beiden Epochen vor. Da sich die als korrespondierend deklarierten Kurvenpunkte sowohl in unterschiedlichen Koordinatensystemen befinden als auch in deformierten Bereichen liegen können, sind annähernd identische Punkte zu definieren, die in stabilen Kurvenbereichen und bei Transformation in ein gemeinsames Koordinatensystem möglichst geringe Koordinatendifferenzen vorweisen sollten.

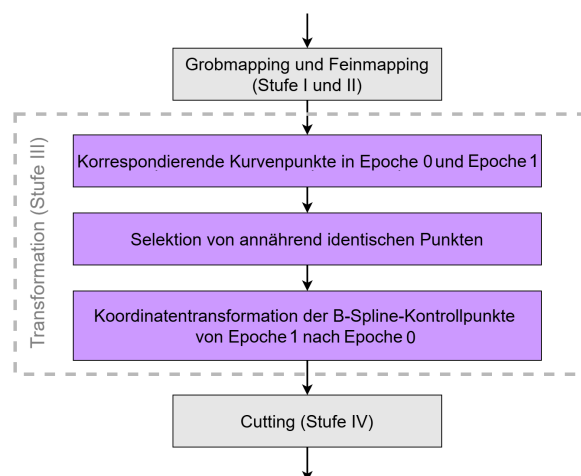


Abb. 4.22: Ablauf von Stufe III des Mapcut-Algorithmus für B-Spline-Kurven (Transformation).

Hierfür werden die beiden zu betrachtenden B-Spline-Kurven in zehn gleichmäßige Teile unterteilt. In jedem dieser zehn Kurventeile wird ein korrespondierender Kurvenpunkt gesucht, der als annähernd identischer Punkt bei einer Koordinatentransformation helfen soll, die Laserscandaten und B-Spline-Kurvenpunkte von Epoche 1 in das System von Epoche 0 zu überführen. Die Unterteilung in zehn gleichmäßige Teile hat für die verwendeten Datensätze empirisch die besten Ergebnisse ergeben. Für andere Datensätze sollte diese Unterteilung überprüft werden.

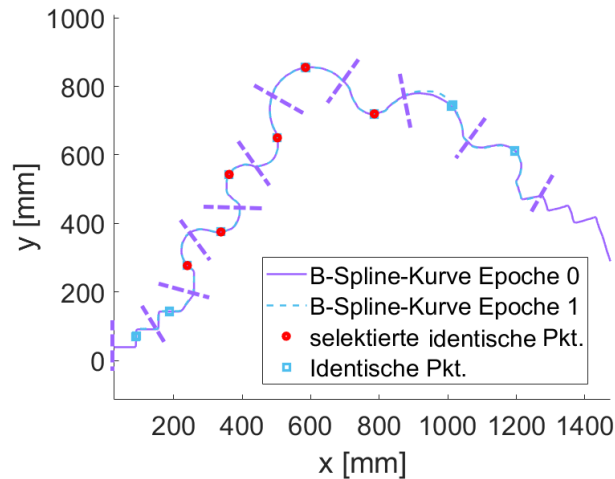


Abb. 4.23: B-Spline-Kurven eines Beispieldatensatzes von Epoche 0 und Epoche 1. Annähernd identische Punkte in 10 Kurventeilen und selektierte identische Punkte in 60 % der Kurventeile (Iteration 1).

Mittels ähnlichem Messaufbau weisen die Epochen insgesamt nur Differenzen im geringen cm-Bereich und kleine Drehwinkel auf, so dass eine Selektion der bereits über das Grob- und Feinmapping zugewiesenen korrespondierenden Kurvenpunkte mittels Abgleich der Koordinatendifferenzen dieser Punktkorrespondenzen erfolgen kann. In dieser Arbeit und bei den verwendeten Datensätzen werden lediglich kleine deformierte Bereiche verwendet. Hierbei hat sich empirisch herausgestellt, dass gute Ergebnisse für die Transformation geliefert werden, wenn 60 % der selektierten vorläufigen annähernd identischen Punkte mit der geringsten Koordinatendifferenz ausgewählt werden. Die sechs ausgewählten annähernd identischen Punkte werden daraufhin dazu genutzt, das gesamte Kontrollpolygon bzw. jeden B-Spline-Kontrollpunkt von Epoche 1 in das Koordinatensystem von Epoche 0 zu transformieren. Abbildung 4.23 veranschaulicht die Selektion von identischen Punkten an einem Beispieldatensatz von B-Spline-Kurven einer Epoche 0 und Epoche 1. Die Unterteilung der Kurve in 10 Kurventeile wird anhand magentafarbener Striche gekennzeichnet. Die annähernd identischen Punkte werden hellblau dargestellt, während final selektierte identische Punkte in rot abgebildet werden. In dem Beispieldatensatz in Abbildung 4.23 wird veranschaulicht, dass im deformierten Kurvenbereich, der sich im zweiten Kurventeil von rechts befindet, kein identischer Punkt selektiert wird. Der Mapcut-Algorithmus verläuft in einem iterativen Prozess, so dass bei den weiteren Iterationsschritten erneut identische Punkte selektiert werden. Ab dem zweiten Iterationsdurchlauf wird eine feinere Unterteilung der Kurve vorgenommen, da davon ausgegangen wird, dass durch den ersten Iterationsdurchlauf eine gute Überlagerung der B-Spline-Kurven hergestellt werden konnte. Ab der zweiten Iteration findet somit eine Auswahl von 15 Kurventeilen statt, in denen annähernd identische Punkte ausfindig gemacht werden. Hiervon werden wiederum 60 % mit der geringsten Koordinatendifferenz als selektierte identische Punkte für die Koordinatentransformation verwendet.

Abbildung 4.24 stellt die Unterteilung der B-Spline-Kurven von Epoche 0 und Epoche 1 im letzten Iterationschritt dar. Die Unterteilung ist durch magentafarbene Striche signalisiert und die selektierten identischen Punkte in rot dargestellt. Als Koordinatentransformation wird für den 2D-Fall eine Helmerttransformation angewendet. Für nähere Informationen zur angewendeten Helmerttransformation sei auf Kahmen (2006) verwiesen.

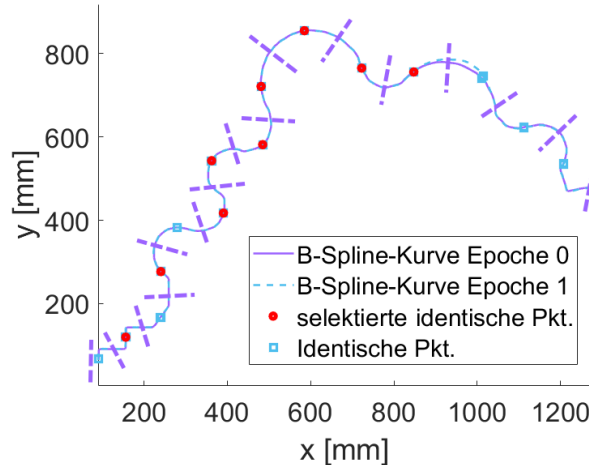


Abb. 4.24: B-Spline-Kurven eines Beispieldatensatzes von Epoche 0 und Epoche 1. Annähernd identische Punkte in 15 Kurventeilen und selektierte identische Punkte in 60 % der Kurventeile (Letzte Iteration).

Ergebnis von Stufe III des Mapcut-Algorithmus sind von Epoche 1 nach Epoche 0 transformierte B-Spline-Kontrollpunkte sowie ursprünglich in Epoche 0 vorliegende B-Spline-Kontrollpunkte. Die beschriebene Stufe III kann entsprechend dem Ablaufdiagramm in Abbildung 4.18 zwischen Feinmapping und Cutting im Mapcut-Algorithmus eingeordnet werden. Hätten zwei zu betrachtende Kontrollpolygone von Epoche 0 und Epoche 1 sowie deren B-Spline-Kurven den gleichen Anfangs- und Endpunkt, könnte der Mapcut-Algorithmus an dieser Stelle abgeschlossen werden. Da in dieser Arbeit allerdings Kontrollpolygone unterschiedlicher Länge bzw. mit unterschiedlichem Anfangs- und Endkontrollpunkt betrachtet werden, ergeben sich bei den B-Spline-Kurven von zwei Epochen abweichende Kurvenformen. Die Verhaltensweise von zwei Kontrollpolygonen, deren Anfangs- und Endpunkt nicht auf eine ähnliche Position fällt, soll anhand von Abbildung 4.25 erklärt werden.

Abbildung 4.25 zeigt in der oberen Grafik zwei Kontrollpolygone, die aus einer unterschiedlichen Anzahl an Kontrollpunkten bestehen. Kontrollpolygon von Epoche 0 ist mit grauen Kontrollpunkten abgebildet und Kontrollpolygon von Epoche 1 mit schwarzen Kontrollpunkten. Das Kontrollpolygon von Epoche 0 enthält zwei Kontrollpunkte mehr als das Kontrollpolygon von Epoche 1, so dass sowohl zu Beginn als auch am Ende dieses Kontrollpolygons übersteht. Zu erkennen ist, dass die zwei aus den unterschiedlichen Kontrollpolygonen generierten Kurvenpunkte der B-Spline-Kurven dritten Grades teilweise eine andere Form vorweisen, obwohl die Kontrollpunkte im mittleren Bereich identisch sind. Wird ein Mapcut-Prozess bis zu Stufe III durchgeführt, werden als Resultat jeweils zwei B-Spline-Kurvenpunkte der beiden Epochen ab dem linken ersten schwarzen Kontrollpunkt des kürzeren Kontrollpolygons zueinander zugewiesen und als korrespondierend angenommen. Dies liegt an den ähnlichen Krümmungsparametern im mittleren Bereich, wodurch eine derartige Überlagerung im Mapcut-Prozess vorgegeben wird. In der unteren Grafik von Abbildung 4.25 ist hierbei zu beobachten, dass die Differenzen zugeordneter B-Spline-Kurvenpunkte anwachsen. Betragen sie im Bereich des linken schwarzen Anfangskontrollpunktes noch weniger als 0,5 mm, wachsen sie bis zum schwarzen Endkontrollpunkt auf 2,5 mm an. Dies kann damit erklärt werden, dass der differierende Startkontrollpunkt mit dem die B-Spline-Kurve

entsprechend der Anfangs- und Endpunkt-Eigenschaft nach Abschnitt 2.2.3 interpolieren muss, die Kurvenkrümmung und somit die Dichte der Kurvenpunkte innerhalb einer Kurve zueinander steuert.

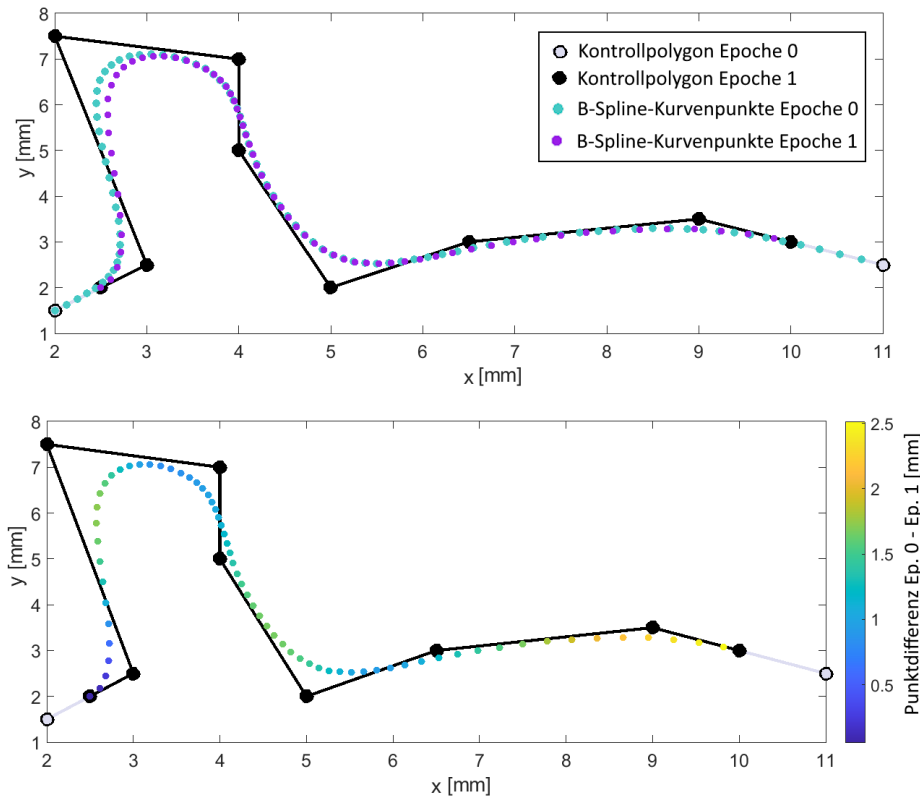


Abb. 4.25: Obere Grafik: Zwei Kontrollpolygone unterschiedlicher Punktanzahl und deren B-Spline-Kurven. Untere Grafik: Punktdifferenzen zugehöriger B-Spline-Kurvenpunkte nach Stufe III des Mapcut-Prozesses.

Ist die Krümmung groß, bedeutet dies eine hohe Kurvenpunktdichte, während die Kurvenpunktdichte geringer ausfällt, wenn auch die Kurvenkrümmung gering ist. Dementsprechend wird durch die unterschiedliche Position der Startkontrollpunkte beider Epochen zu Beginn der Kurven eine differierende Krümmung und ein differierender Kurvenpunkt Abstand zwischen den Kurvenpunkten erzeugt, der eine Zunahme der Differenz zwischen zugewiesenen zugehörigen Kurvenpunkte zweier Epoche forciert.

Eine Lösung dieser Thematik bietet ein Ansatz einer sogenannten Ausstanzung bzw. eines Cuttings des längeren Kontrollpolygons durch das kürzere Kontrollpolygon, so dass die Start- und Endkontrollpunkte beinahe auf die gleiche Position fallen. Die Genauigkeit mit der die Start- und Endkontrollpunkte zweier Epochen auf die gleiche Position festgelegt werden können, hängt vom Punktabstand der verwendeten Punktwolken ab. Liegt der Punktabstand im mm-Bereich, können auch die Start- und Endkontrollpunkte im Cutting-Prozess auf mm-Genauigkeit bestimmt werden. Die Vorgehensweise des Cuttings wird mit den Grafiken in Abbildung 4.26 verdeutlicht. In der oberen Grafik ist zu erkennen, dass die Form der beiden B-Spline-Kurven durch das Entfernen der zwei äußeren grauen Kontrollpunkte wesentlich ähnlicher ausfällt. Die untere Grafik zeigt die deutlich geringeren Punktdifferenzen der zugehörigen B-Spline-Kurvenpunkte in Relation zu Abbildung 4.25. Je mehr sich die Start- und Endkontrollpunkte der Kontrollpolygone beider Epochen in der Position ähneln, desto geringer wird die Differenz im Randbereich. Ein Anwachsen von Differenzen zugehöriger B-Spline-Kurvenpunkte kann mit diesem Ansatz verhindert bzw. verringert werden.

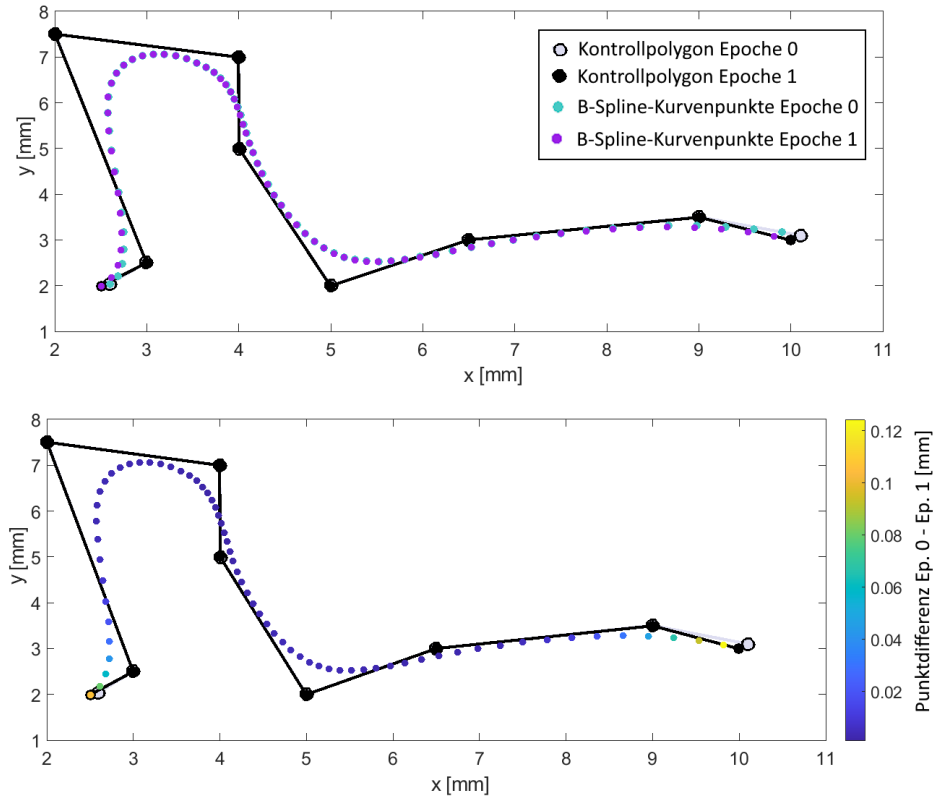


Abb. 4.26: Obere Grafik: Zwei Kontrollpolygone gleicher Punktzahl und deren B-Spline-Kurven. Untere Grafik: Punktdifferenzen zugehöriger B-Spline-Kurvenpunkte nach Stufe IV des Mapcut-Prozesses (Cutting).

Die letzte und vierte Stufe des Mapcut-Algorithmus besteht demnach aus einem Cutting. Hierbei wird das Kontrollpolygon des längeren Kontrollpolygons derart gekürzt, dass die Koordinatendifferenz zwischen dem ausgewählten Anfangskontrollpunkt in Relation zum ersten B-Spline-Kontrollpunkt des kürzeren B-Spline-Kontrollpolygons minimal ausfällt. Genauso wird ein Kontrollpunkt am Ende des längeren Kontrollpolygons als Endkontrollpunkt ausfindig gemacht, dessen Koordinatendifferenz minimal zum letzten Kontrollpunkt des kürzeren Kontrollpolygons ist. Die Angleichung der Kontrollpolygone beider Epochen in Stufe IV des Mapcut-Algorithmus schafft eine Datenbasis, die bei erneuter Anwendung des Mapcut-Algorithmus mit vorheriger Bildung von B-Spline-Kurven und deren Krümmungsparametern entsprechend des Ablaufdiagramms in Abbildung 4.18, die Qualität der Zuweisung korrespondierender Kurvenpunkte verbessert. Auf Grundlage des Mapcut-Algorithmus können B-Spline-Kurven zweier Epochen feinregistriert werden, um in einem nächsten Schritt in einer Deformationsanalyse auf signifikante Deformationen untersucht werden zu können.

4.3.2 Mapcut-Algorithmus (B-Spline-Flächen)

Der entwickelte Mapcut-Algorithmus ist ebenso für B-Spline-Flächen anwendbar und ist in Abbildung 4.27 im Kontext der DBKM als Ablaufdiagramm dargestellt. Er untergliedert sich gleichermaßen in vier Stufen. In einer Korrelationsanalyse mittels Krümmung anhand eines Grobmappings und Feinmappings (Stufe I und II) wird die optimale Überlagerungsposition zwischen zwei Epochen ausfindig gemacht. Auf dieser Basis können identische Punkte selektiert werden, die eine Transformation der Laserscandaten von Epoche 1 in das Koordinatensystem von Epoche 0 gewährleisten (Stufe III).

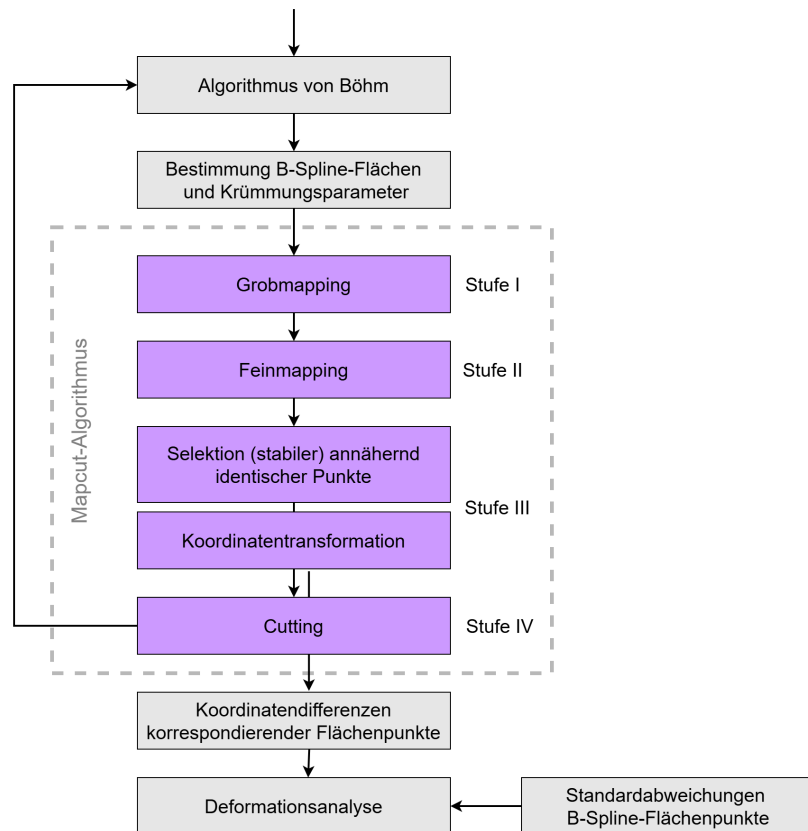


Abb. 4.27: Ablauf des Mapcut-Algorithmus für B-Spline-Flächen im Kontext der DBKM.

Das anschließende Cutting beinhaltet eine Ausstanzung des größeren Kontrollgitters mittels der Randkontrollpunkte des kleineren Kontrollgitters (Stufe IV). Nach dieser Kontrollgitterrandangleichung wird der bereits bekannte Prozess des Algorithmus von Böhm und der Generierung der B-Spline-Flächen vollführt. Durch eine iterative Angleichung der Kontrollgitterränder und Optimierung der Koordinatentransformation soll ein direkter Koordinatenvergleich von zugehörigen B-Spline-Flächenpunkten zweier Epochen hinsichtlich Deformationen ausgeführt werden können. Die einzelnen Stufen des Mapcut-Algorithmus werden im Folgenden für B-Spline-Flächen näher erläutert.

Grobmapping und Feinmapping (Stufe I und II)

Um eine angenäherte Überlagerungsposition von B-Spline-Flächen zweier Epochen zu erzielen, werden statt der 3D-B-Spline-Flächenpunkte deren 1D-Krümmungsparameter als 2D-Gitter verglichen. Schrittweise wird das 2D-Krümmungsgitter, in Form einer 2D-Krümmungsmatrix, der Epoche mit der kleineren Zeilen- und Spaltenanzahl und somit der Punktwolke mit geringerem Ausmaß über das 2D-Krümmungsgitter der Punktwolkenepoche mit dem größerem Ausmaß verschoben. Stufe I beinhaltet ein Grobmapping, bei dem nur jeder zehnte Wert betrachtet wird, um eine Annäherung der gesuchten Überlagerungsposition mit möglichst geringer Rechenleistung zu finden. Hierbei wird analog zum Mapcut-Algorithmus im Kurvenfall ein maximaler Verschiebungsüberstand von 10 % verfolgt. Zeilenweise wird das 2D-Krümmungsgitter kleineren Ausmaßes über das Krümmungsgitter größeren Ausmaßes verschoben und für jede dieser Verschiebungspositionen ein mehrdimensionaler Korrelationskoeffizient berechnet.

Die Formeln zur Berechnung der Korrelationskoeffizienten-Matrix sind eine Weiterentwicklung der Gleichungen vom 2D-Kurvenfall 4.16 bis 4.19. Hierbei gilt k_1 für eine zeilenweise Verschiebung und k_2 für eine spaltenweise Verschiebung. Ebenso wird vorgegeben, dass $n_1 > n_2$ und $n_3 > n_4$. Dies bedeutet, dass die Krümmungsmatrix von Epoche 0 mit n_1 mehr Zeilen und mittels n_3 mehr Spalten als die Krümmungsmatrix von Epoche 1 aufweist. Die Verschiebungsparameter k_1 und k_2 sind wie folgt aufgebaut:

$$k_1 = -\frac{n_2}{10}, -\frac{n_2}{10} + 1, \dots, 0, \dots, n_1 - n_2 + \frac{n_2}{10}, \quad (4.21)$$

$$k_2 = -\frac{n_4}{10}, -\frac{n_4}{10} + 1, \dots, 0, \dots, n_3 - n_4 + \frac{n_4}{10}. \quad (4.22)$$

Damit sowohl zeilenweise als auch spaltenweise ein 10%-iger Überstand bei der Verschiebung gewährleistet werden kann, werden für die Generierung der Korrelationskoeffizienten-Matrix neun Formeln benötigt, die durch die Verschiebungsparameter k_1 und k_2 gesteuert werden. Tabelle 4.2 zeigt die Limitierungen durch die Verschiebungsparameter entsprechend zeilenweiser und spaltenweiser Verschiebung. Die mittige Zelle gibt die Verschiebungsparameter wieder, bei der die Krümmungsmatrizen der Epoche 0 und Epoche 1 ohne 10-%igen Überstand verschoben werden.

zeilenweise	$k_1 \leq 0$	$0 < k_1 \leq n_1 - n_2$	$k_1 > n_1 - n_2$
spaltenweise	$k_2 \leq 0$	$k_2 \leq 0$	$k_2 \leq 0$
zeilenweise	$k_1 \leq 0$	$0 < k_1 \leq n_1 - n_2$	$k_1 > n_1 - n_2$
spaltenweise	$0 < k_2 \leq n_3 - n_4$	$0 < k_2 \leq n_3 - n_4$	$0 < k_2 \leq n_3 - n_4$
zeilenweise	$k_1 \leq 0$	$0 < k_1 \leq n_1 - n_2$	$k_1 > n_1 - n_2$
spaltenweise	$k_2 > n_3 - n_4$	$k_2 > n_3 - n_4$	$k_2 > n_3 - n_4$

Tab. 4.2: Aufstellung der Verschiebungsparameter-Grenzen zur Bestimmung der Formel für die Kreuzkovarianzfunktion des Mapcut-Algorithmus für B-Spline-Flächen.

Mittels Tabelle 4.2 kann die Formel für jeden Verschiebungsparameter erzeugt werden. Beispielhaft ist die Formel der Kreuzkovarianzfunktion für $k_1 \leq 0$ und $k_2 \leq 0$ in Gleichung 4.3.2 dargestellt:

$$k_1 \leq 0 \text{ und } k_2 \leq 0 : \quad Cov_{\kappa_1, \kappa_2}(k_1, k_2) = \frac{1}{n_2 - |k_1| - 2} \frac{1}{n_4 - |k_2| - 2} \sum_{j_1=1}^{n_2+k_1} \sum_{j_2=1}^{n_4+k_2} (\kappa_1(j_1, j_2) - \overline{\kappa_1})(\kappa_2(j_1 - k_1, j_2 - k_2) - \overline{\kappa_2}). \quad (4.23)$$

Aus Gründen der Übersichtlichkeit werden, wie im Kurvenfall, die Formeln der Kreuzkovarianzfunktion aufgeführt. Mit Hilfe von Gleichung 4.20 wird aus der Kreuzkovarianzfunktion eine Kreuzkorrelationsfunktion berechnet.

In Ergänzung zu den dargestellten Formeln verdeutlicht Abbildung 4.28 den Vorgang des Mapcut-Algorithmus Stufe I bildhaft. Das hellblaue Rechteck stellt das Krümmungsgitter einer kleineren Epoche dar, das bei 10 %igem Überstand zeilenweise über das Krümmungsgitter der größeren Epoche verschoben wird. Da jede Verschiebungsposition einen Korrelationskoeffizienten hervorbringt, entsteht eine Korrelationskoeffizienten-Matrix, deren größter Wert eine angenäherte Überlagerungsposition der Krümmungsgitter mit der höchsten Übereinstimmung wiedergibt.

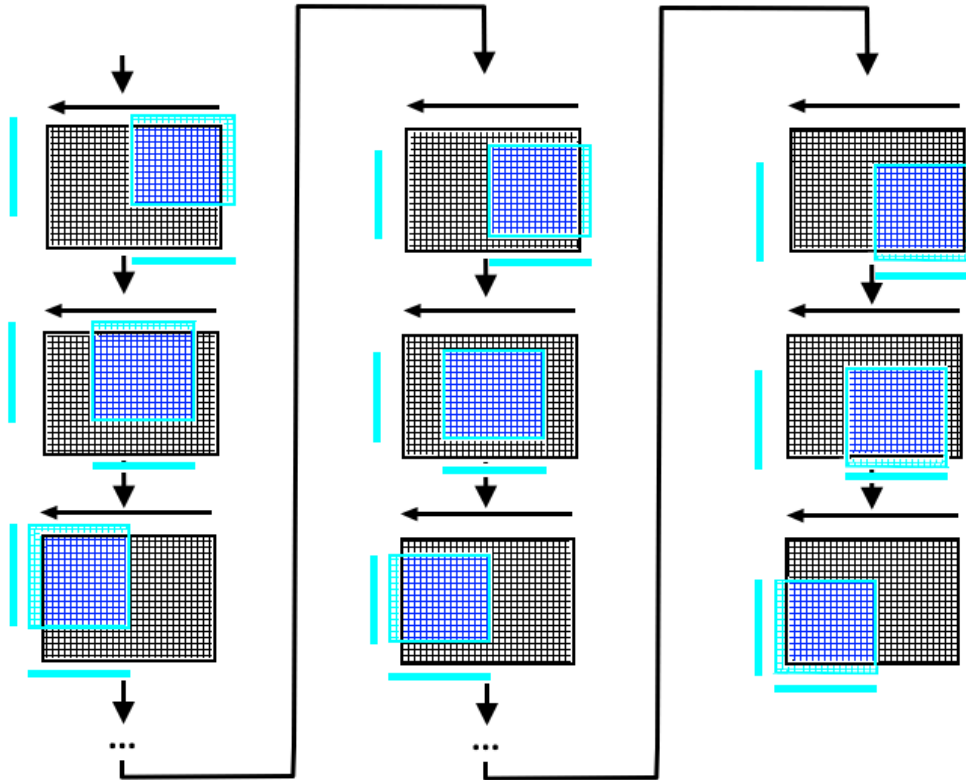


Abb. 4.28: Schrittweise Verschiebung der Krümmungsgitter zweier Epochen im Grobmapping.

Abbildung 4.29 stellt eine beispielhafte Korrelationskoeffizienten-Matrix eines Beispieldatensatzes von TLS-Daten eines Gurkenblatts zweier Epochen im ersten Iterationschritt dar. Die linke Grafik zeigt das Gitter der Korrelationskoeffizienten-Matrix im Grobmapping und die rechte Grafik im Feinmapping. Die vollständige Prozessierung der DBKM inklusive Mapcut-Algorithmus für das Gurkenblatt ist in Kapitel 7 in Abschnitt 7.1.2 dargelegt. Die Korrelationskoeffizienten-Matrix wird als dreidimensionale farbliche Abbildung entsprechend der Größe des Korrelationskoeffizienten für die jeweilige Überlagerungsposition präsentiert. Während die dunkelblauen Werte nahezu keine Übereinstimmung der zwei Epochen signalisieren, zeigt der Peak mit gelber Färbung eine höhere Übereinstimmung.

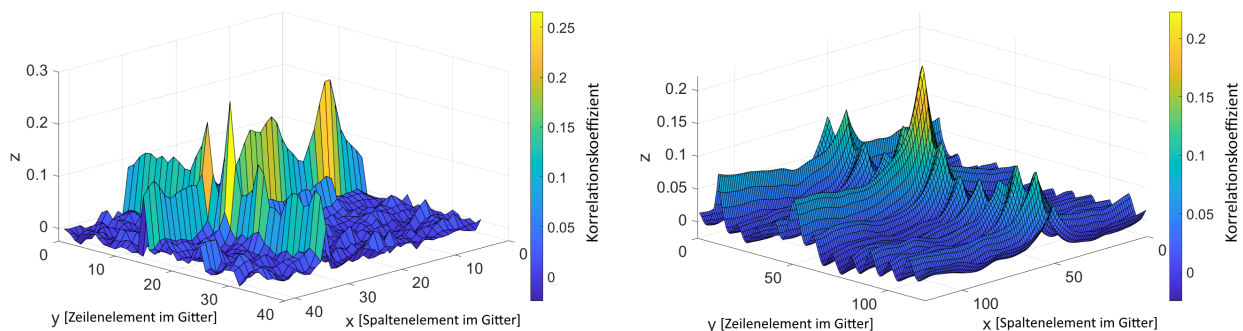


Abb. 4.29: Grobmapping und Feinmapping für einen Beispieldatensatz (Iteration 1).

Durch das Grobmapping mit der Abfrage jedes zehnten Krümmungswerts kann eine annähernde Überlagerungsposition zwischen zwei Epochen ausfindig gemacht werden. In Stufe II des Mapcut-Algorithmus wird im Feinmapping im Bereich der identifizierten Überlagerungsposition die gleiche

Vorgehensweise betreffend schrittweiser Verschiebung zweier Krümmungsgitter vorgenommen wie im Grobmapping. Das Feinmapping unterscheidet sich allerdings darin, dass beim Vergleich der Krümmungsgitter nicht jeder zehnte Krümmungswert, sondern jeder Krümmungswert für die Abfrage und Berechnung des mehrdimensionalen Korrelationskoeffizienten herangezogen wird. Die schrittweise Verschiebung erfolgt nur in einem ausgewählten Bereich ohne 10 %igen Verschiebungsüberstand. Das Feinmapping beschränkt sich auf eine Verschiebungsausdehnung von 120 Krümmungswerten, da eine grobe Position bereits bekannt ist. Dies ermöglicht eine recheneffiziente Umsetzung der zwei Stufen des Mapcut-Algorithmus. Prinzipiell ist der Abfrageumfang im Feinmapping je nach Datensatz zu hinterfragen und gegebenenfalls bei Bedarf zu vergrößern. Bei den in dieser Arbeit verwendeten Datensätzen hat sich die genannte Verschiebungsausdehnung bewährt und wird deshalb für jeden Datensatz angewendet. Analog zum 2D-Kurvenfall wird der Mapcut-Algorithmus für B-Spline-Flächen iterativ durchgeführt. Dies bedeutet, dass die Ergebnisse sukzessive verbessert werden. Die letzte Iteration von Stufe I und II im Mapcut-Algorithmus bringt demzufolge höhere Korrelationen hervor, als in der ersten Iteration.

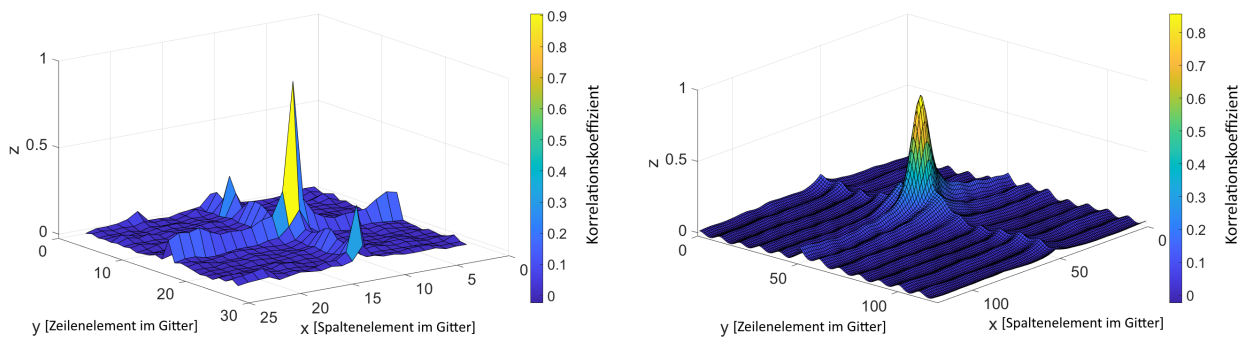


Abb. 4.30: Grobmapping und Feinmapping für einen Beispieldatensatz (Letzte Iteration).

Eine beispielhafte Ausprägung der Korrelationskoeffizienten-Matrix in der letzten Iteration kann in den Grafiken in Abbildung 4.30 eingesehen werden. Die linke Grafik stellt das Ergebnis des Grobmappings dar. Die rechte Grafik zeigt das Resultat des Feinmappings.

Transformation und Cutting (Stufe III und IV)

Durch das Grob- und Feinmapping werden, wie in Abbildung 4.31 als Ablaufdiagramm dargestellt, zugehörige bzw. korrespondierende B-Spline-Flächenpunkte in Epoche 0 und 1 zugeordnet. Dies versetzt in die Lage Flächenpunktpaare in Epoche 0 und Epoche 1 mit geringer Koordinatendifferenz ausfindig zu machen, die als annähernd identische Punkte für eine Koordinatentransformation verwendet werden können. Die Selektion von Flächenpunktpaaren über eine geringe Koordinatendifferenz ist möglich, da eine Aufstellung des terrestrischen Laserscanners auf einer ähnlichen Position über mehrere Epochen erforderlich ist. Dies resultiert in geringen Translations- sowie Drehwinkelwerten. Bei der Selektion der annähernd identischen Punkte werden die B-Spline-Flächen in der ersten Iteration in 4 x 4 Kacheln gleichmäßig aufgeteilt und in jedem dieser 16 Kacheln ein korrespondierender Flächenpunkt mit der geringsten Koordinatendifferenz gesucht. Unter den 16 ausgewählten Flächenpunktpaaren werden sechs mit der geringsten Koordinatendifferenz selektiert und als annähernd identische Punkte in die Transformation überführt.

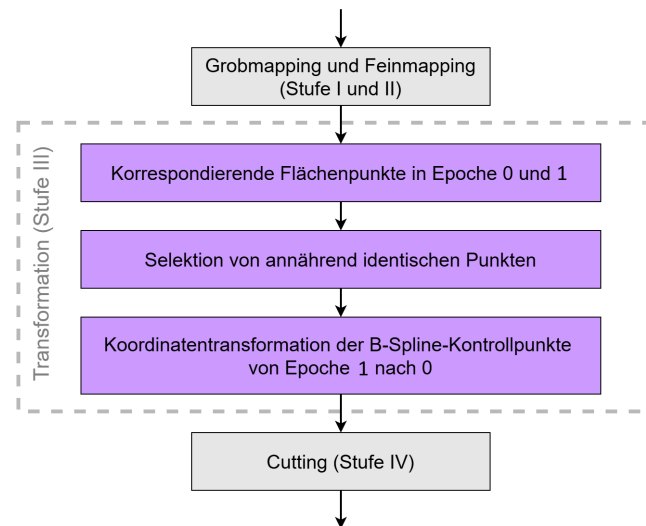


Abb. 4.31: Ablauf von Stufe III des Mapcut-Algorithmus für B-Spline-Flächen (Transformation).

Diese Vorgehensweise soll eine gleichmäßig verteilte Abfrage von Flächenpunktpaaren gewährleisten. Da in der ersten Iteration davon ausgegangen wird, dass durch die differierenden Ränder beider Epochen insgesamt mehr Bereiche größere Koordinatendifferenzen aufweisen, die sich nicht für eine Auswahl an identischen Punkten eignen, werden mit 6 von 16 identischen Punkten etwa 40 % verwendet. Die Untermenge von sechs identischen Punkten hat sich empirisch als vorteilhaft für die verwendeten Daten herausgestellt. Beispielhaft kann eine Selektion identischer Punkte für die erste Iteration mit der linken Grafik in Abbildung 4.32 angegeben werden.

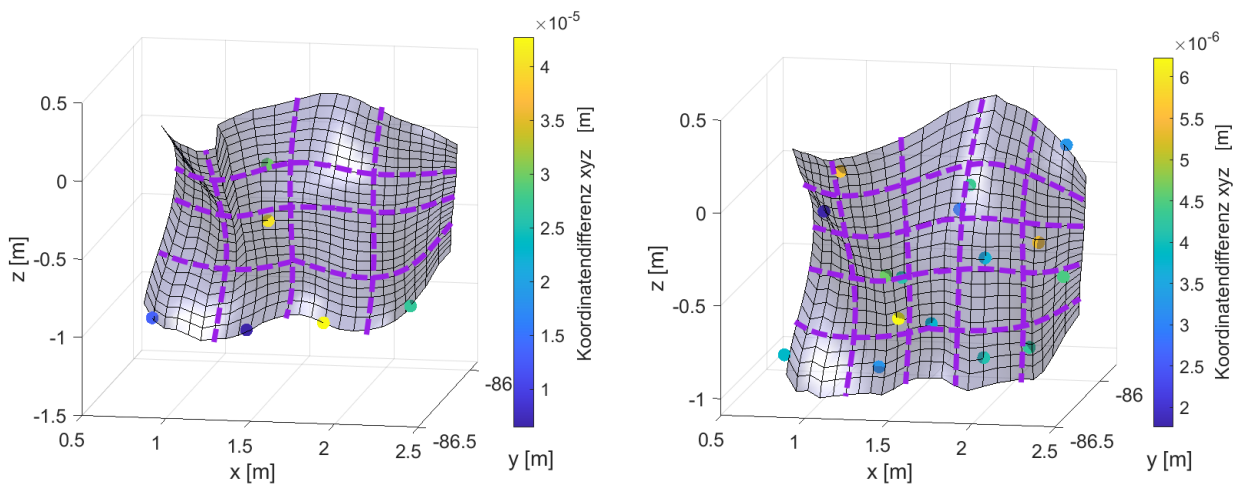


Abb. 4.32: Selektion von annähernd identischen Punkten für einen Beispieldatensatz (links: erste Iteration, rechts: letzte Iteration)

Für gewöhnlich verbessert sich durch die erste Iteration die Zuordnung der korrespondierenden Flächenpunkte. Dies hat zur Folge, dass nach der ersten Iteration bei der Auswahl der identischen Punkte in einer feineren Untergliederung gesucht und mehr Punkte verwendet werden können. Deshalb werden die B-Spline-Flächen ab der zweiten Iteration gleichmäßig in 5×5 Kacheln aufgeteilt und nach korrespondierenden Flächenpunkten mit der geringsten Koordinatendifferenz gesucht. Unter den 25 ausgewählten Flächenpunktpaaren werden 16 mit der geringsten Koordinatendifferenz selektiert und als annähernd identische Punkte in die Transformation überführt. Die Auswahl von 16 von 25

Flächenpunktpaaren entspricht etwas mehr als 60 %. Dieser Prozentsatz hat sich ebenso empirisch für die genutzten Daten herausgebildet. Für andere Datensätze sollte diese numerische Festlegung evaluiert werden. Die rechte Grafik von Abbildung 4.32 zeigt eine beispielhafte Selektion von 16 annähernd identischen Punkten in einer Untergliederung von 25 Teilen.

Nach der Selektion von annähernd identischen Punkten können die Kontrollpunkte von Epoche 1 mittels einer Koordinatentransformation in das System von Epoche 0 transformiert werden. Hierzu wird das Gauß-Helmert-Modell in Bezug auf eine 3D-Koordinatentransformation verwendet. Analog zum 2D-Kurvenfall treten gleichermaßen beim 3D-Flächenfall durch unterschiedliche Randbegrenzungen bei den Kontrollgittern der zwei Epochen in der ersten Iteration größere Koordinatendifferenzen zwischen den zugehörigen Flächenpunktpaaren auf. Da die B-Spline-Fläche aufgrund der Eckpunkt-Eigenschaft mit den Eckpunkten des Kontrollgitters zusammenfällt und sich der Randkurve annähert, entstehen bei unterschiedlichen Rändern der beiden Epochen unterschiedliche Krümmungsverhalten und Abstände zwischen den B-Spline-Flächenpunkten.

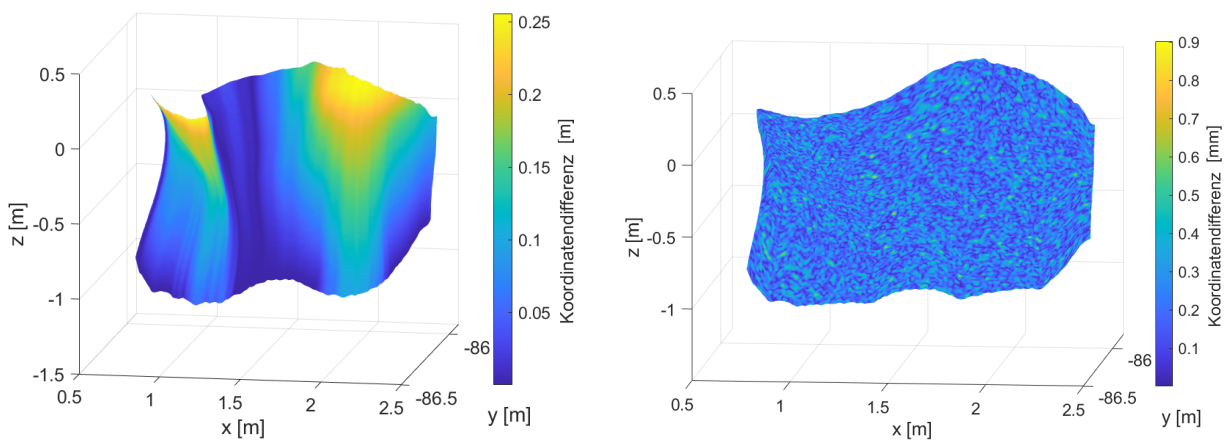


Abb. 4.33: Koordinatendifferenzen von Epoche 0 und Epoche 1 für einen Beispieldatensatz (links: erste Iteration, rechts: letzte Iteration).

Beispielhaft wird dieses Verhalten in der linken Grafik von Abbildung 4.33 ersichtlich. Durch die unterschiedlichen Randabgrenzungen vergrößert sich die Koordinatendifferenz zwischen als korrespondierend zugewiesenen Flächenpunktpaaren zunehmend nach oben hin. Um diese Auswirkung unterschiedlicher Ränder abzumildern, wird das für den Kurvenfall bereits beschriebene Verfahren des Cuttings für Flächen angewendet. Hierbei wird die größere Punktwolke bzw. deren Kontrollgitter mittels der Randkontrollpunkte der kleineren Punktwolke ausgestanzt. Dies führt zu einer Angleichung der Randkontrollpunkte der Kontrollgitter beider Epochen. Da die Angleichung der Ränder in den ursprünglichen Kontrollgittern angewendet wird, ist nach dem Cutting entsprechend des Ablaufdiagramms von Abbildung 4.27 eine erneute Anwendung des Algorithmus von Böhm durchzuführen. Die Anwendung des Algorithmus von Böhm erzeugt eine bidirektionale Kontrollgitterstruktur, die eine Generierung von Tensorprodukt B-Spline-Flächen ermöglicht. Eine iterative Abfolge des Mapcut-Algorithmus innerhalb der DBKM verbessert sukzessive Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte. Eine verbesserte Überlagerung zweier Epochen für den dargestellten Beispieldatensatz kann in der rechten Grafik von Abbildung 4.33 eingesehen werden. Die Koordinatendifferenzen sind hierbei deutlich geringer als im ersten Iterationsschritt. Mit dieser Grundlage können die Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte zusammen mit der Verwendung von den Standardabweichungen der zu betrachtenden B-Spline-Flächenpunkte in einem Signifikanztest auf eine mögliche Deformation untersucht werden. Die stochastische Modellierung der Methode, die im Vorfeld der Deformationsanalyse notwendig ist, wird daher in Kapitel 5 behandelt.

4.4 Zusammenfassung des Kapitels

In diesem Kapitel wird die neuartige DBKM vorgestellt, die sowohl für einen 2D- wie auch 3D-Fall arbeitet. Sie besteht aus einer B-Spline-Modellierung, einem Mapcut-Algorithmus zur Registrierung, einer stochastischen Modellierung und einer abschließenden Deformationsanalyse. Auf die stochastische Modellierung und Deformationsanalyse wird in Kapitel 5 eingegangen. Für die B-Spline-Modellierung werden die TLS-Daten unmittelbar als Kontrollpunkte verwendet. Die Anwendung eines Knoteneinfügealgorithmus erstellt ein bidirektionales Kontrollpunktgitter als Basis für die Tensorprodukt B-Spline-Flächengenerierung. Für jeden Kurven- bzw. Flächenpunkt wird ein Krümmungsparameter ermittelt, der im Mapcut-Algorithmus durch Korrelationsanalyse eine Registrierung zweier Epochen ermöglicht. Der Mapcut-Algorithmus besteht aus vier Stufen: Grobmapping, Feinmapping, Koordinatentransformation und Cutting. Ein iterativer Prozess verbessert dessen Ergebnisse. Das Resultat der B-Spline-Modellierung und des Mapcut-Algorithmus sind korrespondierende B-Spline-Kurven- bzw. -Flächenpunkte im gleichen Koordinatensystem. Mittels stochastischer Modellierung kann mit der zugehörigen Kofaktormatrix ein Signifikanztest betreffend Deformationen durchgeführt werden. Die stochastische Modellierung und der Signifikanztest werden im nächsten Kapitel erläutert.

5 Stochastische Modellierung

Um für jeden B-Spline-Kurven- bzw. -Flächenpunkt eine Entscheidung betreffend einer signifikanten Deformation vornehmen zu können, bedarf es der Hinzunahme von Informationen über die Standardabweichung jedes Kurven- bzw. Flächenpunktes. Dies wird mittels Varianz-Kovarianzfortpflanzung realisiert. Für die in diesem Kapitel durchgeführte Varianz-Kovarianzfortpflanzung sei erwähnt, dass zur Vereinfachung auf die vollständige Benennung des Wortes Varianz-Kovarianzfortpflanzung verzichtet wird und statt dessen das Wort Kovarianzfortpflanzung genannt wird. Hiermit ist jedoch sowohl die Fortpflanzung der Kovarianzen als auch Varianzen gemeint. Im Folgenden wird die Berechnung der gesuchten Kovarianzmatrix anhand eines Ablaufdiagramms (siehe Abbildung 5.1 und 5.3) sowie einer Beschreibung der elementaren Rechenschritte, unterteilt in Kurven- sowie Flächenfall, vorgestellt. Für die initialen Laserscanpunkte wird ein Gaußsches Messrauschen von 0,5 mm angenommen, welches der Standardabweichung des terrestrischen Laserscanners Z+F Imager 5016 und des Leica HDS7000 entspricht (Kersten et al. 2020; Leica Geosystems 2011). Die Annahme des vereinfachten Ansatzes in Form von Gaußschem Rauschen wird herangezogen, da es sich bei dieser Methode um erste Ansätze und Nachforschungen in diesem Themengebiet handelt. Eine Erweiterung des entwickelten Ansatzes kann für zukünftige Arbeiten mit einer korrelierten Varianz-Kovarianzmatrix nach Kerekes und Schwieger (2025) erfolgen.

5.1 Varianz-Kovarianzfortpflanzung (B-Spline-Kurven)

Für die Bestimmung einer Varianz-Kovarianzmatrix der B-Spline-Kurvenpunkte von Epoche 0 und Epoche 1 ist der vollständige Methodenprozess überwiegend mittels Kovarianzfortpflanzungsgesetz stochastisch nachzuvollziehen. Die Abfolge dieser statistischen Auswertung soll anhand des Diagramms in Abbildung 5.1 beschrieben werden.

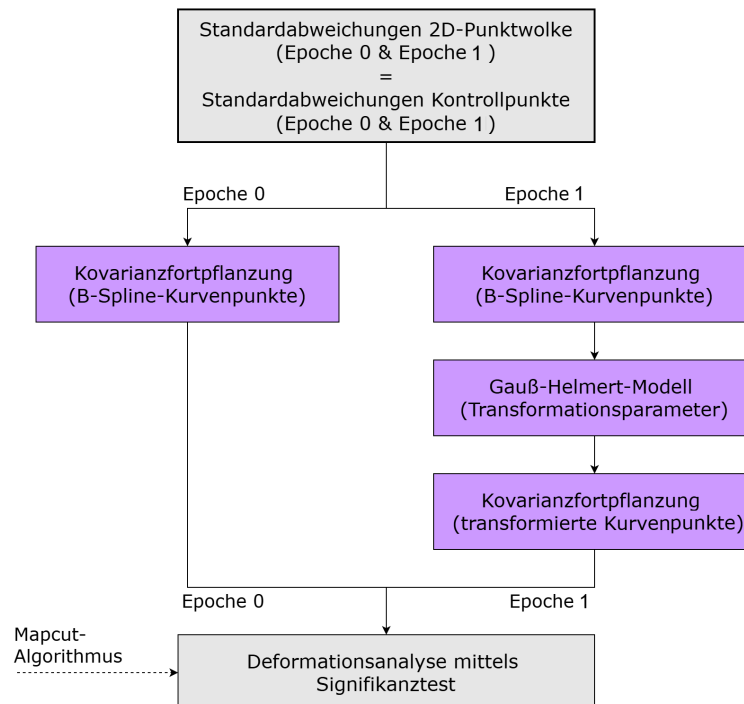


Abb. 5.1: Ablaufdiagramm der statistischen Auswertung der DBKM (B-Spline-Kurven).

Mittels der Standardabweichungen der TLS-Punkte, die identisch mit den Standardabweichungen der Kontrollpunkte sind, kann auf Basis der Kovarianzfortpflanzung sowohl eine Varianz-Kovarianzmatrix der B-Spline-Kurvenpunkte von Epoche 0 als auch von Epoche 1 aufgestellt werden. Von Epoche 0 sind die Berechnungen für die stochastische Modellierung damit abgeschlossen. Für Epoche 1 kann mit Hilfe des Gauß-Helmert-Modells eine Varianz-Kovarianzmatrix der Transformationsparameter bestimmt werden. Auf Basis einer kombinierten Varianz-Kovarianzmatrix, die stochastische Informationen der B-Spline-Flächenpunkte und Transformationsparameter enthält, kann daraufhin in einer Kovarianzfortpflanzung eine Ermittlung der Varianz-Kovarianzmatrix der transformierten Kurvenpunkte von Epoche 1 realisiert werden.

Kovarianzfortpflanzung: B-Spline-Kurvenpunkte

Für die Berechnung der Varianz-Kovarianzmatrix der B-Spline-Kurvenpunkte auf Basis der Standardabweichungen der Kontrollpunkte bzw. der ursprünglichen Laserscanpunkte, kann zur Herleitung des funktionalen Modells auf Gleichung 2.24 von Abschnitt 2.2 zur Berechnung von B-Spline-Kurven zurückgegriffen werden:

$$\mathbf{C}(t) = \sum_{i=0}^m \mathbf{d}_i \cdot N_i^n(t), \quad t \in [u_0, u_k] = [u_n, u_{k-n}].$$

Bezüglich der Definition der in der Formel enthaltenen Indizes sei in diesem Fall sowie bei den folgenden Gleichungen betreffend funktionaler Modelle auf die jeweiligen Abschnitte verwiesen.

Zur Bestimmung der Varianz-Kovarianzmatrix der B-Spline-Kurvenpunkte Σ_C lautet die Formel des zugehörigen Kovarianzfortpflanzungsgesetzes wie folgt:

$$\Sigma_C = \mathbf{F} \cdot \Sigma_d \cdot \mathbf{F}^T, \quad (5.1)$$

wobei Σ_d der Varianz-Kovarianzmatrix der Kontrollpunkte und $\mathbf{F}$ der Jacobi-Matrix mit den jeweiligen partiellen Ableitungen entspricht. Dieser Rechenschritt wird sowohl für die Kurvenpunkte von Epoche 0 als auch für die zu vergleichende Messepoche Epoche 1 durchgeführt. Eine detaillierte Aufschlüsselung der Jacobi-Matrix zu dieser Kovarianzfortpflanzung findet sich im Anhang in Unterkapitel A.2.

Gauß-Helmert-Modell: Transformationsparameter

Im Vorgang der entwickelten DBKM werden identische B-Spline-Kurvenpunkte selektiert, die via Helmertransformation die Berechnung von Transformationsparametern gewährleisten.

Der funktionale Zusammenhang zwischen den unbekannten Transformationsparametern x_0 , y_0 , q und Φ sowie den bekannten identischen Punkten liefert die Transformationsgleichung zur Berechnung der Koordinaten im (x, y) -System (Kahmen 2006):

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + q \cdot \begin{pmatrix} \cos(\Phi) & -\sin(\Phi) \\ \sin(\Phi) & \cos(\Phi) \end{pmatrix} \cdot \begin{pmatrix} \xi_j \\ \eta_j \end{pmatrix} \quad \text{mit } j = 1, \dots, z, \quad (5.2)$$

wobei z der Anzahl an ausgewählten identischen Punkten entspricht. Durch Substitution kann dieser Zusammenhang mit $a = q \cdot \cos(\Phi)$ und $o = q \cdot \sin(\Phi)$ vereinfacht werden:

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} a & -o \\ o & a \end{pmatrix} \cdot \begin{pmatrix} \xi_j \\ \eta_j \end{pmatrix}, \quad (5.3)$$

$$\begin{aligned} x_j &= x_0 + a \cdot \xi_j - o \cdot \eta_j, \\ y_j &= y_0 + o \cdot \xi_j + a \cdot \eta_j. \end{aligned} \quad (5.4)$$

Aus Gleichung 5.4 ist ersichtlich, dass das Gleichungssystem nicht derart umgestellt werden kann, dass die Beobachtungen $\tilde{\mathbf{l}} = (\xi_1, \eta_1, \dots, \xi_z, \eta_z, x_1, y_1, \dots, x_z, y_z)$ von den unbekannten Parametern $\hat{\mathbf{x}} = (\hat{x}_0, \hat{y}_0, \hat{a}, \hat{o})$ separierbar sind. Aufgrund dessen ist eine Anwendung des Gauß-Markov-Modells nicht möglich, weshalb sich eine Durchführung des Gauß-Helmert-Modells für die Helmerttransformation anbietet (Pelzer 1985).

Die Bedingungsgleichungen des Gauß-Helmert-Modells, die Ähnlichkeitstransformation zur Beschaffung von Näherungswerten, die Linearisierung nach Taylor sowie weitere notwendige Rechenschritte können in Kahmen (2006) nachgelesen werden. Am Ende des Ausgleichsprozesses resultiert die Kofaktormatrix der ausgeglichenen Unbekannten $\mathbf{Q}_{\hat{\mathbf{x}}}$ und der Aposteriori-Varianzfaktor $\hat{\sigma}_0^2$ mit deren Unterstützung die Aposteriori-Varianz-Kovarianzmatrix der Unbekannten $\hat{\Sigma}_{\hat{\mathbf{x}}}$ in Form der Transformationsparameter ermittelt werden kann (Jäger et al. 2005):

$$\hat{\Sigma}_{\hat{\mathbf{x}}} = \hat{\sigma}_0^2 \cdot \mathbf{Q}_{\hat{\mathbf{x}}} = \hat{\Sigma}_{\text{Tp}}. \quad (5.5)$$

Kovarianzfortpflanzung: Transformierte B-Spline-Kurvenpunkte

Die Durchführung des Gauß-Helmert-Modells bringt eine Varianz-Kovarianzmatrix der ausgeglichenen Unbekannten bzw. Transformationsparameter $\hat{\Sigma}_{\text{Tp}}$ hervor, die unter Berücksichtigung der Varianz-Kovarianzmatrix der Kurvenpunkte Σ_{C} der Epoche 1, zu einer Varianz-Kovarianzmatrix Σ_{CTp} generiert werden kann, die Elemente beider Matrizen enthält. Eine Einbeziehung von Kovarianzen der Kurvenpunkte und Transformationsparameter findet hierbei nicht statt. Diese werden als unkorreliert angenommen. Mit Hilfe der Kovarianzfortpflanzung auf Basis des funktionalen Modells der Gleichungen 5.4:

$$\begin{aligned} x_j &= x_0 + a \cdot \xi_j - o \cdot \eta_j, \\ y_j &= y_0 + o \cdot \xi_j + a \cdot \eta_j \end{aligned}$$

kann die Varianz-Kovarianzmatrix der transformierten Kurvenpunkten Σ_{tC} von Epoche 1 aufgestellt werden:

$$\Sigma_{\text{tC}} = \mathbf{F} \cdot \Sigma_{\text{CTp}} \cdot \mathbf{F}^T. \quad (5.6)$$

Mittels der Koordinatendifferenzen zweier zugehöriger B-Spline-Kurvenpunkte von Epoche 0 und Epoche 1 sowie deren Standardabweichungen, die mit Hilfe statistischer Auswertung bestimmt wurden, sind die Voraussetzungen zur Anwendung eines Signifikanztests für B-Spline-Kurven (Unterkapitel 5.4) gegeben.

Da die Wahl der Schrittweite Einfluss auf die Größe der Matrizen hat, die in der Kovarianzfortpflanzung berechnet werden, soll untersucht werden, ob bei Anwendung unterschiedlicher Schrittweiten andere Standardabweichungen für die Kurvenpunkte resultieren oder ob ein einheitliches Muster erreicht

wird. Für eine B-Spline-Kurve werden die Standardabweichungen bei den Schrittweiten (0,1), (0,05) und (0,01) berechnet. Abbildung 5.2 zeigt für die ersten 21 Kurvenpunkte einer B-Spline-Kurve die Standardabweichungen je nach Schrittweite.

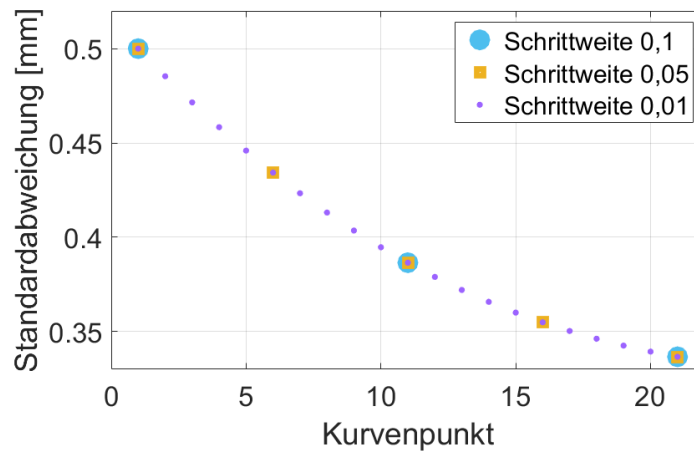


Abb. 5.2: Standardabweichung der Kurvenpunkte im Bereich des Anfangskurvenpunktes je nach Schrittweite. Numerische Darstellung der Werte der Standardabweichungen in Tabelle 5.1.

Kurvenpunkt	Standardabweichung bei Schrittweite 0,1 [mm]	Standardabweichung bei Schrittweite 0,05 [mm]	Standardabweichung bei Schrittweite 0,01 [mm]
1	0,5	0,5	0,5
2 - 5			0,49; 0,47; 0,46; 0,45
6		0,43	0,43
7 - 10			0,42; 0,41; 0,40; 0,39
11	0,39	0,39	0,39
12 - 15			0,38; 0,37; 0,36; 0,36
16		0,35	0,35
17 - 20			0,35; 0,35; 0,34; 0,34
21	0,34	0,34	0,34
> 22 < 1.978	0,34 + / - 0,02	0,34 + / - 0,02	0,34 + / - 0,02

Tab. 5.1: Vergleich des Einflusses der Schrittweite auf die Standardabweichungen von B-Spline-Kurvenpunkten für einen Teilkurvenbereich. Grafische Darstellung der Standardabweichungen in Abbildung 5.2.

Die Kurvenpunkte mit der kleinsten Schrittweite von 0,01 werden mit einem kleinen magentafarbenen Punkt dargestellt. Die größere Schrittweite 0,05 wird für die Kurvenpunkte als gelbes Quadrat und die Standardabweichungen der größten Schrittweite von 0,1 mit einem hellblauen Kreis abgebildet. Die hinterlegten Werte sind in Tabelle 5.1 aufgeführt. Hierbei zeigt sich, dass die initiale Standardabweichung von 0,5 mm für den ersten Kurvenpunkt der B-Spline-Kurve beibehalten wird. Dies kann mit der Anfangs- und Endpunkt-Eigenschaft (siehe Unterkapitel 2.2.3) erklärt werden. Die Anfangs- und Endpunkt-Eigenschaft von B-Spline-Kurven gibt vor, dass der Anfangs- und Endpunkt der B-Spline-Kurve mit dem Anfangs- und Endpunkt des Kontrollpolygons zusammenfallen muss, wodurch die Standardabweichung von 0,5 mm erhalten bleibt. Dieser Einfluss schwächt sich mit zunehmendem Abstand zum Anfangs- bzw. Endkurvenpunkt ab und pendelt sich bei einer Standardabweichung von 0,34 mm + / - 0,02 mm für Kurvenpunkte ab dem 22-ten und bis zum 1978-ten Kurvenpunkt bei einer Gesamtanzahl an

Kurvenpunkten von 2.000 ein. Die Kurvenpunkte 1.980 bis 2.000 folgen wiederum dem Muster analog zu den Standardabweichungen der Kurvenpunkte 1 bis 21. Dies zeigt, dass unterschiedliche Schrittweiten zu den gleichen Standardabweichungen führen und lediglich durch die höhere Diskretisierung ein feineres Abbild der Standardabweichungen der Kurvenpunkte erzeugt wird. Bei empirischen Untersuchungen unterschiedlicher 2D-TLS-Daten und dem Vergleich der Standardabweichungen der mit der DBKM erzeugten B-Spline-Kurven, haben sich durchgängig gleiche Werte und Verhaltensweisen für Epoche 0 und Epoche 1 gezeigt. Die stochastische Modellierung von mittels DBKM erzeugter B-Spline-Kurven bildet sich demzufolge derart ab, dass die Standardabweichung des Anfangs- und Endkurvenpunktes bei der initialen Standardabweichung von 0,5 mm erhalten bleibt. Außerhalb des Einflussbereichs der Anfangs- und Endkurvenpunkte der B-Spline-Kurve kann eine Standardabweichung von $0,34 + / - 0,02$ mm angenommen werden.

5.2 Varianz-Kovarianzfortpflanzung (B-Spline-Flächen)

Für die Berechnung der Varianz-Kovarianzmatrix der B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 bei 3D-Laserscandaten wird in analoger Weise wie im 2D-Kurvenfall vorgegangen. Entsprechend des Ablaufdiagramms in Abbildung 5.3 ist ersichtlich, dass allerdings in Relation zum 2D-Fall in einem ersten Schritt die Auswirkungen des Algorithmus von Böhm berücksichtigt werden müssen.

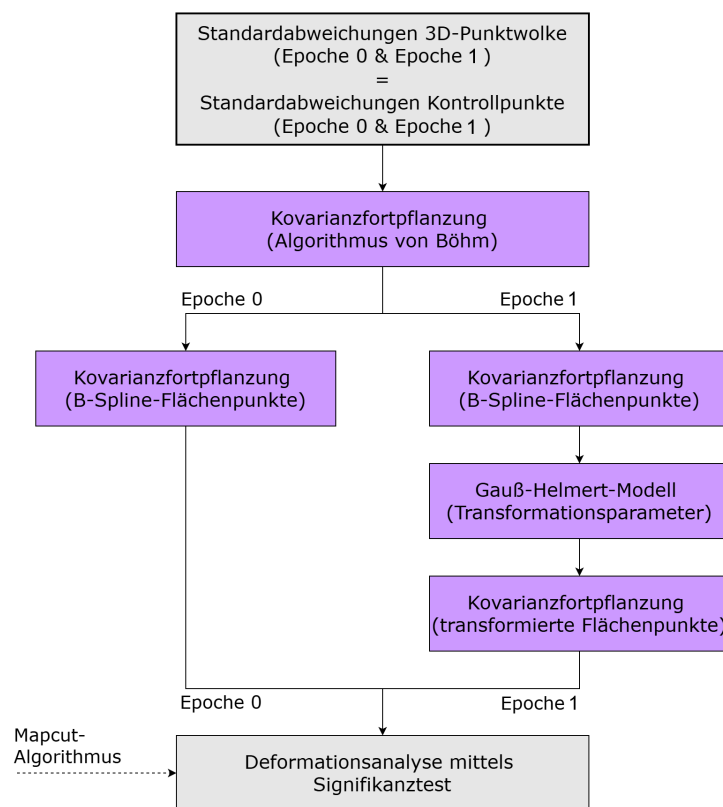


Abb. 5.3: Ablaufdiagramm der statistischen Auswertung der DBKM (B-Spline-Flächen).

Hierzu werden mittels der Standardabweichungen der Laserscanpunkte, die identisch mit den Standardabweichungen der Kontrollpunkte sind, auf Basis der Kovarianzfortpflanzung sowohl eine Varianz-Kovarianzmatrix von Epoche 0 als auch von Epoche 1 für den Prozess des Algorithmus von Böhm aufgestellt, um daraufhin ebenso mit Hilfe von Kovarianzfortpflanzung die Varianz-Kovarianzmatrizen der B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 zu generieren. Während für Epoche 0 keine weiteren Berechnungen notwendig sind, kann für Epoche 1 mit Hilfe

des Gauß-Helmert-Modells eine Varianz-Kovarianzmatrix der Transformationsparameter bestimmt werden. Auf Basis einer kombinierten Varianz-Kovarianzmatrix, die stochastische Informationen der B-Spline-Flächenpunkte und Transformationsparameter enthält, kann in einer Kovarianzfortpflanzung eine Ermittlung der Varianz-Kovarianzmatrix der transformierten Kurvenpunkte von Epoche 1 realisiert werden.

Kovarianzfortpflanzung: Neu eingefügte Kontrollpunkte (Algorithmus von Böhm)

Für die Aufstellung des funktionalen Modells betreffend der Kovarianzfortpflanzung des Algorithmus von Böhm, kann auf Gleichung 4.4 des in Abschnitt 4.2.2.1 beschriebenen Algorithmus von Böhm als Knoteneinfügealgorithmus zurückgegriffen werden:

$$\mathbf{q}_i = (1 - \gamma_i) \cdot \mathbf{d}_{i-1} + \gamma_i \cdot \mathbf{d}_i \quad \text{mit}$$

$$\gamma_i = \begin{cases} 1 & i \leq s - n \\ \frac{\bar{u} - u_i}{u_{i+n} - u_i} & s - n < i \leq s \\ 0 & i < s + 1 \end{cases}.$$

Die Varianz-Kovarianzmatrix der neu erzeugten Kontrollpunkte Σ_q wird dementsprechend mit Hilfe der Varianz-Kovarianzmatrix der ursprünglichen bzw. alten Kontrollpunkte Σ_d mittels Kovarianzfortpflanzung generiert. Eine detaillierte Beschreibung des Rechengangs betreffend Kovarianzfortpflanzung in Bezug auf den Algorithmus von Böhm kann im Anhang im Unterkapitel A.1 nachgelesen werden.

Kovarianzfortpflanzung: B-Spline-Flächenpunkte

Auf Basis der Varianz-Kovarianzmatrix, der durch den Algorithmus von Böhm eingefügten Kontrollpunkte kann mit Hilfe der B-Spline-Flächen-Gleichung 2.31 aus Abschnitt 2.3 das funktionale Modell hergeleitet werden:

$$\mathbf{S}(t_1, t_2) = \sum_{i=0}^m \sum_{j=0}^a N_i^n(t_1) \cdot N_j^q(t_2) \cdot \mathbf{d}_{i,j}, \quad (t_1, t_2) \in [u_0, u_k] \times [v_0, v_r].$$

Durch die Varianz-Kovarianzmatrix der neu erzeugten Kontrollpunkte Σ_q kann mittels Kovarianzfortpflanzungsgesetz die Varianz-Kovarianzmatrix der B-Spline-Flächenpunkte Σ_S berechnet werden. Die Formel zur Kovarianzfortpflanzung sowie eine Aufstellung der Jacobi-Matrix sind im Anhang in Unterkapitel A.3 zu finden.

Gauß-Helmert-Modell: Transformationsparameter

Wie im Kurvenfall wird zur Bestimmung der Standardabweichungen der Transformationsparameter zur Transformation der Laserscandaten und B-Spline-Flächenpunkte von Epoche 1 in das Koordinatensystem von Epoche 0 das Gauß-Helmert-Modell herangezogen, da wie im funktionalen Modell in Gleichung 5.7 einzusehen ist, eine Trennung von Beobachtungen und Unbekannten nicht möglich ist.

$$\begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + (1 + q) \cdot \begin{pmatrix} 1 & \epsilon_z & -\epsilon_y \\ -\epsilon_z & 1 & \epsilon_x \\ \epsilon_y & -\epsilon_x & 1 \end{pmatrix} \cdot \begin{pmatrix} \xi_j \\ \eta_j \\ \zeta_j \end{pmatrix}. \quad (5.7)$$

Die Ausgleichung erfolgt analog zur Vorgehensweise im 2D-Fall. Der Ausgleichungsprozess resultiert in einer Varianz-Kovarianzmatrix der ausgeglichenen Unbekannten bzw. Transformationsparameter $\hat{\Sigma}_{Tp}$.

Kovarianzfortpflanzung: Transformierte B-Spline-Flächenpunkte

Der letzte Schritt für die Berechnung der Standardabweichungen der transformierten B-Spline-Flächenpunkte von Epoche 1 besteht in der Anwendung der Varianz-Kovarianzfortpflanzung unter zu Hilfenahme des funktionalen Modells:

$$\begin{aligned} x_j &= x_0 + (1 + q) \cdot (\xi_j + \epsilon_z \cdot \eta_j - \epsilon_y \cdot \zeta_j), \\ y_j &= y_0 - (1 + q) \cdot (\epsilon_z \cdot \xi_j + \eta_j + \epsilon_x \cdot \zeta_j), \\ z_j &= z_0 + (1 + q) \cdot (\epsilon_y \cdot \xi_j - \epsilon_x \cdot \eta_j + \zeta_j). \end{aligned} \quad (5.8)$$

Die gesuchte Varianz-Kovarianzmatrix der transformierten B-Spline-Flächenpunkte Σ_{tS} wird mittels einer Varianz-Kovarianzmatrix Σ_{STp} , bestehend aus der Varianz-Kovarianzmatrix der B-Spline-Flächenpunkte von Epoche 1 und der Varianzen der Transformationsparameter, generiert:

$$\Sigma_{tS} = \mathbf{F} \cdot \Sigma_{STp} \cdot \mathbf{F}^T. \quad (5.9)$$

Das Vorhandensein der euklidischen Distanz, auf Basis der 3D-Koordinaten der B-Spline-Flächenpunkte von Epoche 0 und 1, sowie der jeweiligen Standardabweichungen der entsprechenden Flächenpunkte, versetzt in die Lage in einem Signifikanztest, der in Abschnitt 5.4 behandelt wird, eine Entscheidung treffen zu können, ob eine signifikante Deformation zwischen zwei zugehörigen B-Spline-Flächenpunkten vorliegt oder nicht. Zuvor soll allerdings auf die praktische Anwendung der stochastischen Modellierung und einen Ansatz zum Umgang mit großen Varianz-Kovarianzmatrizen eingegangen werden.

5.3 Simulation Varianz-Kovarianzfortpflanzung (B-Spline-Flächen)

Nachdem die Varianz-Kovarianzfortpflanzung vollständig theoretisch offengelegt wurde, ist ein Verweis auf die praktische Umsetzung sowie die Dimensionen der zu verwendenden Varianz-Kovarianzmatrizen unabdingbar. Zu beachten ist, dass sich die Größe der stochastischen Matrizen während des in Abschnitt 5.2 beschriebenen Prozesses erhöht. Näheres wird im Folgenden erläutert.

Die Anzahl an Laserscanpunkten, die in die entwickelte Methode einfließt, bestimmt die Dimension der initialen Varianz-Kovarianzmatrix. Durch die Definition von Kontrollpolygone und den Algorithmus von Böhm wird diese Dimension verändert, da teilweise Kontrollpunkte entfernt sowie zusätzliche Kontrollpunkte eingefügt werden. Bei der Bestimmung der B-Spline-Flächenpunkte auf Basis des Kontrollpunktgitters erhöht sich die Anzahl an Punkten und somit die Dimension der Varianz-Kovarianzmatrix in großem Umfang, da Flächenpunkte mit geringerer Schrittweite die Form der Kontrollpolygone dichter nachbilden. Ein Beispiel mit einer Laserscanpunktwolke bestehend aus 500 Punkten, die in Abbildung 5.4 in der linken Grafik dargestellt ist, soll diesen Sachverhalt verdeutlichen. Durch den Prozess der Definition von Kontrollpolygone sowie den Algorithmus von Böhm resultiert eine Varianz-Kovarianzmatrix bestehend aus 469 x 469 Einträgen. Das zugehörige Kontrollgitter kann in der rechten Grafik von Abbildung 5.4 eingesehen werden. Nach der Varianz-Kovarianzfortpflanzung für die Bestimmung von B-Spline-Flächenpunkten ergibt sich eine Matrixgröße von 26.281 x 26.281. Die Anzahl an B-Spline-Flächenpunkten hat sich somit maßgeblich erhöht. Bei der Durchführung der in dieser Arbeit überwiegend genutzten Punktwolkengrößen von mehr als 15.000 Punkten, würden sich demnach Varianz-Kovarianzmatrizen mit einer Größe von etwa 780.000.000 x 780.000.000 bilden. Zwar kann die Schrittweite der zu generierenden B-Spline-Flächenpunkte verringert werden, um kleinere Matrizen zu erhalten, allerdings genügt dies nicht, um Dimensionen zu erreichen, die kleiner eine Million betragen.

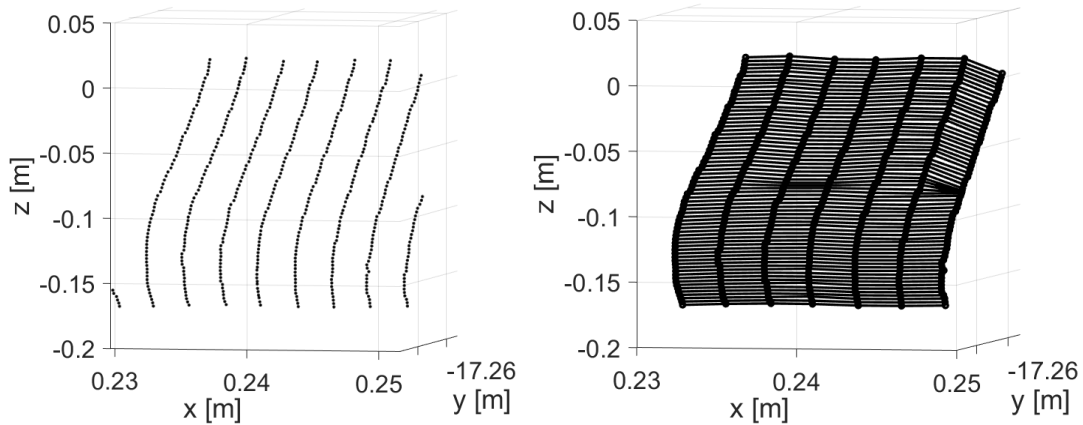


Abb. 5.4: links: 3D-Punktwolke bestehend 500 Laserscanpunkten (Beispieldatensatz), rechts: Kontrollgitter.

Aufgrund dieses Sachverhalts wird nach einer Alternative gesucht, die den stochastischen Prozess möglichst getreu abbildet und gleichzeitig ein Arbeiten mit extrem großen Matrizen vermeidet. Eine intensive Beleuchtung der resultierenden Standardabweichungen jedes B-Spline-Flächenpunktes für den Beispieldatensatz von Abbildung 5.4 zeigt, dass sich die initial angenommene Standardabweichung von 0,5 mm während der stochastischen Modellierung nach einem einheitlichen Schema verhält.

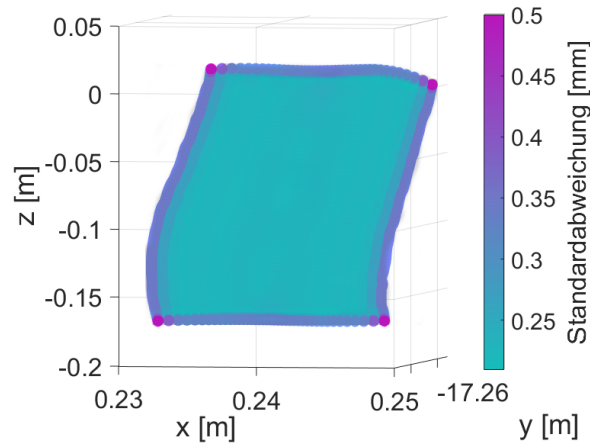


Abb. 5.5: Standardabweichungen der B-Spline-Flächenpunkte (Beispieldatensatz).

Abbildung 5.5 stellt B-Spline-Flächenpunkte des Beispieldatensatzes samt zugehöriger Standardabweichung in einer farblichen Skala dar. Flächenpunkte mit einer Standardabweichung von 0,5 mm werden in magenta dargestellt und Flächenpunkte mit einer geringeren Standardabweichung bis 0,21 mm werden in cyan abgebildet. Bildet man die einzelnen Flächenpunkte inklusive ihrer Standardabweichungen als rasterförmiges Gitter ab, wird ein Schema erkennbar, dass im Folgenden beleuchtet und in Abbildung 5.6 aufgezeigt wird. Die Eckpunkte der B-Spline-Fläche weisen eine Standardabweichung von 0,5 mm auf. Die B-Spline-Flächenpunkte am Rand der B-Spline-Fläche tragen bereits eine geringere Standardabweichung als die Eckpunkte. Die Standardabweichung verringert sich weiter mit der anschließenden Reihe an Flächenpunkten und wird mit Zunahme der Distanz zu den Eck- und Randpunkten nochmals verkleinert.

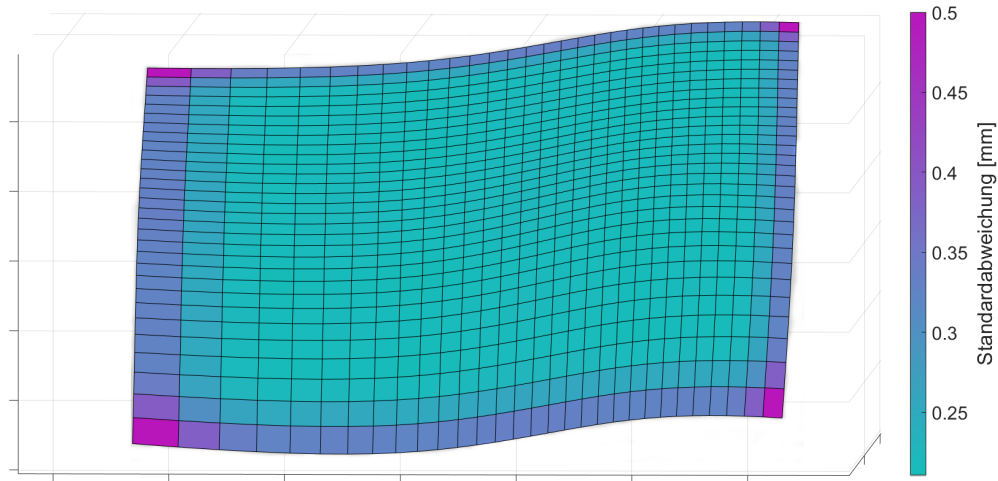


Abb. 5.6: Schema zur Bildung von simulierten Standardabweichungen für B-Spline-Flächen

Für eine Erklärung der Verhaltensweise der Standardabweichungen dienen die zwei Grafiken in Abbildung 5.7, die einen Zusammenhang zwischen der Generierung von B-Spline-Flächen, ihren typischen Eigenschaften sowie der Größe der Standardabweichungen herstellt.

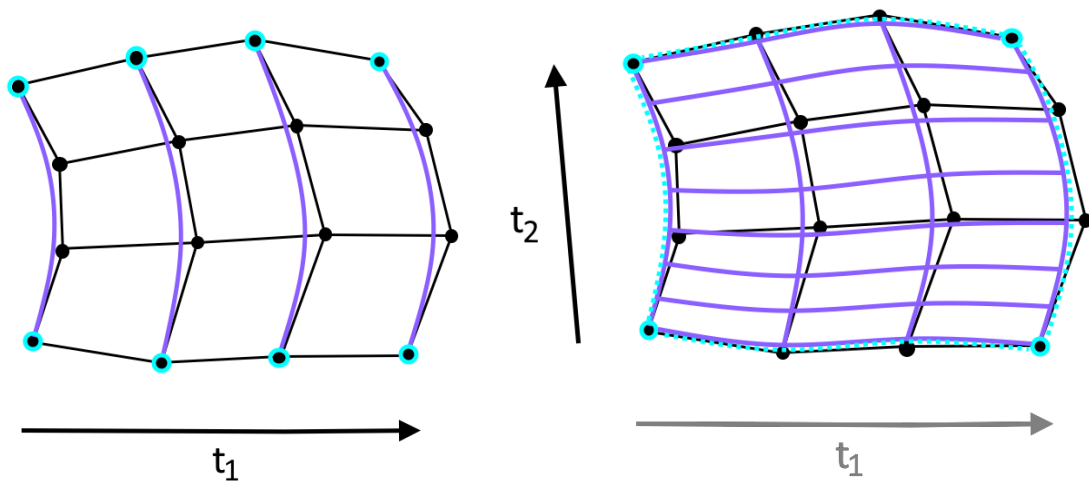


Abb. 5.7: Schematische Darstellung der B-Spline-Flächengenerierung mit Bezugnahme auf die Eckpunkt-Eigenschaft.

Die linke Grafik stellt 16 Kontrollpunkte eines Kontrollgitters dar, mit deren Unterstützung eine B-Spline-Fläche gebildet werden soll. Wie in Unterkapitel 2.3 beschrieben, besteht der erste Schritt aus einer Bildung von B-Spline-Kurven in t_1 -Richtung. Hierbei fallen die Anfangs- und Endpunkte der Kontrollpolygone mit denen der B-Spline-Kurven zusammen. Diese sogenannten Zwangspunkte sind in der linken Grafik in cyan dargestellt. Entsprechend der Verhaltensweise der Standardabweichungen von B-Spline-Kurven enthalten die Anfangs- und Endkurvenpunkte, die in cyan gefärbt sind, eine Standardabweichung von 0,5 mm die magenta gefärbten Kurvenpunkte außerhalb des Einflussbereichs der Anfangs- und Endpunkte eine Standardabweichung von $0,34 + / - 0,02$ mm. Schritt 2 der B-Spline-Flächenbildung besteht aus einer Generierung von B-Spline-Kurven in t_2 -Richtung auf Basis der in Schritt 1 gebildeten Kurven. Auch hierbei gilt die Anfangs- und Endpunkt-Eigenschaft. Somit

bleiben als ursprüngliche Zwangspunkte nur die vier Eckpunkte des Kontrollgitters übrig. Die Position der Randkurven hat ebenso eine Einflussnahme bei der B-Spline-Flächenbildung, auch wenn diese in Relation zu den Eckpunkten etwas abgeschwächer ist. Dies zeigt, warum die Standardabweichungen der B-Spline-Flächenpunkte durch die stochastische Modellierung der entwickelten Methode nach der Schemadarstellung in Abbildung 5.6 vorliegen. Ein vergrößerter Ausschnitt der Schemadarstellung mit expliziter Angabe der Standardabweichungen ist in Abbildung 5.8 dargestellt.

Wie bereits erwähnt, erhalten die Eckpunkt-B-Spline-Flächenpunkte eine Standardabweichung von 0,5 mm [A]. Die direkt benachbarten Flächenpunkte fallen mit 0,39 mm [B] etwas geringer aus. Die B-Spline-Flächenpunkte im gesamten Randbereich mit Ausnahme der Eckpunkte und ihrer Nachbarpunkte beinhalten eine Standardabweichung von 0,34 mm $\pm$ 0,02 mm [D]. Eine Reihe nach innen versetzt weist der zum Eckpunkt diagonal anschließende Flächenpunkt eine Standardabweichung von 0,30 mm [C] auf und ist demgemäß bereits kleiner als alle Eck- und Randpunkte. Die in zweiter Reihe liegenden Rand-Flächenpunkte haben eine Standardabweichung von 0,26 mm $\pm$ 0,01 mm [E]. Noch geringer fallen die Standardabweichungen im mittleren Bereich der B-Spline-Fläche aus, bei der alle Flächenpunkte, die nicht zu den äußeren zwei Reihen gehören Standardabweichungen von 0,23 mm $\pm$ 0,02 mm [F] aufweisen.

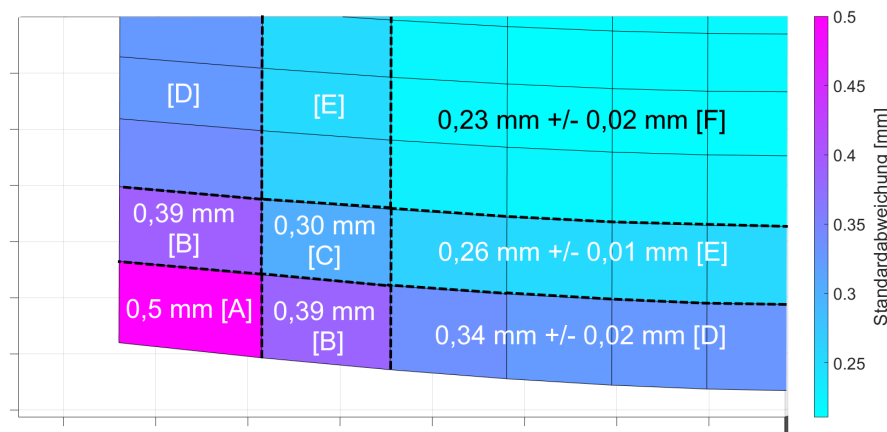


Abb. 5.8: Werte für simulierte Standardabweichungen von B-Spline-Flächen (Vergrößerter Ausschnitt).

Als offene Fragestellung verbleibt, inwiefern die Kontrollpunktanzahl oder die bei der B-Spline-Flächengenerierung festgelegte Schrittweite Einfluss auf die Höhe der Standardabweichungen haben. Um dies herauszufinden, werden Untersuchungen mit Punktwolken mit einer Anzahl an Kontrollpunkten zwischen 200 und 700 durchgeführt. Hierbei zeigen die Ergebnisse der Standardabweichungen, dass das Schema von Abbildung 5.8 exakt widerspiegelt wird. Die Anzahl an Kontrollpunkten hat demnach keine Auswirkung auf die Standardabweichungen. Der Einfluss der Schrittweite mit der die B-Spline-Fläche generiert wird, wird ebenso untersucht. Für die Prozessierung der B-Spline-Flächen wird in dieser Arbeit eine Schrittweite von 0,1 verwendet. Gegenübergestellt werden Standardabweichungen von B-Spline-Flächenpunkten, die mit einer Schrittweite von 0,2 generiert werden. Tabelle 5.2 zeigt die resultierenden Standardabweichungen in Relation zu einer Schrittweite von 0,1. Die erste Spalte beinhaltet die Flächenpunktbereiche, die in Abbildung 5.8 deklariert sind, Spalte zwei zeigt die Standardabweichungen für die Schrittweite 0,1 und Spalte drei für Schrittweite 0,2.

Zeile eins zeigt, dass die ursprüngliche Standardabweichung von 0,5 mm der Eckpunkte identisch ist. Die benachbarten Flächenpunkte im Bereich [B] sind bei einer Schrittweite von 0,2 mit einem Wert von 0,35 mm um 0,04 mm kleiner als bei einer Schrittweite von 0,1. Die diagonal zum Eckpunkt liegende Flächenpunkte im Bereich [C] fallen mit einem Wert von 0,23 mm 0,07 mm geringer aus.

Die Flächenpunkte am Rand im Bereich [D] sind hingegen beinahe identisch und nur minimal kleiner. Die Flächenpunkte der zweiten Reihe im Bereich [E] sind mit 0,23 mm $\pm$ 0,01 mm um etwa 0,03 mm kleiner als bei einer Schrittweite von 0,1. Der Flächenpunktbereich [F] weist keinen Unterschied zwischen den Schrittweiten auf.

Flächenpunktbereich	Standardabweichung bei Schrittweite 0,1 [mm]	Standardabweichung bei Schrittweite 0,2 [mm]
[A]	0,5	0,5
[B]	0,39	0,35
[C]	0,30	0,23
[D]	0,34 + / - 0,02	0,34 + / - 0,02
[E]	0,26 + / - 0,01	0,23 + / - 0,01
[F]	0,23 + / - 0,02	0,23 + / - 0,02

Tab. 5.2: Vergleich Einfluss Schrittweite auf Standardabweichungen von B-Spline-Flächenpunkten.

Zusammenfassend kann festgestellt werden, dass die größere Schrittweite von 0,2 eine Verringerung der Einflussnahme der Standardabweichungen der Eckpunkte bewirkt, da der lokale Einfluss hierdurch geringer wird. Auf diese Weise fallen die Standardabweichungen der direkt in der Nähe der Eckpunkte liegenden Flächenpunkte geringer aus, als bei einer Schrittweite von 0,1. Auf die Standardabweichungen der Flächenpunkte im Inneren der B-Spline-Fläche haben die hier verwendeten unterschiedlichen Schrittweiten allerdings keine Auswirkung.

Abschließend kann festgestellt werden, dass weder eine Änderung der Punktzahl noch der Schrittweite eine deutliche Veränderung des in Abbildung 5.8 dargestellten Schemas, vor allem bei den Standardabweichungen im Inneren ab der zweiten Flächenpunktreihe, bewirkt. Zwar hat die Schrittweite auf die Randbereiche der Flächenpunkte Einfluss, dies ist jedoch zweitrangig, da in dieser Arbeit ausschließlich Flächen mit Schrittweite 0,1 generiert werden und nur auf lokale Deformationen im Innenbereich untersucht wird. In Folge dieser Resultate wird das vorgestellte Schema als heuristische Methode herangezogen, um Standardabweichungen für größere Punktwolken zu simulieren.

5.4 Signifikanztest

Das Vorhandensein der Euklidischen Distanzen zugehöriger B-Spline-Kurven- bzw. Flächenpunkte sowie deren Standardabweichungen versetzt in die Lage einen Signifikanztest für die 3D-Deformationsanalyse durchzuführen. Hierbei wird unmittelbar ein Individualtest verwendet, der jeden B-Spline-Kurven- bzw. -Flächenpunkt auf eine mögliche Deformation untersucht. Die Anwendung eines Globaltests könnte Aufschluss darüber geben, ob grundsätzlich eine Deformation zwischen zwei zu untersuchenden Epochen besteht. Der Rechenaufwand zur Generierung einer inversen Kovaktormatrix der B-Spline-Flächenpunkte würde bei den hier verwendeten Punktwolkengrößen (bis 150.000 Punkte) allerdings Dimensionen im Millionenbereich annehmen. Da dieser Rechenaufwand nicht im Verhältnis zur Relevanz des Erkenntnisgewinns steht, wird auf einen Globaltest verzichtet.

Demzufolge wird ein Individualtest angewendet. Aufgrund der Fragestellung, ob zwischen zwei Epochen signifikante Punktveränderungen verifizierbar sind, bietet sich ein lokaler Kongruenztest mittels F-Test an. Hierbei werden ein Differenzvektor $\mathbf{d}$ für den 2D- und 3D-Fall:

$$\text{2D: } \mathbf{d} = \begin{pmatrix} x_{\text{Ep0}} \\ y_{\text{Ep0}} \end{pmatrix} - \begin{pmatrix} x_{\text{Ep1}} \\ y_{\text{Ep1}} \end{pmatrix} \text{ bzw. 3D: } \mathbf{d} = \begin{pmatrix} x_{\text{Ep0}} \\ y_{\text{Ep0}} \\ z_{\text{Ep0}} \end{pmatrix} - \begin{pmatrix} x_{\text{Ep1}} \\ y_{\text{Ep1}} \\ z_{\text{Ep1}} \end{pmatrix} \quad (5.10)$$

und die zugehörige Kofaktormatrix

$$\mathbf{Q}_{dd} = \mathbf{Q}_{00} + \mathbf{Q}_{11} \quad (5.11)$$

bestimmt. Die unkorrelierten Kofaktormatrizen $\mathbf{Q}_{00}$ von Epoche 0 und $\mathbf{Q}_{11}$ von Epoche 1 sind mithilfe der Standardabweichungen der Unterkapitel 5.1 und 5.3 aufzustellen. Die entsprechenden Hypothesen für einen B-Spline-Kurven- bzw. Flächenpunkt $\mathbf{X}_{\text{Ep0}}$ einer Epoche 0 und $\mathbf{X}_{\text{Ep1}}$ einer Epoche 1 lauten:

$$\text{Nullhypothese: } H_O : \mathbf{X}_{\text{Ep0}} = \mathbf{X}_{\text{Ep1}}, \quad (5.12)$$

$$\text{Alternativhypothese: } H_A : \mathbf{X}_{\text{Ep0}} \neq \mathbf{X}_{\text{Ep1}}. \quad (5.13)$$

Die Testgröße F kann mittels

$$F = \frac{\mathbf{d}' \cdot \mathbf{Q}_{dd}^{-1} \cdot \mathbf{d}}{p \cdot \sigma_0^2} \quad (5.14)$$

gebildet werden, wobei p der Anzahl der Unbekannten (2D: 2 Koordinaten, 3D: 3 Koordinaten) und $\sigma_0 = 1$ entspricht (Niemeier 2008).

Die Testgröße $F_{p,\infty,1-\frac{\alpha}{2}}$ kann aus einer Tabelle der F-Verteilung entnommen werden. Für α wird eine Irrtumswahrscheinlichkeit von $\alpha = 0,05$ herangezogen. Für den 2D-Fall beträgt $F_{2,\infty,0,975} = 3,69$ und für den 3D-Fall $F_{3,\infty,0,975} = 3,12$. Überschreitet die Testgröße das Quantil, wird die Nullhypothese verworfen und es liegt eine signifikante Deformation zwischen den zwei zu vergleichenden Punkten vor.

Damit keine Ausreißer der verwendeten Punktwolken als Deformation deklariert werden, werden im 2D-Fall B-Spline-Kurvenpunkte als deformiert abgebildet, wenn mindestens drei Laserscanpunkte betroffen sind. Im 3D-Fall müssen mindestens fünf Laserscanpunkte von einer Deformation betroffen sein. Diese Werte wurden empirisch ermittelt.

5.5 Zusammenfassung des Kapitels

Durch die stochastische Modellierung der entwickelten DBKM findet eine Berücksichtigung der Unsicherheiten in der Punktwolke bzgl. Messrauschen statt. Sowohl für den 2D-Fall als auch den 3D-Fall werden mittels Varianz-Kovarianzfortpflanzung und Gauß-Helmert-Modell die initial angenommenen Standardabweichungen der Laserscandaten fortgepflanzt. Ergebnis ist eine Kovarianzmatrix der B-Spline-Kurven- bzw. Flächenpunkte zweier zu vergleichender Epochen. Aufgrund der Datenmenge wird für den 3D-Fall eine simulierte Kovarianzmatrix auf Datengrundlage kleinerer Punktwolken erstellt. Ein Signifikanztest in Form eines F-Tests, der Differenzvektor und Kofaktormatrix einbezieht, bietet die Grundlage zur Durchführung einer Deformationsanalyse.

6 Evaluierungsdatensätze

Um die Funktionsfähigkeit und Wirksamkeit der entwickelten DBKM beurteilen zu können, werden vier Evaluierungsdatensätze verwendet. Diese sollen dabei unterstützen den aktuellen Stand der Methode zu dokumentieren sowie mögliche Stärken und Schwächen aufzudecken. Es wurde darauf geachtet jeweils zwei zu vergleichende Epochen mit einer unterschiedlichen Punktzahl in die DBKM einzuführen, um die Funktionsweise für Messepochen mit unterschiedlichen Laserscanausschnitten nachzuweisen. Da der Fokus verstärkt auf der Beurteilung der DBKM im Hinblick auf 3D-Punktwolken liegt, werden als Evaluierungsdatensätze eine 2D-Punktwolke und drei 3D-Punktwolken genutzt. Diese werden im Folgenden vorgestellt.

Fassadenstück (2D-Punktwolke)

Bei der 2D-Punktwolke handelt es sich um simulierte Daten eines Fassadenstücks. Für die simulierten Daten wird ein Gaußsches Rauschen von 0,5 mm angenommen. Die Wahl der Größe der Standardabweichung orientiert sich an der Genauigkeitsuntersuchung betreffend des terrestrischen Laserscanners Z+F Imager 5016 von Kersten et al. (2020).

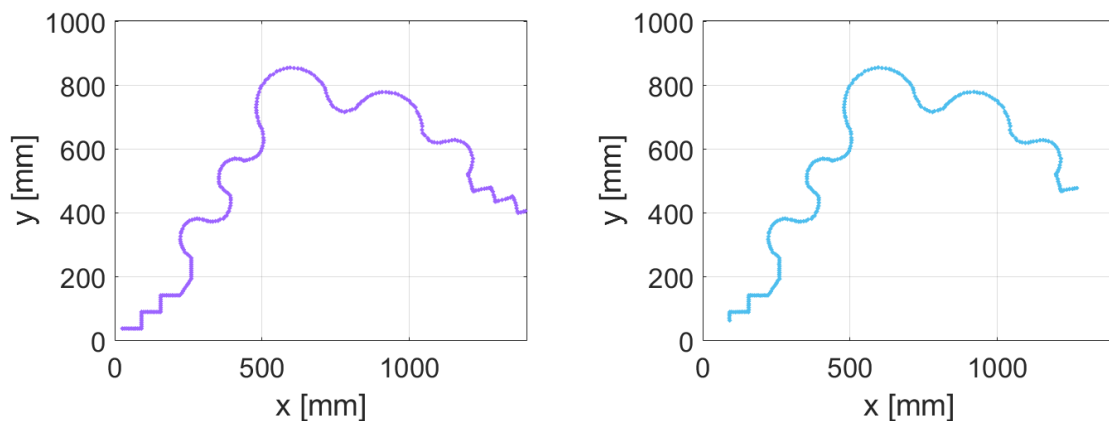


Abb. 6.1: 2D-Punktwolke eines Fassadenstücks von Epoche 0 (links) und Epoche 1 (rechts).

Epoche 0 besteht aus 319 Punkten und Epoche 1 aus 261 Punkten. Die aufeinanderfolgenden Laserscanpunkte, die als Kontrollpunkte verwendet werden sollen, weisen einen mittleren Punktabstand von 9 mm auf.

Gurkenblatt (3D-Punktwolke)

Als Entwicklungsdatensatz und gleichzeitig Evaluierungsdatensatz der Methode wird die 3D-Punktwolke eines Gurkenblatt-Testobjekts genutzt. Der in der linken Grafik von Abbildung 6.2 dargestellte Probekörper in Form eines Gurkenblatts wurde aus einem 0,4 m x 0,4 m x 0,2 m großen Aluminiumblock gefertigt.

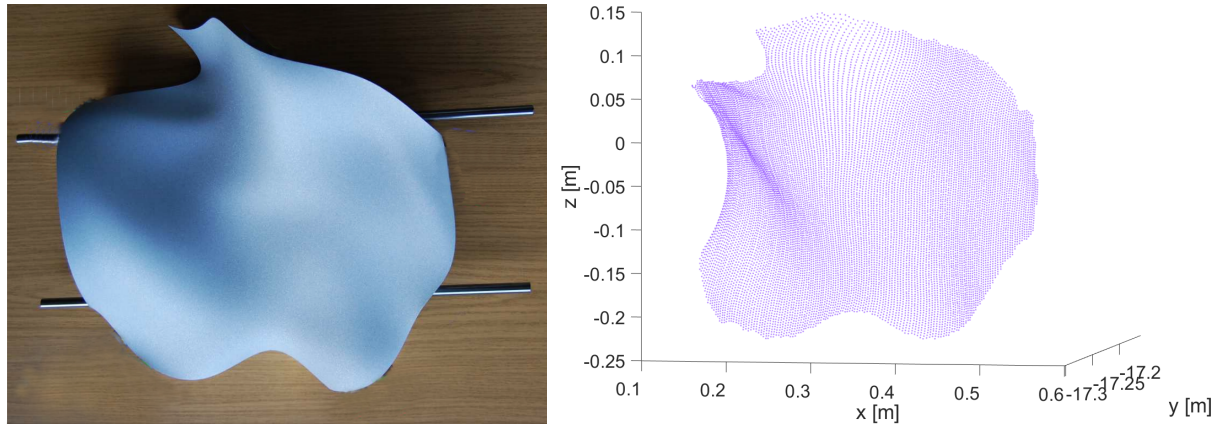


Abb. 6.2: links: Gurkenblatt-Testobjekt nach Harmening (2020), rechts: Laserscan-Punktwolke des Gurkenblatt-Testobjekts.

Die TLS-Punktwolke wurde mit einer Leica Nova MS50 Multistation erfasst und besteht aus 15.017 Punkten, die einen durchschnittlichen Punktabstand von 3 mm vorweisen (Harmening 2020; Harmening und Neuner 2016). Die zugehörige Punktwolke ist in der rechten Grafik von Abbildung 6.2 angegeben. Da der Datensatz für die Entwicklung der DBKM eingesetzt wurde und anfängliche Konflikte mit dem Messrauschen umgegangen werden sollten, wurde der mittlere Punktabstand mit dem Faktor 5 auf 15 mm vergrößert. Für die Simulation von Messepoche 1 wurde die verrauschte Punktwolke von Epoche 0 an den Rändern abgeschnitten, so dass Epoche 1 aus insgesamt 13.486 Punkten besteht.

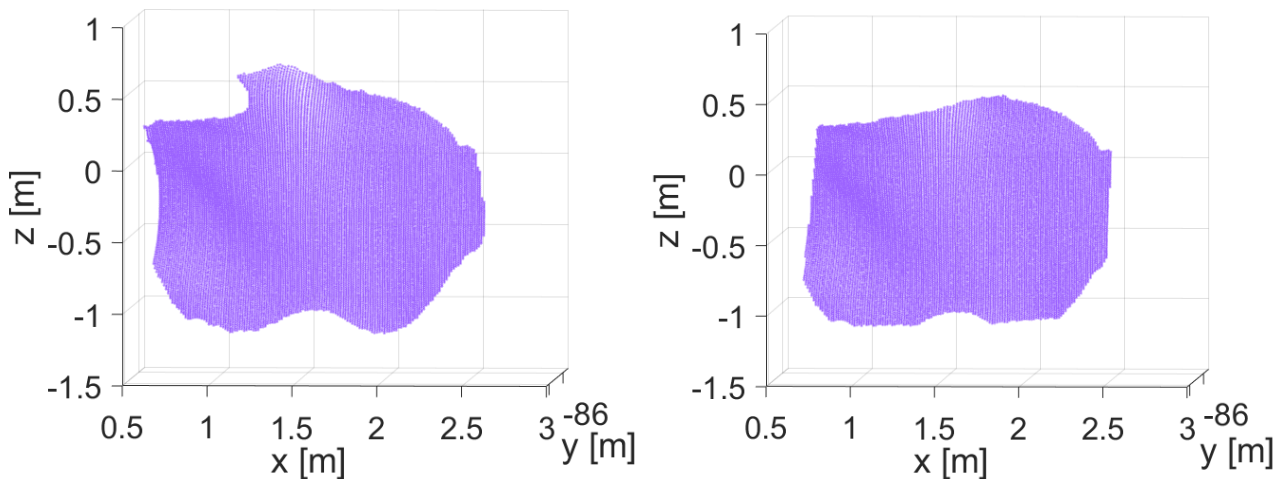


Abb. 6.3: 3D-Punktwolken des Gurkenblatt-Testobjekts von Epoche 0 (links) und Epoche 1 (rechts).

Die beiden in der Arbeit verwendeten 3D-Punktwolken des Gurkenblatts sind in Abbildung 6.3 dargestellt. Abgesehen von simulierten Daten besteht der Fokus auf der Evaluierung der DBKM an Realdaten, weshalb zwei weitere Testobjekte zur Erstellung von 3D-Punktwolken erzeugt wurden.

Tontorso (3D-Punktwolke)

Das erste Testobjekt für Realdaten stellt einen Ausschnitt der berühmten griechischen Skulptur Nike von Samothrake dar, die die Siegesgöttin Nike verkörpert und im Louvre in Paris ausgestellt ist. Die Skulptur zeigt die Siegesgöttin in Landeposition, wodurch ihre Flügel mit Luft erfüllt und ihr flatterndes Gewand am Körper anliegt (Bernhardt 2014). Abbildung 6.4 zeigt das Nikemonument sowie den in magenta markierten Bereich, der für das Testobjekt nachgebaut werden sollte.

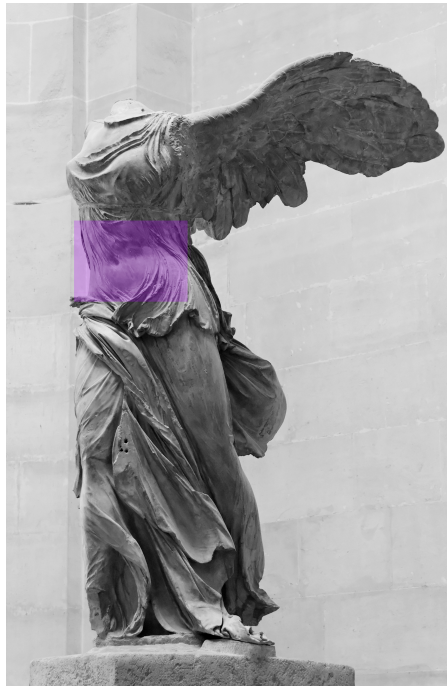


Abb. 6.4: Nike von Samothrake nach Nguyen (2007) (magenta: Modellierter Ausschnitt für Testobjekt aus Ton).

Angefertigt wurde das Testobjekt aus einer Tonmasse, die auf ein verformtes Drahtgeflecht angebracht wurde. Die Grafiken in Abbildung 6.5 zeigen den Herstellungsprozess sowie den fertig gestellten Tontorso der Nike Skulptur, der 10 kg wiegt und Maße von 0,5 m x 0,5 m aufweist.



Abb. 6.5: Herstellungsprozess und fertig gestelltes Testobjekt aus Ton (Tontorso).

Durch die Nachbildung der Skulptur im Bauchraum werden zum einen gleichmäßige Formänderungen abgebildet und zum anderen können durch die Nachmodellierung der Falten des Gewands markante Krümmungswechsel im Testobjekt aufgenommen werden.

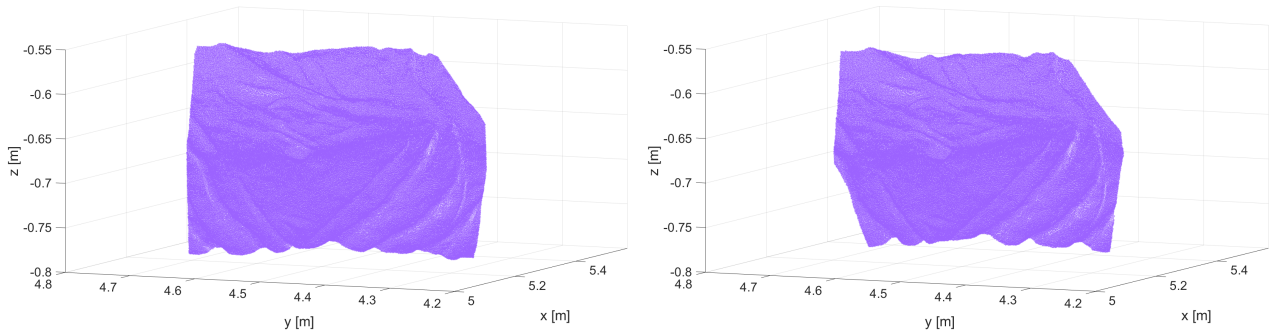


Abb. 6.6: 3D-Punktwolken des Tontorso-Testobjekts von Epoche 0 (links) und Epoche 1 (rechts).

Abbildung 6.6 zeigt die erfassten 3D-Punktwolken des Tontorso-Testobjekts von Epoche 0 und Epoche 1. Der Tontorso wurde mit dem Laserscanner Leica HDS7000 in der Auflösungsstufe High erfasst. Aufgrund des wie in Abbildung 6.9 dargestellten nahen Testaufbaus beträgt der durchschnittliche Punktabstand in den Punktwolken 1,2 mm. Entsprechend des Datenblatts des Leica HDS7000 wird für die erfassten TLS-Punkte eine Standardabweichung von 0,5 mm angenommen (Leica Geosystems 2011). Beide Epochen wurden von einem annähernd gleichen Standpunkt aufgenommen. Die TLS-Punktwolke von Epoche 0 besteht aus 150.074 Punkten und von Epoche 1 aus 128.775 Punkten.

Sandobjekt (3D-Punktwolke)

Das zweite real gemessene Testobjekt, das in Abbildung 6.7 dargestellt ist, wurde mittels Sand modelliert und weist Maße von 1,2 m x 0,7 m auf. Bei der Erstellung von diesem Testobjekt wurden homogene Formverläufe, ähnlich einer Hügellandschaft, nachgebildet.



Abb. 6.7: Testobjekt aus Sand.

Die Laserscanaufnahme wurde ebenso mit dem Leica HDS7000 Laserscanner und der Auflösungsstufe High durchgeführt. Die in Abbildung 6.8 dargestellten zwei 3D-Punktwolken des Sandobjekts enthalten einen durchschnittlichen Punktabstand von 2,3 mm. Die Punktzahl beträgt bei Epoche 0 135.600 Punkte und bei Epoche 1 101.253 Punkte.

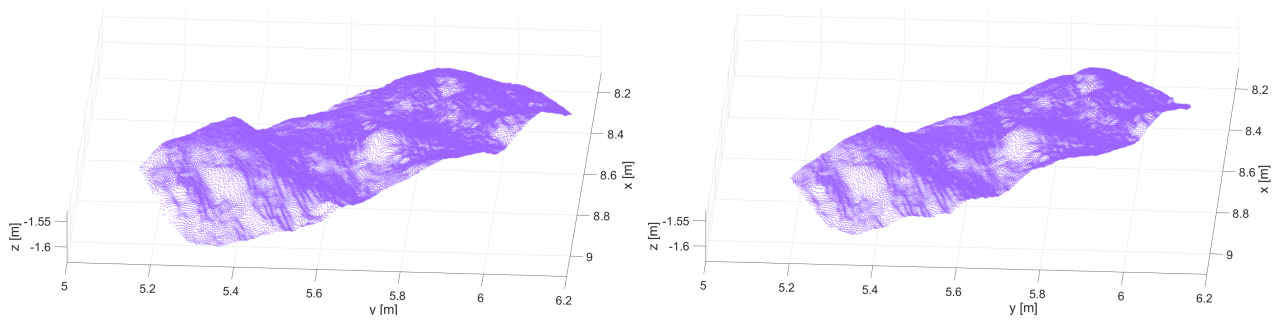


Abb. 6.8: 3D-Punktwolken des Sandobjekts von Epoche 0 (links) und Epoche 1 (rechts).

Abbildung 6.9 zeigt den Instrumentenaufbau des Leica HDS7000 im Messkeller der Universität Stuttgart sowie das Tontorso-Testobjekt in erhöhter Position auf einem Tisch zur linken Seite sowie das Sand-Testobjekt auf dem Boden zur rechten Seite.



Abb. 6.9: Instrumentenaufbau mit dem Tontorso-Testobjekt und Sand-Testobjekt.

7 Ergebnisanalyse

Die in dieser Arbeit vorgestellte DBKM wurde mittels simulierter Daten von 2D- und 3D-Punktwolken sowie eines Testobjekts in Form eines Gurkenblatts entwickelt. Zentrale Kernelemente der DBKM sind die Modellierung von TLS-Daten mittels Tensorprodukt B-Spline-Flächen und die Anwendung des Mapcut-Algorithmus als Registrierungsverfahren. Durch eine parallel geschaltete stochastische Modellierung ist eine Deformationsanalyse ebenfalls durchführbar. In den folgenden zwei Unterkapiteln soll die Funktionsfähigkeit und der Entwicklungsstand der DBKM an den vier in Kapitel 6 präsentierten Evaluierungsdatensätzen überprüft werden. Besonderes Interesse gilt hierbei der Fragestellung, inwieweit die DBKM und die anschließende Deformationsanalyse, die mittels simulierter Daten implementiert wurden, bereits auf Realdaten anwendbar sind und bei welchen Einzelkomponenten noch Entwicklungsbedarf besteht. In Unterkapitel 7.1 wird die DBKM an den vier Datensätzen ohne simulierte Deformationen angewendet und bewertet. Unterkapitel 7.2 fokussiert sich auf die Untersuchung der Ergebnisse bei Anwendung der Deformationsanalyse unter Nutzung der vier Datensätze mit simulierten Deformationen.

7.1 Anwendung der DBKM

7.1.1 Fassadenstück (2D-Punktwolke)

Für den 2D-Fall wird die DBKM anhand eines zweidimensionalen Testdatensatzes eines Fassadenstücks angewendet. Beide Epochen sind in geringem Maß verdreht und verschoben. Abbildung 7.1 zeigt das Kontrollpolygon von Epoche 0 und von Epoche 1.

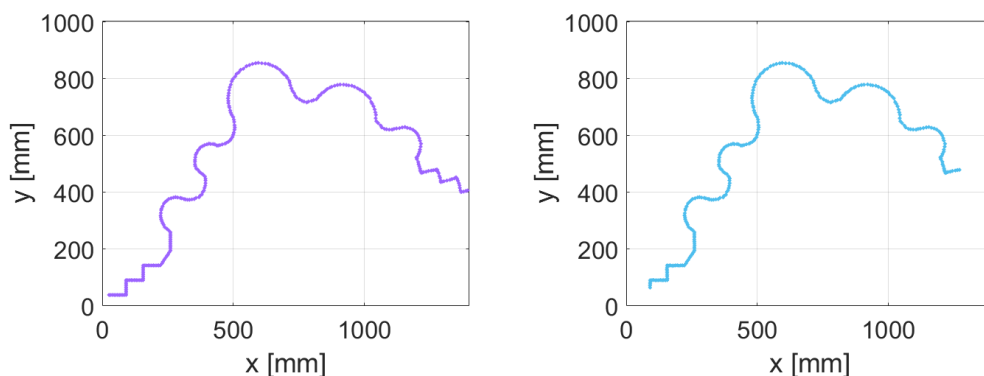


Abb. 7.1: 2D-Kontrollpolygone des Fassadenstücks von Epoche 0 (links) und Epoche 1 (rechts).

Die Größe des Drehwinkels und der Translationen wird derart gewählt, dass die Lage der Kontrollpolygone beider Epochen eine Messung des gleichen Objekts bei ähnlicher Messanordnung widerspiegeln. Im Detail existiert ein Drehwinkel von 0,2 gon und eine Translation in x-Richtung von 3,0 mm und in y-Richtung von 3,0 mm. Abbildung 7.2 verdeutlicht die Differenzen in der x- und y-Koordinate sowie die Euklidische Distanz der Kontrollpunkte. Hierbei kann sowohl das Messrauschen des Datensatzes mit der Standardabweichung von 0,5 mm als auch die geringe Translation und Drehung der beiden Epochen erkannt werden.

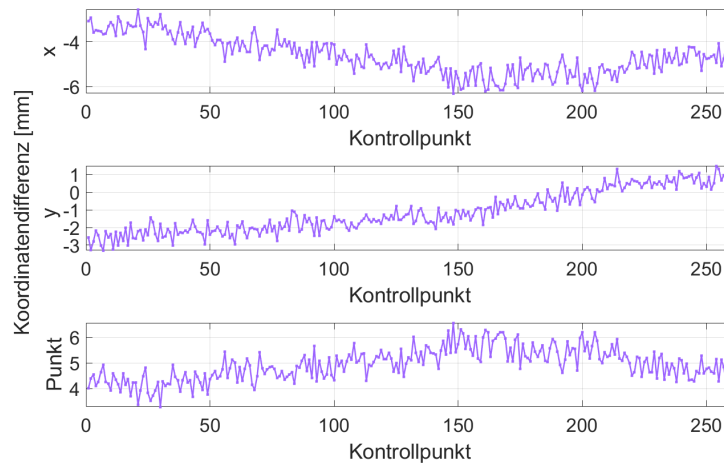


Abb. 7.2: Koordinatendifferenzen der Kontrollpunkte des Fassadenstücks von Epoche 0 und Epoche 1.

Die Darstellung der Koordinatendifferenzen zugehöriger Kontrollpunkte dient als Vergleichsmöglichkeit, um nach der Durchführung der DBKM eine Aussage über deren Einsatzqualität geben zu können. Die Differenzen zwischen den Kontrollpunkten werden sowohl durch das Messrauschen als auch die Drehung und Translation beeinflusst. Das Messrauschen spiegelt sich in Form eines Zickzackmusters, die Drehung durch eine zunehmende Vergrößerung der Differenzen und die Translationen durch eine grundsätzliche Differenz in x- und y-Koordinate von $> 3,0$ mm wider. Die maximale Koordinatendifferenz in x und y liegt etwa bei 6,3 mm und die maximale Euklidische Distanz der Kontrollpunkte bei etwa 6,6 mm. Aus den Kontrollpolygonen werden B-Spline-Kurven dritten Grades gebildet. Diese sind in Abbildung 7.3 präsentiert.

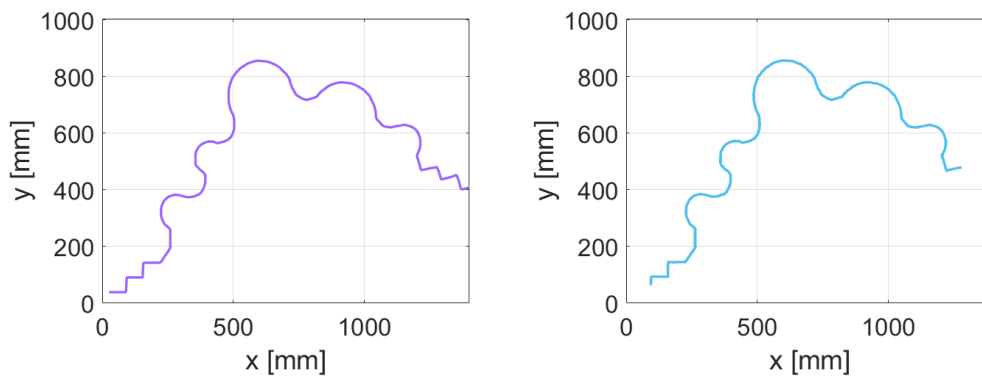


Abb. 7.3: B-Spline-Kurven des Fassadenstücks von Epoche 0 (links) und Epoche 1 (rechts).

Auf Basis der B-Spline-Kurven wird für jeden B-Spline-Kurvenpunkt ein Krümmungsparameter als Merkmalsdeskriptor berechnet. Daraufhin kann der Mapcut-Algorithmus mit Stufe I und II gestartet werden. Durch die 1D-Verschiebung der Krümmungswerte beider Epochen und die Bestimmung eines Korrelationskoeffizienten für jede betrachtete Verschiebungsposition mit einer Schrittweite, bei der jeder zehnte Krümmungswert betrachtet wird, kann die Verschiebungsposition mit der höchsten Korrelation erhalten werden. Die linke Grafik von Abbildung 7.4 zeigt die Korrelationskoeffizienten je nach Verschiebungsposition im Grobmapping. Hierbei ist ein 10%-tiger Überstand in beide Richtungen bereits inkludiert. Die im Grobmapping erhaltene Verschiebungsposition kann im Feinmapping durch Betrachtung jedes Krümmungswerts präzisiert werden. In der rechten Grafik von Abbildung 7.4 sind die Korrelationskoeffizienten für 120 Verschiebungspositionen vor und hinter der im Grobmapping

definierten Position abgebildet. Die insgesamt 240 Verschiebungspositionen im Feinmapping entsprechen im Grobmapping 24 Verschiebungspositionen.

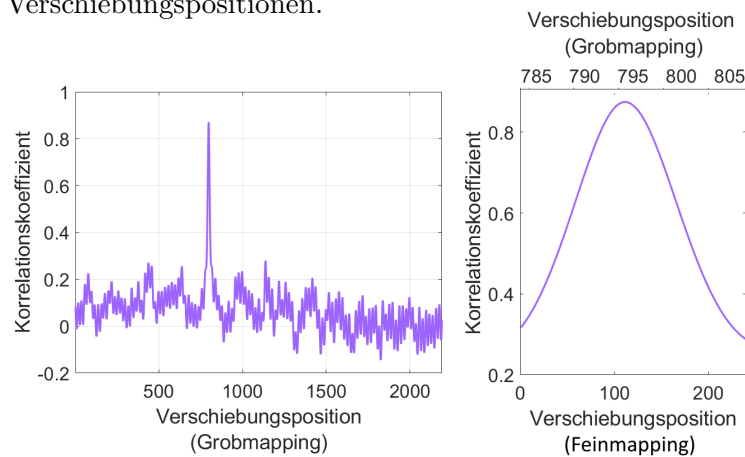


Abb. 7.4: links: Korrelationskoeffizienten im Grobmapping, rechts: Korrelationskoeffizienten im Feinmapping (2D-Fassadenstück, Iteration 1).

Während im Grobmapping der höchste Korrelationskoeffizient einen Wert von 0,871 hat, bringt die detaillierte Abfrage im Feinmapping einen höchsten Korrelationskoeffizienten von 0,874 hervor. Auf Basis der erlangten Verschiebungsposition kann eine Zuordnung zugehöriger B-Spline-Kurvenpunkte vorgenommen werden. Aus zehn Teilbereichen der B-Spline-Kurve werden Punktkorrespondenzen mit der geringsten Koordinatendifferenz ausgewählt. Von dieser Auswahl werden sechs als identische Punkte für die Transformation des Kontrollpolygons und der B-Spline-Kurvenpunkte von Epoche 1 nach Epoche 0 verwendet. Abbildung 7.5 zeigt in der linken Grafik in hellblau die zehn identischen Punkte für die zehn Kurvenbereiche und in rot die sechs selektierten identischen Punkte nach der Transformation.

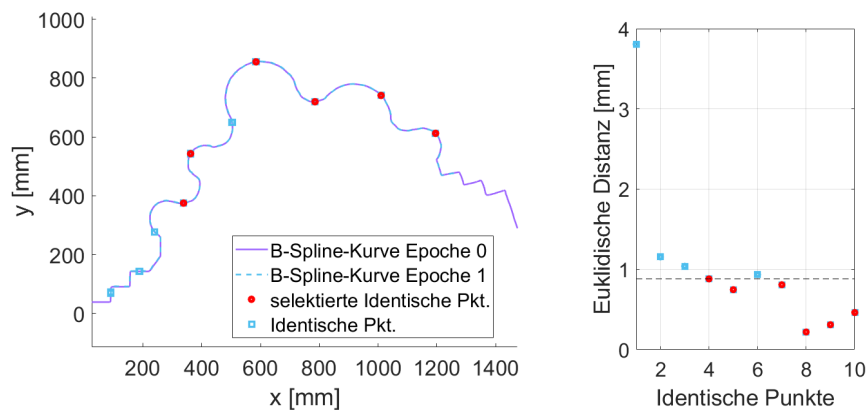


Abb. 7.5: links: Identische Punkte transformiert in Epoche 0, rechts: Euklidische Distanz der identische Punkte zwischen Epoche 0 und Epoche 1 (2D-Fassadenstück, Iteration 1).

Die rechte Grafik stellt die Euklidische Distanz der identischen Punkte nach der Transformation für die erste Iteration des Mapcut-Algorithmus dar. Eine Betrachtung der Koordinatendifferenzen zugehöriger B-Spline-Kurvenpunkte nach Durchführung der Koordinatentransformation (Stufe III Mapcut-Algorithmus) in Abbildung 7.6 zeigt, dass die Differenzen in Relation zu Abbildung 7.2 deutlich kleiner geworden sind. Die Koordinatendifferenzen in x und y liegen überwiegend bei maximal 1,5 mm. Auffällig ist eine starke Vergrößerung der Koordinatendifferenzen bei den x-Koordinaten auf 8,5 mm im Diagramm rechts sowie eine starke Zunahme der y-Koordinatendifferenzen auf 4,5 mm im Diagramm links. Dies zeigt, dass die Koordinatendifferenzen an den Anfangs- und Endkurvenpunkten bei Epoche 0 und Epoche 1 deutlich größer werden.

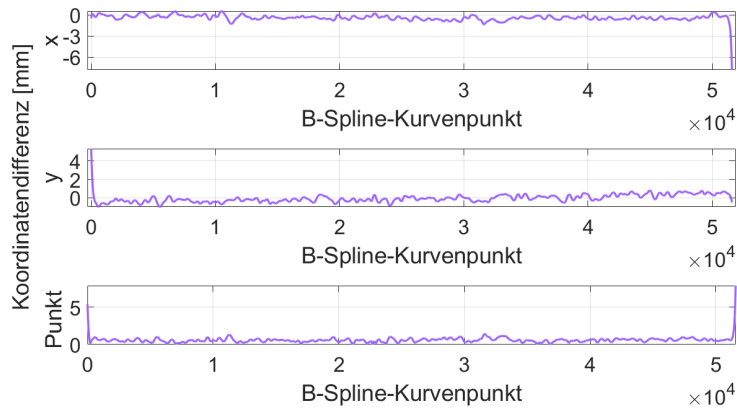


Abb. 7.6: Koordinatendifferenzen nach Stufe III des Mapcut-Algorithmus (2D-Fassadenstück, Iteration 1).

Erklärt werden kann dieses Verhalten mittels der Anfangs- und Endpunkt-Eigenschaft von B-Spline-Kurven (siehe Unterkapitel 2.2.3). Die Ursache liegt an den unterschiedlichen Anfangs- und Endpunkten der Kontrollpolygone von Epoche 0 und Epoche 1. Deshalb wird Stufe IV des Mapcut-Algorithmus angewendet, das sogenannte Cutting, bei dem das Kontrollpolygon von Epoche 0 im Bereich des Anfangs- und Endkontrollpunktes von Epoche 1 abgeschnitten wird. Diese Angleichung ermöglicht ähnliche Positionen der Anfangs- und Endkontrollpunkte beider Epochen.

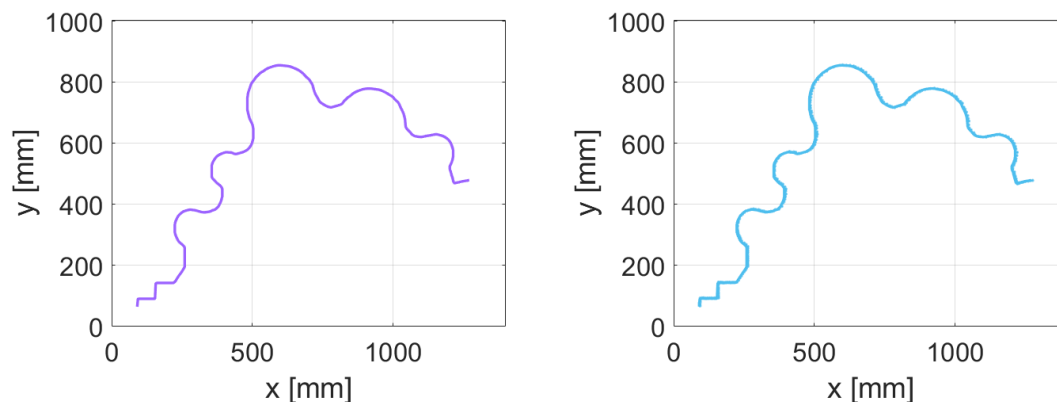


Abb. 7.7: B-Spline-Kurven von Epoche 0 (links) und Epoche 1 (rechts) nach Durchführung der ersten Iteration des Mapcut-Algorithmus (2D-Fassadenstück, Iteration 2).

Danach wird die zweite Iteration des Mapcut-Algorithmus durchgeführt, bei der zunächst aus dem gecutteten Kontrollpolygon von Epoche 0 und dem transformierten Kontrollpolygon von Epoche 1 B-Spline-Kurven und Krümmungsparameter berechnet werden. Die erzeugten B-Spline-Kurven sind in Abbildung 7.7 dargestellt. Im Vergleich zu den B-Spline-Kurven von Abbildung 7.3 sind die Anfangs- und Endkurvenpunkte beider Epochen nun ähnlich. Mittels der Bestimmung der Krümmungsparameter lässt sich daraufhin Stufe I und II des Mapcut-Algorithmus durchführen. Die beiden Diagramme der Korrelationskoeffizienten von Stufe I und II von Iteration 2 sind in Abbildung 7.8 angegeben. Im Grobmapping wird in Iteration 2 des Mapcut-Algorithmus ein maximaler Korrelationskoeffizient von 0,904 und im Feinmapping von 0,906 erreicht. In Relation zur ersten Iteration des Mapcut-Algorithmus hat sich der maximale Korrelationskoeffizient um 0,03 erhöht. Hierdurch kann eine Zuordnung zugehöriger B-Spline-Kurvenpunkte vorgenommen werden.

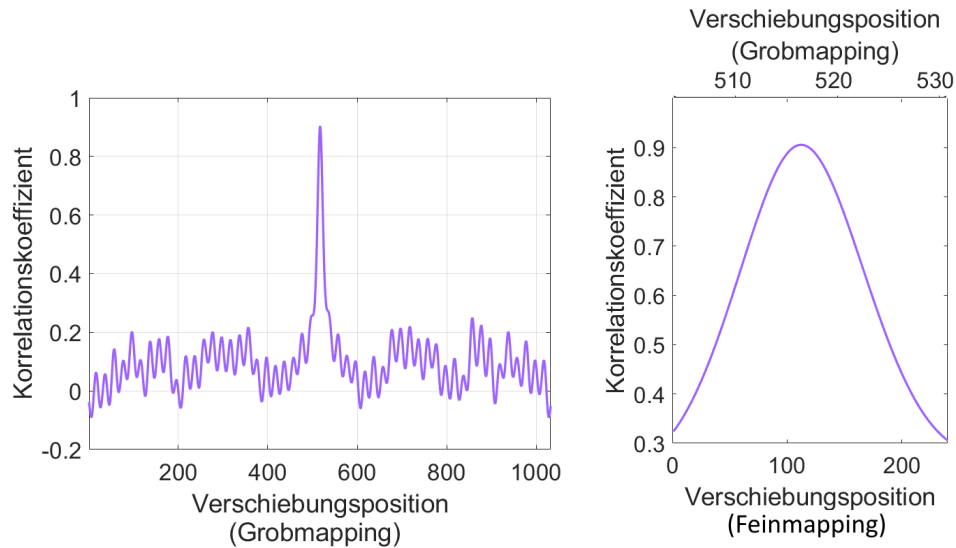


Abb. 7.8: links: Korrelationskoeffizienten im Grobmapping, rechts: Korrelationskoeffizienten im Feinmapping (2D-Fassadenstück, Iteration 2).

In 15 Teilbereichen der B-Spline-Kurve werden daraufhin zugehörige Kurvenpunktpaare mit der geringsten Koordinatendifferenz gesucht. Von diesen 15 Kurvenpunktpaaren werden wiederum neun mit der geringsten Koordinatendifferenz als identische Punkte für die Koordinatentransformation herangezogen. Die linke Grafik von Abbildung 7.9 zeigt die selektierten identischen Punkte in rot. Prinzipiell zeigt die linke Grafik die selektierten identischen Punkte der B-Spline-Kurven bereits im Koordinatensystem von Epoche 0. Die rechte Grafik von Abbildung 7.9 stellt die Euklidischen Distanzen der identischen Punkte ebenso nach Transformation in das Koordinatensystem von Epoche 0 dar. Die neun selektierten identischen Punkte haben eine maximale Euklidische Distanz von 0,5 mm. Dies ist im Vergleich zur ersten Iteration um etwa 0,4 mm geringer.

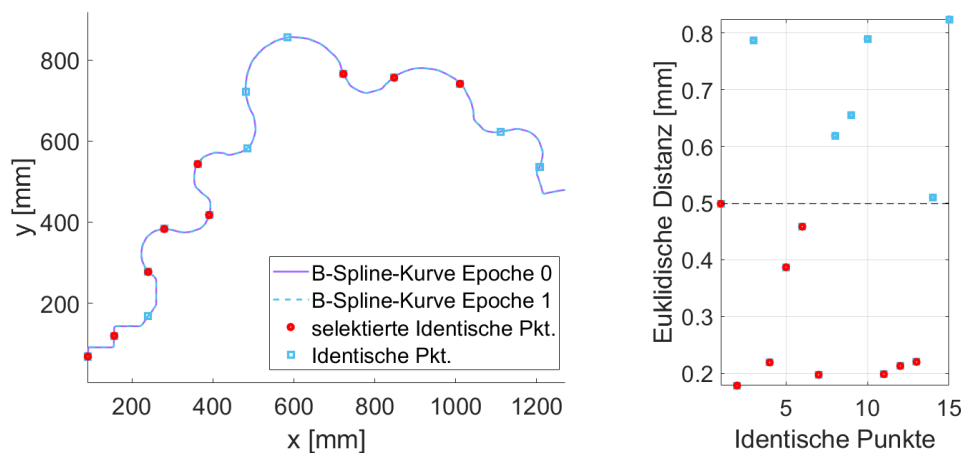


Abb. 7.9: links: Identische Punkte transformiert in Epoche 0, rechts: Euklidische Distanz der identische Punkte zwischen Epoche 0 und Epoche 1 (2D-Fassadenstück, Iteration 2).

Nach Durchführung der zweiten Iteration des Mapcut-Algorithmus zeigt eine Darstellung der Koordinatendifferenzen zugehöriger B-Spline-Kurvenpunkte in Abbildung 7.10, dass die Differenzen reduziert werden konnten. Während zu Beginn bei der x- und y-Koordinate bis zu 4 mm Differenzen auftreten,

liegen diese nach zweimaliger Durchführung des Mapcut-Algorithmus bei maximal 0,9 mm. Die maximale Euklidische Distanz der Kurvenpunkte hat sich von 5 mm auf 1,2 mm verbessert. Der Median der Kurvenpunktdifferenzen liegt bei 0,8 mm. Ebenso sind die im Bereich des Anfangs- und Endkurvenpunktes im ersten Iterationsdurchlauf bestehenden größer werdenden Differenzen (siehe Abbildung 7.6) behoben. Eine dritte Iteration würde lediglich eine Verringerung der Kurvenpunktdifferenz von 0,02 mm erzielen, weshalb der iterative Prozess in Anbetracht des Aufwand-Nutzen-Verhältnisses nach zwei Durchläufen für das 2D-Fassadenstück beendet wird.

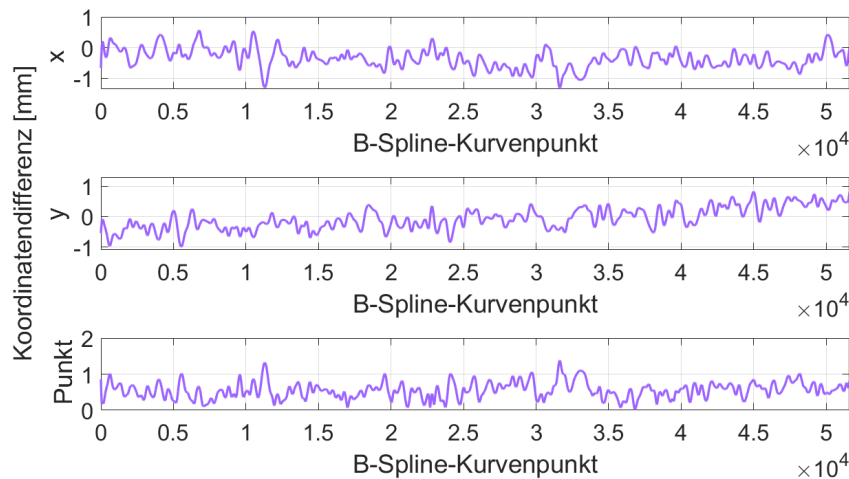


Abb. 7.10: Koordinatendifferenzen nach Stufe III des Mapcut-Algorithmus (2D-Fassadenstück, Iteration 2).

7.1.2 Gurkenblatt (3D-Punktwolke)

Die DBKM wurde mithilfe des Laserscandatensatzes des in Kapitel 6 beschriebenen Gurkenblatts entwickelt.

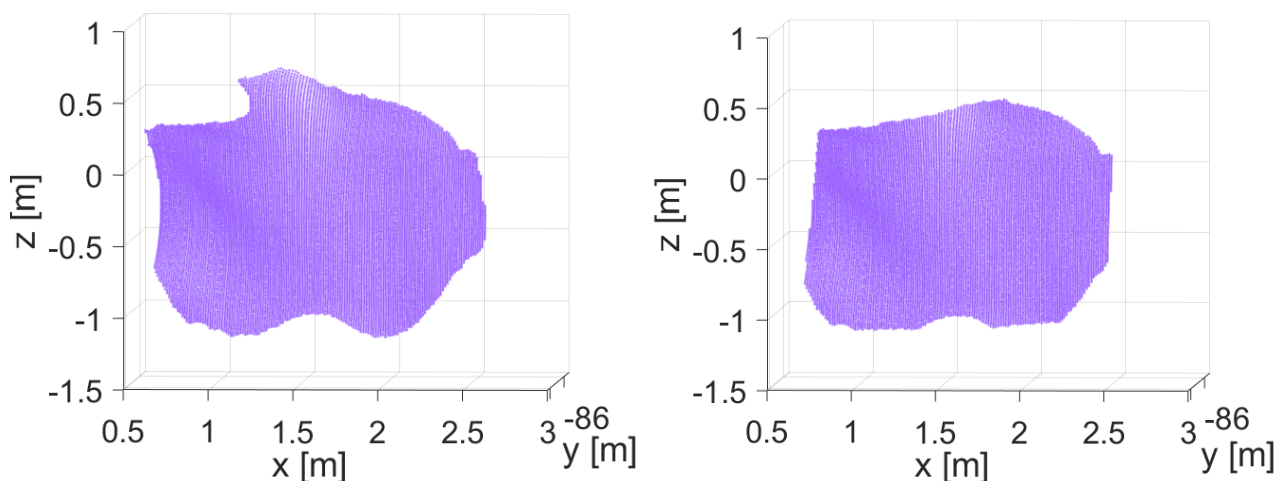


Abb. 7.11: 3D-Punktwolken des Gurkenblatts von Epoche 0 (links) und Epoche 1 (rechts).

Die einzelnen Ergebnisse der Prozessschritte beginnend bei der Definition der Kontrollpolygone bis hin zum Vergleich der Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte werden im

Folgenden dargestellt. Für die Durchführung der Methode werden die ursprünglichen Laserscandaten des Gurkenblatts als Epoche 0 herangezogen. Epoche 1 bildet einen Ausschnitt von Epoche 0 mit einer geringeren Punktzahl ab, der mit einem Gaußschen Rauschen von 0,5 mm versehen ist. Es handelt sich somit um simulierte Daten. Abbildung 7.11 zeigt die zwei verwendeten Punktwolken des Gurkenblatts von Epoche 0 und Epoche 1. Im Rahmen der Modellierung der TLS-Daten mittels Tensorprodukt B-Spline-Flächen wird der entwickelte Algorithmus zur Definition von Kontrollpolygonen angewendet. Beim Vergleich der zwei resultierenden Histogramme in Abbildung 7.12 kann festgestellt werden, dass die meisten gebildeten Kontrollpolygone mehr als 50 Punkte enthalten. Während aus der Punktwolke von Epoche 0 überwiegend Kontrollpolygone mit einer Punktzahl zwischen 50 und 135 entwickelt werden, entstehen aus Epoche 1 größtenteils Kontrollpolygone mit einer Punktzahl zwischen 50 und 115.

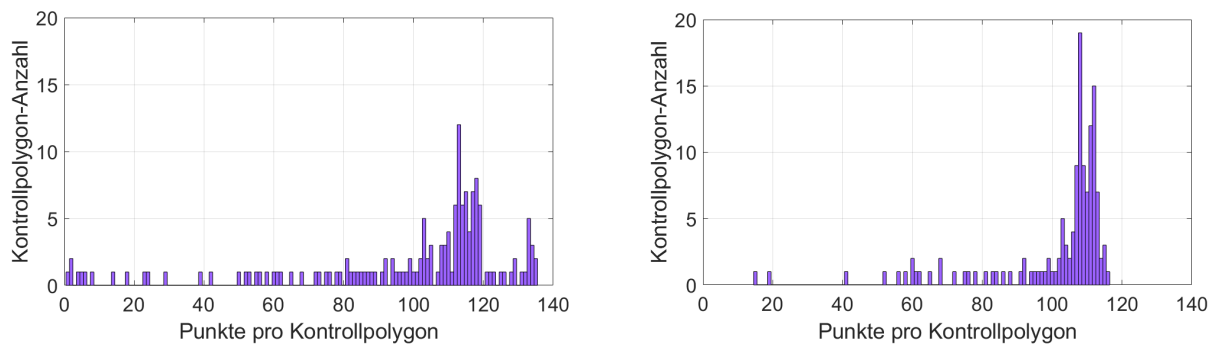


Abb. 7.12: Histogramme der definierten Kontrollpolygone des Gurkenblatts von Epoche 0 (links) und Epoche 1 (rechts).

Epoche 0 enthält dementsprechend mehr Kontrollpolygone mit einer höheren Anzahl an Kontrollpunkten als Epoche 1. Dies ist auf den größeren Ausschnitt der Punktwolke von Epoche 0 zurückzuführen. Die höhere Anzahl an Kontrollpolygonen gleicher Länge im Falle von Epoche 1 in Relation zu Epoche 0 ist wiederum mit der gleichmäßigeren rechteckförmigeren Form der Punktwolke von Epoche 1 zu erklären.

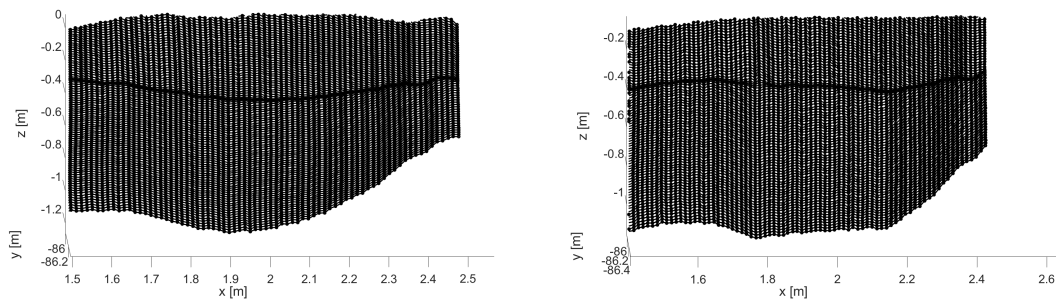


Abb. 7.13: Ausschnitt der Kontrollgitter von Gurkenblatt Epoche 0 (links) und Epoche 1 (rechts).

Mittels der definierten Kontrollpolygone kann der Algorithmus von Böhm angewendet werden. Auf diese Weise werden die Kontrollpolygone mit neuen Kontrollpunkten derart aufgefüllt, dass eine rechteckförmige Kontrollgitterstruktur entsteht. Trotz des Einführens neuer Kontrollpunkte im mittleren Bereich des Kontrollgitters wird die Form der zukünftigen B-Spline-Fläche nicht verändert. Ein Ausschnitt der erzeugten Kontrollgitter ist in den Grafiken von Abbildung 7.13 dargestellt. In diesen kann die zuvor genannte Verdichtung der Kontrollpunkte im mittleren Bereich nachvollzogen werden.

Ein vergrößerter Ausschnitt dieser Kontrollgitter ist in den Grafiken von Abbildung 7.14 einsehbar.

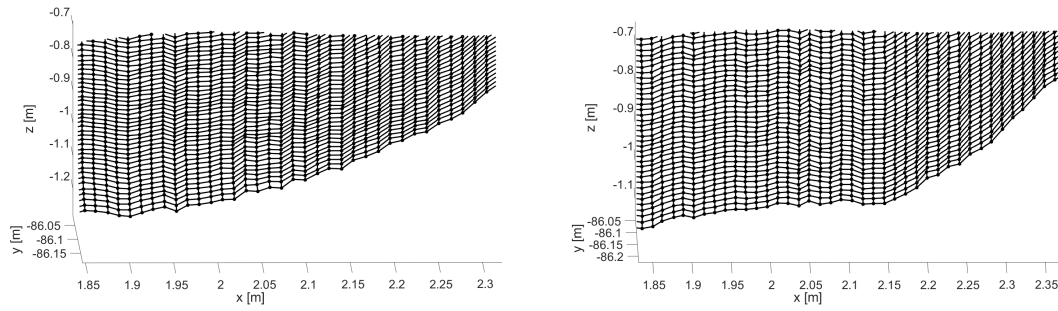


Abb. 7.14: Ausschnitt der Kontrollgitter von Gurkenblatt Epoche 0 (links) und Epoche 1 (rechts).

Auf dieser Basis kann eine Tensorprodukt B-Spline-Fläche für jede Epoche sowie Krümmungsparameter generiert werden. Die B-Spline-Fläche wird, wie in Abbildung 2.28 von Abschnitt 2.3 dargestellt, zeilenweise erzeugt. Für eine bessere Darstellung der Flächen werden die B-Spline-Flächen als Rasterdarstellung in Abbildung 7.15 präsentiert.

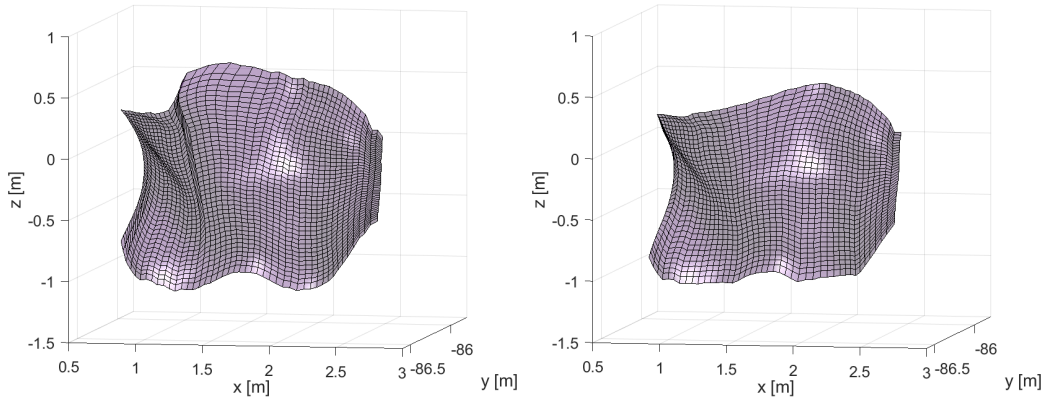


Abb. 7.15: B-Spline-Fläche von Gurkenblatt Epoche 0 (links) und Epoche 1 (rechts) (Rasterdarstellung).

Für die Generierung der Rasterdarstellung wird in diesem Fall jeder 30.-te B-Spline-Flächenpunkt verwendet, um eine bessere Darstellung der durch die B-Spline-Fläche modellierten Objektoberfläche und -form zu erhalten.

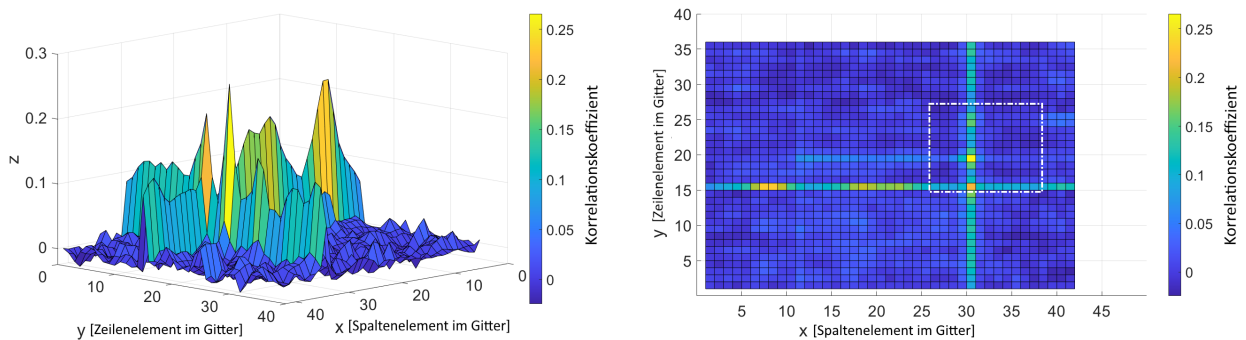


Abb. 7.16: Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 1).

Diese Diskretisierung dient ausschließlich zu Darstellungszwecken und wird nicht für die weiteren Berechnungen genutzt. Nach der Erzeugung einer B-Spline-Fläche und Krümmungsparametern für jeden B-Spline-Flächenpunkt wird der Mapcut-Algorithmus gestartet. Das Grobmapping (Stufe I), dessen Korrelationskoeffizienten-Matrix in Abbildung 7.16 als 3D- und 2D-Ansicht dargestellt ist, liefert eine grobe Eingrenzung für die Zuordnung der Punktkorrespondenzen. Sowohl in der 3D- als auch 2D-Ansicht kann abgelesen werden, dass der Mapping-Vorgang mit der Korrelationsanalyse keine hohen eindeutigen Werte hervorbringt, sondern maximal bei 0,25 liegt. Die für Stufe II des Mapcut-Algorithmus zu betrachtenden Verschiebungsbereiche, sind in der rechten Grafik von Abbildung 7.16 mit einem weißen Rechteck markiert. Eine präzisere Abfrage der Korrelationskoeffizienten im Feinmapping (Stufe II), die in Abbildung 7.17 dargestellt ist, bringt schließlich die Möglichkeit hervor erste Punktkorrespondenzen festzulegen. Der Vergleich der maximalen Werte der Korrelationskoeffizienten im Grobmapping in Relation zum Feinmapping zeigt, dass im Grobmapping ein höherer Korrelationskoeffizient erreicht wird.

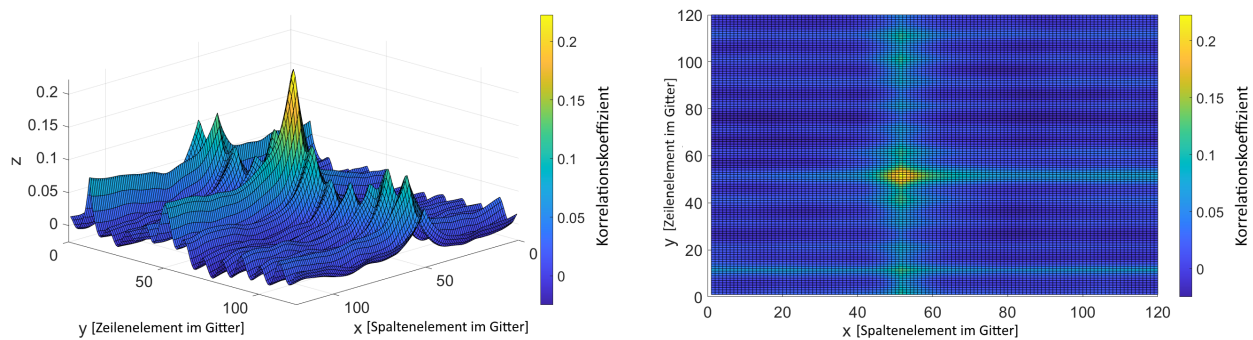


Abb. 7.17: Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 1).

Diese Verhaltensweise kann damit erklärt werden, dass im Grobmapping lediglich jeder zehnte Krümmungswert betrachtet wird und so eine Art Glättungseffekt der Daten stattfindet. Durch die Reduktion der Abtastrate der Krümmungswerte wird das Rauschen abgeschwächt, während trendartige Komponenten dominieren. Der Vergleich der geglätteten Krümmungswerte beider Epochen weist deshalb eine stärkere Übereinstimmung auf, was sich in einem höheren Korrelationskoeffizienten widerspiegelt.

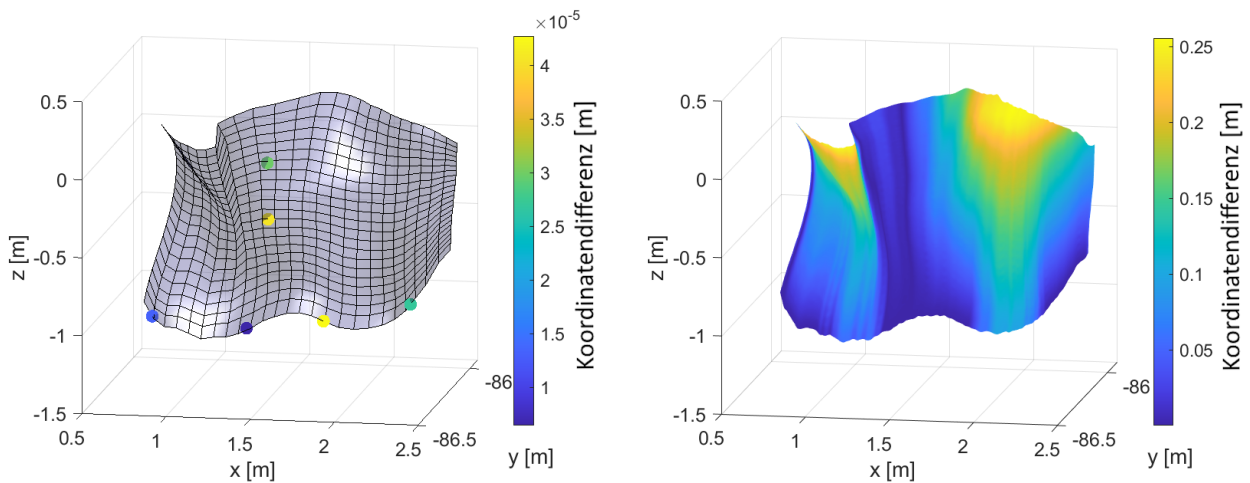


Abb. 7.18: links: Vorläufige annähernd identische Punkte, rechts: Koordinatendifferenzen von Epoche 0 und Epoche 1 (Gurkenblatt, Iteration 1).

Der erhöhte Korrelationskoeffizient spiegelt allerdings keine tatsächliche verbesserte Übereinstimmung wider, da deutlich weniger Werte betrachtet werden als im Feinmapping. Die Hinzunahme jedes Krümmungswerts für die Betrachtung der Verschiebungsposition im Feinmapping führt daher oftmals zu einem geringeren maximalen Korrelationskoeffizienten als im Grobmapping, jedoch hat der Korrelationskoeffizient im Feinmapping eine präzisere verlässlichere Aussagekraft, da eine größere Datenmenge betrachtet wird. In Stufe III des Mapcut-Algorithmus werden durch Einteilung der B-Spline-Fläche in 16 Kacheln und der Selektion von sechs Punktkorrespondenzen mit der geringsten Euklidischen Distanz annähernd identische Punkte ausfindig gemacht (linke Grafik von Abbildung 7.18). Mit deren Unterstützung kann eine Koordinatentransformation der B-Spline-Fläche sowie der Kontrollpunkte von Epoche 1 in das Koordinatensystem von Epoche 0 ermöglicht werden. Wie in der rechten Grafik von Abbildung 7.18 zu erkennen ist, bestehen teilweise große Koordinatendifferenzen zwischen korrespondierend zugeordneten B-Spline-Flächenpunkten der zwei Epochen. Ursache für die deutlichen Koordinatendifferenzen zwischen korrespondierenden B-Spline-Flächenpunkten sind die unterschiedlichen Randabgrenzungen der beiden Epochen.

Die übereinandergelagerten Epochen sowie die eindeutigen Unterschiede im Randbereich der zwei verwendeten Epochen des Gurkenblatts werden in Abbildung 7.19 verdeutlicht. Das Cutting, Stufe IV des Mapcut-Algorithmus, soll die beschriebenen Auswirkungen der unterschiedlichen Randabgrenzungen verringern bzw. beheben. Die Randkontrollpunkte beider Epochen werden beim Cutting angeglichen und versetzt in die Lage durch nachfolgende Anwendung des Algorithmus von Böhm und der Bildung von B-Spline-Flächen und deren Krümmungsparametern den Mapcut-Algorithmus erneut zu durchlaufen. Der Vorgang wird solange wiederholt, bis sich die Koordinatendifferenzen zwischen zwei Durchläufen nicht mehr deutlich verringern. Bei den 3D-Datensätzen handelt es sich um eine Verringerung von 0,02 mm, bei der in Betrachtung des Rechenaufwand-Nutzen-Verhältnisses keine weitere Iteration durchgeführt wird.

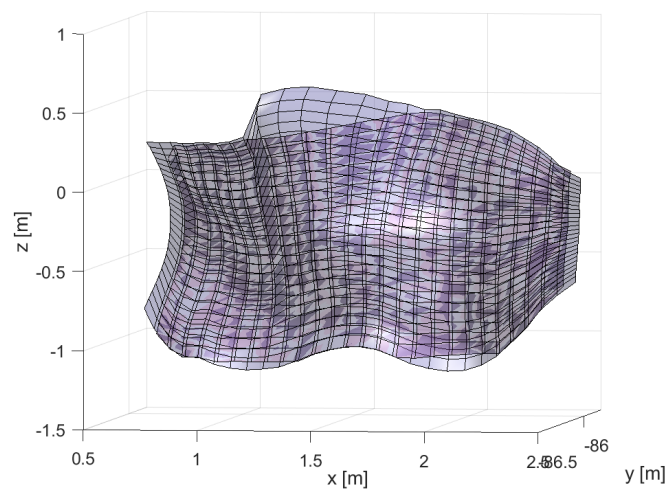


Abb. 7.19: Übereinandergelagerte B-Spline-Flächen von Epoche 0 und Epoche 1 (Gurkenblatt, Iteration 1).

Überwiegend kann die Randabgrenzung bereits nach ein bis zwei Iterationen verbessert werden, so dass meistens maximal drei Iterationen des Mapcut-Algorithmus notwendig sind. Im Falle des Gurkenblatts genügen zwei Iterationsdurchläufe, um eine ausreichende Optimierung zu erreichen. Die Grafiken von Abbildung 7.20 zeigen das Ergebnis des Grobmappings in der zweiten Iteration. Das weiße Rechteck in der rechten Grafik von Abbildung 7.20 verdeutlicht den Verschiebungsbereich, der im Feinmapping, statt mit einer 10-er Schrittweite, für jeden Krümmungswert in der Korrelationsanalyse betrachtet wird.

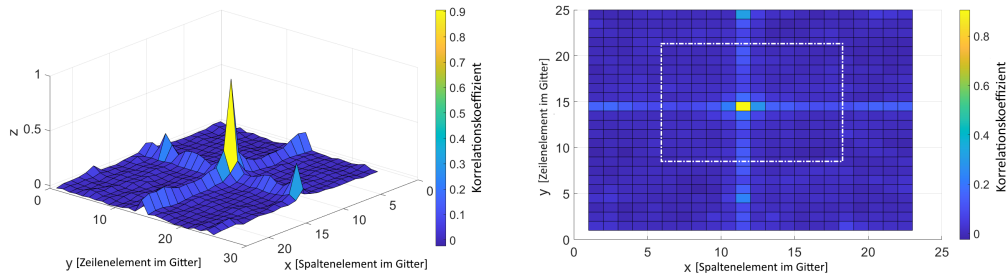


Abb. 7.20: Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 2).

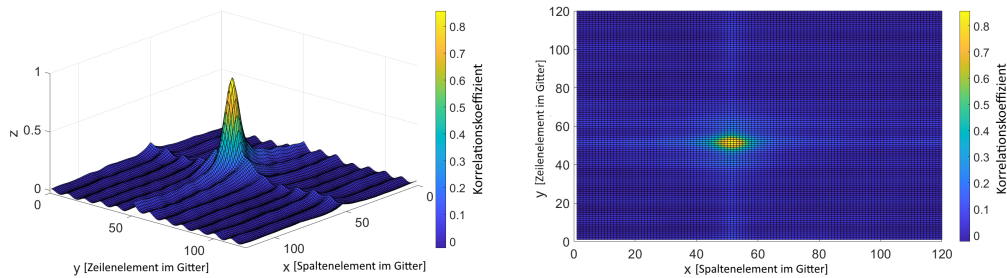


Abb. 7.21: Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Gurkenblatt, Iteration 2).

Im Vergleich zur ersten Iteration von Abbildung 7.16 kann bei den Werten der Korrelationsmatrix ein ausgeprägter und eindeutiger Peak wahrgenommen werden, der eine klare Zuordnung ermöglicht. In Relation zur ersten Iteration, in der der maximale Korrelationswert 0,25 betrug, zeigt die zweite Iteration einen maximalen Korrelationswert von 0,9. Auch die Ergebnisse des Feinmappings, die in Abbildung 7.21 dargestellt werden, fallen deutlich besser aus als in der ersten Iteration (Abbildung 7.17). Die Selektion von annähernd identischen Punkten im zweiten Iterationsschritt, die in Abbildung 7.22 in der linken Grafik präsentiert sind, gestattet eine Transformation der Kontrollpunkte und der B-Spline-Flächenpunkte von Epoche 1 in das Koordinatensystem von Epoche 0. Bei der Auswahl der annähernd identischen Punkte wird die B-Spline-Fläche ab der zweiten Iteration in 25 Kacheln unterteilt und hiervon 16 Punktkorrespondenzen mit der geringsten Euklidischen Distanz für die Transformation genutzt.

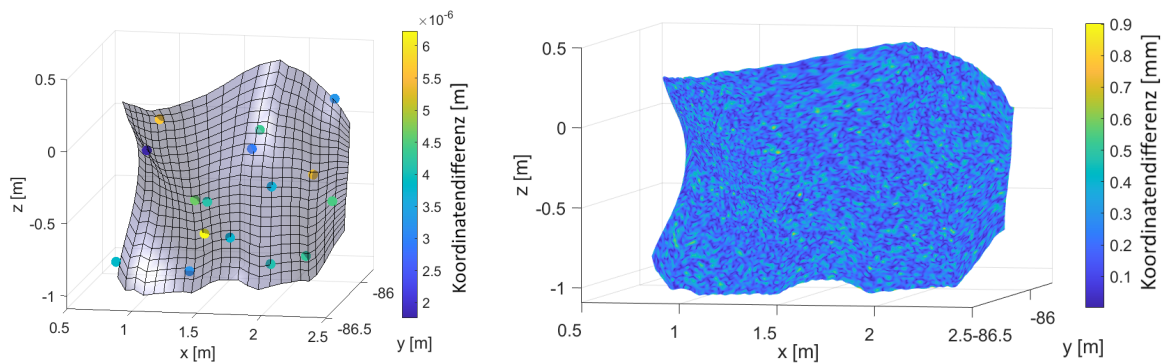


Abb. 7.22: links: Identische Punkte, rechts: Koordinatendifferenzen von Epoche 0 und Epoche 1 (Gurkenblatt, Iteration 2).

Hierdurch können die Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte unmittelbar durch Berechnung der Euklidischen Distanz verglichen werden. Die rechte Grafik von Abbildung 7.22 stellt die Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte in der zweiten Iteration dar. In Relation zur ersten Iteration (Abbildung 7.18, rechte Grafik) kann eine deutliche Verringerung der Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte festgestellt werden. Während bei der ersten Iteration maximale Differenzen von 25 cm auftreten, beträgt das Maximum in der zweiten Iteration lediglich 0,9 mm und der Median beträgt 0,2 mm.

7.1.3 Tontorso (3D-Punktwolke)

Bei dem zweiten 3D-Datensatz, bei dem die Funktionalität der DBKM überprüft werden soll, handelt es sich um die Laserscandaten des Tontorsos, der in zwei Epochen real aufgemessen wurde. Die TLS-Punktwolken sind in den Grafiken in Abbildung 7.23 dargestellt und werden detaillierter in Kapitel 6 beschrieben.

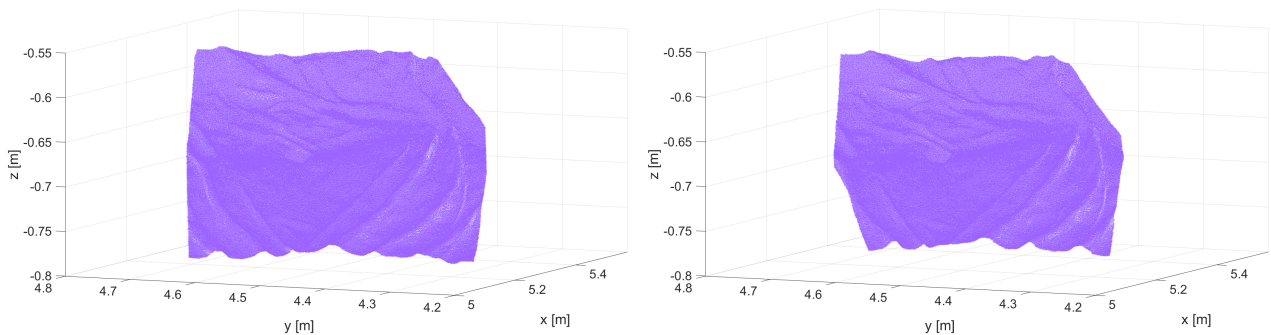


Abb. 7.23: 3D-Punktwolken von Tontorso von Epoche 0 (links) und Epoche 1 (rechts).

Die real gemessenen TLS-Daten des Tontorsos enthalten in Relation zu den simulierten Daten des Gurkenblatts, auf deren Basis die DBKM entwickelt wurde, mit ca. 150.000 Punkten die 10-fache Menge an Punkten und sind demnach deutlich umfangreicher. Der Vorgang der Definition von Kontrollpolygonen bringt die zwei Histogramme von Abbildung 7.24 hervor. Bei der Punktwolke von Epoche 0 werden überwiegend Kontrollpolygone mit einer Punktanzahl zwischen 330 und 345 Kontrollpunkten pro Kontrollpolygon definiert. Dies betrifft in diesem Bereich 329 Polygone.

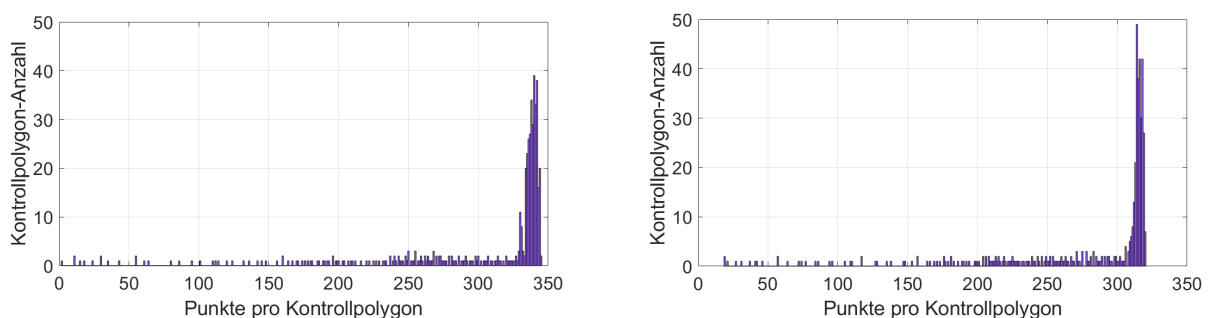


Abb. 7.24: Histogramme der definierten Kontrollpolygone des Tontorsos von Epoche 0 und Epoche 1.

Vor dieser Anhäufung an Kontrollpolygonen werden in einer Spanne zwischen 160 bis 330 Punkten pro Polygon in der Summe etwa 190 Polygone definiert. Die kleinere Punktwolke aus

Epoche 1 enthält nach der Definition der Kontrollpolygone hingegen überwiegend Kontrollpolygone mit 310 bis 320 Punkten. In diesem Bereich sind dies insgesamt 288 Polygone. Vor dieser Anhäufung werden in einem Bereich zwischen 160 und 309 Punkten pro Polygon 145 Polygone gebildet. Aus den beiden Histogrammen kann abgelesen werden, dass der größere Ausschnitt der Punktwolke von Epoche 0 in Polygonen größerer Punktzahl mündet, so dass bei Epoche 1 mehr Polygone mit geringerer Punktzahl als in Epoche 0 vorhanden sind. Die rechteckförmige Ausprägung und der gleichmäßige Verlauf der Punktwolken beider Epochen spiegelt sich in Histogrammen wider, die eine Anhäufung von Kontrollpolygonen mit ähnlicher Punktzahl aufweisen. Auf dieser Basis ist eine Anwendung des Algorithmus von Böhm durchführbar. Das Kontrollpolygon mit der jeweils größten Anzahl an Kontrollpunkten gibt die Anzahl an Kontrollpunkten vor, die hinzugefügt werden muss, um eine bidirektionale Kontrollgitterstruktur zu erlangen. Die Grafiken in Abbildung 7.25 zeigen beispielhaft einen Ausschnitt der erzeugten Kontrollgitter von Epoche 0 und Epoche 1. Hierbei kann festgestellt werden, dass die Kontrollpunkte ungleichmäßig verteilt sind.

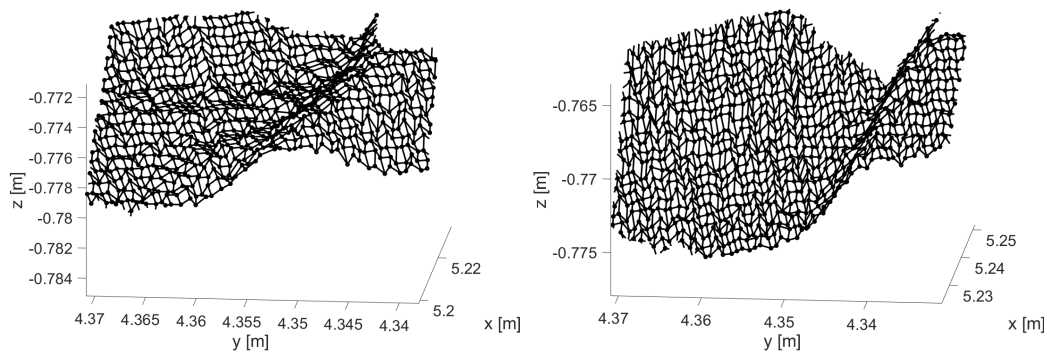


Abb. 7.25: Ausschnitt der Kontrollgitter des Tontorso von Epoche 0 (links) und Epoche 1 (rechts).

Trotz der in Relation zu den simulierten Daten des Gurkenblatts unregelmäßiger verlaufenden Realdaten des Tontorsos, ist eine bidirektionale Kontrollgitterstruktur konstruierbar, so dass Tensorprodukt B-Spline-Flächen gebildet werden können. Die erzeugten B-Spline-Flächen werden als Rasterdarstellung in den Grafiken von Abbildung 7.26 gezeigt.

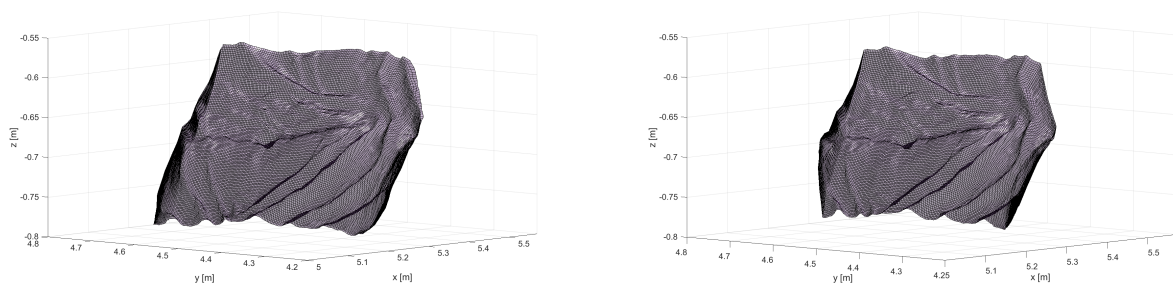


Abb. 7.26: B-Spline-Flächen des Tontorsos von Epoche 0 (links) und Epoche 1 (rechts) (Rasterdarstellung).

Mittels der aus den B-Spline-Flächenpunkten konstruierten Krümmungsparametern kann im Mapcut-Algorithmus in Stufe I ein Grobmapping erfolgen. Die Korrelationen der einzelnen Überlagerungspositionen werden in einer Korrelationsmatrix abgespeichert, die in Abbildung 7.27 für die erste Iteration in einer 3D- und einer 2D-Ansicht dargestellt ist. Ungeachtet der Realdaten performt das Grobmapping analog zur Anwendung mit simulierten Daten. Auffällig bei Betrachtung der 3D-Ansicht

des Grobmappings ist die linienhafte Struktur hoher Korrelationskoeffizientenwerte. Dies könnte damit zusammenhängen, dass der Tontorso einige scharf verlaufende Kanten besitzt, die bei horizontaler Verschiebung der Krümmungsmatrizen beider Epochen im ersten Iterationsprozess besser korrelieren als bei vertikaler Verschiebung.

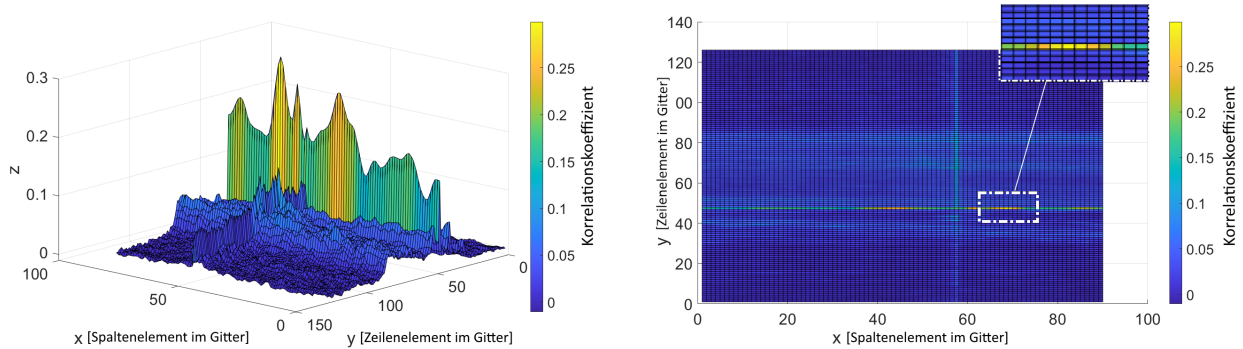


Abb. 7.27: Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 1).

Anhand der höchsten Korrelation von etwa 0,3 kann im Feinmapping eine Präzisierung für die Zuordnung zugehöriger B-Spline-Flächenpunkte erfolgen. Das Ergebnis des Feinmappings wird in Abbildung 7.28 präsentiert.

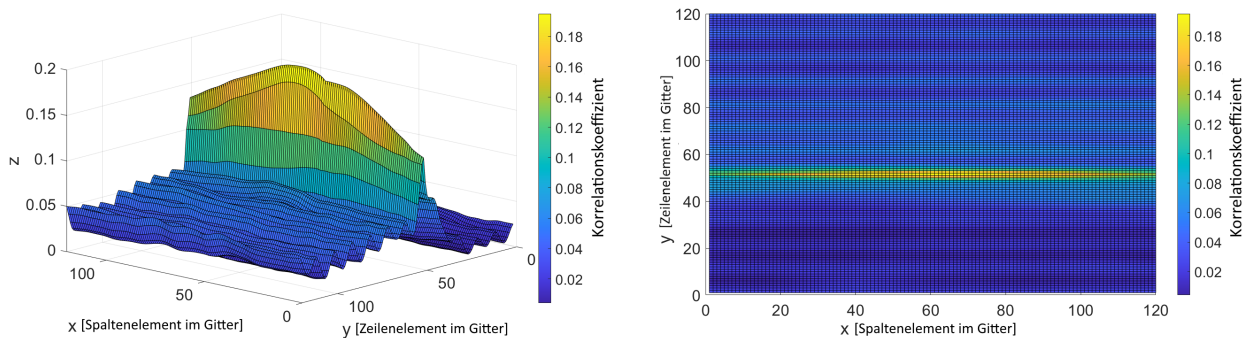


Abb. 7.28: Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 1).

In der dritten Stufe des Mapcut-Algorithmus werden vorläufige annähernd identische Punkte selektiert mit denen die B-Spline-Flächenpunkte und Kontrollpunkte von Epoche 1 in das Koordinatensystem von Epoche 0 transformiert werden können. Die ausgewählten identischen Punkte für die erste Iteration werden in der linken Grafik von Abbildung 7.29 gezeigt. Die Koordinatendifferenzen zwischen den selektierten identischen Punkten nach der Koordinatentransformation liegen zwischen 2 mm und 6,5 mm. Eine Koordinatentransformation der B-Spline-Fläche und der Kontrollpunkte von Epoche 1 in Epoche 0 ermöglicht den direkten Vergleich korrespondierender B-Spline-Flächenpunkte. In der rechten Grafik von Abbildung 7.29 können die Koordinatendifferenzen betrachtet werden. Überwiegend liegen Differenzen von 2 cm vor, allerdings treten in der ersten Iteration, vor allem am Rand sowie im mittleren Bereichen größere Koordinatendifferenzen von bis zu 18 cm auf. Die Darstellung der sehr groß ausfallenden Koordinatendifferenzen zwischen als korrespondierend ausgewiesenen B-Spline-Flächenpunkten der zwei Epochen zeigt erneut die Notwendigkeit der Anwendung des Cuttings des Mapcut-Algorithmus, um auch mit TLS-Daten zweier Epochen arbeiten zu können, deren Ränder nicht durch intensive Vorverarbeitung beinahe identisch sind.

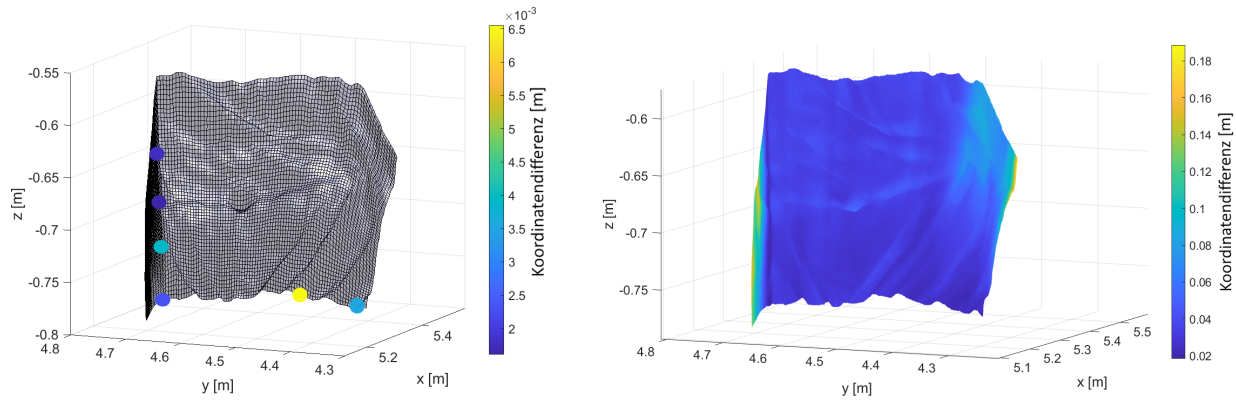


Abb. 7.29: links: Vorläufige annähernd identische Punkte, rechts: Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Tontorso, Iteration 1).

Abbildung 7.30 zeigt die nach der ersten Iteration übereinanderliegenden B-Spline-Flächen in Rasterdarstellung von Epoche 0 und Epoche 1.

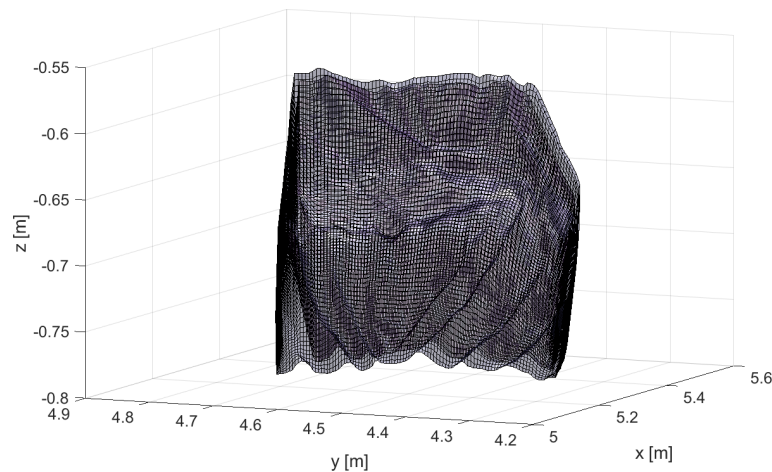


Abb. 7.30: Übereinandergelagerte B-Spline-Flächen von Epoche 0 und Epoche 1 (Tontorso, Iteration 1).

Nach der Koordinatentransformation wird das Cutting des Mapcut-Algorithmus durchgeführt. Für die TLS-Daten des Tontorsos sind drei Iterationsdurchläufe der B-Spline-Flächen Modellierung und des Mapcut-Algorithmus notwendig, um eine Verbesserung der Zuordnung der Punktkorrespondenzen zu erreichen. In der dritten Iteration des Grobmappings zeigt sich in Abbildung 7.31 in Relation zur ersten Iteration eine klare und eindeutige Herausbildung eines maximalen Werts der Korrelationsmatrix.

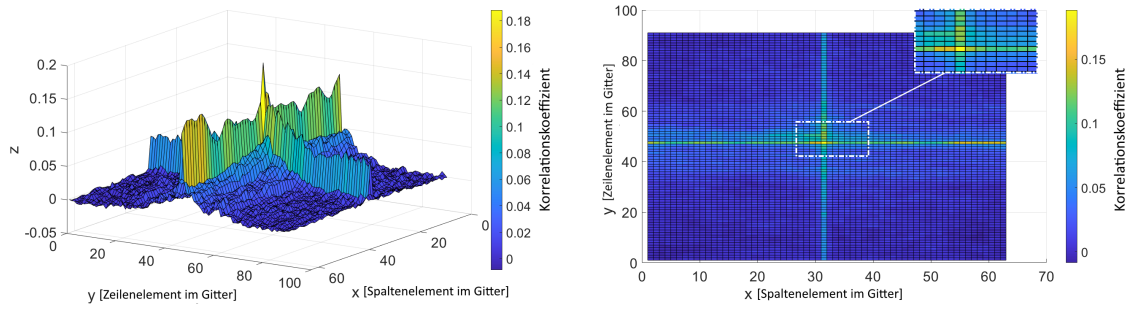


Abb. 7.31: Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 3).

Die verbesserte Zuordnungsmöglichkeit zeigt sich auch im Feinmapping in den Grafiken in Abbildung 7.32, bei dem der maximale Korrelationswert im Vergleich zum linienhaften Verlauf des Feinmappings in Iteration 1 eindeutiger herausgebildet wird.

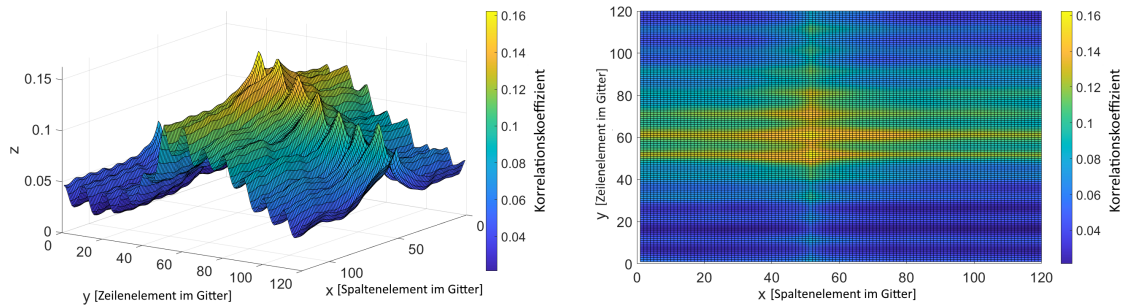


Abb. 7.32: Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Tontorso, Iteration 3).

Nach dem Feinmapping werden die annähernd identischen Punkte für den dritten Iterationsdurchlauf selektiert. Die 16 ausgewählten Punktkorrespondenzen mit der geringsten Euklidischen Distanz werden in Abbildung 7.33 dargestellt. Die Koordinatendifferenzen nach einer Transformation in Epoche 0 haben sich mit 1 mm bis 3 mm im Vergleich zur ersten Iteration halbiert.

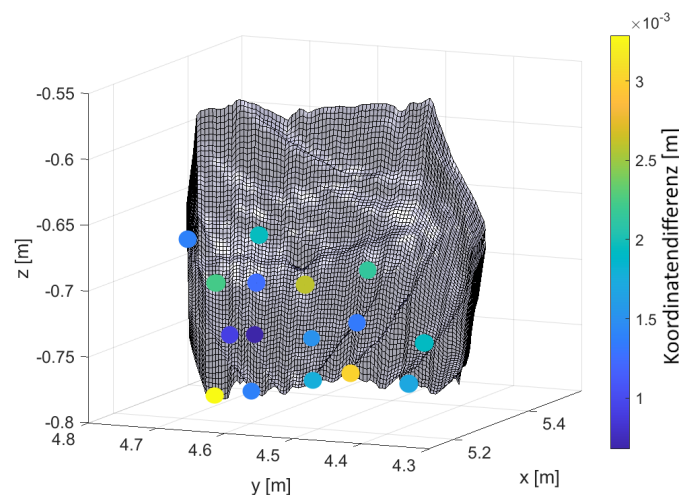


Abb. 7.33: Identische Punkte (Tontorso, Iteration 3).

Durch die Koordinatentransformation der gesamten B-Spline-Fläche und der Kontrollpunkte von Epoche 1 nach Epoche 0 können die korrespondierenden B-Spline-Flächenpunkte verglichen werden.

Die Euklidischen Distanzen der Punktkorrespondenzen sind in Abbildung 7.34 dargestellt und fallen im Vergleich zum ersten Iterationsdurchlauf wesentlich geringer aus. Statt 2 cm bis 18 cm, sind nun Differenzen von 2 mm bis 20 mm vorhanden. Betrachtet man die Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte im Detail, kann festgestellt werden, dass besonders an Stellen von Kanten, Erhebungen oder Vertiefungen größere Differenzen bis zu 2 cm auftreten. Außerhalb der Bereiche mit stärkeren Erhebungen oder Vertiefungen sind überwiegend Koordinatendifferenzen von etwa 2 mm bis 4 mm abzulesen. Im Vergleich zu den Ergebnissen der simulierten Daten des Gurkenblatts, kann bei den Realdaten des Tontorsos durch die iterative Anwendung der DBKM zwar eine Verbesserung der Größenordnung von maximal 18 cm auf 2 cm und außerhalb der Erhebungen auf 0,2 cm erzielt werden, allerdings können die Koordinatendifferenzen der Punktkorrespondenzen bisher nicht soweit verringert werden, dass eine Größenordnung des Messrauschens erreicht wird.

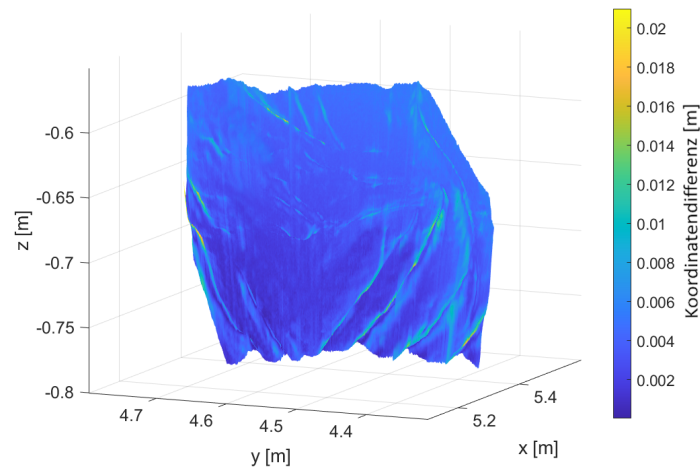


Abb. 7.34: Koordinatendifferenzen von Epoche 0 und Epoche 1 (Tontorso, Iteration 3).

Auch bei Versuchen mit mehr Iterationsdurchläufen ergibt sich keine Änderung. Die Ursache für diese Verhaltensweise wird nach der Vorstellung der Ergebnisse der Anwendung der DBKM des Sandobjekts in Abschnitt 7.1.5 intensiv diskutiert. Im Allgemeinen kann festgestellt werden, dass bei den Realdaten des Tontorsos mit ca. 150.000 Punkten und einem mittleren Punktabstand von 1,2 mm alle entwickelten Einzelkomponenten erfolgreich eingesetzt werden können. Zu diesen Einzelkomponenten zählen die Definition von Kontrollpolygonen und die Durchführung des Algorithmus von Böhm, wodurch eine Modellierung der TLS-Daten mittels Tensorprodukt B-Spline-Flächen abgeschlossen werden kann. Zum anderen können die Funktionsfähigkeit der vier Stufen des Mapcut-Algorithmus, das Grob- und Feinmapping, die Transformation und das Cutting, bestätigt werden. Die Übertragung der auf simulierten Daten basierenden entwickelten DBKM auf reale TLS-Daten ist somit in erster Linie erfolgreich. Die Differenz der Punktkorrespondenzen sollte allerdings zukünftig verringert werden.

7.1.4 Sandobjekt (3D-Punktwolke)

Der dritte 3D-Datensatz, der in dieser Arbeit zur Evaluation der DBKM verwendet wird, beinhaltet real gemessene Laserscandaten zweier Epochen eines Sandobjekts.

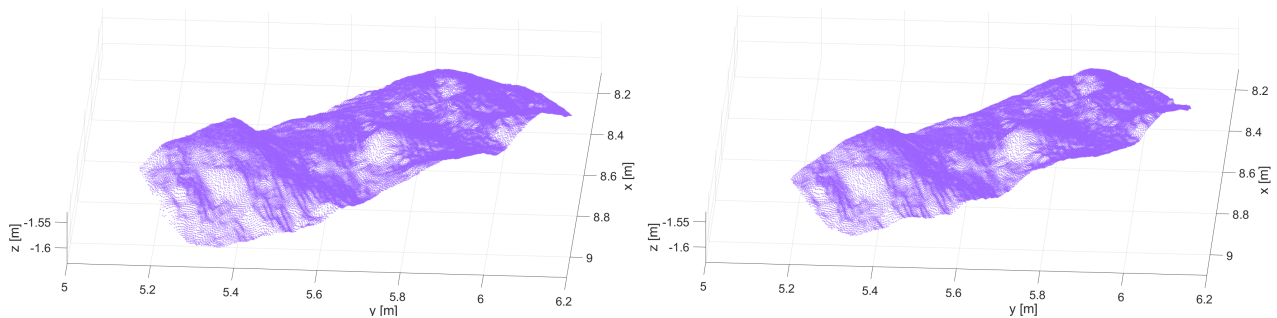


Abb. 7.35: 3D-Punktwolken des Sandobjekt von Epoche 0 (links) und Epoche 1 (rechts).

Mit ca. 140.000 Punkten und einem mittleren Punktabstand von 2,3 mm ergeben sich bei diesem Datensatz ähnliche Dimensionen wie im Falle des Tontorsos. In den Grafiken von Abbildung 7.35 werden die 3D-Punktwolken beider Epochen präsentiert. Zur Vorbereitung der Daten hinsichtlich B-Spline-Flächen-Generierung werden Kontrollpolygone definiert. Die resultierenden Histogramme sind in Abbildung 7.36 für Epoche 0 und Epoche 1 dargestellt.

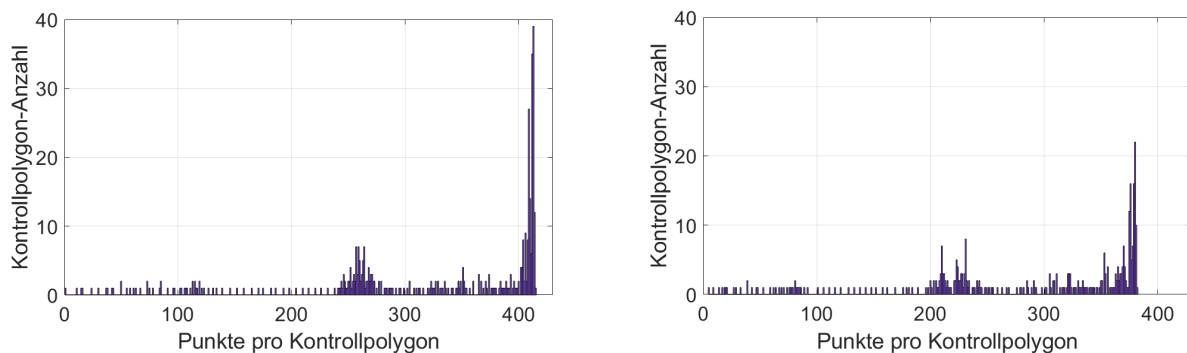


Abb. 7.36: Histogramme der definierten Kontrollpolygone von Sandobjekt Epoche 0 (links) und Epoche 1 (rechts).

Hierin ist festzustellen, dass es jeweils zwei Anhäufungen mit einer größeren Anzahl an Kontrollpolygone gibt. Bei Epoche 0 sind bei Polygonen mit einer Punktzahl zwischen 240 und 273 insgesamt 93 Polygone gebildet. Der Bereich von Punkten vor der zweiten Anhäufung beinhaltet 103 Kontrollpolygone. Die zweite Anhäufung bezieht eine Punktzahl zwischen 402 und 415 Punkten ein, die insgesamt 190 Kontrollpolygone stellen. In Relation dazu findet die erste Anhäufung bei Epoche 1 bei einer Punktzahl zwischen 200 und 234 Punkten mit einer Summe von 74 Kontrollpolygone statt. Vor der zweiten Ausprägung einer größeren Anzahl an Kontrollpolygone sind 105 Polygone gebildet. Die zweite Anhäufung bei Epoche 1 befindet sich bei einer Punktzahl zwischen 362 und 382 Punkten pro Polygon, die insgesamt 125 Kontrollpolygone abbilden. Im Vergleich zu den resultierenden Histogrammen im Falle des Tontorsos ist die Verteilung der definierten Kontrollpolygone hinsichtlich der Anzahl an Kontrollpunkten pro Polygon weiter gestreut. Dies kann mit der länglichen Form der Punktwolkenausschnitte und dem Verlauf der Laserscanaufnahmelinien schräg zur Punktwolkenkante erklärt werden. Mittels der definierten Kontrollpolygone und der Anwendung des Algorithmus von Böhm können

bidirektionale Kontrollgitterstrukturen für Epoche 0 und Epoche 1 erstellt werden. Die zwei Grafiken in Abbildung 7.37 zeigen beispielhaft einen ähnlichen Ausschnitt des Kontrollpunktgitters für Epoche 0 und Epoche 1. Im Vergleich zu den Daten des Gurkenblattobjekts kann bei den Realdaten eine weniger homogene Verteilung der Laserscanpunkte und somit der Kontrollpunkte der Kontrollgitter festgestellt werden.

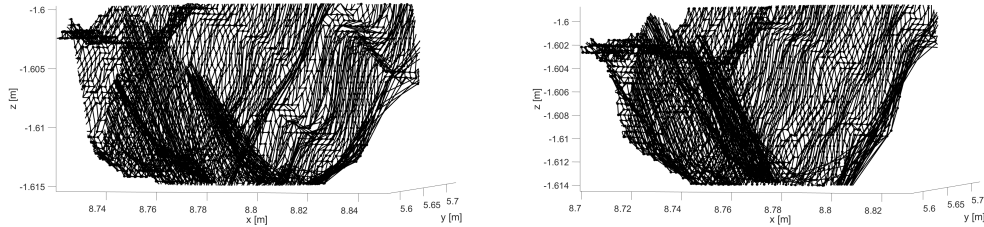


Abb. 7.37: Ausschnitt der Kontrollgitter des Sandobjekts von Epoche 0 (links) und Epoche 1 (rechts).

Aus diesen bidirektionalen Kontrollgitterstrukturen lassen sich Tensorprodukt B-Spline-Flächen generieren, die zur besseren Darstellung als Raster in Abbildung 7.38 dargestellt werden.

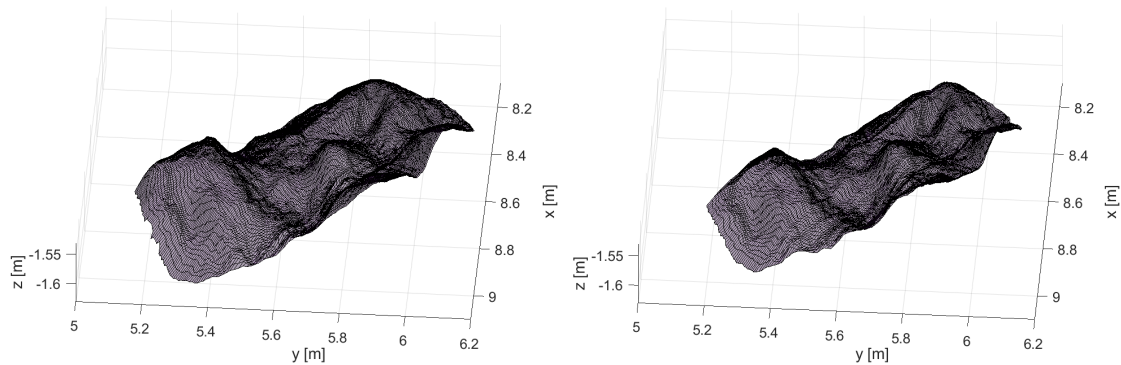


Abb. 7.38: B-Spline-Flächen des Sandobjekts von Epoche 0 (links) und Epoche 1 (rechts) (Rasterdarstellung).

Die nach der B-Spline-Flächen-Generierung erfolgte Bestimmung von Krümmungsparametern versetzt in die Lage mittels Stufe I des Mapcut-Algorithmus ein Grobmapping durchzuführen. In Abbildung 7.39 kann die resultierende Korrelationsmatrix als 3D- und 2D-Ansicht in der ersten Iteration eingesehen werden. Hierin kann eine Tendenz für die beste Überlagerungsposition der zu vergleichenden Krümmungsmatrizen festgestellt werden. Der Parameter mit der höchsten Korrelation liegt hierbei bei etwa 0,2.

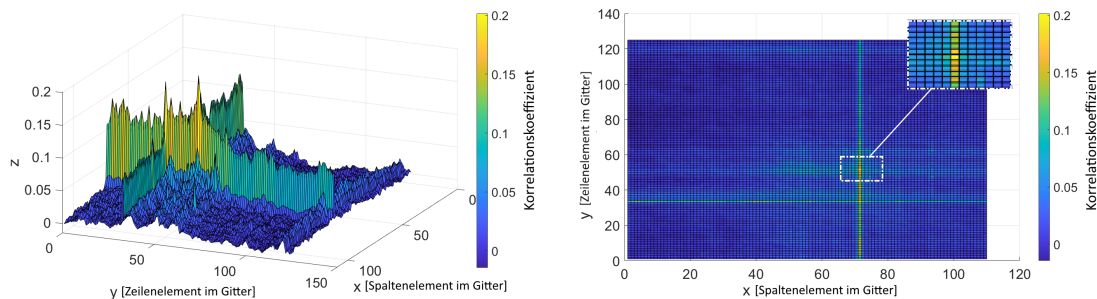


Abb. 7.39: Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 1).

In Stufe II des Mapcut-Algorithmus wird eine Korrelationsmatrix auf Basis der besten Überlagerungsposition, die im Grobmapping ausfindig gemacht wurde, in geringerer Schrittweite aufgestellt. Die zugehörigen Grafiken zum Feinmapping in Iteration 1 finden sich in Abbildung 7.40. In Relation zur Darstellung der Korrelationskoeffizientenmatrix des Tontorsos, ist der Verlauf homogener. Dies könnte mit den nicht vorhandenen Kanten im Sandobjekt zusammenhängen.

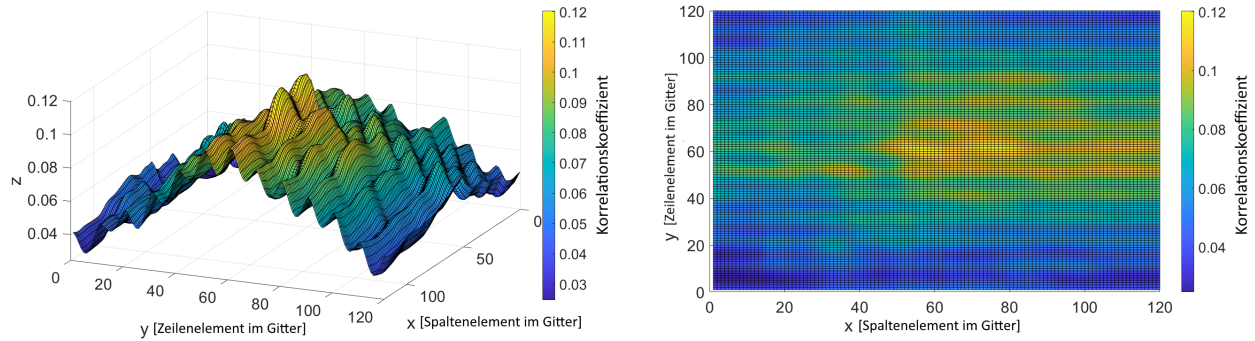


Abb. 7.40: Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 1).

Mit den in Stufe III des Mapcut-Algorithmus selektierten vorläufigen identischen Punkten erfolgt die Transformation der B-Spline-Flächenpunkte und der Kontrollpunkte von Epoche 1 in das Koordinatensystem von Epoche 0. In der linken Grafik von Abbildung 7.41 werden die vorläufigen selektierten identische Punkte samt der Koordinatendifferenzen farbig repräsentiert.

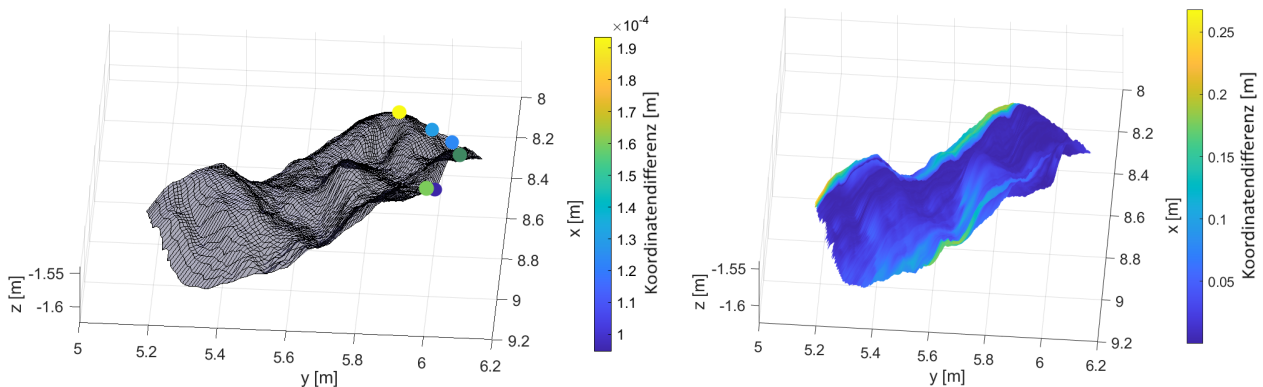


Abb. 7.41: links: Vorläufige annähernd identische Punkte, rechts: Koordinatendifferenzen korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 1).

Die Spanne der Differenzen reicht von 0,1 mm bis 0,2 mm. Diese Werte sind in erster Betrachtung gering, allerdings führt dies wie in der rechten Grafik zu sehen ist, nicht automatisch zu insgesamt geringen Koordinatendifferenzen zugehöriger B-Spline-Flächenpunkte. Diese differieren nach der Transformation in ein gemeinsames Koordinatensystem zwischen 2 cm und 25 cm. Teilweise liegt die Ursache für dieses Verhalten in den unterschiedlichen Begrenzungen der verwendeten Punktwolken. Zwar gibt es Bereiche in denen die Differenzen geringer sind, wie im Bereich, in dem die identischen Punkte zugewiesen wurden, allerdings weist die Mehrzahl an korrespondierenden B-Spline-Flächenpunkten eine wesentlich größere Koordinatendifferenz auf. Dies ist besonders in den länglichen Randbereichen in der rechten Grafik von Abbildung 7.41 zu sehen. In Abbildung 7.42 werden beide B-Spline-Flächen als Rasterdarstellung im Koordinatensystem von Epoche 0 übereinandergelagert gezeigt.

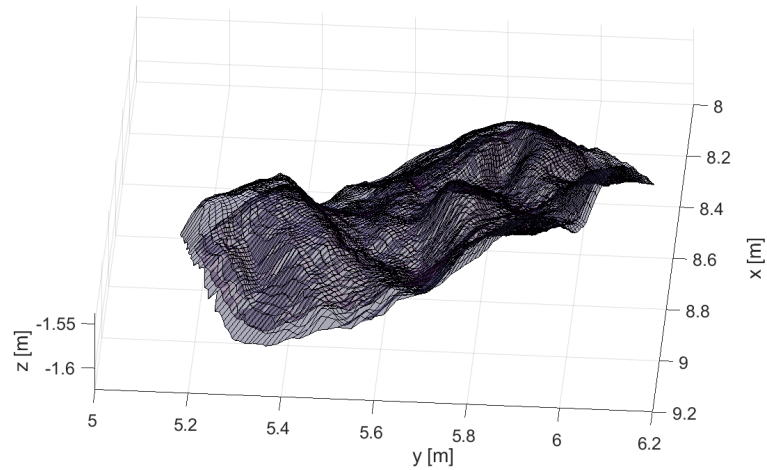


Abb. 7.42: Übereinandergelagerte B-Spline-Flächen von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 1).

Nach der Durchführung des Cuttings als Stufe IV kann durch eine dreimalige iterative Abwicklung der DBKM eine deutliche Verbesserung des Gesamtprozesses erreicht werden. Abbildung 7.43 zeigt die Korrelationsmatrix des Grobmappings bei der dritten Iteration aus zwei Perspektiven. Hierbei kann ein ausgeprägtes Peak mit einer Korrelation von über 0,35 festgestellt werden, das die Basis für ein feineres Mapping bietet, dessen Ergebnis in Abbildung 7.44 veranschaulicht wird.

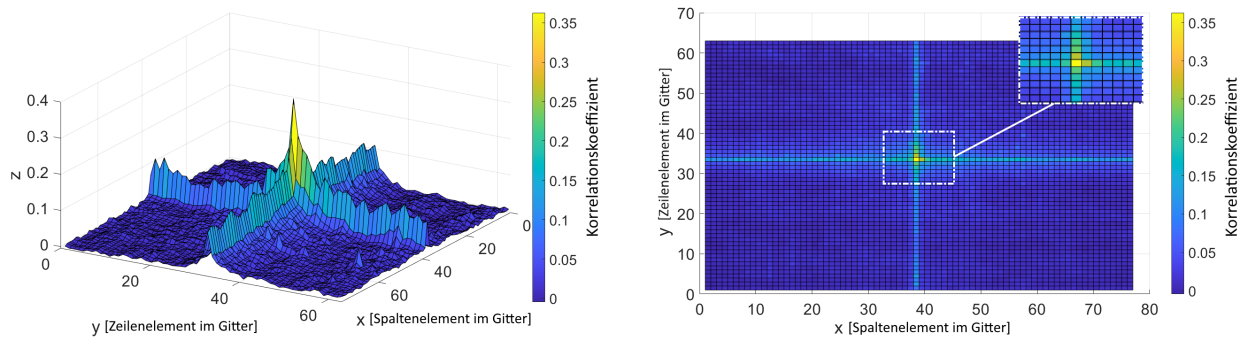


Abb. 7.43: Grobmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 3).

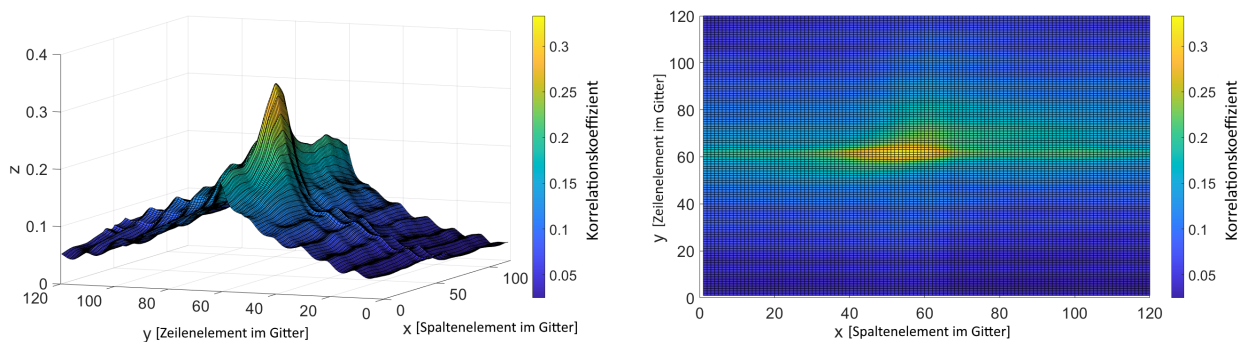


Abb. 7.44: Feinmapping von Epoche 0 und Epoche 1 als 3D- und 2D-Ansicht (Sandobjekt, Iteration 3).

Betrachtet man das Ergebnis der Selektion der annähernd identischen Punkte für die Koordinatentransformation in Abbildung 7.45, können Differenzen zwischen den als identische Punkte definierten zugehörigen Punkten von 0,01 mm bis 0,05 mm abgelesen werden.

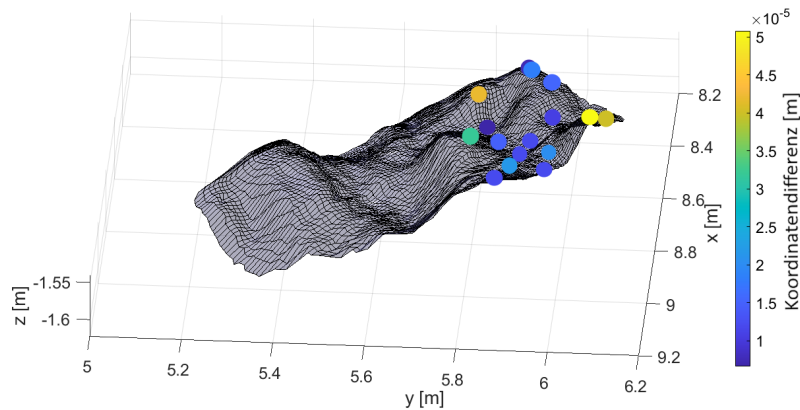


Abb. 7.45: Identische Punkte von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 3).

Trotz dieser geringen Differenzen zwischen den identischen Punkten, ergeben sich bei der Transformation der B-Spline-Flächenpunkte von Epoche 1 in das Koordinatensystem von Epoche 0 nicht ausschließlich geringe Differenzen zwischen korrespondierenden B-Spline-Flächenpunkten. Die Koordinatendifferenzen zugehöriger B-Spline-Flächenpunkte in Abbildung 7.46 zeigen, dass die Spanne bei 0,1 cm bis 2,5 cm liegt. Überwiegend weisen die korrespondierenden B-Spline-Flächenpunkte eine Differenz von 1,1 mm auf.

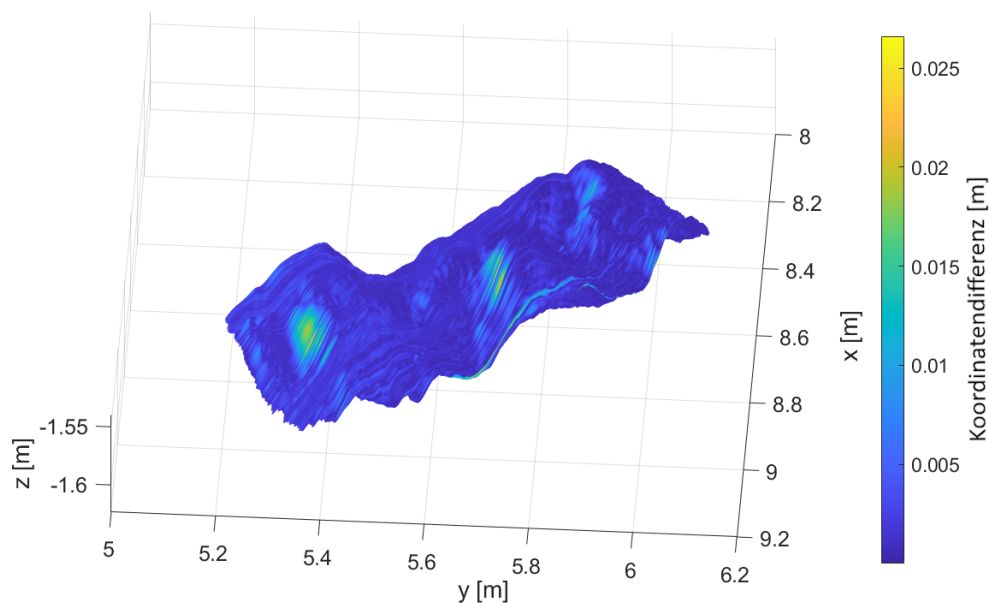


Abb. 7.46: Koordinatendifferenzen von Epoche 0 und Epoche 1 (Sandobjekt, Iteration 3).

Ähnlich wie im Falle des Tontorsos fällt auf, dass große Koordinatendifferenzen von bis zu 2,5 cm an Positionen in der Punktwolke auftreten, die Erhebungen oder Vertiefungen enthalten. Eine nähere Diskussion dieser Verhaltensweise findet im nächsten Abschnitt 7.1.5 statt.

7.1.5 Diskussion der Ergebnisse

Sowohl bei dem Datensatz des Tontorsos als auch des Sandobjekts kann festgestellt werden, dass an Kanten, Erhebungen oder Vertiefungen der Punktwolken größere Koordinatendifferenzen nach Durchführung der DBKM auftreten. Um dieses Verhalten näher zu beleuchten, werden ein vergrößerter Ausschnitt der 3D-Punktwolke des Sandobjekts bei einer Erhebung in Abbildung 7.47 und ein vergrößerter Ausschnitt der 3D-Punktwolke des Tontorsos bei einer Vertiefung sowie einer Kante in Abbildung 7.48 dargestellt.

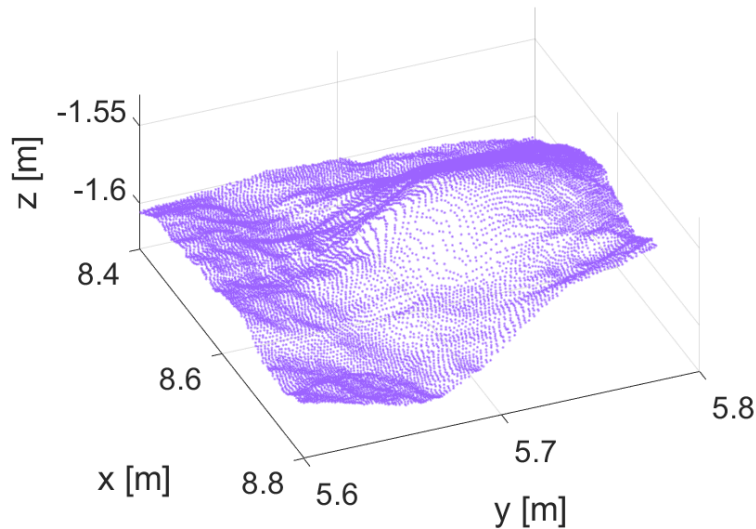


Abb. 7.47: Ausschnitt der 3D-Punktwolke des Sandobjekts im Bereich einer Erhebung.

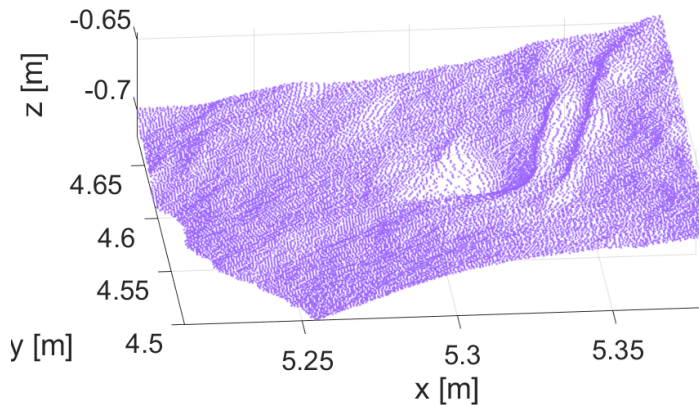


Abb. 7.48: Ausschnitt der 3D-Punktwolke des Tontorso im Bereich einer Vertiefung und von Kanten.

Anhand der beiden Abbildungen, kann beobachtet werden, dass in flachen Bereichen der Punktwolken eine gleichmäßige Verteilung der TLS-Daten vorliegt. In den Bereichen der Punktwolken, die eine Erhebung, Vertiefung oder Kante des Messobjekts diskretisieren, sind die TLS-Daten ungleichmäßiger angeordnet. In diesen Bereichen können deutlich größere Punktabstände zwischen aufeinanderfolgenden TLS-Punkten sowie zwischen benachbarten Laserscanreihen festgestellt werden. Da die Laserscanreihen bei Aufnahme weiterer Epochen unterschiedlich positioniert sind und sich diese unterschiedliche Positionierung in Bereichen von Erhebungen stärker durch vergrößerte und ungleichmäßigere Punktabstände

ausprägt, führt dies auch im Kontrollgitter und der B-Spline-Fläche zu unterschiedlichen Punktpositionierungen. Durch die differierenden Punktabstände und den voneinander abweichenden Verlauf der Laserscanaufnahmelinien zweier Epochen entstehen somit in der B-Spline-Fläche größere Koordinatendifferenzen zwischen zugewiesenen korrespondierenden B-Spline-Flächenpunkten in Bereichen mit Erhebungen als in flachen Bereichen ohne Kanten oder Vertiefungen. Mit der derzeitig entwickelten DBKM können die Auswirkungen von Erhebungen oder Vertiefungen der TLS-Daten in den B-Spline-Flächen bisher nicht kompensiert werden. Ein Lösungsansatz zur Behebung dieser Verhaltensweise, könnte eine Verdichtung der Laserscanpunkte in selektierten Bereichen von Erhebungen oder Vertiefungen darstellen, um die Punktdichte auf die Punktdichte der restlichen Punktwolke anzugleichen. Eine derartige Verdichtung ist bisher allerdings noch kein Bestandteil der entwickelten Methode. In zukünftigen Arbeiten kann dieser Ansatz weiter verfolgt werden.

Für die Evaluation der Anwendung der DBKM an den vier vorgestellten Datensätzen kann festgehalten werden, dass der Einsatz der Kernelemente der DBKM betreffend der Modellierung von TLS-Daten mittels Tensorprodukt B-Spline-Flächen und der Anwendung des Mapcut-Algorithmus als lokales Registrierungsverfahren erfolgreich durchgeführt werden konnte. Der auf simulierten Daten implementierte Ansatz der DBKM ist auf die Anwendung von Realdaten übertragbar. Für die simulierten Daten des 2D-Datensatzes werden korrespondierende B-Spline-Kurvenpunkte generiert, die eine maximale Euklidische Distanz von 1,2 mm vorweisen. Für die simulierten Daten des 3D-Datensatzes des Gurkenblatts kann bei den erzeugten korrespondierenden B-Spline-Flächenpunkten eine maximale Euklidische Distanz von 0,9 mm festgestellt werden. Die Größenordnung der maximalen Euklidischen Distanzen ist durch das in den Punktwolken enthaltene Messrauschen begründbar und stellt durch die direkte funktionale Approximation der TLS-Daten mittels B-Spline-Flächen einen limitierenden Faktor der DBKM dar. Aufgrund der ständigen Entwicklungsoptimierung bei terrestrischen Laserscannern und dem zukünftigen Verbesserungspotential hinsichtlich der Genauigkeit der Punktwolkenerfassung mittels TLS, wird dieser limitierende Faktor zukünftig zunehmend geringer ausfallen. Bei den realen Datensätzen konnten bisher Euklidische Distanzen der korrespondierenden B-Spline-Flächenpunkte zwischen 0,01 cm und 2,5 cm erreicht werden. Zwar enthalten Bereiche der modellierten und lokal registrierten B-Spline-Flächen überwiegend Euklidische Distanzen zwischen den Punktkorrespondenzen von etwa 1,1 mm, allerdings treten in Bereichen von Erhebungen, Vertiefungen oder Kanten Euklidische Distanzen bis zu 2,5 cm auf. Dieses Verhalten bedingt zusätzlichen und zukünftigen Entwicklungsbedarf. Für eine Weiterverwendung im Zusammenhang mit einer Deformationsanalyse sind die resultierenden Koordinatendifferenzen bei Realdaten somit bisher in exponierten Bereichen zu groß, um Deformationen im mm-Bereich ausfindig machen zu können.

Für eine Evaluation der Anwendung der Deformationsanalyse in Unterkapitel 7.2 werden die jeweilige Messepoche 0 der vier Datensätze mit einem Gaußschen Rauschen überlagert und als Messepoche 1 herangezogen. Ebenso werden den vier Datensätzen simulierte Deformationen im mm-Bereich hinzugefügt. Die Position der Deformation in der Punktwolke wird in den jeweiligen Abschnitten der Datensätze gezeigt. Eine Festlegung der Deformationen im geringen mm-Bereich erfolgt aufgrund der bisherigen Ergebnisse aus Unterkapitel 7.1 Abschnitt 7.1.1 und Abschnitt 7.1.2.

7.2 Anwendung der Deformationsanalyse

Für die Deformationsanalyse mittels DBKM im Rahmen von B-Spline-Kurven und -Flächen werden die Koordinatendifferenzen zugehöriger B-Spline-Kurven- und Flächenpunkte mit der entsprechenden Kofaktormatrix in einem Signifikanztest auf signifikante Deformationen geprüft. Die im vorherigen Unterkapitel verwendeten Datensätze werden für die praxisnahe Vorstellung der Deformationsanalyse eingesetzt. Im Detail wird Epoche 0 mit einem Gaußschen Rauschen von 0,5 mm versetzt und eine Deformation angebracht, um Epoche 1 zu generieren.

Die Deformationsanalyse an B-Spline-Kurven wird anhand der simulierten 2D-Punktwolke eines Fassadenstücks mit einer anwachsenden 2D-Deformation bis zu 3,0 mm vorgestellt. Für die Darstellung der Deformationsanalyse bei B-Spline-Flächen werden die drei Testobjekte Gurkenblatt, Tontorso und Sandobjekt verwendet und eine mittig ansteigende 3D-Deformation bis zu 3,0 mm angebracht. Prinzipiell werden die Deformationen bei den Datensätzen im mittleren Bereich platziert. Bei der Positionierung der Deformation in der Punktwolke wurde darauf geachtet, dass diese nicht am Rand liegen, um eine erfolgreiche Durchführung des Mapcut-Algorithmus zu gewährleisten. Bei der Anbringung der 2D- und 3D-Deformation wurden die einzelnen Koordinaten x , y und z unterschiedlich stark deformiert, um die spezielle Eigenschaft der DBKM, Aussagen über Deformationen betreffend einzelner Koordinatenrichtungen treffen zu können, zu verdeutlichen.

7.2.1 Fassadenstück (2D-Punktwolke)

Für die Darstellung der Funktionalität der DBKM für B-Spline-Kurven wird die 2D-Punktwolke des Fassadenstücks aus Unterkapitel 7.1.1 herangezogen. Zusätzlich wird eine Deformation, die 21 der insgesamt 261 Kontrollpunkte und somit 8 % der Kontrollpunkte betrifft, angewendet. Abbildung 7.49 zeigt die Kontrollpolygone beider Epochen. Das Kontrollpolygon von Epoche 0 ist magenta gefärbt und das Kontrollpolygon von Epoche 1 blau. Zur besseren Darstellung der deformierten Kontrollpunkte werden die zwei Kontrollpolygone ohne Drehung und Translation abgebildet. Das gestrichelte Rechteck in Abbildung 7.49 stellt eine Vergrößerung dar und markiert den Bereich, in dem das Kontrollpolygon von Epoche 1 in Relation zu Epoche 0 deformiert ist.

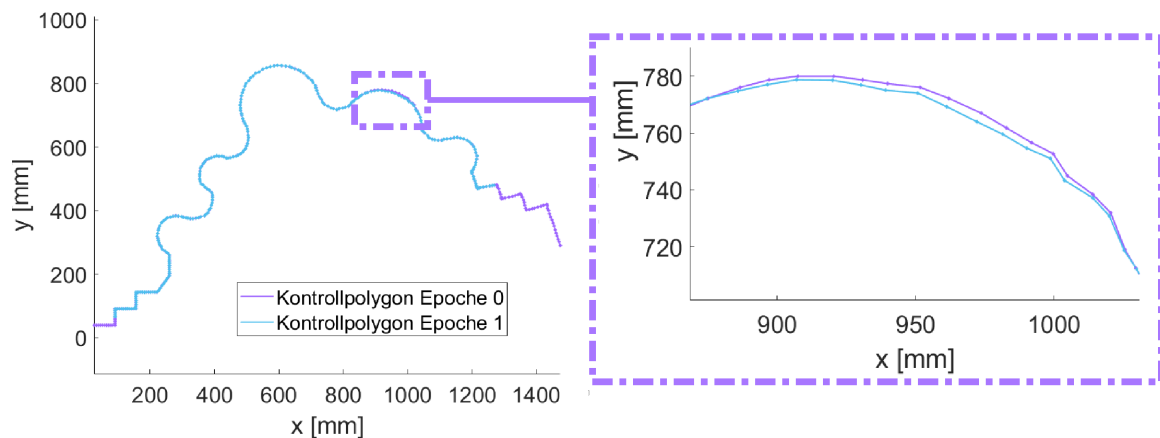


Abb. 7.49: Kontrollpolygone von Epoche 0 und Epoche 1 mit Deformation (2D-Fassadenstück).

Für eine Übersicht der Koordinatendifferenzen naheliegender Kontrollpunkte vor Durchführung der entwickelten Methode sei auf die drei Grafiken mit den Koordinatendifferenzen in x- und y-Koordinate sowie der Euklidischen Distanz der Kontrollpunkte in Abbildung 7.50 verwiesen.

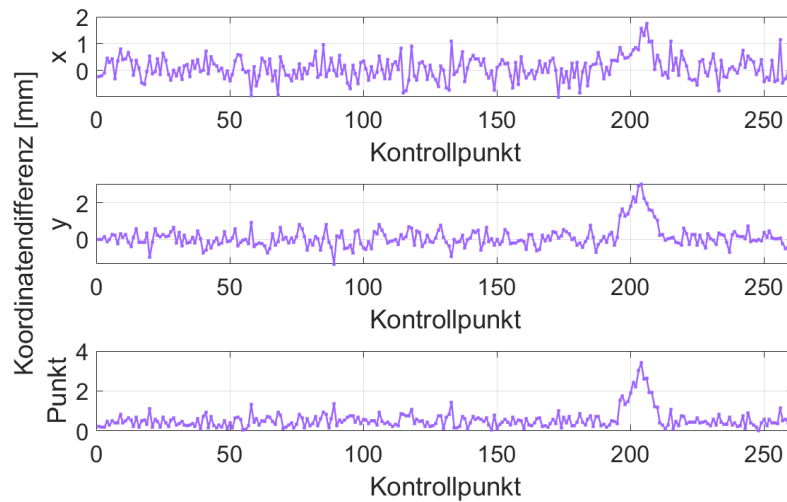


Abb. 7.50: Koordinatendifferenzen der Kontrollpunkte von Epoche 0 und Epoche 1 (2D-Fassadenstück).

Diese spiegeln das Messrauschen beider Epochen zueinander wider. Eine Deformation kann im Bereich des 200-ten Kontrollpunktes erkannt werden. Für eine gezielte Vergleichsmöglichkeit der Funktionsweise der DBKM werden die angebrachten Deformationen in ihrer Reinform ohne Messrauschen in Abbildung 7.51 dargestellt. Die darunterliegende Tabelle 7.1 gibt die maximale Deformation in x- und y-Koordinate sowie im Kontrollpunkt aus.

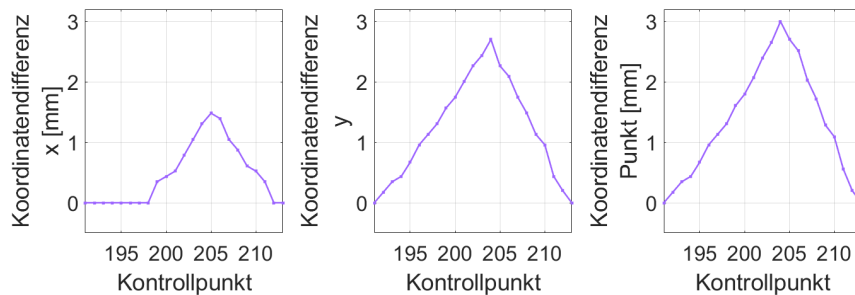


Abb. 7.51: Deformationen von Epoche 1 in x-Koordinate, y-Koordinate und im Kontrollpunkt ohne Messrauschen (Referenz-2D-Deformation, 2D-Fassadenstück).

	x [mm]	y [mm]	2D-Deformation [mm]
max. Deformation	1,5	2,7	3,0

Tab. 7.1: Maximale Deformation von Epoche 1 ohne Messrauschen (2D-Fassadenstück).

In der x-Koordinate beginnt die geringste Deformation bei 0,3 mm und wächst auf 1,5 mm an. In der y-Koordinate beginnt die kleinste Deformation bei 0,2 mm und steigt auf 2,7 mm an. Für die Euklidische Distanz im Kontrollpunkt bzw. die 2D-Deformation bedeutet dies eine maximale Deformation von 3,0 mm. Anhand von Abbildung 7.51 soll, abgesehen von den maximalen Deformationen,

gezeigt werden, dass eine Anbringung von langsam ansteigenden Deformationen verfolgt wird. Auf diese Weise kann die gesamte Deformationsspanne von 0 mm bis 3,0 mm betreffend einer signifikant feststellbaren 2D-Deformation betrachtet werden. Durch die Anwendung der DBKM für die beiden Epochen resultieren korrespondierende B-Spline-Kurvenpunkte, deren Koordinatendifferenzen zusammen mit der Kofaktormatrix aus der stochastischen Modellierung in einen Signifikanztest in Form eines F-Tests übergeben werden können. Der Signifikanztest entscheidet, welche Kurvenpunkte als signifikant deformiert ausgewiesen werden.

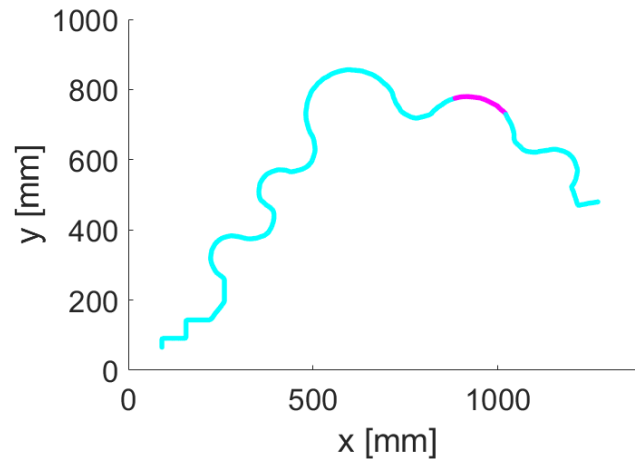


Abb. 7.52: Ergebnis des Signifikanztests (2D-Fassadenstück).

Das Ergebnis des Signifikanztests für die B-Spline-Kurvenpunkte des Fassadenstücks ist in Abbildung 7.52 präsentiert. Hierbei stellen hellblaue Kurvenbereiche keine signifikante Deformation dar. Magenta-farbene Kurvenpunkte präsentieren die signifikante Deformation zwischen Epoche 0 und Epoche 1. Der gleiche deformierte Kurvenbereich ist ebenso in Abbildung 7.53 realisiert, allerdings werden darin die Koordinatendifferenzen der zugehörigen B-Spline-Kurvenpunkte abgebildet.

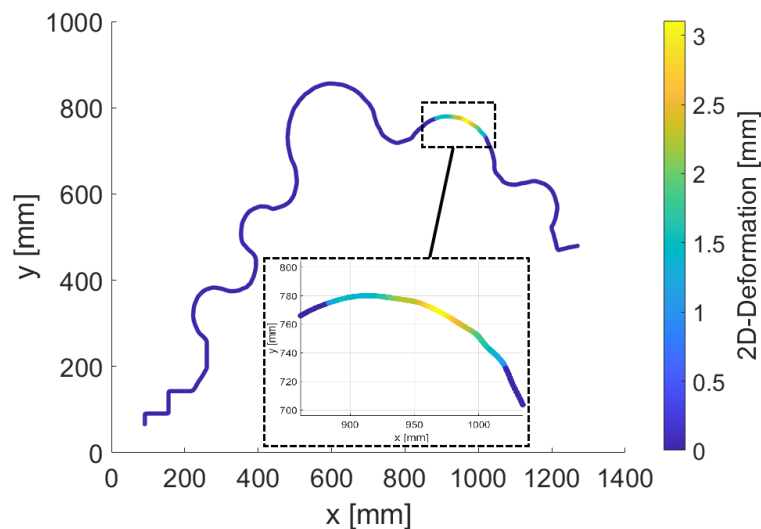


Abb. 7.53: 2D-Deformationen der B-Spline-Kurvenpunkte in signifikant deformierten Bereichen (2D-Fassadenstück).

Hierbei werden ausschließlich die als signifikant deformierten Kurvenpunkte mit ihren 2D-Deformationen gezeigt. Dunkelblau gefärbte Kurvenpunkte sind nicht signifikant deformiert und deshalb automatisch

auf eine 2D-Deformation von 0 mm gesetzt. Das schwarze Rechteck ermöglicht eine vergrößerte Darstellung des Deformationsbereichs und zeigt eine 2D-Deformation von etwa 3 mm an. Genauer können die Koordinatendifferenzen zugehöriger B-Spline-Kurvenpunkte in Abbildung 7.54 und als vergrößerte Darstellung in Abbildung 7.55 betrachtet werden.

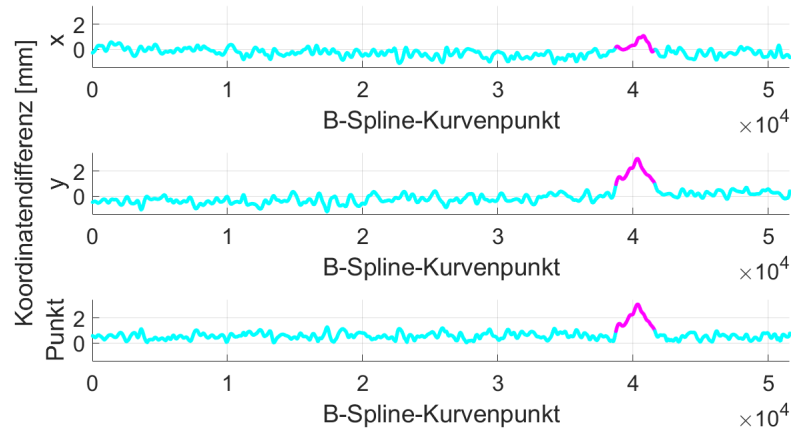


Abb. 7.54: Koordinatendifferenzen der B-Spline-Kurvenpunkte von Epoche 0 und Epoche 1 in signifikant deformierten Bereichen (2D-Fassadenstück).

Hellblau gefärbte Koordinatendifferenzen liegen in nicht signifikant deformierten Bereichen, während magentafarbene Koordinatendifferenzen eine Deformation im Kurvenpunkt aufzeigen. Um einen verbesserten grafischen Vergleich der als signifikant ausgewiesenen Deformationen realisieren zu können, werden in Abbildung 7.55 vergrößerte Ausschnitte der Koordinatendifferenzen korrespondierender B-Spline-Kurvenpunkte in x- und y-Koordinate und in den B-Spline-Kurvenpunkten dargestellt.

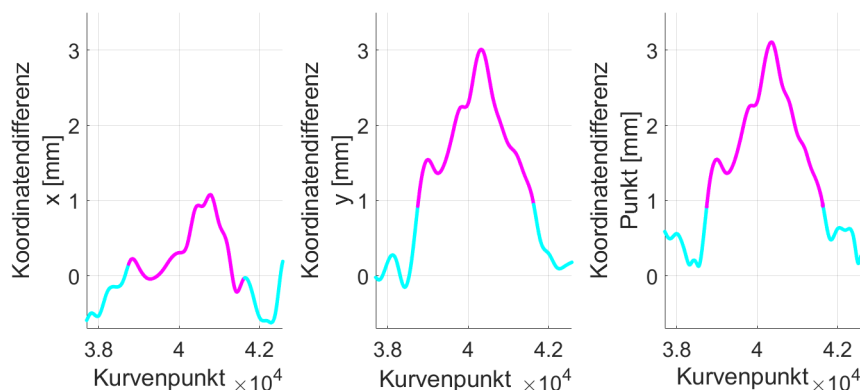


Abb. 7.55: Ausschnitte der signifikant deformierten Koordinatendifferenzen zwischen Epoche 0 und Epoche 1 in x-Koordinate, y-Koordinate und im B-Spline-Kurvenpunkt (2D-Fassadenstück).

In der x-Koordinate kann eine maximale Deformation von 1,1 mm abgelesen werden. In Relation zur angebrachten Deformation mit 1,5 mm sind dies 0,4 mm weniger. In der y-Koordinate ist eine maximale Deformation von 3,0 mm abzugreifen. Im Vergleich zur Initialdeformation mit 2,7 mm ist dies ein Unterschied von 0,3 mm. Im B-Spline-Kurvenpunkt äußert sich dies durch eine maximale 2D-Deformation von 3,1 mm, was einem Unterschied von 0,1 mm entspricht. Dieser vergleichsweise

geringe Unterschied kann damit begründet werden, dass die x-Koordinatendifferenz kleiner und die y-Koordinatendifferenz größer als die angebrachte Deformation von Tabelle 7.1 ausfallen und sich somit ausgleichen. In der rechten Grafik kann, abgesehen von der maximalen Deformation, auch die Euklidische Distanz im Kurvenpunkt abgelesen werden, ab der eine Deformation als signifikant ausgewiesen wird. Dies sind im 2D-Fall etwa 0,9 mm. Somit kann festgestellt werden, dass eine Deformation bei der 2D-Punktwolke des Fassadenstücks von Epoche 1 ab etwa 0,9 mm festgestellt werden kann. Zusätzlich ist anzumerken, dass sich die in Epoche 1 enthaltenen Deformationen trotz des Messrauschens mit Hilfe der Zuordnung der Punktkorrespondenzen durch die DBKM lediglich um 0,4 mm vom ursprünglichen Deformationswert unterscheiden.

7.2.2 Gurkenblatt (3D-Punktwolke)

Für die Deformationsanalyse des Gurkenblatt-Testobjekts werden zwei Epochen der Punktwolken aus Abschnitt 7.1.2 verwendet. Epoche 1, die ein Weißes Gaußsches Rauschen von 0,5 mm enthält, wird mit einer anwachsenden Deformation entsprechend Tabelle 7.2 belegt.

	x [mm]	y [mm]	z [mm]	3D-Deformation [mm]
max. Deformation	0,4	1,8	2,4	3,0

Tab. 7.2: Maximale Deformation von Epoche 1 ohne Messrauschen betreffend x-, y-, z-Koordinate und 3D-Deformation (Gurkenblatt, Tontorso, Sandobjekt).

Die in Tabelle 7.2 angegebenen maximalen Deformationswerte werden bei den drei 3D-Datensätzen Gurkenblatt, Tontorso und Sandobjekt in gleicher Höhe angebracht. Die Position des deformierten Bereichs in der Punktwolke wird mittig gewählt. Empirische Untersuchungen haben gezeigt, dass lediglich eine Deformation im Randbereich vermieden werden sollte, um den Mapcut-Algorithmus im Hinblick auf die Anfangs- und Endpunkt-Eigenschaft der B-Spline-Flächen erfolgreich durchführen zu können. An welcher Position die Deformation im mittleren Bereich in der Punktwolke genau gewählt wird, ist hierbei unbedeutend. Bei der Simulation der Deformationen für Epoche 1 wurde darauf geachtet, dass eine ansteigende Deformation untersucht werden kann, um einen Grenzwert ermitteln zu können, ab dem signifikante Deformationen feststellbar sind. Der maximale Deformationswert wird, wie im 2D-Fall, auf 3,0 mm festgelegt. Für die x-, y- und z-Koordinate werden jeweils unterschiedlich große maximale Deformationen gewählt. Dies soll die Eigenschaft der DBKM aufzeigen, dass Deformationen nicht nur im Hinblick auf Abstände, sondern ebenfalls auf die Änderung einzelner Koordinatenrichtungen untersucht werden können. Zusätzlich wird bei der Deformation einer Koordinate eine maximale Deformation unterhalb des Messrauschens von 0,5 mm verwendet, um zu verdeutlichen, dass auch geringe Deformationen in einer Koordinatenrichtung bestimmbar sein können, solange die gesamte 3D-Deformation den notwendigen Grenzwert überschreitet. Für die x-Koordinate wird eine maximale Deformation von 0,4 mm, für die y-Koordinate von 1,8 mm und für die z-Koordinate von 2,4 mm verwendet.

Um die angebrachte 3D-Deformation grafisch darstellen zu können, werden in Abbildung 7.56 korrespondierende B-Spline-Flächenpunkte auf Basis zweier Epochen ohne Messrauschen abgebildet. Von der Deformation sind 305 von insgesamt 15.017 TLS-Punkten betroffen. Dies entspricht einem Deformationsanteil von etwa 2 %. Anhand des vergrößerten Ausschnitts, kann die anwachsende Deformation zum Zentrum des Deformationsbereichs hin sowie die maximale 3D-Deformation von 3 mm nachvollzogen werden.

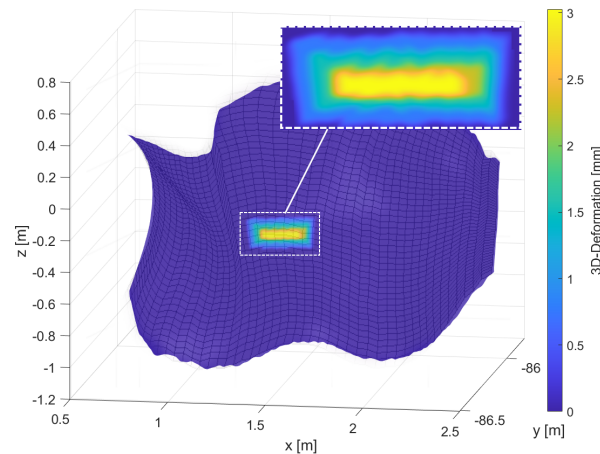


Abb. 7.56: 3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 ohne Messrauschen (Referenz-3D-Deformation, Gurkenblatt).

Nach Anwendung der DBKM kann mittels Durchführung des Signifikanztests eine eindeutige Identifikation des signifikant deformierten Bereichs festgestellt werden. Die signifikant deformierten B-Spline-Flächenpunkte von Epoche 1 sind in Abbildung 7.57 in gelb dargestellt.

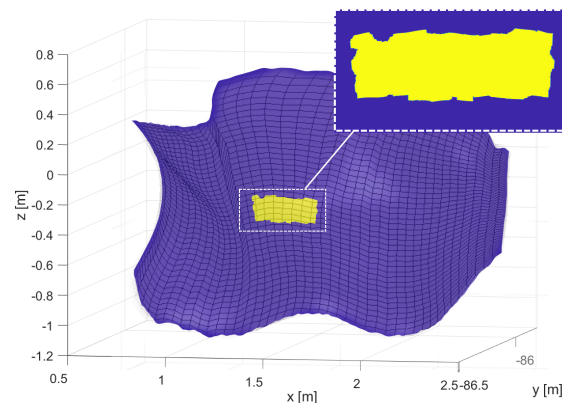


Abb. 7.57: Ergebnis des Signifikanztests (Gurkenblatt).

Um einen Eindruck zu erhalten, welche deformierten Bereiche im Signifikanztest erkannt werden und ab welchem Wert der 3D-Deformation dies erfolgt, stellt die obere Grafik von Abbildung 7.58 die unverrauschten 3D-Deformationen der korrespondierenden B-Spline-Flächenpunkte als Referenzbild farblich dar. Die äußere graue Konturlinie in der oberen Grafik bildet eine Begrenzung der 3D-Deformation im Referenzbild. Die innere graue Konturlinie verdeutlicht die Begrenzung der mit der DBKM und dem Signifikanztest bestimmten 3D-Deformation. Demzufolge bilden die 3D-Deformationen zwischen äußerer und innerer Konturlinie diese 3D-Deformationen ab, die durch die DBKM aufgrund des Rauschniveaus nicht bestimmt werden können. Alle 3D-Deformationen zwischen korrespondierenden B-Spline-Flächenpunkten innerhalb der inneren grauen Konturlinie können dagegen vollständig festgestellt werden. Die untere Grafik von Abbildung 7.58 bildet einen Längsschnitt durch die 3D-Deformation ab, deren Verlauf in der oberen Grafik durch die weiß gestrichelte Linie markiert wird. Die graue Linie in der unteren Grafik stellt hierbei die 3D-Deformation zwischen korrespondierenden B-Spline-Flächenpunkten auf Basis von unverrauschten 3D-Punktvolken dar und soll wiederum als Referenz dienen.

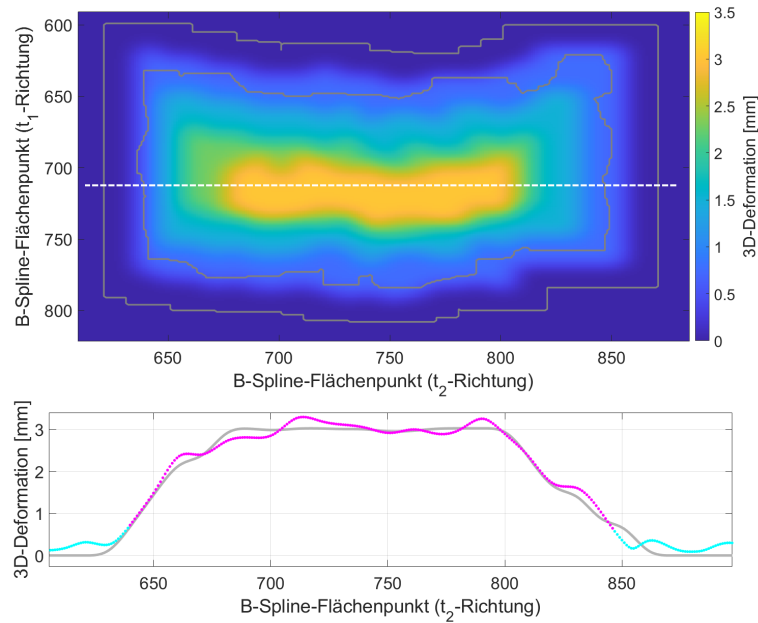


Abb. 7.58: Oben: 3D-Deformation der unverrauschten Messdaten als 2D-Darstellung. Graue äußere Konturlinie: Begrenzung der Referenz-3D-Deformation, innere Konturlinie: Begrenzung der durch die DBKM signifikant festgestellten 3D-Deformation. Unten: Längsschnitt durch 3D-Deformation. Graue Linie: Referenz, Magentafarbene Linie: signifikant festgestellte 3D-Deformation (Gurkenblatt).

Die hellblaue und magentafarbene Linie zeigt die 3D-Deformation korrespondierender B-Spline-Flächenpunkte auf Basis der DBKM und der verrauschten 3D-Punktwolken des Gurkenblatts. Die hellblaue Linie verdeutlicht nicht signifikante Punktdifferenzen zwischen den B-Spline-Flächenpunkten, während die magentafarbene Linie signifikant deformierte B-Spline-Flächenpunkte abbildet. Aus dem Übergang von hellblauer zu magentafarbener Linie kann abgelesen werden, dass 3D-Deformationen ab etwa 0,7 mm als signifikant eingestuft werden. Der als signifikant deformiert festgelegte Bereich der B-Spline-Fläche kann zusätzlich in Bezug auf eine jeweilige Deformation im Hinblick auf die x-, y- und z-Koordinate untersucht werden. Ein vergrößerter Ausschnitt der signifikanten Deformation bzgl. 3D-Deformation und Deformation in der x-Koordinate ist in den Grafiken in Abbildung 7.59 dargestellt.

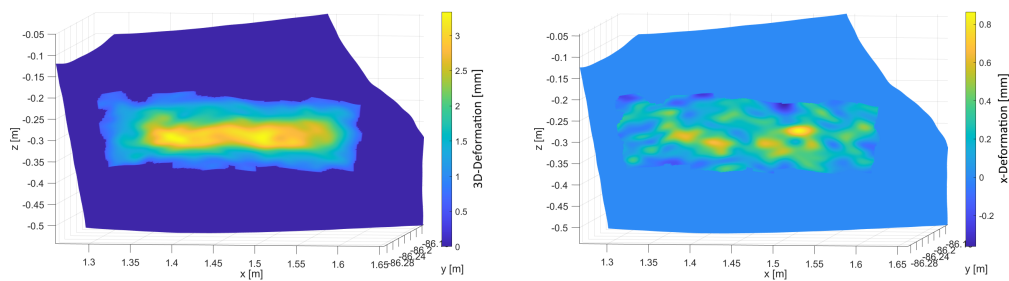


Abb. 7.59: Vergrößerter Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Gurkenblatt). Links: B-Spline-Flächenpunkte, rechts: x-Koordinate.

Die Deformation in der y-Koordinate und z-Koordinate ist als vergrößerter Ausschnitt in den Grafiken in Abbildung 7.60 abgebildet.

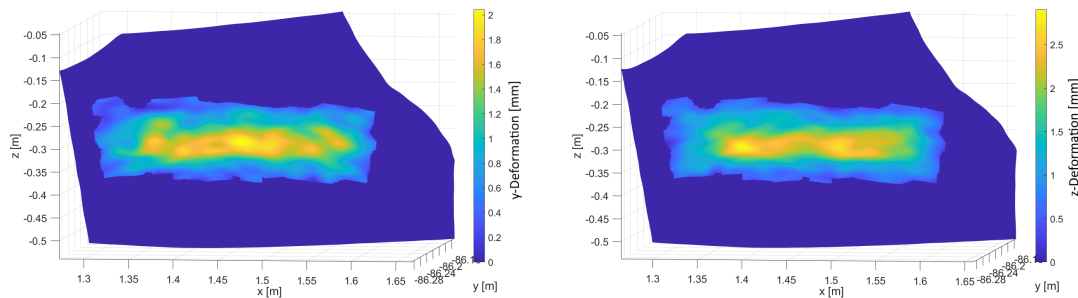


Abb. 7.60: Vergrößerter Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Gurkenblatt). Links: y-Koordinate, rechts: z-Koordinate.

In den genannten Grafiken werden zur leichteren optischen Identifikation der signifikant deformierten Bereiche, Koordinatendifferenzen zwischen korrespondierenden B-Spline-Flächenpunkten außerhalb des Signifikanzbereichs mit dem Wert 0 belegt. Bei Betrachtung der 3D-Deformationen in der linken Grafik von Abbildung 7.59 kann eine maximale Deformation von 3,3 mm statt der angebrachten 3,0 mm abgelesen werden. Die rechte Grafik zeigt eine maximale Deformation in der x-Koordinate von 0,8 mm statt der angebrachten 0,4 mm an. Bei der y-Koordinate beträgt die maximale Deformation 2,0 mm statt 1,8 mm und bei der z-Koordinate 2,8 mm statt 2,4 mm. Somit unterscheiden sich die signifikanten maximalen Deformationen korrespondierender B-Spline-Flächenpunkte, die mittels DBKM zugewiesen wurden, in der 3D-Deformation sowie den einzelnen Koordinaten um bis zu 0,4 mm. Bei Betrachtung der maximalen Differenz von 0,4 mm ist darauf hinzuweisen, dass ein Zusammenhang mit den nach der Normalverteilung generierten Punktdaten berücksichtigt werden sollte.

7.2.3 Tontorso (3D-Punktwolke)

Für den Tontorso wird die Punktwolke mit 150.074 Laserscanpunkten von Abschnitt 7.1.3 als Epoche 0 und eine verrauschte Variante als Epoche 1 herangezogen.

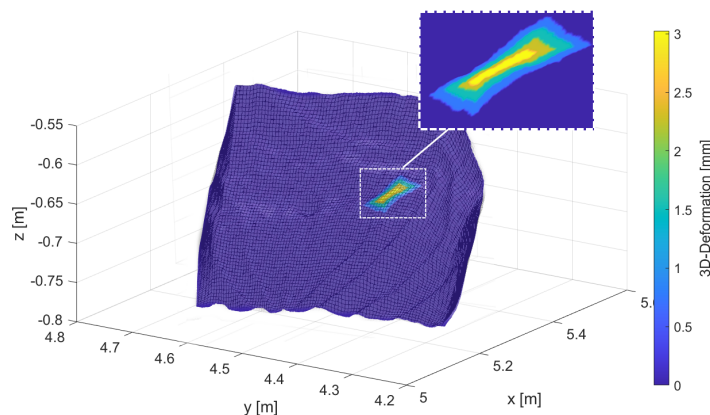


Abb. 7.61: 3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 ohne Messrauschen (Referenz-3D-Deformation, Tontorso).

Entsprechend Tabelle 7.2 wird eine langsam ansteigende 3D-Deformation von maximal 3,0 mm angebracht. Die maximalen Deformationen in der x-, y- und z-Koordinate orientieren sich ebenso an den Werten in Tabelle 7.2. Abbildung 7.61 zeigt die angebrachte 3D-Deformation in Bezug auf korrespondierende B-Spline-Flächenpunkte zweier Epochen. Für eine bessere Darstellung der reinen Deformation werden für die Visualisierung zwei Epochen ohne Messrauschen verwendet. Die 3D-Deformation befindet sich im mittleren Bereich der Punktwolke und nimmt mit 2.600 von 150.074 Laserscanpunkten etwa 2 % der TLS-Punkte in Anspruch. Nach Anwendung der DBKM kann innerhalb der Deformationsanalyse durch einen Signifikanztest eine Identifikation signifikant deformierter B-Spline-Flächenpunkte vorgenommen werden. Diese werden in Abbildung 7.62 in gelb dargestellt.

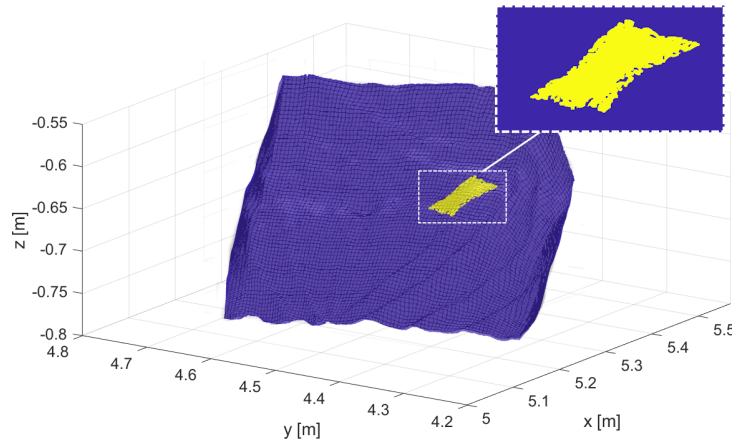


Abb. 7.62: Ergebnis des Signifikanztests (Tontorso).

Für eine Untersuchung des Grenzwerts ab dem 3D-Deformationen innerhalb der DBKM als signifikant deformiert festgestellt werden können, stellt die linke Grafik von Abbildung 7.63 die unverrauschten 3D-Deformationen der korrespondierenden B-Spline-Flächenpunkte als Referenzbild farblich dar.

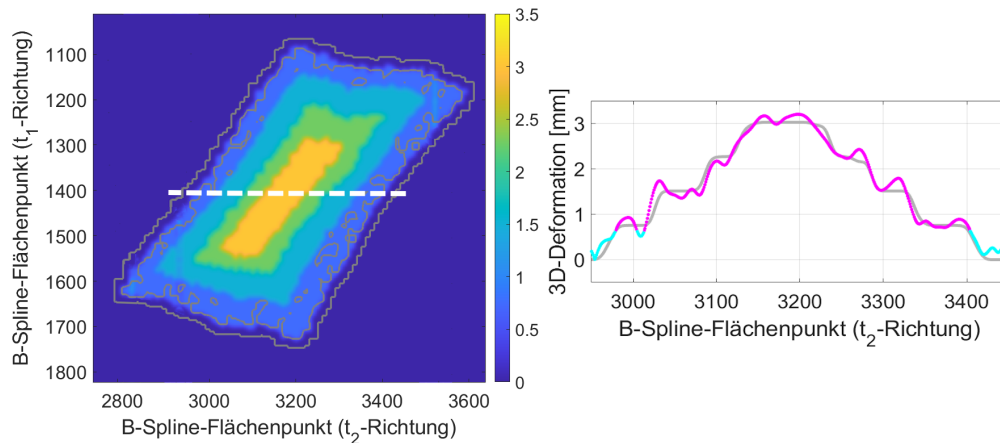


Abb. 7.63: Links: 3D-Deformation der unverrauschten Messdaten als 2D-Darstellung. Graue äußere Konturlinie: Begrenzung der Referenz-3D-Deformation, innere Konturlinie: Begrenzung der durch die DBKM signifikant festgestellten 3D-Deformation. Rechts: Längsschnitt durch 3D-Deformation. Graue Linie: Referenz, Magentafarbene Linie: signifikant festgestellte 3D-Deformation (Tontorso)

Gleichzeitig zeigen die zwei grauen Konturlinien den Unterschied zwischen Referenz-3D-Deformation und signifikant festgestellter 3D-Deformation an. Die äußere graue Konturlinie markiert den Beginn

der Referenz-3D-Deformation von korrespondierenden B-Spline-Flächenpunkten ohne Rauscheinfluss. Die innere graue Konturlinie verdeutlicht die Grenze ab der 3D-Deformationen zwischen den korrespondierenden B-Spline-Flächenpunkten verrauschter 3D-TLS-Daten nach Durchführung der DBKM erkannt werden. Ein Querschnitt entlang der weiß gestrichelten Linie in der linken Grafik, stellt den Schnitt durch die 3D-Deformation dar, die in der rechten Grafik von Abbildung 7.63 als 2D-Darstellung aufgezeigt wird. Während die graue Linie die Referenz-3D-Deformation abbildet, präsentiert die hellblaue Linie nicht signifikante Deformationen und die magentafarbene Linie signifikante Deformationen korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1. Hierbei kann festgestellt werden, dass eine signifikante 3D-Deformation ab 0,7 mm ausgewiesen wird.

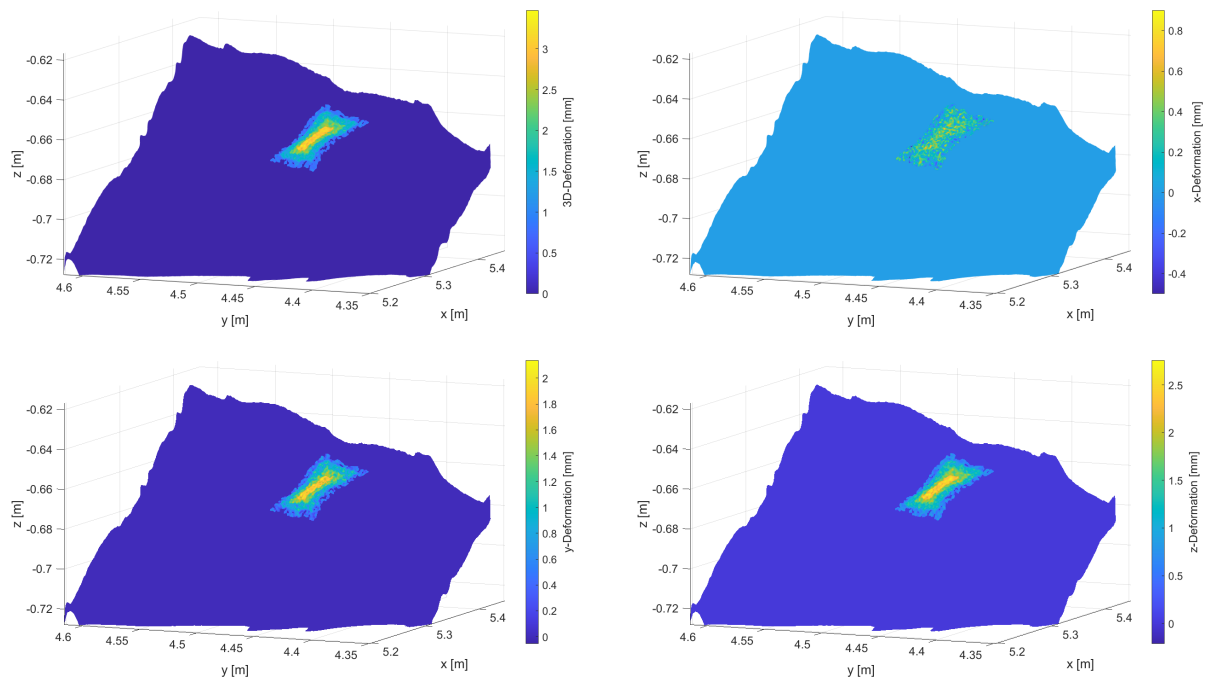


Abb. 7.64: Vergrößerter Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Tontorso). Links oben: B-Spline-Flächenpunkte, rechts oben : x-Koordinate, links unten: y-Koordinate, rechts unten: z-Koordinate.

Zur Untersuchung der in der DBKM ermittelten Deformationen hinsichtlich x-, y- und z-Koordinate sei auf die vier Grafiken in Abbildung 7.64 verwiesen. Zur besseren Darstellung der Ergebnisse werden in den Grafiken nicht signifikant deformierte Differenzen zwischen korrespondierenden B-Spline-Flächenpunkten mit dem Wert 0 mm belegt und sind deshalb einheitlich blau dargestellt. Die obere linke Grafik zeigt die 3D-Deformation korrespondierender B-Spline-Flächenpunkte bis zu maximal 3,4 mm an. Bei Betrachtung der einzelnen Koordinaten, kann bei der x-Koordinate eine Deformation von maximal 0,8 mm betrachtet werden. Die linke untere Grafik zeigt eine maximale Deformation in y-Koordinatenrichtung von 2,1 mm an und die untere rechte Grafik eine maximale Deformation in der z-Koordinate von 2,6 mm. Beim Vergleich der ermittelten Werte mit den Referenzdaten in Tabelle 7.2 ist eine Abweichung von bis zu 0,4 mm feststellbar.

7.2.4 Sandobjekt (3D-Punktwolke)

Für das Sandobjekt wird die Punktwolke von Abschnitt 7.1.4 mit 135.600 Laserscanpunkten als Epoche 0 verwendet und Epoche 1 mit einem Gaußschen Rauschen von 0,5 mm versehen. Analog zu der 3D-Punktwolke des Gurkenblatts und Tontorsos wird für Epoche 1 des Sandobjekts eine ansteigende Deformation mit maximalen Werten entsprechend Tabelle 7.2 angebracht. Die simulierte Deformation nimmt mit 2.400 Kontrollpunkten 2 % der Gesamtanzahl an TLS-Daten bzw. Kontrollpunkten in Anspruch. Abbildung 7.65 zeigt zur besseren Visualisierung die 3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und einer Referenz-Epoche 1 ohne Messrauschen.

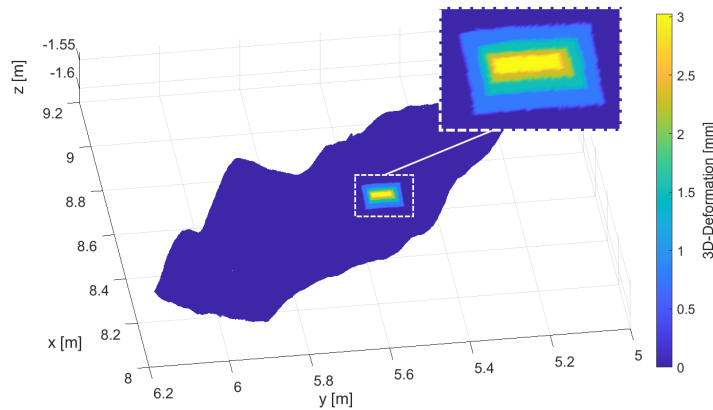


Abb. 7.65: 3D-Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Sandobjekt).

Nach Durchführung der DBKM und Anwendung des Signifikanztests können signifikant deformierte korrespondierende B-Spline-Flächenpunkte identifiziert werden. Diese werden in Abbildung 7.66 in gelb markiert dargestellt.

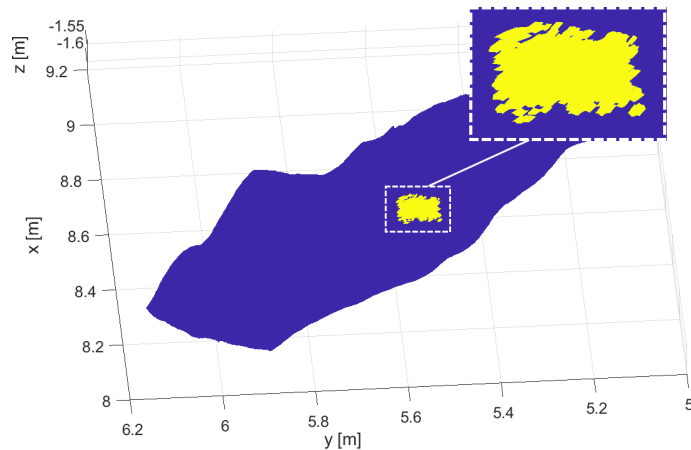


Abb. 7.66: Ergebnis des Signifikanztests (Sandobjekt).

Durch eine Gegenüberstellung der Referenz-3D-Deformation mit der mittels DBKM unter Berücksichtigung des Messrauschens bestimmten 3D-Deformation, kann eine Aussage über die feststellbaren Deformationen getätigt werden.

Die obere Grafik von Abbildung 7.67 stellt die Referenz-3D-Deformation farblich dar. Die äußere Konturlinie markiert die Begrenzung der Deformation in den Referenzdaten, während die innere graue Konturlinie die Begrenzung der feststellbaren 3D-Deformation in den Testdaten widerspiegelt. Dementsprechend können mittels DBKM alle 3D-Deformationen innerhalb der inneren grauen Konturlinie als signifikant deformiert bestimmt werden.

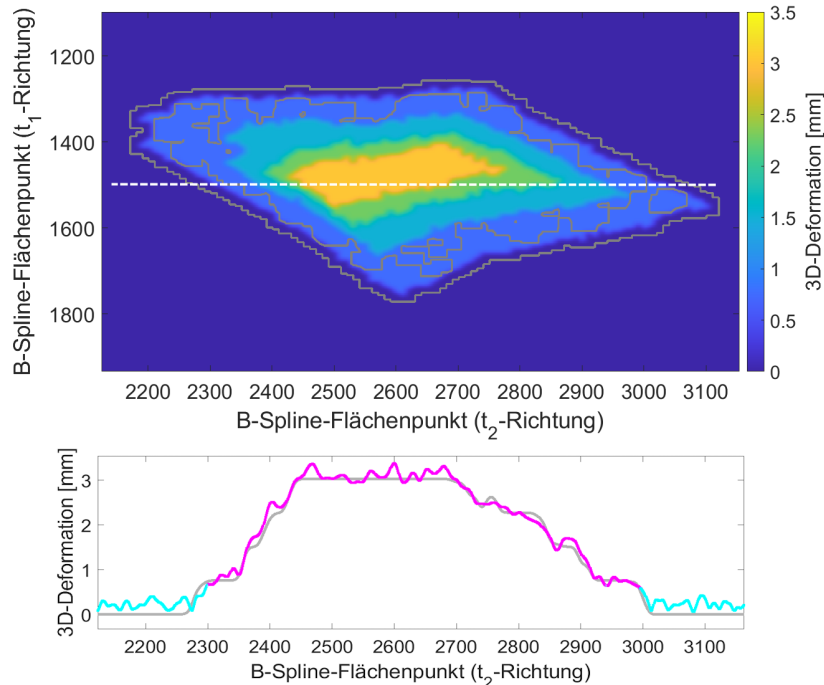


Abb. 7.67: Oben: 3D-Deformation der unverrauschten Messdaten als 2D-Darstellung. Graue äußere Konturlinie: Begrenzung der Referenz-3D-Deformation, innere Konturlinie: Begrenzung der durch die DBKM signifikant festgestellten 3D-Deformation. Unten: Längsschnitt durch 3D-Deformation. Graue Linie: Referenz, Magentafarbene Linie: signifikant festgestellte 3D-Deformation (Sandobjekt).

Die weiß gestrichelte Linie in der oberen Grafik zeigt den Verlauf der Schnittlinie, deren 2D-Darstellung in der unteren Grafik abgebildet ist. Hierbei zeigt die graue Linie die Referenz-3D-Deformation an, während die hellblaue Linie nicht signifikante und die magentafarbene Linie signifikant deformierte 3D-Deformationen darstellt. Analog zu den Ergebnissen des Gurkenblatts und Tontorsos, kann beim Sandobjekt ein Grenzwert für signifikant feststellbare 3D-Deformationen von 0,7 mm bestimmt werden. Die vier Grafiken in Abbildung 7.68 zeigen ausschließlich die als signifikant ausgewiesenen Deformationen für die korrespondierenden B-Spline-Flächenpunkte als 3D-Deformation, in der x-Koordinate, in der y-Koordinate und in der z-Koordinate. Nicht signifikant deformierte B-Spline-Flächenpunkte sind mit dem Wert 0 mm belegt. Bei dem Datensatz des Sandobjekts kann die 3D-Deformation bei einem Wert von maximal 3,2 mm abgelesen werden. Die korrespondierenden B-Spline-Flächenpunkte von Epoche 1 haben in x-Koordinatenrichtung eine maximale Deformation von 0,8 mm, in y-Koordinatenrichtung von 2,2 mm und in z-Koordinatenrichtung von 2,7 mm. Die Abweichung zur ursprünglich angebrachten Deformation beträgt somit maximal 0,4 mm.

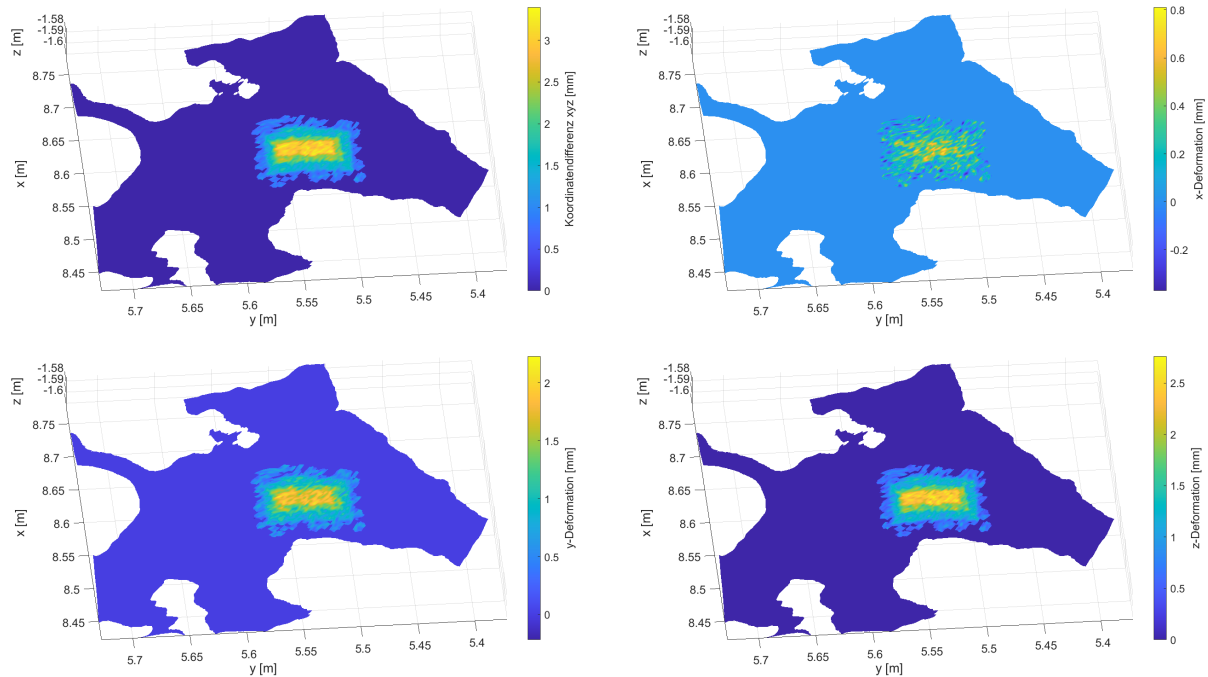


Abb. 7.68: Ausschnitt der signifikanten Deformation korrespondierender B-Spline-Flächenpunkte von Epoche 0 und Epoche 1 (Sandobjekt). Links oben: B-Spline-Flächenpunkte, rechts oben: x-Koordinate, links unten: y-Koordinate, rechts unten: z-Koordinate.

7.2.5 Diskussion der Ergebnisse

Bei der Deformationsanalyse von zugehörigen B-Spline-Flächenpunkten, die auf Basis der DBKM und simulierten Daten durchgeführt wurde, kann festgestellt werden, dass ein Signifikanztest mittels F-Test Deformationen aufdecken und eine Zuweisung hinsichtlich der Koordinatenrichtung der Deformation tätigen kann. Zu beachten ist an diesem Punkt, dass für die Durchführung der Deformationsanalyse die Punktwolke von Epoche 1 durch Anbringung von Gaußschem Rauschen von 0,5 mm und einer mittig anwachsenden Deformation bis maximal 3,0 mm simuliert wurde. Bei dem untersuchten 2D-Datensatz des Fassadenstücks mit insgesamt 261 Kontrollpunkten kann festgestellt werden, dass 2D-Deformationen kleiner 0,9 mm zwischen korrespondierenden B-Spline-Kurvenpunkten im Signifikanztest nicht als signifikant ausgewiesen werden. Ab dem Grenzwert von 0,9 mm können bei dem verwendeten 2D-Datensatz signifikante Deformationen ermittelt werden. Bei den untersuchten drei 3D-Datensätzen Gurkenblatt, Tontorso und Sandobjekt mit einer Kontrollpunktanzahl zwischen 15.000 und 150.000 kann festgestellt werden, dass 3D-Deformationen ab 0,7 mm als signifikant eingestuft werden. Die Diskrepanz zwischen den ermittelten Grenzwerten von 0,9 mm im 2D-Fall und 0,7 mm im 3D-Fall ist bei zukünftigen umfassenderen Untersuchungen näher zu beleuchten und wurde in dieser Arbeit, aufgrund der größeren Gewichtung der Evaluation der DBKM hinsichtlich 3D-Datensätzen, nicht weiter verfolgt. Eine mögliche Ursache kann der im Vergleich zu den 3D-Datensätzen sehr kleine 2D-Datensatz mit lediglich 261 Punkten sein. Eine andere Erklärung kann in der höheren Standardabweichung von B-Spline-Kurvenpunkten im Vergleich zu B-Spline-Flächenpunkten liegen, die nicht am Rand der Punktwolke positioniert sind. Da B-Spline-Kurvenpunkte entsprechend Tabelle 5.1 von Unterkapitel 5.1 eine Standardabweichung von etwa 0,34 mm im mittleren Kurvenbereich enthalten, ist die Standardabweichung von B-Spline-Kurvenpunkten etwa 0,11 mm höher als die Standardabweichung von B-Spline-Flächenpunkten mit etwa 0,23 mm (vergleiche Abbildung 5.8, Unterkapitel 5.3). Diese Aussage kann getätigt werden, da für die Untersuchung von Deformationen

der 3D-Datensätze ausschließlich Deformationen im inneren der Punktwolke und nicht im Randbereich gewählt wurden. Um allerdings hierzu eine präzisere Aussage treffen zu können, ist eine Eingrenzung und Beleuchtung des Grenzwerts für den 2D-Fall in zukünftigen Untersuchungen an 2D-Datensätzen mit größerem Umfang geplant. Abgesehen von dem ermittelten Grenzwert, ab dem Deformationen als signifikant ausgegeben werden können, haben sowohl 2D- als auch 3D-Datensätze gezeigt, dass die angebrachte Deformation mittels der DBKM mit einer maximalen Abweichung von 0,4 mm bestimmt werden kann. Die ermittelten Deformationen lassen sich ebenso durch die Zuweisung korrespondierender B-Spline-Kurven- bzw. Flächenpunkte in x-, y- und z-Deformation spezifizieren, wodurch genauere Aussagen getätigt werden können, in welche Richtung sich eine Deformation entwickelt. Für den 2D-Datensatz wurde eine 2D-Deformation mit einer Kontrollpunktanzahl von 8 % der Gesamtanzahl angebracht. Die drei 3D-Datensätze wurden mit einer 3D-Deformation mit einer Kontrollpunktanzahl von 2 % der Gesamtanzahl versehen. Bei zukünftigen Analysen kann der Prozentanteil der Deformation und dessen Auswirkung auf die Feststellbarkeit von Deformationen ebenso intensiver untersucht werden.

Die Methodik der DBKM in Verbindung mit der Deformationsanalyse enthält somit zwar eine begrenzende Eigenschaft durch die Höhe des Messrauschens, allerdings bietet sie den ausgeprägten Vorteil einer automatischen Feinregistrierung zweier Punktwolken mittels B-Spline-Modellierung und Mapcut-Algorithmus und kann in Berücksichtigung stochastischer Bedingungen signifikant deformierte Bereiche ausweisen. Ein großer Vorteil der DBKM liegt außerdem darin, dass, abgesehen von der Analyse signifikanter Deformationen, gleichzeitig eine Richtungsangabe der Deformation hinsichtlich x-, y- und z-Koordinate getätigt werden kann.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Die in dieser Arbeit entwickelte DBKM stellt einen neuartigen Ansatz vor, um in TLS-Daten zweier Epochen Deformationen zu analysieren. Die DBKM besteht aus drei Hauptkomponenten. Hierzu zählen eine Modellierung der TLS-Daten mittels B-Spline-Kurven bzw. Tensorprodukt B-Spline-Flächen, ein lokales Zielzeichen-freies Registrierungsverfahren unter Nutzung der Krümmung als Merkmalsdeskriptor namens Mapcut-Algorithmus und eine stochastische Modellierung der Prozesskette. Die Ergebnisse der DBKM ermöglichen mittels Signifikanztest eine Deformationsanalyse.

In Kapitel 2 werden die Grundlagen zu B-Spline-Kurven und Tensorprodukt B-Spline-Flächen erläutert und auf die besonders vorteilhaften Eigenschaften der B-Spline Technik zur Modellierung von Punktwolken eingegangen. Drei der wichtigsten Eigenschaften stellen die Lokalität, die Anfangs- und Endpunkt-Eigenschaft sowie die affine Invarianz dar. Im Speziellen wird für B-Spline-Kurven bzw. -Flächen die Bestimmung der Krümmung bzw. eines speziellen Krümmungsparameters vorgestellt, der als Merkmalsdeskriptor in der DBKM verwendet wird. Kapitel 3 ordnet die entwickelte Methode in einen wissenschaftlichen Kontext ein. Hierbei werden aktuelle Ansätze zur Modellierung von TLS-Daten mittels B-Splines vorgestellt, die überwiegend Approximationsverfahren mittels Kleinste-Quadrate-Schätzung einsetzen, um B-Spline bezogene Parameter zu bestimmen. Diese Parameterschätzungen enthalten oftmals ein hohes Maß an Komplexität und die Güte der Parameterdefinition beeinflusst maßgebend wie gut die geschätzte B-Spline-Kurve bzw. -Fläche die reale Objektform wiedergibt. Aufgrund dessen wird in dieser Arbeit ein direkter funktionaler Approximationsansatz verfolgt, der keine Parameterschätzung benötigt, sondern die B-Spline-Kurve bzw. Tensorprodukt B-Spline-Fläche direkt aus den TLS-Daten modelliert. Da bei TLS keine Erfassung von Punktkorrespondenzen zwischen Epochen im Messvorgang erfolgt, sind TLS-Daten vor Durchführung einer Deformationsanalyse immer zu registrieren. Hierzu wird auf bestehende Ansätze zu geometrie- und merkmalsbasierten Registrierungsverfahren eingegangen. Das in dieser Arbeit entwickelte lokale Zielzeichen-freie Registrierungsverfahren soll auf Basis eines Merkmalsdeskriptors eine Feinregistrierung durchführen und dabei ausschließlich stabile Bereiche berücksichtigen.

Kapitel 4 stellt die entwickelte DBKM vor. Damit diese angewendet werden kann, sind einige Rahmenbedingungen betreffend der TLS-Daten und des Messaufbaus zu erfüllen. Die Messungen sollten von einem annähernd gleichen Standpunkt mit ähnlichem Messaufbau und gleichen Messparametern erfolgen. Die zu erfassende Deformation sollte in Relation zur Objektgröße klein sein. Die TLS-Daten sind in Scanreihenfolge sortiert auszugeben, sollten keine Lücken oder Versprünge enthalten, eine Punktzahl bis 150.000 Punkte und einen mittleren Punktabstand größer als das Messrauschen enthalten, um Überschneidungen der Scanlinien zu vermeiden. Als Initialwerte werden ein Kurvengrad der B-Spline-Basisfunktionen von $n = 3$ und ein offener uniformer Knotenvektor verwendet. Eine der drei Kernelemente der DBKM ist die Modellierung der TLS-Daten mittels Tensorprodukt B-Spline-Flächen. Grundvoraussetzung für Tensorprodukt B-Spline-Flächen bildet eine bidirektionale Kontrollpunktgitterstruktur. Um diese zu generieren, werden aus den TLS-Daten einzelne Kontrollpolygone definiert und anschließend mit einem Knoteneinfügealgorithmus in jedes Kontrollpolygon so viele Kontrollpunkte eingefügt, bis jedes Kontrollpolygon aus der gleichen Anzahl an Kontrollpunkten besteht. Auf Basis dieses Kontrollgitters können Tensorprodukt B-Spline-Flächen generiert sowie für jeden B-Spline-Flächenpunkt ein Krümmungsparameter extrahiert werden. Diese Krümmungsparameter dienen als Merkmalsdeskriptoren, um das entwickelte Registrierungsverfahren, den Mapcut-Algorithmus, anzuwenden. Der Mapcut-Algorithmus ist ein iteratives vierstufiges Verfahren, das bisher zur Feinregistrierung von B-Spline-Kurven und -Flächen implementiert vorliegt. Eine Erweiterung zur Grobregistrierung

ist infolge der invarianten Eigenschaften des Krümmungsparameters möglich. Die ersten zwei Stufen des Mapcut-Algorithmus bestehen aus einem Grobmapping und Feinmapping. Hierbei wird für die Registrierung zweier B-Spline-Kurven bzw. zweier Tensorprodukt B-Spline-Flächen eine Korrelationsanalyse auf Basis der Krümmungsparameter durchgeführt. In Stufe III des Mapcut-Algorithmus werden identische Punkte in stabilen Bereichen selektiert, um Transformationsparameter für eine Koordinatentransformation der B-Spline-Kurven bzw. -Flächenpunkte und deren Kontrollpunkte von Epoche 1 in Epoche 0 zu transformieren. Stufe IV bildet im Anschluss ein Cutting, das aus einer Anpassung des Kontrollpolygons bzw. Kontrollgitters von Epoche 0 zu Epoche 1 besteht. Durch das Cutting werden negative Auswirkungen unterschiedlicher Randabgrenzungen der TLS-Daten wesentlich verringert. Mittels einer iterativen Vorgehensweise werden die Koordinatendifferenzen korrespondierender Kurven- bzw. Flächenpunkte zunehmend verringert.

Ergänzt wird die Prozesskette der DBKM durch eine stochastische Modellierung (Kapitel 5). Auf diese Weise findet eine Berücksichtigung der Unsicherheiten in der Punktwolke statt. Als initiale Annahme wird ein Gaußsches Rauschen von 0,5 mm vorgegeben und eine unkorrelierte Varianz-Kovarianzmatrix aufgestellt. Sowohl für den 2D-Fall als auch den 3D-Fall findet je nach Berechnungsschritt der DBKM eine Varianz-Kovarianzfortpflanzung unter anderem auch mit überbestimmten Ansätzen (Gauß-Helmert-Modell) statt. Das Ergebnis ist eine Kofaktormatrix der B-Spline-Kurven- bzw. Flächenpunkte zweier zu vergleichender Epochen. Aufgrund der Datenmenge wird für den 3D-Fall eine simulierte Kofaktormatrix auf Datengrundlage kleinerer Punktwolken erstellt. Ein Signifikanztest in Form eines F-Tests, der Differenzvektor und Kofaktormatrix einbezieht, bietet die Grundlage zur Durchführung einer Deformationsanalyse.

Um die Funktionalität der entwickelten DBKM untersuchen zu können, werden vier Evaluierungsdatensätze herangezogen (Kapitel 6). Datensatz 1 stellt eine simulierte 2D-Punktwolke eines Fassadenstücks mit etwa 300 Punkten dar. Datensatz 2 bildet eine simulierte 3D-Punktwolke eines Testobjekts in Form eines Gurkenblatts mit insgesamt etwa 15.000 Punkten. Der dritte Datensatz einer realen 3D-Punktwolke wurde durch eine Messung eines Testobjekts in Form eines Tontorsos erzeugt und enthält etwa 150.000 Punkte. Datensatz 4 ist ebenso eine reale 3D-Punktwolke mit etwa 135.000 Punkten, die ein Sandobjekt präsentiert.

In Kapitel 7 wird die DBKM und die Deformationsanalyse an den vier Evaluierungsdatensätzen angewendet. Für die beiden simulierten Datensätze kann die Modellierung der TLS-Daten zu B-Spline-Kurven bzw. Tensorprodukt B-Spline-Flächen und der anschließende Mapcut-Algorithmus erfolgreich durchgeführt werden. Die korrespondierenden B-Spline-Kurven- bzw. -Flächenpunkte können derartig zugeordnet und in ein gemeinsames Koordinatensystem transformiert werden, dass lediglich der Einfluss des Messrauschens zum Tragen kommt. Um die Funktionalität und den Entwicklungsstand der DBKM für reale Datensätze abzufragen, wird die Modellierung der TLS-Daten zu Tensorprodukt B-Spline-Flächen und der Mapcut-Algorithmus für die zwei Realdaten des Tontorsos und Sandobjekts angewendet. Hierbei kann festgestellt werden, dass die DBKM erfolgreich auf Realdaten übertragbar ist, da sowohl die Modellierung zu Tensorprodukt B-Spline-Flächen als auch die Korrelationsanalyse mittels Krümmung, die Zuordnung von Punktkorrespondenzen, die Transformation und das Cutting erfolgreich sind. Bei der Analyse der Euklidischen Distanzen zwischen korrespondierenden B-Spline-Flächenpunkten fällt auf, dass zwar überwiegend Euklidische Distanzen von etwa 1,1 mm vorliegen, allerdings in Bereichen von Vertiefungen, Erhebungen oder Kanten der Objektoberfläche Euklidische Distanzen bis zu 2,5 cm auftreten. Dieses Verhalten hängt mit den unterschiedlichen Punktabständen der TLS-Punkte im Bereich von Erhebungen zusammen. In flachen Bereichen sind diese gleichmäßig, während sie bei Erhebungen oder Vertiefungen ein unregelmäßiges Muster annehmen. Dies bewirkt eine starke Diskrepanz der Kontrollgitter und B-Spline-Flächen beider Epochen innerhalb der exponierten Bereiche. Aufgrund der teilweise großen Euklidischen Distanzen zwischen korrespondierenden Flächenpunkten bei den Realdaten, werden für die Anwendung der Deformationsanalyse aus den Daten von Epoche 0 simulierte Daten der Epoche 1 erzeugt, die mit einer Deformation versehen werden.

Hierbei wird die Deformationsanalyse jeweils mit einer ansteigenden Deformation bis maximal 3,0 mm an den vier Datensätzen getestet. Die simulierten Deformationen bei den 3D-Punktwolken betreffen etwa 2 % der Gesamtanzahl an Punkten der jeweiligen 3D-Punktwolke. Es kann festgestellt werden, dass bei dem 2D-Datensatz mit geringer Kontrollpunktanzahl Deformationen ab 0,9 mm signifikant festgestellt werden können. Bei den drei 3D-Datensätzen können 3D-Deformationen ab bereits 0,7 mm bestimmt werden. Unterhalb von 0,7 mm scheint die Beeinflussung durch das Messrauschen dementsprechend zu ausgeprägt zu sein, als dass signifikante Deformationen ermittelbar sind. Sowohl bei dem 2D-Datensatz als auch den drei 3D-Datensätzen kann die angebrachte Deformation mit einer maximalen Abweichung von 0,4 mm bestimmt werden. Eine Beurteilung der resultierenden maximalen Abweichung in Zusammenhang mit dem angebrachten Messrauschen ist zu beachten.

Zwar besteht in dem Ansatz der DBKM in Verbindung mit der Deformationsanalyse ein limitierender Faktor durch die Höhe des Messrauschens, jedoch bietet die entwickelte Methode den erheblichen Vorteil einer automatischen Zielzeichen-freien Registrierung zweier Punktwolken mittels B-Spline-Modellierung und Mapcut-Algorithmus und kann in Berücksichtigung stochastischer Bedingungen signifikant deformierte Bereiche eindeutig identifizieren. Der Vorteil signifikante Deformationen korrespondierender B-Spline-Flächenpunkte analysieren zu können und gleichzeitig eine Annahme über die Richtung der Deformation tätigen zu können, macht die DBKM zu einem vielversprechenden Ansatz für weitere zukünftige Entwicklungen.

8.2 Ausblick

Ein Anknüpfungspunkt für zukünftige Untersuchungen besteht im Umgang mit TLS-Daten realer Punktwolken, die Erhebungen, Vertiefungen oder Kanten aufweisen und deshalb in diesen Bereichen keiner gleichmäßigen Punktverteilung folgen. Diese Unregelmäßigkeiten führen bei der Generierung von Tensorprodukt B-Spline-Flächen bisher zu deutlichen Differenzen zwischen den zu vergleichenden Epochen. Um dieses Verhalten abzumildern, könnte man die Punktwolken in exponierten Bereichen mit zusätzlichen Punkten verdichten, um auf diese Weise gleichmäßigere Punktabstände zu erzeugen. Inwiefern eine Refinement-Strategie angewendet werden kann, ist somit zu untersuchen. Ähnliche Schlussfolgerungen machen Harmening und Butzer (2024), die approximierte Tensorprodukt B-Spline-Flächen mittels Parameterschätzung erfolgreich für künstliche Objekte einsetzen, allerdings bei natürlichen Oberflächen mit vielen lokalen Strukturen Weiterentwicklungspotentiale sehen. Hierfür sollen lokale Parametrisierungen als Ergänzung zu den modellierten B-Spline-Flächen eingearbeitet werden.

Für zukünftige weitere Untersuchungen und Entwicklungen könnte die stochastische Modellierung der DBKM realitätsnaher aufgebaut werden, indem statt einer unkorrelierten Varianz-Kovarianzmatrix als Initialwert eine korrelierte Varianz-Kovarianzmatrix entsprechend Kerekes und Schwiager (2025) eingeführt wird.

Abgesehen von der Möglichkeit eine Verbesserung der Methode durch eine vollbesetzte Varianz-Kovarianzmatrix zu erzielen, ist zu beachten, dass durch die direkte funktionale Approximation der DBKM eine unmittelbare Einbeziehung der Höhe des Messrauschens in die stochastische Modellierung stattfindet. Die Höhe des Messrauschens beeinflusst somit maßgebend die Parameter der resultierenden Kofaktormatrix, die im Signifikanztest eingesetzt werden. Demzufolge korreliert die Qualität der Deformationsanalyse der DBKM in direkter Beziehung mit der Genauigkeit des terrestrischen Laserscanners. Die stetige Verbesserung des TLS kommt der DBKM somit zugute und wird zukünftig zu einer automatischen Genauigkeitssteigerung erfassbarer Deformationen führen.

Um die Eigenschaften und Einsatzfähigkeit der DBKM weiter zu präzisieren, sollen bei künftigen Analysen ebenfalls 2D-Datensätze mit einer größeren Punktzahl untersucht werden. Außerdem soll

geprüft werden, wie groß der prozentuale Anteil deformierter Bereiche in den 2D- und 3D-Punktwolken maximal sein darf, damit die DBKM noch funktionsfähig ist.

Aufgrund des Potentials der entwickelten DBKM 2D- und 3D-Deformationen korrespondierender B-Spline-Kurven- und Flächenpunkte bestimmen zu können, soll die DBKM für Realdaten weiter optimiert werden. Hierzu könnten lokale Punktverdichtungen von 3D-Punktwolken vielversprechende Ansätze liefern.

Literatur

- Abbas, M. A., Setan, H., Majid, Z., Chong, A. K., Lichti, D. D., Idris, K. M. (2014): Improvement in Accuracy Through Self-Calibration for Panoramic Scanner. In: Rahman, A. A.; Boguslawski, P.; Anton, F.; Said, M. N.; Omar, K. M. [Hrsg], *Geoinformation for Informed Decisions*. Springer, S. 201–217.
- Abramowski, S., Müller, H. (1991): *Geometrisches Modellieren*. Bd. 75. Reihe Informatik. BI-Wissenschaftsverl., Mannheim.
- Aichinger, J., Schwieger, V. (2022): Studies on deformation analysis of TLS point clouds using B-splines - A control point based approach (Part I). *Journal of Applied Geodesy*, Jahrgang16, Heft3, S. 279–298.
- Aiger, D., Mitra, N., Cohen-Or, D. (2008): 4-Points Congruent Sets for Robust Pairwise Surface Registration. *ACM Transactions on Graphics*, Jahrgang27, Heft3, 85.
- Akca, D. (2003): Full automatic registration of laser scanner point clouds. In: Gruen, A.; Kahmen, H. [Hrsg.]: *Optical 3-D Measurement Techniques VI*, 22. - 25. September 2003, Zürich, Schweiz, Jahrgang1, S. 330–337.
- Aspert, N., Santa-Cruz, D., Ebrahimi, T. (2002): MESH: Measuring Errors between Surfaces Using the Hausdorff Distance. *Proceedings of the IEEE International Conference in Multimedia and Expo (ICME)*, vol.1, 26.-29. August 2002, Lausanne, Schweiz, Jahrgang1, S. 705–708.
- Bae, K.-H., Lichti, D. D. (2008): A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, Jahrgang63, Heft1, S. 36–54.
- Barlow, R. (1999): *Statistics: A guide to the use of statistical methods in the physical sciences*. 5. The Manchester physics series. Wiley, Chichester.
- Bender, M., Bill, M. (2006): *Computergrafik – Ein anwendungsorientiertes Lehrbuch*. 2. Aufl., Carl Hanser Verlag, München, Wien.
- Bernhardt, J. C. (2014): *Das Nikemonument von Samothrake und der Kampf der Bilder*. Franz Steiner Verlag, Stuttgart.
- Besl, P. J., Mc Kay, N. D. (1992): A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jahrgang14, Heft2, S. 239–256.
- Boehm, W. (1980): Inserting new knots into B-spline curves. *Computer Aided Design*, Jahrgang12, Heft4, S. 199–201.
- Boehm, W. (1985): On the efficiency of knot insertion algorithms. *Computer Aided Geometric Design*, Jahrgang2, Heft1–3, S. 141–143.
- Boergens, E., Schmidt, M., Seitz, F. (2021): The use of B-splines to represent the topography of river networks. *Water Resources Research*, Jahrgang12, Heft21.
- Bureick, J. (2020): *Robuste Approximation von Laserscan-Profilen mit B-Spline-Kurven*. Deutsche Geodätische Kommission, Reihe C (Dissertationen), Heft852, München.
- Carmo, M. P. d. (1993): *Differentialgeometrie von Kurven und Flächen*. Bd. 55. 3., durchgesehenes Vieweg-Studium Aufbaukurs Mathematik. Vieweg, Braunschweig.
- Censi, A. (2008): An ICP variant using a point-to-line metric. *Proceedings of the IEEE International Conference on Robotics and Automation*, 19. - 23. Mai 2008, Pasadena, Californien, USA, S. 19–25.
- Chen, S., Nan, L., Renbo, X., Jibin, Z., Wonka, P. (2020): PLADE: A Plane-based Descriptor for Point Cloud Registration with Small Overlap. *IEEE Transactions on Geoscience and Remote Sensing*, Jahrgang58, Heft4, S. 2530–2540.
- Chen, Y., Medioni, G. (1992): Object modelling by registration of multiple range images. *Image and Vision Computing*, Jahrgang10, Heft3, S. 145–155. *Range Image Understanding*.
- Cignoni, P., Rocchini, C., Scopigno, R. (1998): METRO: Measuring error on simplified surfaces.

- Computer Graphics Forum, Jahrgang17, S. 167–174.
- Cohen, E., Lyche, T., Riesenfeld, R. F. (1980): Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer Graphics and Image Processing*, Jahrgang14, Heft2, S. 87–111.
- Cox, M. G. (1972): The numerical evaluation of B-splines. *Journal of the Institute of Mathematics and its Applications*, Jahrgang10, S. 134–149.
- Curry, H. B., Schoenberg, I. J. (1947): On spline distributions and their limits: the Pölya distribution functions. *Bulletin of the American Mathematical Society*, Jahrgang53, S. 109, Abstract 380t.
- de Boor, C. (1972): On calculating with B-splines. *Journal of Approximation Theory*, Jahrgang6, Heft1, S. 50–62.
- de Boor, C. (1978): *A Practical Guide to Splines*. Bd. 27. *Applied Mathematical Sciences*. Springer, New York.
- de Boor, C., Höllig, K. (1985): *B-Splines without Divided Differences*. University of Wisconsin-Madison, Mathematics Research Center, Madison, Wisconsin.
- de Boor, C. (2001): *A Practical Guide to Splines: With 32 figures*. Bd. 27, überarb. Auflage. *Applied Mathematical Sciences*. Springer, New York.
- Dong, Z., Liang, F.; Yang, B., Xu, Y., Zang, Y., Li, J., Wang, Y., Dai, W., Fan, H., Hyypä, J., Stilla, U. (2020): Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, Jahrgang163, S. 327–342.
- Eling, D. (2009): *Terrestrisches Laserscanning für die Bauwerksüberwachung*. Deutsche Geodätische Kommission, Reihe C (Dissertationen), Heft641, München.
- Epstein, M. P. (1976): On the Influence of Parameterization in Parametric Interpolation. *SIAM Journal on Numerical Analysis*, Jahrgang13, Heft2, S. 261–268.
- Eschenburg, J.-H., Jost, J. (2007): *Differentialgeometrie und Minimalflächen*. 2., Springer-Verlag, Berlin, Heidelberg.
- Fang, J.-J., Hung, C.-L. (2013): An improved parameterization method for B-spline curve and surface interpolation. *Computer-Aided Design*, Jahrgang45, Heft6, S. 1005–1028.
- Farin, G. (1994): *Kurven und Flächen im Computer Aided Geometric Design: Eine praktische Einführung*. Vieweg Lehrbuch. *Angewandte Mathematik*. Vieweg, Braunschweig.
- Farin, G. (2002): *Curves and surfaces for CAGD: A practical guide*. 5., Morgan Kaufmann Publishers, San Francisco.
- Fischler, M. A., Bolles, R. C. (1981): Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the Association for Computing Machinery*, Jahrgang24, Heft6, S. 381–395.
- Foley, T. A., Nielson, G. M. (1989): Knot Selection for Parametric Spline Interpolation. In: Lyche, T. und Schumaker, L. L. [Hrsg.], *Mathematical Methods in Computer Aided Geometric Design*. Academic Press, S. 261–271.
- Franaszek, M., Cheok, G. S., Witzgall, C. (2009): Fast automatic registration of range images from 3D imaging systems using sphere targets. *Automation in Construction*, Jahrgang18, Heft3, S. 265–274.
- Friedli, E. (2020): *Point Cloud Registration and Mitigation of Refraction Effects for Geomonitoring Using Long-Range Terrestrial Laser Scanning*. Dissertation. ETH Zürich..
- Ge, X., Wunderlich, T. (2016): Surface-based matching of 3D point clouds with variable coordinates in source and target system. *ISPRS Journal of Photogrammetry and Remote Sensing*, Jahrgang111, S. 1–12.
- Girardeau-Montaut, D., Roux, M., Marc, R., Thibault, G. (2005): Change detection on point cloud data acquired with a ground laser scanner. *ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005"*, 12. - 14. September 2005, Enschede, Niederlande, S. 30–35.

- Gojcic, Z., Schmid, L., Wieser, A. (2021): Dense 3D displacement vector fields for point cloud-based landslide monitoring. *Landslides*, Jahrgang18, S. 3821–3832.
- Goldman, R. N. (1990): Blossoming and knot insertion algorithms for B-spline curves. *Computer-Aided Geometric Design*, Jahrgang7, Heft1, S. 69–81.
- Gordon, S., Lichti, D., Franke, J., Stewart, M. (2004): Structural deformation measurements using terrestrial laser scanners. *Proceedings of the 1st FIG Symposium on Engineering Surveys for Construction Works and Structural Engineering*, 28. Juni - 1. Juli 2004, Nottingham, United Kingdom.
- Gordon, S., Lichti, D., Stewart, M., Franke, J. (2003): Structural deformation measurements using terrestrial laser scanners. *Proceedings of the 11th FIG Symposium on Deformation Measurements*, Santorin, Griechenland.
- Gordon, W. J., Riesenfeld, R. F. (1974): Bernstein-Bézier methods for the computer-aided design of free-form curves and surfaces. *Journal of the ACM*, Jahrgang21, S. 293–310.
- Handels, H. (2009): Medizinische Bildverarbeitung: Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie. 2., überarb. und erw., Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden.
- Harmening, C. (2013): Raum-zeitliche Segmentierung von natürlichen Objekten in stark verdeckten Szenen. Unveröffentlichte Masterarbeit, Leibniz Universität Hannover, Hannover.
- Harmening, C. (2020): Spatio-temporal deformation analysis using enhanced B-spline models of laser scanning point clouds. Dissertation, E120-05 Department für Geodäsie und Geoinformation, TU Wien.
- Harmening, C., Butzer, R. (2024): Improving the approximation quality of tensor product B-spline surfaces by local parameterization. *Journal of Applied Geodesy*, Jahrgang18, S. 575–596.
- Harmening, C., Hobmaier, C., Neuner, H. (2021): Laser Scanner-Based Deformation Analysis Using Approximating B-Spline Surfaces. *Remote Sensing*, Jahrgang13 (18), Heft3551.
- Harmening, C., Neuner, H. (2020): A spatio-temporal deformation model for laser scanning point clouds. *Journal of Geodesy*, Jahrgang94, Heft5, S. 1–15.
- Harmening, C., Neuner, H.-B. (2014): Raumkontinuierliche Modellierung mit Freiformflächen. In: *Terrestrisches Laserscanning 2014* (11. - 12. Dezember 2014, Fulda). Bd. 78. Wißner-Verlag, Augsburg, S. 105–122.
- Harmening, C., Neuner, H.-B. (2015): A constraint-based parameterization technique for B-spline surfaces. *Journal of Applied Geodesy*, Jahrgang9, Heft3, S. 143–161.
- Harmening, C., Neuner, H.-B. (2016): Detecting rigid body movements from TLS-based areal deformation measurements. *Talk 78th FIG Working Week 2016*, Christchurch, Neuseeland, 2016-05-02 – 2016-05-06. In: *Proceedings of 78th FIG Working Week 2016*, S. 1–10.
- Hartung, J., Elpelt, B., Klösener, K.-H. (2002): Statistik: Lehr- und Handbuch der angewandten Statistik. 13., De Gruyter Oldenbourg, München.
- Hausdorff, F. (1935): Mengenlehre. 3. Aufl., Walter de Gruyter, Berlin.
- Hebel, M., Stilla, U. (2012): Simultaneous Calibration of ALS Systems and Alignment of Multi-view LiDAR Scans of Urban Areas. *IEEE Transactions on Geoscience and Remote Sensing*, Jahrgang50, Heft6, S. 2364–2379.
- Heunecke, O., Kuhlmann, H., Welsch, W., Eichhorn, A., Neuner, H.-B. (2013): Auswertung geodätischer Überwachungsmessungen. 2., neu bearb. und erw. Handbuch Ingenieurgeodäsie. Wichmann, Berlin und Offenbach.
- Holst, C., Schmitz, B., Schraven, A., Kuhlmann, H. (2017): Eignen sich in Standardsoftware implementierte Punktwolkenvergleiche zur flächenhaften Deformationsanalyse von Bauwerken? Eine Fallstudie anhand von Laserscans einer Holzplatte und einer Staumauer. In: *zfv. (Zeit-*

- schrift für Geodäsie, Geoinformation und Landmanagement). Jahrgang142, Heft2, S. 98–110.
- Holst, C., Kuhlmann, H. (2016): Challenges and Present Fields of Action at Laser Scanner Based Deformation Analyses. *Journal of Applied Geodesy*, Jahrgang10, Heft1, S. 17–25.
- Holst, C., Zeimet, P., Nothnagel, A., Schauerte, W., Kuhlmann, H. (2012): Estimation of Focal Length Variations of a 100-m Radio Telescope's Main Reflector by Laser Scanner Measurements. *Journal of Surveying Engineering*, Jahrgang138, S. 126–135.
- Hoschek, J., Lasser, D. (1992): Grundlagen der geometrischen Datenverarbeitung. 2., neu bearb. und erw., Vieweg+Teubner Verlag, Wiesbaden.
- Huang, R., Xu, Y., Stilla, U. (2021): GraNet: Global relation-aware attentional network for semantic segmentation of ALS point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, Jahrgang177, S. 1–20.
- Jäger, R., Müller, T., Saler, H., Schwäble, R. (2005): Klassische und robuste Ausgleichungsverfahren: Ein Leitfaden für Ausbildung und Praxis von Geodäten und Geoinformatikern. Wichmann, Heidelberg.
- Kahmen, H. (2006): Angewandte Geodäsie: Vermessungskunde. 20., neu bearb. De Gruyter Lehrbuch. De Gruyter, Berlin.
- Kerekes, G., Schwieger, V. (2025): Correlations in TLS point clouds: Should we care about them?. 6th Joint International Symposium on Deformation Monitoring (JISDM), 7. bis 9. April 2025, Karlsruhe.
- Kermarrec, G., Kargoll, B., Alkhatib, H. (2020): Deformation Analysis Using B-Spline Surface with Correlated Terrestrial Laser Scanner Observations — A Bridge Under Load. *Remote Sensing*, Jahrgang12, Heft5, 829.
- Kersten, T. P., Stange, M., Lindstaedt, M., Melchke, K. (2020): Geometrische Genauigkeitsuntersuchungen der terrestrischen Laserscanner Leica RTC360 und Z+F Imager 5016 im Labor und im Feldprüfverfahren. In: Luhmann, T.; Schuhmacher, C. [Hrsg.], *Photogrammetrie, Laserscanning, Optische 3D-Messtechnik - Beiträge der Oldenburger 3D-Tage 2020*. Wichmann, Berlin, S. 2–13.
- Koch, K. R. (2010): NURBS Surface with Changing Shape. *Allgemeine Vermessungs-Nachrichten*, Jahrgang117, Heft3, S. 83–89.
- Kuhlmann, H., Schwieger, V., Wieser, A., Niemeier, W. (2014): Engineering Geodesy - Definition and Core Competencies. *Journal of Applied Geodesy*, Jahrgang8, Heft4, S. 327–334.
- Kühnel, W. (2010): Differentialgeometrie: Kurven - Flächen - Mannigfaltigkeiten. 5., aktual. Aufbaukurs Mathematik. Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden.
- Lague, D., Brodu, N., Leroux, J. (2013): Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z). *ISPRS Journal of Photogrammetry and Remote Sensing*, Jahrgang82, S. 10–26.
- Lancaster, P., Salkauskas, K. (1990): Curve and Surface Fitting. An Introduction. 3., Academic Press, London.
- Lee, E. (1985): Some remarks concerning B-splines. *Computer Aided Geometric Design*, Jahrgang2, Heft4, S. 307–311.
- Lee, E. (1989a): Choosing nodes in parametric curve interpolation. *Computer Aided Geometric Design*, Jahrgang21, Heft6, S. 363–370.
- Lee, E. (1989b): Yet another proof of knot insertion. *Computer Aided Geometric Design*, Jahrgang6, Heft1, S. 83–84.
- Leica Geosystems (2011): HDS7000 Laser Scanner. Leica Geosystems AG, Heerbrugg.
- Lindenbergh, R., Pfeifer, N. (2005): A statistical deformation analysis of two epochs of terrestrial laser data of a lock. *Proceedings of the 7th Conference on Optical 3-D Measurement Techniques*, Wien, Österreich, Jahrgang2, S. 61–70.
- Little, M. J. (2006): Slope monitoring at PPRust open pit operation. In: *Proceedings of the International Symposium on Stability of Rock Slopes in Open Pit Mining and Civil Engineering*

- ring. Symposium Series S44 and Metallurgy. The South African Institute of Mining, S. 211–230.
- Lovas, T., Barsi, A., Detrekoi, A., Dunai, L., Csak, Z., Polgar, A., Berényi, A., Szocs, K. (2008): Terrestrial Laserscanning in Deformation Measurements of Structures. *International Archives of Photogrammetry and Remote Sensing*, Jahrgang37, S. 527–532.
- Lyche, T. (1976): Discrete polynomial spline approximation methods. In: Böhmer, K.; Meinardus, G.; Schempp, W. [Hrsg.], *Spline Functions. Lecture Notes in Mathematics*, Jahrgang501, Heft1, S. 144–176.
- Lyche, T. (1988): Note on the Oslo algorithm. *Computer-Aided Design*, Jahrgang20, Heft6, S. 353–355.
- Ma, W., Kruth, J. P. (1995): Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, Jahrgang12, Heft9, S. 663–675.
- Merziger, G., Wirth, T. (2010): *Repetitorium Höhere Mathematik: Mehr als 1000 durchgerechnete Beispiele*. 6. Aufl., Binomi, Barsinghausen.
- Metzner, M., Schwieger, V. (2014): Analyse von Messreihen im Zeitbereich; DVW-Seminar, Hannover 26.02.2014 - 27.02.2014. In: *Zeitabhängige Messgrößen - ihre Daten haben (Mehr-)Wert*. Bd. 74. Wißner-Verlag, Augsburg, S. 65–84.
- Meyer, M. (2017): *Signalverarbeitung. Analoge und digitale Signale, Systeme und Filter*. Springer Vieweg, Wiesbaden.
- Mohammadivojdan, B., Lorenz, F., Artz, T., Weiß, R., Hake, F., Alkhatib, Y., Neumann, I., Alkhatib, H. (2025): Robust algorithm for automatic surface-based outlier detection in MBES point clouds. *Marine geodesy*, Jahrgang48, Heft2, S. 141–172.
- Möser, M. (2018): *Digitale Signalverarbeitung in der Messtechnik*. Fachwissen Technische Akustik. Springer Vieweg, Berlin.
- Mukupa, W., Roberts, G., Hancock, C., Al-Manasir, K. (2016): A review of the use of terrestrial laser scanning application for change detection and deformation monitoring of structures. *Survey Review*, Jahrgang49, Heft353, S. 99–116.
- Neumann, I., Kutterer, H. (2007): Congruence tests and outlier detection in deformation analysis with respect to observation imprecision. *Journal of Applied Geodesy*, Jahrgang1, S. 1–7.
- Neuner, H.-B., Harmening, C., Schmitt, C. (2015): Die Modellierung mit geometrischen Freiformelementen - Eine mögliche Grundlage für die raumkontinuierliche Bauwerksüberwachung. In: *Interdisziplinäre Messaufgaben im Bauwesen* (26. - 27. März 2015, Darmstadt). Bd. 79. Wißner-Verlag, Augsburg, S. 47–65.
- Neuner, H.-B., Holst, C., Kuhlmann, H. (2016): Overview on current modelling strategies of point clouds for deformation analysis. *avn - Allgemeine Vermessungs-Nachrichten*, Jahrgang123, S. 328–339.
- Nguyen, M.-L. (2007): Nike of Samothrace Louvre Ma2369 n4. URL: https://no.m.wikipedia.org/wiki/Nike_fra_Samothrace (Zugriff 26.08.2024).
- Niemeier, W. (2008): *Ausgleichsrechnung: Statistische Auswertemethoden*. De Gruyter Lehrbuch. De Gruyter, Tübingen.
- Ohlmann-Lauber, J., Schäfer, T. (2011): Ansätze zur Ableitung von Deformationen aus TLS-Daten. In: 106. DVW-Seminar *Terrestrisches Laserscanning – TLS 2011*, Wißner-Verlag, Augsburg. Jahrgang66, S. 161–180.
- Ötsch, E., Harmening, C., Neuner, H. (2025): Employing variance component estimation for point cloud based geometric surface representation by B-splines. *Journal of Applied Geodesy*.
- Ötsch, E., Neuner, H. (2023): Detektion und Analyse von temperaturbedingter Deformation eines konischen Industrieschlotes. In: Wieser, A. [Hrsg.]: *Ingenieurvermessung 23. Beiträge zum 20. Internationalen Ingenieurvermessungskurs* Zürich, 2023. Wichmann, Berlin/Offenbach, S. 321–334.

- Paffenholz, J.-A., Stenz, U., Wujanz, D., Neitzel, F., Neumann, I. (2017): 3D-Punktwolken-basiertes Monitoring von Infrastrukturbauwerken am Beispiel einer historischen Gewölbebrücke. In: *Terrestrisches Laserscanning 2017 (TLS 2017)*. DVW-Schriftenreihe, Wißner-Verlag, Augsburg. Jahrgang88, S. 115–127.
- Park, S.-Y., Subbarao, M. (2003): An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*, Jahrgang24, Heft16, S. 2967–2976.
- Pelzer, H. (1985): *Geodätische Netze in Landes- und Ingenieurvermessung: Vorträge des Kontaktstudiums*. Bd. 13. Vermessungswesen bei Konrad Wittwer. Wittwer, Stuttgart.
- Piegl, L. (1989): Key Developments in Computer-Aided Geometric Design. *Computer-Aided Design*, Jahrgang21, Heft5, S. 262–273.
- Piegl, L., Tiller, W. (1997): *The NURBS Book*. 2. Aufl., Springer, Berlin und New York.
- Piegl, L., Tiller, W. (2000): Least-Squares B-Spline Curve Approximation with Arbitrary End Derivatives. *Engineering with Computers*, Jahrgang16, S. 109–116.
- Prautzsch, H. (1984): A short proof of the Oslo algorithm. *Computer Aided Geometric Design*, Jahrgang1, Heft1, S. 95–96.
- Prautzsch, H., Boehm, W., Paluszny, M. (2002): Bézier and B-spline techniques. *Mathematics and visualization*. Springer, New York, NY.
- Ramshaw, L. (1987): *Blossoming: A Connect-the-Dots Approach to Splines*. Digital Systems Research Center, Palo Alto, Kalifornien.
- Rashidi, M., Mohammadi, M., Sadeghlou Kivi, S., Abdolvand, M. M., Truong-Hong, L., Samali, B. (2020): A Decade of Modern Bridge Monitoring Using Terrestrial Laser Scanning: Review and Future Directions. *Remote Sensing*, Jahrgang12, Heft22, 3796.
- Rencher, A. C., Schaalje, G. B. (2008): *Linear models in statistics*. 2., Wiley-Interscience, Hoboken, NJ.
- Richter, T., Wick, T. (2017): *Einführung in die Numerische Mathematik: Begriffe, Konzepte und zahlreiche Anwendungsbeispiele*. Springer Spektrum, Berlin, Heidelberg.
- Riel, B., Simons, M., Ponti, D., Agram, P., Jolivet, R. (2018): Quantifying Ground Deformation in the Los Angeles and Santa Ana Coastal Basins Due to Groundwater Withdrawal. *Water Resources Research*, Jahrgang54.
- Riesenfeld, R. F. (1973): *Application of B-spline Approximation to Geometric Problems of Computer-aided Design*. Dissertation, Syracuse University, Syracuse, NY.
- Rogers, D. F. (2001): *An introduction to NURBS: With historical perspective*. Morgan Kaufmann Publishers, San Francisco, Kalifornien.
- Rusu, R. B., Blodow, N., Beetz, M. (2009): Fast Point Feature Histograms (FPFH) for 3D registration. *2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, S. 3212–3217.
- Sadiq, S., Harmening, C. (2025): Investigating the potential of stochastic relationships to model deformations. *Journal of Applied Geodesy*, Jahrgang19, Heft3, S. 395–402.
- Schäfer, T., Weber, T., Kyrinovic, P., Zámečníková, M. (2004): Deformation Measurement Using Terrestrial Laser Scanning at the Hydropower Station of Gabčíkovo. *INGEO 2004 and FIG Regional Central and Eastern European Conference on Engineering Surveying*, 11.-13. November 2004, Bratislava, Slowakei.
- Schill, F. J. (2020): *Überwachung von Tragwerken mit Profilsclannern*. Dissertation. Technische Universität Darmstadt.
- Schmitt, C., Neuner, H.-B. (2015): Knot estimation on B-Spline curves. *Österreichische Zeitschrift für Vermessung und Geoinformation (VGI)*, Jahrgang103, Heft2-3, S. 188–197.
- Schoenberg, I. J. (1946): Contribution to the problem of approximation of equidistant data by analytic functions. *Quarterly of Applied Mathematics*, Jahrgang4, Heft1, S. 45–99, 112–141.

- Schoenberg, I. J. (1967): On spline functions. In: Sisha, O. [Hrsg.], *Inequalities*. Academic Press, New York, NY, S. 255–291.
- Schumaker, L. L. (2007): *Spline Functions: Basic Theory*. 3. Aufl. Cambridge Mathematical Library. Cambridge University Press, Cambridge.
- Seidel, H.-P. (1988): Knot insertion from a blossoming point of view. *Computer Aided Geometric Design*, Jahrgang5, Heft1, S. 81–86.
- Serantoni, E., Wieser, A. (2016): TLS-based Deformation Monitoring of Snow Structures. In: *Terrestrisches Laserscanning 2016 (TLS 2016)*. DVW-Schriftenreihe, Wißner-Verlag, Augsburg. Jahrgang85, S. 89–102.
- Steinbach, J., Harmening, C. (2025): Laser scanning based deformation analysis of a wooden dome under load. *Journal of Applied Geodesy*.
- Stilla, U., Xu, Y. (2023): Change detection of urban objects using 3D point clouds: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, Jahrgang197, S. 228–255.
- Theiler, P., Wegner, J., Schindler, K. (2013): Markerless point cloud registration with keypoint-based 4-points congruent sets. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, JahrgangII-5/W2.
- Theiler, P., Wegner, J., Schindler, K. (2014): Fast registration of laser scans with 4-point congruent sets - what works and what doesn't. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Jahrgang2, Heft3, S. 149–156.
- Tombari, F., Salti, S., Di Stefano, L. (2010): Unique Signatures of Histograms for Local Surface Description. In: Daniilidis, K.; Maragos, P.; Paragios, N. [Hrsg.]: *Computer Vision – ECCV 2010*. Springer. Berlin, Heidelberg. Jahrgang6313, S. 356–369.
- Tombari, F., Salti, S., Di Stefano, L. (2011): A combined texture-shape descriptor for enhanced 3D feature matching. *Proceedings of the 18th IEEE International Conference on Image Processing, ICIP 2011*, 11. - 14. September 2011, Brüssel, Belgien, S. 809–812.
- Truong-Hong, L., Lindenbergh, R. (2019): Measuring deformation of bridge structures using laser scanning data. *Proceedings of the 4th Joint International Symposium on Deformation Monitoring (JISDM)*, 15.-17. Mai 2019, Athen, Griechenland.
- Tsakiri, M., Lichti, D., Pfeifer, N. (2006): Terrestrial laser scanning for deformation monitoring. In: *Proceedings of the 12th FIG Symposium*, März 22-24, Baden, S. 211–230.
- Vežočnik, R., Ambrožič, T., Sterle, O., Bilban, G., Pfeifer, N., Stopar, B. (2009): Use of Terrestrial Laser Scanning Technology for Long Term High Precision Deformation Monitoring. *Sensors*, Jahrgang9, S. 9873–9895.
- Vosselman, G., Maas, H.-G. (2010): *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.
- Wagner, A., Wiedemann, W., Wunderlich, T. (2017): Fusion of Laser Scan and Image Data for Deformation Monitoring-Concept and Perspective. *INGEO 2017 – 7 th International Conference on Engineering Surveying*, 18.-20. Oktober 2017, Lissabon, Portugal, S. 157–164.
- Walton, G., Delaloye, D., Diederichs, M. (2014): Development of an elliptical fitting algorithm to improve change detection capabilities with applications for deformation monitoring in circular tunnels and shafts. *Tunnelling and Underground Space Technology*, Jahrgang43, S. 336–349.
- Wang, Z., Xu, X., He, X., Wei, X., Yang, H. (2023): A Method for Convergent Deformation Analysis of a Shield Tunnel Incorporating B-Spline Fitting and ICP Alignment. *Remote Sensing*, Jahrgang15, Heft21, 5112.
- Weitz, E. (2019): *Elementare Differentialgeometrie (nicht nur) für Informatiker: Mit vielen Grafiken und Visualisierungen in JavaScript*. 1., Springer Spektrum, Berlin.
- Wiedemann, W., Holst, C. (2022): Identifying individual rocks within laser scans for a rigorous deformation analysis of water dams. *Proceedings of the 5th Joint International Symposium*

- on Deformation Monitoring, (JISDM2022), 20.- 22. Juni 2022, Valencia, Spanien, S. 327–334.
- Wujanz, D. (2016): Terrestrial Laser Scanning for Geodetic Deformation Monitoring. Deutsche Geodätische Kommission, Reihe C (Dissertationen), Heft775, München.
- Wujanz, D., Krueger, D., Neitzel, F. (2014): Der ICPprox-Algorithmus: Oberflächenbasierte Registrierung terrestrischer Laserscans für die Deformationsanalyse. Beiträge zum 17. Internationalen Ingenieurvermessungskurs, S. 245–256.
- Wunderlich, T., Niemeier, W., Wujanz, D., Holst, C., Neitzel, F., Kuhlmann, H. (2016): Areal Deformation Analysis from TLS Point Clouds – The Challenge. AVN Allgemeine Vermessungsnachrichten, Jahrgang123, S. 340–351.
- Xiao, P., Zhao, R., Li, D., Zeng, Z., Qi, S., Yang, X. (2022): As-Built Inventory and Deformation Analysis of a High Rockfill Dam under Construction with Terrestrial Laser Scanning. Sensors, Jahrgang22(2):521.
- Yang, B., Zang, Y. (2014): Automated registration of dense terrestrial laser-scanning point clouds using curves. ISPRS Journal of Photogrammetry and Remote Sensing, Jahrgang95, S. 109–121.
- Yang, B., Zang, Y., Dong, Z., Huang, R. (2015): An automated method to register airborne and terrestrial laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, Jahrgang109, S. 62–76.
- Yang, Y. (2024): Towards Improved Targetless Registration and Deformation Analysis of TLS Point Clouds Using Patch-based Segmentation. Deutsche Geodätische Kommission, Reihe C (Dissertationen), Heft929, München.
- Yang, Y., Schwieger, V. (2022): Supervoxel-based targetless registration and identification of stable areas for deformed point clouds. Journal of Applied Geodesy, Jahrgang17, Heft2, S. 161–170.
- Yuan, W., Yuan, X., Xu, S., Gong, J., Shibasaki, R. (2019): Dense Image-Matching via Optical Flow Field Estimation and Fast-Guided Filter Refinement. Remote Sensing, Jahrgang11, Heft20, 2410.
- Zaczek-Peplinska, J., Kowalska, M. (2017): Terrestrial laser scanning in monitoring of anthropogenic objects. Geodesy and Cartography, Jahrgang66, Heft2, S. 347–364.

Abkürzungsverzeichnis

C2M	Cloud-to-Mesh
DBKM	Direkte B-Spline-Kontrollpunkt-Methode
FPFH	Fast Point Feature Histogram
ICP	Iterative-Closest-Point-Algorithmus
K-4PCS	Keypoint 4-Points Congruents Set
M2M	Mesh-to-Mesh
M3C2	Multiscale Model to Model Cloud Comparison
RANSAC	Random Sample Consensus
SHOT	Signature of Histograms of Orientations
TLS	Terrestrisches Laserscanning
4PCS	4-Points Congruents Set

A Anhang

A.1 Algorithmus von Böhm - Stochastischer und numerischer Nachweis

Der folgende Abschnitt soll zum einen überprüfen, wie sich die Varianzen und Kovarianzen einer Varianz-Kovarianzmatrix von Kontrollpunkten verhalten, wenn der Algorithmus von Böhm auf ein Kontrollpolygon einer B-Spline-Kurve bzw. das Kontrollgitter einer B-Spline-Fläche angewendet wird, zum anderen soll durch einen mathematischen Nachweis ermittelt werden, ob die Form der B-Spline-Kurve hiervon tangiert wird. Hierzu werden zunächst die Kovarianzfortpflanzung für die Kontrollpunkte einer B-Spline-Kurve und danach einer B-Spline-Fläche erarbeitet. Für einen Nachweis der Verhaltensweise der Varianz-Kovarianzmatrix im Zusammenhang mit dem Algorithmus von Böhm werden daraufhin die Varianz-Kovarianzmatrix eines beispielhaften 2D-Kontrollpolygons sowie eines beispielhaften 3D-Kontrollgitters berechnet. Die zwei letzten Abschnitte von Unterkapitel A.1 untersuchen die Einflussnahme der Wahl des neu einzufügenden Knotenwerts auf die Standardabweichung und zeigen, ob numerisch ermittelbare Abweichungen von B-Spline-Kurvenpunkten vor und nach der Anwendung des Algorithmus von Böhm auftreten.

Varianz-Kovarianzfortpflanzung - Algorithmus von Böhm (2D)

Für den 2D-Fall werden die alten Kontrollpunkte $\mathbf{d}$ vor Ausführung des Algorithmus von Böhm als Beobachtungen angesehen, während als Zielgrößen die neu gebildeten de Boor-Punkte $\mathbf{q}$ interpretiert werden:

$$\mathbf{d} = (d_0, d_1, \dots, d_m) \quad (\text{A.1})$$

$$\mathbf{q} = (q_0, q_1, \dots, q_{m+1}). \quad (\text{A.2})$$

Für die Durchführung der Kovarianzfortpflanzung ist zu Beginn die Kovarianzmatrix der alten Kontrollpunkte $\mathbf{S}_d$ zu generieren, in der die Varianzen $s_{d_0}^2, s_{d_1}^2, \dots, s_{d_m}^2$ von $d_0, d_1, \dots, d_m$ sowie die Kovarianzen s_{d_α, d_β} für alle $\alpha \neq \beta$ mit $\alpha = \beta = 0, 1, \dots, m$ aufgeführt sind. Für den 2D-Fall ergibt sich die Matrixgröße aus der Anzahl der $m + 1$ Kontrollpunkte:

$$\mathbf{S}_d = \begin{pmatrix} s_{d_0}^2 & s_{d_0, d_1} & s_{d_0, d_2} & \dots & s_{d_0, d_m} \\ s_{d_1, d_0} & s_{d_1}^2 & s_{d_1, d_2} & \dots & s_{d_1, d_m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{d_m, d_0} & s_{d_m, d_1} & s_{d_m, d_2} & \dots & s_{d_m}^2 \end{pmatrix} [m + 1 \times m + 1]. \quad (\text{A.3})$$

Siehe hierzu Rencher und Schaalje (2008). Im weiteren Fortgang besteht die Aufgabe darin, ein funktionales Modell aufzustellen, das eine Verbindung der gesuchten Zielgrößen mit den Beobachtungen herstellt. Diese Verknüpfung liefert Gleichung 4.4 des Algorithmus von Böhm:

$$\mathbf{q}_i = (1 - \gamma_i) \cdot \mathbf{d}_{i-1} + \gamma_i \cdot \mathbf{d}_i \quad \text{mit}$$

$$\gamma_i = \begin{cases} 1 & i \leq s - n \\ \frac{\bar{u} - u_i}{u_{i+n} - u_i} & s - n < i \leq s \\ 0 & i < s + 1 \end{cases}.$$

Für jeden der $m + 2$ neuen Kontrollpunkte ist dementsprechend ein funktionales Modell zu erzeugen, in dem alle Beobachtungen enthalten sind:

$$\begin{aligned}\Phi_1(\mathbf{d}) &= f_1(d_0, d_1, \dots, d_m) \\ \Phi_2(\mathbf{d}) &= f_2(d_0, d_1, \dots, d_m) \\ &\vdots \\ \Phi_{m+2}(\mathbf{d}) &= f_{m+2}(d_0, d_1, \dots, d_m).\end{aligned}\tag{A.4}$$

In einem zweiten Schritt sind $m + 2$ totale Differentiale für die Funktion F zu generieren. Hierbei ist für jede Beobachtung eine partielle Ableitung zu bilden:

$$\begin{aligned}dF_1 &= \frac{\partial f_1}{\partial d_0} dd_0 + \frac{\partial f_1}{\partial d_1} dd_1 + \dots + \frac{\partial f_1}{\partial d_m} dd_m \\ dF_2 &= \frac{\partial f_2}{\partial d_0} dd_0 + \frac{\partial f_2}{\partial d_1} dd_1 + \dots + \frac{\partial f_2}{\partial d_m} dd_m \\ &\vdots \\ dF_{m+2} &= \frac{\partial f_{m+2}}{\partial d_0} dd_0 + \frac{\partial f_{m+2}}{\partial d_1} dd_1 + \dots + \frac{\partial f_{m+2}}{\partial d_m} dd_m.\end{aligned}\tag{A.5}$$

Aus den partiellen Ableitungen kann daraufhin eine sogenannte Jacobi-Matrix $\mathbf{F}$ konstruiert werden:

$$\mathbf{F} = \begin{pmatrix} \frac{\partial f_1}{\partial d_0} & \frac{\partial f_1}{\partial d_1} & \frac{\partial f_1}{\partial d_2} & \dots & \frac{\partial f_1}{\partial d_m} \\ \frac{\partial f_2}{\partial d_0} & \frac{\partial f_2}{\partial d_1} & \frac{\partial f_2}{\partial d_2} & \dots & \frac{\partial f_2}{\partial d_m} \\ \frac{\partial f_3}{\partial d_0} & \frac{\partial f_3}{\partial d_1} & \frac{\partial f_3}{\partial d_2} & \dots & \frac{\partial f_3}{\partial d_m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_{m+1}}{\partial d_0} & \frac{\partial f_{m+1}}{\partial d_1} & \frac{\partial f_{m+1}}{\partial d_2} & \dots & \frac{\partial f_{m+1}}{\partial d_m} \\ \frac{\partial f_{m+2}}{\partial d_0} & \frac{\partial f_{m+2}}{\partial d_1} & \frac{\partial f_{m+2}}{\partial d_2} & \dots & \frac{\partial f_{m+2}}{\partial d_m} \end{pmatrix} = \begin{pmatrix} \gamma_0 & 0 & 0 & \dots & 0 \\ (1 - \gamma_1) & \gamma_1 & 0 & \dots & 0 \\ 0 & (1 - \gamma_2) & \gamma_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \gamma_m \\ 0 & 0 & 0 & \dots & (1 - \gamma_m) \end{pmatrix} \tag{A.6}$$

[$m + 2 \times m + 1$].

Auf Basis der partiellen Ableitungen von Gleichung A.6 und den Varianzen und Kovarianzen aus Gleichung A.3 findet entsprechend des Schemas des Kovarianzfortpflanzungsgesetzes eine Bestimmung der Kovarianzmatrix Σ_q statt (Barlow 1999):

$$\begin{matrix} \Sigma_q & = & \mathbf{F} & \cdot & \Sigma_d & \cdot & \mathbf{F}^T \\ [m + 2 \times m + 2] & & [m + 2 \times m + 1] & & [m + 1 \times m + 1] & & [m + 1 \times m + 2] \end{matrix} \tag{A.7}$$

Hierbei entspricht Σ_d der Kovarianzmatrix der Beobachtungen in Form der Varianzen der alten Kontrollpunkte vor Anwendung des Algorithmus von Böhm und $\mathbf{F}$ der Jacobi-Matrix mit den partiellen Ableitungen. Als Resultat wird eine Kovarianzmatrix Σ_q der neuen Kontrollpunkte erzielt, die Aufschluss über die Einflussnahme des Algorithmus von Böhm auf die Standardabweichung der de Boor-Punkte geben kann.

Das Wurzelziehen jeder Varianz bringt die Standardabweichung der neuen Kontrollpunkte $\mathbf{q}$ des Algorithmus von Böhm hervor:

$$\sigma_{q_i} = +\sqrt{\sigma_{q_i}^2}.\tag{A.8}$$

Zu beachten ist, dass pro zusätzlich angeführtem Kontrollpunkt, der in einem iterativen Prozess mittels Algorithmus von Böhm berechnet wird, eine Kovarianzmatrix Σ_q generiert werden muss. Demzufolge ermittelt sich bei h zusätzlichen Kontrollpunkten die finale Kovarianzmatrix aus h Kovarianzfortpflanzungsberechnungen.

Sollen abgesehen von der Varianz sowie Kovarianz eines Kontrollpunktes mit jedem der anderen Kontrollpunkte zusätzlich die Korrelationen zwischen den neu generierten de Boor-Punkten bestimmt werden, ist eine Korrelationsmatrix $\mathbf{P}_\rho$ der Form:

$$\mathbf{P}_\rho = (\rho_{d_\gamma, d_\delta}) = \begin{pmatrix} 1 & \rho_{d_0, d_1} & \rho_{d_0, d_2} & \cdots & \rho_{d_0, d_{m+1}} \\ \rho_{d_1, d_0} & 1 & \rho_{d_1, d_2} & \cdots & \rho_{d_1, d_{m+1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{d_{m+1}, d_0} & \rho_{d_{m+1}, d_1} & \rho_{d_{m+1}, d_2} & \cdots & 1 \end{pmatrix} \quad [m+2 \times m+2] \quad (\text{A.9})$$

aufzustellen. Die einzelnen Elemente der Korrelationsmatrix lassen sich mittels Dividieren der Kovarianz $\sigma_{d_\gamma, d_\delta}$ durch das Produkt der Standardabweichungen von d_γ und d_δ mit $\gamma = \delta = 0, 1, \dots, m+1$ ermitteln:

$$\rho_{d_\gamma, d_\delta} = \frac{\sigma_{d_\gamma, d_\delta}}{\sigma_{d_\gamma} \cdot \sigma_{d_\delta}} \quad (\text{A.10})$$

Nimmt $\rho_{d_\gamma, d_\delta}$ einen Wert gleich 0 an, bestehen keine Korrelationen zwischen d_γ und d_δ . Beide Kontrollpunkte sind stochastisch unabhängig. Bewegt sich $\rho_{d_\gamma, d_\delta}$ allerdings nahe dem Wert 1 oder -1 existieren starke Korrelationen zwischen beiden de Boor-Punkten und es bestehen stochastische Abhängigkeiten. Das Vorzeichen des Korrelationskoeffizienten entscheidet, ob sich die Parameter proportional oder reziprok proportional verhalten. Eine reziproke Proportionalität liegt zwischen zwei Variablen vor, wenn d_γ proportional zum Kehrwert von d_δ , also $1/d_\delta$, ist (Niemeier 2008).

Varianz-Kovarianzfortpflanzung - Algorithmus von Böhm (3D)

Die stochastische Modellierung von B-Spline-Kontrollpunkten einer B-Spline-Fläche in Verbindung mit dem Algorithmus von Böhm erfordert, wie im Kurvenfall, die Durchführung der Kovarianzfortpflanzung nach jedem Hinzufügen eines Kontrollpunktes. Das funktionale Modell, dass eine Verbindung zwischen den alten sowie neuen Kontrollpunkten herstellt, liefert somit ebenfalls Gleichung 4.4 des Algorithmus von Böhm. Der Beobachtungsvektor setzt sich aus einer Aufreihung von Kontrollpunkten zusammen, die ursprünglich in einem Kontrollgitter in Matrixform mit $m+1$ Zeilen und $a+1$ Spalten vorliegen. Für eine bessere Übersicht der einzelnen Beobachtungsgrößen, wird die Notation der Kontrollpunkte auf eine einstellige Nummerierung reduziert. Der Beobachtungsvektor besteht demgemäß aus $c+1$ de Boor-Punkten, wobei $c+1 = (m+1) \cdot (a+1)$ bedeutet:

$$\begin{aligned} \mathbf{d} &= (d_{0,0}, d_{1,0}, d_{2,0}, d_{3,0}, d_{0,1}, d_{1,1}, d_{2,1}, d_{3,1}, d_{4,1}, d_{5,1}, d_{0,2}, \dots, d_{m,a}) \\ &= (d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, \dots, d_c). \end{aligned} \quad (\text{A.11})$$

In jedem Iterationsschritt des Algorithmus von Böhm wird ein neuer Knoten und somit ein zusätzlicher de Boor-Punkt eingefügt, wodurch der Zielgrößenvektor einen Parameter mehr als der Beobachtungsvektor enthält:

$$\mathbf{q} = (q_0, q_1, q_2, \dots, q_{c+1}). \quad (\text{A.12})$$

Mit Hilfe der Jacobi-Matrix, die mittels partiellen Ableitungen entsprechend Gleichung A.13 aufgestellt wird, sowie der Varianz-Kovarianzmatrix der alten Kontrollpunkte Σ_{aK} , kann die Varianz-

Kovarianzmatrix der neuen Kontrollpunkte Σ_{nK} mittels Kovarianzfortpflanzung gemäß Gleichung A.14 erstellt werden:

$$\begin{aligned}
 \mathbf{F} &= \begin{pmatrix} \frac{\partial q_0}{\partial d_0} & \frac{\partial q_0}{\partial d_1} & \frac{\partial q_0}{\partial d_2} & \cdots & \frac{\partial q_0}{\partial d_c} \\ \frac{\partial q_1}{\partial d_0} & \frac{\partial q_1}{\partial d_1} & \frac{\partial q_1}{\partial d_2} & \cdots & \frac{\partial q_1}{\partial d_c} \\ \frac{\partial q_2}{\partial d_0} & \frac{\partial q_2}{\partial d_1} & \frac{\partial q_2}{\partial d_2} & \cdots & \frac{\partial q_2}{\partial d_c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial q_c}{\partial d_0} & \frac{\partial q_c}{\partial d_1} & \frac{\partial q_c}{\partial d_2} & \cdots & \frac{\partial q_c}{\partial d_c} \\ \frac{\partial q_{c+1}}{\partial d_0} & \frac{\partial q_{c+1}}{\partial d_1} & \frac{\partial q_{c+1}}{\partial d_2} & \cdots & \frac{\partial q_{c+1}}{\partial d_c} \end{pmatrix} \\
 &= \begin{pmatrix} \gamma_0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ (1-\gamma_1) & \gamma_1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & (1-\gamma_2) & \gamma_2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & (1-\gamma_3) & \gamma_3 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & (1-\gamma_4) & \gamma_4 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma_5 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-\gamma_6) & \gamma_6 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & (1-\gamma_{c+1}) \end{pmatrix} \quad [c+2 \times c+1]. \quad (\text{A.13})
 \end{aligned}$$

$$\Sigma_{\text{q}} = \mathbf{F} \cdot \Sigma_{\text{d}} \cdot \mathbf{F}^T. \quad (\text{A.14})$$

Eine Besonderheit des 3D-Falls kann dahingehend festgestellt werden, dass in der Jacobi-Matrix $\mathbf{F}$ der Übergang zweier Kontrollpolygone derart bemerkbar wird, dass sich die partiellen Ableitungen des jeweils ersten de Boor-Punktes eines Kontrollpolygons nur aus dem Term γ_i zusammensetzen, statt aus den Termen γ_i und $(1-\gamma_i)$. Dies kann in Gleichung A.13 etwa bei γ_5 festgestellt werden. Hierdurch wird deutlich, dass der Einfluss der Kontrollpolygone untereinander weiterhin, trotz des Algorithmus von Böhm, unbeeinflusst ist, da die Bildung einer identischen Kurvenform kontrollpolygonweise im Algorithmus erfolgt.

Beispielhafte Berechnungen für den 2D-Fall

Nachdem die theoretischen Aspekte der Kovarianzfortpflanzung im Bezug auf den Algorithmus von Böhm offengelegt wurden, sollen im Folgenden die Kovarianzmatrizen Σ_q einer B-Spline-Kurve und im darauffolgenden Abschnitt einer B-Spline-Fläche bestimmt werden, um einen Eindruck der Verhaltensweise der Varianzen und Kovarianzen durch die Anwendung des Algorithmus von Böhm zu erlangen.

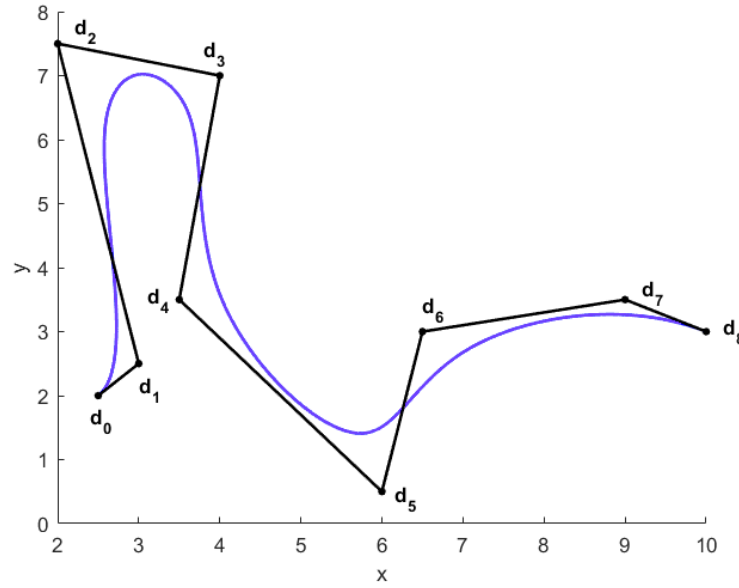


Abb. A.1: B-Spline-Kurve dritten Grades.

Die in Abbildung A.1 dargestellte beispielhafte B-Spline-Kurve besteht aus $m + 1 = 9$ Kontrollpunkten und erhält mittels dreimaliger Ausführung des Algorithmus von Böhm drei neue Kontrollpunkte. Für die B-Spline-Kurve dritten Grades und einem Knotenvektor von

$$\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6) \quad (\text{A.15})$$

wird ein neuer Knoten $\bar{u} = 3$ eingefügt. Dieser entspricht dem Median der Knotenvektorwerte. Der Inhalt der Varianz-Kovarianzmatrix der alten Kontrollpunkte Σ_d wird in erster Linie auf Basis einer Diagonalmatrix mit einer Varianz von 1 mm^2 gebildet. Obwohl im Hauptteil dieser Arbeit eine Standardabweichung von $0,5 \text{ mm}$ verwendet wird, wird hier der Wert 1 mm gewählt, um Größenunterschiede in berechneten Varianz-Kovarianzmatrizen leichter feststellen zu können.

$$\Sigma_d = \begin{pmatrix} 1 \text{ mm}^2 & 0 & 0 & \dots & 0 \\ 0 & 1 \text{ mm}^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \text{ mm}^2 \end{pmatrix} \quad [9 \times 9] \quad (\text{A.16})$$

Die Generierung der Matrix $\mathbf{F}$ erfolgt daraufhin mittels Bildung der partiellen Ableitungen der Funktionen zur Berechnung der 10 neuen Kontrollpunkte $\mathbf{q}$ nach den 9 alten de Boor-Punkten $\mathbf{d}$. Die

am Rand der Matrix aufgeführten grauen Parameter verdeutlichen die Funktion sowie den Kontrollpunkt nach dem die Funktion abgeleitet wird.

$$\mathbf{F} = \begin{pmatrix} & \mathbf{d}_0 & \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 & \mathbf{d}_4 & \mathbf{d}_5 & \mathbf{d}_6 & \mathbf{d}_7 & \mathbf{d}_8 \\ \mathbf{q}_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_4 & 0 & 0 & 0 & 0,3333 & 0,6667 & 0 & 0 & 0 & 0 \\ \mathbf{q}_5 & 0 & 0 & 0 & 0 & 0,6667 & 0,3333 & 0 & 0 & 0 \\ \mathbf{q}_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{q}_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \mathbf{q}_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad [10 \times 9] \quad (\text{A.17})$$

Auf Basis von Gleichung A.7, entsprechend des Kovarianzfortpflanzungsgesetzes, ist eine Ermittlung der Kovarianzmatrix $\mathbf{\Sigma}_q$ durchführbar.

$$\mathbf{\Sigma}_q = \begin{pmatrix} & \mathbf{q}_0 & \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 & \mathbf{q}_4 & \mathbf{q}_5 & \mathbf{q}_6 & \mathbf{q}_7 & \mathbf{q}_8 & \mathbf{q}_9 \\ \mathbf{q}_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_3 & 0 & 0 & 0 & 1 & 0,3333 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{q}_4 & 0 & 0 & 0 & 0,3333 & 0,5556 & 0,4444 & 0 & 0 & 0 & 0 \\ \mathbf{q}_5 & 0 & 0 & 0 & 0 & 0,4444 & 0,5556 & 0,3333 & 0 & 0 & 0 \\ \mathbf{q}_6 & 0 & 0 & 0 & 0 & 0 & 0,3333 & 1 & 0 & 0 & 0 \\ \mathbf{q}_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{q}_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \mathbf{q}_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad [\text{mm}^2] \quad [10 \times 10] \quad (\text{A.18})$$

Betrachtet man die Elemente der resultierenden Kovarianzmatrix, kann festgestellt werden, dass sich Kovarianzen zwischen den Kontrollpunkten gebildet haben und die Varianz der in Folge des Böhm Algorithmus koordinatenmäßig geänderten de Boor-Punkte in Relation zu den in der Position unveränderten Kontrollpunkten geringer ausfällt, als die zu Beginn angenommenen Varianzen von 1 mm^2 .

Durch Wurzelziehen erhält man die Standardabweichungen der mittels Anwendung des Algorithmus von Böhm generierten Kontrollpunkte. Gleichung A.19 zeigt, dass die Standardabweichungen der zwei neuen Kontrollpunkte $\mathbf{q}_4$ und $\mathbf{q}_5$, die den alten de Boor-Punkt $\mathbf{d}_4$ ersetzen, mit $0,75 \text{ mm}$ um $0,25 \text{ mm}$ geringer ausfallen, als die originäre Standardabweichung.

In Abbildung A.2 kann das neu gebildete Kontrollpolygon in schwarz sowie das Kontrollpolygon vor Ausführung des Knoteneinfügealgorithmus in grau eingesehen werden.

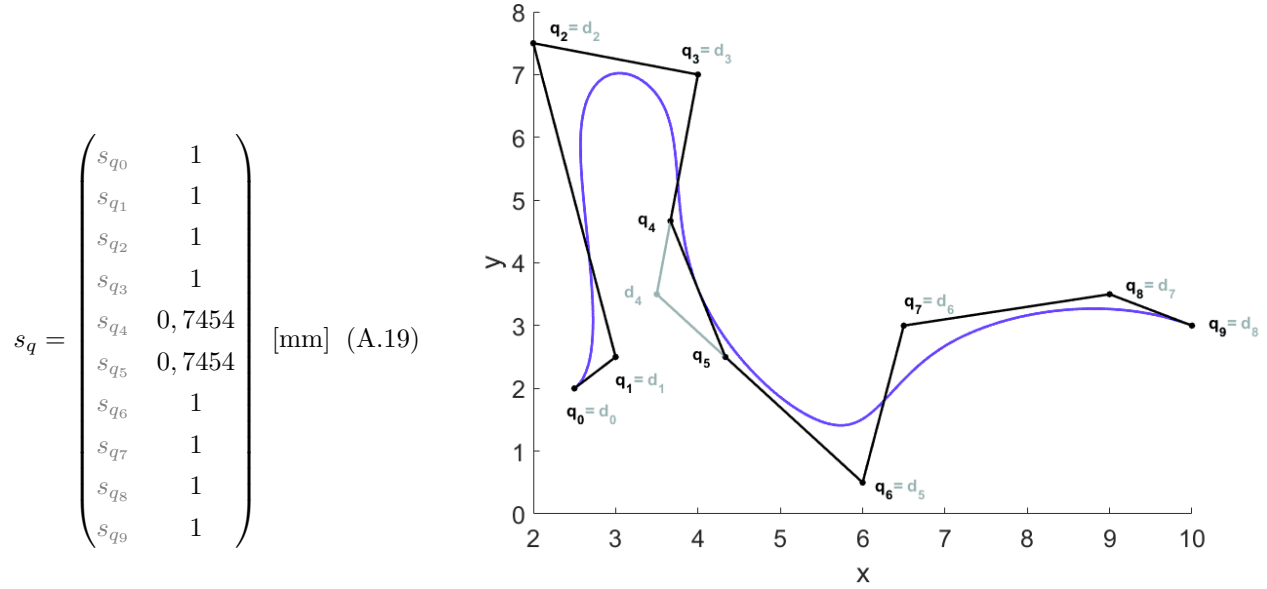


Abb. A.2: B-Spline-Kurve dritten Grades nach erster Anwendung des Algorithmus von Böhm.

In einem zweiten Durchgang wird ein weiterer Kontrollpunkt mittels Algorithmus von Böhm durch Einfügen des Knotens $\bar{u} = 3$ in den inzwischen veränderten Knotenvektor

$$U = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6) \quad (\text{A.20})$$

erhalten. Die Kovarianzfortpflanzung mit einer hierauf basierenden Matrix F und der im ersten Berechnungsschritt generierten Kovarianzmatrix Σ_d liefert die in Gleichung A.21 dargestellte Matrix Σ_q . Auffällig sind besonders die Kovarianzen, die im Vergleich zum ersten Berechnungsdurchgang nicht nur einen Wert versetzt zur Hauptdiagonalen auftreten, sondern bereits in zweiter Reihe sichtbar und von Null abweichend sind.

$$\Sigma_q = \begin{pmatrix} & q_0 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} \\ q_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q_3 & 0 & 0 & 0 & 1 & 0,3333 & 0,1667 & 0 & 0 & 0 & 0 & 0 \\ q_4 & 0 & 0 & 0 & 0,3333 & 0,5556 & 0,5 & 0,4444 & 0 & 0 & 0 & 0 \\ q_5 & 0 & 0 & 0 & 0,1667 & 0,5 & 0,5 & 0,5 & 0,1667 & 0 & 0 & 0 \\ q_6 & 0 & 0 & 0 & 0 & 0,4444 & 0,5 & 0,5556 & 0,3333 & 0 & 0 & 0 \\ q_7 & 0 & 0 & 0 & 0 & 0 & 0,1667 & 0,3333 & 1 & 0 & 0 & 0 \\ q_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ q_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ q_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad [\text{mm}^2] \quad [11 \times 11] \quad (\text{A.21})$$

Bei der Standardabweichung des de Boor-Punktes q_5 , der als zusätzliches Element zum Kontrollpolygon hinzukommt, ist eine Standardabweichung von 0,71 mm in Gleichung A.22 auszumachen. Diese liegt demnach 0,29 mm unterhalb der Anfangsstandardabweichung. Die Standardabweichungen der Kontrollpunkte q_4 und q_6 bleiben in Folge der unveränderten Position bei 0,75 mm. Die zugehörige

B-Spline-Kurve und deren Kontrollpolygon nach zweimaliger Anwendung des Algorithmus von Böhm sind in Abbildung A.3 dargestellt.

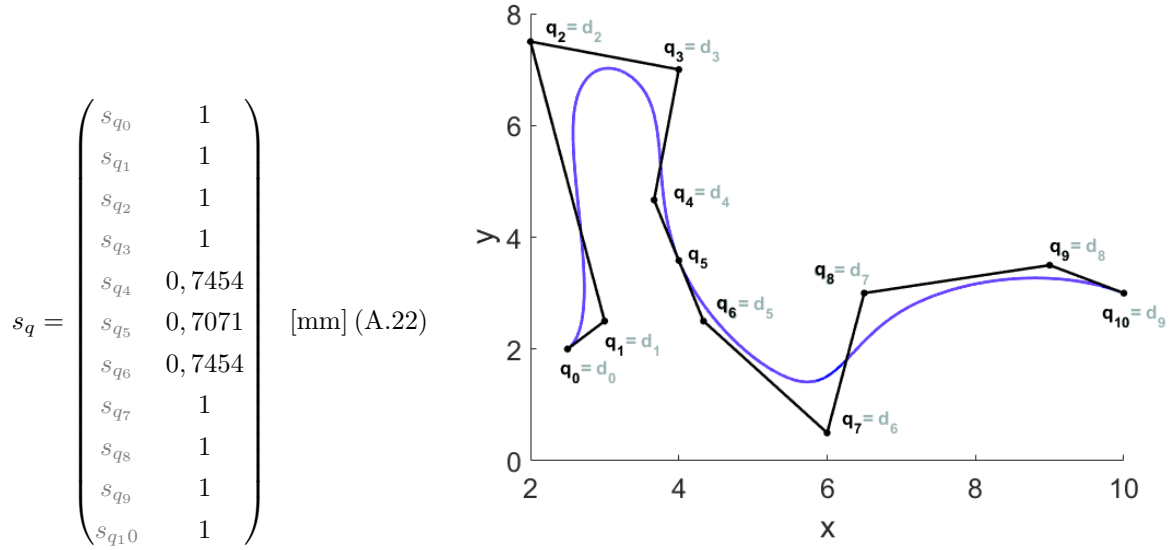


Abb. A.3: B-Spline-Kurve dritten Grades nach zweiter Anwendung des Algorithmus von Böhm.

Im dritten Durchgang, in dem erneut ein zusätzlicher Knoten mit dem Wert $\bar{u} = 3$ in den vorhandenen Knotenvektor

$$U = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6) \quad (\text{A.23})$$

eingefügt wird, können bei der berechneten Kovarianzmatrix Σ_q bereits drei mit Werten ungleich Null befüllte Reihen zur Hauptdiagonalen ausgemacht werden.

$$\Sigma_q = \begin{pmatrix} & q_0 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} & q_{11} \\ q_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q_3 & 0 & 0 & 0 & 1 & 0,3333 & 0,1667 & 0,1667 & 0 & 0 & 0 & 0 & 0 \\ q_4 & 0 & 0 & 0 & 0,3333 & 0,5556 & 0,5 & 0,5 & 0,4444 & 0 & 0 & 0 & 0 \\ q_5 & 0 & 0 & 0 & 0,1667 & 0,5 & 0,5 & 0,5 & 0,5 & 0,1667 & 0 & 0 & 0 \\ q_6 & 0 & 0 & 0 & 0,1667 & 0,5 & 0,5 & 0,5 & 0,5 & 0,1667 & 0 & 0 & 0 \\ q_7 & 0 & 0 & 0 & 0 & 0,4444 & 0,5 & 0,5 & 0,5556 & 0,3333 & 0 & 0 & 0 \\ q_8 & 0 & 0 & 0 & 0 & 0 & 0,1667 & 0,1667 & 0,3333 & 1 & 0 & 0 & 0 \\ q_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ q_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ q_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad [\text{mm}^2] \quad [12 \times 12] \quad (\text{A.24})$$

Die Standardabweichungen des neuen Kontrollpunktes q_6 des dritten Berechnungsvorgangs, in Gleichung A.25, zeigen ähnliche Verhaltensweisen wie in den Berechnungen zuvor und weisen eine Standardabweichung von 0,71 mm vor. q_4 , q_5 und q_7 bleiben in ihrer Position bestehen und behalten demgemäß die jeweilige Standardabweichung bei, lediglich die Punktnummerierung ändert von d_6 auf q_7 . Das durch die dreimalige Anwendung des Algorithmus von Böhm entstandene Kontrollpolygon ist in Abbildung A.4 einsehbar.

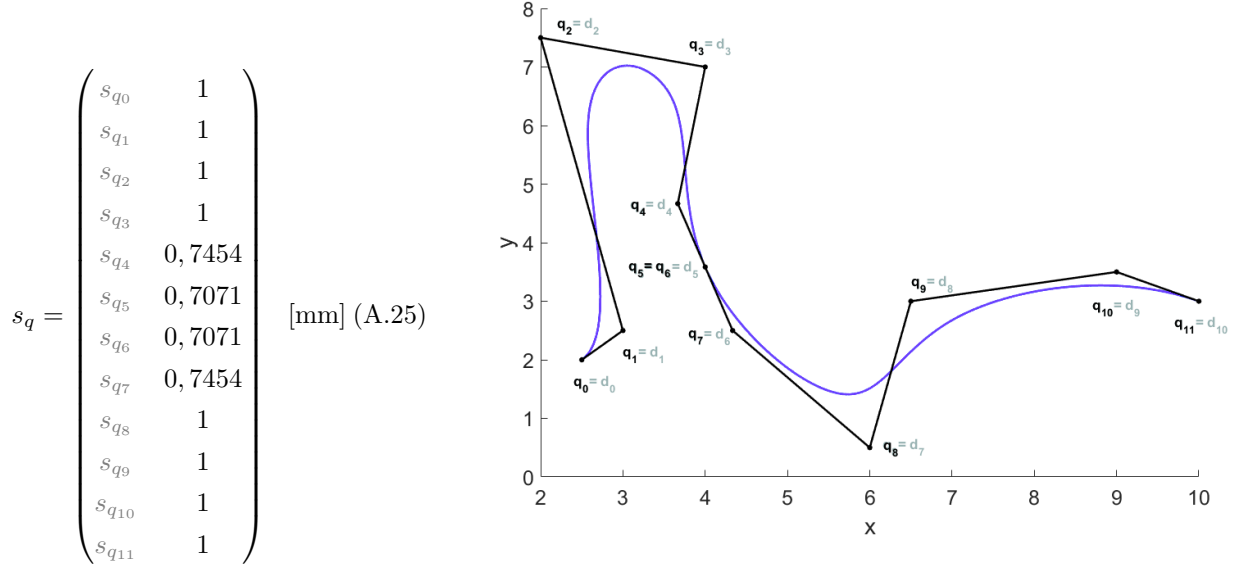


Abb. A.4: B-Spline-Kurve dritten Grades nach dritter Anwendung des Algorithmus von Böhm.

Die Verringerung der Standardabweichung der neu generierten Kontrollpunkte kann dahingehend erklärt werden, dass eine Standardabweichung eines neuen de Boor-Punktes aus einer unterschiedlichen Gewichtung der Standardabweichungen zweier benachbarter Kontrollpunkte resultiert. Die Multiplikation der mit dem Wert von 1 mm angenommenen Varianzen mit Werten < 0 bildet Varianzen, die kleiner ausfallen als die Ausgangsgröße.

$$\mathbf{P}_\rho = \begin{pmatrix}
 & \mathbf{q}_0 & \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 & \mathbf{q}_4 & \mathbf{q}_5 & \mathbf{q}_6 & \mathbf{q}_7 & \mathbf{q}_8 & \mathbf{q}_9 & \mathbf{q}_{10} & \mathbf{q}_{11} \\
 \mathbf{q}_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \mathbf{q}_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \mathbf{q}_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \mathbf{q}_3 & 0 & 0 & 0 & 1 & 0,4472 & 0,2357 & 0,2357 & 0 & 0 & 0 & 0 & 0 \\
 \mathbf{q}_4 & 0 & 0 & 0 & 0,4472 & 1 & 0,9487 & 0,9487 & 0,8 & 0 & 0 & 0 & 0 \\
 \mathbf{q}_5 & 0 & 0 & 0 & 0,2357 & 0,9487 & 1 & 1 & 0,9487 & 0,2357 & 0 & 0 & 0 \\
 \mathbf{q}_6 & 0 & 0 & 0 & 0,2357 & 0,9487 & 1 & 1 & 0,9487 & 0,2357 & 0 & 0 & 0 \\
 \mathbf{q}_7 & 0 & 0 & 0 & 0 & 0,8 & 0,9487 & 0,9487 & 1 & 0,4472 & 0 & 0 & 0 \\
 \mathbf{q}_8 & 0 & 0 & 0 & 0 & 0 & 0,2357 & 0,2357 & 0,4472 & 1 & 0 & 0 & 0 \\
 \mathbf{q}_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 \mathbf{q}_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 \mathbf{q}_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix} \quad [12 \times 12] \quad (\text{A.26})$$

Die Berechnung der Korrelationsmatrix $\mathbf{P}_\rho$ in Gleichung A.26 nach Gleichung A.10 verdeutlicht die Abhängigkeiten der einzelnen neu entstandenen Kontrollpunkte untereinander. So weisen etwa die Kontrollpunkte $\mathbf{q}_5$ und $\mathbf{q}_6$ Korrelationswerte vom Wert 1 in Folge 100% Korrelation auf, da beide Punkte die gleiche Position inne haben. Ebenfalls starke stochastische Abhängigkeiten zeigen mit einem Korrelationswert von 0,95 die Kontrollpunkte $\mathbf{q}_4$ und $\mathbf{q}_5$, $\mathbf{q}_5$ und $\mathbf{q}_7$, $\mathbf{q}_4$ und $\mathbf{q}_6$ sowie $\mathbf{q}_6$ und $\mathbf{q}_7$. Diese Kontrollpunkte liegen entweder unmittelbar nebeneinander oder dienen bei der Anwendung des Algorithmus von Böhm zur Berechnung des neuen Kontrollpunktes. Je größer der Abstand eines Kontrollpunktes im Kontrollpolygon zum neu einzufügenden Kontrollpunkt besteht, desto geringer ist der lokale Einfluss und desto weniger stochastische Abhängigkeiten kommen in der dargestellten

Korrelationsmatrix zum Vorschein. Demzufolge sind zwischen den Kontrollpunkten $\mathbf{q}_4$ und $\mathbf{q}_3$ sowie $\mathbf{q}_7$ und $\mathbf{q}_8$ lediglich Korrelationen von 0,45 gegeben. $\mathbf{q}_3$ und $\mathbf{q}_8$ weisen nur noch geringfügige stochastische Abhängigkeiten auf, während die Kontrollpunkte $\mathbf{q}_0$ bis $\mathbf{q}_2$ und $\mathbf{q}_9$ bis $\mathbf{q}_{11}$ keinerlei Abhängigkeiten von den anderen Kontrollpunkten zeigen.

Fraglich ist bis zu dieser Stelle noch, ob die Wahl eines anderen Wertes für den neu einzufügenden Knoten in Relation zum jeweiligen Median des Knotenvektors eventuell größere Standardabweichungen hervorbringt und, ob es Unterschiede in der Form der zu bildenden B-Spline-Kurve durch die Wahl des neuen Knotens geben kann. Beide Aspekte werden in den nächsten Abschnitten behandelt.

Überprüfung des Einflusses des neuen Knotens auf die Standardabweichung

Um zu überprüfen, in welcher Art die Wahl des Wertes des einzufügenden Knotens $\bar{u}$ Einfluss auf die Standardabweichung der Kontrollpunkte bei Anwendung des Algorithmus von Böhm hat, werden im Folgenden unterschiedliche Faktoren für die einzufügenden Knoten verwendet. Wie in Abbildung A.5 einsehbar ist, werden für die einzufügenden Knoten $\bar{u}$ Werte zwischen 1 und 5 für den Knotenvektor

$$\mathbf{U} = (0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6) \quad (\text{A.27})$$

herangezogen. Die Grafik A.5 zeigt die Standardabweichungen der jeweiligen Kontrollpunkte der B-Spline-Kurve von Abbildung A.1. Zu beobachten ist, dass sich entsprechend des Verhaltens der B-Spline-Basisfunktionen ein symmetrisches Abbild der Standardabweichungen ergibt. Je größer der Wert des Knotens $\bar{u}$ gewählt wird, desto weiter verschiebt sich der Einfluss durch die Veränderung im Knotenvektor auf Kontrollpunkte im hinteren Bereich des Kontrollpolygons.

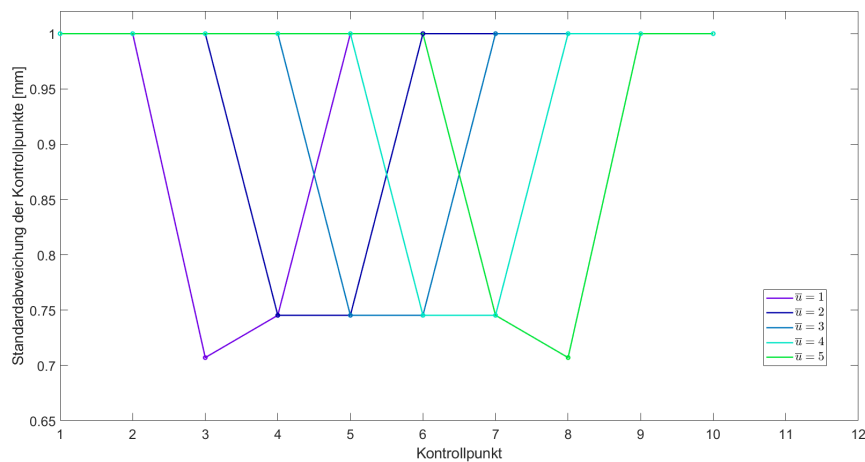


Abb. A.5: Standardabweichungen der Kontrollpunkte bei einmaliger Ausführung des Algorithmus von Böhm.

Dieses Verhalten spiegelt in Abbildung A.6 und A.7 durch zweimalige und dreimalige Anwendung des Algorithmus von Böhm, bei der erneut zusätzliche Knoten eingefügt werden, wieder.

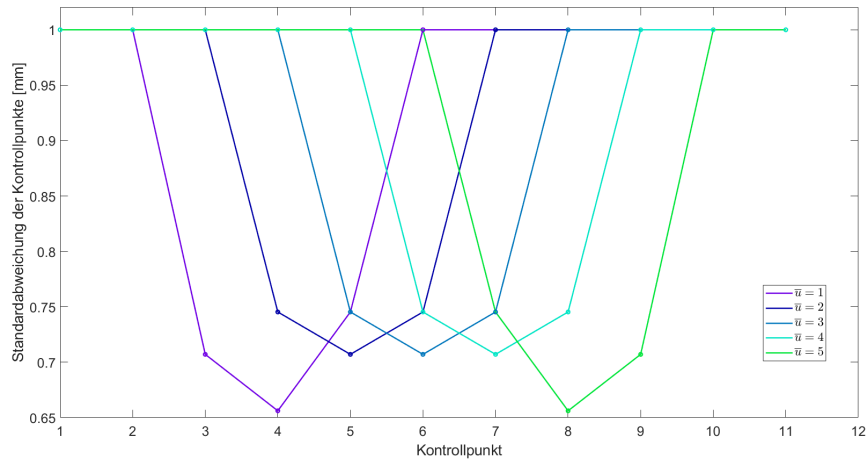


Abb. A.6: Standardabweichungen der Kontrollpunkte bei zweimaliger Ausführung des Algorithmus von Böhm.

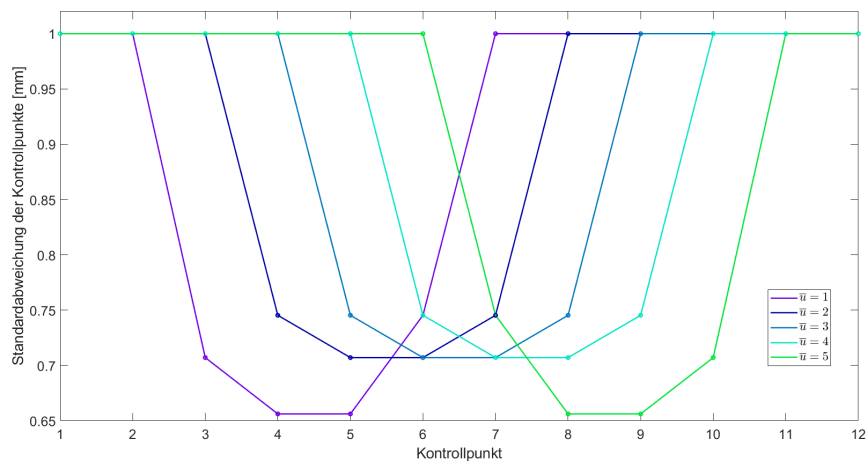


Abb. A.7: Standardabweichungen der Kontrollpunkte bei dreimaliger Ausführung des Algorithmus von Böhm.

Mittels der vorangegangenen Berechnungen kann gezeigt werden, dass sich die Standardabweichungen der Kontrollpunkte der B-Spline-Kurve durch die Anwendung des Algorithmus von Böhm nicht erhöhen, sondern im Gegenteil im jeweiligen Bereich des Einflusses des Knoteneinfügens sogar abnehmen. Da die Unterschiede der resultierenden Standardabweichungen auf die Kontrollpunkte je nach Knotenposition des einzufügenden Knotens mit maximal 0,06 mm gering ausfallen, wird für die Anwendung des Algorithmus von Böhm für die einzufügende Knotenposition die mittlere Position des Knotenvektors verwendet.

Numerischer Nachweis betreffend Kurvenänderung

Um zu überprüfen, ob die aus den veränderten Kontrollpolygonen zu generierenden B-Spline-Kurven in ihrer Form noch der ursprünglichen B-Spline-Kurve entsprechen, werden aus den de Boor-Polygonen, die in Folge unterschiedlicher einzufügender Knoten wie auch Durchgängen im Algorithmus von Böhm entstanden sind, B-Spline-Kurven dritten Grades erzeugt und in Relation zur originären B-Spline-Kurve vom Grad 3 gesetzt.

	Eingefügte Knotenanzahl	Maximale Abweichung zur ursprünglichen B-Spline-Kurve - X [mm]				
		$\bar{u} = 1$	$\bar{u} = 2$	$\bar{u} = 3$	$\bar{u} = 4$	$\bar{u} = 5$
Durchgang 1	1	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$3,55 \cdot 10^{-15}$
Durchgang 2	2	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$2,66 \cdot 10^{-15}$	$3,55 \cdot 10^{-15}$
Durchgang 3	3	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$2,66 \cdot 10^{-15}$	$3,55 \cdot 10^{-15}$

Tab. A.1: Differenz der X-Koordinaten der mittels Algorithmus von Böhm generierten B-Spline-Kurven in Relation zur ursprünglichen B-Spline-Kurve.

Die zwei Tabellen A.1 und A.2 stellen die maximale absolute Abweichung der jeweiligen berechneten B-Spline-Kurve zur ursprünglichen B-Spline-Kurve in der x- sowie y-Koordinate dar.

	Eingefügte Knotenanzahl	Maximale Abweichung zur ursprünglichen B-Spline-Kurve - Y [mm]				
		$\bar{u} = 1$	$\bar{u} = 2$	$\bar{u} = 3$	$\bar{u} = 4$	$\bar{u} = 5$
Durchgang 1	1	$2,66 \cdot 10^{-15}$	$2,66 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,33 \cdot 10^{-15}$
Durchgang 2	2	$2,66 \cdot 10^{-15}$	$2,66 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$8,88 \cdot 10^{-16}$
Durchgang 3	3	$2,66 \cdot 10^{-15}$	$2,66 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,78 \cdot 10^{-15}$	$1,33 \cdot 10^{-15}$

Tab. A.2: Differenz der Y-Koordinaten der mittels Algorithmus von Böhm generierten B-Spline-Kurven in Relation zur ursprünglichen B-Spline-Kurve.

Anhand der Differenzen beider Tabellen, kann festgestellt werden, dass zwischen den Kurvenpunkten der in Relation gesetzten B-Spline-Kurven, Abweichungen im Bereich von $10 \cdot 10^{-15}$ mm existieren und demnach von einer nahezu identischen Kurvenform ausgegangen werden kann. Die Abweichungen der Kurvenpunkte, die mit Hilfe eines neuen Knotens mit dem Wert von $\bar{u} = 3$ bestimmt wurden, weisen sowohl in den Differenzen der x-Koordinaten wie auch der y-Koordinaten sehr niedrige Werte auf. In Einbeziehung dieser Resultate wie auch der Ergebnisse der vergleichenden Anschauung der Standardabweichungen der Kontrollpunkte bei unterschiedlichen neuen Knoten, wird für die Anwendung des Algorithmus von Böhm in Unterkapitel 4.2 der Median des jeweiligen Knotenvektors zur Generierung eines Trägervektors mit einem zusätzlichen Träger verwendet. Für die Berechnung der Kovarianzmatrix im 3D-Fall werden demzufolge ebenso die Mediane der Knotenvektoren herangezogen.

Beispielhafte Berechnungen für den 3D-Fall

Nachdem die Kovarianzmatrix der neuen Kontrollpunkte nach Ausführung des Algorithmus von Böhm anhand eines 2D-Falls bestimmt wurde, soll im Folgenden die Kovarianzmatrix eines Kontrollgitters ermittelt werden, das als Basis zur Generierung einer B-Spline-Flächen dient.

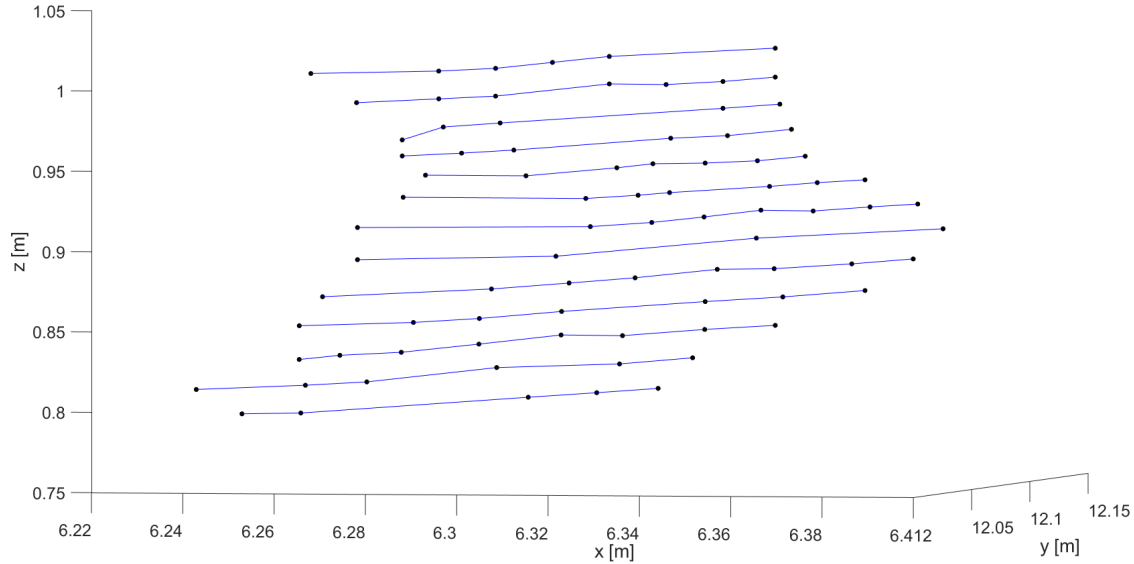


Abb. A.8: Kontrollpolygone.

Hierzu werden 13 Kontrollpolygone unterschiedlicher Punktzahl als Grundlage in einer Matrix $\mathbf{D}$ aufgestellt. Da das längste de Boor-Polygon aus 8 Kontrollpunkten besteht, hat die in Gleichung A.28 dargestellte Matrix eine Dimension von $[8 \times 13]$.

$$\mathbf{D} = \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} & d_{1,6} & d_{1,7} & d_{1,8} & d_{1,9} & d_{1,10} & d_{1,11} & d_{1,12} & d_{1,13} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} & d_{2,6} & d_{2,7} & d_{2,8} & d_{2,9} & d_{2,10} & d_{2,11} & d_{2,12} & d_{2,13} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} & d_{3,6} & d_{3,7} & d_{3,8} & d_{3,9} & d_{3,10} & d_{3,11} & d_{3,12} & d_{3,13} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} & d_{4,6} & d_{4,7} & d_{4,8} & d_{4,9} & d_{4,10} & d_{4,11} & d_{4,12} & d_{4,13} \\ d_{5,1} & d_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} & \text{NaN} & d_{5,7} & d_{5,8} & d_{5,9} & d_{5,10} & d_{5,11} & d_{5,12} & d_{5,13} \\ \text{NaN} & d_{6,2} & d_{6,3} & d_{6,4} & d_{6,5} & \text{NaN} & d_{6,7} & d_{6,8} & d_{6,9} & d_{6,10} & \text{NaN} & d_{6,12} & d_{6,13} \\ \text{NaN} & \text{NaN} & d_{7,3} & d_{7,4} & d_{7,5} & \text{NaN} & d_{7,7} & d_{7,8} & d_{7,9} & \text{NaN} & \text{NaN} & d_{7,12} & \text{NaN} \\ \text{NaN} & \text{NaN} & d_{7,3} & \text{NaN} & d_{7,5} & \text{NaN} & d_{7,7} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} & \text{NaN} \end{pmatrix} \quad (\text{A.28})$$

Nach Abzug der nicht belegten Werte (NaN) in der Matrix, wird die Kovarianzmatrix der alten Kontrollpunkte auf Basis der Varianzen von 84 de Boor-Punkten jeweils mit einem Wert von 1 mm aufgestellt. Die finale Kovarianzmatrix der neuen Kontrollpunkte, besteht in Folge der 13 Kontrollpolygone mit einer Maximallänge von 8 de Boor-Punkten aus $8 \cdot 13 = 104$ Varianzen der neuen Kontrollpunkte. Bedingt durch die 20 nicht belegten Werte in der Matrix $\mathbf{D}$, ist der Algorithmus von Böhm 20 mal auszuführen, um die fehlenden Kontrollpunkte zu generieren. Eine Kovarianzfortpflanzung ist somit 20 mal durchzuführen.

Abbildung A.9 stellt die resultierenden Standardabweichungen jeder der 104 Kontrollpunkte dar. Die vertikalen grau gestrichelten Linien markieren die Unterscheidung in einzelne Kontrollpolygone. Zu beobachten ist, dass eine Verringerung der Standardabweichung mit der Anzahl der einzufügenden

Knoten einhergeht, so dass diese Kontrollpolygone mit den meisten einzufügenden Kontrollpunkten die de Boor-Punkte mit den geringsten Standardabweichungen enthalten. Da neue Knoten jeweils mittig in das Polygon eingefügt werden, äußert sich die Standardabweichung des zusätzlichen Kontrollpunktes durch mittige Ausschläge.

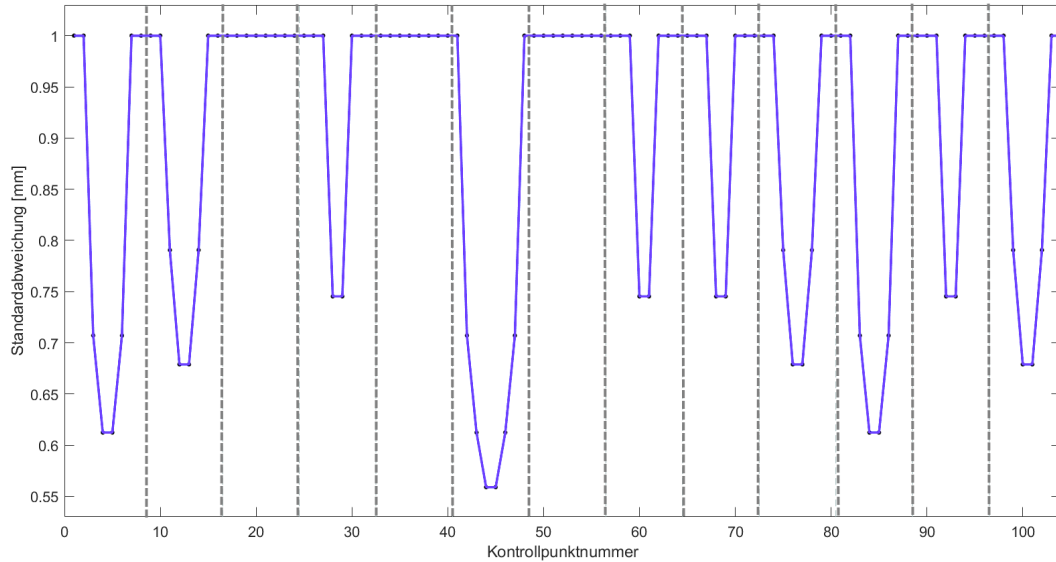


Abb. A.9: Standardabweichungen der Kontrollpunkte eines Kontrollgitters nach Anwendung des Algorithmus von Böhm.

Abbildung A.10 zeigt die Befüllung der Korrelationsmatrix mit Werten ungleich 0. Eine Lesbarkeit einzelner Werte in der Matrix ist nicht vorgesehen, da lediglich die Abhängigkeiten bzw. Unabhängigkeiten der Kontrollpunkte zwischen den Kontrollpolygonen verdeutlicht werden sollen. Die Darstellung veranschaulicht die stochastische Unabhängigkeit der Kontrollpunkte zwischen den einzelnen Kontrollpolygonen. Diese Verhaltensweise ist schlüssig, da zur Erhaltung der Form einer B-Spline-Kurve bzw. -Fläche lediglich im jeweiligen Kontrollpolygon einzelne Punkte eingefügt werden müssen. Die Beibehaltung der Flächenform benötigt demgemäß keine Interaktion zwischen den einzelnen de Boor-Polygonen.

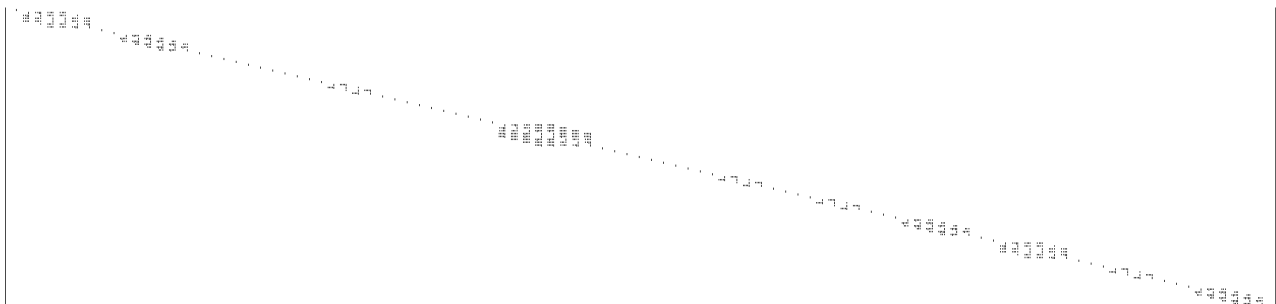


Abb. A.10: Befüllung der Korrelationsmatrix mit Werten ungleich 0 eines Kontrollgitters nach Anwendung des Algorithmus von Böhm.

A.2 Varianz-Kovarianzfortpflanzung für B-Spline-Kurvenpunkte

Für die Berechnung der Varianz-Kovarianzmatrix der B-Spline-Kurvenpunkte wird für das funktionale Modell Gleichung 2.24 von Abschnitt 2.2 herangezogen:

$$C(t) = \sum_{i=0}^m d_i \cdot N_i^n(t), \quad t \in [u_0, u_k] = [u_n, u_{k-n}].$$

Die Jacobi-Matrix kann daraufhin mittels der partiellen Ableitungen entsprechend Gleichung A.29, bei der von einem Kurvengrad von 3 für die B-Spline-Basisfunktionen ausgegangen wird, aufgestellt werden:

$$F = \begin{pmatrix} \frac{\partial C(u_0)}{\partial d_0} & \frac{\partial C(u_0)}{\partial d_1} & \cdots & \frac{\partial C(u_0)}{\partial d_m} \\ \frac{\partial C(u_1)}{\partial d_0} & \frac{\partial C(u_1)}{\partial d_1} & \cdots & \frac{\partial C(u_1)}{\partial d_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C(u_k)}{\partial d_0} & \frac{\partial C(u_k)}{\partial d_1} & \cdots & \frac{\partial C(u_k)}{\partial d_m} \end{pmatrix} = \begin{pmatrix} N_0^3(u_0) & N_1^3(u_0) & \cdots & N_m^3(u_0) \\ N_0^3(u_1) & N_1^3(u_1) & \cdots & N_m^3(u_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_0^3(u_k) & N_1^3(u_k) & \cdots & N_m^3(u_k) \end{pmatrix} \quad (\text{A.29})$$

Mit Hilfe der Varianz-Kovarianzmatrix der Kontrollpunkte Σ_d sowie der Jacobi-Matrix kann mittels Kovarianzfortpflanzungsgesetz die Varianz-Kovarianzmatrix der B-Spline-Kurvenpunkte Σ_S berechnet werden:

$$\Sigma_C = F \cdot \Sigma_d \cdot F^T. \quad (\text{A.30})$$

Zu beachten ist, dass die Größe der entstehenden Varianz-Kovarianzmatrix Σ_C von der Anzahl der B-Spline-Kurvenpunkte abhängig ist. Die Anzahl der Kurvenpunkte ist wiederum abhängig von der festgelegte Schrittweite.

A.3 Varianz-Kovarianzfortpflanzung für B-Spline-Flächenpunkte

Für die Berechnung der Varianz-Kovarianzmatrix der B-Spline-Flächenpunkte wird für das funktionale Modell Gleichung 2.31 herangezogen:

$$\mathbf{S}(t_1, t_2) = \sum_{i=0}^m \sum_{j=0}^a N_i^n(t_1) \cdot N_j^e(t_2) \cdot \mathbf{d}_{i,j}, \quad (t_1, t_2) \in [u_0, u_k] \times [v_0, v_r]$$

Die Jacobi-Matrix kann daraufhin mittels der partiellen Ableitungen entsprechend Gleichung A.31, bei der von einem Kurvengrad von 3 für die B-Spline-Basisfunktionen ausgegangen wird, aufgestellt werden:

$$F = \begin{pmatrix} \frac{\partial S(u_0, v_0)}{\partial d_{0,0}} & \frac{\partial S(u_0, v_0)}{\partial d_{1,0}} & \cdots & \frac{\partial S(u_0, v_0)}{\partial d_{m_d,0}} & \frac{\partial S(u_0, v_0)}{\partial d_{0,1}} & \cdots & \frac{\partial S(u_0, v_0)}{\partial d_{m_d,a_d}} \\ \frac{\partial S(u_0, v_1)}{\partial d_{0,0}} & \frac{\partial S(u_0, v_1)}{\partial d_{1,0}} & \cdots & \frac{\partial S(u_0, v_1)}{\partial d_{m_d,0}} & \frac{\partial S(u_0, v_1)}{\partial d_{0,1}} & \cdots & \frac{\partial S(u_0, v_1)}{\partial d_{m_d,a_d}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S(u_0, v_r)}{\partial d_{0,0}} & \frac{\partial S(u_0, v_r)}{\partial d_{1,0}} & \cdots & \frac{\partial S(u_0, v_r)}{\partial d_{m_d,0}} & \frac{\partial S(u_0, v_r)}{\partial d_{0,1}} & \cdots & \frac{\partial S(u_0, v_r)}{\partial d_{m_d,a_d}} \\ \frac{\partial S(u_1, v_0)}{\partial d_{0,0}} & \frac{\partial S(u_1, v_0)}{\partial d_{1,0}} & \cdots & \frac{\partial S(u_1, v_0)}{\partial d_{m_d,0}} & \frac{\partial S(u_1, v_0)}{\partial d_{0,1}} & \cdots & \frac{\partial S(u_1, v_0)}{\partial d_{m_d,a_d}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S(u_1, v_r)}{\partial d_{0,0}} & \frac{\partial S(u_1, v_r)}{\partial d_{1,0}} & \cdots & \frac{\partial S(u_1, v_r)}{\partial d_{m_d,0}} & \frac{\partial S(u_1, v_r)}{\partial d_{0,1}} & \cdots & \frac{\partial S(u_1, v_r)}{\partial d_{m_d,a_d}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S(u_k, v_r)}{\partial d_{0,0}} & \frac{\partial S(u_k, v_r)}{\partial d_{1,0}} & \cdots & \frac{\partial S(u_k, v_r)}{\partial d_{m_d,0}} & \frac{\partial S(u_k, v_r)}{\partial d_{0,1}} & \cdots & \frac{\partial S(u_k, v_r)}{\partial d_{m_d,a_d}} \end{pmatrix}$$

$$= \begin{pmatrix} N_0^3(u_0) \cdot N_0^3(v_0) & N_1^3(u_0) \cdot N_0^3(v_0) & \cdots & N_{m_d}^3(u_0) \cdot N_0^3(v_0) & N_0^3(u_0) \cdot N_1^3(v_0) & \cdots & N_{m_d}^3(u_0) \cdot N_{a_d}^3(v_0) \\ N_0^3(u_0) \cdot N_0^3(v_1) & N_1^3(u_0) \cdot N_0^3(v_1) & \cdots & N_{m_d}^3(u_0) \cdot N_0^3(v_1) & N_0^3(u_0) \cdot N_1^3(v_1) & \cdots & N_{m_d}^3(u_0) \cdot N_{a_d}^3(v_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ N_0^3(u_0) \cdot N_0^3(v_r) & N_1^3(u_0) \cdot N_0^3(v_r) & \cdots & N_{m_d}^3(u_0) \cdot N_0^3(v_r) & N_0^3(u_0) \cdot N_1^3(v_r) & \cdots & N_{m_d}^3(u_0) \cdot N_{a_d}^3(v_r) \\ N_0^3(u_1) \cdot N_0^3(v_0) & N_1^3(u_1) \cdot N_0^3(v_0) & \cdots & N_{m_d}^3(u_1) \cdot N_0^3(v_0) & N_0^3(u_1) \cdot N_1^3(v_0) & \cdots & N_{m_d}^3(u_1) \cdot N_{a_d}^3(v_0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ N_0^3(u_1) \cdot N_0^3(v_r) & N_1^3(u_1) \cdot N_0^3(v_r) & \cdots & N_{m_d}^3(u_1) \cdot N_0^3(v_r) & N_0^3(u_1) \cdot N_1^3(v_r) & \cdots & N_{m_d}^3(u_1) \cdot N_{a_d}^3(v_r) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ N_0^3(u_k) \cdot N_0^3(v_r) & N_1^3(u_k) \cdot N_0^3(v_r) & \cdots & N_{m_d}^3(u_k) \cdot N_0^3(v_r) & N_0^3(u_k) \cdot N_1^3(v_r) & \cdots & N_{m_d}^3(u_k) \cdot N_{a_d}^3(v_r) \end{pmatrix} \quad (\text{A.31})$$

Durch die Varianz-Kovarianzmatrix der neu erzeugten Kontrollpunkte Σ_q sowie der Jacobi-Matrix kann mittels Kovarianzfortpflanzungsgesetz die Varianz-Kovarianzmatrix der B-Spline-Flächenpunkte Σ_S berechnet werden:

$$\Sigma_S = \mathbf{F} \cdot \Sigma_q \cdot \mathbf{F}^T. \quad (\text{A.32})$$

Zu beachten ist, dass die Größe der entstehenden Varianz-Kovarianzmatrix Σ_S von der Anzahl der B-Spline-Flächenpunkte abhängig ist. Die Anzahl der Flächenpunkte ist wiederum abhängig von der festgelegte Schrittweite.

A.4 Beispielrechnung zur Kreuzkovarianzfunktion (2D)

Krümmungssignal κ_1 , $n_1 = 100$

Krümmungssignal κ_2 , $n_2 = 60$

$$k = -6, -5, \dots, 0, \dots, 46$$

$$\begin{aligned}
 k \leq 0 : \quad Cov_{\kappa_1, \kappa_2}(-6) &= \frac{1}{52} \sum_{j=1}^{54} (\kappa_1(j) - \overline{\kappa_1})(\kappa_2(j+6) - \overline{\kappa_2}) \\
 &= \frac{1}{52} ((\kappa_1(1) - \overline{\kappa_1})(\kappa_2(7) - \overline{\kappa_2}) + \dots + (\kappa_1(54) - \overline{\kappa_1})(\kappa_2(60) - \overline{\kappa_2})) \\
 &\quad \vdots \\
 Cov_{\kappa_1, \kappa_2}(0) &= \frac{1}{58} \sum_{j=1}^{60} (\kappa_1(j) - \overline{\kappa_1})(\kappa_2(j) - \overline{\kappa_2}) \\
 &= \frac{1}{58} ((\kappa_1(1) - \overline{\kappa_1})(\kappa_2(1) - \overline{\kappa_2}) + \dots + (\kappa_1(60) - \overline{\kappa_1})(\kappa_2(60) - \overline{\kappa_2})) \\
 &\quad \vdots \\
 0 < k \leq 40 : \quad Cov_{\kappa_1, \kappa_2}(1) &= \frac{1}{58} \sum_{j=2}^{61} (\kappa_1(j) - \overline{\kappa_1})(\kappa_2(j-1) - \overline{\kappa_2}) \\
 &= \frac{1}{58} ((\kappa_1(2) - \overline{\kappa_1})(\kappa_2(1) - \overline{\kappa_2}) + \dots + (\kappa_1(61) - \overline{\kappa_1})(\kappa_2(60) - \overline{\kappa_2})) \\
 &\quad \vdots \\
 Cov_{\kappa_1, \kappa_2}(40) &= \frac{1}{58} \sum_{j=41}^{100} (\kappa_1(j) - \overline{\kappa_1})(\kappa_2(j-40) - \overline{\kappa_2}) \\
 &= \frac{1}{58} ((\kappa_1(41) - \overline{\kappa_1})(\kappa_2(1) - \overline{\kappa_2}) + \dots + (\kappa_1(100) - \overline{\kappa_1})(\kappa_2(60) - \overline{\kappa_2})) \\
 &\quad \vdots \\
 k > 40 : \quad Cov_{\kappa_1, \kappa_2}(41) &= \frac{1}{57} \sum_{j=42}^{100} (\kappa_1(j) - \overline{\kappa_1})(\kappa_2(j-41) - \overline{\kappa_2}) \\
 &= \frac{1}{57} ((\kappa_1(42) - \overline{\kappa_1})(\kappa_2(1) - \overline{\kappa_2}) + \dots + (\kappa_1(100) - \overline{\kappa_1})(\kappa_2(59) - \overline{\kappa_2})) \\
 &\quad \vdots \\
 Cov_{\kappa_1, \kappa_2}(46) &= \frac{1}{52} \sum_{j=47}^{100} (\kappa_1(j) - \overline{\kappa_1})(\kappa_2(j-46) - \overline{\kappa_2}) \\
 &= \frac{1}{52} ((\kappa_1(47) - \overline{\kappa_1})(\kappa_2(1) - \overline{\kappa_2}) + \dots + (\kappa_1(100) - \overline{\kappa_1})(\kappa_2(54) - \overline{\kappa_2}))
 \end{aligned}$$

Danksagung

An dieser Stelle möchte ich meinem Doktorvater Prof. Dr.-Ing. habil. Dr. h.c. Volker Schwieger für seine Betreuung und seine stetige Unterstützung danken. Hierbei bin ich besonders dankbar für das entgegengebrachte Vertrauen und die Möglichkeit diese Arbeit nebenberuflich durchführen zu dürfen. Danke, Volker für die unzähligen aufschlussreichen Diskussionen, die Anregungen und die Flexibilität bei den Terminfestlegungen. Als Nächstes möchte ich mich bei meiner Gutachterin Prof. Dr. Corinna Harmening für die freundliche Übernahme des Korreferats und sowie den anregenden Austausch in Sachen B-Spline-Technik bedanken.

Für die sehr angenehme und familiäre Atmosphäre, die ich am Institut für Ingenieurgeodäsie erfahren habe, möchte ich mich bei allen Kolleginnen und Kollegen herzlich bedanken. Hervorzuheben sind Dr.-Ing. Laura Balangé und Dipl.-Ing. Lyudmila Gorokhova, die mich bei der TLS-Messung der Testobjekte im Messkeller sowie der Nachbearbeitung der Punktwolken unterstützt haben.

In besonderem Maße danke ich meinem Partner Fabian Kretschmaier für sein unendliches Verständnis, seine stetige Rückendeckung und seine ermutigenden Worte. Weiter möchte ich meinen Eltern Gabriele und Peter und meiner Schwester Mona für die Ermöglichung der Rahmenbedingungen und Ihre beständige Unterstützung danken.

Lebenslauf

Persönliche Daten:

Name	Julia Jasmin Aichinger
Geburtsdatum	15.10.1991
Geburtsort	Stuttgart Bad-Cannstatt, Deutschland

Beruf:

2023 - heute	Geschäftsführende Partnerin, Öffentlich bestellte Vermessungsingenieurin AICHINGER - Ingenieurbüro für Vermessung PartG
2019 - 2023	Vermessungsingenieurin, Vermessungsassessorin AICHINGER - Ingenieurbüro für Vermessung

Ausbildung:

2017 - 2026	Promotion Institut für Ingenieurgeodäsie (IIGS), Universität Stuttgart
2017 - 2019	Referendariat Höherer vermessungstechnischer Verwaltungsdienst, Große Staatsprüfung Ministerium für ländlichen Raum und Verbraucherschutz Baden-Württemberg
2015 - 2017	Masterstudium Geodäsie und Geoinformatik, Universität Stuttgart
2011 - 2015	Bachelorstudium Geodäsie und Geoinformatik, Universität Stuttgart
Juni 2011	Abitur Wirtemberg-Gymnasium, Stuttgart-Untertürkheim