
Westfälische Wilhelms-Universität Münster

Masterarbeit

Optical Flow Approximation mit Hilfe
multivariater n -dimensionaler Thin-Plate Spline
Interpolation

Optical Flow Approximation Using Multivariate n -dimensional Thin-Plate Spline Interpolation

Oliver Schmitz

Matr. Nr.: 405 905

Betreuung und Erstgutachter: Prof. Dr. Benedikt Wirth
Zweitgutachter: Dr. Frank Wübbeling
eingereicht am: 26. Februar 2021

Zusammenfassung

In der folgenden Arbeit wird, vor dem Hintergrund einer medizinischen Anwendung, ein gradientenbasiertes Verfahren zur Erschließung von *Optical Flow* Feldern anhand von wenigen unregelmäßig verteilten Punkten vorgestellt. Dabei wird die lokale Schlechtgestellttheit von *Optical Flow* Problemen durch die *Brightness Constancy Assumption* und *Thin-Plate Spline* Interpolation überwunden.

Einführend wird eine Übersicht über *Optical Flow* Problemstellungen erarbeitet, sowie die klassische *Thin-Plate Spline* Interpolation vorgestellt. Der Hauptteil beschäftigt sich mit einer angepassten Version der *Thin-Plate Spline* Interpolation, bei der die Interpolationsbedingungen durch die *Optical Flow Constraints* gegeben sind. Die Formulierung des mathematischen Problems erfolgt durch die Minimierung eines Funktional im Distributionenraum nach dem Vorbild von Meinguet [14]. Die Existenz einer eindeutigen Lösung des n-dimensionalen Problems wird bewiesen. Hierbei wird zwischen einem exakten und inexakten Verfahren unterschieden, also solches in dem die Interpolationsbedingung streng erfüllt werden müssen, bzw. lediglich als Regularisierungsterm Einfluss nehmen. Die erschlossene Theorie wird durch Angabe der Algorithmen, Konditionsuntersuchung und Ergebnisanalyse ergänzt. Letztere erfolgt anhand konstruierten Daten, sowie mittels eines Middelbury *Optical Flow* Datensatzes [15]. Während der theoretische Anteil der Arbeit den multivariaten n-dimensionalen Fall behandelt, beschränken sich die konkreten Lösungsverfahren auf den anschaulichen zweidimensionalen Fall.

Inhaltsverzeichnis

1	Motivation	1
1.1	Medizinischer Hintergrund	1
1.2	<i>Adaptive Optics</i>	2
2	Einführung <i>Optical Flow</i>	2
2.1	Grundlagen <i>Optical Flow</i>	3
2.2	<i>Optical Flow Constraint</i>	4
2.3	Lokale Lösungsverfahren	6
2.3.1	Das Aperture-Problem	7
2.4	Globale Lösungsverfahren	8
3	<i>Thin-Plate Spline</i> Interpolation	9
3.1	Klassische TPS Interpolation	9
3.2	TPS Interpolation entlang der Koordinatenachsen	10
4	Meinguets multivariate eindimensionale Interpolation	11
4.1	Mathematische Grundlagen	12
4.2	Multivariate Interpolation nach Meinguet	13
5	Multivariate n-dimensionale Interpolation	15
5.1	Unisolvenz der Punktmenge	16
5.2	Formulierung von Problem (P)	20
5.3	Darstellungsformeln in Beppo Levi Räumen	20
5.4	Vektorwertiger <i>reproducing kernel</i>	25
5.5	Lösung des Problems (P)	27
5.6	Lösen des Gleichungssystems	32
6	Inexaktes Interpolationsproblem	34
7	Matrixformen der Interpolationsprobleme	35
7.1	Exakte Interpolation	35
7.2	Inexakte Interpolation	37
8	Analytische Konditionsuntersuchungen	41
8.1	Kondition	41
8.2	Störung der bezüglich der Zeitableitung	42
8.3	Fehler der gesamten Funktion	46
8.4	Systemmatrix ist singulär	48
9	Algorithmus und Fehlermaße	48
9.1	Diskretisierung	49
9.2	Algorithmus	49
9.3	Fehlermaße	51
9.3.1	Schlechtgestelltheit bei der Berechnung des Fehlers	51
9.3.2	Angular Error (AE)	52
9.3.3	Absolute Endpoint Error (AEE)	53

9.3.4	SSD Interpolation Error (IE)	53
9.3.5	Normierter SSD Interpolation Error (NIE)	53
10	Ergebnisse	53
10.1	Grundlegende Bewegungsformen	54
10.2	Gaussglocke	54
10.2.1	Monotone Gradientenausrichtung	57
10.2.2	Gradient orthogonal zur Bewegung	59
10.2.3	Störung in u_t	60
10.3	Middlebury Datensatz „Hydrangea“	62
10.3.1	Glättung von Daten	63
10.3.2	Verhalten für verschwindende Gradienten	64
11	Fazit	67
A	Programmcode	68
A.1	Exaktes Verfahren	68
A.2	Inexaktes Verfahren	69
A.3	Funktionsauswertung	71

1 Motivation

In diesem einleitenden Abschnitt möchten wir die Problemstellung dieser Arbeit motivieren. Nachdem der medizinische Kontext beleuchtet wird, folgt eine kurze Einführung in den gewählten Lösungsansatz.

1.1 Medizinischer Hintergrund

Mikroskopie an lebenden Organismen (Intravitalmikroskopie) ist ein bildgebendes Verfahren, welches bei der Untersuchung von arteriellen Blutgefäßen mit Einlagerungen wie Cholesterin verwendet wird. Dabei werden weiße Blutkörperchen, oder andere Zellen des blutbildenden Systems, beobachtet, um Zellbewegung, Zellwachstum, Zellsterben sowie Zell-zu-Zell-Interaktionen zu visualisieren. Als Versuchsobjekte dienen häufig Mäuse, bei denen eine kleine Operation durchgeführt und so Zugang zu einer Arterie erlangt wird.

Bei allen Verfahren, welche die Beobachtung von Zellverhalten zum Ziel haben, sind folgende drei grundlegenden Schritte gemein: Markierung der Zellen, Stabilisierung des Gewebes und Analysieren der Daten. Beim Markieren von Zellen handelt es sich um eine Kennzeichnung mit einem fluoreszierenden Material. Dies kann durch Veränderung des genetischen Materials, sodass dieses fluoreszierende Eigenschaften erlangt, oder durch die Zuführung von chemischen Stoffen geschehen. Ohne auf weitere Details der beiden Verfahren einzugehen, wollen wir festhalten, dass nach der Behandlung die markierten Zellen (mit oder ohne Anregung von externen Lichtquellen) im Mikroskop erkennbar sind.

Die beiden Hauptprobleme der Intravitalmikroskopie sind einerseits die Gewinnung von Bildmaterial ohne übermäßigen Schaden am Gewebe auszulösen, andererseits die Vermeidung von Bewegungsartefakten, ausgelöst durch Herzschlag und Atmung der lebenden Maus.

Obwohl bei sogenannten *Laser-Scanning-Mikroskopen*, also solche die ein Gebiet durch Abtastung einzelner Punkte messen, eine Abtasttiefe von 0.05 cm und somit ein dreidimensionales Bild möglich ist, werden operative Eingriffe benötigt, um Zugang zu Arterien zu erhalten. Dabei ist es wichtig Verletzungen am Gewebe zu minimieren, um Faktoren wie dessen Durchblutung nicht zu beeinflussen.

Um Bildartefakte, ausgelöst durch die Bewegung der Maus, zu minimieren, werden üblicherweise drei verschiedene Ansätze zur Stabilisierung genutzt: mechanische, *post-processing* und *triggering*. Bei der mechanischen Stabilisierung wird versucht das Gewebe durch mechanische Möglichkeiten, wie z.B. Gewebekleber oder Kompression des Objekts, zu fixieren. Bei *post-processing* Verfahren wird versucht, durch die Sammlung sehr vieler Daten und anschließender Auswahl bestimmter Daten bessere Bilder zu erzielen. Dies kann durch die Wahl einzelner *Frames* oder Pixel, sowie der Modifikation der Bilder durch Verschiebungen bzw. Rotationen geschehen. Schließlich wird bei *Triggering*-Verfahren versucht, die Erfassung des Bildmaterials mit der Bewegung des Objekts zu synchronisieren. Ein Bild soll nur dann entstehen, wenn die Bewegung des Zellverbands minimal ist. Dabei wird ein möglichst regelmäßiger Herzschlag bzw. Atemrhythmus und somit eine gute Betäubung und operationelle Fähigkeiten vorausgesetzt. Ebenso gibt es Verfahren, die eine Kombination aus all diesen Methoden nutzen. [Vgl. 12]

1.2 Adaptive Optics

Motivation für den Inhalt der vorliegenden Arbeit ist ein anderer Lösungsansatz. Durch *Remote Focussing* kann versucht werden, entweder die Position der Probe (also z.B. der Maus), oder die der Linse des Mikroskops, während der Bildnahme so anzupassen, sodass diese entgegengesetzt bzw. synchron zu der Bewegung des Gewebes verläuft. Im Fokus des Mikroskops soll somit immer der relevante Punkt der Probe liegen. Nachteil ist jedoch, dass die Masse der Probe bzw. Linse zu groß für die benötigte Frequenz der Bewegung ist und es erneut zu Abbildungsfehlern kommen kann.

Im vorliegenden Fall soll der Ausgleich hingegen im Inneren des Mikroskops durchgeführt (*adaptive optics*) werden. In diesem wird eine spiegelnde, flexible Membran, die durch elektrische Signale verformt werden kann, entsprechend angepasst. Mit dieser Anpassung kann der Fokus des Mikroskops im Bruchteil einer Sekunde verschoben werden. Interessant ist somit insbesondere, inwiefern die Membran zu welchem Zeitpunkt verformt werden muss.

Der grobe Ablauf einer Aufnahme mit Hilfe dieses Verfahrens sollte dann die folgende Form annehmen.

1. Messung der Geschwindigkeit und Bewegungsrichtung mit Hilfe eines *Optical Flow* Verfahrens in wenigen Punkten auf dem Gebiet.
2. Bestimmung der Geschwindigkeit und Richtung für jeden Punkt auf dem Untersuchungsgebiet mit Hilfe der *Thin-Plate Spline* Interpolation.
3. Berechnung der Position des Gewebes für einen gewünschten Punkt innerhalb eines Zeitintervalls, z.B. Zeit zwischen Herzschlägen/ im Atemrhythmus, mittels der bekannten Richtungsgeschwindigkeit.
4. Bilderfassung für das gewünschte Gebiet entlang der berechneten Position mit angepasster spiegelnder Membran.
5. Ggfs. Verbesserung des Bewegungsprofils durch Hinzunahme von mehr Messpunkten und erneuter Interpolation.

Die ersten beiden Punkte, die Bestimmung der Richtungsgeschwindigkeit und die Interpolation, sind in Wirklichkeit eng miteinander verknüpft. Ein essentieller Teil des Hauptteils beschäftigt sich mit der Herleitung der Theorie, die gradientenbasierte *Optical Flow* Verfahren mit *Thin-Plate Spline* Interpolation verbindet. Die mathematische Herleitung wird hier für einen allgemeinen n -dimensionalen Fall gerechnet. Die Umsetzung als Algorithmus ab Kapitel 9 beschränkt sich auf den zweidimensionalen Fall. Das grundlegende Vorgehen, also die Bestimmung des *Optical Flow* Feldes mit Hilfe der *Thin-Plate Spline* Interpolation, basiert auf der Idee und nach Anleitung von Prof. Wirth.

2 Einführung *Optical Flow*

Die Bestimmung der Geschwindigkeit der Probe ist ein bedeutender Punkt im Ablauf des Verfahrens. Die Bestimmung von Geschwindigkeitsrichtungen anhand von Bildfolgen fällt unter den Begriff des *Optical Flow*.

2.1 Grundlagen Optical Flow

Die Problematik beim *Optical Flow* beschäftigt sich mit Bewegungen von Objekten im (meist) dreidimensionalen Raum. Dabei wird versucht die Informationen, die im zweidimensionalen Bild zu Verfügung stehen, auf die tatsächliche physikalische Bewegung zurückzuführen. Dabei ist zu beachten, dass diese sich signifikant voneinander unterscheiden.

Betrachten wir den optimalen Fall eines solchen Szenarios, die Projektion eines 3D-Bewegungsfeldes auf die 2D-Bildebene (vgl. Abb. 1). Das Fortschreiten des Flugzeugs kommt auf dem Bild (*image domain*) als 2D-Bewegungsfeld zum Vorschein. Dieses 2D-Bewegungsfeld ist für den Betrachter jedoch nicht messbar. Wir können nur auf die Veränderungen der Helligkeit oder Intensität des Bildes in jedem Punkt zurückgreifen, dem *Optical Flow*. Dieses wird durch die Funktion $u(x, t)$, der Intensität an der Stelle $x \in \mathbb{R}^2$ zum Zeitpunkt t beschrieben.

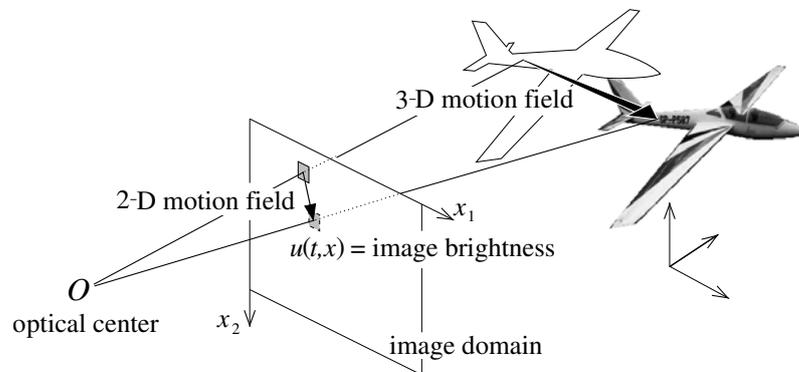


Abbildung 1: Projektion einer Bewegung im dreidimensionalen Raum auf eine Bildebene aus dem Blickwinkel eines Betrachters am optischen Zentrum O . Messbar ist nur die Intensität $u(t, x)$ im Bild. [3]

Den Unterschied zwischen den Bewegungsfeldern und der gemessenen Intensität kann man sich an zwei einfachen Beispielen verdeutlichen. In Abbildung 2a (1) wird auf einer rotierenden Sphäre eine konstante Intensität des Bildes gemessen. Während der *Optical Flow* keine Bewegung anzeigt, befindet sich die Sphäre sowohl im 3D-, als auch im 2D-Bewegungsfeld in Bewegung. Im Gegensatz dazu, wird bei schwenkender Lichtquelle und unbewegter Sphäre eine Veränderung der Intensität gemessen (Abb. 2a (2)). Hier ändert sich die Intensität der Bildpunkte, während keine tatsächliche Veränderung der Position stattfindet.

Abbildung 2b zeigt einen sich gegen den Uhrzeigersinn drehenden Zylinder mit Helixmuster. Das 2D-Bewegungsfeld in (1) orientiert sich nach rechts entlang der x -Achse, während der *Optical Flow* (2) auf der Rolle gerade nach oben, also entlang der y -Achse, verläuft. Für den Betrachter sieht es durch das spiralförmige Muster so aus, als ob sich die Streifen auf dem Zylinder nach oben bewegen. In diesem Fall stimmt also weder die augenscheinliche Bewegung mit der projizierten 2D-Bewegung, noch die 2D-Bewegung mit der tatsächlichen dreidimensionalen Bewegung überein.

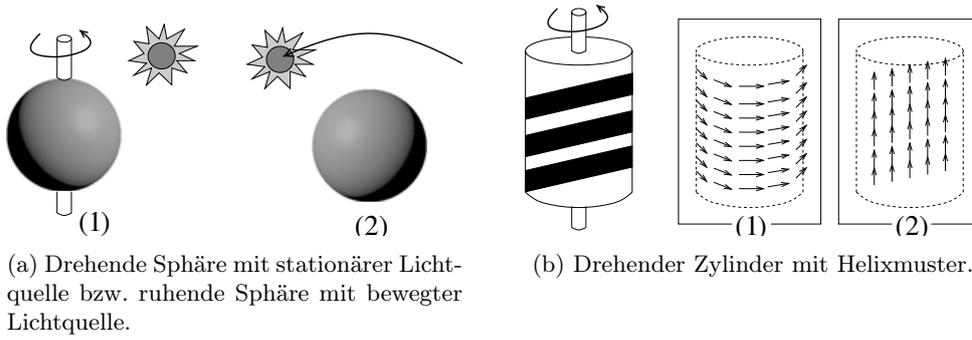


Abbildung 2: Beispiele für die Diskrepanz zwischen *Optical Flow* und den Bewegungsfeldern. [3]

Obwohl der *Optical Flow* und die Bewegungsfelder offensichtlich unterschiedlich sind, teilen sie sich gewisse Eigenschaften, die nützliche Informationen enthalten. Zum Beispiel ist in Abbildung 2b am Rand des Zylinders eine Unstetigkeit im *Optical Flow*, sowie im Bewegungsfeld zu erkennen. Anhand solcher und ähnlicher Informationen versuchen verschiedene Verfahren, Rückschlüsse vom *Optical Flow* auf 2D-, bzw. 3D- Bewegungen zu schließen. Drei grundlegende Kategorien von Verfahren lassen sich hierbei unterscheiden: korrelationsbasierte, featurebasierte und gradientenbasierte Verfahren.

Bei korrelationsbasierten Verfahren wird versucht eine Korrelation zwischen zwei Abbildungen in passenden Intensitätsbereichen herzustellen, wohingegen bei featurebasierten Verfahren der *Optical Flow* an wenigen Stellen mit bestimmten Eigenschaften (*Features*) berechnet wird. Dabei werden in einem ersten Schritt z.B. Kanten oder Ecken identifiziert, welche dann in beiden Bildern verknüpft werden sollen. Die dritte und letzte Methode, die auch in der vorliegenden Arbeit genutzt wird, nutzt hingegen die zeitlichen und räumlichen partiellen Ableitungen der vorliegenden Eingangsbilder, um den *Optical Flow* zu berechnen. [vgl. 3, Kap. 5.3.2, S.250 ff.]

Im folgenden Abschnitt werden die Prinzipien von gradientenbasierten Verfahren erläutern und eine mathematische Grundlage für deren Berechnung schaffen.

2.2 *Optical Flow Constraint*

Sei Ω der Bildbereich und $[0, T]$ das betrachtete Zeitintervall. Wir bezeichnen mit $u(x, t) : \Omega \times [0, T] \rightarrow [0, 1]$ die Funktion der Intensität des Bildes im Punkt $x \in \Omega$ zum Zeitpunkt $t \in [0, T]$. Im Beispiel in Abbildung 1 ist $\Omega \subseteq \mathbb{R}^2$ ein Gebiet in der Ebene, für die Anwendung mit z.B. Laser-Scanning-Mikroskope sind auch Gebiete in \mathbb{R}^3 denkbar.

Im zweidimensionalen Fall ist das gesuchte Geschwindigkeitsfeld $v = (v_1, v_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, also eine Funktion, die einem Punkt $x \in \Omega$ zum Zeitpunkt $t \in [0, T]$ eine Geschwindigkeit $v(x)$ zuordnet.

Ein beliebter Startpunkt für gradientenbasierte Verfahren ist die *Brightness Constancy Assumption*. Diese besagt, dass es angemessen ist anzunehmen, dass sich Bildintensitäten nur an andere Stelle verschieben, jedoch nicht verschwinden oder Neue entstehen. Formal bedeutet dies, dass für einen kleinen Zeitschritt $\delta_t > 0$ mit Taylor-Entwicklung gilt [nach 7, S. 85]:

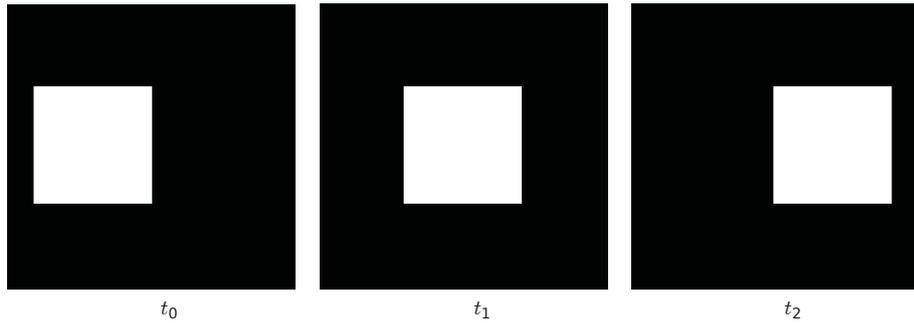
$$\begin{aligned} u(x, t) &= u(x + \delta_t v(x, t), t + \delta_t) \\ &= u(x, t) + \delta_t v(x, t) \bullet \nabla u(x, t) + \delta_t u_t(x, t) + O(\delta_t^2). \end{aligned}$$

Hierbei bezeichnet $\nabla u(x, t) \in \mathbb{R}^2$ den Gradienten von $u(x, t)$ entlang x , $u_t(x, t) = \frac{\partial}{\partial t} u(x, t) \in \mathbb{R}$ die partielle Ableitung in Richtung t und \bullet das Standardskalarprodukt. Subtraktion von $u(x, t)$ und Division durch δ_t liefert

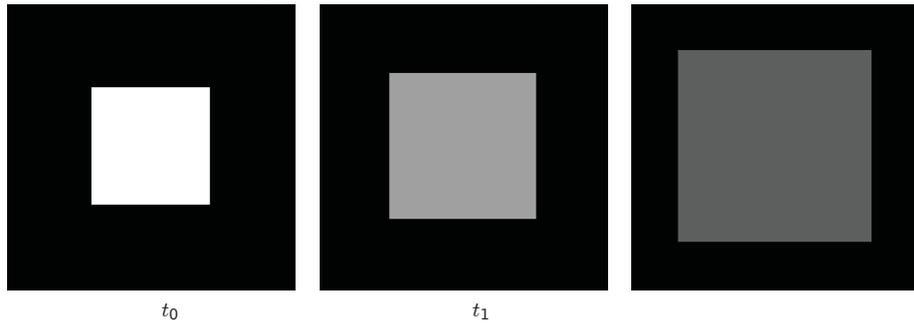
$$0 = v(x, t) \bullet \nabla u(x, t) + u_t(x, t) + O(\delta_t).$$

Für kleine Zeitschritte können wir den Restterm vernachlässigen und erhalten den *Optical Flow Constraint*

$$0 = v(x, t) \bullet \nabla u(x, t) + u_t(x, t). \quad (2.1)$$



(a) Bildfolge eines weißen Blocks mit konstanter Intensität, welcher von links nach rechts wandert. Die *Brightness Constancy Assumption* wird nicht erfüllt



(b) Bildfolge zeigt einen verlaufenden Block. Die Intensität eines Pixels wird somit auf mehrere Pixel aufgeteilt. Die *Brightness Constancy Assumption* wird nicht erfüllt.

Abbildung 3: Beispiele zweier Bildfolgen, die die *Brightness Constancy Assumption* erfüllen bzw. nicht erfüllen. [Aus 7, S. 89]

Ein anderer Ansatz [nach 9], der ebenfalls zum *Optical Flow Constraint* führt, ist die Vorstellung, dass die Intensität des Bildes im Bildraum entlang einer Kurve

$$\vec{x}(t) : [0, T] \rightarrow \Omega, \quad t \mapsto (\vec{x}_1(t), \vec{x}_2(t))$$

verschoben wird. Dies entspricht genau der *Brightness Constancy Assumption*, wir fordern somit einen konstanten Wert $c \in [0, 1]$ entlang der Kurve \vec{x} zu allen Zeitpunkten, formal:

$$u(\vec{x}(t), t) = c, \quad \forall t \in [0, T].$$

Differentiation nach t liefert einerseits

$$\frac{d}{dt}u(\vec{x}(t), t) = \frac{d}{dt}c = 0. \quad (2.2)$$

Mit Hilfe der Kettenregel erhalten wir andererseits

$$\begin{aligned} \frac{d}{dt}u(\vec{x}(t), t) &= \frac{\partial u}{\partial x}(\vec{x}(t), t) \frac{d}{dt}\vec{x}_1(t) + \frac{\partial u}{\partial y}(\vec{x}(t), t) \frac{d}{dt}\vec{x}_2(t) + u_t(\vec{x}(t), t) \\ &= \frac{\partial u}{\partial x}(\vec{x}(t), t)v_1(\vec{x}(t), t) + \frac{\partial u}{\partial y}(\vec{x}(t), t)v_2(\vec{x}(t), t) + u_t(\vec{x}(t), t) \\ &= \nabla u(\vec{x}(t), t) \bullet v(\vec{x}(t), t) + u_t(\vec{x}(t), t), \end{aligned} \quad (2.3)$$

wobei $\frac{d}{dt}\vec{x}_i(t) = v_i(\vec{x}(t), t)$ gilt, da die Ableitung der Kurve genau die Geschwindigkeit in diesem Punkt ist.

Aus Gleichung 2.2 und 2.3 folgt somit der *optical flow constraint* wie in 2.1.

2.1 Definition (INTERPOLATIONSBEDINGUNGEN)

Sei $(a_i)_{i \in I} = A \subset \mathbb{R}^n$ eine endliche Menge von Punkten. Die *Interpolationsbedingungen* zum Zeitpunkt t zum Lösen des *Optical Flow Problems* ist durch den *Optical Flow Constraint* gegeben durch

$$\begin{aligned} v(a_i, t) \bullet \nabla u(a_i, t) + u_t(a_i, t) &= 0, & \forall i \in I \\ \Leftrightarrow v(a_i, t) \bullet \nabla u(a_i, t) &= -u_t(a_i, t), & \forall i \in I \\ \Leftrightarrow v(a_i) \bullet \varphi_i &= \alpha_i, & \forall i \in I \end{aligned}$$

mit $\varphi_i := \nabla u(a_i, t)$ und $\alpha_i := -u_t(a_i, t)$.

Diese Gleichung ist jedoch nicht ausreichend, um bei gegebenen Daten ∇u und u_t die zwei (oder mehr) Unbekannten v_1, v_2 zu berechnen. Es existieren unendlich viele Lösungen. Das Problem ist schlecht gestellt.

2.3 Lokale Lösungsverfahren

Bei lokalen Verfahren wird versucht zusätzliche Informationen aus der direkten Umgebung eines Punktes zu erschließen, um doch eine eindeutige Lösung von (2.1) erhalten. Eine denkbare zusätzliche Annahme hierbei ist, dass in einer kleinen Umgebung um einen Punkt x das Geschwindigkeitsfeld nahezu konstant ist. Somit können die gemessenen Daten in einer Umgebung von x , also die Richtungs- und Zeitableitungen, zur Bestimmung von v genutzt werden. Je nachdem wie groß die Umgebung ist, wird ein Geschwindigkeitsfeld, welches den *Optical Flow Constraint* für jeden Punkt der Umgebung erfüllt, nicht existieren. Somit suchen wir das Geschwindigkeitsfeld welches den *Optical Flow Constraint* z.B. im kleinsten Quadrat minimiert. Minimiere also ein Funktional E

$$E(v) = \sum_{x \in \Omega_x} g(x)[v \bullet \nabla u(x, t) + u_t(x, t)]^2,$$

wobei $g(x)$ eine Gewichtungsfunktion für die Umgebung $\Omega_x \subseteq \Omega$ ist. Mit dieser können Punkte, die näher an dem ursprünglichen Punkt sind, stärker gewichtet werden. Das Minimum der Funktion $E(v) = E(v_1, v_2)$ kann durch die Betrachtung der kritischen Punkte bestimmt werden.

$$\frac{\partial E(v_1, v_2)}{\partial v_1} = \sum_{x \in \Omega_x} g(x) [v_1 u_x^2 + v_2 u_x u_y + u_x u_t] = 0$$

$$\frac{\partial E(v_1, v_2)}{\partial v_2} = \sum_{x \in \Omega_x} g(x) [v_2 u_y^2 + v_1 u_x u_y + u_y u_t] = 0$$

In Matrixschreibweise lassen sich diese Gleichungen schreiben als:

$$Lv = b,$$

wobei die Einträge von L und b definiert sind durch:

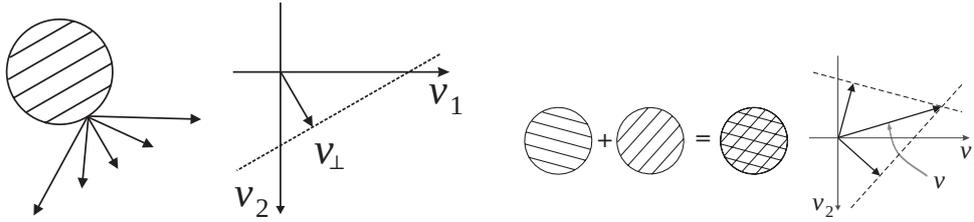
$$L = \begin{bmatrix} \sum g u_x^2 & \sum g u_x u_y \\ \sum g u_x u_y & \sum g u_y^2 \end{bmatrix}, \quad b = - \begin{bmatrix} \sum g u_x u_t \\ \sum g u_y u_t \end{bmatrix}.$$

Gilt $\text{rang}(L) = 2$, so ist das optimale kleinste Quadrate-Geschwindigkeitsfeld im Punkt x gegeben durch $v = L^{-1}b$. [Vgl.: 9, S.3f]

Die Beschränkung der Informationsgewinnung auf eine kleine Umgebung kann sich jedoch auch als problematisch erweisen. Ist L eine singuläre Matrix, so können wir nicht nach v lösen. Die Situation, in der dieses Problem auftritt, wird üblicherweise als *Aperture-Problem* bezeichnet.

2.3.1 Das Aperture-Problem

Die Untersuchung eines Punktes in einer kleinen Umgebung wird häufig mit dem Blick durch eine kleine Öffnung bzw. Blende (*Aperture*) verglichen. Stellen wir uns ein Streifenmuster um den Beobachtungspunkt vor (vgl. Abb. 4a). Für den Betrachter sind nur die Bewegungen orthogonal zu den Streifen erkennbar. Die Anteile einer Bewegung parallel zum Streifenmuster sind durch die kleine Öffnung nicht zu identifizieren. Erst durch eine Kante im Streifen, oder durch ein zweites Muster, welches über dem Ersten liegt, wird die Bewegung für einen Betrachter eindeutig. [1] Trotzdem enthält die Gleichung (2.1) Informationen, die man sich, auch bei lokaler Betrachtung, zu Nutze machen kann.



(a) Unabhängig von der tatsächlichen Bewegungsrichtung scheint das Gitter nur entlang v_{\perp} zu wandern. Bewegungsanteile orthogonal zu v_{\perp} , machen sich nur in der scheinbaren Geschwindigkeit bemerkbar.

(b) Bei zwei übereinanderliegenden Gittern ist nur eine Richtung v zulässig. Diese liegt am Schnittpunkt der möglichen Bewegungsvektoren der Einzelgitter.

Abbildung 4: Durch eine Blende scheinen sich Streifen immer nur orthogonal zu den Balken zu bewegen. Erst durch ein zweites Muster wird diese (fast immer) eindeutig [1]

Betrachten wir die orthogonale Projektion einer Lösung v auf den Gradienten ∇u für $\|\nabla u\| \neq 0$, so erhalten wir

$$v_{\perp} = \left(v \cdot \frac{\nabla u}{\|\nabla u\|} \right) \frac{\nabla u}{\|\nabla u\|} \stackrel{2.1}{=} \frac{-u_t}{\|\nabla u\|} \frac{\nabla u}{\|\nabla u\|}.$$

Wir können also anhand eines Punktes bestimmen, dass der Anteil der Lösung v entlang des Gradienten die Länge

$$\|v_{\perp}\| = \left\| \frac{-u_t}{\|\nabla u\|} \frac{\nabla u}{\|\nabla u\|} \right\| = \frac{|u_t|}{\|\nabla u\|}$$

besitzt. Alle möglichen Lösungen liegen somit auf einer Orthogonalen zum Gradienten $\|\nabla u\|$, mit der Entfernung $\frac{|u_t|}{\|\nabla u\|}$ zum Ausgangspunkt, wie in Abbildung 4a die gestrichelte Linie.

Da in einem Punkt der Gradient nur in eine Richtung zeigt, müssen die Informationen anderweitig erschlossen werden. Andererseits kann die gesuchte Richtung v nicht eindeutig bestimmt werden.

2.4 Globale Lösungsverfahren

Um die Schwierigkeiten der lokalen Betrachtung zu überwinden, versuchen *globale Verfahren* Annahmen an das gesamte Gebiet zu stellen. Beispielsweise kann gefordert werden, dass das gesuchte Geschwindigkeitsfeld entsprechend glatt gewählt wird. Mathematisch bedeutet dies eine Minimierung des Funktionals

$$E(v) = \int_{\Omega} (\nabla u(x, y) \cdot v(x, y) + u_t(x, y))^2 + \lambda (\|\nabla v_1(x, y)\|^2 + \|\nabla v_2(x, y)\|^2) dx dy.$$

Die Art der Glättung kann hierbei durch die Norm und den Glättungsfaktor λ , bzw. die Wahl der zulässigen Funktionen v gewählt werden.

Die Nachteile von globalen Verfahren ist jedoch die aufwändige Minimierung, die über das gesamte Bild durchgeführt werden soll. Somit müssen für jeden Punkt auf dem Gebiet die partiellen Gradienten in alle räumliche Dimensionen und die zeitliche Dimension berechnet werden. Der Herzschlag einer Maus kann durchaus die Frequenz von ca. 20 Hz haben, eine derart schnelle Berechnung der Geschwindigkeitsvektoren über diesen kurzen Intervallen ist unrealistisch.

Der Ansatz der vorliegenden Arbeit, der ein Mittelweg zwischen einem lokalen und globalen Verfahren aufzeigt, nutzt einerseits die Information, die ausschließlich lokal in einem Punkt zur Verfügung steht, nämlich die Orthogonalität zum Gradienten mit gegebenem Abstand, und andererseits eine Art der Glättung über ein größeres Gebiet in wenigen Punkten. So hoffen wir einerseits den Fallstrick des Aperture-Problems zu umgehen und andererseits die benötigte Anzahl der Gradienten und somit die Rechenzeit zu reduzieren.

Der Grundgedanke hierbei ist, dass wir die lokal vorhandenen Informationen mit Hilfe der *Thin-Plate Spline* Interpolation zu anderen Punkten auf dem Gebiet in einer gewissen glatten Art weitergeben.

Um ein gewisses Verständnis für die klassische *Thin-Plate Spline* Interpolation zu erhalten folgt im nächsten Abschnitt eine kleine Einführung in das Thema. Die strenge mathematische Formulierung im neuen Kontext findet sich ab Kapitel 4.

3 *Thin-Plate Spline* Interpolation

Thin-Plate Splines (TPS) stammen ursprünglich aus dem Flugzeugbau zum Berechnen von Verformungen von Tragflächen.¹ Hierbei wurde die Oberfläche der Tragfläche zwischen zweidimensionalen Punkten auf eine Weise interpoliert, die, wie der Name bereits verrät, der Verformung eines dünnen Bleches entspricht.

In einem eindimensionalen Setting wurden ursprünglich lineare Splines zur Interpolation genutzt, in höheren Dimensionen diese dementsprechend auf mehrdimensionale Gitter erweitert. Der Vorteil der *Thin-Plate Splines* ist, dass auf ein regelmäßiges Gitter, und somit auch auf die Bestimmung der entsprechenden eindimensionalen linearen Splines, verzichtet werden kann. Es stellt sich heraus, dass zum Berechnen der *Thin-Plate Spline* Interpolation nach dem Lösen eines linearen Gleichungssystems lediglich eine endliche Summe aus reellen Parametern und Werten des Logarithmus benötigt wird. [vgl. 11]

3.1 Klassische TPS Interpolation

Die mathematische Beschreibung des Problems basiert auf der Formulierung von Duchon [8]. Dieser hält an der Vorstellung fest, dass *TPS* der Verformung eines dünnen Blechs entsprechen, diese jedoch nicht wie in früheren Formulierungen auf einem beschränkten Gebiet betrachtet werden, sondern auf den gesamten \mathbb{R}^n erweitert. Diese Vorstellung eines unendlich ausgedehnten Bleches auf \mathbb{R}^n führt zu einer sehr einfachen Beschreibung der *TPS*.

Ausgangspunkt ist hierbei immer die Minimierung des Funktionals $\int_{\mathbb{R}^n} |D^2 v|^2$. Für $n = 2$ ist dies ähnlich zu der Biegeenergie eines Bleches. Das Hauptresultat, welches als Vorbild und Orientierung für den weiteren Verlauf der Arbeit dienen soll lautet folgendermaßen:

3.1 Satz (TPS- INTERPOLATION)

Sei A eine *gültige* Menge von Punkten von \mathbb{R}^2 . Dann existiert genau eine stetige Funktion w der

¹Im früheren Setting wurden die *Thin-Plate Splines* noch *Surface Splines* genannt. Vergleiche hierzu [11].

Form

$$w(x) = \sum_{a \in A} \gamma_a |a - x|^2 \log(|a - x|) + \alpha \bullet x + \beta, \quad \alpha \in \mathbb{R}^2, \beta \in \mathbb{R}$$

welche die gegebenen Interpolationsbedingungen auf A erfüllt und wobei

$$\sum_{a \in A} \gamma_a = 0 \quad \text{und} \quad \sum_{a \in A} \gamma_a a = 0$$

gilt.

Dann ist $\int_{\mathbb{R}^2} |D^2 w|^2 \leq \int_{\mathbb{R}^2} |D^2 f|^2$ für alle f , die mit w auf A übereinstimmen.

Die Funktion $r \mapsto r^2 \log(r)$ ist dabei in $r = 0$ stetig auf 0 zu erweitern.

Eine Funktion von der Form wie w nennen wir *Thin-Plate Spline*.

Die genaue Bedingung an die Menge A , sodass diese *gültig* ist, wollen wir auf einen späteren Zeitpunkt verschieben. Hier soll lediglich verdeutlicht werden, dass die resultierende Form der *TPS* eine sehr einfache Form besitzt.

3.2 TPS Interpolation entlang der Koordinatenachsen

Eine Abwandlung der *Thin-Plate Splines*, die sich von der anschaulichen Idee eines deformierten Blechs fortbewegt, beruht auf der Idee von Bookstein. In seinem Paper "Principal warps: thin-plate splines and the decomposition of deformations" schlägt er vor, die Verformung, die ursprünglich orthogonal zur Ebene stattfindet, umzuinterpretieren. So wird diese genutzt, um Verschiebungen *entlang* der Koordinatenachsen zu interpolieren. Als Anschauungsbeispiel dient ein Quadrat mit der Seitenlänge $\sqrt{2}$ in der Ebene. An den zwei gegenüberliegenden Punkten an $(0, \pm 1)$ ist eine (unendliche ausgedehnte) dünne Blechplatte in einer beliebigen aber festen Entfernung über dem Quadrat fixiert. An den übrigen zwei Punkten $(\pm 1, 0)$ in gleicher Distanz unter dem Quadrat. Die resultierende verformte Blechplatte entspricht der klassischen Anschauung der TPS-Interpolation (Siehe Abb. 5).

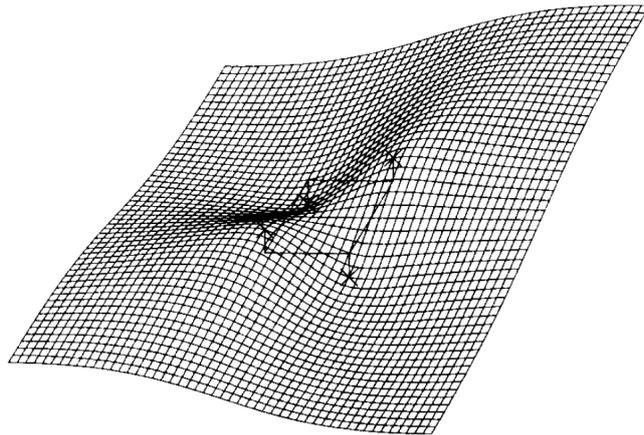


Abbildung 5: Klassisches Beispiel einer TPS-Interpolation. Ein dünnes Metallblech ist entlang eines Quadrates nach unten bzw. oben verformt. [Aus 5]

Für die neue Interpretation nach Bookstein betrachten wir erneut das gleiche Quadrat mit Seitenlängen $\sqrt{2}$ in der Ebene. Anstelle dieses in orthogonaler Richtung zu der Ebene zu verbiegen,

werden zwei Punkte in gegenüberliegenden Ecken entlang der y -Achse innerhalb der Koordinatenachsen verschoben, bis diese die Form wie in Abb. 6 (rechts) annehmen. Die direkte Intuition von der Funktion als Abbild eines verbogenen Blechs geht verloren, eine Verformung in die dritte Dimension findet nicht statt. Lediglich die Intensität der Manipulation in jedem Punkt wird dadurch bestimmt, als hätte diese orthogonal zu der Ebene stattgefunden. Die neue Funktion entspricht somit der Abbildung $(x, y) \mapsto (x, y + z(x, y))$, wobei $z(x, y)$ die Verformung der klassischen Anwendung aus Abb. 5 entspricht. Die oben getroffene Beschränkung der Verschiebung auf die y -Achse, ist nur in diesem Beispiel so gewählt, im Allgemeinen wirkt sich die Veränderung auf x - und y -Achse gleichermaßen aus.

Mit diesem Ansatz lösen die *TPS* ein zweidimensionales Interpolationsproblem mit beliebigen Punkten von \mathbb{R}^2 nach \mathbb{R}^2 . Dabei minimiert es eindeutig die Biegeenergie, die ein dünnes Metallblech innehat, als hätte die Verformung orthogonal zu der Koordinatenebene stattgefunden.

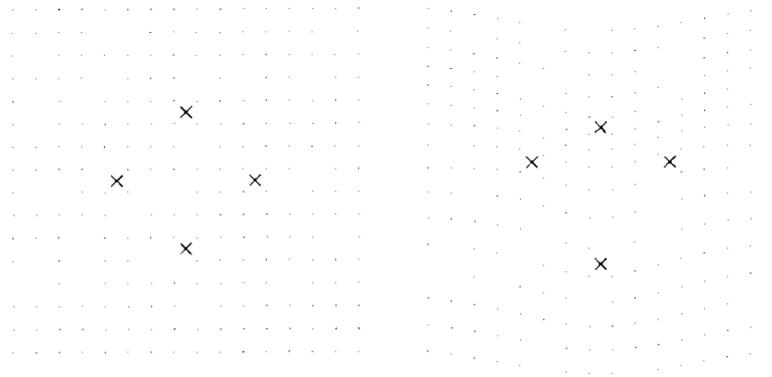


Abbildung 6: Interpretation der *TPS* als Verformung in Richtung der Koordinatenachsen. An den Punkten im Hintergrund lässt sich erkennen wie sich die Verformung auf die restlichen Punkte der Ebene auswirkt. [Aus 5]

Diese Intuition wollen wir im Hinterkopf behalten, wenn wir die *TPS*-Interpolation nutzen, um lokale Geschwindigkeitsinformationen zwischen den Punkten auszutauschen. Die Interpolationsbedingungen sind jedoch nicht mehr als Verschiebung innerhalb der Koordinatenachsen zu verstehen, schließlich sind lokal in jedem Punkt nur die Informationen der Gleichung (2.1), also ein skalarer Wert berechenbar.

4 Meinguets multivariate eindimensionale Interpolation

Nachdem wir nun ein Gefühl dafür entwickelt haben, welches Problem wir zu lösen versuchen, werden wir die mathematische Theorie rigoros erörtern. Als Grundlage dient hierbei die Veröffentlichung “Multivariate interpolation at arbitrary points made simple” von Jean Meinguet [14]. In Kapitel 5 folgen wir, sofern nicht anders notiert, der Beweisstruktur dieser Veröffentlichung. In diesem wird die Interpolation, entsprechend dem *Thin-Plate Spline* Vorbild für multivariate Funktionen $v : \mathbb{R}^n \rightarrow \mathbb{R}$ entwickelt. Diese Arbeit diente auch Bookstein als Grundlage der *TPS* Interpolation entlang der Koordinatenachse. In diesem soll die Funktion v für gegebene Interpolationsbedingungen $v(a_i) = \alpha_i, \forall i$ für eine endliche Menge an Punkten $A = (a_i)$ entsprechend minimiert werden. Für das *Optical Flow* Problem haben wir jedoch eine Funktion $v : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ mit den Interpolationsbedingungen $v(a_i) \bullet \varphi_i = \alpha_i$ entsprechend mit $\varphi_i = (\nabla u)_i$ und $\alpha_i = (-u_t)_i$

gegeben (vgl. Def. 2.1).

In Kapitel 5 soll also die Theorie für diesen angepassten Fall entwickelt werden, mit dem Hauptergebnis entsprechend dem Vorbild von Satz 3.1. Alle Ergebnisse werden für den allgemeinen Fall $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ bewiesen, somit können die Ergebnisse auch auf höherdimensionale Probleme übertragen werden.

4.1 Mathematische Grundlagen

Bis zu diesem Zeitpunkt haben wir auf die Definition von Rahmenbedingungen, die wir auf der Suche nach einer entsprechenden Funktion v stellen, verzichtet. Dies wollen wir nun nachholen. Es folgen einige grundlegende Definitionen [vgl. 10].

4.1 Definition (RAUM DER TESTFUNKTIONEN)

Sei $X \subset \mathbb{R}^n$ eine offene Menge.

Die Elemente des Vektorraumes $C_c^\infty(X)$ heißen *Testfunktionen* auf X , die Menge $\mathcal{D} := C_c^\infty(X)$ dementsprechend *der Raum der Testfunktionen* auf X .

Dabei ist $C^\infty(X)$ die Menge aller unendlich oft differenzierbarer Funktionen und $C_c^\infty(X) \subseteq C^\infty(X)$ die Menge aller unendlich oft differenzierbarer Funktionen mit kompakten Träger. Der *Träger* einer Funktion $f : X \rightarrow \mathbb{R}$ ($\text{supp } f$) definieren wir durch den Abschluss der Menge $\{x \in X \mid f(x) \neq 0\} \subseteq X$.

4.2 Definition (MULTI-INDEX)

Ein *(n-)Multi-Index* ist ein n -Tupel $\alpha = (\alpha_1, \dots, \alpha_n)$ mit nicht-negativen ganzen Zahlen α_i .

Die Länge von α ist definiert durch

$$|\alpha| := |\alpha_1| + \dots + |\alpha_n|.$$

Für eine $|\alpha|$ -fach differenzierbare Funktion $f \in C^{|\alpha|}(X)$ schreiben wir

$$\partial^\alpha f := \frac{\partial^{|\alpha|} f}{\partial^{\alpha_1} x_1, \dots, \partial^{\alpha_n} x_n}.$$

4.3 Definition (DISTRIBUTION)

Sei $X \subset \mathbb{R}^n$ eine offene Menge.

Eine lineare Abb. $u : C_c^\infty(X) \rightarrow \mathbb{R}$ nennen wir *Distribution*, wenn für jede kompakte Menge $K \subset X$ eine reelle Konstante $C \geq 0$ und eine nicht-negative ganze Zahl N existiert mit

$$|\langle u, \phi \rangle| \leq C \sum_{|\alpha| \leq N} \sup |\partial^\alpha \phi|,$$

für alle $\phi \in \mathcal{D}$ mit $\text{supp } \phi \subset K$.

Dabei bezeichnen wir mit $\langle u, \phi \rangle$ die duale Paarung der Distribution u mit der Testfunktion $\phi \in \mathcal{D}$.

Die Menge aller Distributionen auf X bezeichnen wir mit $\mathcal{D}'(X)$.

4.4 Erinnerung

Die Ableitung erster Ordnung einer Distribution $v \in \mathcal{D}'$ ist definiert durch

$$\langle \partial_i v, \phi \rangle := - \langle v, \partial_i \phi \rangle, \quad 1 \leq i \leq n, \quad \phi \in \mathcal{D}$$

entsprechend für die Ableitung der Ordnung $|\alpha|$ eines Multi-Index α

$$\langle \partial^\alpha v, \phi \rangle := (-1)^{|\alpha|} \langle v, \partial^\alpha \phi \rangle, \quad \phi \in \mathcal{D}.$$

4.5 Definition (L^p -RÄUME)

Sei (X, \mathcal{A}, μ) ein Maßraum und sei $1 \leq p < \infty$. Dann setzen wir

$$L^p(X, \mu) := \{f : X \rightarrow \mathbb{R} \mid f \text{ messbar, } \int_X |f|^p d\mu < \infty\}$$

und für $f \in L^p(X, \mu)$ setzen wir

$$\|f\|_p := \left(\int_X |f|^p d\mu \right)^{\frac{1}{p}}.$$

4.2 Multivariate Interpolation nach Meinguet

In diesem Abschnitt wollen wir kurz die Problemstellung aus Meinguets Originalveröffentlichung [14] darstellen, da diese Elemente im späteren Verlauf für uns relevant sein werden. Ausgangspunkt ist die Suche nach einem minimalen $v : \mathbb{R}^n \rightarrow \mathbb{R}$, welches gegebene Interpolationsbedingungen $v(a_i) = \alpha_i$ für alle $1 \leq i \leq N$ erfüllt.

Sei $m, n \geq 1 \in \mathbb{N}$. Als ein geeigneter Funktionenraum zur Lösung des Interpolationsproblems stellt sich der *Beppo-Levi Raum* heraus, definiert durch

$$X := BL^m(\mathbb{R}^n) = \{v \in \mathcal{D}'(\mathbb{R}^n) \mid \partial^\alpha v \in L^2(\mathbb{R}^n) \text{ für } |\alpha| = m\}, \quad (4.1)$$

wobei α ein Multi-Index ist.

In Worten betrachten wir also alle Distributionen, für die alle partiellen Ableitungen, im distributionellen Sinne, L^2 integrierbar sind.

Auf X kann ein semi-inneres Produkt $(\cdot, \cdot)_m$ definiert werden, also ein inneres Produkt, welches eine Seminorm

$$|v|_m := \left\{ \sum_{i_1, \dots, i_m=1}^n \int_{\mathbb{R}^n} |\partial_{i_1, \dots, i_m} v(x)|^2 dx \right\}^{1/2} \quad (4.2)$$

induziert.

4.6 Definition (SEMINORM)

Eine Abbildung $|\cdot|_m : X \rightarrow \mathbb{R}$ auf einem Vektorraum X über einen Körper \mathbb{K} heißt *Seminorm*, wenn folgende Eigenschaften für $v, w \in X$ und $\lambda \in \mathbb{K}$ erfüllt sind:

- $|v + w|_m \leq |v|_m + |w|_m$ (Dreiecksungleichung)
- $|\lambda v|_m = |\lambda| |v|_m$ (absolute Homogenität)

Die fehlende Eigenschaft, die $|\cdot|_m$ von einer Norm unterscheidet ist die positive Definitheit. Es ist von Bedeutung für welche Elemente von X die Seminorm verschwindet.

4.7 Lemma

Der Kern von $|\cdot|_m$ ist \mathcal{P}_{m-1} , der Raum aller Polynome $p : \mathbb{R}^n \rightarrow \mathbb{R}$ mit maximalen Grad $m - 1$, wobei

$$p(x) = \sum_{|e_j| \leq m-1} \gamma_{e_j} x^{e_j}, \quad x \in \mathbb{R}^n, \gamma_{e_i} \in \mathbb{R}$$

und $e_j = (e_{j1}, \dots, e_{jn}) \in \mathbb{N}^n$ ein Multi-Index mit $x^{e_j} := \prod_{i=1}^n x_i^{e_{ji}}$.

Beweis. Die Seminorm $|\cdot|_m$ verschwindet genau dann, wenn alle partiellen Ableitungen mit Ordnung m der Distribution auf \mathbb{R}^n verschwinden. Nach [19, S.60] sind dies die Polynome vom Grad $m - 1$, also \mathcal{P}_{m-1} . \square

4.8 Definition (SEMI-HILBERTRAUM)

Sei X ein reeller (komplexer) Vektorraum mit einem semi-inneren Produkt, welcher bezüglich der induzierten Seminorm vollständig ist. Dann bezeichnen wir X als *Semi-Hilbertraum*.

Als essentieller Satz, welcher die Wahl der genannten Funktionenräume und Normen rechtfertigt, und den Grundstein zur mathematisch korrekter Formulierung der *Thin-Plate Spline* Interpolation legt, nennen wir den folgenden Satz von Meinguet:

4.9 Satz

Sei $m > n/2$.

Dann ist der Raum $X := BL^m(\mathbb{R}^n)$ mit der Seminorm $|\cdot|_m$ definiert durch (4.1) und (4.2) ein semi-Hilbert Funktionenraum stetiger Funktionen auf \mathbb{R}^n , sodass alle lineare Auswertungsfunktionale mit endlichen Träger auf \mathbb{R}^n , welche auf \mathcal{P}_{m-1} verschwinden, entsprechend beschränkt sind. Insbesondere ist X vollständig bezüglich der Seminorm $|\cdot|_m$. Ein lineares Auswertungsfunktional mit endlichen Träger auf \mathbb{R}^n welches auf \mathcal{P}_{m-1} verschwindet, ist hier Maß der Form

$$\mu = \sum_{k \in K} \lambda_k \delta_{(b_k)},$$

wobei K endlich, die Punkte $(b_k)_{k \in K} \in \mathbb{R}^n$ paarweise verschieden und die Koeffizienten die Orthogonalitätsbedingung

$$\langle \mu, p \rangle := \sum_{k \in K} \lambda_k p(b_k) = 0, \quad \forall p \in \mathcal{P}_{m-1}$$

erfüllen.

Die Elemente in X sind in erster Linie keine Funktionen auf \mathbb{R}^n , sondern Äquivalenzklassen, welche sich auf der Nullmenge von $|\cdot|_m$ unterscheiden. Satz 4.9 garantiert uns die Existenz eines Repräsentanten jeder Äquivalenzklasse in X , der sogar eine stetige Funktion ist. Für diesen macht also eine Punktauswertung auf \mathbb{R}^n Sinn. Wenn wir im Folgenden $v(x)$, $x \in \mathbb{R}^n$ für $v \in X$ schreiben, dann ist der entsprechende stetige Repräsentant gemeint. Zudem sind die Funktionale, die genau auf der Menge der Polynome \mathcal{P}_{m-1} verschwinden, beschränkt. Wir werden später sehen, dass wir

diese Beschränktheit auf einer Teilmenge von X , die sogar ein echter Hilbertraum ist, wichtig ist. Die Bedingung $m > n/2$ setzen wir von nun an also voraus.

Ein ausführlicher Beweis, der Faltungen von Distributionen und Funktionen nutzt, ist in [13] zu finden.

4.10 Definition (MEINGUETS PROBLEM (P))

Sei $A = (a_i)_{i \in I}$ eine endliche Menge paarweise verschiedener Punkte in \mathbb{R}^n , welche eine \mathcal{P}_{m-1} -unisolvante Teilmenge $B = (a_j)_{j \in J}$ enthält. Sei zudem für eine Menge reeller Skalare $(\alpha_i)_{i \in I}$ die lineare Vielfalt definiert durch

$$V := \{v \in X \mid v(a_i) = \alpha_i, \forall i \in I\}, \quad (4.3)$$

vorausgesetzt $m > n/2$.

Dann beschreibt Meinguet das Problem (P) mit:

Suche $u \in V$, sodass

$$|u|_m := \inf_{v \in V} |v|_m \quad (4.4)$$

Dabei bezeichnen wir $B = (a_j)_{j \in J}$ als \mathcal{P}_{m-1} -unisolvante Teilmenge, wenn ein eindeutiges $p \in \mathcal{P}_{m-1}$ existiert, welches die Interpolationsbedingungen $p(a_j) = \alpha_j$ für die gegebenen Daten $\alpha_j, \forall j \in J$ erfüllt.

5 Multivariate n-dimensionale Interpolation

Meinguets mathematische Grundlage nutzen wir nun als Ausgangspunkt, um die Funktionenräume auf höhere Dimensionen zu erweitern und so an die veränderten Interpolationsbedingungen anzupassen. Wir sind weiterhin an gewissen Eigenschaften des Beppo-Levi Raumes X interessiert, es ist also natürlich diesen auf unsere Voraussetzung, also eine Erweiterung in n Dimensionen zu verallgemeinern. Wir definieren entsprechend den Produktraum

$$X^n := \underbrace{X \times \dots \times X}_{n\text{-mal}}$$

mit $X = BL^m(\mathbb{R}^n)$ wie in (4.1). Es ist also $v = (v_1, v_2, \dots, v_n) \in X^n$ genau dann, wenn $v_i \in X$ für alle $i = 1, \dots, n$.

Für $v \in X^n$ wollen wir analog zu Sektion 4.2 eine Seminorm auf X^n definieren.

$$|v|_{m,n} := \{ |v_1|_m^2 + \dots + |v_n|_m^2 \}^{1/2} \quad (5.1)$$

Diese Seminorm ist wohldefiniert, denn für $v \in X^n$ ist $v_i \in X$ für $i = 1 \dots n$, somit $|v_i|_m < \infty, 1 \leq i \leq n$ und somit $|v|_{m,n} < \infty$. Außerdem ist die absolute Homogenität und Subadditivität von $|\cdot|_{m,n}$ gewährleistet, da $|\cdot|_m$ absolut homogen und subadditiv ist.

Für $v \in X^n$ zeigt sich außerdem

$$\begin{aligned}
|v|_{m,n}^2 &= |v_1|_m^2 + \dots + |v_n|_m^2 \\
&= \sum_{i_1, \dots, i_m=1}^n \int_{\mathbb{R}^n} |\partial_{i_1, \dots, i_m} v_1(x)|^2 dx + \dots + \sum_{i_1, \dots, i_m=1}^n \int_{\mathbb{R}^n} |\partial_{i_1, \dots, i_m} v_n(x)|^2 dx \\
&= \sum_{i_1, \dots, i_m=1}^n \int_{\mathbb{R}^n} |\partial_{i_1, \dots, i_m} v_1(x)|^2 + \dots + |\partial_{i_1, \dots, i_m} v_n(x)|^2 dx \\
&= \sum_{i_1, \dots, i_m=1}^n \int_{\mathbb{R}^n} |\partial_{i_1, \dots, i_m} v(x)|_2^2 dx.
\end{aligned}$$

Die Seminorm $|\cdot|_{m,n}$ auf X^n unterscheidet sich zu der Seminorm auf X somit nur aufgrund der höheren Dimension und somit des Betrags des Vektors in \mathbb{R}^n . Der Einfachheit halber werden wir somit die Seminorm $|\cdot|_{m,n}$ und das zugehörige semi-innere Produkt $(\cdot, \cdot)_{m,n}$ mit $|\cdot|_m$, bzw. $(\cdot, \cdot)_m$ bezeichnen.

Die Interpretation einer Distribution $v \in X^n$ als stetige Funktion $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ist weiterhin durch Satz 4.9 gerechtfertigt, da die Stetigkeit von v von der komponentenweisen Stetigkeit abhängt. Wir halten somit weiterhin an der Bedingung $m > n/2$ fest. Der Kern von $|\cdot|_m$ ist offensichtlich, mit Blick auf Lemma 4.7, durch \mathcal{P}_{m-1}^n , die n -dimensionalen Polynome vom Grad $(m-1)$ auf \mathbb{R}^n gegeben.

Um das Problem (P) vollständig in dem neuen Setting formulieren zu können, müssen wir noch die Voraussetzungen an die Punkte $A = (a_i)_{i \in I}$ anpassen. In der Definition von Problem (P) in Def. 5.7 wird die \mathcal{P}_{m-1} -Unisolvenz einer Teilmenge B gefordert. Ähnliche Ansprüche wollen wir auch an die gegebenen Punkte und Daten in diesem Fall stellen. Wir definieren dazu den Begriff der \mathcal{P}_{m-1}^n -Unisolvenz. Es sei angemerkt, dass dieser etwas von der üblichen Formulierung abweicht, in unserem Setting jedoch stimmig scheint.

5.1 Unisolvenz der Punktmenge

Sei $A = (a_i)_{i \in I}$ eine endliche Menge paarweise verschiedener Punkte im \mathbb{R}^n mit entsprechenden Daten $(\alpha_i)_{i \in I} \in \mathbb{R}$, $(\varphi_i)_{i \in I} \in \mathbb{R}^n$ in diesen Punkten gegeben.

Um die eindeutige Existenz einer Lösung für eine gegebene Punktmenge zu garantieren, interessiert uns die Dimension des Polynomraums \mathcal{P}_{m-1}^n und somit von \mathcal{P}_{m-1} .

5.1 Lemma

Sei \mathcal{P}_{m-1} der Raum aller Polynome vom Grad $m-1$ auf \mathbb{R}^n . Dann hat \mathcal{P}_{m-1} Dimension

$$M = \binom{n+m-1}{n}. \quad (5.2)$$

Beweis. Die multivariaten Monome mit Grad $\leq m-1$ und der Form

$$x^e = \prod_{i=1}^n x_i^{e_i},$$

mit den Multiindize $e \in \mathbb{N}^n$, $|e| \leq m-1$ bilden eine Basis des Polynomraumes \mathcal{P}_{m-1} . Wir wollen die Anzahl der Basiselemente bestimmen.

Die Anzahl der Monome eines bestimmten Grades k sind genau die Anzahl der k -elementigen Multimengen über eine n -elementige Menge. In einer Multimenge kann ein Element gegenüber einer Menge mehrmals vorkommen. Anschaulich steht die Vielfachheit einer Variable in der Multimenge für den entsprechenden Exponenten des zugehörigen Monoms. Die Multimenge $\{x_1, x_1, x_1, x_2, x_3, x_3\}$ entspricht z.B. dem Monom $x_1^3 \cdot x_2 \cdot x_3^2$. Somit kann jede Variable x_i , $i = 1, \dots, n$ maximal k -mal in die Multimenge aufgenommen werden.

Die mögliche Anzahl k -elementiger Multimengen einer n -elementigen Menge wird mit

$$\binom{\binom{n}{k}}{k} := \binom{k+n-1}{n-1}$$

berechnet. Die Anzahl aller Monome mit maximalem Grad $m-1$, und somit die aller Basiselemente, ist somit die Summe

$$\sum_{k=0}^{m-1} \binom{k+n-1}{n-1} = \binom{n+m-1}{n}$$

was mit Induktion leicht folgt. □

Aus Lemma 5.1 folgt direkt, dass die Dimension von \mathcal{P}_{m-1}^n genau $n \cdot M$ ist. Wir wollen nun definieren wann wir eine Menge \mathcal{P}_{m-1}^n -unisolvent bezeichnen.

5.2 Definition

Sei $B = (a_j)_{j \in J}$ eine Untermenge von A mit $N := n \cdot M$ Punkten, M definiert durch (5.2).

Dann heißt B \mathcal{P}_{m-1}^n -unisolvent, wenn ein eindeutiges $p \in \mathcal{P}_{m-1}^n$ existiert mit

$$p(a_j) \bullet \varphi_j = \alpha_j, \quad \forall j \in J$$

für gegebene Daten $(\alpha_j)_{j \in J}, (\varphi_j)_{j \in J}$.

Die Existenz eines solchen $p \in \mathcal{P}_{m-1}^n$ hängt von der Anordnung der Koordinaten $(a_j)_{j \in J}$ und der Vektoren $(\varphi_j)_{j \in J}$ ab. Grundsätzlich lässt sich folgendes über die Existenz einer solchen Lösung sagen:

5.3 Lemma

Sei $(a_i)_{i \in I}$ die Menge der gewählten Koordinaten in \mathbb{R}^n und $\varphi_i \in \mathbb{R}^n$, $\alpha_i \in \mathbb{R}$, $\forall i \in I$ die gegebenen Daten in den Punkten. Dann existiert ein $p \in \mathcal{P}_{m-1}^n$ mit $p(a_i) \bullet \varphi_i = \alpha_i$, $\forall i \in I$ genau dann, wenn die Vandermonde-Matrix

$$V = \begin{bmatrix} \varphi_{11}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] & \cdots & \varphi_{1n}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] \\ \varphi_{21}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] & \cdots & \varphi_{2n}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] \\ \vdots & & \vdots \\ \varphi_{S1}[a_S^{e_1}, a_S^{e_2}, \dots, a_S^{e_M}] & \cdots & \varphi_{Sn}[a_S^{e_1}, a_S^{e_2}, \dots, a_S^{e_M}] \end{bmatrix} \quad (5.3)$$

invertierbar ist. Dabei ist $S \in \mathbb{N}$ und $M = \binom{m-1+n}{n}$ die maximale Anzahl aller Monome eines Polynoms $p \in \mathcal{P}_{m-1}$.

Beweis. Ein Polynom $p \in \mathcal{P}_{m-1}^n$ hat die Form

$$p(x) = \sum_{|e_j| \leq m-1} \gamma_{e_j} x^{e_j}, \quad x \in \mathbb{R}^n, \gamma_{e_i} = \begin{pmatrix} \gamma_{e_{i1}} \\ \vdots \\ \gamma_{e_{in}} \end{pmatrix} \in \mathbb{R}^n$$

dabei ist $e_j = (e_{j1}, \dots, e_{jn}) \in \mathbb{N}^n$ ein Multi-Index mit $|e_j| = \sum_i^n e_{ji}$ und $x^{e_j} := \prod_{i=1}^n x_i^{e_{ji}}$ das entsprechende Monom vom Grad $|e_j|$.

Für die oben gegebene Matrix V betrachten wir dann das Gleichungssystem

$$V \cdot \begin{bmatrix} \gamma_{e_{11}}, \dots, \gamma_{e_{M1}}, \gamma_{e_{12}}, \dots, \gamma_{e_{M2}}, \dots, \gamma_{e_{1n}}, \dots, \gamma_{e_{Mn}} \end{bmatrix}^t = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_S \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \varphi_{11}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] & \dots & \varphi_{1n}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] \\ \varphi_{21}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] & \dots & \varphi_{2n}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] \\ \vdots & & \vdots \\ \varphi_{S1}[a_S^{e_1}, a_S^{e_2}, \dots, a_S^{e_M}] & \dots & \varphi_{Sn}[a_S^{e_1}, a_S^{e_2}, \dots, a_S^{e_M}] \end{bmatrix} \cdot \begin{bmatrix} \gamma_{e_{11}} \\ \vdots \\ \gamma_{e_{M1}} \\ \gamma_{e_{12}} \\ \vdots \\ \gamma_{e_{M2}} \\ \vdots \\ \gamma_{e_{1n}} \\ \vdots \\ \gamma_{e_{Mn}} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_S \end{bmatrix}.$$

Dieses Gleichungssystem entspricht der Gleichung $p(a_i) \bullet \varphi_i = \alpha_i, \forall i \in I$ und hat genau dann eine eindeutige Lösung wenn V invertierbar ist. \square

5.4 Korollar

Grundvoraussetzung für die Invertierbarkeit von V ist, dass V quadratisch ist. Somit muss $S \in \mathbb{N}$ der Anzahl der Spalten von V entsprechen. Wir benötigen somit genau $N := n \cdot M$ Punkte, um ein eindeutiges Polynom $p \in \mathcal{P}_{m-1}^n$ zu finden welches die Interpolationsbedingungen erfüllt.

5.5 Bemerkung

Der für uns besonders interessante Fall $n = m = 2$ vereinfacht das gesuchte Polynom auf die Form $p(x) = \begin{pmatrix} \gamma_{e_{11}} \\ \gamma_{e_{12}} \end{pmatrix} + \begin{pmatrix} \gamma_{e_{21}} & \gamma_{e_{31}} \\ \gamma_{e_{22}} & \gamma_{e_{32}} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$. Die benötigte Anzahl von Punkten ist $N = n \cdot M = 2 \cdot \binom{3}{2} = 2 \cdot 3 = 6$ und die Vandermonde-Matrix entspricht

$$V = \begin{bmatrix} \varphi_{11}[1, a_{11}, a_{12}] & \varphi_{12}[1, a_{11}, a_{12}] \\ \varphi_{21}[1, a_{21}, a_{22}] & \varphi_{22}[1, a_{21}, a_{22}] \\ \vdots & \vdots \\ \varphi_{61}[1, a_{61}, a_{62}] & \varphi_{62}[1, a_{61}, a_{62}] \end{bmatrix}.$$

Die minimale Anzahl von Punkten, um die Lösbarkeit zu garantieren ist somit durch $N = 6$ nach unten beschränkt.

Die genauen geometrischen Anforderungen an die Position der Koordinaten sind nicht so einfach zu bestimmen, aber auch nicht weiter entscheidend. Wir wollen einfach nur fordern, dass ein entsprechendes $p \in \mathcal{P}_{m-1}^n$ existiert.

Mit der Existenz geht folgende Erkenntnis mit ein:

5.6 Satz

Ein eindeutiges $p \in \mathcal{P}_{m-1}^n$ welches die Interpolationsbedingungen $p(a_j) \bullet \varphi_j = \alpha_j$, $\forall j \in J$ erfüllt, existiert genau dann, wenn eine Lagrange-Basis $(p_i)_{i \in J} \in \mathcal{P}_{m-1}^n$ existiert mit $p_i(a_j) \bullet \varphi_j = \delta_{ij}$, $\forall i, j \in J$.

Beweis. Der Beweis folgt einer Beweisskizze von Prof. Wirth.

” \Rightarrow ”

Sei $p \in \mathcal{P}_{m-1}^n$ eindeutig, welches die Interpolationsbedingungen erfüllt. Da die Existenz eines eindeutigen p nach Lemma 5.3 nur von der Invertierbarkeit der Vandermonde-Matrix (5.3) abhängt, existieren auch eindeutige Lösungen p_i für die Interpolationsprobleme mit $\alpha_j = \delta_{ij}$, $j \in J$ für alle $i \in J$.

Die $(p_i)_{i \in J}$ sind linear unabhängig und somit eine Basis der multivariaten Polynome, denn:

$$\begin{aligned} \sum_{i \in J} \beta_i p_i(a_j) &= 0, \quad 1 \leq j \leq N \\ \Rightarrow \sum_{i \in J} \beta_i p_i(a_j) \bullet \varphi_j &= 0, \quad 1 \leq j \leq N \\ \Leftrightarrow \sum_{i \in J} \beta_i \delta_{ij} &= 0, \quad 1 \leq j \leq N \\ \Leftrightarrow \beta_j &= 0, \quad 1 \leq j \leq N. \end{aligned}$$

Somit existiert eine Lagrange-Basis $(p_i)_{i \in J} \in \mathcal{P}_{m-1}^n$ mit $p_i(a_j) \bullet \varphi_j = \delta_{ij}$.

” \Leftarrow ”

Sei umgekehrt $(p_i)_{i \in J} \in \mathcal{P}_{m-1}^n$ eine Lagrange-Basis, welche die Interpolationsbedingung erfüllt. Setze dann $p = \sum_{i \in J} \alpha_i p_i$. Dann gilt für alle $j \in J$

$$p(a_j) \bullet \varphi_j = \sum_{i \in J} \alpha_i p_i(a_j) \bullet \varphi_j = \sum_{i \in J} \alpha_i \delta_{ij} = \alpha_j.$$

Zudem ist p eindeutig, denn sei $q = \sum_{i \in J} \beta_i p_i$ eine andere Lösung, welche die Interpolationsbedingung erfüllt, so ist

$$\alpha_j = q(a_j) \bullet \varphi_j = \sum_{i \in J} \beta_i p_i(a_j) \bullet \varphi_j = \sum_{i \in J} \beta_i \delta_{ij} = \beta_j, \quad j \in J,$$

somit $\beta_j = \alpha_j$ für alle $j \in J$ und daher $p = q$. Also ist p ein eindeutiges Polynom p , welches die Interpolationsbedingungen erfüllt. \square

5.2 Formulierung von Problem (P)

Wir sind nun zu vollständigen mathematischen Formulierung des Problems bereit.

5.7 Definition (PROBLEM (P))

Sei $n, m \in \mathbb{N}$ mit $n, m \geq 1$ und $m > n/2$.

Zudem sei $A = (a_i)_{i \in I}$ eine endliche Menge paarweise verschiedener Punkte in \mathbb{R}^n , sodass eine \mathcal{P}_{m-1}^n -unisolvante Untermenge $B = (a_j)_{j \in J}$ mit $N = n \cdot M$ Punkten existiert. M ist bestimmt durch Lemma 5.1. Beschreibe für gegebene Daten $(\alpha_i)_{i \in I}$ in \mathbb{R} und $(\varphi_i)_{i \in I}$ in \mathbb{R}^n die Menge der linearen Variationen V^n durch

$$V^n = \{v \in X^n \mid v(a_i) \bullet \varphi_i = \alpha_i \quad \forall i \in I\}. \quad (5.4)$$

Dann bezeichnet das *Problem (P)* die Suche nach $u \in V^n$, sodass

$$|u|_m = \inf_{v \in V^n} |v|_m,$$

gilt.

5.3 Darstellungsformeln in Beppo Levi Räumen

Im folgenden Abschnitt erarbeiten wir uns eine Darstellung von v , welche uns näher an die einfache Formulierung bringt, die wir uns von der *Thin-Plate Spline* Interpolation erhoffen. Dabei werden wir auf Eigenschaften des Beppo-Levi Raumes zurück greifen. Die Voraussetzungen des Problems (P) aus Def. 5.7 setzen wir nun als erfüllt an.

Durch die \mathcal{P}_{m-1}^n -Unisolvenz und damit durch die Existenz eines Polynoms $p \in \mathcal{P}_{m-1}^n$, welches die Interpolationsbedingung erfüllt, erhalten wir mit Satz 5.6 den Interpolanten Pv von v definiert durch

$$Pv = \sum_{j \in J} v(a_j) \bullet \varphi_j p_j, \quad (5.5)$$

mit $(p_j)_{j \in J}$ der entsprechenden Lagrange-Basis von \mathcal{P}_{m-1}^n .

Die Abbildung $P : X^n \rightarrow X^n$ ist offensichtlich linear und eine Projektion, denn

$$\begin{aligned} P(Pv) &= \sum_{j \in J} \left(\sum_{i \in J} v(a_i) \bullet \varphi_i p_i(a_j) \right) \bullet \varphi_j p_j \\ &= \sum_{j \in J} \sum_{i \in J} v(a_i) \bullet \varphi_i \underbrace{(p_i(a_j) \bullet \varphi_j)}_{=\delta_{ij}} p_j \\ &= \sum_{i \in J} v(a_i) \bullet \varphi_i p_i = Pv. \end{aligned}$$

Somit ist P eine lineare Projektion mit Bild \mathcal{P}_{m-1}^n und Kern

$$X_0^n := \{v \in X^n \mid v(a_j) \bullet \varphi_j = 0 \quad \forall j \in J\}.$$

Mit Hilfe der Projektion P würden wir gerne den Semi-Hilbertraum X^n als direkte Summe

$$X^n = X_0^n \oplus \mathcal{P}_{m-1}^n$$

schreiben. Im nächsten Satz werden wir sehen, dass X_0^n ein Hilbertraum und somit abgeschlossen ist, die Zerlegung von X^n als innere direkte Summe somit gerechtfertigt ist [Vgl. 6, S.223, Theorem 9.3.10.].

5.8 Bemerkung

Diese Zerlegung des Raumes ist bei Meinguet in der Form $X = X_0 \oplus \mathcal{P}_{m-1}$ zu finden. [14] Hierbei entspricht

$$X_0 := \{v \in X \mid v(a_j) = 0 \quad \forall j \in J\}. \quad (5.6)$$

Wie wir später sehen werden, kann das ursprüngliche X_0 (bzw. X_0^n) alternativ als Kern stetiger linearer Funktionale aufgefasst werden, welches die Abgeschlossenheit ebenfalls zeigt.

Auf dem Unterraum X_0^n ist die Seminorm $|\cdot|_m$ sogar eine Norm, dies impliziert folgenden Satz:

5.9 Satz

X_0^n ist zusammen mit der Seminorm $|\cdot|_m$ aus (5.1) nun sogar ein (echter) Hilbertraum.

Beweis. Wir betrachten zunächst den Quotientenraum X^n/\mathcal{P}_{m-1}^n und zeigen dort die Vollständigkeit. Die entsprechende Quotientenabbildung auf X^n hat die Form

$$\begin{aligned} \pi : X^n &\rightarrow X^n/\mathcal{P}_{m-1}^n \\ v &\mapsto v + \mathcal{P}_{m-1}^n, \end{aligned}$$

d.h. $\pi(v) = \pi(w)$ genau dann, wenn $\pi(v) - \pi(w) \in \mathcal{P}_{m-1}^n$.

Dann ist X^n/\mathcal{P}_{m-1}^n aber ein Hilbertraum mit der Norm $\|\pi(v)\|$ definiert durch $\|\pi(v)\| := |v|_m$ [vgl. 17, S.32]. Denn nun ist für gleiche Elemente $\pi(v), \pi(w) \in X^n/\mathcal{P}_{m-1}^n$

$$\begin{aligned} \pi(v) = \pi(w) &\Leftrightarrow \pi(v - w) = 0 \\ &\Leftrightarrow \|\pi(v - w)\| = \|0\| \\ &\Leftrightarrow |v - w|_m = 0 \end{aligned}$$

und außerdem gilt die umgekehrte Dreiecksungleichung für die Seminorm $|\cdot|_m$

$$\left| |v|_m - |w|_m \right| \leq |v - w|_m.$$

Also ist $0 = |v - w|_m \geq ||v|_m - |w|_m| = ||\pi(v)\| - \|\pi(w)\|$ und somit $\|\pi(v)\| = \|\pi(w)\|$, die Norm somit wohldefiniert. Die Normeigenschaften von $\|\cdot\|$ folgen direkt aus den Eigenschaften der Seminorm $|\cdot|_m$.

Außerdem ist X^n/\mathcal{P}_{m-1}^n vollständig bzgl. $\|\cdot\|$, denn ist $\pi(v_n)$ eine Cauchyfolge in X^n/\mathcal{P}_{m-1}^n , ein beliebiges $\varepsilon > 0$, dann existieren $N, M \in \mathbb{N}$ mit

$$\varepsilon > \|\pi(v_M) - \pi(v_N)\| = \|\pi(v_M - v_N)\| = |v_M - v_N|_m \quad (5.7)$$

und somit ist trivialerweise auch v_n eine Cauchyfolge in X^n bezüglich $|\cdot|_m$. Aufgrund der Vollständigkeit von X^n nach Satz 4.9, sofern $m > n/2$, konvergiert diese Cauchyfolge gegen ein $w \in X^n$ und analog zu (5.7) folgt die Konvergenz von $\pi(v_n)$ gegen $\pi(w)$. Somit ist der Quotientenraum X^n/\mathcal{P}_{m-1}^n vollständig bzgl. $\|\cdot\|$.

Für die Identitätsabbildung I und die Projektion P aus (5.5) ist $(I - P) : X^n \rightarrow X_0^n$ ebenfalls eine lineare Projektion mit $\ker((I - P)) = \mathcal{P}_{m-1}^n$, es folgt aufgrund des Isomorphiesatzes [16, S.85, Theo: 2.70], dass $X^n/\ker((I - P))$ isomorph zum Bild von $I - P$ ist. Dies entspricht

$$X^n/\mathcal{P}_{m-1}^n \cong X_0^n.$$

Per Definition von $\|\cdot\|$ folgt zusätzlich die Isometrie von X^n/\mathcal{P}_{m-1}^n und X_0^n . Da X^n/\mathcal{P}_{m-1}^n vollständig ist, folgt somit die Vollständigkeit von X_0^n und somit ist X_0^n ein Hilbertraum mit Norm $|\cdot|_m$. \square

Nach Satz 4.9 ist X_0 für $m > n/2$ ebenfalls ein Hilbert-Funktionenraum von stetigen Funktionen auf \mathbb{R}^n , sowie sind alle (linearen) Auswertungsfunktionale $\delta_{(x)}$ für feste x auf X_0 beschränkt und somit stetig.

Betrachten wir nun für ein festes aber beliebiges $\varphi \in \mathbb{R}^n$ das Funktional $v \mapsto v(x) \bullet \varphi \in \mathbb{R}$, so ist dieses, als Linearkombination stetiger Funktionale, ebenfalls linear und stetig. Wir können die Auswertung eines solchen Funktionalen schreiben als

$$v(x) \bullet \varphi = \langle \varphi \delta_{(x)}, v \rangle = (K_x^\varphi, v)_m, \quad \forall v \in X_0^n, \forall \varphi, x \in \mathbb{R}^n. \quad (5.8)$$

Dabei bezeichnen wir mit $\langle \cdot, \cdot \rangle$ die duale Paarung zwischen X_0^n und $X_0^{n'}$ und mit $K_x^\varphi \in X_0^n$ der entsprechende eindeutige Fréchet-Riesz Repräsentant von $\varphi \delta_{(x)}$. Dieser existiert nach dem Satz von Fréchet-Riesz, da $v \mapsto v(x) \bullet \varphi \in \mathbb{R}$ ein stetiges Funktional ist und X_0^n ein Hilbertraum mit dem Skalarprodukt $(\cdot, \cdot)_m$.

5.10 Satz (FRÉCHET-RIESZ [6, S.218, THEO: 9.3.3])

Sei L ein beschränktes lineares Funktional über einen Hilbertraum X mit Skalarprodukt (\cdot, \cdot) . Dann existiert ein eindeutiges Element $x_0 \in X$, sodass gilt

$$L(x) = (x, x_0), \quad \forall x_0 \in X.$$

Für die Projektion P ist $I - P$ ebenfalls eine Projektion mit dem Bild X_0^n , dabei ist I die Identitätsabbildung. Somit erfüllt $\phi - P\phi \in X_0^n$ für Testfunktionen $\phi \in \mathcal{D}^n$ die Gleichung 5.8.

Dann folgt für $\phi - P\phi \in X_0^n$ mit partieller Integration und der Definition von partiellen Ableitungen auf Distributionen

$$\begin{aligned}
(K_x^\varphi, \phi - P\phi)_m &= (K_x^\varphi, \phi)_m - \underbrace{(K_x^\varphi, P\phi)}_{\in X_0^n, \in P_{m-1}^n}_m \\
&= (K_x^\varphi, \phi)_m - 0 \\
&= \sum_{i_1, \dots, i_m=1}^n \int_{\mathbb{R}^n} \partial_{i_1, \dots, i_m} K_x^\varphi(y) \bullet \partial_{i_1, \dots, i_m} \phi(y) dy \\
&= \int_{\mathbb{R}^n} (-1)^m \sum_{i_1, \dots, i_m=1}^n K_x^\varphi(y) \bullet (\partial_{i_1, \dots, i_m})^2 \phi(y) dy \\
&= \langle (-1)^m \Delta^m K_x^\varphi, \phi \rangle \quad \forall \phi \in \mathcal{D}^n
\end{aligned} \tag{5.9}$$

mit $\Delta^m := \sum_{i_1, \dots, i_m=1}^n (\partial_{i_1, \dots, i_m})^2$.

Zudem gilt die Identität

$$(\phi - P\phi)(x) \bullet \varphi = (\phi(x) - \sum_{j \in J} \phi(a_j) \bullet \varphi_j p_j(x)) \bullet \varphi = \langle \varphi \delta_{(x)} - \sum_{j \in J} p_j(x) \bullet \varphi \varphi_j \delta_{(a_j)}, \phi \rangle, \tag{5.10}$$

da $\phi(x) \bullet \varphi = \langle \varphi \delta_{(x)}, \phi \rangle$ und $\phi(a_j) \bullet \varphi = \langle \varphi \delta_{(a_j)}, \phi \rangle$.

Mit Hilfe der Gleichungen 5.8 und 5.10 folgt, dass K_x^φ eine Lösung der partiellen Differentialgleichung

$$(-1)^m \Delta^m K_x^\varphi = \varphi \delta_{(x)} - \sum_{j \in J} p_j(x) \bullet \varphi \varphi_j \delta_{(a_j)}, \tag{5.11}$$

im distributionellen Sinn ist, denn

$$\begin{aligned}
\langle \varphi \delta_{(x)} - \sum_{j \in J} p_j(x) \bullet \varphi \varphi_j \delta_{(a_j)}, \phi \rangle &\stackrel{5.10}{=} (\phi - P\phi)(x) \bullet \varphi \\
&\stackrel{5.8}{=} (K_x^\varphi, \phi - P\phi)_m \\
&\stackrel{5.9}{=} \langle (-1)^m \Delta^m K_x^\varphi, \phi \rangle \quad \forall \phi \in \mathcal{D}^n,
\end{aligned}$$

in $X_0^n \subset \mathcal{D}^{n'}$.

5.11 Lemma

Eine Lösung der Differentialgleichung (5.11) in \mathcal{D}' lässt sich über die Fundamentallösung der Differentialgleichung Δ^m in \mathbb{R}^n herleiten. Eine solche Lösung entspricht der distributionellen Differentialgleichung $\Delta^m E = \delta$, $E \in \mathcal{D}'$ und ist nach [13, S.169] die rotationsinvariante Funktion auf dem Komplement des Ursprungs von \mathbb{R}^n gegeben in [19, S.288, Example 2] durch

$$E(y) := \begin{cases} cr^{2m-n} \log r & \text{falls } 2m \geq n \text{ und } n \text{ gerade} \\ dr^{2m-n} & \text{sonst.} \end{cases} \tag{5.12}$$

Dabei ist $r := |y|$ die euklidische Norm des Vektors $y \in \mathbb{R}^n$ und

$$c := \frac{(-1)^{n/2+1}}{2^{2m-1}\pi^{n/2}(m-1)!(m-n/2)!}, \quad (5.13)$$

$$d := \frac{(-1)^m \Gamma(n/2 - m)}{2^{2m}\pi^{n/2}(m-1)!}. \quad (5.14)$$

Für $y = 0$ ist betrachten wir erneut die stetige Erweiterung auf null.

5.12 Beispiel

Für $m = n = 2$ vereinfacht sich der Parameter c zu

$$c = \frac{1}{8\pi},$$

die Lösung der Differentialgleichung (5.12) entspricht dann der Formel

$$E(y) = cr^{2m-n} \log r = \frac{1}{8\pi} |y|^2 \log(|y|),$$

für alle $y \in \mathbb{R}^2 \setminus \{0\}$.

Nach dem Superpositionsprinzip von linearen Operatoren sind alle Linearkombinationen von Lösungen einer Gleichung wiederum Lösungen der Gleichungen. Wir erhalten somit mit (5.12) eine Lösung der Differentialgleichung (5.11), gegeben durch

$$H_x^\varphi(y) = (-1)^m \left[E(x-y)\varphi - \sum_{j \in J} p_j(x) \bullet \varphi E(a_j - y)\varphi_j \right], \quad \forall x, y \in \mathbb{R}^n. \quad (5.15)$$

Dabei ergibt sich die Lösung für das verschobene Dirac-Maß $\varphi\delta_{(a)}$ durch die komponentenweise Translation der Lösung $E(\cdot - a)$ und dem inneren Produkt mit φ .

5.13 Lemma

Für $m > n/2$ ist $H_x^\varphi \in X^n$ und diese Lösung ist eindeutig bis auf \mathcal{P}_{m-1}^n .

Beweis. Für das vereinfachte Problem nach [14] ist

$$H_x(y) = (-1)^m \left[E(x-y) - \sum_{j \in J} p_j(x) E(a_j - y) \right] \in \mathbb{R}, \quad \forall x, y \in \mathbb{R}^n.$$

analog zu (5.15) formuliert. Nach [13, S. 174, Theorem 2] ist $H_x \in X$, sofern $m > n/2$.

Dann ist aber auch $H_x^\varphi \in X^n$ für beliebiges $\varphi \in \mathbb{R}^n$.

Angenommen H_x^φ ist nicht eindeutig, d.h. es existiert ein weiteres \tilde{H}_x^φ welches die Differentialgleichung (5.12) ebenfalls erfüllt. Dann gilt aber

$$(-1)^m \Delta^m (H_x^\varphi - \tilde{H}_x^\varphi) = (-1)^m \Delta^m (H_x^\varphi) - (-1)^m \Delta^m (\tilde{H}_x^\varphi) = 0,$$

also ist $H_x^\varphi - \tilde{H}_x^\varphi$ eine Lösung der iterativen Laplace Gleichung. Andererseits sind die einzigen Lösungen der iterativen Laplace Gleichung in X^n die Polynome vom Grad $\leq m-1$ [vgl. 14, S.

296], d.h. $H_x^\varphi - \tilde{H}_x^\varphi \in \mathcal{P}_{m-1}^n$. Somit ist $H_x^\varphi \sim \tilde{H}_x^\varphi$ in X/\mathcal{P}_{m-1}^n und H_x^φ eine eindeutige Lösung. Mit der Isomorphie $X_0^n \cong X/\mathcal{P}_{m-1}^n$ (vgl. Beweis von Satz 5.9) folgt, dass ein eindeutiges $K_x^\varphi \in X_0^n \cong X/\mathcal{P}_{m-1}^n$ existiert, welches die Differentialgleichung (5.11) erfüllt. \square

Dieses eindeutige $K_x^\varphi \in X_0^n$ erhalten wir durch die Projektion von H_x^φ auf X_0^n , also

$$\begin{aligned}
K_x^\varphi(y) &= (I - P)H_x^\varphi(y) \\
&= H_x^\varphi(y) - P(H_x^\varphi)(y) \\
&= (-1)^m \left[E(x-y)\varphi - \sum_{i \in J} p_i(x) \bullet \varphi E(a_i - y)\varphi_i \right] \\
&\quad - (-1)^m \left[\sum_{j \in J} E(x - a_j)\varphi \bullet \varphi_j p_j(y) - \sum_{i \in J} \left(\sum_{j \in J} p_j(x) \bullet \varphi E(a_j - a_i)\varphi_j \bullet \varphi_i \right) p_i(y) \right] \\
&= (-1)^m \left[E(x-y)\varphi - \sum_{i \in J} p_i(x) \bullet \varphi E(a_i - y)\varphi_i \right. \\
&\quad \left. - \sum_{j \in J} E(x - a_j)\varphi \bullet \varphi_j p_j(y) + \sum_{i \in J} \sum_{j \in J} p_j(x) \bullet \varphi E(a_j - a_i)\varphi_j \bullet \varphi_i p_i(y) \right]. \tag{5.16}
\end{aligned}$$

Dies ist also die Form des entsprechenden Fréchet-Riesz Repräsentanten der Gleichung (5.8)

$$v(x) \bullet \varphi = \langle \varphi \delta_{(x)}, v \rangle = (K_x^\varphi, v)_m, \quad \forall v \in X_0^n, \forall \varphi, x \in \mathbb{R}^n,$$

der sogenannte vektorwertige *reproducing kernel*.

5.4 Vektorwertiger reproducing kernel

Angelehnt an [14] wollen wir die besondere Eigenschaft der gerade berechneten Funktion $K^\varphi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ hervorheben. So ist die bei Meinguet entwickelte Funktion $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ ein sogenannter *reproducing kernel*.

5.14 Definition (REPRODUCING KERNEL, [6])

Sei $A \subseteq \mathbb{R}^n$ eine Punktmenge. X ein Hilbert Funktionenraum mit Funktionalen auf A . Eine Funktion $K : A \times A \rightarrow \mathbb{R}$ heißt *reproducing kernel* für einen Raum X , falls

1. Für jedes feste $a \in A$ ist $K(\cdot, a)$ eine Funktion in X .
2. Für jede Funktion $v \in X$ und für jeden Punkt $a \in A$ ist die *reproducing property*

$$v(a) = (v(\cdot), K(\cdot, a))_X$$

erfüllt.

Die entscheidende *reproducing property* haben wir nach Meinguets Vorbild ebenfalls entwickelt (5.8), jedoch lässt sich Definition 5.14 nicht direkt auf den höherdimensionalen Fall anwenden. Alternativ definieren wir im Folgenden die verallgemeinerte Version, den vektorwertigen *reproducing kernel* (vgl. z.B. [2]).

5.15 Definition (VEKTORWERTIGER REPRODUCING KERNEL)

Seien $A, B \subseteq \mathbb{R}^n$ zwei Punktengen. X^n ein Hilbert Funktionenraum mit Funktionen $v : A \rightarrow \mathbb{R}^n$. Eine Funktion $K^\varphi : A \times A \rightarrow \mathbb{R}^n$ heißt *vektorwertiger reproducing kernel* für einen Raum X^n , falls

1. Für jedes feste $a \in A$, $\varphi \in B$ ist $K^\varphi(\cdot, a)$ eine Funktion in X^n .
2. Für jede Funktion $v \in X^n$ und für jeden Punkt $a \in A$ ist die *reproducing property*

$$v(a) \bullet \varphi = (v(\cdot), K^\varphi(\cdot, a))_X$$

erfüllt.

Dass $K^\varphi(x, y) := K_x^\varphi(y)$ genau die Eigenschaften, insbesondere die *reproducing property*, von Definition 5.15 erfüllt zeigt sich in (5.8), immer sofern $m > n/2$ gilt und somit X_0^n ein Hilbertraum ist.

In folgendem Lemma beweisen wir zwei elementare Eigenschaften des grade definierten vektorwertigen *reproducing kernels*.

5.16 Lemma

Sei K^φ ein vektorwertiger *reproducing kernel*. Dann gelten die Eigenschaften

1. $K^\varphi(x, y)$ ist *symmetrisch* in $x, y \in \mathbb{R}^n$
2. $K^\varphi(x, y) \bullet \varphi$ ist *positiv definit* für alle $\varphi \in \mathbb{R}^n$ in dem Sinne, dass für eine beliebige Menge $b_1, \dots, b_N \in \mathbb{R}^n$, $N \geq 1$ die $N \times N$ Matrix mit Einträgen $K^\varphi(b_i, b_j) \bullet \varphi$, $i, j \in [1, N]$ positiv semidefinit ist.

Beweis. zu 1):

Mit Hilfe der Darstellung durch den entsprechenden Frechet-Riesz Repräsentanten nach (5.8), bzw. allgemeiner durch die Anwendung der *reproducing property* zeigt sich

$$K^\varphi(x, y) \bullet \varphi = K_x^\varphi(y) \bullet \varphi = \langle \varphi \delta_{(y)}, K_x^\varphi \rangle \stackrel{5.8}{=} (K_x^\varphi, K_y^\varphi)_m. \quad (5.17)$$

Die Punktauswertung an $K^\varphi(x, y) \bullet \varphi$ kann somit als Skalarprodukt in X_0^n der Fréchet-Riesz Repräsentanten von $\varphi \delta_{(x)}$, bzw. $\varphi \delta_{(y)}$ interpretiert werden. Somit ist $K^\varphi(x, y)$ offensichtlich symmetrisch, denn

$$K^\varphi(x, y) \bullet \varphi = (K_x^\varphi, K_y^\varphi)_m = (K_y^\varphi, K_x^\varphi)_m = K^\varphi(y, x) \bullet \varphi.$$

zu 2):

Sei $x, \varphi \in \mathbb{R}^n$ und B die $N \times N$ Matrix mit den oben beschriebenen Einträgen. Dann folgt mit der

Identität (5.17) und der Bilinearität des Skalarprodukts

$$\begin{aligned}
x^t B x &= \sum_{i,j=1}^N x_i x_j K^\varphi(b_i, b_j) \bullet \varphi \stackrel{5.17}{=} \sum_{i,j=1}^N x_i x_j (K_{b_i}^\varphi, K_{b_j}^\varphi)_m \\
&= \sum_{i,j=1}^N (x_i K_{b_i}^\varphi, x_j K_{b_j}^\varphi)_m \\
&= \left(\sum_{i=1}^N x_i K_{b_i}^\varphi, \sum_{j=1}^N x_j K_{b_j}^\varphi \right)_m \\
&= \left| \sum_{i=1}^N x_i K_{b_i}^\varphi \right|_m^2 \geq 0.
\end{aligned} \tag{5.18}$$

Somit ist B positiv semidefinit. \square

5.5 Lösung des Problems (P)

Mit der neuen Darstellung einer Funktion $v \in X_0^n$ als *reproducing kernel* setzen wir nun die Suche nach dem minimalen $u \in V^n \subset X^n$ bezüglich $|\cdot|_m$ fort. Im folgenden Abschnitt erarbeiten wir eine Lösung des Problems (P).

Da die Seminorm $|\cdot|_m$ auf \mathcal{P}_{m-1}^n verschwindet, X^n zudem zu X_0^n isometrisch isomorph ist, können wir das Bild der Projektion $I - P : X^n \rightarrow X_0^n$ eingeschränkt auf V^n betrachten.

Wir suchen also nach minimalen $w \in W^n$, wobei

$$\begin{aligned}
W^n &:= \{v \in X_0^n \mid v = (I - P)w, w \in V^n\} \\
&= \{v \in X_0^n \mid v(a_i) \bullet \varphi_i = (I - P)w(a_i) \bullet \varphi_i, \forall i \in I\} \\
&= \{v \in X_0^n \mid v(a_k) \bullet \varphi_k = \alpha'_k \quad \forall k \in K\}
\end{aligned} \tag{5.19}$$

mit $K := I - J = \{k \in S \mid 1 \leq k \leq S, S := |K|\}$ und

$$\alpha'_k := \alpha_k - \sum_{j \in J} \alpha_j p_j(a_k) \bullet \varphi_k.$$

Hierbei ergibt sich (5.19) aus Anwendung der Projektion und Ausnutzung der Eigenschaft der Lagrange-Basis aus Satz 5.6, also

$$\begin{aligned}
v(a_i) \bullet \varphi_i &= (I - P)w(a_i) \bullet \varphi_i, & \forall i \in I \\
\Leftrightarrow v(a_i) \bullet \varphi_i &= w(a_i) \bullet \varphi_i - \sum_{j \in J} w(a_j) \bullet \varphi_j p_j(a_i) \bullet \varphi_i, & \forall i \in I \\
\Leftrightarrow v(a_i) \bullet \varphi_i &= \begin{cases} w(a_i) \bullet \varphi_i - w(a_i) \bullet \varphi_i = 0, & \forall i \in J \\ w(a_i) \bullet \varphi_i - \sum_{j \in J} \alpha_j p_j(a_i) \bullet \varphi_i, & \forall i \in K \end{cases} \\
\Leftrightarrow v(a_k) \bullet \varphi_k &= \alpha_k - \sum_{j \in J} \alpha_j p_j(a_k) \bullet \varphi_k, & \forall k \in K, \forall v \in X_0^n.
\end{aligned}$$

Bei Gleichung (5.8) haben wir bereits argumentiert, dass $\varphi_{\delta(x)}$ lineare stetige Funktionale sind. Wir

werden später sehen, dass die Existenz einer Lösung insbesondere von der linearen Unabhängigkeit der Funktionale $\varphi\delta_{(x)}$ abhängt. Im folgenden Satz geben wir Kriterien für diese wichtige Eigenschaft an.

5.17 Satz (LINEARE UNABHÄNGIGKEIT DER FUNKTIONALE)

Für $\varphi \neq 0$ und paarweise verschiedene $(a_k)_{k \in K}$ sind die stetigen linearen Funktionale $\varphi\delta_{(a_k)}$, $1 \leq k \leq S$ linear unabhängig.

Beweis. Seien also $(a_k)_{k \in K} \subset \mathbb{R}^n$ eine Menge paarweise verschiedener Punkte. Wir zeigen als erstes die lineare Unabhängigkeit der Funktionale $(\delta_{(a_k)})_{k \in K}$ auf $X_0 = \{v \in X \mid v(a_j) = 0 \ \forall j \in J\}$ (vgl. (5.6)).

Angenommen die entsprechenden linearen stetigen Funktionale $\delta_{(a_k)}$, $1 \leq k \leq S$ seien linear abhängig, d.h. o.B.d.A. existiere eine Darstellung für $\delta_{(a_1)}$, sodass nicht alle $c_k = 0$ für $c_k \in \mathbb{R}$, $2 \leq k \leq S$

$$\delta_{(a_1)} = \sum_{k=2}^S c_k \delta_{(a_k)}.$$

Dann gilt für beliebiges $v \in X_0$

$$\begin{aligned} \delta_{(a_1)}(v) &= \sum_{k=2}^S c_k \delta_{(a_k)}(v) \\ v(a_1) &= \sum_{k=2}^S c_k v(a_k) \end{aligned}$$

Da die Punkte $a_k \in \mathbb{R}^n$, $k \in K$ paarweise verschieden sind, existiert ein offener Ball U_{a_1} mit Radius r um a_1 , sodass $a_k \notin U_{a_1}$, $\forall 2 \leq k \leq S$, bzw. sogar $a_j \notin U_{a_1}$, $\forall j \in J$. Wähle dann z.B. $\phi : \mathbb{R} \rightarrow \mathbb{R}$ mit

$$\phi(x) = \begin{cases} \exp(-\frac{r^2}{r^2-x^2}), & \text{falls } x \in (-r, r) \\ 0, & \text{sonst,} \end{cases}$$

und entsprechend für $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$, $\psi(x) = \prod_{i=1}^n \phi(x_i - (a_1)_i)$. Dann ist ψ eine glatte Funktion mit kompakten Träger, und es gilt einerseits $\psi(x) \neq 0$ für alle $x \in U_{a_1}$, andererseits $\psi(x) = 0$ für alle $x \notin U_{a_1}$ und somit auch $\psi(a_k) = 0$ für alle $k > 1$. Außerdem ist ψ glatt mit kompaktem Träger, somit in X und weiterhin $\psi(a_j) = 0$ für $j \in J$, somit $\psi \in X_0$. Dann ist aber für $v = \psi$

$$0 \neq \psi(a_1) = \sum_{k=2}^S c_k \psi(a_k) = 0.$$

Dies ist ein Widerspruch. Die Funktionale $\varphi_{(a_k)}$ sind also linear unabhängig.

Für die Funktionale $\varphi\delta_{(a_k)}$ auf X_0^n für $\varphi \neq 0$ folgt die lineare Unabhängigkeit fast analog. Angenommen es existiere wieder eine Darstellung für $\varphi\delta_{(a_1)}$, also für nicht alle $c_k = 0$

$$\begin{aligned} \varphi\delta_{(a_1)} &= \sum_{k=2}^S c_k \varphi\delta_{(a_k)} \\ \Leftrightarrow v(a_1) \bullet \varphi &= \sum_{k=2}^S c_k v(a_k) \bullet \varphi, \quad \forall v \in X_0^n \end{aligned}$$

Definiere dann aber wie oben das entsprechende $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$ und damit ein n -dimensionales ψ^n mit der komponentenweisen Auswertung an ψ , also

$$\begin{aligned}\psi^n : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\ x &\mapsto (\psi(x), \dots, \psi(x))^t\end{aligned}$$

Dann ist ψ^n in X_0^n und da $\varphi \neq 0$ gilt

$$0 \neq \psi^n(a_1) \bullet \varphi = \sum_{k=2}^S c_k \psi^n(a_k) \bullet \varphi = 0,$$

denn $\psi^n(a_k) = (0, \dots, 0)^t$ für alle $k > 1$. Somit sind für $\varphi \neq 0$ die Funktionale $\varphi \delta_{(a_k)}$ linear unabhängig. \square

Die Elemente, die im Bild der Projektion $(I - P)|_{V^n}$ liegen, also genau die Funktionen in W^n nach (5.19), lassen sich als eine affine Verschiebung der Menge W_0 schreiben

$$W_0^n := \{v \in X_0^n \mid v(a_k) \bullet \varphi_k = 0 \ \forall k \in K\}. \quad (5.20)$$

Die Menge $\{v \in X_0^n \mid v(a_k) \bullet \varphi_k = 0\}$ ist als Urbild des Kerns der stetigen linearen Funktionale $\varphi \delta_{(a_k)}$ für beliebige $k \in K$ ein abgeschlossener Untervektorraum. Somit ist W_0^n ebenfalls ein abgeschlossener Untervektorraum als Schnitt der Untervektorräume über die (endliche) Anzahl $k \in K$. Dieser ist nichtleer, da nach Satz 5.17 die Funktionale $\varphi \delta_{(a_k)}$ linear unabhängig sind und somit existiert nach [6, S.226, Theo 9.4.1.] mindestens ein solches $v \in X_0^n$.

Sei nun $v \in X_0^n$ mit $v(a_k) \bullet \varphi_k = \alpha'_k$ beliebig. Dann lässt sich W^n schreiben als der affine Unterraum zu W_0^n , also

$$W^n = v + W_0^n.$$

5.18 Satz (ORTHOGONALITÄTSBEDINGUNG FÜR w)

In $W^n \subset X_0^n$ existiert ein eindeutiges w minimaler Norm, welches die Orthogonalitätsbedingung

$$(w, v)_m = 0, \ \forall v \in W_0^n$$

erfüllt.

Beweis. Wähle $v \in W^n \subset X_0^n$ beliebig.

Dann lässt sich W_0^n beschreiben durch $W_0^n = W^n - v$. Definiere nun $y := -v$. Da W_0^n ein abgeschlossener Untervektorraum ist, existiert nach dem Projektionssatz [21, S.350, Theo.A.5.] ein eindeutiges $x \in W_0^n$, sodass die Orthogonalitätseigenschaft

$$(x - y, z)_m = 0, \ \forall z \in W_0^n$$

erfüllt ist, sowie der Abstand zu y minimal ist, d.h.

$$|x - y|_m \leq |\hat{x} - y|_m, \ \forall \hat{x} \in W_0^n.$$

Dann erhalten wir mit Einsetzen von $y := -v \in W^n$ und entsprechendem $x = w - v \in W_0^n$ mit $w \in W^n$

$$\begin{aligned} |x - y|_m &\leq |\hat{x} - y|_m, & \forall \hat{x} \in W_0^n \\ \Leftrightarrow |w - v - (-v)|_m &\leq |\hat{w} - v - (-v)|_m, & \forall \hat{w} \in W^n \\ \Leftrightarrow |w|_m &\leq |\hat{w}|_m, & \forall \hat{w} \in W^n. \end{aligned}$$

Somit ist w minimal in W^n . Weiter gilt nach der Orthogonalitätseigenschaft

$$\begin{aligned} (x - y, z)_m = 0 &\Leftrightarrow (w - v + v, z)_m = 0 & \forall z \in W_0^n \\ \Leftrightarrow (w, z)_m &= 0 & \forall z \in W_0^n. \end{aligned}$$

□

Die gerade bewiesene Orthogonalitätsbedingung eines minimalen $w \in W^n$ können wir nun zur Lösung des Problems nutzen. Alternativ zu der Definition (5.20) können wir die Menge W_0^n nach (5.8) schreiben als Menge aller $v \in X_0^n$, die orthogonal zu den Riesz-Repräsentanten $K_{a_k}^{\varphi_k}, \forall k \in K$ liegen, nämlich

$$W_0^n \stackrel{(5.20)}{=} \{v \in X_0^n \mid v(a_k) \bullet \varphi_k = 0 \ \forall k \in K\} \stackrel{(5.8)}{=} \{v \in X_0^n \mid (v, K_{a_k}^{\varphi_k})_m = 0 \ \forall k \in K\}.$$

Dann folgt mit der Orthogonalitätsbedingung aus Satz 5.18, dass ein minimales $w \in W^n$ im Spann von $\{K_{a_k}^{\varphi_k} \mid 1 \leq k \leq S\}$ liegt und dementsprechend von der Form

$$w = \sum_{k=1}^S \gamma_k K_{a_k}^{\varphi_k}, \quad \gamma_k \in \mathbb{R} \quad (5.21)$$

ist. Mit Hilfe dieser Summendarstellung und der Definition von W^n in (5.19) erhalten wir das lineare Gleichungssystem zum Bestimmen der reellen Koeffizienten γ_i

$$\sum_{j=1}^S \gamma_j K^{\varphi_j}(a_i, a_j) \bullet \varphi_i = \alpha'_i, \quad 1 \leq i \leq S. \quad (5.22)$$

5.19 Satz (EXISTENZ EINER EINDEUTIGEN LÖSUNG)

Das Problem (P), nach Definition 5.7, hat eine eindeutige Lösung, sofern $\varphi_k \neq 0$ für alle $k \in K$.

Beweis. Die Koeffizientenmatrix von Gleichung (5.22) ist nach Lemma 5.16 symmetrisch und positiv semidefinit. Für die paarweise verschiedenen $(a_i) \in A$, $1 \leq i \leq S$ und $\varphi_k \neq 0$, sind die entsprechenden $K_{a_i}^{\varphi_i} \in X_0^n$ nach Satz 5.17 linear unabhängig und somit gilt für Gleichung (5.18)

$$\left| \sum_{i=1}^S x_i K_{b_i}^{\varphi_i} \right|_m^2 = 0 \Leftrightarrow x_i = 0, \quad 1 \leq i \leq S.$$

Somit ist die Koeffizientenmatrix sogar positiv definit und damit nichtsingulär.

Mit der Nichtsingularität folgt die eindeutige Lösbarkeit der Gleichung (5.22) und ebenso des

Problems (P) mit

$$u = w + p = \sum_{k=1}^S \gamma_k K_{a_k}^{\varphi_k} + \sum_{j \in J} \alpha_j p_j.$$

Denn w ist minimal auf X_0^n und somit auf X^n bezüglich $|\cdot|_m$ und es gilt für alle $j \in J$

$$\begin{aligned} u(a_j) \bullet \varphi_j &= \sum_{k=1}^S \gamma_k K_{a_k}^{\varphi_k}(a_j) \bullet \varphi_j + \sum_{i \in J} \alpha_i p_i(a_j) \bullet \varphi_j \\ &= 0 + \alpha_j, \end{aligned}$$

denn $K_{a_k}^{\varphi_k} \in X_0^n$, $1 \leq k \leq S$ und $p_j(a_i) \bullet \varphi_i = \delta_{ji}$, $\forall i, j \in J$.

Für $k \in K$ gilt entsprechend

$$\begin{aligned} u(a_k) \bullet \varphi_k &= \sum_{i=1}^S \gamma_i K_{a_i}^{\varphi_i}(a_k) \bullet \varphi_k + \sum_{i \in J} \alpha_i p_i(a_k) \bullet \varphi_k \\ &\stackrel{(5.22)}{=} \alpha'_k + \sum_{i \in J} \alpha_i p_i(a_k) \bullet \varphi_k \\ &\stackrel{(Def.)}{=} \alpha_k - \sum_{j \in J} \alpha_j p_j(a_k) \bullet \varphi_k + \sum_{i \in J} \alpha_i p_i(a_k) \bullet \varphi_k = \alpha_k. \end{aligned}$$

Insgesamt sind also die Interpolationsbedingungen für alle $a \in A$ eindeutig erfüllt, $u \in V^n$ ist eine eindeutige Lösung. \square

Jedes optimale $u \in X$ liegt somit im Schnitt der Räume

$$u = V_0^{n\perp} \cap V^n,$$

wobei V^n , definiert durch (5.4), der affine Unterraum des Untervektorraumes

$$V_0^n := \{v \in X^n \mid v(a_i) \bullet \varphi_i = 0, \forall i \in I\} = \{v \in X_0^n \mid v(a_k) \bullet \varphi_k = 0 \forall k \in K\} = W_0^n, \quad (5.23)$$

ist. Das $u \in V^n$ gilt, wurde in Satz 5.19 aufgezeigt und es ist offensichtlich, dass V^n die affine Variation aller X^n -Interpolanten auf A zu *gegebenen Daten* $(\alpha_i)_{i \in I}, (\varphi_i)_{i \in I}$ ist. Hingegen ist $V_0^{n\perp}$ der Untervektorraum aller optimalen X^n -Interpolanten zu *beliebigen Daten* $(\alpha_i)_{i \in I}$ auf A mit $(\varphi_i)_{i \in I}$. Dies ist ersichtlich aus der Orthogonalitätseigenschaft des minimalen $w \in W_0^{n\perp} = V_0^{n\perp}$, aufgezeigt im Beweis von Satz 5.18.

5.20 Bemerkung

Auch wenn in Satz 5.19 nicht explizit die Bedingung

$$\varphi_j \neq 0$$

für alle $j \in J$ genannt wird, so ist diese in der Formulierung von Problem (P) (Def. 5.7) enthalten. Dort hatten wir die \mathcal{P}_{m-1}^n -Unisolvenz der Menge $(a_j)_{j \in J}$ gefordert und diese garantiert nach Satz 5.6 die Existenz einer Lagrange-Basis $(p_i)_{i \in J} \in \mathcal{P}_{m-1}^n$ mit

$$p_i(a_j) \bullet \varphi_j = \delta_{ij}, \forall i, j \in J.$$

Ist $\varphi_{\hat{j}} = 0$ für beliebiges $\hat{j} \in J$, wird diese Bedingung selbstverständlich verletzt und die \mathcal{P}_{m-1}^n -Unisolvenz ist nicht gegeben.

Somit wird für die Existenz einer Lösung insgesamt $\varphi_i \neq 0$ für alle $i \in I$ gefordert.

5.6 Lösen des Gleichungssystems

Mit Hilfe des Gleichungssystems (5.22) und einer entsprechenden Lagrange-Basis können wir bereits eine Lösung für ein optimales u berechnen. Folgender Satz liefert uns zusätzlich Gleichungen für ein lineares Gleichungssystem, um diese in nur einem Schritt zu berechnen.

5.21 Satz

Sei $(a_i)_{i \in I}$, $(\varphi_i)_{i \in I} \in \mathbb{R}^n$ mit $\varphi_i \neq 0, \forall i \in I$ wie im vorhergegangenen Kapitel und es gelte $m > n/2$. Dann hat jedes $w \in V_0^{n \perp}$ eine eindeutige Darstellung der Form

$$w(y) = (-1)^m \sum_{i \in I} \gamma_i E(a_i - y) \varphi_i + q(y),$$

mit $q \in \mathcal{P}_{m-1}^n$ und $\gamma_i \in \mathbb{R}$ und für diese $(\gamma_i)_{i \in I}$ gilt

$$\sum_{i \in I} \gamma_i \varphi_i \bullet p(a_i) = 0 \quad \forall p \in \mathcal{P}_{m-1}^n.$$

Beweis. Sei $w \in V_0^{n \perp}$. Aus der eindeutigen Darstellung des für $w \in W_0^{n \perp} = V_0^{n \perp}$ in (5.21) folgt

$$\begin{aligned} w(y) &= \sum_{k=1}^S \gamma_k K_{a_k}^{\varphi_k}(y) \\ &\stackrel{(5.16)}{=} \sum_{k=1}^S \gamma_k (-1)^m \left[E(a_k - y) \varphi_k - \sum_{i \in J} p_i(a_k) \bullet \varphi_k E(a_i - y) \varphi_i \right. \\ &\quad \left. - \sum_{j \in J} E(a_k - a_j) \varphi_k \bullet \varphi_j p_j(y) + \sum_{i \in J} \sum_{j \in J} p_j(a_k) \bullet \varphi_k E(a_j - a_i) \varphi_j \bullet \varphi_i p_i(y) \right], \end{aligned}$$

nach Einsetzen der Definition von $K_{a_k}^{\varphi_k}$ nach (5.16). Umstellen der Summanden und die Definition der Parameter γ_i für $i \in J$ führt zu

$$\begin{aligned} &= (-1)^m \left[\sum_{k=1}^S \gamma_k E(a_k - y) \varphi_k + \underbrace{\sum_{i \in J} - \sum_{k=1}^S \gamma_k p_i(a_k) \bullet \varphi_k E(a_i - y) \varphi_i}_{=:\gamma_i, i \in J} \right. \\ &\quad \left. - \sum_{k=1}^S \gamma_k \sum_{j \in J} E(a_k - a_j) \varphi_k \bullet \varphi_j p_j(y) + \sum_{k=1}^S \gamma_k \sum_{i \in J} \sum_{j \in J} p_j(a_k) \bullet \varphi_k E(a_j - a_i) \varphi_j \bullet \varphi_i p_i(y) \right], \\ &\qquad\qquad\qquad =: q(y) \end{aligned}$$

dabei ist $q(y)$ ein n -dimensionales Polynom vom Grad $\leq m-1$ als Linearkombination der Lagrange-Basis $p_i(y) \in \mathcal{P}_{m-1}^n$ und somit

$$\begin{aligned} &= (-1)^m \left[\sum_{k=1}^S \gamma_k E(a_k - y) \varphi_k + \sum_{i \in J} \gamma_i E(a_i - y) \varphi_i + q(y) \right] \\ &= (-1)^m \sum_{i \in I} \gamma_i E(a_i - y) \varphi_i + q(y). \end{aligned}$$

Nun bleibt die zweite Behauptung

$$\sum_{i \in I} \gamma_i \varphi_i \bullet p(a_i) = 0 \quad \forall p \in \mathcal{P}_{m-1}^n$$

zu zeigen. Sei dazu $(p_l)_{l \in J} \in \mathcal{P}_{m-1}^n$ eine Basis mit $p_l(a_k) \bullet \varphi_k = \delta_{lk} \quad \forall l, k \in J$, welche nach Satz 5.6 existiert. Dann gilt mit Einsetzen der Definition von $(\gamma_i)_{i \in J}$

$$\begin{aligned} \sum_{i \in I} \gamma_i \varphi_i \bullet p_l(a_i) &= \sum_{j \in J} \gamma_j \varphi_j \bullet p_l(a_j) + \sum_{k=1}^S \gamma_k \varphi_k \bullet p_l(a_k) \\ &= \sum_{j \in J} \left(- \sum_{k=1}^S \gamma_k p_j(a_k) \bullet \varphi_k \right) \underbrace{\varphi_j \bullet p_l(a_j)}_{\delta_{lj}} + \sum_{k=1}^S \gamma_k \varphi_k \bullet p_l(a_k) \\ &= - \sum_{k=1}^S \gamma_k p_l(a_k) \bullet \varphi_k + \sum_{k=1}^S \gamma_k \varphi_k \bullet p_l(a_k) \\ &= 0. \end{aligned}$$

Die Eindeutigkeit der Darstellung folgt direkt aus der Eindeutigkeit der Parameter γ_i nach Satz 5.19, immer sofern $\varphi_i \neq 0, \forall i \in I$. \square

5.22 Beispiel

Für $m = n = 2$ hat ein minimales w die Form

$$\begin{aligned} w(y) &= (-1)^m \sum_{i \in I} \gamma_i E(a_i - y) \varphi_i + q(y) \\ &= \sum_{i \in I} \gamma_i \frac{1}{8\pi} |a_i - y|^2 \log(|a_i - y|) \varphi_i + \begin{pmatrix} \beta_1 \\ \beta_4 \end{pmatrix} + \begin{pmatrix} \beta_2 & \beta_3 \\ \beta_5 & \beta_6 \end{pmatrix} y. \end{aligned}$$

Wenn wir uns an Satz 3.1 aus dem Kapitel über *Thin-Plate Splines* erinnern, erkennen wir, dass wir unser Ziel erreicht haben. Wir haben eine eindeutige Form von w entwickelt, welche die Interpolationsbedingungen, die durch ein *Optical Flow* Problem gestellt werden, nach dem Vorbild der *Thin-Plate Splines* interpoliert.

Im nächsten Kapitel wollen wir einen alternativen Ansatz erarbeiten, in dem die Interpolationsbedingung nicht als Beschränkung des Funktionenraums dient, sondern lediglich die zu minimierende Norm in einem Regularisierungsterm bestraft. Eine exakte Erfüllung der Interpolationsbedingung wird somit nicht mehr gefordert.

6 Inexaktes Interpolationsproblem

Wie bei Gleichung (5.23) argumentiert, bezeichnet die Menge $V_0^{n\perp}$ die lineare Variation aller Interpolanten auf X^n zu beliebigen Daten $(\alpha_i)_{i \in I}$ an den Punkten $A = (a_i)_{i \in I} \in \mathbb{R}^n$, in Abhängigkeit von $(\varphi_i)_{i \in I} \in \mathbb{R}^n$. In diesem Ansatz, für klassische *Thin-Plate Spline* Interpolation, z.B. beschrieben in [21, S.199ff.], wollen wir auf die strenge Erfüllung der Interpolationsbedingungen für ein $v \in X^n$ verzichten. Welche Werte v genau auf A annimmt, wollen wir zu dessen Bestimmung nicht benutzen, ausschließlich die Abweichung von den Interpolationsbedingung bestrafen. Die Forderung, dass v aus $V_0^{n\perp}$ gewählt werden soll ist somit gerechtfertigt. Weiterhin minimieren wir die Funktion bezüglich der Norm $|\cdot|_m$, die Intuition der TPS Interpolation bleibt erhalten.

Glücklicherweise haben wir bereits in Satz 5.21 einen Ausdruck von $v \in V_0^{n\perp}$ in Abhängigkeit von den Punkten $(a_i)_{i \in I}, (\varphi_i)_{i \in I} \in \mathbb{R}^n$ entwickelt und können diesen im Folgenden weiter nutzen.

6.1 Definition (INEXAKTES INTERPOLATIONSPROBLEM)

Sei $n, m \in \mathbb{N}$ mit $n, m \geq 1$ und $m > n/2$.

Zudem sei $A = (a_i)_{i \in I}$ eine endliche Menge paarweise verschiedener Punkte in \mathbb{R}^n , sodass eine \mathcal{P}_{m-1}^n -unisolvante Untermenge $B = (a_j)_{j \in J}$ mit $N = n \cdot M$ Punkten existiert, M erneut durch Lemma 5.1 bestimmt. Dann bezeichnet die Suche nach einem $u \in V_0^{n\perp}$, sodass

$$U(u) = \inf_{v \in V_0^{n\perp}} U(v) = \inf_{v \in V_0^{n\perp}} |v|_m^2 + \lambda \sum_{i \in I} (v(a_i) \bullet \varphi_i - \alpha_i)^2$$

für den Regularisierungsparameter $\lambda > 0 \in \mathbb{R}$ erfüllt ist, das *inexakte Interpolationsproblem*.

Wir wollen das Funktional U im Anbetracht der Ergebnisse aus dem vergangenen Kapitel genauer betrachten.

Nach Satz 5.21 hat jedes $v \in V_0^{n\perp}$ die Form

$$v(x) = \sum_{i \in I} \gamma_i E(a_i - x) \varphi_i + q(x)$$

für $q(x) \in \mathcal{P}_{m-1}^n, \gamma_i \in \mathbb{R}$, sofern $\varphi_i \neq 0, \forall i \in I$.

Wir betrachten als erstes den vorderen Teil des Funktionals $U(v)$, die Seminorm von v . Einsetzen der Summenschreibweise von v liefert

$$\begin{aligned} |v|_m^2 &= (v, v)_m \\ &= \left(\sum_{i \in I} \gamma_i E(a_i - \cdot) \varphi_i + q(\cdot), \sum_{j \in I} \gamma_j E(a_j - \cdot) \varphi_j + q(\cdot) \right)_m \\ &= \sum_{i, j \in I} \gamma_i \gamma_j (E(a_i - \cdot) \varphi_i, E(a_j - \cdot) \varphi_j)_m \\ &= \sum_{i, j \in I} \gamma_i \gamma_j \langle (-1)^m \Delta E(a_i - \cdot) \varphi_i, E(a_j - \cdot) \varphi_j \rangle \\ &= \sum_{i, j \in I} \gamma_i \gamma_j \langle \varphi_i \delta_{(a_i)}, E(a_j - \cdot) \varphi_j \rangle, \end{aligned}$$

da $E(\cdot)$ nach Lemma 5.11 genau die (distributionelle) Lösung von $\Delta E = \delta$ ist. Schließlich folgt mit der Auswertung an $\varphi_i \delta_{(a_i)}$

$$= \sum_{i,j \in I} \gamma_i \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i. \quad (6.1)$$

Der zweite Term des Funktionals U , der Abweichungen von den Interpolationsbedingungen im Quadrat bestraft, lässt sich mit der Summenform von v schreiben als

$$\begin{aligned} \sum_{i \in I} (v(a_i) \bullet \varphi_i - \alpha_i)^2 &= \sum_{i \in I} \left(\sum_{j \in I} \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + q(a_i) \bullet \varphi_i - \alpha_i \right)^2 \\ &= \sum_{i \in I} \left(\sum_{j \in I} \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + \sum_{|e_j| \leq m-1} \beta_{e_j} a_i^{e_j} \bullet \varphi_i - \alpha_i \right)^2. \end{aligned}$$

Wir können das zu minimierende Funktional nun also in Abhängigkeit von den Parametern γ und β schreiben. Dann ist für $\gamma = (\gamma_1, \dots, \gamma_T) \in \mathbb{R}^T$ und $\beta = (\beta_{e_1}, \dots, \beta_{e_M})$

$$U(\gamma, \beta) = \sum_{i,j \in I} \gamma_i \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + \lambda \sum_{i \in I} \left(\sum_{j \in I} \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + \sum_{|e_j| \leq m-1} \beta_{e_j} a_i^{e_j} \bullet \varphi_i - \alpha_i \right)^2. \quad (6.2)$$

Diese Darstellung ist, sofern $\varphi_i \neq 0$, $\forall i \in I$, aufgrund der Eindeutigkeit der Parameter γ für ein festes $\lambda \in \mathbb{R}$ wieder eindeutig.

Die Existenz einer Lösung für das inexakte Interpolationsproblem wird im folgenden Kapitel bewiesen.

7 Matrixformen der Interpolationsprobleme

Um das exakte, sowie das inexakte Interpolationsproblem zu implementieren, ist die Darstellung als lineares Gleichungssystem sinnvoll. Im folgenden Kapitel erarbeiten wir eine Matrixdarstellung, die diesem Gleichungssystem entspricht.

7.1 Exakte Interpolation

Nach Satz 5.21 können wir ein optimales $v \in X^n$ für beliebige Daten schreiben als

$$v(x) = (-1)^m \sum_{i \in I} \gamma_i E(a_i - x) \varphi_i + q(x),$$

mit $q \in \mathcal{P}_{m-1}^n$. Außerdem muss nach der zweiten Gleichung des Satzes die Bedingung

$$\sum_{i \in I} \gamma_i \varphi_i \bullet q(a_i) = 0$$

an ein solches q erfüllt sein.

Anhand dieser Gleichungen und den Interpolationsbedingungen

$$\begin{aligned} v(a_i) \bullet \varphi_i &= \alpha_i, \quad \forall i \in I \\ \Leftrightarrow (-1)^m \sum_{j \in I} \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + q(a_i) \bullet \varphi_i &= \alpha_i, \quad \forall i \in I, \end{aligned}$$

können wir folgende Matrixschreibweise entwickeln.

Bezeichne mit \mathcal{S} eine $T \times T$ Matrix mit Einträgen

$$s_{ij} = \begin{cases} E(a_j - a_i) \varphi_j \bullet \varphi_i, & i \neq j \\ 0, & \text{sonst,} \end{cases}$$

T die Anzahl der paarweise verschiedenen Punkte $(a_i)_{i \in I}$. Aufgrund der Symmetrie von $E(\cdot, \cdot)$ und \bullet , gilt

$$s_{ij} = E(a_j - a_i) \varphi_j \bullet \varphi_i = E(a_i - a_j) \varphi_i \bullet \varphi_j = s_{ji},$$

\mathcal{S} ist offensichtlich eine symmetrische Matrix mit Diagonaleinträgen 0.

Nach Satz 5.1 hat \mathcal{P}_{m-1} maximal $M = \binom{n+m-1}{n}$ Monome und somit auch M Koeffizienten, für \mathcal{P}_{m-1}^n ergeben sich also $N := n \cdot M$ gesuchte Koeffizienten. Wir beschreiben mit

$$P := \begin{pmatrix} \varphi_{11}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] & \cdots & \varphi_{1n}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] \\ \varphi_{21}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] & \cdots & \varphi_{2n}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] \\ \vdots & & \vdots \\ \varphi_{T1}[a_T^{e_1}, a_T^{e_2}, \dots, a_T^{e_M}] & \cdots & \varphi_{Tn}[a_T^{e_1}, a_T^{e_2}, \dots, a_T^{e_M}] \end{pmatrix} \quad (7.1)$$

die $T \times N$ Matrix der Monome bis zum Grad $m-1$, ausgewertet an den Punkten $(a_i)_{i \in I}$.

Dann entspricht das lineare Gleichungssystem

$$\begin{bmatrix} \mathcal{S} & P \\ P^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad (7.2)$$

genau den Interpolationsbedingungen,

mit den gegebenen Daten $\alpha^t = (\alpha_1, \dots, \alpha_T)$ und $\gamma^t = (\gamma_1, \dots, \gamma_T)$ die gesuchten Koeffizienten, und $\beta^t = (\beta_1, \dots, \beta_N)$ die Koeffizienten des Polynoms $q \in \mathcal{P}_{m-1}^n$.

Die Lösung des linearen Gleichungssystems ist dann durch

$$\begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \mathcal{S} & P \\ P^t & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad (7.3)$$

gegeben.

Nach Satz 5.21 existiert zu gegebenen Daten $(\alpha_i)_{i \in I}, (\varphi_i)_{i \in I}$ mit $\varphi_i \neq 0, \forall i \in I$ an den Punkten $(a_i)_{i \in I} \in \mathbb{R}^n$ genau eine eindeutige Lösung des Problems. Das entsprechende lineare Gleichungssystem (7.2) hat somit eine eindeutige Lösung, die Koeffizientenmatrix ist invertierbar.

7.1 Beispiel

Sei $m = n = 2$. Dann ist $N = n \cdot M = 6$ und das gesuchte Polynom $q \in \mathcal{P}_1^2$ hat die Form

$$q(x) = \begin{pmatrix} \beta_1 \\ \beta_4 \end{pmatrix} + \begin{pmatrix} \beta_2 & \beta_3 \\ \beta_5 & \beta_6 \end{pmatrix} x.$$

Die zugehörigen Matrizen des Gleichungssystems für $|I| = T$ Punkte in \mathbb{R}^2 , wobei $T \geq N$ erfüllt sein muss, sind daher

$$\mathcal{S} = \begin{bmatrix} 0 & E(a_1 - a_2)\varphi_1 \bullet \varphi_2 & E(a_1 - a_3)\varphi_1 \bullet \varphi_3 & \cdots & E(a_1 - a_T)\varphi_1 \bullet \varphi_T \\ E(a_2 - a_1)\varphi_2 \bullet \varphi_1 & 0 & E(a_2 - a_3)\varphi_2 \bullet \varphi_3 & \cdots & E(a_2 - a_T)\varphi_2 \bullet \varphi_T \\ E(a_3 - a_1)\varphi_3 \bullet \varphi_1 & E(a_3 - a_2)\varphi_3 \bullet \varphi_2 & 0 & \cdots & E(a_3 - a_T)\varphi_3 \bullet \varphi_T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E(a_T - a_1)\varphi_T \bullet \varphi_1 & E(a_T - a_2)\varphi_T \bullet \varphi_2 & E(a_T - a_3)\varphi_T \bullet \varphi_3 & \cdots & 0 \end{bmatrix}$$

mit $\mathcal{S} \in \mathbb{R}^{T \times T}$ und

$$P = \begin{bmatrix} \varphi_{11} & \varphi_{11}a_{11} & \varphi_{11}a_{12} & \varphi_{12} & \varphi_{12}a_{11} & \varphi_{12}a_{12} \\ \varphi_{21} & \varphi_{21}a_{21} & \varphi_{21}a_{22} & \varphi_{22} & \varphi_{22}a_{21} & \varphi_{22}a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{T1} & \varphi_{T1}a_{61} & \varphi_{T1}a_{T2} & \varphi_{T2} & \varphi_{T2}a_{T1} & \varphi_{T2}a_{T2} \end{bmatrix} \in \mathbb{R}^{T \times 6}.$$

7.2 Inexakte Interpolation

Auch die Funktion $U(\gamma, \beta)$ aus (6.2) des inexakten Interpolationsproblems können wir erneut mit der Matrixschreibweise übersichtlicher gestalten. Wie sich herausstellt, ist für die Aufstellung der Matrixform des inexakten Problems nur eine leichte Modifikation der vorausgegangen exakten Version nötig.

7.2 Satz (MATRIXFORM DES INEXAKTEN PROBLEMS)

Das inexakte Interpolationsproblem (Def. 6.1) kann als lineares Gleichungssystem ausgedrückt werden durch

$$\begin{bmatrix} \mathcal{S} + 1/\lambda Id & P \\ P^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad (7.4)$$

mit Matrizen \mathcal{S} , P definiert wie in Abschnitt 7.1, sofern $\varphi_i \neq 0$, $\forall i \in I$.

Beweis. Definiere dazu, genau so wie in Abschnitt 7.1, die $T \times T$ Matrix \mathcal{S} mit Einträgen

$$s_{ij} = \begin{cases} E(a_j - a_i)\varphi_j \bullet \varphi_i, & i \neq j \\ 0, & \text{sonst.} \end{cases}$$

Anhand der Gleichung (6.1) können wir die Seminorm $|v|_m^2$ durch die Matrix \mathcal{S} ausdrücken durch

$$\begin{aligned} |v|_m^2 &\stackrel{(6.1)}{=} \sum_{i,j \in I} \gamma_i \gamma_j E(a_j - a_i)\varphi_j \bullet \varphi_i \\ &= \gamma^t \mathcal{S} \gamma, \end{aligned}$$

mit $\gamma^t = (\gamma_1, \dots, \gamma_T)$.

Für den Regulierungsterm definiere genau wie in (7.1) die Matrix $P \in \mathbb{R}^{T \times N}$ mit

$$P = \begin{pmatrix} \varphi_{11}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] & \cdots & \varphi_{1n}[a_1^{e_1}, a_1^{e_2}, \dots, a_1^{e_M}] \\ \varphi_{21}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] & \cdots & \varphi_{2n}[a_2^{e_1}, a_2^{e_2}, \dots, a_2^{e_M}] \\ \vdots & & \vdots \\ \varphi_{T1}[a_T^{e_1}, a_T^{e_2}, \dots, a_T^{e_M}] & \cdots & \varphi_{Tn}[a_T^{e_1}, a_T^{e_2}, \dots, a_T^{e_M}] \end{pmatrix}.$$

Dann ist

$$\begin{aligned} & v(a_i) \bullet \varphi_i - \alpha_i \\ &= \sum_{j \in J} \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + \sum_{|e_j| \leq m-1} \beta_{e_j} a_i^{e_j} \bullet \varphi_i - \alpha_i \\ &= (S\gamma)_i + (P\beta)_i - (\alpha)_i \end{aligned}$$

mit $\beta^t = ([\beta_{e_{11}}, \dots, \beta_{e_{M1}}], \dots, [\beta_{e_{1n}}, \dots, \beta_{e_{Mn}}]) \in \mathbb{R}^N$, $\alpha = (\alpha_1, \dots, \alpha_T) \in \mathbb{R}^T$.

Für den quadratischen Fehler erhalten wir in Matrixschreibweise den Ausdruck

$$\begin{aligned} \lambda \sum_{i \in I} (v(a_i) \bullet \varphi_i - \alpha_i)^2 &= \lambda \sum_{i \in I} \left(\sum_{j \in J} \gamma_j E(a_j - a_i) \varphi_j \bullet \varphi_i + \sum_{|e_j| \leq m-1} \beta_{e_j} a_i^{e_j} \bullet \varphi_i - \alpha_i \right)^2 \\ &= \lambda (S\gamma + P\beta - \alpha)^t (S\gamma + P\beta - \alpha). \end{aligned}$$

Die gesamte Funktion $E(\gamma, \beta)$ schreiben wir also mit

$$E(\gamma, \beta) = \gamma^t S \gamma + \lambda (S\gamma + P\beta - \alpha)^t (S\gamma + P\beta - \alpha). \quad (7.5)$$

Da wir diese Funktion nach den Parametern γ und β minimieren möchten, berechnen wir nun deren partielle (vektorwertige) Ableitungen:

$$\begin{aligned} \frac{\partial E}{\partial \gamma} &= \gamma^t (S + S^t) + \lambda [S(S\gamma + P\beta - \alpha) + (S\gamma + P\beta - \alpha)^t S] \\ &= 2S + 2\lambda S(S\gamma + P\beta - \alpha), \end{aligned}$$

da S symmetrisch, und

$$\begin{aligned} \frac{\partial E}{\partial \beta} &= \lambda [P^t (S\gamma + P\beta - \alpha) + (S\gamma + P\beta - \alpha)^t P] \\ &= 2\lambda P^t (S\gamma + P\beta - \alpha). \end{aligned}$$

Notwendigerweise müssen die partiellen Ableitungen am Minimum verschwinden, somit folgt für

$$\frac{\partial E}{\partial \gamma} = 0$$

$$\begin{aligned}
\frac{\partial E(\gamma, \beta)}{\partial \gamma} = 0 &\Leftrightarrow 0 = 2\mathcal{S} + 2\lambda\mathcal{S}(\mathcal{S}\gamma + P\beta - \alpha) \\
&\Leftrightarrow 0 = \gamma + \lambda\mathcal{S}\gamma + \lambda P\beta - \lambda\alpha \\
&\Leftrightarrow 0 = (\mathcal{S} + 1/\lambda Id)\gamma + P\beta - \alpha
\end{aligned} \tag{7.6}$$

mit Id die Identitätsmatrix. Für den Parameter β hingegen gilt

$$\begin{aligned}
\frac{\partial E(\gamma, \beta)}{\partial \beta} = 0 &\Leftrightarrow 0 = P^t(\mathcal{S}\gamma + P\beta - \alpha) \\
&\Leftrightarrow 0 = P^t\mathcal{S}\gamma + P^tP\beta - P^t\alpha \\
&\Leftrightarrow P^t\alpha = P^t\mathcal{S}\gamma + P^tP\beta.
\end{aligned} \tag{7.7}$$

Aus $P^t \cdot (7.6) - (7.7)$ folgt außerdem $P\gamma = 0$, welches das lineare Gleichungssystem

$$\begin{bmatrix} \mathcal{S} + 1/\lambda Id & P \\ P^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \tag{7.8}$$

ergibt. Für entsprechende γ und β , die das lineare Gleichungssystem (7.8) erfüllen, verschwindet, somit die partiellen Ableitungen von E . Da E als Positivkombination von konvexen Funktionen selber konvex ist, nimmt die Funktion hier sogar ihr globales Minimum an. \square

Um die Existenz einer eindeutigen Lösung für das inexakte Problem zu beweisen, benötigen wir das Schur-Komplement.

7.3 Definition

Sei M eine Blockmatrix definiert durch

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

sodass A, B, C und D jeweils $p \times p, p \times q, q \times p$ und $q \times q$ Matrizen sind. Ist A nichtsingulär, so nennt man die Matrix

$$M/A := D - CA^{-1}B$$

das *Schur-Komplement* von A der Matrix M .

7.4 Lemma

Sei M eine Blockmatrix wie in 7.3 und sei A nichtsingulär. Dann gilt:

1. Die Lösungen des linearen Gleichungssystems

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

lassen sich schreiben durch

$$y = (M/A)^{-1}(g - CA^{-1}f), \quad x = A^{-1}(f - By).$$

2. Die Matrix M/A ist nichtsingulär genau dann, wenn M nichtsingulär ist.

Beweis. 1. Folgt direkt mit Umformen der linearen Gleichungen $Ax + By = f$ und $Cx + Dy = g$.

2. Folgt mit [22, Theorem 1.1] nachdem $\det(M/A) = \det(M)/\det(A)$.

□

7.5 Satz

Das inexakte Interpolationsproblem, definiert in 6.1, hat eine eindeutige Lösung.

Beweis. Im Gegensatz zu dem exakten Interpolationsproblem, können wir die Bedingung der Gradienten

$$\varphi_i \neq 0, \forall i \in I$$

abschwächen. Angenommen es existiert ein $T \in I$ mit $\varphi_T = 0$. Sei dies o.B.d.A der letzte Punkt der Indexmenge, also $T = |I|$. Bezeichne dann mit $U^T(v)$ die Energie von v einschließlich aller Punkte bis T , also

$$U^T(v) = |v|_m^2 + \lambda \sum_{i=1}^T (v(a_i) \bullet \varphi_i - \alpha_i)^2.$$

Dann ist aber

$$U^T(v) = U^{T-1}(v) + \lambda \alpha_T^2,$$

und somit auch

$$\inf_{v \in X^n} U^T(v) = \inf_{v \in X^n} U^{T-1}(v) + \lambda \alpha_T^2.$$

Ein Minimierer von $U^{T-1}(v)$ ist somit auch ein Minimierer von $U^T(v)$. Ist $\tilde{i} < |I|$ die Anzahl der Punkte, an denen der Gradient verschwindet, so ist v dann ein (eindeutiger) Minimierer von $U^T(v)$, wenn v ein Minimierer von $U^{T-\tilde{i}}$ ist. Sei also $I \setminus L$ die Indexmenge aller Punkte mit $\varphi_i \neq 0$ für $i \in I \setminus L$. Sofern eine \mathcal{P}_{m-1}^n -unisolvente Untermenge $B = (a_j)_{j \in J}$ mit $N = n \cdot M$ Punkten existiert, dann können wir das Minimierungsproblem nach Satz 7.2 schreiben als lineares Gleichungssystem

$$K_\lambda \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \mathcal{S} + 1/\lambda Id & P \\ P^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}. \quad (7.9)$$

Wir zeigen nun mit Hilfe des Schur-Komplements, dass die Systemmatrix K_λ eine Inverse besitzt. Da \mathcal{S} eine symmetrische, reelle Matrix ist, können wir diese Diagonalisieren, d.h. es existiert eine unitäre Matrix U und eine Diagonalmatrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{T-\tilde{i}})$ mit

$$\mathcal{S} = U^t \Lambda U,$$

wobei die Diagonaleinträge von Λ die Eigenwerte von \mathcal{S} sind. Dann ist aber

$$\mathcal{S} + 1/\lambda Id = U^t \Lambda U + 1/\lambda U^t U = U^t (\Lambda + 1/\lambda Id) U.$$

Somit ist $\mathcal{S} + 1/\lambda Id$ invertierbar genau dann, wenn $U^t (\Lambda + 1/\lambda Id) U$, bzw.

$$(\Lambda + 1/\lambda Id) = \text{diag}\left(\frac{\lambda_1 \lambda + 1}{\lambda}, \dots, \frac{\lambda_{T-\tilde{i}} \lambda + 1}{\lambda}\right)$$

invertierbar ist, da $U^t = U^{-1}$ gilt. Sofern die Eigenwerte $\lambda_i \neq -1/\lambda$ für alle i in $I \setminus L$ sind, ist $(\Lambda + 1/\lambda Id)$ invertierbar und somit durch Lemma 7.4 auch die Systemmatrix K_λ .

Ist $\lambda_i = -1/\lambda$ für eine Untermenge von $I \setminus L$, so hat die Systemmatrix nach dem Diagonalisieren die Form

$$\left[\begin{array}{cc|c} \tilde{\Lambda} & 0 & P_1 \\ 0 & 0 & P_2 \\ \hline P_1^t & P_2^t & 0 \end{array} \right] \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}.$$

Definieren wir nun die Matrizen $\tilde{P} := [0, P_1]$ und $A := [0, P_2; P_2^t, 0]$ erhalten wir das neue Gleichungssystem

$$\begin{bmatrix} \tilde{\Lambda} & \tilde{P} \\ \tilde{P}^t & A \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}.$$

Nun ist $\tilde{\Lambda}$ invertierbar und mit Lemma 7.4 folgt die Invertierbarkeit der Systemmatrix K_λ .

Also existiert eine eindeutige Lösung für das inexacte Interpolationsproblem. □

7.6 Bemerkung

Im Gegensatz zu dem exakten Interpolationsproblem kann hier eine Lösung auch dann gefunden werden, wenn einige der Punkte einen verschwindenden Gradienten haben. Auf die Bedingung, dass für die Menge der Punkte A eine \mathcal{P}_{m-1}^n -unisolvante Untermenge $B = (a_j)_{j \in J}$ mit $N = n \cdot M$ Punkten existiert, kann jedoch nicht verzichtet werden. Somit müssen beispielsweise für $m = n = 2$ mindestens $N = 6$ Punkte mit nichtverschwindenden Gradienten existieren.

7.7 Beispiel

Für $m = n = 2$ bleibt die Form sehr ähnlich zu Beispiel 7.1. Lediglich entlang der Diagonalen verändert sich die Matrix \mathcal{S} zu $\mathcal{S} + 1/\lambda Id =$

$$\begin{bmatrix} 1/\lambda & E(a_1 - a_2)\varphi_1 \bullet \varphi_2 & E(a_1 - a_3)\varphi_1 \bullet \varphi_3 & \cdots & E(a_1 - a_T)\varphi_1 \bullet \varphi_T \\ E(a_2 - a_1)\varphi_2 \bullet \varphi_1 & 1/\lambda & E(a_2 - a_3)\varphi_2 \bullet \varphi_3 & \cdots & E(a_2 - a_T)\varphi_2 \bullet \varphi_T \\ E(a_3 - a_1)\varphi_3 \bullet \varphi_1 & E(a_3 - a_2)\varphi_3 \bullet \varphi_2 & 1/\lambda & \cdots & E(a_3 - a_T)\varphi_3 \bullet \varphi_T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E(a_T - a_1)\varphi_T \bullet \varphi_1 & E(a_T - a_2)\varphi_T \bullet \varphi_2 & E(a_T - a_3)\varphi_T \bullet \varphi_3 & \cdots & 1/\lambda \end{bmatrix} \in \mathbb{R}^{T \times T}$$

Die Matrix P hier kann direkt aus Beispiel 7.1 übernommen werden.

8 Analytische Konditionsuntersuchungen

Anhand der Matrixformen aus Kapitel 7 können wir einige Konditionsüberlegungen der Gleichungssysteme für das exakte bzw. das inexacte Interpolationsproblem anstellen.

8.1 Kondition

Das Fehlerverhalten der Lösungen bei Störung der Eingangsdaten können wir durch die Kondition beschreiben.

8.1 Definition (NACH [20])

Seien X, Y normierte Vektorräume und $f : X \rightarrow Y$ eine Funktion. Sei Δx eine Störung von $x \in X$. Bezeichne mit $\Delta f = \|f(x) - f(x + \Delta x)\|$ die Störung von f .

Die *absolute Konditionszahl* κ_{abs} eines Problems f an der Stelle x ist dann definiert durch

$$\kappa_{abs} := \kappa_{abs}(x) = \lim_{\delta \rightarrow 0} \sup_{\|\Delta x\| \leq \delta} \frac{\|\Delta f\|}{\|\Delta x\|}. \quad (8.1)$$

Die *relative Konditionszahl* κ_{rel} eines Problems f an der Stelle x ist definiert durch

$$\kappa_{rel} := \kappa_{rel}(x) = \lim_{\delta \rightarrow 0} \sup_{\|\Delta x\| \leq \delta} \left(\frac{\|\Delta f\|}{\|f(x)\|} / \frac{\|\Delta x\|}{\|x\|} \right). \quad (8.2)$$

Im Folgenden wollen wir den Grenzwert des Supremums in (8.1) und (8.2) als Supremum aller infinitesimalen Störungen Δx betrachten, somit wollen wir uns mit der folgenden Schreibweise begnügen

$$\begin{aligned} \kappa_{abs} &= \sup_{\Delta x} \frac{\|\Delta f\|}{\|\Delta x\|} \\ \kappa_{rel} &= \sup_{\Delta x} \left(\frac{\|\Delta f\|}{\|f(x)\|} / \frac{\|\Delta x\|}{\|x\|} \right) \end{aligned}$$

Ist f eine differenzierbare Funktion, so ist offensichtlich $\kappa_{abs} = \|Df(x)\|$ und $\kappa_{rel} = \|Df(x)\| \frac{\|x\|}{\|f(x)\|}$.

Für gewisse Parameter des Problems können wir die Kondition analytisch berechnen. Wir beginnen mit einer Störung der rechten Seite des Gleichungssystems, also der Werte $\alpha := u_t$ der Zeitableitung.

8.2 Störung der bezüglich der Zeitableitung

Sei K die Systemmatrix des exakten Interpolationsproblems, also

$$K := \begin{bmatrix} \mathcal{S} & P \\ P^t & 0 \end{bmatrix}.$$

Bezeichne mit $f : \mathbb{R}^T \rightarrow \mathbb{R}^{T+N}$ die Funktion zur Lösung des Gleichungssystems des exakten Interpolationsproblems mit

$$\alpha \mapsto \begin{bmatrix} \mathcal{S} & P \\ P^t & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \quad (8.3)$$

Wir möchten gerne die absolute Kondition, also die Anfälligkeit des Systems für Fehler in Daten α von f bestimmen.

Für eine Störung Δb der rechten Seite eines nichtsingulären linearen Gleichungssystems $Ax = b$ ist

die absolute Kondition genau die Matrixnorm der inversen Matrix A^{-1} , denn dann ist:

$$\begin{aligned}\kappa_{abs} &= \sup_{\Delta b} \frac{\|\Delta f\|}{\|\Delta b\|} \\ &= \sup_{\Delta b} \frac{\|A^{-1}b - A^{-1}(b + \Delta b)\|}{\|\Delta b\|} \\ &= \sup_{\Delta b} \frac{\|A^{-1}\Delta b\|}{\|\Delta b\|} = \|A^{-1}\|,\end{aligned}$$

und für die relative Kondition

$$\begin{aligned}\kappa_{rel} &= \sup_{\Delta b} \left(\frac{\|\Delta f\|}{\|f(b)\|} / \frac{\|\Delta b\|}{\|b\|} \right) \\ &= \sup_{\Delta b} \left(\frac{\|A^{-1}b - A^{-1}(b + \Delta b)\|}{\|A^{-1}b\|} / \frac{\|\Delta b\|}{\|b\|} \right) = \|A^{-1}\| \frac{\|b\|}{\|A^{-1}b\|}.\end{aligned}\quad (8.4)$$

Wir wissen jedoch, dass Teile der rechten Seite des Gleichungssystems nicht von Störungen in den Daten betroffen sind (die N Zeilen, die per Definition 0 sind). Die Norm der gesamten Matrix K beschreibt also nicht die Kondition unseres Problems. Wir müssen also eine leicht veränderte Matrix betrachten.

8.2 Satz

Die absolute Kondition des exakten Interpolationsproblems für die Störung der Parameter α bezüglich der euklidischen Norm ist gegeben durch den betraglich größten Singulärwert σ_1 der Matrix $K^{-1}\mathcal{T}$. Dabei ist \mathcal{T} die Projektion eines \mathbb{R}^T Vektors auf $\mathbb{R}^{(T+N)}$.

Beweis. Sei $\mathcal{T} \in \mathbb{R}^{(T+N) \times T}$ die Projektion eines Vektors $\alpha \in \mathbb{R}^T$ auf $[\alpha; 0] \in \mathbb{R}^{(T+N)}$, also

$$\mathcal{T} = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}\quad (8.5)$$

Für $\alpha \in \mathbb{R}^T$ bezeichne mit $\alpha_0 := (\alpha, 0, \dots, 0)^t \in \mathbb{R}^{(T+N)}$.

Dann ist $K^{-1}\alpha_0 = K^{-1}\mathcal{T}\alpha$. Wir können die Störung des Gleichungssystem nun in Abhängigkeit von α bezüglich einer Matrix schreiben.

Für die absolute Kondition des Problems f von (8.3) erhalten wir also

$$\begin{aligned}
\kappa_{abs} &= \sup_{\Delta\alpha} \frac{\|\Delta f\|}{\|\Delta\alpha\|} \\
&= \sup_{\Delta\alpha} \frac{\|K^{-1}\alpha_0 - K^{-1}(\alpha_0 + \Delta\alpha_0)\|}{\|\Delta\alpha\|} \\
&= \sup_{\Delta\alpha} \frac{\|K^{-1}\Delta\alpha_0\|}{\|\Delta\alpha\|} \\
&= \sup_{\Delta\alpha} \frac{\|K^{-1}\mathcal{T}\Delta\alpha\|}{\|\Delta\alpha\|} = \|K^{-1}\mathcal{T}\|
\end{aligned} \tag{8.6}$$

Sei nun $U\Sigma V^t = K^{-1}\mathcal{T}$ die Singulärwertzerlegung von $K^{-1}\mathcal{T}$. Es sind also $U \in \mathbb{R}^{(N+T) \times (N+T)}$, $V \in \mathbb{R}^{T \times T}$ unitäre Matrizen und $\Sigma \in \mathbb{R}^{(T+N) \times T}$ eine Diagonalmatrix mit Einträgen $\sigma_1, \dots, \sigma_T$, den Singulärwerten von $K^{-1}\mathcal{T}$. Dann ist für die euklidische Norm $\|\cdot\|_2$

$$\begin{aligned}
\kappa_{abs}^2 &\stackrel{(8.6)}{=} \|K^{-1}\mathcal{T}\|_2^2 = \sup_{\|\alpha\|_2 \neq 0} \frac{\|K^{-1}\mathcal{T}\alpha\|_2}{\|\alpha\|_2} \\
&= \sup_{\|\alpha\|_2=1} (K^{-1}\mathcal{T}\alpha, K^{-1}\mathcal{T}\alpha)_2 \\
&= \sup_{\|\alpha\|_2=1} (U\Sigma V^t\alpha, U\Sigma V^t\alpha)_2 \\
&= \sup_{\|\alpha\|_2=1} ((U\Sigma V^t)^t U\Sigma V^t\alpha, \alpha)_2 \\
&= \sup_{\|\alpha\|_2=1} (V\Sigma^t U^t U\Sigma V^t\alpha, \alpha)_2 \\
&= \sup_{\|\alpha\|_2=1} (V\Sigma^2 V^t\alpha, \alpha)_2 \quad , \text{da } U^t U = Id \\
&= \sup_{\|\alpha\|_2=1} (\Sigma V^t\alpha, \Sigma V^t\alpha)_2 = \sup_{\|\alpha\|_2=1} \|\Sigma V^t\alpha\|_2^2 = \sup_{\|\alpha\|_2=1} \|\Sigma\alpha\|_2^2 = \sigma_1^2,
\end{aligned}$$

denn für unitäre Matrizen U gilt bezüglich der euklidischen Norm $\|Ux\|_2 = \|x\|_2$ für alle x . Außerdem ist σ_1 der betragsmäßig größte Singulärwert von $K^{-1}\mathcal{T}$. Somit ist $\kappa_{abs} = \sigma_1$, was zu zeigen war. \square

Für die relative Kondition müssen benötigen wir erneut das Schur-Komplement aus Definition 7.3. Mit Hilfe von Lemma 7.4 können wir die Lösung des linearen Gleichungssystems vom exakten Interpolationsproblem

$$\begin{bmatrix} S & P \\ P^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \tag{8.7}$$

schreiben als

$$\begin{aligned}
\beta &= (K/S)^{-1}(0 - P^t S^{-1}\alpha), & \gamma &= S^{-1}(\alpha - P\beta) \\
\Leftrightarrow \beta &= (P^t S^{-1}P)^{-1} P^t S^{-1}\alpha, & \gamma &= S^{-1}(\alpha - P\beta),
\end{aligned}$$

da auch S und K nichtsingulär und somit K/S nichtsingulär ist. Einsetzen der ersten Gleichung

in die zweite liefert

$$\gamma = (S^{-1} - S^{-1}P(P^tS^{-1}P)^{-1}P^tS^{-1})\alpha. \quad (8.8)$$

Bezeichne nun mit $A_1 := (S^{-1} - S^{-1}P(P^tS^{-1}P)^{-1}P^tS^{-1})$ und $A_2 := (P^tS^{-1}P)^{-1}P^tS^{-1}$ die neuen Systemmatrizen. Dann wird durch $f : \alpha \mapsto (A_1; A_2)\alpha$ die Lösung des exakten Interpolationsproblems beschrieben. Mit Hilfe dieser Charakterisierung können wir also die relative Kondition bestimmen.

8.3 Satz

Die relative Kondition des exakten Interpolationsproblems für die Störung der Parameter α bezüglich der euklidischen Norm ist gegeben durch

$$\kappa_{rel} = \sigma_1 \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2} \leq \sigma_1/\sigma_T.$$

Es herrscht Gleichheit, falls α ein rechter Singulärwert von $(A_1; A_2)$ zu dem entsprechenden minimalen Singulärwert σ_T ist.

Beweis. Nach Definition 8.2 ergibt sich

$$\begin{aligned} \kappa_{rel} &= \sup_{\Delta\alpha} \frac{\|\Delta(A_2; A_2)\|_2}{\|\Delta\alpha\|_2} \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2} \\ &= \sup_{\Delta\alpha} \frac{\|(A_2; A_2)(\Delta\alpha)\|_2}{\|\Delta\alpha\|_2} \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2} \\ &= \sigma_1 \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2}, \end{aligned}$$

genau so wie oben mit σ_1 der größte Singulärwert von $(A_1; A_2)$. Dann ist weiter

$$\begin{aligned} &\leq \sup_{\|\alpha\|>0} \sigma_1 \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2} = \sup_{\|\alpha\|=1} \frac{1}{\|\alpha\|_2} \sigma_1 \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2} \\ &= \sigma_1 \frac{1}{\inf_{\|\alpha\|=1} \|(A_1; A_2)\alpha\|_2} \\ &= \sigma_1/\sigma_T, \end{aligned}$$

mit σ_T der kleinste Singulärwert von $(A_1; A_2)$. □

8.4 Bemerkung

Die gesamten Ergebnisse lassen sich direkt auf das inexacte Interpolationsproblem mit der Systemmatrix

$$K_\lambda := \begin{bmatrix} \mathcal{S} + 1/\lambda Id & P \\ P^t & 0 \end{bmatrix}$$

übertragen.

Es folgt direkt, dass

$$\kappa_{abs} = \tilde{\sigma}_1,$$

für $\tilde{\sigma}_1$ der betraglich größte Singulärwert von $K_\lambda^{-1}\mathcal{T}$ und

$$\kappa_{rel} = \sigma_1 \frac{\|\alpha\|_2}{\|(\tilde{A}_1; \tilde{A}_2)\alpha\|_2} \leq \tilde{\sigma}_1/\tilde{\sigma}_T$$

für $\tilde{\sigma}_1$ bzw. $\tilde{\sigma}_T$ der betraglich größte bzw. kleinste Singulärwert von $(\tilde{A}_1; \tilde{A}_2)$.

Die angepassten Formen von A_1, A_2 folgen analog zu (8.8) mit

$$\tilde{A}_1 := ((S + 1/\lambda Id)^{-1} - (S + 1/\lambda Id)^{-1}P(P^t(S + 1/\lambda Id)^{-1}P)^{-1}P^t(S + 1/\lambda Id)^{-1})$$

und

$$\tilde{A}_2 := (P^t(S + 1/\lambda Id)^{-1}P)^{-1}P^t(S + 1/\lambda Id)^{-1}.$$

8.3 Fehler der gesamten Funktion

Wir betrachten erneut die Störung der rechten Seite von

$$\begin{bmatrix} S & P \\ P^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}. \quad (8.9)$$

Uns interessiert nun die Störung der Funktion

$$w(y) = (-1)^m \sum_{i \in I} \gamma_i E(a_i - y) \varphi_i + q(y),$$

bezüglich der Norm $|\cdot|_m$ auf dem Funktionenraum X^n . Nach der Definition der absoluten Kondition benötigen wir $\Delta w = w_\alpha - w_{\alpha + \Delta\alpha}$. Sei γ_α die Lösung bezüglich α . Durch Ersetzen der γ_{α_i} durch die Lösung des Schur-Komplements nach (8.8) erhalten wir

$$\begin{aligned} w_\alpha - w_{\alpha + \Delta\alpha} &= (-1)^m \sum_{i \in I} \gamma_{\alpha_i} E(a_i - \cdot) \varphi_i + q_\alpha(\cdot) - (-1)^m \sum_{i \in I} \gamma_{(\alpha + \Delta\alpha)_i} E(a_i - \cdot) \varphi_i + q_{\alpha + \Delta\alpha}(\cdot) \\ &= (-1)^m \sum_{i \in I} ((A_1 \alpha)_i - (A_1(\alpha + \Delta\alpha))_i) E(a_i - \cdot) \varphi_i + q_\alpha(\cdot) - q_{\alpha + \Delta\alpha}(\cdot) \\ &= w_{\Delta\alpha} + q_{\Delta\tilde{\alpha}}, \end{aligned}$$

für den entsprechenden Polynomanteil $q_{\Delta\tilde{\alpha}}$.

Einsetzen der Definition der absoluten Kondition liefert

$$\begin{aligned} \kappa_{abs} &= \sup_{\|\Delta\alpha\|=1} \|\Delta w\|_m^2 \\ &= \sup_{\|\Delta\alpha\|=1} \|w_{\Delta\alpha}\|_m^2 \\ &\stackrel{(6.1)}{=} \sup_{\Delta\alpha} \sum_{i \in I} \sum_{j \in J} \gamma_{\Delta\alpha_i} \gamma_{\Delta\alpha_j} E(a_j - a_i) (\varphi_i \bullet \varphi_j) \\ &= \sup_{\Delta\alpha} \sum_{i \in I} \sum_{j \in J} (A_1 \Delta\alpha)_i (A_1 \Delta\alpha)_j E(a_j - a_i) (\varphi_i \bullet \varphi_j). \end{aligned}$$

Nutzung der Matrixschreibweise für S ergibt

$$\begin{aligned} &= \sup_{\Delta\alpha} (A_1 \Delta\alpha)^t S (A_1 \Delta\alpha) \\ &= \sup_{\Delta\alpha} \Delta\alpha^t A_1^t S (A_1 \Delta\alpha), \end{aligned}$$

und nach Einsetzen der Definition der Matrix A_1 mit elementaren Umformungen

$$\begin{aligned} &= \sup_{\Delta\alpha} \Delta\alpha^t (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1})^t S (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1}) \Delta\alpha \\ &= \sup_{\Delta\alpha} \Delta\alpha^t (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1}) (Id - P (P^t S^{-1} P)^{-1} P^t S^{-1}) \Delta\alpha \\ &= \sup_{\Delta\alpha} \Delta\alpha^t (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1} \\ &\quad + S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1}) \Delta\alpha \\ &= \sup_{\Delta\alpha} \Delta\alpha^t (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1}) \Delta\alpha \\ &= \sup_{\Delta\alpha} \Delta\alpha^t A_1 \Delta\alpha \\ &= \|A_1\|_2. \end{aligned}$$

Dabei hatten wir A_1 definiert mit $A_1 := (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1})$.

Die Gültigkeit der letzten Gleichung zeigt sich durch die Überlegung, dass A_1 symmetrisch ist, denn

$$\begin{aligned} A_1^t &= (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1})^t \\ &= (S^{-1t} - S^{-1t} P^t (P^t S^{-1} P)^{-1t} P^t S^{-1t}) \\ &= (S^{-1} - S^{-1} P (P^t S^{-1} P)^{-1} P^t S^{-1}) = A_1, \end{aligned}$$

da S symmetrisch ist. Außerdem gilt für eine symmetrische Matrix A mit Singulärwertzerlegung $A = U \Sigma U^t$ mit U einer orthogonalen Matrix

$$\sup_{\|x\|_2=1} |x^t A x| = \sup_{\|x\|_2=1} |x^t U \Sigma U^t x| = \sup_{\|y\|_2=1} |y^t \Sigma y| = |\sigma_1|, \text{ für } y = U^t x,$$

wobei aufgrund der Orthogonalität von U gilt $\|Ux\|_2 = \|x\|_2$. Andererseits gilt, wie schon oben gesehen,

$$\|A\|_2^2 = \sup_{\|x\|_2=1} \|Ax\|_2^2 = \sigma_1^2.$$

Wir stellen also fest, dass die absolute Kondition bezüglich der gesamten Funktion w in der Norm $|\cdot|_m$ ausschließlich von den Einträgen der Matrix A_1 abhängt. Dies ist nicht ganz unerwartet, da die Matrix A_2 zur Bestimmung der Koeffizienten des Polynoms vom Grad $\leq m-1$ genutzt werden, diese auf $|\cdot|_m$ jedoch verschwinden.

8.4 Systemmatrix ist singulär

Bisher haben wir an der Bedingung $\varphi \neq 0$ festgehalten. Dies schließt natürlich den Fall $\varphi_{\tilde{i}} = 0$ und $\alpha_{\tilde{i}} \neq 0$ für ein $\tilde{i} \in I$ aus, denn dann existiert kein v , welches den Optical Flow Constraint erfüllt. Auch anhand der Matrix

$$A := \begin{bmatrix} \mathcal{S} & P \\ P^t & 0 \end{bmatrix}$$

wird deutlich, dass dies zu einer singulären Matrix und so zu einem nicht lösbaarem Gleichungssystem führt, denn dort verschwindet eine gesamte Zeile wobei die rechte Seite nicht verschwindet.

$$\begin{aligned} (A)_{\tilde{i}} &= (-1)^m \sum_{j \in I} \gamma_j E(a_j - a_{\tilde{i}}) \varphi_j \bullet \varphi_{\tilde{i}} + q(a_{\tilde{i}}) \bullet \varphi_{\tilde{i}} = \alpha_{\tilde{i}} \\ &\Rightarrow 0 = \alpha_{\tilde{i}} \end{aligned}$$

In diesem Fall erfüllt entweder das zu Grunde liegende Bild nicht die *Optical Flow Constraints* und somit unter Umständen nicht die *Brightness Constancy Assumption*, oder ein Messfehler der Daten, der zu einem verschwindenden Gradienten führt, stört die Lösung des Problems.

Beim inexakten Problem sind entlang des ersten Blocks der Blockmatrix

$$A_\lambda = \begin{bmatrix} \mathcal{S} + 1/\lambda Id & P \\ P^t & 0 \end{bmatrix}$$

durch die Regularisierung nicht-null Einträge entstanden. Für einen verschwindenden Gradienten $\varphi_{\tilde{i}} = 0$ kann somit trotzdem eine Lösung der Matrix berechnet werden.

$$\begin{aligned} (A_\lambda)_{\tilde{i}} &= \sum_{j \in J} \gamma_j E(a_j - a_{\tilde{i}}) \varphi_j \bullet \varphi_{\tilde{i}} + \frac{1}{\lambda} \gamma_{\tilde{i}} + \sum_{|e_j| \leq m-1} \beta_{e_j} a_{\tilde{i}}^{e_j} \bullet \varphi_{\tilde{i}} = \alpha_{\tilde{i}} \\ &\frac{1}{\lambda} \gamma_{\tilde{i}} = \alpha_{\tilde{i}}, \end{aligned}$$

Mit Hilfe des inexakten Verfahrens können also problematische Berechnungspunkte überwunden werden. Das Verfahren ist hier stabiler gegenüber Störungen des Gradienten. Wie wir später sehen werden (Kap. 10.3.2) sind auch sehr kleine Gradienten problematisch, was für den inexakten Interpolationsansatz spricht.

9 Algorithmus und Fehlermaße

Im Folgenden wollen wir auf die Umsetzung der Theorie zu einem Algorithmus vorbereiten. Wir gehen auf die Diskretisierung, den Pseudocode der beiden Verfahren und schließlich auf die Evaluation der numerischen Ergebnisse ein.

9.1 Diskretisierung

Mit Blick auf den *Optical Flow Constraint*

$$v \bullet \nabla u + u_t = 0$$

aus Kapitel 2 wird deutlich, dass wir zwei Operatoren für gegebene Bilddaten u diskretisieren müssen. Wir wählen für u_t den Vorwärtsdifferenzenquotient und für die räumlichen Ableitungen ∇u den zentralen Differenzenquotient. Die Daten, die in einem Punkt zu Verfügung stehen, werden also maximal vom direkten Nachbarn erschlossen. Des weiteren beschränken wir uns der Einfachheit halber auf eine Anzahl von zwei Bildern einer Bildfolge.

Für eine diskrete Version von u , bezeichne mit $U(i, j, t) := u(i, j)$ die Intensität eines Pixels zum Zeitpunkt t für $i = 0, \dots, n_x$, $j = 0, \dots, n_y$, und $t = 0, 1$, definiere somit

$$U_x(i, j, t) = \begin{cases} \frac{U(i+1, j, t) - U(i-1, j, t)}{2\delta_x} & \text{falls } i > 0 \text{ und } i < n_x \\ 0 & \text{sonst,} \end{cases}$$

$$U_y(i, j, t) = \begin{cases} \frac{U(i, j+1, t) - U(i, j-1, t)}{2\delta_y} & \text{falls } j > 0 \text{ und } j < n_y \\ 0 & \text{sonst,} \end{cases}$$

$$U_t(i, j) = \frac{U(i, j, t+1) - U(i, j, t)}{\delta_t},$$

für Schrittweiten δ_x, δ_y und δ_t .

9.2 Algorithmus

Die Implementierung des exakten Verfahrens benötigt nur wenige Zeilen Code. Im ersten Schritt werden die Raum-, und Zeitgradienten in den betrachteten Punkten berechnet. Anschließend müssen die Beträge r_{ij} zwischen den Punkten $(a_i)_{i \in I}$ und das innere Produkt dot_{ij} der Raumgradienten bestimmt werden. Aufgrund der Symmetrieeigenschaften reicht einmaliges Berechnen und Setzen von $r_{ij} = r_{ji}$ bzw. $dot_{ij} = dot_{ji}$.

Es folgt die Auswertung am *reproducing kernel* und Konstruktion der Matrizen \mathcal{S}, P und schließlich A . Im letzten Schritt wird die Inverse der Matrix A berechnet und wir erhalten die gesuchten Koeffizienten γ und β , welche zurück gegeben werden.

Algorithm 1 Exakte Interpolation

```

1: procedure EXAKTINTERPOLATION( $U_0, U_1, A$ )
2:   for  $a_i \in A = (a_i)_{i \in I}$  do
3:      $U_x(a_{i1}, a_{i2}, 0), U_y(a_{i1}, a_{i2}, 0), U_t(a_{i1}, a_{i2}) \leftarrow \text{calcDiff}(U_0, U_1, a_i)$ 
4:      $\varphi_i := [U_x, U_y](a_{i1}, a_{i2}, 0)$ 
5:      $\alpha_i := U_t(a_{i1}, a_{i2}, 0)$ 
6:   end for
7:   for  $i, j \in I$  do
8:      $r_{ij} \leftarrow |a_i - a_j|$ 
9:      $\text{dot}_{ij} \leftarrow \varphi_i \bullet \varphi_j$ 
10:    if  $r_{ij} > 0$  then
11:       $s_{ij} \leftarrow \frac{1}{8\pi} r_{ij}^2 \log(r_{ij}) \cdot \text{dot}_{ij}$ 
12:    else
13:       $s_{ij} \leftarrow 0$ 
14:    end if
15:  end for
16:   $\mathcal{S} := (s_{ij})_{i,j \in I}$ 
17:   $p_i := [\varphi_{i1}, a_{i1}\varphi_{i1}, a_{i2}\varphi_{i1}, \varphi_{i2}, a_{i1}\varphi_{i2}, a_{i2}\varphi_{i2}]$ 
18:   $P := (p_i)_{i \in I}$ 
19:   $A := [\mathcal{S}, P; P^t, 0]$ 
20:   $\alpha_0 := [\alpha, 0]$ 
21:   $[\gamma, \beta] \leftarrow A^{-1}\alpha_0$  return  $\gamma, \beta$ 
22: end procedure

```

Der Algorithmus des inexakten Verfahrens unterscheidet sich nur geringfügig von dem Exakten. Hier wird lediglich zusätzlich der Regularisierungsparameter λ übergeben. Im Konstruktionsschritt der Matrix A wird dieser dann mit der Identitätsmatrix multipliziert und auf \mathcal{S} addiert. Das restliche Verfahren verläuft analog.

Algorithm 2 Inexakte Interpolation

```

1: procedure INEXAKTEINTERPOLATION( $U_0, U_1, A, \lambda$ )
2:   for  $a_i \in A = (a_i)_{i \in I}$  do
3:      $U_x(a_{i1}, a_{i2}, 0), U_y(a_{i1}, a_{i2}, 0), U_t(a_{i1}, a_{i2}) \leftarrow \text{calcDiff}(U_0, U_1, a_i)$ 
4:      $\varphi_i := [U_x, U_y](a_{i1}, a_{i2}, 0)$ 
5:      $\alpha_i := U_t(a_{i1}, a_{i2}, 0)$ 
6:   end for
7:   for  $i, j \in I$  do
8:      $r_{ij} \leftarrow |a_i - a_j|$ 
9:      $\text{dot}_{ij} \leftarrow \varphi_i \bullet \varphi_j$ 
10:    if  $r_{ij} > 0$  then
11:       $s_{ij} \leftarrow \frac{1}{8\pi} r_{ij}^2 \log(r_{ij}) \cdot \text{dot}_{ij}$ 
12:    else
13:       $s_{ij} \leftarrow 0$ 
14:    end if
15:  end for
16:   $\mathcal{S} := (s_{ij})_{i,j \in I}$ 
17:   $p_i := [\varphi_{i1}, a_{i1}\varphi_{i1}, a_{i2}\varphi_{i1}, \varphi_{i2}, a_{i1}\varphi_{i2}, a_{i2}\varphi_{i2}]$ 
18:   $P := (p_i)_{i \in I}$ 
19:   $A := [\mathcal{S} + 1/\lambda Id, P; P^t, 0]$ 
20:   $\alpha_0 := [\alpha, 0]$ 
21:   $[\gamma, \beta] \leftarrow A^{-1}\alpha_0$  return  $\gamma, \beta$ 
22: end procedure

```

Die Auswertung der Funktion an einem Punkt z bei gegebenen Koeffizienten γ, β erfordert die Berechnung des Abstands r_i zu jedem der Ausgangspunkte a_i für den *reproducing kernel* und des Polynomsanteils p .

Algorithm 3 Funktionsauswertung

```

1: procedure FUNKTIONSAUSWERTUNG( $z, \gamma, \beta, \varphi, A$ )
2:   for  $a_i \in A$  do
3:      $r_i \leftarrow |z - a_i|$ 
4:     if  $r_i > 0$  then
5:        $w_i \leftarrow \frac{1}{8\pi} r_i^2 \log(r_i)$ 
6:     else
7:        $w_i \leftarrow 0$ 
8:     end if
9:   end for
10:   $p_x \leftarrow \beta[0] + \beta[1]z_1 + \beta[2]z_2$ 
11:   $p_y \leftarrow \beta[3] + \beta[4]z_1 + \beta[5]z_2$ 
12:   $v_x \leftarrow \sum_i \gamma_i w_i \varphi_{i1} + p_x$ 
13:   $v_y \leftarrow \sum_i \gamma_i w_i \varphi_{i2} + p_y$  return  $v = [v_x, v_y]$ 
14: end procedure

```

9.3 Fehlermaße

Die Evaluierung von *Optical Flow* Feldern ist nicht unproblematisch. Im Folgenden gehen wir auf diese Problematik ein und erläutern welche Fehlermaße zur Evaluierung der Ergebnisse genutzt wurde. Die Beschreibung der Problematik und die Wahl der Fehlermaße orientieren sich an [7, S.114 ff.]

9.3.1 Schlechtgestellttheit bei der Berechnung des Fehlers

Eine Möglichkeit ein Geschwindigkeitsfeld v zu bewerten ist der Vergleich mit einem vorhandenen *Ground Truth Flow*, also einem Geschwindigkeitsfeld, welches die Veränderung in einer Bildfolge hinreichend beschreibt. Es stellt sich heraus, dass verschiedene Geschwindigkeitsfelder die gleichen Veränderungen in einer Bildfolge beschreiben können. Betrachten wir hierzu die beiden Matrizen einer vereinfachten Abfolge von Bildern.

$$u_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Für diese Bildsequenz sind verschiedene Vektorfelder denkbar, die die Bewegung hinreichend beschreiben, z.B.

$$v_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

beschreibt die Verschiebung der beiden Punkte um einen Pixel nach unten, hingegen

$$v_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, v_2 = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

zusätzlich deren Position vertauscht.

Die Geschwindigkeitsvektoren

$$v_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, v_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

verschieben alle Pixel eine Position nach unten, während in x-Richtung keine Vertauschung geschieht.

Wie man unschwer erkennen kann, erwirken alle Kombinationen v_1, v_2 die selbe Bewegung im Bild u_1 und erfüllen somit auch den *Optical Flow Constraint* $\nabla u \bullet v + u_t = 0$. Es ist fragwürdig ein Paar von Geschwindigkeitsvektoren als grundlegende Wahrheit (*Ground Truth*) v_{GT} zu bezeichnen und andere Vektorfelder gegen dieses aufzuwiegen. Dies sollte bei der Evaluation im Hinterkopf behalten werden. Ein Teil der Evaluationsmethoden verlangt ein solches v_{GT} , andere Methoden kommen ohne dieses aus.

Die Auswahl der Fehlermaße wurde aufgrund [7] getroffen. AEE, AE und SSD sind klassische Fehlermaße, ebenfalls in [4] zu finden.

9.3.2 Angular Error (AE)

Das nach Baker u. a. [4] am häufigsten genutzte Fehlermaß ist der *Angular Error (AE)*. Bei diesem wird der durchschnittliche Winkel zwischen (normierten) Vektorfeld und (normierten) *Ground Truth* bestimmt. Dazu werden die Vektorfelder vom \mathbb{R}^2 in den \mathbb{R}^3 mit $v \mapsto (v_1, v_2, 1)$ projiziert, um die Division durch 0 auszuschließen. Es ergeben sich die projizierten und normierten Felder

$$\bar{v} := \frac{(v_1, v_2, 1)}{\sqrt{\|v\|^2 + 1}}, \bar{v}_{GT} := \frac{(v_{GT1}, v_{GT2}, 1)}{\sqrt{\|v_{GT}\|^2 + 1}}.$$

Anschließend wird der durchschnittliche Winkel zwischen den entsprechenden Punkten der beiden Vektorfelder berechnet.

$$AE := \frac{1}{|\Omega|} \sum_{i=1}^{|\Omega|} \arccos(\bar{v}(i) \bullet \bar{v}_{GT}(i))$$

Die Werte von dem Arkuskosinus liegen im Intervall $[0, \pi]$ und entsprechen dem Winkel der Vektoren im Radiant. Durchschnittswerte nahe an Null deuten somit auf eine gleichgerichtete Ausrichtung, Werte nahe an π hingegen auf eine entgegengesetzte Ausrichtung. Liegt der Durchschnitt bei $\pi/2 \approx 1.57$ ist mit keinem Zusammenhang zwischen v und v_{GT} zu rechnen.

Wie Dirks in [7, S.117] aufzeigt, ist der AE dahingehend problematisch, da für betraglich große Geschwindigkeitsvektoren der Fehler geringer ist als für betraglich Kleine. Da die Elemente eines Bildes, die viel Bewegung aufzeigen, häufig besonders interessant sind, werden diese unter Umständen zu gering gewichtet. Fehler mit wenig Bewegung, z.B. im Hintergrund werden hingegen übergewichtet.

9.3.3 Absolute Endpoint Error (AEE)

Der *Average Endpoint Error (AEE)* misst die absolute Entfernung der endgültigen Position eines Pixels nach seiner Verschiebung im Vergleich zu der tatsächlichen Position. Für $v = (v_1, v_2)$ ist der AEE definiert durch die Formel

$$AEE := \frac{1}{|\Omega|} \sum_{i=1}^{|\Omega|} \sqrt{(v_1(i) - v_{GT_1}(i))^2 + (v_2(i) - v_{GT_2}(i))^2},$$

für $v_{GT} = (v_{GT_1}, v_{GT_2})$ ein gegebener *Ground Truth* und $|\Omega|$ die Gesamtzahl der Pixel.

9.3.4 SSD Interpolation Error (IE)

Ein Fehlermaß, welches ohne einen gegebenen *Ground Truth* auskommt, ist der *SSD (Sum of Square Differences) Interpolation Error (IE)*. Hierzu wird zuerst ein zweites Bild \tilde{u}_2 mit Hilfe von bikubischer Interpolation berechnet. Mit dem ersten Bild der Sequenz u_1 und dem Vektorfeld v errechnen wir

$$\tilde{u}_2 := u_1(x + v).$$

Nun wird die Summe der Abweichungsquadrate zwischen dem interpolierten Bild \tilde{u}_2 und dem Originalbild u_2 bestimmt und dieses über die Größe des Bildes gemittelt. Wir erhalten

$$IE := \sqrt{\frac{1}{|\Omega|} \sum_{i=1}^{|\Omega|} (\tilde{u}_2(i) - u_2(i))^2}.$$

9.3.5 Normierter SSD Interpolation Error (NIE)

Der *normierter Interpolation Error (NIE)* ist eine leicht veränderte Version des IE, definiert durch:

$$NIE := \sqrt{\frac{1}{|\Omega|} \sum_{i=1}^{|\Omega|} \frac{(\tilde{u}_2(i) - u_2(i))^2}{\|\nabla u_2(x)\|^2 + 1}}.$$

Hierbei wird der Fehler zwischen dem interpolierten Bild \tilde{u}_2 und dem zweiten Bild u_2 durch die Norm des Gradienten geteilt. Daher wird der Fehler an den Kanten eines Bildes weniger stark gewichtet, denn dort ist die Norm des Gradienten $\|\nabla u_2(x)\|^2$ groß. Für kleine Gradienten verhält sich $\|\nabla u_2(x)\|^2 + 1$ wie 1 und wird somit nicht gewichtet. Begründet wird dies in [7] damit, dass die Approximation von Gradienten für *Optical Flow* Berechnungen verrauscht ist, daher diese gewichtet werden sollten.

Beide der IE und NIE messen die Unterschiede zwischen den Intensitäten der Bilder. Die Werte im Bild, die in Kapitel 10 betrachtet werden, liegen alle im Intervall $[0, 255]$.

10 Ergebnisse

Im folgenden Kapitel betrachten wir Lösungen der bisher betrachteten Algorithmen. Wir beschränken uns auf schwarz/weiß Sequenzen von zwei Bildern. Die Algorithmen wurden in Python implementiert und ausgewertet. Ein Ausschnitt des Programmcodes ist in Appendix A zu finden.

10.1 Grundlegende Bewegungsformen

Inspiziert von [7] wollen wir zunächst drei grundlegende Arten von Bewegung analysieren; Translation, Rotation und Skalierung. Dies wird damit begründet, dass alle anderen Arten von Bewegung in der Ebene lediglich eine Kombination aus diesen ist. Eine Untersuchung dieser basalen Bewegungen in Isolation zum besseren Verständnis ist somit gerechtfertigt.

Da die Darstellung von Vektorfeldern mit Hilfe von Pfeilen in der Ebene (siehe Abb. 7 (a)-(c)) im Detail ungenau ist, greifen wir auf die Visualisierung *Flow Field Color Coding* aus [4] zurück. In dieser Farbcodierung wird jedem (Bewegungs-)Vektor eine Farbe aus dem Farbraum aus Abb. 8 zugeordnet. Dabei ändert sich der Farbton mit Ausrichtung des Vektors, die Farbsättigung mit dessen Betrag. Die maximale Sättigung ist im Folgenden durch den betraglich größten Vektor bestimmt, außer ggf. im direkten Vergleich mit anderen Vektorfeldern durch das gemeinsame Maximum. Die konkrete Implementierung in Python der Visualisierung ist eine leichte Modifikation von [18].

Die codierten Bewegungsfelder der grundlegenden Bewegungsformen sind in Abb. 7 (d)-(f) dargestellt.

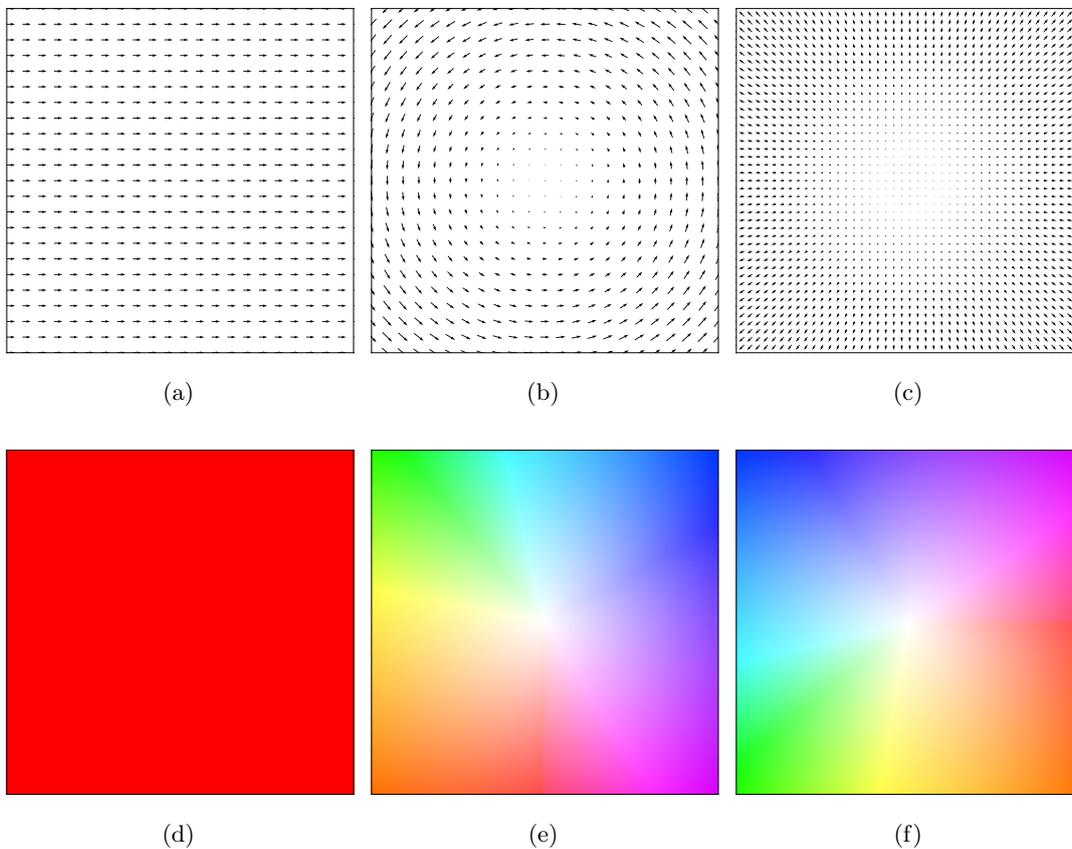


Abbildung 7: Grundarten der Bewegung in der Ebene. Von (a) – (c) die Darstellung als Vektorfeld mit Pfeilen und (d) – (f) die entsprechende Darstellung mit *Flow* Visualisierung nach [4].

10.2 Gaussglocke

Für erste Versuche wurden folgende künstliche Bilddaten konstruiert. Auf einem $n \times n$ großen Gebiet wurde der Graph einer 2D-Gaussglocke gezeichnet, um ein glattes Bild mit möglichst nichtverschwindenden Gradienten als erste Versuchsfläche zu erhalten. Betrachten wir also die kontinu-

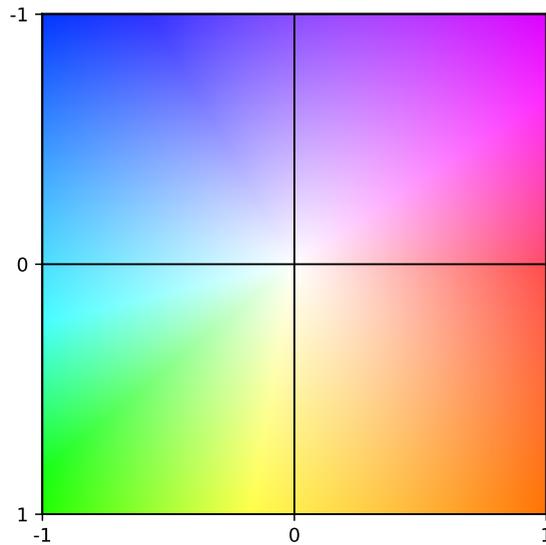


Abbildung 8: *Flow Field Color Coding* nach [4]. Ein Vektor ausgehend vom Zentrum entspricht der Farbe seiner Koordinaten.

ierliche Funktion

$$u : [-3, 3] \times [-3, 3] \rightarrow \mathbb{R}^2$$

$$(x, y) \mapsto \frac{1}{\sqrt{2\pi}} (e^{-\frac{x^2}{2}}, e^{-\frac{y^2}{2}}),$$

bzw. die Diskretisierung dieser mit

$$U(i, j) = u(-3 + i\Delta x, -3 + j\Delta y),$$

für $0 \leq i, j \leq n$ und $\Delta x, \Delta y = 6/n$ (siehe Abb. 9). Anschließend wird die Funktion noch auf das Intervall $[0, 255]$ normiert.

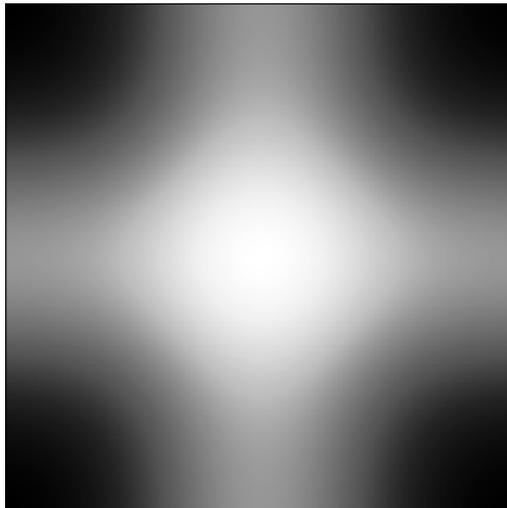


Abbildung 9: $2D$ -Gaussglocke auf einem 200×200 px Gebiet. In der diskreten Version dieser Größe sind sogar alle (räumlichen) Gradienten nichtverschwindend.

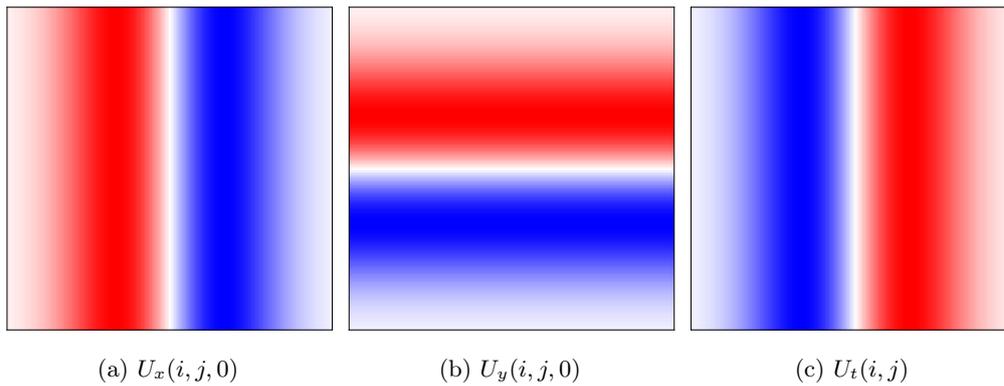


Abbildung 10: Gradienten in räumlichen Dimensionen x und y und Differential in Richtung t für die zweidimensionale Gaussglocke bei Translation nach rechts mit Diskretisierung nach Kapitel 9.1. Negative Werte sind blau, positive Werte in rot eingefärbt.

Das zweite Bild der Bildsequenz wurde durch Anwendung einer der drei grundlegenden Bewegungen des ersten Bildes erstellt. Da die Diskretisierung des Gradienten lediglich den direkten Nachbarn eines Punktes berücksichtigt, wurden die *Ground Truth* Bewegungsfelder der Grundarten der Bewegung, betraglich durch Eins beschränkt. Nach Konstruktion ist also $|v_{GT}| \leq 1$. Wir erhalten das zweite Bild der Bildfolge mit

$$U_2(x, y) := U_1(x + v_1, y + v_2).$$

Dazu wurden die Werte von U_1 mit einer bikubischen Interpolation berechnet. Natürlich ist es hier auch möglich die Werte von U_2 anhand der ursprünglichen kontinuierlichen Funktion u zu berechnen. Sollten jedoch später die grundlegenden Bewegungen auch auf andere Bilder angewendet werden, die nicht direkt zu berechnen sind, verfällt diese Möglichkeit. Es wurde daher ein einheitlicher Weg gewählt.

Der entsprechende *Ground Truth* ist beispielsweise bei der Translation nach rechts um einen Pixel, konstant 1 entlang der x-Achse und 0 entlang der y-Achse.

In Abbildung 11 sind die durch das exakte Interpolationsproblem berechnete Geschwindigkeitsfelder dargestellt. Es wurde für neun äquidistante Koordinaten (Anordnung siehe Abb. 11a) die räumlichen und zeitlichen Gradienten berechnet und anschließend die eindeutige Lösung v auf dem gesamten Gebiet ausgewertet.

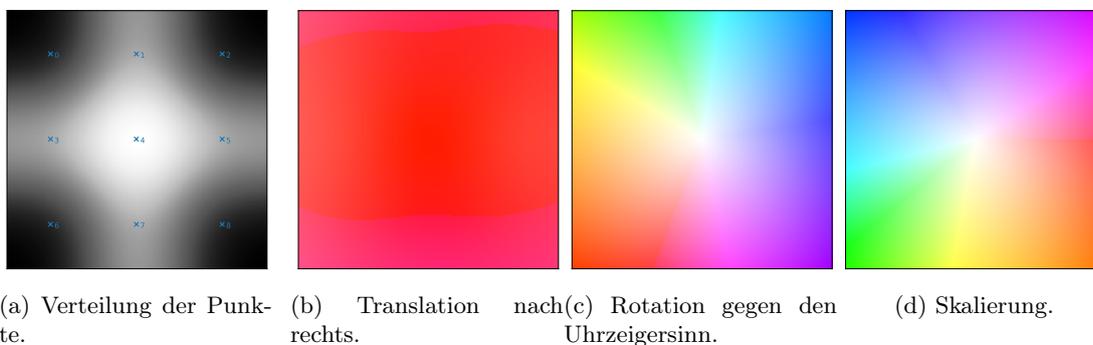


Abbildung 11: Errechnete Vektorfelder der drei Grundarten der Bewegung bei der Gaussglocke anhand von 9 Punkten.

Obwohl die Vektorfelder aus Abb. 11 den *Ground Truth* Feldern aus Abb. 7 ähneln, sind feine Unterschiede zu erkennen, so ist z.B. bei der Translation in (b) am unteren und oberen Bildrand eine leichte Pinkfärbung zu erkennen.

Für eine steigende Anzahl an Interpolationspunkten betrachten wir die Fehlerwerte im Graphen von Abb. 12. Auffällig ist, dass für eine steigende Anzahl an Punkten, die Qualität der Geschwindigkeitsfelder nicht zuverlässig zunimmt, d.h. der Fehler nicht stetig abnimmt. Stattdessen schwanken die Werte für verschiedene Fehlermaße mehr oder weniger chaotisch. Eine mögliche Erklärung ist, dass die komplette Bewegung für das gesamte glatte Bild schon vollständig durch die minimale Anzahl an Koordinaten beschrieben werden kann. Zur Erinnerung: die minimale Anzahl wird durch die Dimension der n -dimensionalen Polynome vom Grad $m - 1$ bestimmt, für $n = m = 2$ also 6. Das würde also bedeuten, dass die Bewegungen ausschließlich linearer Natur ist, die Schwankungen in den Fehlerwerten von anderen Faktoren bestimmt wird.

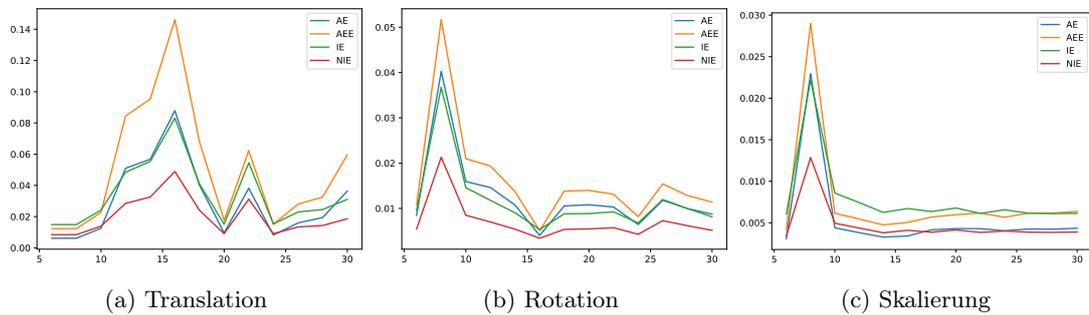


Abbildung 12: Fehlerverhalten für ansteigende Anzahl an Koordinaten der Gaussglocke für verschiedene Grundbewegungsarten.

Im folgenden Abschnitt wollen wir auf einige dieser Fehlerfaktoren eingehen und Erklärungen für Artefakte liefern, die zu Schwankungen im *Optical Flow* Feld führen können.

10.2.1 Monotone Gradientenausrichtung

Wie bereits im vorherigen Abschnitt angesprochen, sind in Abb. 11b am oberen und unteren Bildrand eine pinke Einfärbung zu erkennen. Anhand einer etwas anderen Anordnung von Koordinaten lässt sich dieses Artefakt ebenfalls erkennen und liefert mögliche Erklärungen. Für acht Koordinaten (Abb. 13a) wurde erneut ein Geschwindigkeitsfeld auf einem -aus Symmetriegründen- 201×201 großen Gebiet berechnet.

In der *Flow* Visualisierung in Abbildung 13b ist ein Bruch in der Färbung entlang der horizontalen Bildmitte zu erkennen. Dieser Bruch zeigt an, dass oberhalb der Bildmitte das Vektorfeld (minimal) nach unten gerichtet, unterhalb der Bildmitte (minimal) nach oben gerichtet ist (vgl. Abb. 13d). Zusätzlich können wir an den Werten der Verschiebung entlang der x -Achse v_x in Abb. 13e ablesen, dass der Betrag $|v_x|$ sich von dem *Ground Truth* Wert 1 entfernt, wenn wir uns von der vertikalen Bildmitte entfernen.

Dies sind Artefakte, die wir auf die Ausrichtung der Gradienten des Bildes u_1 zurückführen können, welche auf die Bildmitte bzw. in Richtung der Bildachsen gerichtet sind (Vgl. Abb. 13c). Alle Bildgradienten oberhalb der horizontalen Bildmitte haben positive Y -Anteile. Für die gegebenen Werte der Zeitableitung sind daher ausschließlich Abweichungen von dem *Ground Truth* nach unten möglich. Es gibt keine Punkte, die diese fehlenden Informationen oberhalb der Mitte ergänzen.

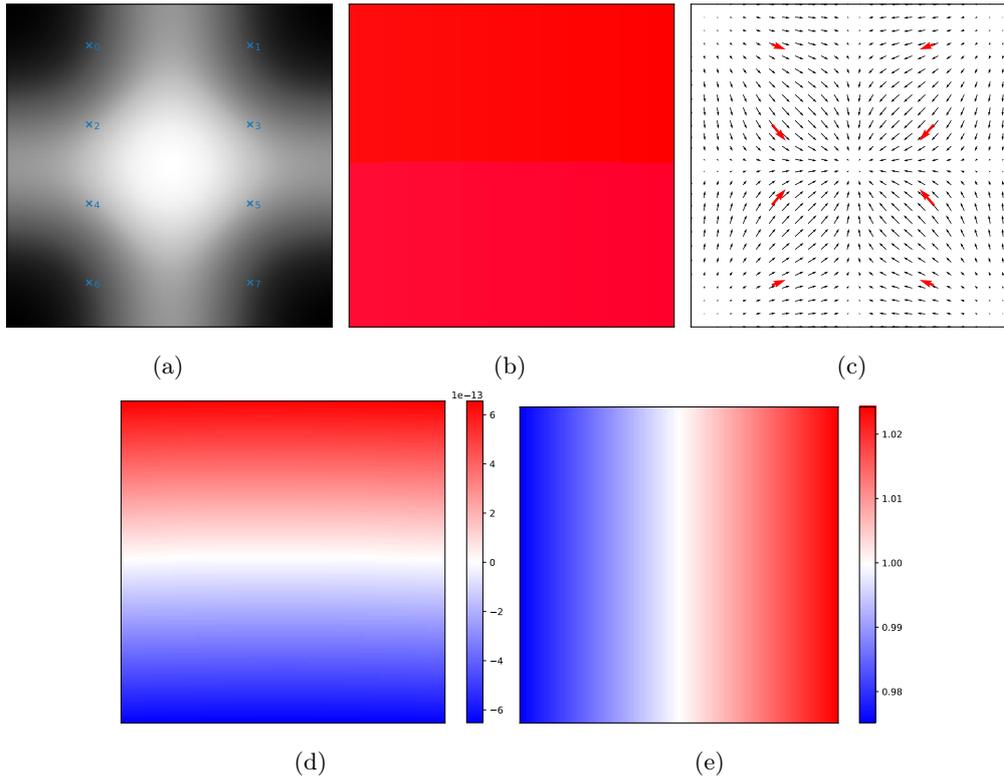


Abbildung 13: (a) Verteilung der Punkte. (b) Flow Visualisierung für Translation rechts. (c) Gradientenvektorfeld. In rot die Gradienten in den Berechnungspunkten (d) Abweichung von v_y von dem *Ground Truth* Wert $v_{GT_y} = 0$ (e) Abweichung von v_x von dem *Ground Truth* Wert $v_{GT_x} = 1$.

zen könnten. Zusätzlich verschlechtern sich die Abweichungen zu dem oberen/ unteren Bildrand. Dort sind die Gradienten der Punkte 0, 1 bzw. 6, 7 fast parallel zu dem *Ground Truth* Vektorfeld. Wie wir aus Kapitel 2.3.1 wissen, können lokal Bewegungen, die orthogonal zur Bewegung liegen, nicht unterschieden werden. Daher besteht an den Rändern eine Unsicherheit für Bewegung in Y-Richtung. An der horizontalen Mitte kann diese Unsicherheit einerseits durch die diagonalen Gradienten, andererseits durch die Informationen der naheliegenden Punkte unterhalb der Mitte, ergänzt werden.

Ähnlich verhält es sich mit den Abweichungen entlang der x-Achse aus Abb. 13e. Auf der rechten Seite der vertikalen Bildmitte wird der Optimalwert $v_x = 1$ tendenziell überschätzt, auf der linken Seite eher unterschätzt. Auch hier sind die Gradienten der rechten Bildhälfte nach links ausgerichtet und vice versa. Die Werte sind nur dann präzise, wenn die verschiedenen Informationen in der Bildmitte aufeinander treffen. Sicherheit für die tatsächliche Größe von v_x würde ein Punkt orthogonal zu der Bewegung liefern. In Kapitel 2.3.1 haben wir festgestellt, dass lokal die Länge der orthogonalen Projektion von v auf den Gradienten ∇u gegeben ist durch

$$\|v_{\perp}\| = \left\| \frac{-u_t}{\|\nabla u\|} \frac{\nabla u}{\|\nabla u\|} \right\| = \frac{|u_t|}{\|\nabla u\|}.$$

Läge v orthogonal auf dem Gradienten ∇u , so gilt $\|v_{bot}\| = \|v\|$ und dies würde mehr lokale Informationen liefern.

Eine mögliche Erklärung dafür, dass die Abweichungen in X-Richtung um ein vielfaches größer sind als in Y-Richtung ist, dass hier nur in ausschließlich einer Reihe von Punkten gemessen wird.

Die *Thin-Plate Spline* Interpolation verstärkt die Ausschläge zum Rand dementsprechend. Insgesamt lässt sich zusammenfassen, dass bei einer monotonen Ausrichtung der Gradienten, die richtigen Bewegungsvektoren nicht ausreichen bestimmt werden können. Die lokale Information, die in einem Punkt zur Verfügung steht, kann nicht durch die benötigten Informationen in der Umgebung ergänzt werden.

10.2.2 Gradient orthogonal zur Bewegung

Ein Extremfall des vorausgegangenen Problems liegt dann vor, wenn der Gradient orthogonal zur Bewegungsrichtung steht. Da die Richtungsinformationen lokal verloren gehen, fehlt diese Information auch den anderen Punkten der Umgebung. Im folgenden Beispiel wurde der Punkt 2 aus Abbildung 11a um nur einen Pixel nach rechts verschoben. Es resultiert das folgende *Flow Field* mit den entsprechenden Fehlerwerten in Tabelle 1.

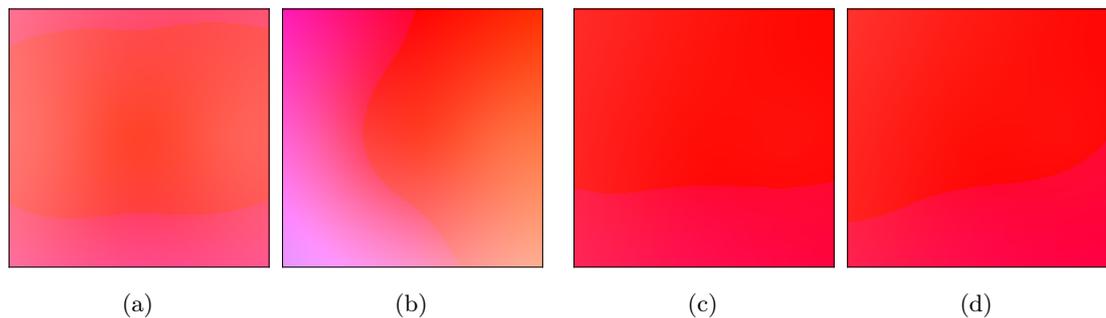


Abbildung 14: Geschwindigkeitsfelder für die Koordinatenkombination aus Abb. 11a in (a) bzw. den zweiten Punkt um einen Pixel nach rechts verschoben in (b). Im Kontrast dargestellt sind die selben Koordinaten mit dem inexakten Interpolationsverfahren für $\lambda = 1/30$ in (c) bzw. (d).

	AE	AEE	IE	NIE
(a)	0,0737	0,1267	0,2196	0,1177
(b)	0,2161	0,3693	0,4849	0,2747
(c)	0,0322	0,0541	0,0542	0,0299
(d)	0,0335	0,0560	0,0574	0,0318

Tabelle 1: Fehlerwerte zu den Geschwindigkeitsfeldern aus Abbildung 14.

Die minimale Verschiebung auf der glatten Gausskurve führt zu einer starken Verschlechterung des Vektorfeldes bezüglich aller vier getesteten Fehlermaße. Die Fehlerwerte erleben eine Verdopplung- bis Verdreifachung des Wertes vor der minimalen Veränderung. Die fehlenden Informationen in nur einem Punkt, wirken sich sehr stark auf die restlichen Punkte des gesamten Ergebnisses und nicht nur lokal aus. Dies macht es sehr schwierig bei schlechten Ergebnissen nachzuvollziehen, warum ein Vektorfeld besonders schlechte Ergebnisse liefert. Theoretisch kann dies der Einfluss eines einzigen Punktes mit orthogonalen Gradienten sein, der nichtlokal Einfluss nimmt.

Beim inexacten Interpolationsverfahren hingegen ist die Veränderung des Vektorfeldes verschwindend gering. Dies liegt vermutlich daran, dass die in dem Punkt fehlenden Informationen durch die Minimierung des Geschwindigkeitsfeldes ausgeglichen werden. Für das konstante Vektorfeld ist dies natürlich sehr einfach, daher resultiert die geringe Verschlechterung der Fehlerwerte in Tabelle 1, (c) und (d).

10.2.3 Störung in u_t

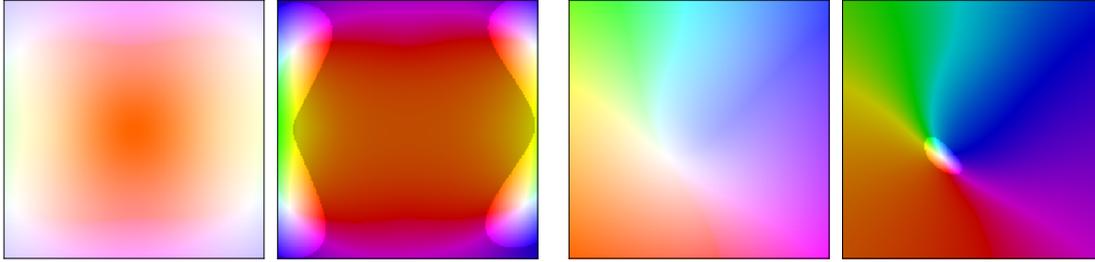
In Kapitel 8.2 haben wir bereits Formeln für die absolute und relative Kondition des Problems bzgl. Störungen der rechten Seite des Gleichungssystems, also Störungen der Zeitableitung des Bildes, erarbeitet. Diese waren gegeben durch

$$\kappa_{abs} = \sigma_1,$$

der betraglich größte Singulärwert von $K^{-1}\mathcal{T}$ und

$$\kappa_{rel} = \sigma_1 \frac{\|\alpha\|_2}{\|(A_1; A_2)\alpha\|_2} \leq \sigma_1 / \sigma_T.,$$

mit σ_1 bzw. σ_T der betraglich größte bzw. kleinste Singulärwert von (A_1, A_2) , definiert durch $A_1 := (S^{-1} - S^{-1}P(P^t S^{-1}P)^{-1}P^t S^{-1})$ und $A_2 := (P^t S^{-1}P)^{-1}P^t S^{-1}$.



(a) Koordinatenkombination aus Abb. 11a.

(b) Modifizierte Koordinaten, sodass Punkt 1 aus 11a einen Pixel nach rechts verschoben ist. Dadurch steht der Gradient orthogonal zum Ground Truth.

Abbildung 15: Geschwindigkeitsfelder mit Worst-Case Störung. Die rechte Version zeigt Fluss in gleicher Skalierung wie in Abb. 14 mit Geschwindigkeiten die über die ursprünglichen Maximalgeschwindigkeit hinaus geht geclippt und abgedunkelt.

Betrachten wir beispielsweise wieder das Problem mit neun Koordinaten in der Anordnung aus Abb. 11a für die Translation nach rechts, so erhalten wir die Werte $\kappa_{abs} \approx 25.692$ und $\kappa_{rel} \approx 30.512$. Für diese Wahl der Berechnungspunkte erwarten wir also eine Vervielfachung der Störung mit Faktor 25 bzw. 30 relativ zu der Größe von α und γ, β . Für den Fall mit orthogonalem Gradienten (Abb. 15b) verschlechtern sich die Werte sogar weiter auf $\kappa_{abs} \approx 2,879 \cdot 10^{23}$ und $\kappa_{rel} \approx 45.896$. Um den Effekt der größtmöglichen Störung der rechten Seite des Gleichungssystems zu sehen, können wir unsere vorhandenen Daten u_t mit dem Singulärvektor s_1 zu dem entsprechenden Singulärwert σ_1 stören.

Die errechnete Kondition bezieht sich dabei natürlich auf die Störung der Koeffizienten γ und β aus dem Gleichungssystem (7.2). Die Auswirkungen auf die gesamte Funktion

$$v(x) = (-1)^m \sum_{i \in I} \gamma_i E(a_i - x) \varphi_i + q_\beta(x)$$

zeigen sich im Geschwindigkeitsfeld und den berechneten Fehlerwerten. Das aus den gestörten Daten errechnete Geschwindigkeitsfeld hat die Form in Abb. 15a mit den folgenden Fehlerwerten in Tabelle 2.

Betrachten wir die *worst case* Störung für das inexakte Interpolationsproblem werden andere Probleme deutlich. Der Regularisierungsparameter $\lambda = 1/30$ wurde aufgrund des minimalen Fehlers

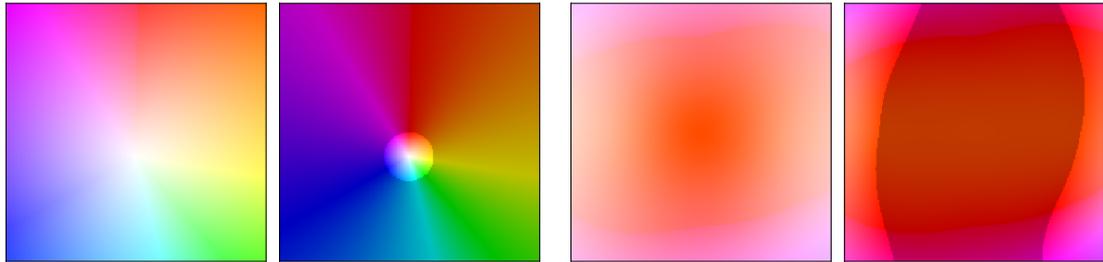
	AE	AEE	IE	NIE	κ_{abs}	κ_{rel}
Orig.:	0,0737	0,1267	0,2196	0,1177		
(a)	0,7138	2,0777	3,9095	2,0955	25,69	30,51
Orig.:	0,2161	0,3693	0,4849	0,2747		
(b)	1,4750	8,5216	10,9042	6,2852	2,88e + 23	45,90

Tabelle 2: Fehlerwerte des exakten Lösungsverfahrens bei *worst case* Störung entsprechend der *Flow* Visualisierung aus Abb. 15. Zeile „Orig.“ zeigt erneut die Fehlerwerte für ungestörte Version aus Tabelle 1. Daneben die entsprechenden Konditionswerte.

bzgl. des Geschwindigkeitsfeldes gewählt (vgl. Fehlerwerte in Tabelle 1 (c)). Berechnen wir anhand dieser Werte nun den *worst case* Fehler, erneut durch die Singulärwertzerlegung und den Singulärvektor zum größten Singulärwert, erhalten wir ein Vektorfeld welches sehr wenig mit dem *Ground Truth* Geschwindigkeitsfeld zu tun hat (siehe Abb. 16a). Hier wird deutlich, dass für ein so kleines λ die Glattheit von v überbewertet wird, dieses somit sehr klein bzgl. $|v|_m$, jedoch fern von den Interpolationsbedingungen liegt. Dies spiegelt sich natürlich in den Fehlerwerten in Tabelle 3 (a) wieder.

Ein für Störungen stabileres Ergebnis wird nur dann erreicht, wenn der Parameter λ neu gewählt wird. Ein besseres λ für diese Zusammensetzung der Koordinaten (Abb. 11a), sodass der Fehler bzgl. des gestörten Geschwindigkeitsfeldes minimal ist, ist ca. $\lambda = 1/15$. Die resultierende *Flow* Visualisierung ist in Abb. 16b mit den Fehlerwerten in Tabelle 3 (b) dargestellt.

Somit ist eine „gute“ Wahl eines λ unter Umständen anfälliger für Störungen der Eingangsdaten α .



(a) Für Parameter $\lambda = 1/30$.

(b) Für Parameter $\lambda = 1/15$.

Abbildung 16: Geschwindigkeitsfelder des inexakten Interpolationsverfahren mit Worst-Case Störung. Die rechte Version zeigt den *Optical Flow* in gleicher Größenordnung wie die ungestörte Version mit Geschwindigkeiten, die über die ursprünglichen Maximalgeschwindigkeit hinaus geht, abgeschnitten und visuell abgedunkelt.

	AE	AEE	IE	NIE	κ_{abs}	κ_{rel}
Orig.:	0,0322	0,0541	0,0542	0,0299		
(a)	1,2241	4,2968	5,4079	3,1184	8,06	22,95
(b)	0,2991	0,6378	1,2195	0,6527	8,16	16,73

Tabelle 3: Fehlerwerte des inexakten Lösungsverfahrens bei *worst case* Störung entsprechend der *Flow* Visualisierung aus Abb. 16. Daneben die Konditionswerte.

Schon an dem konstruiertem glatten Bild der Gaussglocke werden schon einige Schwächen des Verfahrens aufgedeckt. Einzelne Eigenschaften in einem Punkt werden sehr schnell auf das gesamte Gebiet verstärkt und so das Ergebnis verfälscht und die Identifikation des ursprünglichen

Problempunktes gestaltet sich als schwierig. Hinzu kommt, dass im Normalfall kein *Ground Truth* Geschwindigkeitsfeld bekannt ist. Daher gestaltet es sich als schwierig, problematische Punkte, z.B. solche mit einem Gradienten orthogonal zu der Geschwindigkeit, zu identifizieren. Auch die Wahl eines optimalen Parameters λ für das inexakte Verfahren ist problematisch.

10.3 Middlebury Datensatz „Hydrangea“



(a) Bildfolge zeigt eine Hortensie vor monotonem Hintergrund. (b) *Ground Truth* Visualisierung

Abbildung 17: „Hydrangea“ aus dem Middlebury Datensatz [15]

Mit den identifizierten Schwierigkeiten des letzten Kapitels im Hinterkopf, wollen wir nun Fehlerwerte für echte Daten des *Middlebury Optical Flow* Datensatzes berechnen. Die Bildfolge „Hydrangea“ stellt andere Ansprüche an das Verfahren. Während der Hintergrund eine sehr homogene Bewegungsrichtung nach rechts zeigt (pink in Abb. 17b), ist die Bewegung in den Blüten sehr unregelmäßig. Zudem kommen hier nun ein natürliches Bildrauschen hinzu und die Bewegungen sind nicht mehr betragslich durch Eins beschränkt.

Ziel des Verfahrens war es anhand sehr weniger Punkte ein aussagekräftiges Bewegungsfeld zu erhalten. In Abbildung 18 sind die Ergebnisse für eine maximale Anzahl von zwölf Punkten für das exakte und inexakte Verfahren zu sehen, die zugehörigen Fehlerwerte in Tabelle 4. Die Punkte wurden gleichverteilt auf dem gesamten Bild gewählt und sind in 18c zu sehen. Wir stellen fest, dass das inexakte Verfahren minimal bessere Ergebnisse liefert.

Als Parameter für das inexakte Verfahren wurde $\lambda = 1/20$ gewählt und in beiden Verfahren wurden die Eingangsbilder entsprechend geglättet (vgl. Abschnitt 10.3.1).



(a) Exaktes Verfahren. (b) Inexaktes Verfahren für $\lambda = 1/20$. (c) Verteilung der zwölf Punkte.

Abbildung 18: *Flow* Visualisierung für zwölf Koordinaten.

Möchte man über die geringe Anzahl der Koordinaten hinausgehen, ist mit einer langsamen Verbesserung der Fehlerwerte zu rechnen. In Abb. 19 (a) und (b) ist der Fehlerverlauf für das exakte und das inexakte Verfahren für bis zu 30 gleichverteilten Punkten abgebildet. Die Funktion mit

	AE	AEE	IE	NIE
(a)	0,6575	3,0370	6,4568	2,9500
(b)	0,6484	3,0030	6,4359	2,9307

Tabelle 4: Fehlerwerte für zwölf Koordinaten entsprechend Abb. 18

dem kleinsten Fehler für $\lambda = 1/20$ und 30 Koordinaten ist in Abb. 20 zu sehen. Während im Hintergrund keine deutliche Verbesserung zu erkennen ist, sind die anderen Bewegungsrichtungen innerhalb der Blüte besser zu erkennen.

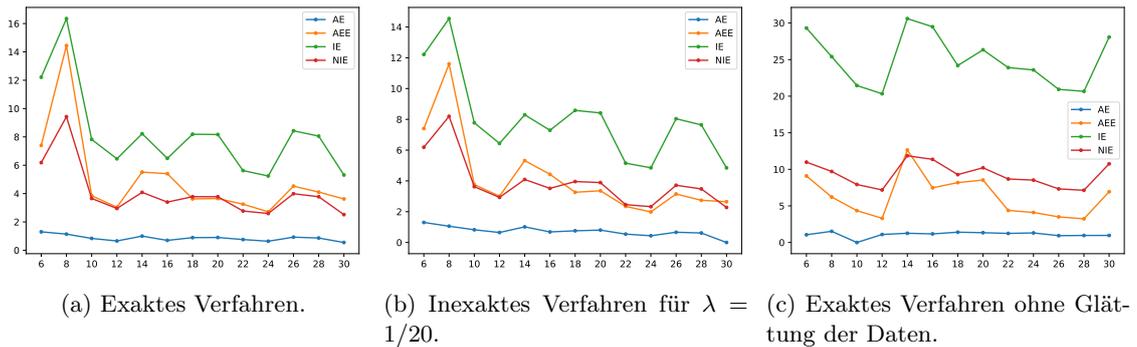


Abbildung 19: Fehlerverhalten für eine wachsende Anzahl an gleichmäßig verteilter Punkte.

Abbildung 20: Flow Visualisierung des inexakten Verfahrens für $\lambda = 1/20$, anhand dreißig Berechnungspunkten.

In Abbildung 19 (c) sind die Fehlerwerte für eine steigende Anzahl an Koordinaten bei ungeglätteten Bildern abgebildet. Wie unschwer zu erkennen ist, verbessern diese sich nicht. Im folgenden Kapitel wollen wir auf das Problem der Glättung der Eingangsdaten eingehen.

10.3.1 Glättung von Daten

Wie in Abschnitt 9.1 erwähnt, wurde die Diskretisierung so gewählt, dass nur die direkte Umgebung eines Punktes bei der Berechnung seines Gradienten berücksichtigt werden. Da wir nur wenige Punkte zur Berechnung des *Optical Flow* nutzen wollen, diese jedoch repräsentativ für die Umgebung eines Punktes sein sollen, müssen wir die Bildsequenz entsprechend glätten.

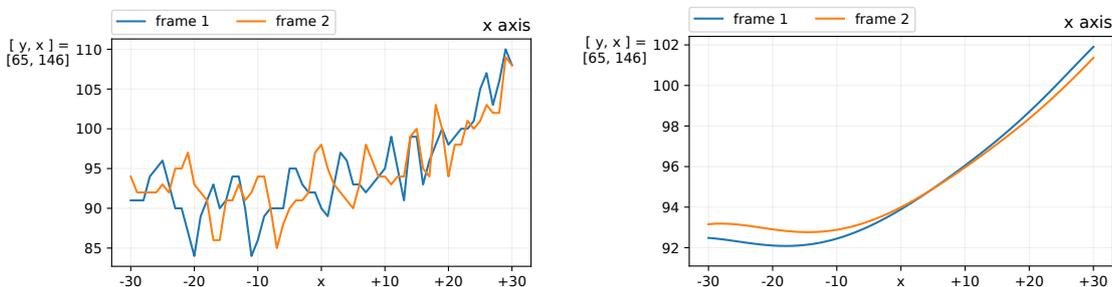
Deutlich wird dies in Abb. 21a. Dort sehen wir einen Schnitt der Bildintensität entlang der x-Achse. Abgebildet wird der Graph der Intensität links bzw. rechts des Punktes $\binom{146}{65}$. Wie in Kapitel 2.2 beschrieben, haben wir mit der *Brightness Constancy Assumption* angenommen, dass die Intensi-

tät eines Bildes entlang einer Kurve auf dem Gebiet pro Zeitschritt konstant bleibt. Entlang der x-Achse in Abb. 21a können wir die Anzeichen einer solchen gleichbleibende Intensität erkennen, der orangefarbene Graph scheint eine verzerrte und verschobene Kopie des blauen Graph zu sein. Innerhalb eines Zeitschrittes können wir mit Hilfe des *Optical Flow* eine lineare Transformation vollziehen. Dies entspricht auf dem Achsendurchschnitt einer Verschiebung des Graphen entlang der x-Achse. Betrachten wir nun den Punkt mittig im Graphen, so stellen wir fest, dass wir anhand des Gradienten keine Verbindung zwischen diesem und dem (vermuteten) Zielpunkt, circa 5 Pixel weiter rechts, herstellen können. Der Gradient im ersten *Frame* (blau) im Punkt x ist in Abb. 21a sogar gegenläufig zu dem in zweiten *Frame* (orange). Die Berechnung einer linearen Verschiebung, also eine Veränderung entsprechend der Tangente im Punkt x , macht hier keinen Sinn. In der direkten Umgebung von x sind keine Informationen enthalten, welche die Punkte miteinander verknüpfen. Die gesuchte lineare Verschiebung ist zu groß für die gegebene Datenlage. Daher müssen die Daten geglättet werden.

In Abbildung 22 ist dies anhand konstruierter Daten verdeutlicht dargestellt. Sind die Daten ausreichend geglättet (a), bzw. liegen nah genug beieinander, kann der Punkt x_2 durch die Tangente des ersten *Frames* in x_1 approximiert werden. Ist der Abstand groß (b), zeigt diese ins Leere.

In Abb. 21b ist erneut der Schnitt durch die Bilddaten entlang der Achsen dargestellt, jedoch wurde vorher ein Gaussfilter über die Eingangsbilder gelegt. Wird nun die Ableitung des ersten *Frames* im Punkt x berechnet, verläuft die Tangente in x entlang des orangefarbenen Graphs. Die Berechnung anhand dieser geglätteten Datenlage scheint gerechtfertigt.

Für die Hortensien Bildfolge haben wir für brauchbare Ergebnisse einen Gaussfilter mit einem Radius von 4 Pixel gewählt. Die Wahl erfolgte empirisch, stimmt aber ungefähr mit der durchschnittlichen Länge der *Ground Truth* Vektoren überein.

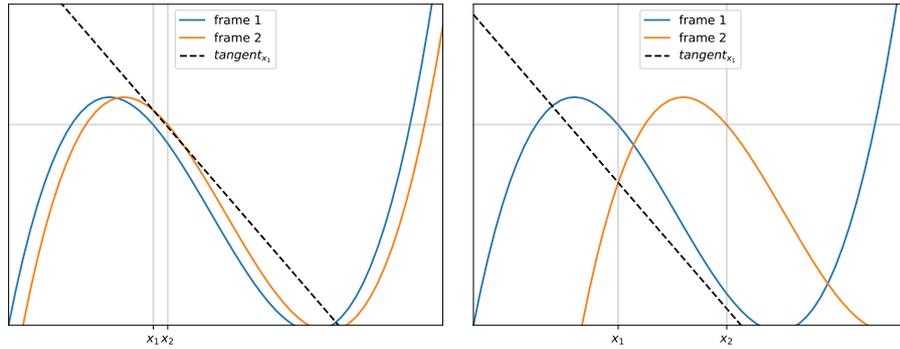


(a) Daten sind nicht geglättet. Parallelen der beiden *Frames* sind zu erkennen, jedoch ist anhand der Gradienten keine Beziehung herstellbar. (b) Daten sind geglättet. Gradienten sind aussagekräftig für Approximation.

Abbildung 21: Querschnitt der Intensität der Bildfolge „Hydrangea“ entlang der x-Achse um den Punkt $[y, x] = [65, 146]$.

10.3.2 Verhalten für verschwindende Gradienten

Das Verhalten der Fehlerwerte für verschwindende Gradienten bei anderweitig gleichbleibenden Parameter lässt sich anhand der Abbildung 23 erkennen. Dies ist interessant, da das exakte Verfahren für $\nabla u = 0$ scheitert. Störungen der Daten, die zu verschwindend kleinen Gradienten führen, sind ebenfalls problematisch. Das Verhalten der Lösung für verschwindende Parameter wurde folgendermaßen untersucht:



(a) Abstand zwischen den Punkten nach dem Zeitintervall ist klein genug, sodass die Tangente des Punktes x_1 genutzt werden kann, um die Distanz zu x_2 linear zu approximieren. (b) Abstand zwischen x_1 und x_2 ist zu groß, sodass die Tangente von x_1 in keiner Beziehung mehr zu x_2 steht. Berechnung des Flusses mit Hilfe der Gradienten macht keinen Sinn.

Abbildung 22: Vergleich von zwei *Optical Flow* Problemen mit ausreichend kleiner Distanz innerhalb eines Zeitraumes in (a) bzw. zu großer Distanz in (b).

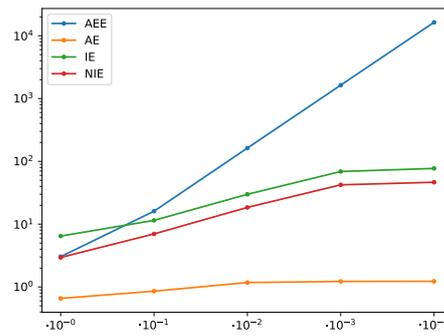


Abbildung 23: Verhalten des Fehlers für einen verschwindenden Gradienten in einem Punkt bei insgesamt neun Koordinaten.

	AE	AEE	IE	NIE
$\cdot 10^{-0}$	0,6575	3,0370	6,4568	2,9500
$\cdot 10^{-1}$	0,8570	16,0819	11,4796	6,9848
$\cdot 10^{-2}$	1,1755	162,6600	29,8830	18,4619
$\cdot 10^{-3}$	1,2243	1634,3916	68,8873	42,2738
$\cdot 10^{-4}$	1,2299	16352,3372	77,1370	46,4572

Tabelle 5: Fehlerwerte für einen verschwindenden Gradienten entsprechend Abb. 23

Vor der Berechnung der Lösungen wurden für einen Punkt die Werte beider räumlicher Ableitungen um den Faktor 10^{-i} , $0 \leq i \leq 4$ skaliert. Die Werte der zeitlichen Ableitung, sowie die Werte der verbleibenden Punkte wurden nicht verändert. Wir beobachten das Verhalten einer Lösung für ausschließlich einen Punkt, der nahezu null ist. Es zeigt sich, dass der AEE mit dem gleichen Faktor steigt, wie der Gradient sinkt. Dies ist im Einklang mit der Beobachtung, dass der durchschnittliche Betrag $|v|$ für das berechnete Geschwindigkeitsfeld v mit der gleichen Größenordnung steigt.

Dieses Verhalten ist durch die Interpolationsbedingung erklärbar. In Kapitel 2.3.1 haben wir bereits gesehen, dass die Länge der orthogonalen Projektion von v auf den Gradienten gegeben ist durch $\|v_{\perp}\| = \frac{|u_t|}{\|\nabla u\|}$. Für schnell sinkende Werte von $\|\nabla u\|$ erwarten wir somit einen wachsenden Betrag von v_{\perp} und somit v . Der Einfluss eines Punktes auf die gesamte Lösung des Problems ist so stark, dass dies sich auf das gesamte Geschwindigkeitsfeld auswirkt. Auch bei den Fehlerwerten von IE und NIE zeigt sich eine deutliche Verschlechterung des Fehlerwertes. Auch beim AE sehen wir einen signifikanten Anstieg des Fehlers. Schon bei dem Faktor 10^{-1} überschreitet die durchschnittliche Abweichung 45° und steigt bei 10^{-4} bis ca. 70° . Ein zuverlässiger Zusammenhang zwischen Geschwindigkeitsfeld und *Ground Truth* ist nicht mehr gegeben.

11 Fazit

In der vorliegenden Arbeit wurde, motiviert durch die medizinische Bildgebung, eine *Optical Flow* Problemstellung formuliert. Um Artefakte bei Aufnahmen der Intravitalmikroskopie zu vermindern, soll die Bewegung der lebendigen Probe ausgeglichen werden. Dazu muss die Geschwindigkeit und Richtung von Bewegungen mit Hilfe von Bildern bestimmt werden. Als Basis wurde ein Einstieg in das Themengebiet „*Optical Flow*“ gegeben und dort die Problematik der lokalen Schlechtgestellttheit aufgezeigt. Zudem wurde das Konzept der *Thin-Plate Spline* Interpolation erläutert.

Anhand der *Brightness Constancy Assumption* wurden die *Optical Flow Constraints* erschlossen und diese als Interpolationsbedingungen für ein angepasstes *Thin-Plate Spline* Verfahren definiert. Mit Meinguets Veröffentlichung zu *Thin-Plate Spline* Interpolation als Vorbild, wurde die eindeutige Lösbarkeit des (exakten) Interpolationsproblems nachgewiesen. Eine alternative inexakte Version wurde vorgestellt, in der die Interpolationsbedingung nur als Strafterm vorkommt. Es wurde der große Vorteil des inexakten Verfahrens gezeigt, da in einzelnen Punkten ein verschwindender Gradient zugelassen ist.

Beide Interpolationsansätze konnten als lineares Gleichungssystem formuliert werden, für das jeweils ein Algorithmus angegeben wurde. Für die Lösung der Gleichungssysteme wurde eine analytische Konditionsuntersuchung durchgeführt. Um die Lösungen der Algorithmen interpretieren zu können, wurden verschiedene Fehlermaße für *Optical Flow* Geschwindigkeitsfelder angegeben.

Bei der Untersuchung der Ergebnisse wurde festgestellt, dass grundlegende Bewegungsformen auf glatten Bildern, auch an wenigen Punkten, gut reproduziert werden können. Problematisch erwies sich, schon bei konstruierten Daten, insbesondere ein zur Bewegung orthogonaler Gradient, sowie die Instabilität bezüglich Störungen der rechten Seite des Gleichungssystems. Auch bei der Bildfolge des Middlebury Flow Datensatzes konnte einerseits für eine sehr große Anzahl an Punkten ein erkennbares Geschwindigkeitsfeld reproduziert werden, andererseits setzen sich die Problematiken hier fort. So zeigten wir hier das ungünstige Verhalten für einen verschwindenden Gradienten, insbesondere beim exakten Verfahren. Des Weiteren haben wir die ausreichende Glättung von Eingangsdaten behandelt und begründet warum die Verfahren auf rauschenden Daten keine sinnvollen Ergebnisse liefert.

Insgesamt lässt sich zusammenfassen, dass die lokale Instabilität einen sehr großen Einfluss auf die restlichen Berechnungspunkte hat. Der Ursprung der Störung ist nur schwer zu identifizieren und Ergebnisse sind daher schlecht zu interpretieren. Obwohl die Fehlerwerte mit dem inexakten Verfahren verbessert werden konnten, müssen daher noch weitere Ansätze in Betracht gezogen werden. Eine Hinzunahme von anderen oder weiteren Regularisierungstermen ist denkbar, z.B. die Bestrafung von großen Funktionswerten oder großen Gradienten. Für die praktische Anwendung zur Verminderung der Artefakte bei medizinischen Bildgebung reichen die erreichten Ergebnisse nicht aus. Insbesondere die benötigte Anzahl der Berechnungspunkte, um brauchbare Geschwindigkeitsfelder zu liefern genügt nicht den gesetzten Ansprüchen. Für besonders glatte Objekte mit rauschfreien Daten, die sich über das gesamte Bild gleichmäßig bewegen, kann das vorgestellte Verfahren jedoch bereits sinnvolle Ergebnisse liefern.

A Programmcode

A.1 Exaktes Verfahren

```

import numpy as np
import time

def thin_plate_spline_exact(coord, vec_value, scalar_value, **kwargs):
    '''
    Calculates solution to exact thin-plate spline interpolation problem.

    Parameters
    -----
    coord :
    Coordinates to the corresponding vector and scalar values

    vec_value :
    usually the gradient

    scalar_value :
    usually the time differential

    Returns
    -----
    alpha,
    poly_coeff_reord
    '''

    t = time.perf_counter()
    n = np.shape(coord)[0]

    # Calculate pairwise distance between original coordinates
    # and pairwise dot product between vector values
    distance = []
    dot_prod = []
    for i, val_i in enumerate(coord):
        for j, val_j in enumerate(coord):
            if j > i:
                distance = np.append(distance, np.linalg.norm(val_i - val_j))
                dot_prod = np.append(dot_prod, np.dot(vec_value[i], vec_value[j]))
            # Only need to calc values once since K is symmetric
        else:
            distance = np.append(distance, 0)
            dot_prod = np.append(dot_prod, 0)

    distance = np.reshape(distance, (n * n, 1))
    dot_prod = np.reshape(dot_prod, (n * n, 1))

    # Calculate matrix K
    np.seterr(divide='ignore')
    log_val = np.where(distance > 0,
        1/(8*np.pi) * distance**2 * np.nan_to_num(np.log(distance)) * dot_prod, 0)
    np.seterr(divide='warn')
    log_val = np.nan_to_num(log_val)
    K = np.reshape(log_val, (n, n))

```

```

K = K + np.transpose(K)

# Construct P
P = np.ones((n, 6))
P[:, 0] = vec_value[:, 0]
P[:, 3] = vec_value[:, 1]

for i in range(n):
    P[i, 1:3] = coord[i, :] * vec_value[i, 0]
    P[i, 4:] = coord[i, :] * vec_value[i, 1]

# Construct L
L = np.zeros((n + 6, n + 6))
L[0:n, 0:n] = K
L[0:n, n:] = P
L[n:, 0:n] = np.transpose(P)

# Construct Y
Y = np.zeros(n + 6)
Y[0:n] = scalar_value

# Solve L**-1*Y

try:
    solution = np.matmul(np.linalg.inv(L), Y)
except np.linalg.LinAlgError:
    print('Singular Matrix. Returning empty velocity field.')
vx = np.zeros((1, np.shape(target)[0]))
vy = np.zeros((1, np.shape(target)[0]))
return vx, vy

solution = np.nan_to_num(solution)
alpha = solution[0:n]
poly_coeff = solution[n:]

elapsed_time = time.perf_counter() - t
timed = kwargs.get('timed', True)
if timed:
    print('Solved TPS equation succesfully in ', elapsed_time, ' seconds.')

# Reorder poly_coeff so that we can reuse evaluate function

new_order = [1, 2, 4, 5, 0, 3]
poly_coeff_reord = np.asarray([poly_coeff[i] for i in new_order])

return alpha, poly_coeff_reord

```

A.2 Inexaktes Verfahren

```

import numpy as np
import time

def thin_plate_spline_inexact(coord, vec_value, scalar_value, lambd, **kwargs):
    '''

```

```

Calculates solution to inexact thin-plate spline interpolation problem.

Parameters
-----
coord :
Coordinates to the corresponding vector and scalar values

vec_value :
usually the gradient

scalar_value :
usually the time differential

lambda :
sets weight for the regularization parameter

Returns
-----
alpha,
poly_coeff_reord
'''

t = time.perf_counter()
n = np.shape(coord)[0]

# Calculate pairwise distance between original coordinates
# and pairwise dot product between vector values
distance = []
dot_prod = []
for i, val_i in enumerate(coord):
    for j, val_j in enumerate(coord):
        if j > i:
            distance = np.append(distance, np.linalg.norm(val_i - val_j))
            dot_prod = np.append(dot_prod, np.dot(vec_value[i], vec_value[j]))
        # Only need to calc values once since K is symmetric
    else:
        distance = np.append(distance, 0)
        dot_prod = np.append(dot_prod, 0)

distance = np.reshape(distance, (n * n, 1))
dot_prod = np.reshape(dot_prod, (n * n, 1))

# Calculate matrix K
np.seterr(divide='ignore')
log_val = np.where(distance > 0,
1/(8*np.pi) * distance**2 * np.nan_to_num(np.log(distance)) * dot_prod, 0)
np.seterr(divide='warn')
log_val = np.nan_to_num(log_val)
K = np.reshape(log_val, (n, n))
K = K + np.transpose(K)

# Construct Q
Q = np.ones((n, 6))
Q[:, 4] = vec_value[:, 0]
Q[:, 5] = vec_value[:, 1]

```

```

for i in range(n):
    Q[i, 0:2] = coord[i, :] * vec_value[i, 0]
    Q[i, 2:4] = coord[i, :] * vec_value[i, 1]

Q_trans = np.transpose(Q)

# Construct L

S_lambda = K + (1/lambda) * np.identity(n)
L = np.zeros((n + 6, n + 6))
L[0:n, 0:n] = S_lambda
L[0:n, n:] = Q
L[n:, 0:n] = Q_trans

Y = np.zeros(n + 6)
Y[0:n] = scalar_value

# Solve L**-1 * Y

try:
    solution = np.matmul(np.linalg.inv(L), Y)
except np.linalg.LinAlgError:
    print('Singular Matrix. Returning empty velocity field.')
vx = np.zeros((1, np.shape(target)[0]))
vy = np.zeros((1, np.shape(target)[0]))
return vx, vy

solution = np.nan_to_num(solution)
alpha = solution[0:n]
poly_coeff = solution[n:]

elapsed_time = time.perf_counter() - t
timed = kwargs.get('timed', True)
if timed:
    print('Solved TPS equation succesfully in ', elapsed_time, ' seconds.')

return alpha, poly_coeff

```

A.3 Funktionsauswertung

```

import numpy as np

def evaluate_function(coord, poly_coeff, alpha, vec_value, target):
    '''
    Calculate and return value of optical flow function v for requested points
    specified with target

    Parameters
    -----
    coord :
    Coordinates to the corresponding vector and scalar values

    poly_coeff :

```

```

Coefficients with shape [a1, a2, a3, a4, b1, b2] corresponding the polynom Ax +
                        b, A = [[a1, a2], [a3, a4]]

alpha :
Coefficients for the reproducing kernel

vec_value :
usually gradients

target:
[x,y] coordinates where v should be evaluated at

Returns
-----
v_x, v_y : function values evaluated at coordinates specified with 'target'
,,'

vx = []
vy = []

for i in target:

distance_target = np.empty((np.shape(coord)[0]))

for pos, j in enumerate(coord):
distance_target[pos] = np.linalg.norm(j-i)

np.seterr(divide='ignore')
log_val_target = np.where(distance_target > 0,
1/(8*np.pi) * distance_target**2
* np.nan_to_num(np.log(distance_target)), 0)
np.seterr(divide='warn')

sum_weights = np.multiply(alpha, log_val_target)
sum_weights = np.nan_to_num(sum_weights)

vx = np.append(vx, poly_coeff[4] + poly_coeff[0] * i[0] + poly_coeff[1]
* i[1] + np.sum(np.multiply(sum_weights, vec_value[:, 0])))
vy = np.append(vy, poly_coeff[5] + poly_coeff[2] * i[0] + poly_coeff[3]
* i[1] + np.sum(np.multiply(sum_weights, vec_value[:, 1])))

return vx, vy

```

Literatur

- [1] Edward H. Adelson und J. Anthony Movshon. “Phenomenal coherence of moving visual patterns”. In: *Nature* 300.5892 (1982), S. 523–525. ISSN: 0028-0836. DOI: 10.1038/300523a0.
- [2] Mauricio A. Alvarez, Lorenzo Rosasco und Neil D. Lawrence. “Kernels for Vector-Valued Functions: a Review”. In: (2011). arXiv: 1106.6251 [stat.ML].
- [3] Gilles Aubert und Pierre Kornprobst. *Mathematical Problems in Image Processing. Partial Differential Equations and the Calculus of Variations*. 2006. DOI: 10.1007/978-0-387-44588-5.
- [4] Simon Baker u. a. *A Database and Evaluation Methodology for Optical Flow*. 2007. DOI: 10.1.1.157.659.
- [5] F. L. Bookstein. “Principal warps: thin-plate splines and the decomposition of deformations”. In: 11 (1989), S. 567–585. ISSN: 0162-8828. DOI: 10.1109/34.24792.
- [6] Philip J. Davis. *Interpolation and Approximation*. Dover Publications, 1975. ISBN: 9780486624952.
- [7] Hendrik M. Dirks. “Variational Methods for Joint Motion Estimation and Image Reconstruction. Partial Differential Equations and the Calculus of Variations”. Diss. Westfälische Wilhelms-Universität Münster, 2015.
- [8] Jean Duchon. “Splines minimizing rotation-invariant semi-norms in Sobolev spaces”. In: *Constructive Theory of Functions of Several Variables*. Hrsg. von Walter Schempp und Karl Zeller. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, S. 85–100. ISBN: 978-3-540-37496-1.
- [9] David J. Fleet und Yair Weiss. “Optical Flow Estimation”. In: *Handbook of Mathematical Models in Computer Vision*. Hrsg. von Paragios N., Chen Y. und Faugeras O. Springer, Boston, MA., 2006. Kap. 15, S. 239–258. DOI: 10.1007/0-387-28831-7_15.
- [10] Friedrich Gerard Friedlander und Mark Joshi. *Introduction to the theory of distributions*. 2. ed. Cambridge [u.a.]: Cambridge Univ. Press, 1998. VII, 175. ISBN: 0521649714.
- [11] Robert L. Harder und Robert N. Desmarais. “Interpolation using surface splines.” In: *Journal of Aircraft* 9.2 (1972), S. 189–191. DOI: 10.2514/3.44330. eprint: <https://doi.org/10.2514/3.44330>. URL: <https://doi.org/10.2514/3.44330>.
- [12] Sara McArdle, Zbigniew Mikulski und Klaus Ley. “Live cell imaging to understand monocyte, macrophage, and dendritic cell function in atherosclerosis”. In: 213 (2016), 2136OIA120. ISSN: 0021-9525. DOI: 10.1083/jcb.2136oia120.
- [13] Jean Meinguet. *An Intrinsic Approach to Multivariate Spline Interpolation at Arbitrary Points*. 1979. DOI: 10.1007/978-94-009-9443-0_12.
- [14] Jean Meinguet. “Multivariate interpolation at arbitrary points made simple”. In: 30 (1979), S. 292–304. ISSN: 0044-2275. DOI: 10.1007/bf01601941.
- [15] *Middlebury Optical Flow Dataset*. 2011. URL: vision.middlebury.edu/flow/ (besucht am 20.05.2020).
- [16] Joseph J. Rotman. *Advanced Modern Algebra*. Second Printing. Prentice Hall, 2003. ISBN: 0130878685.
- [17] Walter Rudin. *Functional analysis*. Second Edition. McGraw-Hill, Inc., 1991. ISBN: 0-07-054236-8.

- [18] Tom Runia. *Optical Flow Visualization*. 2018. URL: https://github.com/tomrunia/OpticalFlow_Visualization (besucht am 20.05.2020).
- [19] Laurent Schwartz. *Théorie des Distributions*. Hermann, 1950.
- [20] Lloyd Nicholas Trefethen und David Bau. *Numerical Linear Algebra*. Philadelphia, Pa.: siam, 1997. XII, 361. ISBN: 0898713617.
- [21] Laurent Younes. *Shapes and Diffeomorphisms*. 2010. DOI: 10.1007/978-3-642-12055-8.
- [22] Fuzhen Zhang. *The Schur Complement and Its Applications*. Boston, MA, 2005.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel „*Optical Flow Approximation mit Hilfe multivariater n -dimensionaler Thin-Plate Spline Interpolation*“ eigenständig verfasst habe. Es wurden keine anderen als die angegebenen Hilfsmittel und Quellen verwendet. Alle Textstellen, die im Wortlaut oder Sinn benutzt wurden, sind entsprechend unter Angabe der Quelle gekennzeichnet, einschließlich Tabellen, Skizzen, Zeichnungen und bildlicher Darstellungen.

Zudem stimme ich der Speicherung der Arbeit in einer Datenbank, sowie ihren Abgleich mit anderen Texten zum Zweck der Plagiatskontrolle zu.

Köln, 26. Februar 2021

(Oliver Schmitz)