Diffusion Maps und ihre Anwendung bei der Analyse von Automobildaten

Anna-Luisa Schwartz

Geboren am 22. Oktober 1987 in Aachen

30. August 2013

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Jochen Garcke

Institut für Numerische Simulation

Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Inhaltsverzeichnis

Notation

$\mathbb{N}$	Die Menge der natürlichen Zahlen: $\{0, 1, 2, 3, \ldots\}$
$\mathbb{R}$	Die Menge der reellen Zahlen
$[1 : n]$	Die Teilmenge von $\mathbb{N}$, die alle Elemente von 1 bis n enthält
m	Anzahl der betrachteten Datenpunkte
$\mathcal{Y}$	Die Menge der betrachteten Datenpunkte: $\{y_1, \ldots, y_m\}$
$\mathcal{M}$	(unbekannte) Mannigfaltigkeit, aus welcher die Daten stammen
$\mathbf{A}$	Eine Matrix
a_{ij}	Eintrag der Matrix $\mathbf{A}$ an der Kreuzung von i-ter Zeile und j-ter Spalte
$\mathbf{A}[i, :]$	Die i-te Zeile der Matrix $\mathbf{A}$ als Zeilenvektor
$\mathbf{A}[:, j]$	Die j-te Spalte der Matrix $\mathbf{A}$ als Spaltenvektor
$\mathbf{I}$	Die Identitätsmatrix
$\mathbf{v}$	Ein Vektor
v_i	i-ter Eintrag von Vektor $\mathbf{v}$
$\mathbf{1}$	Spaltenvektor, dessen Einträge alle eins sind

Abkürzungen

CAD	Computer Aided Design
EMD	Earth Mover's Distance
FE	Finite Elemente
FEM	Finite-Elemente-Methoden
LM	Laplace-Matrix
ML	Maschinelles Lernen
PCA	Hauptachsentransformation, *principal component analysis*

1 Einleitung

Die numerische Simulation von Automobil-Crashtests unter Anwendung von Finite-Elemente-Methoden ist in den letzten 30 Jahren elementarer Bestandteil der Entwicklung neuer Automobile geworden. Ohne aufwändige physikalische Crashtests an Prototypen durchführen zu müssen, können so rechnergestützt Variationen von Bauteilen und Geometrie des Fahrzeugs detailliert auf mögliche Auswirkungen auf das Crashverhalten untersucht werden. Ein Beispiel für einen Crashtest mit der zugehörigen Simulation zeigt Abbildung 1.1.

Abbildung 1.1: Darstellung eines Automobil-Crashtests und der dazu passenden Computersimulation (Quelle: [6])

Mit wachsenden Rechenleistungen steigt jedoch auch der Umfang der bei der Simulation anfallenden Daten massiv an. Heute wird ein Fahrzeug durch mehr als eine Million Finite-Element-Knoten (FE-Knoten) dargestellt und der Crash in bis zu 100 Zeitschritten gespeichert, was zu Daten von Dimensionen in einer Größenordnung von 10^8 für einen einzigen Simulationsdurchlauf führt.
Um diese Datenmengen effizient auswerten und die einzelnen Simulationsdurchläufe in Bezug zueinander setzen zu können, werden Methoden des *Maschinellen Lernens* (ML) und insbesondere der *Dimensionsreduktion* eingesetzt.

Maschinelles Lernen in der Crashtest-Simulation

Ziel der Anwendung von Maschinellem Lernen bei der Analyse von Simulationsdaten ist es, computergestützt Ähnlichkeiten und Unterschiede zwischen verwandten Simulationsdurchläufen zu identifizieren, welche durch Variation von Parametern wie zum Beispiel der Blechdicke generiert wurden. So ist in praktischen Beispielen oftmals eine *Bifurkation* zu erkennen, also eine Aufspaltung der Ergebnisse in zwei unterschiedliche Zustände, abhängig von Variationen bestimmter Parameter. Klassischerweise werden solche Zusammenhänge von einem Ingenieur durch Betrachtung der visualisierten Simulationsläufe identifiziert. Im Idealfall sollte ein ML-Algorithmus diese Aufgabe übernehmen und die Gruppierung und Auswertung der Daten damit deutlich vereinfachen. In anderen Feldern erfolgreiche Verfahren sind jedoch häufig nur bedingt auf die Anwendung bei Simulationsdaten übertragbar, da vergleichsweise wenige Datenpunkte (weniger als 1000 Simulationsdurchläufe) in einem sehr hochdimensionalen Raum vorliegen – wie oben erwähnt sind Dimensionen von 10^8 durchaus üblich. In klassischen ML-Problemstellungen dagegen ist die Anzahl der Daten meist deutlich höher als die der Dimensionen.
Ein gängiger Ablauf in der Automobilindustrie ist eine dreistufige Dimensionsreduktion [2]. In einem Vorverarbeitungsschritt werden die Daten vorbereitet (z.B. durch Clustering-Algorithmen), dann wird die tatsächliche Dimensionsreduktion durchgeführt, welche die Daten unter Berücksichtigung der intrinsischen Geometrie in einen niedrigerdimensionalen Raum abbildet. In einem letzten Schritt können die Daten dann visualisiert und weiter analysiert werden.
Das am weitesten verbreitete Verfahren für die Dimensionsreduktion ist die *Hauptachsentransformation*. Diese Methode setzt jedoch voraus, dass die Daten in einem linearen Unterraum liegen. Um Daten auf nichtlinearen Mannigfaltigkeiten adäquat erfassen und in einen passenden niedrigdimensionalen Raum einbetten zu können, werden für die Analyse der stark nichtlinearen Automobildaten daher Methoden der *Nichtlinearen Dimensionsreduktion* angewandt. Bewährt hat sich hierfür unter anderem das 2006 von Coifman und Lafon [4] vorgestellte Verfahren *Diffusion Maps* [2, 12].

Problemstellung

Die vorliegende Arbeit konzentriert sich auf die Dimensionsreduktion mittels Diffusion Maps. Konkret wird der Fall betrachtet, bei dem die einzelnen Datenpunkte in unterschiedlich großen Räumen liegen.
Grundlage von Diffusion Maps ist der Einsatz einer Kernfunktion, welche die Ähnlichkeiten unterschiedlicher Datenpunkte widerspiegeln soll. Üblicherweise basieren diese auf Abstandsfunktionen zwischen den betrachteten Daten. Bisher wird bei der Auswertung von Automobildaten mit Diffusion Maps der euklidische Abstand zwischen *Verschiebungsvektoren* betrachtet, welche für jeden FE-Gitterpunkt die absolute Verschiebung enthalten [2, 12]. Ein solches Vorgehen erfordert jedoch, dass die Verschiebungsvektoren aller verglichenen Simulationsläufe die gleiche Dimension – also die gleiche Anzahl verwendeter FE-Knoten – aufweisen. Dies ist in der Praxis nicht immer gegeben. Häufig sollen beim Vergleich

von Simulationsdaten zum Beispiel bestimmte Bauteile variiert werden, welche die Geometrie des Fahrzeugs und damit auch das individuelle FE-Gitter verändern. Um derart unterschiedliche Simulationsläufe in Bezug zu setzen, kann man Projektionen auf Hilfsgitter verwenden, welche jedoch rechenintensiv sind und insbesondere bei variabler Geometrie schnell an ihre Grenzen stoßen.
Fraglich ist somit, welche Abstandsbegriffe man auf Simulationsläufe mit variablen Geometrien anwenden kann, um bei der Dimensionsreduktion und folgenden Analyse adäquate Ergebnisse zu erlangen.

Ziel dieser Arbeit

In dieser Arbeit wird erstmals der Ansatz untersucht, Abstände zwischen Simulationsläufen durch eine Verwendung von *Histogrammen* der Verschiebungsvektoren zu realisieren. Durch die Darstellung der hochdimensionalen Daten als Histogramme wird bereits in einem Vorverarbeitungsschritt eine Dimensionsreduktion durchgeführt, welche dann durch den Einsatz von Diffusion Maps auf Grundlage eines passenden Histogrammabstandes noch einmal verfeinert wird.
Ziel der Arbeit ist es, Diffusion Maps für Automobildaten unter Verwendung einer besonderen Metrik für Histogramme, der *Earth Mover's Distance* (EMD) zu implementieren. Anschließend soll das Verfahren anhand mehrerer Datensätze in Hinblick auf Ergebnisqualität und Laufzeit getestet und der Einfluss von unterschiedlichen Parametern analysiert werden.

Eigene Beiträge

- Implementierung eines Diffusion Maps-Verfahrens für Histogramme (Datenextraktion, Preprocessing, Dimensionsreduktion und Visualisierung) auf Grundlage eines Entwurfs von Rodrigo Iza-Teran [12].
- Untersuchung des Einflusses der unterschiedlichen Parameter auf Stabilität und Ergebnisqualität.
- Testen des gewählten Verfahrens im Vergleich mit anderen Metriken.

Aufbau der Arbeit

Zunächst erläutert *Kapitel 2* die Herkunft und Struktur der verwendeten Crashtest-Daten in der Automobilindustrie.
In *Kapitel 3* werden anschließend mathematische Grundlagen vorgestellt: Nach einer kurzen Einführung in die Dimensionsreduktion wird hier genauer auf das Konzept von Diffusion Maps eingegangen. Anschließend werden einige Histogrammabstände präsentiert, wobei ein deutlicher Schwerpunkt auf der Earth Mover's Distance liegt.
Das folgende *Kapitel 4* beschreibt Konzept und Implementierung einer möglichen Lösung für die maschinelle Analyse der betrachteten Automobildaten. Dass diese Lösung auch sinnvoll ist, welche Einschränkungen zu beachten sind und das unterschiedliche Verhalten bei unterschiedlichen Abstandsfunktionen wird dann

in *Kapitel 5* untersucht. Zum Abschluss dieses Kapitels werden die Ergebnisse kritisch diskutiert und mögliche Ungenauigkeiten abhängig von den zu untersuchenden Datensätzen gezeigt.
Abschließend fasst *Kapitel 6* die Resultate dieser Arbeit zusammen und stellt sie in einen größeren Zusammenhang.

Danksagung

An dieser Stelle möchte ich all denjenigen danken, die mich während der Anfertigung dieser Arbeit unterstützt und motiviert haben. Ich bedanke mich bei Prof. Dr. Jochen Garcke für die jederzeit hervorragende Betreuung, sowie bei Prof. Dr. Marc Alexander Schweitzer für die Übernahme der Zweitkorrektur.
Mein besonderer Dank gilt meinem Betreuer bei Fraunhofer SCAI, Rodrigo Iza-Teran, der mir dieses spannende Thema und erste Entwürfe des Programms für diese Bachelorarbeit überlassen hat, für seine stets offene Tür, seine Hilfsbereitschaft und seine vielen Anregungen und Hilfestellungen.

Andreas Müller stand mir von Anfang an mit Rat und Tat zu Python und zu Machine Learning zur Seite und hat trotz großer eigener Belastung immer Zeit für meine Fragen gefunden. Tobias Gödderz, Julia Volmer, Anne Wertenbruch und Lars Wallenborn haben tatkräftig den Kampf gegen Fehler und Unklarheiten in dieser Arbeit aufgenommen und mir damit in der letzten Phase der Arbeit sehr geholfen. Dafür danke ich ihnen von Herzen.

Anna und Julia danke ich für ihre Freundschaft und die großartige Unterstützung in jeder Hinsicht.
Abschließend danke ich Lars für seinen moralischen Beistand, seine fachliche Hilfe und vor allem für die stetige Motivation, an Herausforderungen zu wachsen.

2 Automobildaten

2.1 Crashberechnung

Motivation

Bei der Betrachtung der Sicherheit von Kraftfahrzeugen unterscheidet man zwischen aktiver und passiver Sicherheit. Während in das Fahrzeug eingebaute Systeme wie ABS und ESP *aktiv* zum Vermeiden von Unfällen beitragen, soll durch *passive* Sicherheit das Verletzungsrisiko für Fahrzeuginsassen und andere Verkehrsteilnehmer im Falle eines Unfalls verringert werden. Hierzu werden begleitend zum gesamten Entwicklungsprozess des Fahrzeugs Crashtests mit Prototypen und menschenähnlichen Puppen (Dummys) durchgeführt, die die Verformung des Fahrzeugs und die Beeinträchtigung von Insassen untersuchen sollen. In Abbildung 2.1 wird beispielhaft der Aufbau für den standardisierten Euro NCAP-Frontalaufpralltest gezeigt.

Wichtige Kennwerte der passiven Sicherheit sind z.B. die *Stirnwandintrusion*, also die Verformung der Blechwand, welche Motor- und Passagierraum trennt, die Kraftabsorbtion der Barriere oder die Beschleunigungen und Geschwindigkeiten, die auf die Dummys wirken. So misst zum Beispiel der sogenannte *HIC-Wert* (Head Injury Criterion) den Verletzungsgrad im Kopfbereich, abhängig von dessen Beschleunigung a:

$$\mathrm{HIC} = \max_{t_2 - t_1 = 36\mathrm{ms}} \left[(t_2 - t_1) \left(\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} a(t)\,\mathrm{d}t \right)^{2{,}5} \right]$$

Ein HIC-Wert von 1000 gilt als kritisch in Bezug auf lebensgefährliche Verletzungen [20].

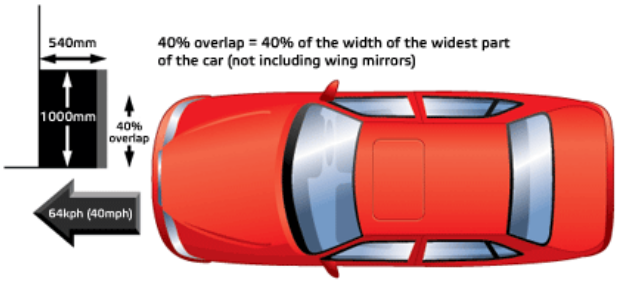

Abbildung 2.1: Schematische Darstellung des Frontalaufpralltests von EuroNCAP (Quelle: `www.euroncap.com`)

Die Idee, computergestützte Crashsimulationen in der Automobilindustrie zusätzlich zu tatsächlichen, destruktiven Tests an Prototypen einzuführen, kam in den 1980'er Jahren in Deutschland auf [10] und ist seitdem essentieller Bestandteil der Entwicklung neuer Fahrzeugmodelle. Ohne dass kosten- und zeitaufwändig neue Prototypen gebaut werden müssen, können dadurch schon in der Frühphase der Entwicklung Variationen von Geometrie und Material in Hinblick auf Veränderungen des Crashverhaltens analysiert werden. Physische, destruktive Tests werden meist nur noch zu Verifikationszwecken zusätzlich durchgeführt.
Über die Hälfte der in der Automobilentwicklung verwendeten Rechnerressourcen fließen in die Crashsimulation; mit der über die Jahre wachsenden Rechenleistung stieg auch die Komplexität der Modelle weiter an. Auf leistungsstarken Parallelrechnern läuft eine Simulation üblicherweise eine Nacht lang.
Grundlage der Simulationen sind mathematische Modelle, die anhand von physikalisch motivierten Differentialgleichungen und geometrischen Daten aufgestellt und anschließend mittels numerischer Methoden berechnet werden.

Modellbildung

Um einen Crashtest bestmöglich zu modellieren, werden eine Vielzahl an Gleichungen aus unterschiedlichen Teilbereichen der Physik benötigt: Mechanik des starren Körpers, Kontinuumsmechanik fester, deformierbarer Körper, Hydromechanik, Aeromechanik und Thermodynamik [20]. Als Unterbereich der Kontinuumsmechanik spielt vor allem die *Elastoplastizität* eine große Rolle, welche die elastischen und plastischen Verformungen des Materials beschreibt. Wichtige Materialparameter in diesem Bereich sind die Spannungs-Dehnungs-Kurven, die das Verhalten eines Werkstoffs bezüglich seiner Verformungen beschreiben, sowie als spezielle Konstante daraus der E-Modul, der beschreibt, welchen Widerstand das Material seiner elastischen Verformung entgegensetzt. Aus allen betrachteten Gleichungen sowie gegebenenfalls vereinfachenden Annahmen lässt sich dann ein physikalisches Modell erstellen. Für die geometrische Modellbildung liegen als Ausgangspunkt gewöhnlich bereits *CAD (Computer Aided Design)*-Daten des Fahrzeugs vor. Diese werden zunächst für die weitere Verwendung in der Simulation aufbereitet, indem die Geometrie vervollständigt und gegebenenfalls Details vereinfacht werden (Vernachlässigung kleiner Bohrungen oder kleiner Radien) [20].

Die anschließende mathematische Modellierung besteht darin, die physikalischen Gleichungen auf das geometrische Modell anzuwenden, was zu einem System aus unterschiedlichen Typen gewöhnlicher und partieller Differentialgleichungen führt. Da im Allgemeinen keine exakte Lösung möglich ist, werden die Gleichungen mit numerischen Methoden gelöst und dabei nur an diskreten Punkten ausgewertet. Für die räumliche Diskretisierung werden dabei *Finite-Elemente (FE)*-Methoden verwendet. Hierfür wird das Fahrzeugmodell von einem engmaschigen Netz an *FE-Knoten* überzogen. Durch dieses Netz wird das Fahrzeugmodell in üblicherweise drei- oder viereckige Elemente unterteilt, die als Grundlage für die Berechnung dienen.

Unter Verwendung von Kombinationen aus elementweise definierten *Ansatzfunktionen*, welche auch die entsprechenden Materialeigenschaften einbeziehen, sowie vorgegebenen Randbedingungen (z.B. äußere Kräfte, die auf das System wirken) kann auf Grundlage dieses FE-Netzes dann ein Gleichungssystem aufgestellt werden. Zur Lösung dieses – bei Crashberechnungen nichtlinearen – Gleichungssystems werden iterative Verfahren, wie z.B. das Newton-Verfahren, eingesetzt. Ist neben der örtlichen auch eine zeitliche Diskretisierung nötig, so erfolgt diese gewöhnlich durch explizite Runge-Kutta-Verfahren [5].
Um den Rahmen dieser Arbeit nicht zu sprengen, sei bezüglich der mathematischen Details zu Finite-Elemente-Methoden auf die einschlägige Literatur verwiesen, z.B. auf [3] und auf [20].

Kontaktberechnung

Wichtigen Einfluss auf die Crashsimulation haben auch Algorithmen für Kontaktberechnungen: Durch die große Anzahl einzelner Komponenten und die starken Krafteinwirkungen kann es sowohl zu Kontakten unterschiedlicher Bauteile untereinander als auch eines stark verformten Bauteils mit sich selbst kommen, wodurch diese Bauteile weiter deformiert würden. Dadurch, dass a priori nicht klar ist, welche Bereiche in Kontakt miteinander kommen, müssen Algorithmen große Teile des Fahrzeugs auf mögliche Kontakte überprüfen und dann die entsprechenden Berechnungen ausführen. Obwohl moderne Algorithmen dieses Verfahren sehr effizient durchführen können, ergibt sich ein großer Anteil des Rechenaufwandes der Simulation in diesem Bereich [20].

2.2 Simulationsdaten und deren Verwaltung

Die computerbasierte Crashberechnung läuft damit technisch wie folgt ab:

Präprozessor In der Vorverarbeitung werden zunächst die Daten z.B. über eine CAD-Schnittstelle importiert und dann entsprechend individueller Vorgaben das FE-Netz erstellt. Die Materialdaten (Verformungsverhalten, Materialkonstanten, Blechdicken, ...) werden vom Ingenieur zugewiesen. Anschließend werden Randbedingen und im System wirkende Kräfte bestimmt und daraus ein Gleichungssystem erstellt.

Gleichungslöser Das entstandene nichtlineare Gleichungssystem wird durch ein (iteratives) Verfahren numerisch gelöst.

Postprozessor Die berechnete Ausgabe, welche etwa Verschiebungen, Spannungen und Beschleunigungen zu unterschiedlichen Zeitschritten enthält, kann weiter analysiert werden. Hierunter fallen insbesondere die Visualisierung der Verformungen sowie die Berechnung der Kennwerte zur Crashsicherheit wie HIC-Wert und Stirnwandintrusion.

Für diese Prozesse wurden hoch spezialisierte kommerzielle Programme entwickelt, welche eine Vielzahl von Materialeigenschaften, Lösern und Elementen zur

Auswahl stellen. Beispiele hierfür sind LS-DYNA [17] mit dem Vor- und Nachbearbeitungspaket LS-Prepost sowie PAM-CRASH [6]. Ebenfalls verwenden diese Programme komplexe Datenstrukturen, um die Simulationsdaten zu speichern. Hierbei werden Information aus der Eingabe zu sämtlichen Bauteilen und Details (Metadaten), sowie alle Ergebnisse der Simulation abgelegt. Für diese Arbeit wurden alle Daten in das von der GNS mbH entwickelte *.a4db*-Format umgewandelt [9], eine spezielle Version des allgemeinen HDF5-Formats (*Hierarchical Data Format*) [33].
Da die bei der Simulation erzeugten Daten sehr groß sind und während der Entwicklung eines Fahrzeugmodells tausende Simulationsläufe durchgeführt werden, ist eine effiziente Datenverwaltung notwendig. Die großen Automobilkonzerne verwenden bereits derartige *SDM* (simulation data management oder Simulationsdatenverwaltungs)-Systeme, welche jedoch wenige Möglichkeiten zur Analyse bieten; Auswertung und Einordnung der Daten erfolgen häufig noch durch einen Nutzer unter Verwendung von 3D-Visualisierungen der Simulationsläufe. Hier wurden in den letzten Jahren durch Fraunhofer SCAI [2, 27, 12] Ansätze zu einer computergestützten Auswertung der Daten mithilfe von Methoden der Dimensionsreduktion vorgestellt. Aufbauend auf dem in [12] vorgeschlagenen Vorgehen von Rodrigo Iza-Teran wird sich diese Bachelorarbeit einem Aspekt davon widmen. Hierfür werden im nächsten Kapitel mathematische Grundlagen vorgestellt.

3 Mathematische Grundlagen

Im Folgenden sei $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\} \subset \mathbb{R}^n$ eine endliche Datenmenge.

3.1 Dimensionsreduktion

Beim Arbeiten mit hochdimensionalen Daten – und hier bezeichnet *hochdimensional* bereits Räume mit Dimension $d > 10$ – treten häufig unerwünschte Nebeneffekte auf: der sogenannte *Fluch der Dimension*. Die Komplexität der Approximation von Funktionen oder der Abschätzung von Wahrscheinlichkeitsdichten hängt exponentiell von d ab. Für klassische Anwendungen des Maschinellen Lernens, in denen man aus einer kleinen Stichprobe von Daten Informationen über zugrunde liegende Strukturen ableiten möchte, ist eine hohe Dimension daher sehr ungünstig. Auch eine geeignete Visualisierung von Daten lässt sich bereits ab $d \geq 5$ nicht mehr umsetzen.
Häufig hat die hohe Dimension der vorliegenden Daten jedoch nur mit ihrer Darstellung zu tun, und nicht mit der Komplexität der sie erzeugenden Prozesse; ein Großteil der Dimensionen ist *redundant*, da die betrachteten Variablen eng miteinander korrelieren.
Es ist also zu hoffen, die vorliegenden Daten auch mit deutlich weniger Variablen effizient beschreiben und in der Folge eine bessere Anschauung des die Daten erzeugenden Prozesses gewinnen zu können. Das Finden dieser vereinfachten Darstellung bezeichnet man als *Dimensionsreduktion*, konkret die Suche nach einer Abbildung (der sogenannten *Einbettung*) $f : \mathcal{Y} \subset \mathbb{R}^n \longrightarrow \mathbb{R}^p$ mit $p < n$, welche die Informationen aus $\mathcal{Y}$ weitgehend erhält. Welche der Informationen erhaltenswert sind und welche verworfen werden können, hängt von der Anwendung ab.
Ein wichtiges Verfahren zur Dimensionsreduktion ist die *Hauptachsentransformation* (Principal Component Analysis, kurz PCA). PCA baut auf der Modellannahme auf, dass die Daten aus $\mathcal{Y}$ in einem linearen Unterraum des $\mathbb{R}^n$ liegen. Es wird der p-dimensionale Unterraum gesucht, auf den eine Orthogonalprojektion der betrachten Daten die größtmögliche Varianz erhält [16]. Obwohl viele Phänomene nichtlinearer Natur und damit nicht angemessen durch PCA erfassbar sind, gehört PCA aufgrund seiner Einfachheit zu den beliebtesten Methoden der Dimensionsreduktion.
Methoden, die von der Annahme ausgehen, dass die Daten auf einer nichtlinearen Mannigfaltigkeit liegen, bezeichnet man als *Nichtlineare Dimensionsreduktion* oder auf Englisch als *Manifold Learning*. Klassischerweise ist das Ziel dieser Methoden die Bewahrung geometrischer Informationen, also von Abständen: Liegen $x, y \in \mathcal{Y}$ nahe beieinander, so soll diese Nähe sich auch zwischen $f(x)$ und $f(y)$ im *Einbettungsraum* $\mathbb{R}^p$ wiederfinden. Es ist also zunächst nötig, geeignete Abstände zu wählen, welche der Struktur der (unbekannten) Mannigfaltigkeit

entsprechen. Euklidische Abstände sind hierfür häufig ungeeignet.
Alternativ kann man z.B. den *geodätischen Abstand* auf der Mannigfaltigkeit verwenden oder sich diesem durch das Suchen von kürzesten Wegen in einem Graphen über der Datenmenge (den sogenannten *Graph-Abstand*, vgl. Abbildung 3.1) annähern. Bekanntester Algorithmus hierfür ist *ISOMAP* [32].

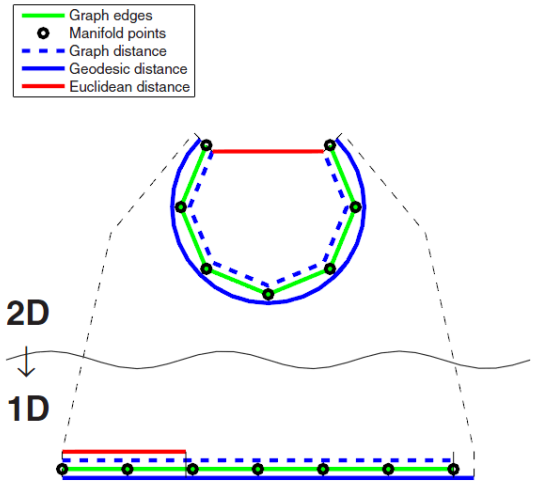

Abbildung 3.1: Abstände der Endpunkte einer C-förmigen Kurve (als eindimensionale Untermannigfaltigkeit des $\mathbb{R}^2$): Euklidischer, geodätischer und Graph-Abstand, oben im Ursprungsraum $\mathbb{R}^2$, unten in einer Einbettung in den $\mathbb{R}^1$. Wenn genügend Punkte auf der Mannigfaltigkeit bekannt sind, bietet der Graph-Abstand eine gute Approximation an den geodätischen Abstand. (Quelle: Lee, Verleysen: Nonlinear Dimensionality Reduction, [16])

Im Gegensatz zu *globalen Methoden* wie ISOMAP, welche Abstände in Bezug auf den gesamten Raum betrachten, wird bei den sogenannten *lokalen Methoden* ein Fokus auf lokale Nachbarschaften zwischen den Datenpunkten gelegt. Diese Verfahren zielen darauf ab, Informationen über die globale Struktur des Datenraums aus sich überlappenden lokalen Strukturen zu gewinnen [15]. Hierfür werden häufig Kernfunktionen statt klassischer Abstände verwendet, welche nähere Abstände höher gewichten können.
Eine Familie von Verfahren, die einen Abstand betrachten, der sich aus lokalen Strukturen zusammensetzt und dabei einem Diffusionsprozess ähnelt, ist *Diffusion Maps*.

3.2 Diffusion Maps

Angenommen, die Punkte aus $\mathcal{Y}$ liegen auf einer Mannigfaltigkeit $\mathcal{M}$. Um die Geometrie dieser Mannigfaltigkeit zu beschreiben, verwendet man eine *Kernmatrix* $\mathbf{K} \in \mathbb{R}^{m \times m}$ mit $k_{ij} = k(y_i, y_j)$ und k einer positiv semidefiniten Kernfunktion (*Mercer-Kern*) [29].

Definition 1 (Positiv semidefiniter Kern)**.** *Eine Kernfunktion* $k : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ *heißt* positiv semidefinit*, falls*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\bar{y}_i, \bar{y}_j) \geq 0$$

für alle endlichen Folgen $(\bar{y}_i)_{i=1}^n$ *über* $\mathcal{Y}$ *und* $(\alpha_i)_{i=1}^n$ *über* $\mathbb{R}$*.*

Die genaue Wahl des Kerns hängt von der Anwendung ab. Da er generell als Maß der Ähnlichkeit zwischen Punkten fungieren soll, ist der Kern häufig abhängig von einer Abstandsfunktion $d : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$. Sehr üblich bei Betrachtung lokaler Geometrien ist die Verwendung des Gauß-Kerns mit Parameter ε:

Definition 2 (Gauß-Kern)**.**

$$k_\varepsilon(x, y) := \exp\left(-\frac{d(x, y)^2}{\varepsilon}\right).$$

Im Folgenden wird ein Abstandsbegriff auf $\mathcal{Y}$ durch die Definition einer *Irrfahrt* (Random Walk) über die Datenpunkte eingeführt:
Betrachtet man die Menge $\mathcal{Y}$ als Knotenmenge eines vollständigen Graphen, so kann man den Kanten die Ähnlichkeit der inzidenten Knoten als Gewichte zuweisen: $\omega(\{y_i, y_j\}) := k(y_i, y_j)$. Die zuvor definierte Kernmatrix $\mathbf{K} \in \mathbb{R}^{m \times m}$ entspricht dann der Adjazenzmatrix des derart konstruierten Graphen.
Wird $\mathbf{K}$ nun zeilenweise mit Hilfe der diagonalen Gradmatrix $\mathbf{D} \in \mathbb{R}^{m \times m}$, $d_{ii} := \sum_j k_{ij}$ normalisiert, so führt dies zu der Matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{K}$. Wegen der engen Verwandtschaft zu $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{K}$, der *normalisierten Laplacematrix* des Graphen[1], wird auch $\mathbf{P}$ in der Literatur gelegentlich als Laplacematrix bezeichnet.
Da die Matrix $\mathbf{P}$ per Konstruktion eine stochastische Matrix ist, kann sie als Übergangsmatrix einer zeitlich homogenen Markovkette betrachtet werden, die eine Irrfahrt auf $\mathcal{Y}$ definiert. Der Eintrag p_{ij} entspricht der Wahrscheinlichkeit, in einem Zeitschritt von y_i zu y_j zu gelangen, formal notiert: $\forall\, s \geq 0 : p_{ij} = \Pr(x_{s+1} = y_j | x_s = y_i)$. Die zu $\mathbf{P}$ gehörige Kernfunktion bezeichnen wir als $p(y_i, y_j)$.
Zieht man nun auch einen Zeitparameter $t \in \mathbb{N}$ hinzu, so beschreibt die Potenz $\mathbf{P}^t$ die Übergangswahrscheinlichkeiten über t Zeitschritte, und der zugehörige

[1] Je nach Anwendung wird der Begriff *Laplacematrix* (LM) unterschiedlich definiert. Häufig wird die hier genannte Matrix auch als Random-Walk-Laplacematrix $\mathbf{L}_{\text{rw}}$ bezeichnet, und der Begriff normalisierte LM bezieht sich auf die Matrix $\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$.

Kern $p_t(y_i, y_j) = (\mathbf{P}^t)_{ij}$ entsprechend die Wahrscheinlichkeit, in t Zeitschritten von Startpunkt x_i zu x_j zu gelangen: $\forall\, s \geq 0 : \mathbf{P}^t_{ij} = \Pr(x_{s+t} = y_j | x_s = y_i)$.

Die Markovkette enthält also Informationen über eine Ausbreitung über die Zeit, welche im Folgenden verwendet werden, um einen zeitabhängigen Abstandsbegriff auf $\mathcal{Y}$ zu definieren. Hierfür nehmen wir an, dass $k(y_i, y_j) > 0$ für alle $y_i, y_j \in \mathcal{Y}$.

Definition 3 (Diffusionsabstand)**.** *Gegeben sei die oben beschriebene Irrfahrt auf $\mathcal{Y}$. Es seien $x, y, z \in \mathcal{Y}$, π die stationäre Verteilung der Markovkette und $t \in \mathbb{R}_{\geq 0}$. Dann ist $D_t : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ mit*

$$D_t(x, y)^2 := \sum_{z \in \mathcal{Y}} (p_t(x, z) - p_t(y, z))^2 \frac{1}{\pi(z)}$$

der Diffusionsabstand mit Parameter t.

Die stationäre Verteilung[2] π existiert nach dem Ergodensatz [8], da die durch $\mathbf{P}$ beschriebene Markovkette auf einem endlichen Zustandsraum operiert und alle Einträge der Matrix $\mathbf{P}$ positiv sind.
Der Diffusionsabstand summiert somit über alle möglichen Verbindungen zwischen x und y nach $2t$ Zeitschritten. $D_t(x, y)$ wird sehr klein, wenn auf dem Graphen viele kurze Wege zwischen x und y existieren und die Übergangswahrscheinlichkeit von x nach y sehr groß ist; analog wird der Abstand größer, je unwahrscheinlicher der Übergang von x nach y ist.
Die Gewichtung des Abstands mit $\pi(z)^{-1}$ erfolgt, um sich der statistischen Datendichte anzupassen und Bereiche geringer Wahrscheinlichkeitsdichte stärker zu gewichten.
Durch die Abhängigkeit von der Zeit ergeben sich unterschiedliche Abstandsstrukturen zu unterschiedlichen Zeitpunkten, was eine differenzierte Analyse der zugrundeliegenden Struktur des Graphen ermöglicht. Der Parameter t übernimmt hier eine skalierende Rolle [4].
Um das zeitabhängige Verhalten der Irrfahrt zu analysieren, werden klassischerweise die Eigenwerte und -vektoren der Übergangsmatrix $\mathbf{P}$ betrachtet [8]. Es ist jedoch nicht selbstverständlich, dass überhaupt eine Eigenwertzerlegung durchgeführt werden kann. Hierfür sei $\mathbf{A}$ die zu $\mathbf{P}$ adjungierte Matrix

$$\mathbf{A} = \mathbf{D}^{\frac{1}{2}} \mathbf{P} \mathbf{D}^{-\frac{1}{2}}$$

Die Matrix $\mathbf{A}$ ist – im Gegensatz zu $\mathbf{P}$ – symmetrisch und teilt das Spektrum mit $\mathbf{P}$. Aus dem Spektralsatz folgt: $\mathbf{A}$ hat m reelle Eigenwerte $\{\lambda_j\}_{j=1}^m$, deren zugehörige Eigenvektoren $\{\mathbf{v}_j\}_{j=1}^m$ eine Orthonormalbasis bilden. Da $\mathbf{P}$ ergodisch ist, folgt mit dem Satz von Perron-Frobenius außerdem (für entsprechend sortierte Eigenwerte):

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n \tag{3.1}$$

[2] Die stationäre Verteilung ist die Verteilung, gegen die die Kette konvergiert, $\lim_{t \to \infty} p_t(x, y) = \pi(y)$ für alle $x, y \in \mathcal{Y}$.

Die rechten und linken Eigenvektoren $\{\phi_j\}$ und $\{\psi_j\}$ von $\mathbf{P}$ erhält man wiederum durch Konjugation mit denen von $\mathbf{A}$,

$$\phi_j = \mathbf{v}_j \mathbf{D}^{\frac{1}{2}}, \qquad \psi_j = \mathbf{v}_j \mathbf{D}^{-\frac{1}{2}}.$$

Der Diffusionsabstand kann auf diese Art auch auf Basis der Eigenwertzerlegung dargestellt werden [4]:

Satz 4.

$$D_t(x,y)^2 = \sum_{l=2}^{n} \lambda_l^{2t} (\psi_l(x) - \psi_l(y))^2.$$

Die Notation $\psi(x)$ bezeichnet für ein $x \in \mathcal{Y}$ den Eintrag des Vektors $\psi \in \mathbb{R}^m$ an der Stelle i, mit i dem Index von $x = y_i$ in der indizierten Menge $\mathcal{Y}$.
Der Term für $l = 1$ fällt in der Gleichung weg, da der Eigenvektor ψ_1 zum Eigenwert $\lambda_1 = 1$ konstant ist.

Aus Gleichung (3.1) folgt, dass der Diffusionsabstand durch Verwendung der ersten Terme der Summe approximiert werden kann. Hierfür gibt man eine Genauigkeit δ für die Approximation vor und definiert $s(\delta, t) := \max\{l \in \mathbb{N} : |\lambda_l|^t > \delta |\lambda_2|^t\}$, sowie den approximierten Diffusionsabstand

$$D_{t,\delta}(x,y)^2 := \sum_{l=2}^{s(\delta,t)} \lambda_l^{2t} (\psi_l(x) - \psi_l(y))^2$$

Um eine Dimensionsreduktion zu erreichen, welche die Abstände der Elemente aus $\mathcal{Y}$ weitgehend erhält, wählt man nun die folgende Einbettung:

Definition 5 (Diffusionsabbildungen $\{\Psi_t\}_{t\in\mathbb{N}}$)**.**

$$\Psi_t : \mathcal{Y} \longrightarrow \mathbb{R}^{s(\delta,t)-1}, \qquad x \mapsto (\lambda_2^t \psi_2(x), \lambda_3^t \psi_3(x), \ldots, \lambda_s^t \psi_s(x))^T.$$

Somit können die Datenpunkte aus $\mathcal{Y} \subset \mathbb{R}^n$ in den $\mathbb{R}^{s(\delta,t)-1}$ eingebettet werden, und man erreicht für $s(\delta,t) - 1 < n$ wie gewünscht eine Dimensionsreduktion. Die Nachbarschaften im Einbettungsraum entsprechen dabei bis auf einen von δ abhängigen Fehler den Diffusionsabständen D_t im Ursprungsraum $\mathcal{Y}$ [4]:

Satz 6. *Die Diffusionsabbildung ist eine Einbettung in den $\mathbb{R}^{s(\delta,t)-1}$ mit der Eigenschaft:*

$$||\Psi_t(x) - \Psi_t(y)|| = D_{t,\delta}(x,y).$$

Anisotrope Diffusion

Statistisch kann man $\mathcal{Y}$ als eine auf Basis einer bestimmten Wahrscheinlichkeitsverteilung aus $\mathcal{M}$ gezogene Stichprobe betrachten. Die Verwendung einer isotropen Kernfunktion wie des Gauß-Kerns führt je nach statistischem Ursprung der Daten zu unterschiedlichen Ergebnissen. Belkin und Niyogi zeigen in [1], dass bei

einer uniform zufällig aus $\mathcal{M}$ gezogenen Stichprobe die Eigenvektoren der Laplace-Matrix eine diskrete Approximation von Eigenfunktionen des Laplace-Beltrami-Operators auf der Mannigfaltigkeit $\mathcal{M}$ darstellen und damit sehr gut geeignet sind, die geometrische Struktur der Mannigfaltigkeit aufzudecken. Ist jedoch – wie häufig in der Praxis – die Verteilung der Stichprobe nicht bekannt, so konvergieren die betrachteten Eigenvektoren der Laplace-Matrix statt gegen die Eigenfunktionen des Laplace-Beltrami-Operators gegen die des allgemeineren Fokker-Planck-Operator auf $\mathcal{M}$,

$$\mathcal{H}\psi = \Delta\psi - 2\nabla\psi \cdot \nabla U$$

mit $U = -\log(\mu(x))$ und μ der Wahrscheinlichkeitsdichte, entsprechend welcher gezogen wurde [22]. Der Term $2\nabla\psi \cdot \nabla U$ entspricht also einer Tendenz in Richtung von Regionen mit höherer Wahrscheinlichkeitsdichte. Die Einbettung, die von Diffusion Maps gefunden wird, hängt daher nicht nur von der Geometrie, sondern auch von der Wahrscheinlichkeitsdichte auf der Mannigfaltigkeit ab. Will man die Wahrscheinlichkeitsverteilung für die Einbettung außer Acht lassen, kann man den isotropen Gauß-Kern $k_\varepsilon(x, y)$ durch den anisotropen Gauß-Kern, $k_\varepsilon(x, y)/D(x)D(y)$ ersetzen. Die Terme $D(x) = \sum_i k(x, x_i)$ entsprechen hierbei den Einträgen der diagonalen Gradmatrix $\mathbf{D}$, welche als Approximation der Wahrscheinlichkeitsverteilung, mit der $\mathcal{Y}$ gezogen wurde, verwendet wird. Mit diesem Kern konvergieren die Eigenvektoren der betrachteten Laplace-Matrix auch bei nicht uniformer Stichprobe gegen die des Laplace-Beltrami-Operators [4].

3.3 Spektrales Clustering

Da Diffusion Maps die Einbettung über die Eigenvektoren einer Laplace-Matrix umsetzt, gehört es zu den sogenannten *Spectral Embedding*-Methoden der Dimensionsreduktion, ebenso wie z.B. *Laplacian Eigenmaps* [1]. Führt man Spectral Embedding nur als Vorverarbeitung durch, um die eingebetteten Daten anschließend mittels Clustering-Algorithmen zu gruppieren, so bezeichnet man das kombinierte Verfahren als *Spectral Clustering*. Meist werden für diese Methoden die ersten $k-1$ nicht-trivialen Eigenvektoren verwendet, um insgesamt k Cluster zu finden. Die bekanntesten Algorithmen sind die von Shi & Malik [30], Ng, Jordan & Weiß [23] sowie Meila & Shi [19]. Die Spectral Clustering-Algorithmen unterscheiden sich vor allem in der Wahl der Normalisierung der Laplace-Matrix sowie in der Wahl des Clustering-Verfahrens im Einbettungsraum: Bei [30] werden durch ein verallgemeinertes Eigenwertproblem die Eigenvektoren von $\mathbf{I}-\mathbf{D}^{-1}\mathbf{K}$ für die Einbettung verwendet, bei [23] die Eigenvektoren von $\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$ und bei [19] diejenigen von $\mathbf{D}^{-1}\mathbf{K}$.

Die betrachteten Eigenvektoren sind durch ihren Ursprung sehr eng miteinander und mit denen für Diffusion Maps verwandt. So hat die zuvor betrachtete Matrix $\mathbf{A} = \mathbf{D}^{\frac{1}{2}}(\mathbf{D}^{-1}\mathbf{K})\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$ die gleichen Eigenvektoren wie die von Ng et al. verwendete, die Eigenwerte sind bei $\frac{1}{2}$ gespiegelt. Die Eigenvektoren der anderen Verfahren erlangt man durch Adjungieren, was die Eigenschaften und Strukturen der Einbettung nicht wesentlich verändert. Ergebnisse und Analysen im Rahmen von Spectral Clustering-Verfahren sind somit weitgehend auf Diffu-

sion Maps übertragbar, weshalb in dieser Arbeit im Rahmen der Experimente in *Kapitel 5* teils darauf zurückgegriffen wird.

3.4 Histogramme

Wie eben betrachtet, hängt die für die Dimensionsreduktion verwendete Kernfunktion vor allem von der Wahl einer Abstandsfunktion d auf der Menge $\mathcal{Y}$ ab. Für die Anwendung auf Simulationsdaten wird bisher der euklidische Abstand der n-dimensionalen Verschiebungsvektoren gewählt [12], wobei n der Anzahl an FE-Knoten in der betrachteten Simulationsschar entspricht. Diese Verschiebungsvektoren enthalten für jeden Knoten dessen absolute Verschiebung im dreidimensionalen Raum.
Werden in einer Schar von Simulationsdurchläufen jedoch Änderungen in der Geometrie des Fahrzeugmodells vorgenommen, so ändern sich dabei auch die FE-Gitter und die Anzahl der verwendeten Knoten ist über die Schar nicht mehr konstant. Paarweise Abstände zwischen Simulationsläufen können also nicht mehr mit dem euklidischen Abstand bestimmt werden. Um trotzdem die Simulationsläufe in Relation zueinander setzen zu können, ist – unter Verlust der Informationen über die Position der Verschiebung – eine einfache Darstellung der Verschiebungen als Histogramme und die Verwendung von Histogrammabständen möglich.

Definition 7 (Histogramm)**.** *Für eine endliche Menge $\mathcal{I} \subset \mathbb{N}^d$ und eine Abbildung $h : \mathcal{I} \longmapsto \mathbb{R}_{\geq 0}$ heißt die Menge $H = (h(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}}$* Histogramm*. Die Elemente von $\mathcal{I}$ repräsentieren* Klassen (Bins) *und die zugeordneten Werte $h(\mathbf{i})$ werden als* Gewichte *bezeichnet.*

Den Verschiebungsvektor $v = (v_1, \ldots, v_n)^T$ eines Simulationslaufs kann man nun als ein eindimensionales Histogramm ($d = 1$) mit $\mathcal{I} = [1 : n_{\text{bins}}]$ darstellen: Partitioniere eine zusammenhängende, beschränkte Teilmenge von $\mathbb{R}$, welche alle Einträge von v enthält, in n_{bins} gleichlange Intervalle $\{J_i\}_{i\in\mathcal{I}}$ und definiere die Abbildung h als: $h(i) = \frac{1}{n} |\{\, j \mid v_j \in J_i \,\}|$.

Eine wichtige Frage bei der Erstellung von Histogrammen in allen Anwendungsbereichen ist die Wahl der Klassenanzahl n_{bins} beziehungsweise der Intervalllängen $h \sim \frac{1}{n_{\text{bins}}}$. Diese Wahl hat großen Einfluss auf die Aussagekraft der Darstellung. Sowohl zu kleine als auch zu große Klassen werden den zugrundeliegenden Strukturen nicht gerecht: Kleine Klassen betonen Abweichungen sehr stark, große verschleiern die Struktur gänzlich.
In vielen statistischen Anwendungen, bei denen ungefähr bekannt ist, welche Wahrscheinlichkeitsverteilung den Stichproben zugrunde liegt, haben sich Faustregeln für die Wahl einer geeigneten Klassenanzahl abhängig von der Anzahl der betrachteten Daten m etabliert, wie Sturges-Regel [31], Scott-Regel [28] und Freedman-Diaconis-Regel [7].
Im Gegensatz zu den üblichen Anwendungen liegt bei den Automobildaten keine feste Wahrscheinlichkeitsverteilung zugrunde. Trotzdem sind aufgrund der Herkunft dieser Daten Strukturen zu erwarten – und auch tatsächlich zu erkennen

– die mehreren Normalverteilungen ähneln. Dies liegt an den Verbindungen zwischen den Gitterpunkten; verschiebt sich ein FE-Knoten im Netz, so sind auch die benachbarten Knoten davon betroffen. Aufgrund ihrer relativen Einfachheit und der Ähnlichkeit aller Empfehlungen für kleine Datenmengen, wurde als Richtwert für diese Arbeit die Sturges-Regel

$$n_{\text{bins}} = \lceil \log_2(m) + 1 \rceil \tag{3.2}$$

verwendet. Im Rahmen der numerischen Experimente in *Kapitel 5* wurde diese Anzahl variiert und der Einfluss auf das Ergebnis beobachtet.

3.5 Abstände zwischen Histogrammen

Vor allem zur Analyse und inhaltsbasierten Suche digitaler Bilder sind Histogramme und deren Abstände in den letzten Jahren ausführlich untersucht worden. So können zum Beispiel Graustufenbilder durch eindimensionale Histogramme über Intervalle der Farbwerte dargestellt werden, wobei die Gewichte dem Anteil der Pixel in dieser Farbstufe entsprechen [13].
Abstandsbegriffe zwischen Histogrammen lassen sich grob aufteilen in *Bin-to-Bin* und *Cross-Bin*-Abstände. Erstere verwenden nur die Gewichte einander entsprechender Klassen. Sie sind also von der Form

$$d(H,K) = \sum_{\mathbf{i}\in\mathcal{I}} f(h(\mathbf{i}),k(\mathbf{i})) \text{ mit Histogrammen } H = (h(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}} \text{ und } K = (k(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}}\,.$$

Je nach Abstandsbegriff wird die Funktion f sehr unterschiedlich gewählt. Insbesondere wird bei dieser Art von Abständen die Topologie des Histogramms völlig außer acht gelassen, Nachbarschaften zwischen Klassen spielen keine Rolle. Bei Cross-Bin-Abständen hingegen werden auch benachbarte oder sogar alle weiteren Klassen in Betracht gezogen. Hierfür verwendet man einen diskreten *Grundabstand* $d_{\mathbf{ij}}$, welcher als Abstand zwischen den durch $\mathbf{i}$ und $\mathbf{j}$ beschriebenen Klassen definiert ist. Diese Konstruktion gibt zum einen die Möglichkeit, Abstände zu formulieren, die deutlich mehr mit menschlicher Wahrnehmung übereinstimmen (vgl. Abbildung 3.2), und ist zum anderen robuster bezüglich der Wahl der Klassengrenzen. Naheliegenderweise sind Cross-Bin-Abstände jedoch gleichzeitig rechenintensiver.

Durch das Einbeziehen des Grundabstandes können Cross-Bin-Abstände auch auf sogenannte *Signaturen* angewandt werden, eine Art Erweiterung von Histogrammen für variable Indexmengen $\mathcal{I}$:

Definition 8 (Signatur)**.** *Sei $X \subset \mathbb{R}^s$ ein Raum und für $j \in [1:n]$ seien $S_j = (\mathbf{m}_j, \omega_j)$ mit Punkten $\mathbf{m}_j \in X$ und Gewichten $\omega_j \in \mathbb{R}_{\geq 0}$. Eine Menge $\{S_1, \ldots, S_n\}$ sei als* Signatur *bezeichnet.*

In obiger Definition entsprechen die Punkte $\mathbf{m}_j$ beispielsweise Mittelpunkten von Clustern und die Gewichte ω_j dem Durchmesser eines Clusters. Darum wird das Paar S_j auch häufig selbst als *Cluster* bezeichnet.

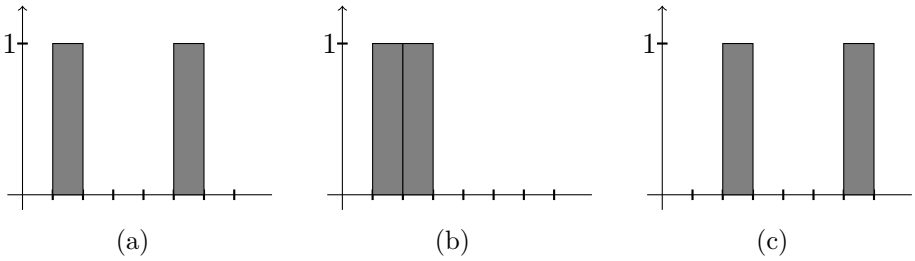

Abbildung 3.2: Histogrammabstände: Bei Betrachtung der Histogramme erscheinen (a) und (c) dem menschlichen Betrachter als ähnlich. Ein Bin-to-Bin-Histogrammabstand kann die Verschiebung um eine Klasse jedoch nicht erfassen und wird daher den Abstand von (a) und (b) entgegen der menschlichen Wahrnehmung größer einordnen als den zwischen (a) und (c).

Bemerkung. *Ein Histogramm $H = (h(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}}$ kann als Signatur gesehen werden, wobei die Cluster der Signatur den* a priori *durch $\mathcal{I}$ bestimmten Klassen des Histogramms zugeordnet sind.*

In vielen Anwendungen ist die Verwendung individuell aufgestellter Signaturen pro Datensatz flexibler und erlaubt es zudem, die relevanten Daten sparsamer zu speichern. Rubner et al. [26] schlagen die Nutzung von Signaturen im Zusammenhang mit inhaltsbasierter Bildersuche (Content Based Image Retrieval) vor. Für die Anwendung in dieser Arbeit wird im Rahmen der Darstellung der Implementierung in *Kapitel 4* kurz eine Verwendung von Signaturen anstelle von Histogrammen diskutiert.

3.5.1 Earth Mover's Distance (EMD)

Die Earth Mover's Distance (EMD) [26] ist eine Cross-Bin-Abstandsfunktion zwischen zwei Signaturen, die z.B. in der Bildersuche verwendet wird.
Die Bezeichnung entstammt der folgenden Analogie: Gegeben seien die beiden Signaturen $S = \{(\mathbf{s}_1, \omega_1), \ldots, (\mathbf{s}_k, \omega_k)\}$ mit k Clustern und $T = \{(\mathbf{t}_1, \nu_1), \ldots, (\mathbf{t}_n, \nu_n)\}$ mit n Clustern und eine *Grundabstandsmatrix* $\mathbf{D}$, mit d_{ij} dem *Grundabstand*, der die Kosten des Transports einer Masseeinheit von $\mathbf{s}_i$ nach $\mathbf{t}_j$ beschreibt.
Betrachtet man nun eine der Signaturen bildlich als Menge von Erdhaufen an den Positionen $\mathbf{s}_i$ mit Größe ω_i und die andere als Menge von Erdlöchern an Positionen $\mathbf{t}_i$ mit Kapazitäten ν_i, so gibt die EMD die optimalen Kosten für den Transport der Erde von S nach T an. Hierfür wird ein diskretes Optimierungsproblem gelöst:

Gesucht ist ein Fluss $\mathbf{F} = [f_{ij}]$ mit f_{ij} dem Fluss zwischen Quellen $\mathbf{s}_i$ und Senken $\mathbf{t}_j$, der die Gesamtkosten minimiert, also

$$\min_{\mathbf{F}} \mathrm{COST}(S, T, \mathbf{F}) \quad \text{mit } \mathrm{COST}(S, T, \mathbf{F}) := \sum_{i=1}^{k}\sum_{j=1}^{n} d_{ij} f_{ij}$$

unter den Nebenbedingungen

$$f_{ij} \geq 0 \qquad \text{für alle } 1 \leq i \leq k, 1 \leq j \leq n \tag{3.3}$$

$$\sum_{j=1}^{n} f_{ij} \leq \omega_i \qquad \text{für alle } 1 \leq i \leq k \tag{3.4}$$

$$\sum_{i=1}^{k} f_{ij} \leq \nu_j \qquad \text{für alle } 1 \leq j \leq n \tag{3.5}$$

$$\sum_{i=1}^{k}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{k}\omega_i, \sum_{j=1}^{n}\nu_j\right). \tag{3.6}$$

Bedingung (3.3) stellt sicher, dass der Transport nur von den Quellen S zu den Senken T erfolgt und nicht in die Gegenrichtung. Die Nebenbedingungen (3.4) und (3.5) beschränken den ausgehenden Fluss bei den Quellen S und den eingehenden Fluss bei den Senken T auf die Gewichte der jeweiligen Cluster und Bedingung (3.6) erzwingt, dass die größtmögliche Menge an Masse transportiert wird. Diese wird im Folgenden als *Gesamtfluss* bezeichnet.

Definition 9 (Earth Mover's Distance)**.** *Sei* $\boldsymbol{F^*} = [f_{ij}^*]$ *der optimale Fluss, der das beschriebene Transportproblem löst. Dann entspricht die Earth Mover's Distance* $EMD(S,T)$ *dessen Gesamtkosten normalisiert durch den Gesamtfluss:*

$$EMD(S,T) := \frac{COST(S,T,\boldsymbol{F^*})}{\sum_{i=1}^{k}\sum_{j=1}^{n} f_{ij}^*} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n} d_{ij} f_{ij}^*}{\sum_{i=1}^{k}\sum_{j=1}^{n} f_{ij}^*}$$

Durch die Normalisierung mit dem Gesamtfluss, der dem Gesamtgewicht der kleineren Signatur entspricht (vgl. Gleichung (3.6)), wird sichergestellt, dass kleinere Signaturen nicht bevorzugt werden, wenn S und T unterschiedliches Gewicht haben.

Bemerkung. *Wenn alle Signaturen gleiches Gesamtgewicht haben und der Grundabstand eine Metrik ist, so ist auch EMD eine Metrik.*

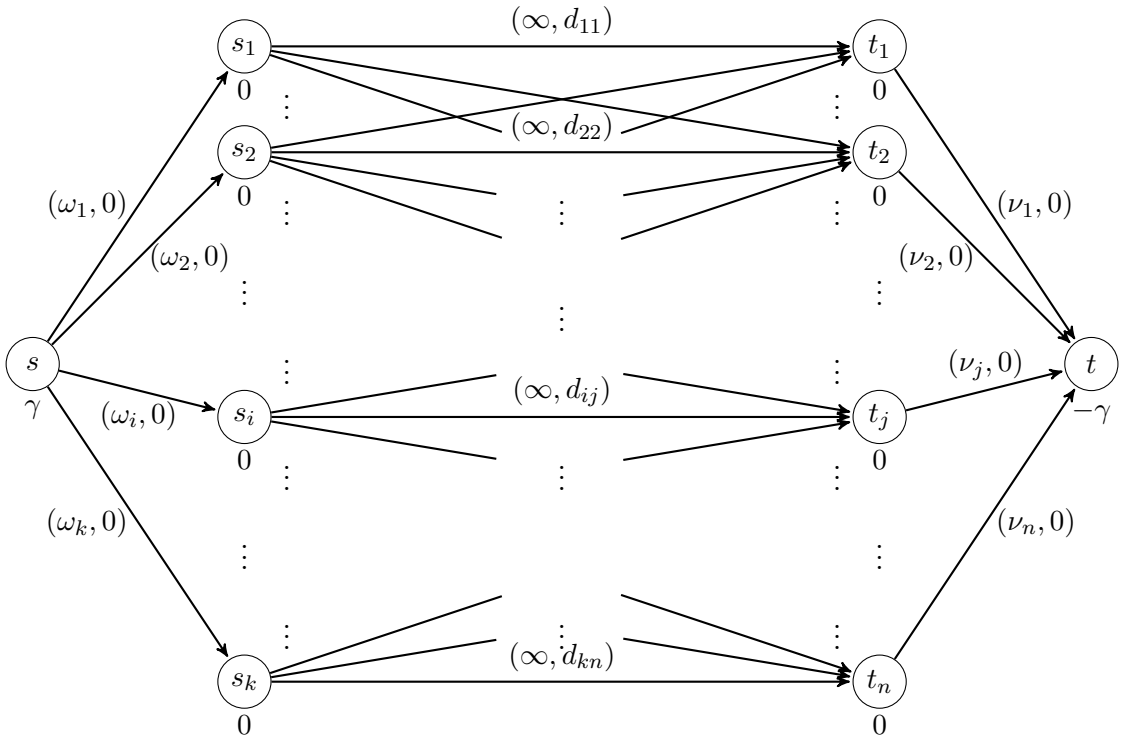

Abbildung 3.3: Schematische Darstellung der Modellierung von EMD als Min-Cost-Flow-Problem. Die Knoten $\{s_i\}_i$ und $\{t_j\}_j$ entsprechen den jeweiligen Clustern der Signaturen. Die Kanten sind mit (u, c) entsprechend ihrer Kapazität u und ihren Kosten c beschriftet, die Knoten mit ihrer Balance b.

Für die algorithmische Bestimmung von EMD muss also ein diskretes Optimierungsproblem gelöst werden. Hierfür kann man das Problem als *Minimum-Cost-Flow*-Problem in einem Netzwerk formulieren. Das allgemeine Minimum-Cost-Flow-Problem ist wie folgt definiert [14]:

Definition 10 (Minimum-Cost-Flow-Problem)**.**

Gegeben *Ein gerichteter Graph $G = (V, E)$ mit Kapazitäten $u : E \to \mathbb{R}_{\geq 0}$, Kosten $c : E \to \mathbb{R}$ und einer Balance $b : V \to \mathbb{R}$ mit $\sum_{v \in V} b(v) = 0$*

Gesucht *Eine Funktion $f : E \to \mathbb{R}_{\geq 0}$ mit $f(e) \leq u(e)$ für alle $e \in E$ und $\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(v)$ für alle $v \in V$, welche die Kosten $COST(f) := \sum_{e \in E} f(e)c(e)$ minimiert (falls es eine solche Funktion gibt).*

Für die Modellierung seien $S = \{ s_i \}_{i=1}^{k}$ und $T = \{ t_j \}_{j=1}^{n}$. Definiere einen gerichteten kantengewichteten Graphen $G = (V, E)$ mit Kantenkapazitäten durch $V = S \dot\cup T \dot\cup \{ s, t \}$ mit zwei künstlichen Knoten s und t und E, u und c wie folgt:

- Für alle $i \in [1 : k]$ sei $e = (s, s_i) \in E$, $u(e) = \omega_i$ und $c(e) = 0$.
- Für alle $j \in [1 : n]$ sei $e = (t_j, t) \in E$, $u(e) = \omega_i$ und $c(e) = 0$.
- Für alle $i \in [1 : k]$ und $j \in [1 : n]$ sei $e = (s_i, t_j) \in E$, $u(e) = \infty$ und $c(e) = d_{ij}$.

Es sei $\gamma := \min \left(\sum_{i=1}^{k} \omega_i, \sum_{j=1}^{n} \nu_j \right)$ die maximal zu transportierende Masse. Die Balance b ist nur für s und t von 0 verschieden, alle anderen Knoten haben weder Angebot noch Bedarf: Für alle $i \in [1 : k]$ und $j \in [1 : n]$ gelte $b(s_i) = 0$ und $b(t_j) = 0$. Das Angebot von Knoten s und der Bedarf von Knoten t wird als γ definiert: $b(s) = \gamma$, $b(t) = -\gamma$. Das so erzeugte Netzwerk ist schematisch in Abbildung 3.3 dargestellt.
Eine Lösung des derart modellierten Minimum-Cost-Flow-Problems entspricht also genau dem minimalen Fluss aus der Problemstellung für EMD. Um diesen Fluss zu finden kann auf dem Modell zum Beispiel der Netzwerk-Simplex oder der Algorithmus von Orlin verwendet werden [14]. Im Fall $n = k$ kann Orlins Algorithmus auf diesem Netzwerk mit Laufzeit $O(n^3 \log n)$ durchgeführt werden.
Alternativ zu der Modellierung über ein Flussnetzwerk können auch Lösungsverfahren für Lineare Programme eingesetzt werden.

3.5.2 Weitere Histogramm-Vergleichsmaße

Im Folgenden werden weitere übliche Abstands- oder Ähnlichkeitsmaße für Histogramme dargestellt.

Histogrammschnitt

Beim Histogrammschnitt handelt es sich nicht um einen Abstand, sondern um ein *Ähnlichkeitsmaß* zwischen zwei Histogrammen, das direkt als Kern für Maschinelles Lernen genutzt werden kann.

$$k_{\mathrm{HI}}(H, K) = \sum_{\mathbf{i} \in \mathcal{I}} \min\{h(\mathbf{i}), k(\mathbf{i})\}$$

Für auf ein Gesamtgewicht von 1 normalisierte Histogramme entspricht ein Wert von 1 völliger Übereinstimmung und 0 dem größtmöglichen Unterschied.

χ^2-Abstand

Der χ^2-Abstand ist ein Bin-to-Bin-Abstand, der die Intuition berücksichtigt, dass Unterschiede zwischen „vollen“ Bins weniger schwerwiegend sind als Unterschiede zwischen „wenig gefüllten“ Bins [25]. Er basiert auf dem in der Statistik verwendeten χ^2-Test und ist wie folgt definiert:

$$\chi^2(H, K) = \frac{1}{2} \sum_{\mathbf{i} \in \mathcal{I}} \frac{(h(\mathbf{i}) - k(\mathbf{i}))^2}{h(\mathbf{i}) + k(\mathbf{i})}$$

Darüber hinaus können auch die klassischen Abstandsmaße im $\mathbb{R}^{n_{\text{bins}}}$ für die Gewichtsvektoren des Histogramms verwendet werden:

Histogrammdifferenz (L_1-Abstand)

Die sogenannte Histogrammdifferenz HD_{abs} ist ein Bin-to-Bin-Abstand und entspricht der klassenweisen Differenz der Histogramme:

$$\text{HD}_{abs}(H, K) = \sum_{\mathbf{i} \in \mathcal{I}} |h(\mathbf{i}) - k(\mathbf{i})|$$

Euklidischer Abstand (L_2-Abstand)

$$d(H, K) = \sqrt{\sum_{\mathbf{i} \in \mathcal{I}} (h(\mathbf{i}) - k(\mathbf{i}))^2}$$

4 Algorithmen und Implementierung

Im vorherigen Kapitel wurde mit Diffusion Maps ein Verfahren der Nichtlinearen Dimensionsreduktion vorgestellt, das bereits erfolgreich für Automobildaten getestet wurde [2, 12]. Um Probleme bei unterschiedlich feinen FE-Gittern oder gar variabler Geometrie zu lösen, bei denen kein euklidischer Abstand zwischen den betrachteten Verschiebungsvektoren anwendbar ist, stellen wir die Verschiebungen zunächst als Histogramme dar, wie in Kapitel 3.5 definiert. Die Gewichtsvektoren der Histogramme werden auf ein Gesamtgewicht von 1 normalisiert, um auch bei unterschiedlicher Auflösung vergleichbare Histogramme zu erlangen.
Dieses Kapitel stellt dar, wie die Histogrammerstellung, die Berechnung von Histogrammabständen und die Dimensionsreduktion mittels Diffusion Maps kombiniert werden können, um das vorgestellte Problem zu lösen.

4.1 Methodik

Die vorgeschlagene Analyse der Automobildaten setzt sich aus den folgenden Schritten zusammen (vgl. auch [12]):

1. Extraktion der für die Analyse relevanten Daten aus den Simulationsdaten, hier: Extraktion der Verschiebungsvektoren $\mathbf{v}_i \in \mathbb{R}^{n_i}$, die für jeden der n_i FE-Gitterpunkte die absoluten Verschiebungen zu einem vorher bestimmten Zeitpunkt t_k enthalten.

2. Vorbehandlung (*Preprocessing*) der Daten, hier: Darstellung der n-dimensionalen Verschiebungsvektoren $\mathbf{v}_i$ als Histogramme $\left(h^i(j)\right)_{j\in[1:n_{\text{bins}}]}$.

3. Erstellen eines Kerns unter Verwendung einer passend gewählten Abstandsfunktion.

4. Dimensionsreduktion durch Anwendung von Diffusion Maps oder verwandten Methoden.

5. Aufbereitung und Deutung der Daten (z.B. Anwenden von Clustering-Algorithmen), hier: 3D-Visualisierung.

4.2 Algorithmen

Im Folgenden werden die verwendeten Algorithmen für Datenextraktion und Vorverarbeitung (Algorithmus 1) sowie für Diffusion Maps (Algorithmus 2) vorgestellt.

Algorithmus 1 : Vorverarbeitung der Automobildaten

Eingabe : Liste *filelist* der m Automobildatensätze $\{y_1, \ldots, y_m\}$,
Parameter n_{bins}

Ausgabe : Histogramme $\{(h^1(j))_{j\in[1:n_{\text{bins}}]}, \ldots, (h^m(j))_{j\in[1:n_{\text{bins}}]}\}$,
gespeichert als Matrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}}\times m}$ mit $w_{kl} = \left(h^l(k)\right)$

$i = 1$;
Initialisiere Verschiebungsmatrix $\mathbf{V} \in \mathbb{R}^{n\times m}$;
Initialisiere Histogrammmatrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}}\times m}$;
for *file* $\in$ *filelist* **do**
  Auslesen und Speichern der Verschiebungsvektoren in x, y, und z-Richtung $\mathbf{v}^x, \mathbf{v}^y, \mathbf{v}^z \in \mathbb{R}^n$ (mit n Anzahl der FE-Knoten des Modells);
  Berechnen des absoluten Verschiebungsvektors $\mathbf{v}^{\text{abs}}$:
  $\forall\, k : v_k^{\text{abs}} := \sqrt{(v_k^x)^2 + (v_k^y)^2 + (v_k^z)^2}$;
  $\mathbf{V}[:, i] := \mathbf{v}^{\text{abs}}$;
  i++;
end
Bestimmen von Minimum $\min_{\text{glob}}$ und Maximum $\max_{\text{glob}}$ der Einträge der Verschiebungsmatrix $\mathbf{V}$;
Definieren von n_{bins} gleich großen Klassen (aufgeteilt entsprechend $\min_{\text{glob}}$ und $\max_{\text{glob}}$) für die Histogramme;
$i = 1$;
for $k \in [1 : m]$ **do**
  Erstellen eines Histogramms $(h(j))_{j\in[1:n_{\text{bins}}]}$ des Verschiebungsvektors $\mathbf{V}[:, k]$ entsprechend der definierten Klassenaufteilung;
  Normalisieren des Histogramms auf Gesamtgewicht 1;
  Ablegen des Gewichtsvektors in $\mathbf{W}[:, i]$;
  i++;
end

Algorithmus 1 beschreibt die ersten beiden Schritte der Methode: Datenextraktion und Histogrammerstellung. Die vorherige Festlegung der Klassen entsprechend der globalen Extrema über alle Verschiebungsvektoren erfolgt, damit die Histogramme im nächsten Schritt sinnvoll verglichen werden können.
Ohne diesen Schritt kann man Algorithmus 1 auch auf die Verwendung von Signaturen verallgemeinern. Da für diese Arbeit jedoch stets mit dem Spezialfall Histogramme gearbeitet wurde, ist hier entsprechend der speziellere Algorithmus dargestellt. Insbesondere ist bei Signaturen im Allgemeinen keine Speicherung in einer einfachen Matrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}}\times m}$, wie hier verwendet, möglich und es können nicht alle der in der Arbeit betrachteten Abstände auf Signaturen verallgemeinert werden.
Zwar ist der hier hauptsächlich eingesetzte EMD-Abstand anwendbar, bei Signaturen muss aber durch die für jeden Datenpunkt unterschiedliche Klassenanzahl und -aufteilung für jede Abstandsberechnung aufs Neue der entsprechende Grundabstand berechnet werden. Dadurch, dass die Repräsentanten sich unter-

scheiden können, ist die Grundabstandsmatrix im Allgemeinen nicht symmetrisch und repräsentiert daher keine Metrik mehr. Dies beeinträchtigt die Berechnung von EMD, da weniger effiziente Algorithmen eingesetzt werden müssen.

Algorithmus 2 : Diffusion Maps mit Histogrammen für Automobildaten

Eingabe : m Histogramme aus dem Preprocessing-Algorithmus; (Histogramm-) Abstandsmaß[a] d; Skalierungsparameter γ für Gauß-Kern, Zieldimension p

Ausgabe : Diffusionskoordinaten der Daten für Einbettung in den $\mathbb{R}^p$

Erstelle Matrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}} \times m}$ mit Gewichtsvektoren der Histogramme;
Initialisiere Abstandsmatrix $\mathbf{D} \in \mathbb{R}^{m \times m}$;
for $i \in [1:m]$ **do**
  for $j \in [1:m]$ **do**
    $d_{ij} := d(\mathbf{W}[:,i], \mathbf{W}[:,j])$;
  end
end
$D_{\text{sum}} := \sum_{i \in [1:m]} \min_{j \in [1:m]} \{\, d_{ij} \mid d_{ij} \neq 0 \,\}$ (für Skalierung des Gauß-Kerns in Größenordnung des durchschnittlich niedrigsten Abstandes);
$\varepsilon := \frac{\gamma}{m} D_{\text{sum}}$;
Erstelle Kernmatrix $\mathbf{K}^{(1)}$ mit Einträgen $k_{ij}^{(1)} = \exp\left(-\frac{d_{ij}^2}{\varepsilon}\right)$;
$\mathbf{p} := \mathbf{K}^{(1)} \cdot \mathbf{1}$, mit $\mathbf{1} = (1 \dots 1)^T$;
Erstelle[b] $\mathbf{K}^{(2)} := \mathbf{K}^{(1)} ./ (\mathbf{p} \cdot \mathbf{p}^T)$ (Approximation des anisotropen Gauß-Kerns, vgl. 3.2) ;
$\mathbf{v} := \text{sqrt}(\mathbf{K}^{(2)} \cdot \mathbf{1})$;
Erstelle zu $\mathbf{K}^{(2)}$ adjungierte symmetrische Matrix $\mathbf{K} := \mathbf{K}^{(2)} ./ (\mathbf{v} \cdot \mathbf{v}^T)$;
Führe Eigenwertzerlegung von $\mathbf{K}$ durch und erhalte Eigenwerte und Eigenvektoren $\{\lambda_i\}_{i \in [1:n_{\text{bins}}]}, \{\psi_i\}_{i \in [1:n_{\text{bins}}]}$;
Gebe $(\lambda_i \psi_i)^T_{i \in [2:(p+1)]}$ aus;

[a] Bei einem Cross-Bin-Abstand beinhaltet die Eingabe d insbesondere den zugehörigen Grundabstand

[b] Die Operation ./ bezeichnet die elementweise Division

Algorithmus 2 beschreibt den dritten Schritt, die Ausführung der Dimensionsreduktion mittels Diffusion Maps. Die Matrix $\mathbf{M} \in \mathbb{R}^{m \times p}$ aufgebaut aus den Ausgabevektoren des Algorithmus, $\mathbf{M}[:,i] = (\lambda_i \psi_i)^T$ für $i \in [2 : (p+1)]$, enthält in ihren Zeilen die (Diffusions-)Koordinaten der m Datenpunkte für die Einbettung in den $\mathbb{R}^p$, welche nun zur weiteren Analyse und bei passendem p auch zur Visualisierung verwendet werden können.
Den ersten Eigenvektor zum Eigenwert $\lambda_1 = 1$ betrachten wir nicht, da es sich hier um den konstanten Vektor handelt und er somit keine für die Einbettung relevante Information beinhaltet.

4.3 Implementierung

Die Implementierung des Algorithmus erfolgt für diese Bachelorarbeit mit der dynamisch typisierten Programmiersprache *Python* in der Version 2.7.1 unter umfangreicher Verwendung der Pakete *NumPy* [34] und *matplotlib* [11].
Grundlage des für die Arbeit geschriebenen Python-Programms war ein Rohentwurf von Rodrigo Iza-Teran, aufbauend auf [12]. Für EMD wurde eine Fast-EMD-Implementierung in C++ von Ofir Pele [25] mit einem Python-Wrapper von Olivier Schwander verwendet, für den χ^2-Kern eine Implementierung aus dem Paket Scikit-Learn [24].
Die 3D-Visualisierung beim Postprocessing erfolgte auf Grundlage des 3D-Viewers von Thilo Vorderbrück, die Simulationsläufe wurden mithilfe des Programms *Animator 4* der GNS mbH [9] visualisiert.

5 Numerische Experimente

Um die Diskriminierungsfähigkeiten der in *Kapitel 4* vorgestellten Methode zu testen, wurden drei unterschiedliche Scharen von Simulationsläufen verwendet. Es wurde jeweils ein einzelner Zeitschritt t_k ausgewählt, zu dem die Verformung gut erkennbar, aber noch nicht voll ausgeprägt ist.

TRUCK Modell eines Chevrolet C2500 Pick-Ups bei frontalem Aufprall, insgesamt $m = 132$ Simulationsdurchläufe (unter Variation von 9 Parametern) mit jeweils $n = 66120$ FE-Knoten, erstellt mit LS-DYNA [17], vgl. Abbildung 5.1.

TRUCK-Beam Extrahierter Längsträger aus dem *TRUCK*-Modell, Variationen wie oben, insgesamt $m = 132$ Simulationsdurchläufe mit jeweils $n = 1714$ FE-Knoten, vgl. Abbildung 5.1.

PKW-Seite Seitenteil eines Mittelklasse-PKWs bei lateralem Aufprall. Die Geometrie der betrachten Seitenteile ist variabel aus unterschiedlichen Bauteilen kombiniert (40 bis 44 einzelne Bauteile, Originaldaten des Herstellers). Insgesamt $m = 143$ Simulationsdurchläufe (unter Variation der Geometrie), jeweils $n_i \approx 26.100$ FE-Knoten, erstellt mit PAM-CRASH [6].

Für die Untersuchungen in diesem Kapitel wurde hauptsächlich der *TRUCK-Beam*-Datensatz herangezogen. Zum einen sind die Verformungen der Längsträger wichtige Indikatoren des gesamten Crashverhalten eines Fahrzeugs, zum anderen ist es anhand der Visualisierung auch gut möglich, die tatsächlichen Simulationsergebnisse mit der errechneten Einbettung in Bezug zu setzen.
Insbesondere weist der betrachtete Längsträger bei unterschiedlichen Parametervariationen eine Bifurkation (vgl. Abbildung 5.2) im Biegeverhalten auf, welche in der Anordnung der Datenpunkte in der Einbettung erkennbar sein sollte.
Für die Messung der Ergebnisqualität ist es sehr schwierig, objektive Maßstäbe zu finden. In der folgenden Auswertung wird vor allem darauf Wert gelegt, inwiefern in der Visualisierung des Crashverhaltens ähnliche Simulationsläufe in der Einbettung nahe beieinander platziert werden, dass optisch erkennbare Ausreißer in der Einbettung ebenfalls separat liegen, und dass bei klaren Bifurkationen die beiden Modi in der Einbettung deutlich getrennt sind.

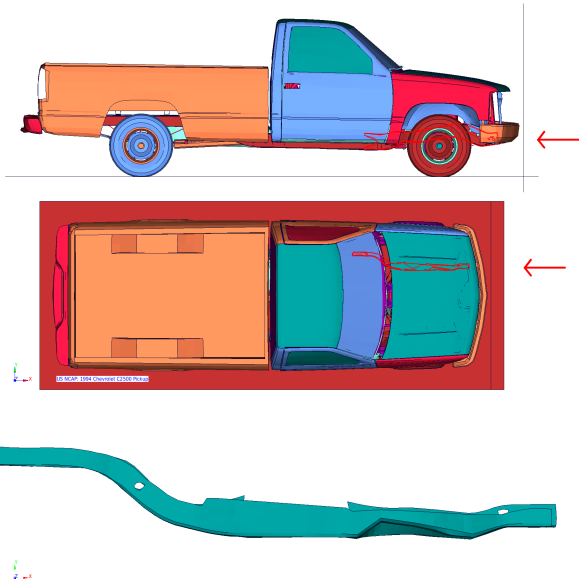

Abbildung 5.1: Darstellung des *TRUCK*-Datensatzes. Von oben nach unten: Seitenansicht des Modells (Position des ausgewählten Längsträgers hervorgehoben); Draufsicht des Modells (Position des ausgewählten Längsträgers hervorgehoben); Draufsicht des ausgewählten Längsträgers (*TRUCK-Beam*).

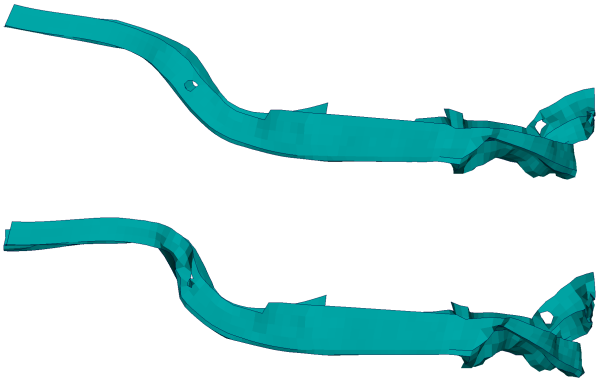

Abbildung 5.2: Verformung des Längsträgers beim Aufprall. Darstellung der beiden Modi der Bifurkation: Im ersten Bild bleibt im hinteren Bereich des Trägers die urspüngliche Krümmung weitgehend erhalten, im zweiten fällt die starke Ausbeulung (*buckling*) auf.

Um die Ergebnisse der Dimensionsreduktion visuell aufzubereiten, wurden für die folgenden Auswertungen drei- und zweidimensionale Einbettungen graphisch dargestellt. Die Einträge der Ausgabevektoren von Diffusion Maps werden als kartesische Koordinaten des $\mathbb{R}^3$ interpretiert. Der ursprüngliche Datenpunkt y_i wird also auf $(\lambda_2\phi_{2,i}, \lambda_3\phi_{3,i}, \lambda_4\phi_{4,i})$ abgebildet. $\phi_{i,j}$ steht hierbei für den j-ten Eintrag des i-ten Eigenvektors ϕ_i.
Da die Eigenwerte λ_i zum Teil sehr klein werden und nur skalierenden Einfluss auf die Einbettungen haben, wurde für die hier dargestellten Einbettungen auf die Skalierung mit ihnen verzichtet.
Neben der dreidimensionalen Einbettung umfassen die gezeigten Plots außerdem die zweidimensionale Einbettung entsprechend der Koordinaten $(\phi_{2,i}, \phi_{3,i})$, sowie zur Anschauung die Projektionen der 3D-Darstellung auf die anderen beiden Koordinatenebenen. Die Einfärbung der Datenpunkte erfolgte entsprechend der Werte der Koordinate zum zweiten Eigenvektor, da dieser theoretisch die meisten Informationen über die globale Struktur der Ursprungsmenge enthält [18].

In Abbildung 5.3 ist die durch die in *Kapitel 4* vorgestellte Methode erzeugte Einbettung in den $\mathbb{R}^3$ dargestellt[1]. Für Abbildung 5.4 wurde nur die durch erste und zweite Diffusionskoordinaten erzeugte zweidimensionale Einbettung betrachtet. Ausgewählte Punkte wurden durch die Visualisierungen der zugrundeliegenden Verformung dargestellt. Hier ist erkennbar, das beide Koordinatenachsen deutlich mit dem Crashverhalten des Trägers korrelieren: Der zweite Eigenvektor trennt anhand seines Vorzeichens klar die beiden Modi der Bifurkation und der Wert des dritten Eigenvektors verhält sich proportional zur Ausprägung der Verformung im mittleren Bereich des Trägers (erkennbar an der Änderung des Farbverlaufs in den Bildern des Bauteils): Die am linken Rand liegenden Modelle sind in der Crashberechnung die am wenigsten, die am rechten Rand liegenden die am stärksten verformten.
Zunächst lässt sich also festhalten, dass die gewählte zweidimensionale Einbettung bereits zwei wichtige Informationen über die Schar der Simulationsläufe enthält. Im Folgenden wird untersucht, inwiefern Variationen der Parameter und des gewählten Abstandsmaßes die beobachteten Korrelationen beeinflussen und ob sich eine ähnlich sinnvolle Einbettung auch bei komplexeren Datensätzen findet.

[1] Gewählte Parameter: $\gamma = 32$ und $n_{\text{bins}} = 10$; Details zu den Parametern folgen in Absätzen 5.1.1 und 5.1.2.

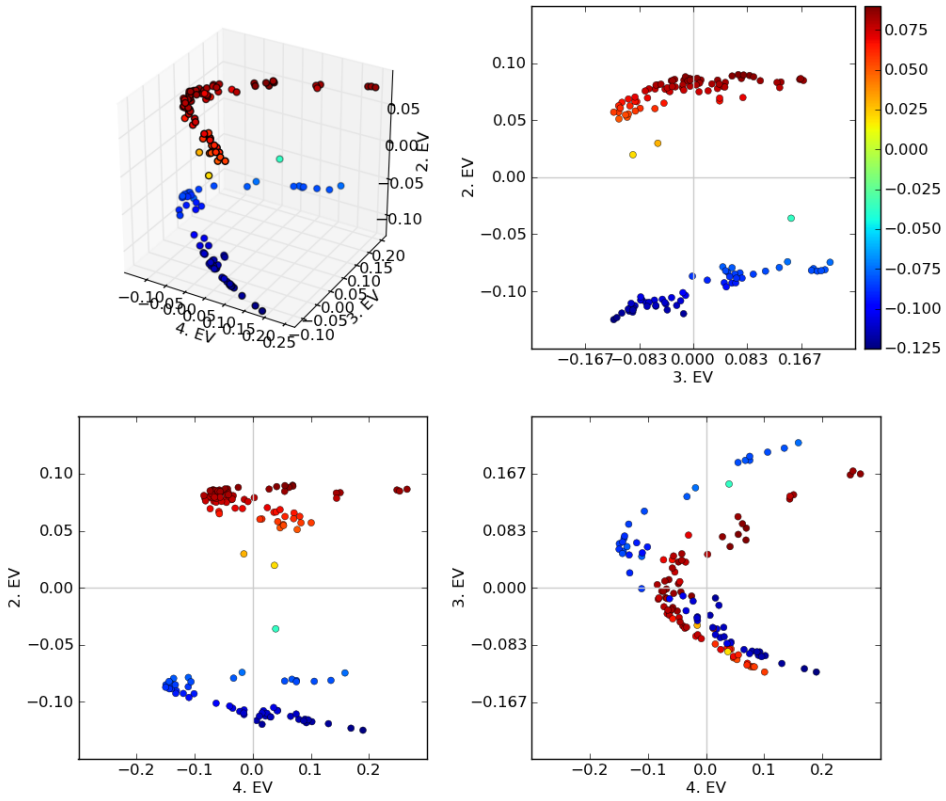

Abbildung 5.3: 3D-Darstellung der Einbettung des *TRUCK-Beam*-Datensatzes in den $\mathbb{R}^3$. Die Farben wurden entsprechend dem 2. Eigenvektor zugeordnet. (Parameter: $\gamma = 32$, $n_{\text{bins}} = 10$)

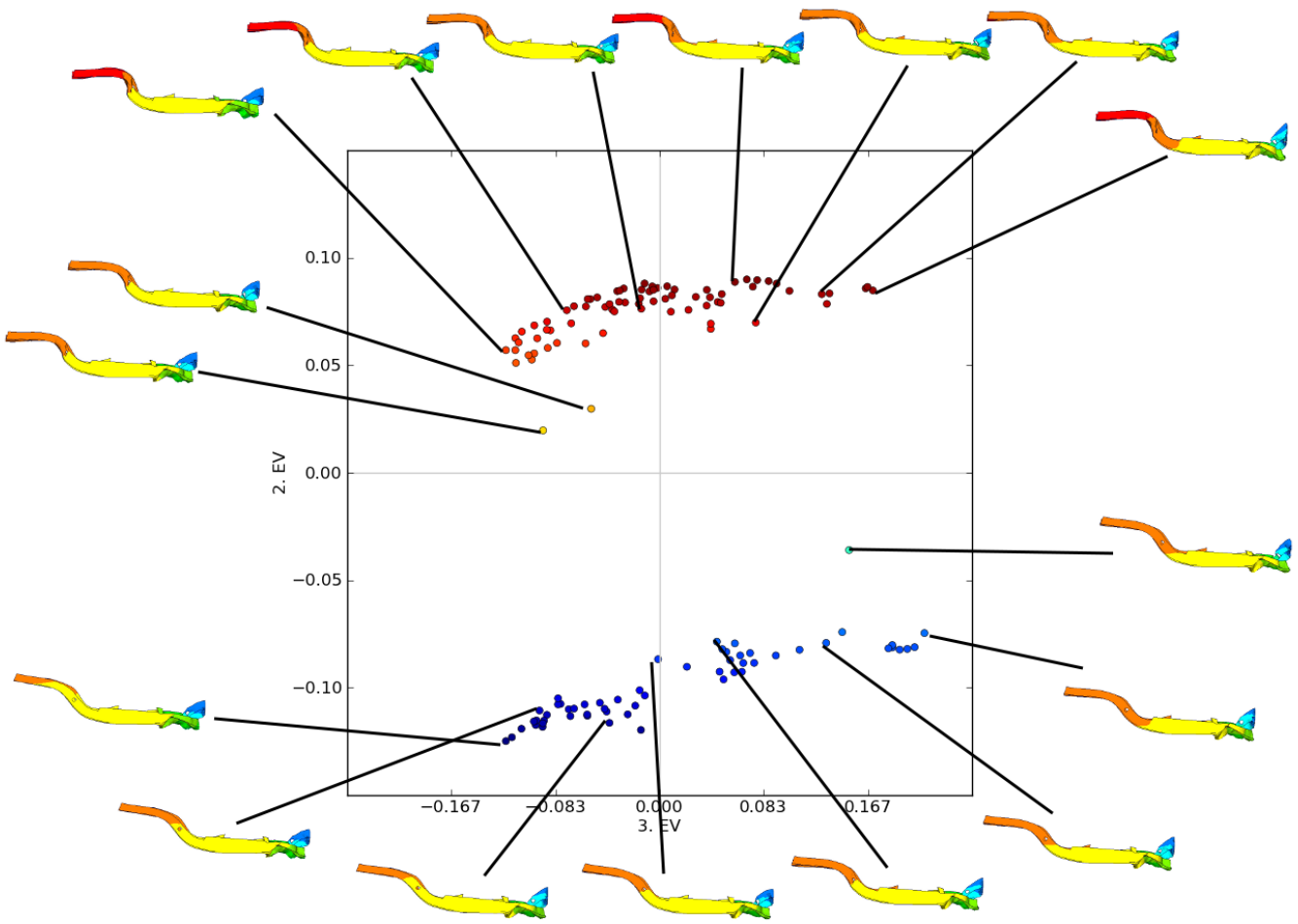

Abbildung 5.4: Die Einbettung in den $\mathbb{R}^2$ entsprechend zweitem und drittem Eigenvektor aus Abbildung 5.3 mit ausgewählten Visualisierungen der Simulationsläufe. Die beiden Bifurkationsmodi sind klar durch das Vorzeichen des zweiten Eigenvektors getrennt; die Achse des dritten Eigenvektors korreliert mit der gesamten Verformung des Bauteils, wie man an der Einfärbung der Modelle erkennt. (Parameter: $\gamma = 32$, $n_{\text{bins}} = 10$)

5.1 Einfluss unterschiedlicher Parameter

5.1.1 Wahl von ε für den Gauß-Kern

Der Gauß-Kern $k_\varepsilon(x, y) = \exp\left(-\frac{d(x,y)^2}{\varepsilon}\right)$ wird über einen Wert ε parametrisiert. Von diesem Parameter hängt die Größe der Nachbarschaft von x ab, in der Punkte y einen Wert $k_\varepsilon(x, y)$ von deutlich > 0 haben, vgl. Abbildung 5.5. Wann eine Wahl von ε für die konkrete Anwendung geeignet ist, lässt sich nur schwer quantifizieren. Generell sollten die entstehenden Nachbarschaften „weder zu groß noch zu klein sein“ [18].
Es haben sich einige Faustregeln zur Wahl der Größenordnung von ε etabliert, so zum Beispiel eine Orientierung an einer längsten Kante eines minimalen Spannbaums über dem Graphen der Daten oder am durchschnittlichen Abstand eines Punktes zu seinem $\log(n) + 1$-nächsten Nachbarn [18]. Lafon [15] hat für Diffusion Maps den durchschnittlichen Abstand eines Punktes zu seinem nächsten Nachbarn gewählt. Hieran orientiert sich auch diese Arbeit, unter zusätzlicher Skalierung mit einem Parameter γ:

$$\varepsilon := \frac{\gamma}{m} \sum_{x \in \mathcal{Y}} \min_{y \in \mathcal{Y}} \{\, d(x, y) \mid d(x, y) \neq 0 \,\}$$

Eine über Faustregeln hinausgehende, fundierte Theorie zur Größenordnung von ε gibt es bisher noch noch nicht [18], insbesondere für kleine Datensätze.
Im Irrfahrt-Modell von Diffusion Maps entspricht eine Verringerung von ε bzw. γ einer sinkenden Varianz: Die Wahrscheinlichkeit, dass die Irrfahrt einen bestimmten Nachbarschaftsbereich verlässt, schrumpft; Entfernungen zwischen unterschiedlichen Häufungsbereichen wachsen an. Zu große wie auch zu kleine Wahlen von γ führen also zu einer schlechteren Ausdifferenzierung der Einbettung; eine zu niedrige Wahl gibt zu viele getrennte Punkte zurück, eine zu große Wahl lässt alle Datenpunkte zu einer einzigen Häufung verschmelzen. Um den Einfluss von γ auf die Ergebnisse der Dimensionsreduktion zu analysieren, werden im Folgenden sowohl die Plots der Einbettungen in den $\mathbb{R}^3$ als auch die Werte der Eigenwerte und Eigenvektoren abhängig von γ betrachtet.

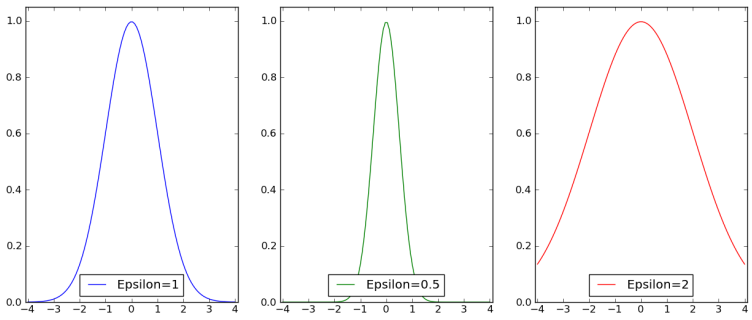

Abbildung 5.5: Plot von $k_\varepsilon(0, x)$ mit unterschiedlichen Werten für ε.

Einbettungen

Für den *TRUCK-Beam*-Datensatz ergeben sich für alle getesteten Wahlen von γ sinnvolle Einbettungen, wobei jedoch die Achsen zum Teil andere Funktionen übernahmen. Für $\gamma = 1$ zum Beispiel (Abbildung 5.7) kann man entlang des zweiten Eigenvektors neben der generellen Zweiteilung bezüglich der Bifurkation zusätzlich das Maß der gesamten Verformung ablesen. Bei Betrachtung des 2D-Plots bezüglich dritten und vierten Eigenvektors (Teilabbildung rechts unten in Abbildung 5.6) erkennt man außerdem, dass diese beiden Eigenvektoren jeweils nur Informationen über eine der beiden Teilmengen der Daten, die den Bifurkations-Modi entsprechen, enthalten. Diese Vektoren fungieren also auch als Indikator für jeweils einen Modus.
Mit steigendem γ ändert sich die Einbettung kontinuierlich, bis ab $\gamma = 16$ die Form der Punktwolke gleich bleibt und sich nur noch die Skala verändert. Plots für unterschiedliche γ von 2^0 bis 2^4 finden sich in Kapitel A.1 im Anhang (Abbildungen A.1 bis A.4).

Verhalten der Eigenwerte und Eigenvektoren

In Abbildung 5.8 ist dargestellt, wie sich das Spektrum von $\mathbf{K}$ abhängig von γ entwickelt. Mit Ausnahme von $\lambda_1 = 1$ sinken die Eigenwerte mit wachsendem γ; λ_2 und λ_3 etwas langsamer als die anderen. Dies ist nicht weiter überraschend, spielt doch der zweite Eigenwert einer Laplace-Matrix (*Fiedler-Wert*) eine wichtige Rolle in der Spektralen Graphentheorie als Maß des Zusammenhangs des Graphen [21].
Das Verhalten der Eigenvektoren, die als die Koordinaten der Einbettung fungieren, bei Variation von γ ist in Abbildung 5.9 dargestellt. Auf der x-Achse sind die Indizes der verwendeten Datenpunkte des Datensatzes *TRUCK-Beam* abgetragen, geplottet sind die jeweiligen Einträge der Eigenvektoren (also die Koordinaten der entsprechenden Dimension) für diesen Datenpunkt. Klar erkennbar ist insbesondere die fast völlige Invarianz des zweiten Eigenvektors bei Variation von γ.
Anschaulich wird dies auch bei Betrachten der Plots in Anhang A.1: Die Zweiteilung der Einbettung entsprechend der Bifurkation im Biegeverhalten des Trägers ist stets deutlich erkennbar. Dies lässt sich ebenfalls mit Ergebnissen aus der Spektralen Graphen-Theorie begründen: Seien $\{\bar{\lambda}_i\}_i$ die nach Größe sortierten Eigenwerte der nicht normalisierten Laplace-Matrix $\bar{\mathbf{L}} = \mathbf{I} - \mathbf{K}$. Der zu $\bar{\lambda}_2$ gehörige Eigenvektor von $\bar{\mathbf{L}}$, der sogenannte *Fiedler-Vektor*, wird verwendet, um anhand der Vorzeichen seiner Einträge eine Bipartitionierung des Graphen vorzunehmen, welche die Lösung eines *RatioCut*[2] approximiert [21].

[2] Gegeben ein Graph $G = (V, E)$ ist das Ziel die Bisektion von V in disjunkte U und W, sodass $\omega(U, W)/|U||W|$ minimiert wird, wobei $\omega(U, W)$ für das Gesamtgewicht der Kanten zwischen U und W steht.

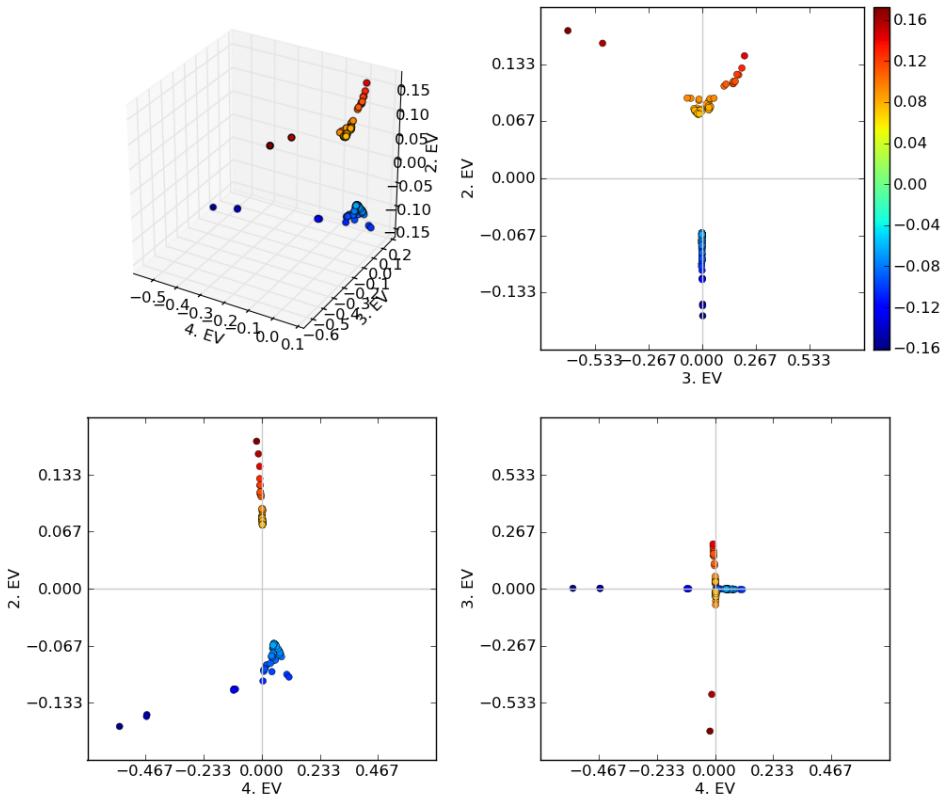

Abbildung 5.6: 3D-Darstellung der Einbettung in den $\mathbb{R}^3$ und Projektionen auf die von jeweils zwei Eigenvektoren aufgespannten Ebenen, $\gamma = 1$.

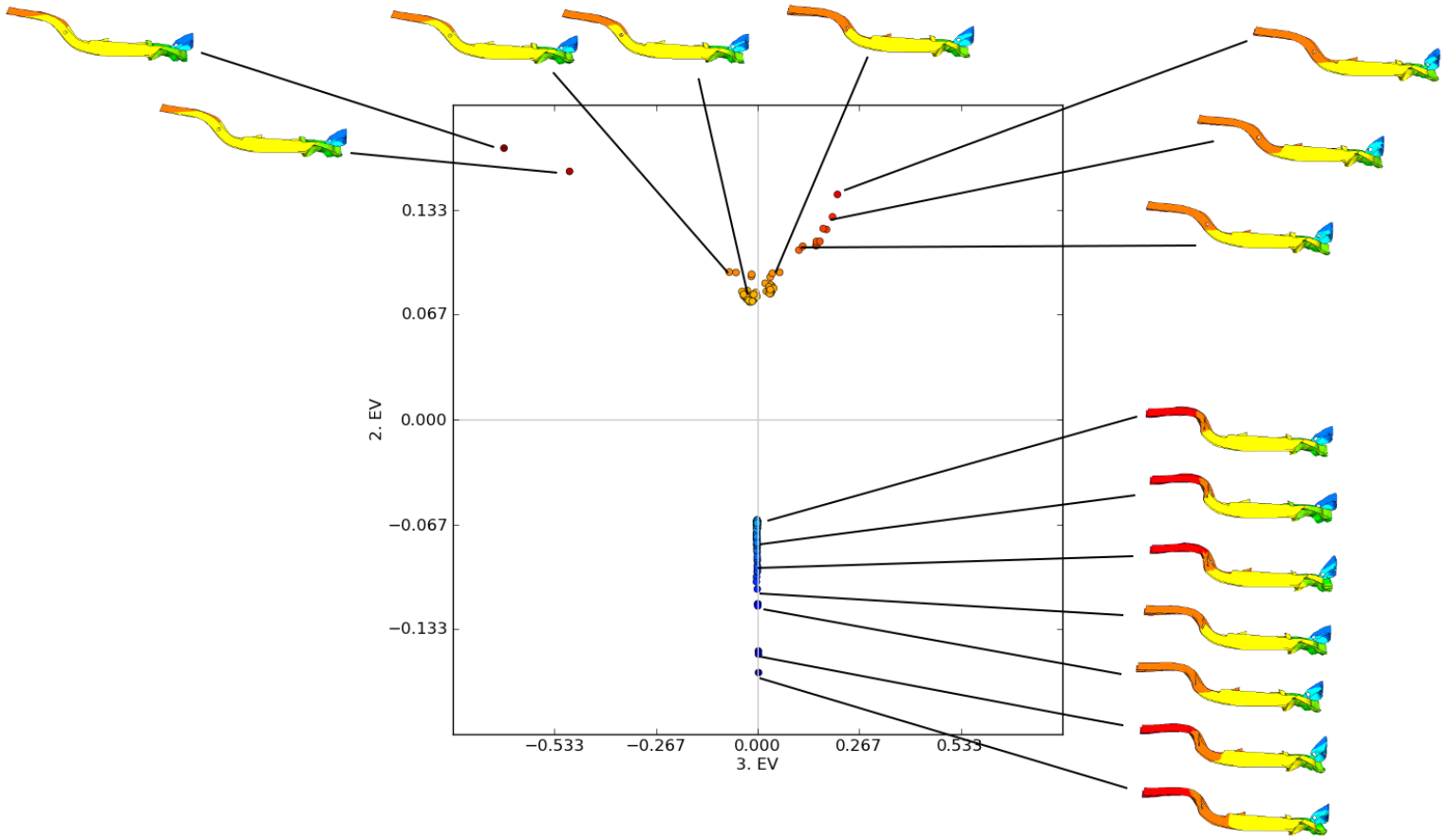

Abbildung 5.7: Die Einbettung in den $\mathbb{R}^2$ entsprechend zweitem und drittem Eigenvektor aus Abbildung 5.6 mit ausgewählten Visualisierungen der Simulationsläufe. Die beiden Bifurkationsmodi sind klar durch das Vorzeichen des zweiten Eigenvektors getrennt. (Parameter: $\gamma = 1$, $n_{\text{bins}} = 10$)

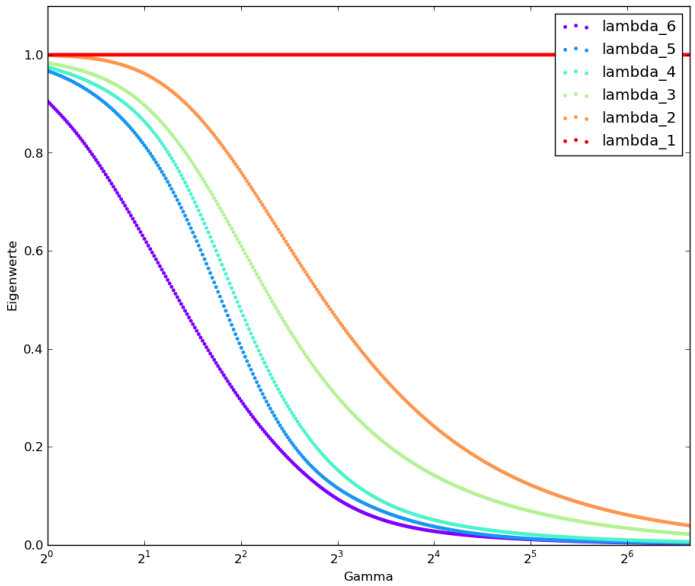

Abbildung 5.8: Verlauf der Eigenwerte λ_1 bis λ_6 von $\mathbf{K}$ bei Analyse des *TRUCK-Beam*-Datensatzes abhängig von der Wahl eines γ. (Parameter: $n_{\text{bins}} = 10$)

Auch der dritte und vierte Eigenvektor verhalten sich bei variablem γ sehr ähnlich, wobei zu erkennen ist, dass sich für niedrige γ die Werte nur für jeweils einen der beiden Cluster signifikant von 0 unterscheiden. Derartiges Verhalten der Eigenvektoren ist ausführlich im Rahmen von *Spectral Clustering*-Methoden (siehe auch Absatz 3.3) untersucht worden: Bei klar geteilten Datensätzen (oder in Bezug auf die zugehörigen Graphen: bei Graphen aus mehreren Zusammenhangskomponenten) können die Vorzeichen der relevanten Eigenvektoren als Indikator für die Gruppierung dienen [18].
Für den fünften Eigenvektor ist kein Zusammenhang zu den Daten mehr zu erkennen, er reagiert deutlich empfindlicher auf die Änderungen von γ; der sechste nimmt für alle γ keine signifikanten Werte mehr an. Aus der Theorie zum Spectral Clustering ist bekannt, dass die kleineren Eigenwerten zugehörigen Eigenvektoren eher lokale als globale Tendenzen einfangen und sich daher sensibler gegenüber kleinen Störungen oder Änderungen verhalten.

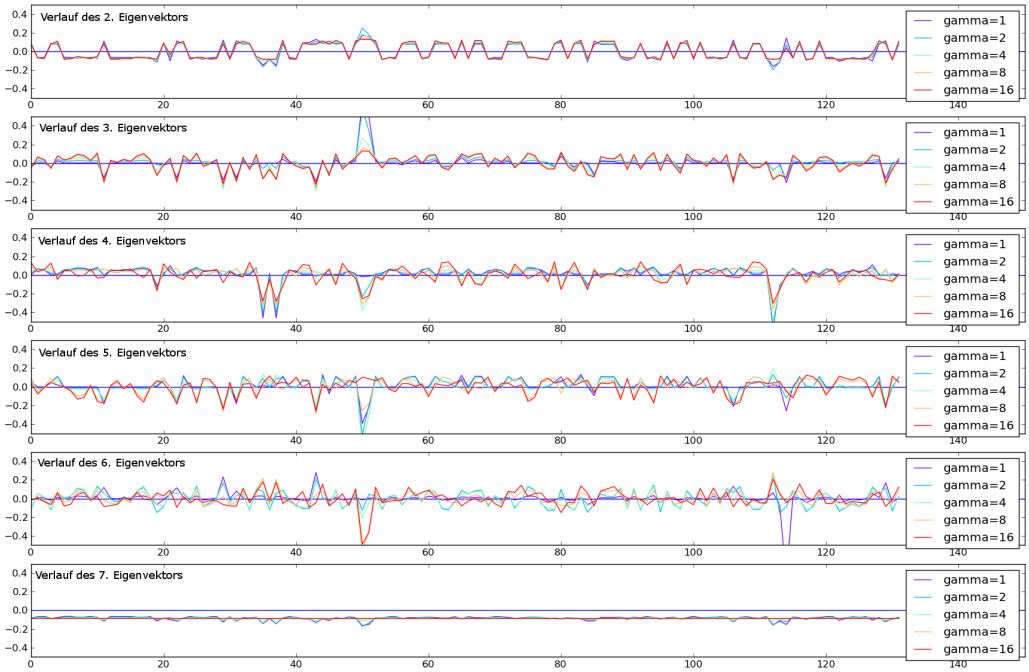

Abbildung 5.9: Werte des 2. bis 7. Eigenvektors bei Anwendung von Diffusion Maps und EMD (Histogramme mit $n_{\text{bins}} = 10$) für variable γ, geplottet gegen die Datenpunkte. Diese Werte stellen die Koordinaten der jeweiligen Datenpunkte für die Einbettung dar. Bemerkenswert ist insbesondere die Stabilität des zweiten Eigenvektors.

5.1.2 Anzahl der Klassen beim Histogramm, n_{bins}

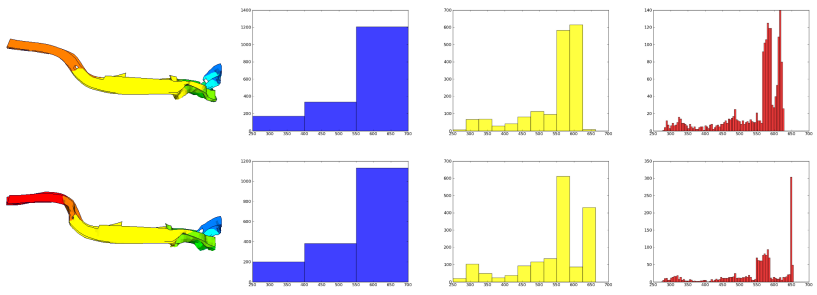

Abbildung 5.10: Zwei unterschiedliche Simulationsdurchläufe des *TRUCK-Beam*-Datensatzes mit den zugeordneten Histogrammen. Von links nach rechts: 3, 12 und 100 Klassen. $n_{\text{bins}} = 12$ ist laut Sturges-Regel optimal.

Die Auswahl der Histogramme beeinflusst sowohl die Laufzeit des Algorithmus als auch die resultierende Einbettung stark, da sie die Ausgangskonfiguration für die Abstandsberechnung mit Histogrammmetriken liefert. Wie bereits in Abschnitt 3.4 erwähnt, ist bei der Darstellung von Daten durch Histogramme die Wahl der Klassenanzahl essentiell, da sowohl zu kleine als auch zu große Klassen die zugrunde liegenden Strukturen nicht mehr angemessen widerspiegeln, vgl. auch Abbildung 5.10. Die Wahl von $n_{\text{bins}} = 12$ für den *TRUCK-Beam*-Datensatz entsprechend der Sturges-Regel (Gleichung 3.2) ergab sehr gute Ergebnisse für alle getesteten Datensätze.[3] Bei weniger Klassen traten Abweichungen von der Struktur auf und die eingebetteten Daten waren unpassend angeordnet, für die beispielhafte Wahl von $n_{\text{bins}} = 5$ ist die Einbettung in Abbildung A.5 in Anhang A abgebildet. Bei mehr Klassen hängen die Effekte von der verwendeten Metrik ab.
Durch die Verwendung von EMD als Histogrammmetrik können die Effekte durch zu hohe Varianz bei zu großer Klassenanzahl recht gut beschränkt werden, da auch benachbarte Klassen in die Abstandsberechnung einfließen und durch den Grundabstand auch passend gewichtet werden. Trotzdem beeinflusst auch hier eine steigende Klassenanzahl das Ergebnis negativ. Generelle Strukturen und Tendenzen in der Einbettung bleiben zwar sehr lange erhalten, so zum Beispiel die Anordnung entsprechend der Gesamtverformung, die Bifurkation ist jedoch bereits ab $n_{\text{bins}} = 15$ nicht mehr klar erkennbar. Entsprechende Plots sind in Kapitel A.2 des Anhangs zu betrachten.
Bei Analyse der Eigenvektoren abhängig von γ, wie im vorigen Absatz für $n_{\text{bins}} = 10$ durchgeführt, ergaben sich etwas schlechtere Ergebnisse bezüglich der Stabilität. Für $n_{\text{bins}} = 30$, vgl. Abbildung 5.11, änderte sich für einige wenige Datenpunkte das Vorzeichen des zweiten Eigenvektors abhängig von der Wahl eines γ, was sich beim betrachteten Datensatz in der schlechteren Einteilung der Bifurkation deutlich macht. Auch der dritte Eigenvektor weist im Vergleich weniger Stabilität auf.

[3] Die Einbettungen für $n_{\text{bins}} = 12$ gleichen den in bisher in diesem Kapitel besprochenen Ergebnissen für $n_{\text{bins}} = 10$.

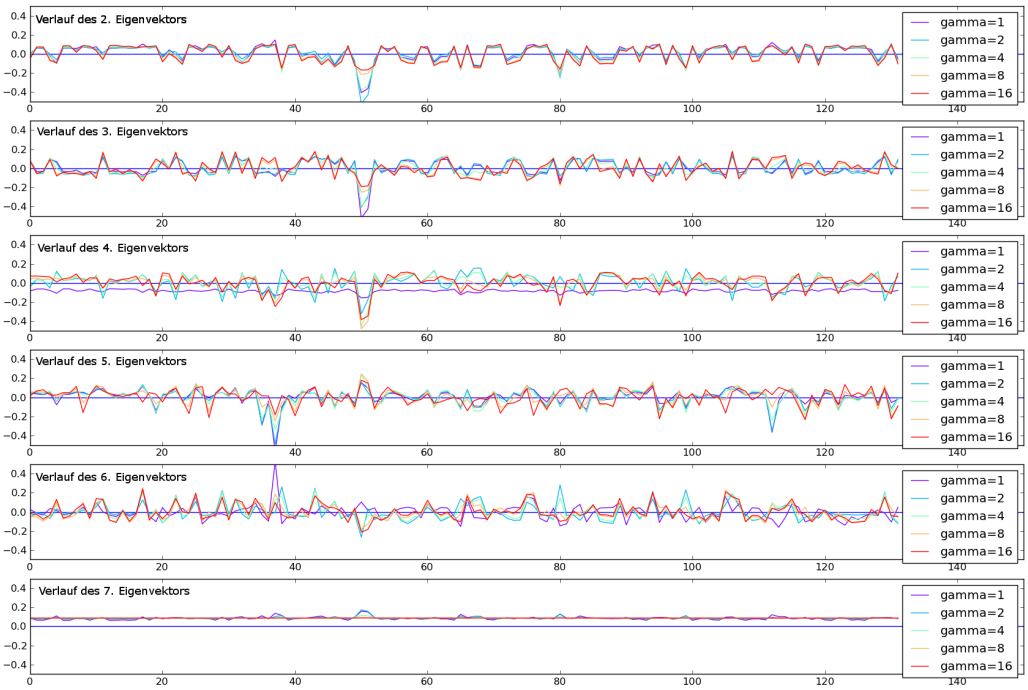

Abbildung 5.11: Werte des 2. bis 7. Eigenvektors bei Anwendung von Diffusion Maps und EMD (Histogramme mit $n_{\text{bins}} = 30$) für variable γ, geplottet gegen die Datenpunkte. Diese Werte stellen die Koordinaten der jeweiligen Datenpunkte für die Einbettung dar. Im Vergleich zu Abbildung 5.9 fällt auf, dass die Werte der Eigenvektoren abhängig von γ weniger stabil verlaufen als für $n_{\text{bins}} = 10$.

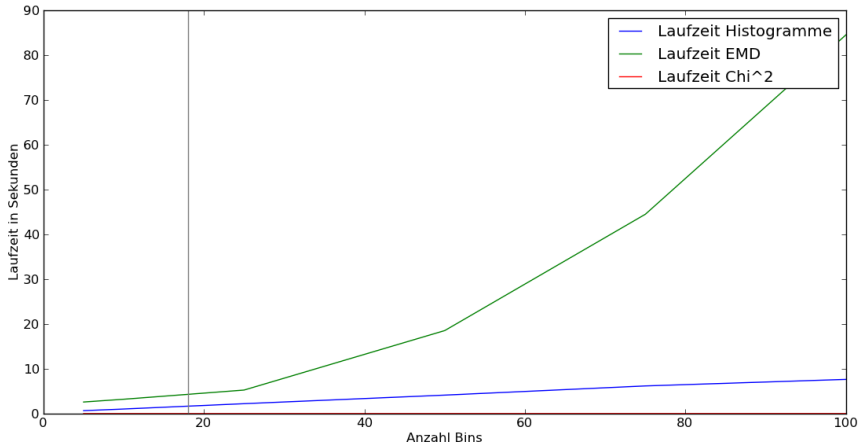

Abbildung 5.12: Praktische Messungen der Laufzeit abhängig von der Anzahl an Klassen: Subroutine „Histogramme erstellen“ und Subroutine „EMD berechnen“. Im Vergleich zu EMD ist außerdem die Berechnung des in Abschnitt 5.3.2 genauer betrachteten χ^2-Abstands dargestellt. Die Berechnung von Diffusion Maps ist aufgrund der niedrigen Dimension der Laplace-Matrix und vorhandener schneller Eigenwertlöser hier stets sehr schnell ($< 0,02$ Sekunden) und daher nicht dargestellt. Die vertikale Markierung wurde bei $n_{\text{bins}} = 16$, der mit der Sturges-Regel gewählten Binanzahl, gesetzt.

Neben den Ergebnissen wird bei steigender Klassenanzahl auch die Laufzeit beeinträchtigt, da deutlich mehr paarweise Abstände analysiert werden müssen. EMD beruht auf der Lösung eines Optimierungsproblems (vgl. Abschnitt 3.5.1), was polynomiell (beziehungsweise hier insbesondere: nicht linear) von n_{bins} abhängige Laufzeiten bedeutet. Die Ergebnisse einer praktischen Laufzeitanalyse für die beiden Subroutinen „Erstellung der Histogramme“ und „EMD bestimmen“ sind in Abbildung 5.12 dargestellt. Das Erstellen der Histogramme hängt linear von der Anzahl der Klassen ab, die Berechnung von EMD hingegen polynomiell. Für dieses Laufzeitdiagramm wurde der *PKW-Seite*-Datensatz verwendet, welcher ungefähr 26.000 FE-Knoten umfasst. Eine Anzahl von 16 Klassen wäre laut Sturges-Regel optimal. Damit weist das Verfahren in der praktischen Anwendung vertretbar kurze Laufzeiten auf.

Bei der Verwendung von Bin-to-Bin-Histogrammabständen (vgl. Abschnitt 3.5) treten die Unregelmäßigkeiten bereits etwas früher auf, jedoch bei deutlich niedrigeren Laufzeiten. Einige Details werden in Absatz 5.3.2 dargestellt.

5.2 Unterschiedliche Datensätze

Auch für den deutlich komplexeren *PKW-Seite*-Datensatz ergaben sich aussagekräftige Einbettungen. Ein Plot ist in Abbildung 5.13 zu sehen. Eine detailliertere Darstellung mit Visualisierungen ist leider in dieser Arbeit nicht möglich, da es sich um vertrauliche Industriedaten handelt. Erkennbar ist die Bildung von vier

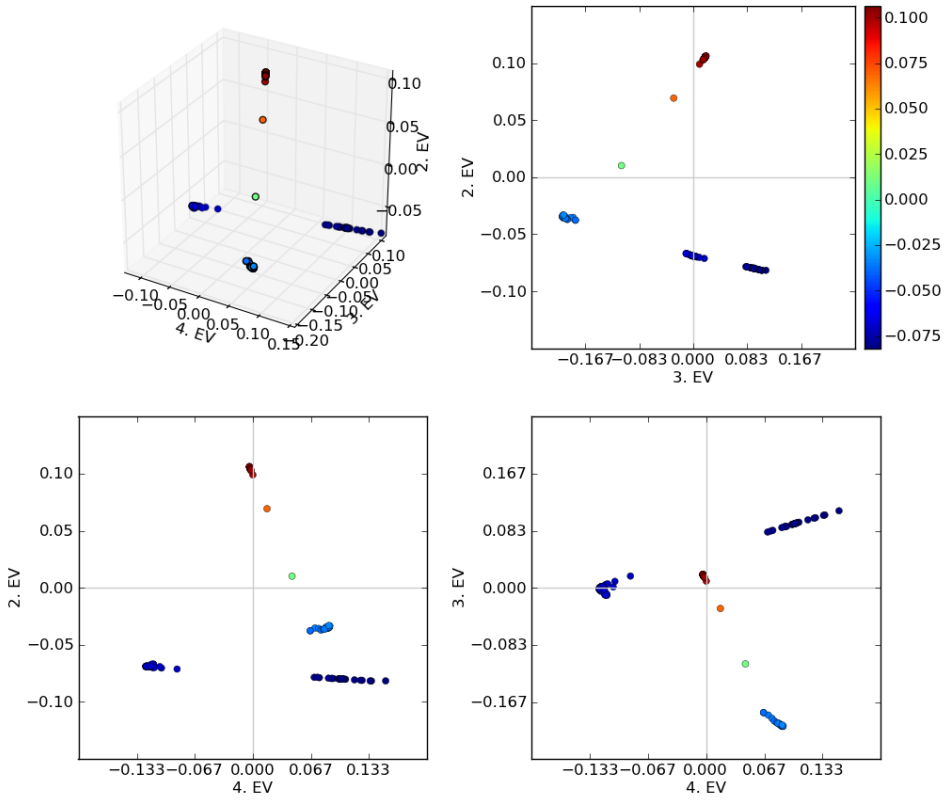

Abbildung 5.13: 3D-Darstellung der Einbettung des *PKW-Seite*-Datensatzes in den $\mathbb{R}^3$ mit Diffusion Maps und EMD, Parameter: $n_{\text{bins}} = 16$, $\gamma = 32$. Der Verlauf des zweiten Eigenvektors korreliert mit den Verformungen im Modell.

größeren Häufungen und zwei Ausreißern (hellgrün und orange). Vor allem diese Ausreißer sowie die Unterteilung in zwei Gruppen (entsprechend der Farben rot und blau) decken sich mit den Ergebnissen der Simulationsläufe. Die weitere Unterteilung der größeren Gruppe ist im Vergleich mit den Visualisierungen der Simulationsdaten nicht völlig offensichtlich, insbesondere zwischen den Ergebnissen der beiden dunkelblauen Gruppen ist optisch kein Unterschied erkennbar. Der zweite Eigenvektor korreliert auch hier mit dem Ausmaß der Verformungen. Je größer der Wert des zweiten Eigenvektors, desto deutlichere Verformungen sind in den Auswertungen erkennbar.
Insgesamt bewährt sich das vorgeschlagene Verfahren also auch für diesen Datensatz – welcher mit Diffusion Maps und euklidischem Abstand aufgrund der variablen Geometrie bisher nur umständlich analysiert werden konnte.
Auch hier wurden Versuche zum Einfluss der Parameter γ und n_{bins} durchgeführt. Der betrachtete Datensatz ist mit EMD als Abstandsmaß sehr stabil, was die Anzahl der Klassen angeht. Bei geeigneten Werten für γ ergab sich eine bis auf Skalierung identische Einbettung für alle untersuchten Werte zwischen 4 und 200. Gleichzeitig reagiert der Datensatz allerdings etwas sensibler auf niedrige Wahlen von γ. Bei $\gamma < 4$ konvergiert die numerische Eigenwertbestimmung nicht, und bei Werten von $\gamma < 8$ sind noch keine Korrelationen der Einbettung zu den tatsächlichen Simulationsergebnissen erkennbar. Dies liegt an der Struktur des Datensatzes: die Unterschiede zwischen den einzelnen Datenpunkten sind so groß, dass die mittels Gaußkern berechneten Ähnlichkeiten bei niedrigem γ für fast alle Paare numerisch 0 betragen.

5.3 Vergleich mit anderen Abstandsfunktionen

5.3.1 Euklidischer Abstand

Da beim *TRUCK-Beam*-Datensatz die Geometrie invariant ist, umfassen die Verschiebungsvektoren der Simulationsläufe insbesondere die gleiche Anzahl von FE-Knoten und können daher direkt mit dem euklidischen Abstand anstelle eines Histogrammabstandes verglichen werden. Somit kann für diesen Datensatz auch das ursprüngliche Diffusion Maps-Verfahren verwendet werden, wie zum Beispiel in [12].
Eine Betrachtung der Einbettung zeigt, dass hier die gleichen Korrelationen erkennbar sind, wie bei Verwendung der Histogrammmetrik: Die berechneten Diffusionskoordinaten bilden sowohl den Modus der Bifurkation als auch einen Verlauf der Gesamtverformung ab. Hier verläuft allerdings die Abgrenzung der beiden Modi „stark deformiert“ und „leicht deformiert“ weniger scharf als bei Histogrammabständen (für eine geeignete Wahl von $n_{\text{bins}} \approx 10$), vgl. Abbildungen 5.14 und 5.15. Die Aufspaltung der Bifurkationsmodi verläuft auch nicht mehr entsprechend des Vorzeichens des zweiten Eigenvektors: Die beiden hellgrünen Datenpunkte in Abbildung 5.15 sind durch den zweiten Eigenvektor in die unpassende Gruppe sortiert worden.

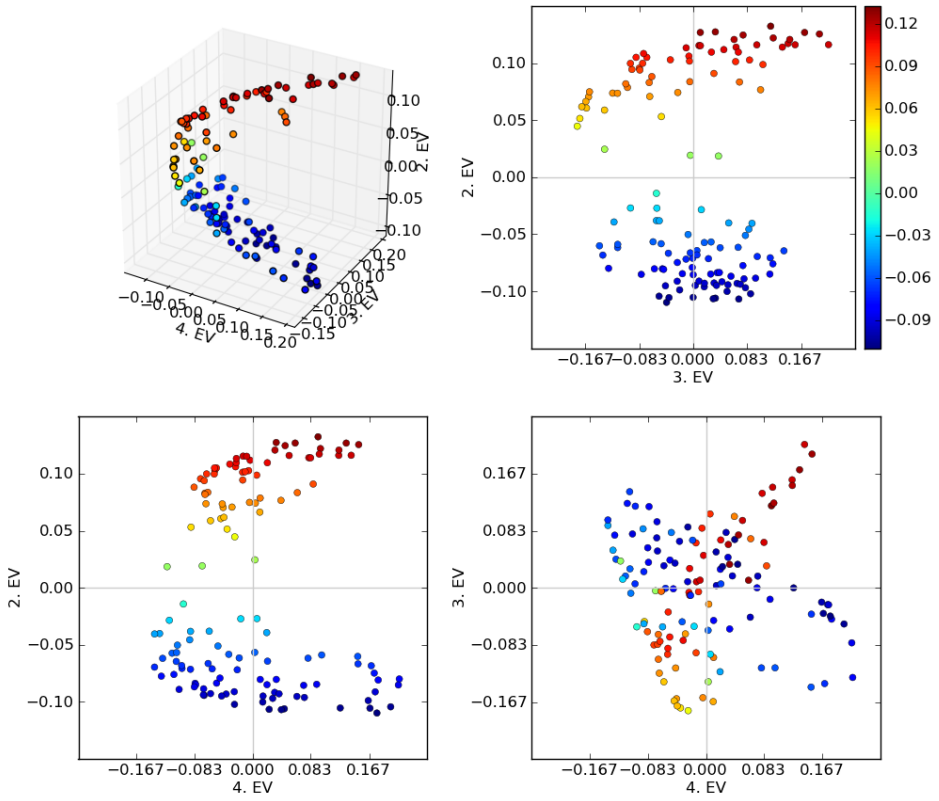

Abbildung 5.14: 3D-Darstellung der Einbettung des Datensatzes *TRUCK-Beam* in den $\mathbb{R}^3$ mittels Diffusion Maps und euklidischem Abstand. (Parameter: $\gamma = 64$, andere Wahlen verhielten sich analog)

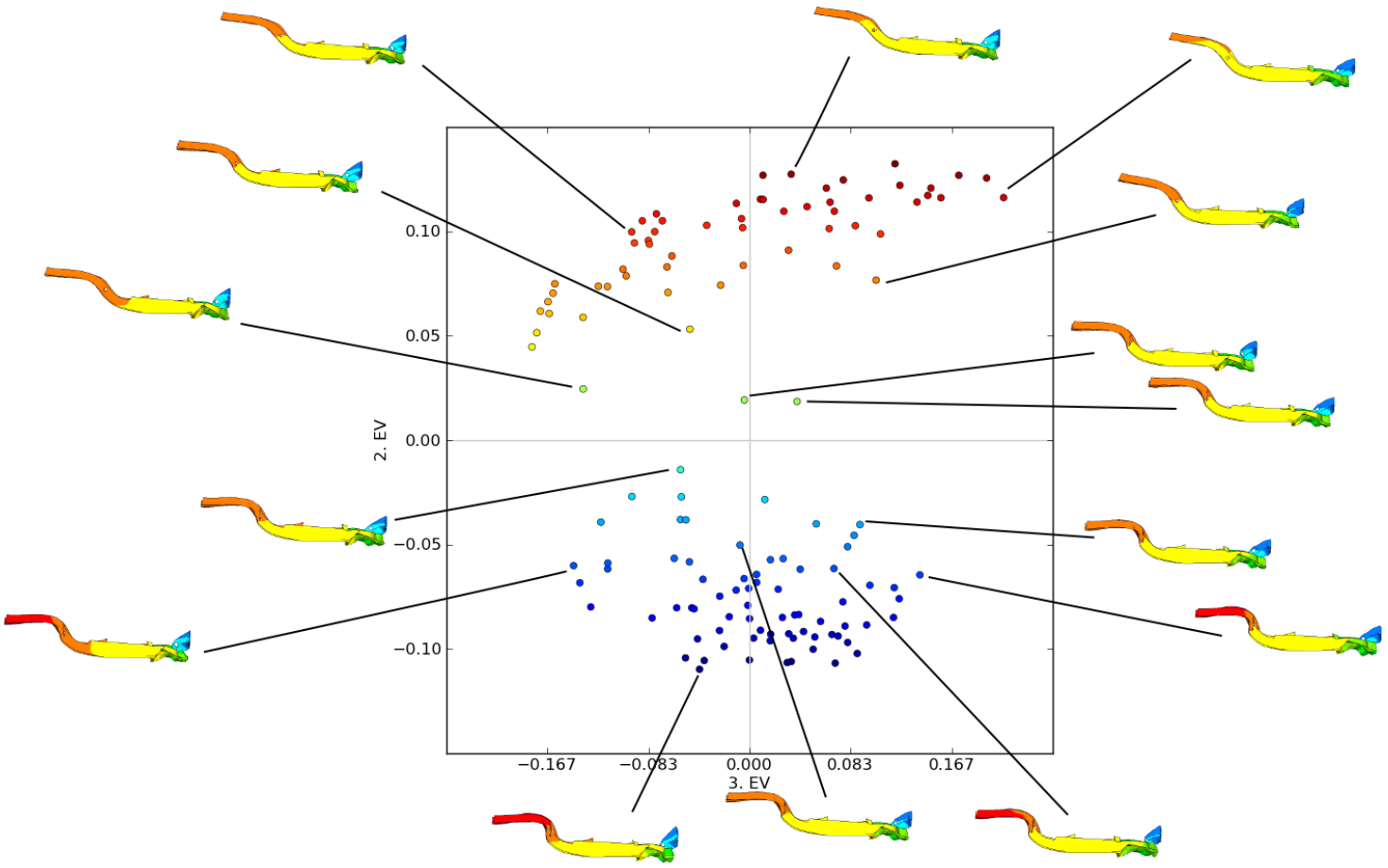

Abbildung 5.15: Die Einbettung in den $\mathbb{R}^2$ mit Diffusion Maps und euklidischem Abstand im Detail. Der zweite Eigenvektor korreliert zwar noch mit der Bifurkation, die Trennung ist aber weniger deutlich als zuvor; die beiden Datenpunkte in der Mitte (hellgrün) wären passender auf die negative Halbebene (starkes Ausbeulen) sortiert worden. (Parameter: $\gamma = 64$, andere Wahlen verhielten sich analog)

Dieser Unterschied wurde auch bei Variationen von γ bestätigt und ist sehr überraschend, da damit die Verwendung von Histogrammen die Resultate im anfangs definierten Sinne sogar verbessert – die Zweiteilung ist bei der eigentlich ungenaueren Methode viel besser zu erkennen und die anderen Maßstäbe werden bei beiden Verfahren gleich gut erreicht.

5.3.2 Histogrammabstände

Als weiterer Histogrammabstand wurde der Bin-to-Bin-Abstand χ^2 gewählt, da dieser im Bereich der digitalen Bildverarbeitung gute Ergebnisse erzielt. Bei $n_{\text{bins}} = 10$ und für alle getesteten Wahlen von γ ergaben sich, ähnlich wie für EMD, sehr gut zu den Daten passende Einbettungen. Dies ist auch in den Abbildungen 5.16 und 5.17 ersichtlich.
Der χ^2-Abstand zeigte sich im Vergleich mit EMD etwas sensibler in Bezug auf die Klassengrößen. Für 12 Klassen sind zwar die gleichen Strukturen erkennbar, die Form der Punktwolke ist jedoch weniger definiert. Die Darstellung der entsprechenden Einbettung findet sich in Abbildung A.10 im Anhang. Bereits ab einer Anzahl von 13 Klassen wird die Abgrenzung bezüglich der Bifurkation weniger scharf, wie für $n_{\text{bins}} = 50$ in Abbildungen 5.18 und 5.19 erkennbar. Dies war mit der Argumentation aus Abschnitt 5.1.2 zu erwarten, da χ^2 als Bin-to-Bin-Abstand empfindlich auf eine größere Varianz bei den Klassen reagiert. Verglichen mit dem eigentlich deutlich robusteren EMD ist der Unterschied jedoch unerwartet klein. Auch bei großen Klassenzahlen bleibt die generelle Struktur der Anordnung den Daten angemessen, die Anwendung von χ^2 liefert hier vergleichbare Einbettungen wie der Einsatz des euklidischen Abstands, was man bei Vergleich der Abbildungen 5.18 und 5.14 sowie 5.19 und 5.15 feststellen kann. Auch beim Test auf dem komplexeren *PKW-Seite*-Datensatz ergaben sich für den χ^2-Abstand vergleichbar gute Ergebnisse wie für die Verwendung von Diffusion Maps mit EMD. Die gefundenen Einbettungen weichen nur sehr wenig von den in Abschnitt 5.2 gezeigten ab. Dies ist auch bei Betrachtung von Abbildung A.11 im Anhang feststellbar.

In die praktische Laufzeitanalyse in Abbildung 5.12 wurde auch die Berechnung des χ^2-Abstands aufgenommen. Für keine der im Rahmen dieser Arbeit betrachteten Klassenanzahlen war diese Berechnung langsamer als $0,01$ Sekunden. Damit ist χ^2 bedeutend schneller als EMD, bei vergleichbaren Ergebnissen.

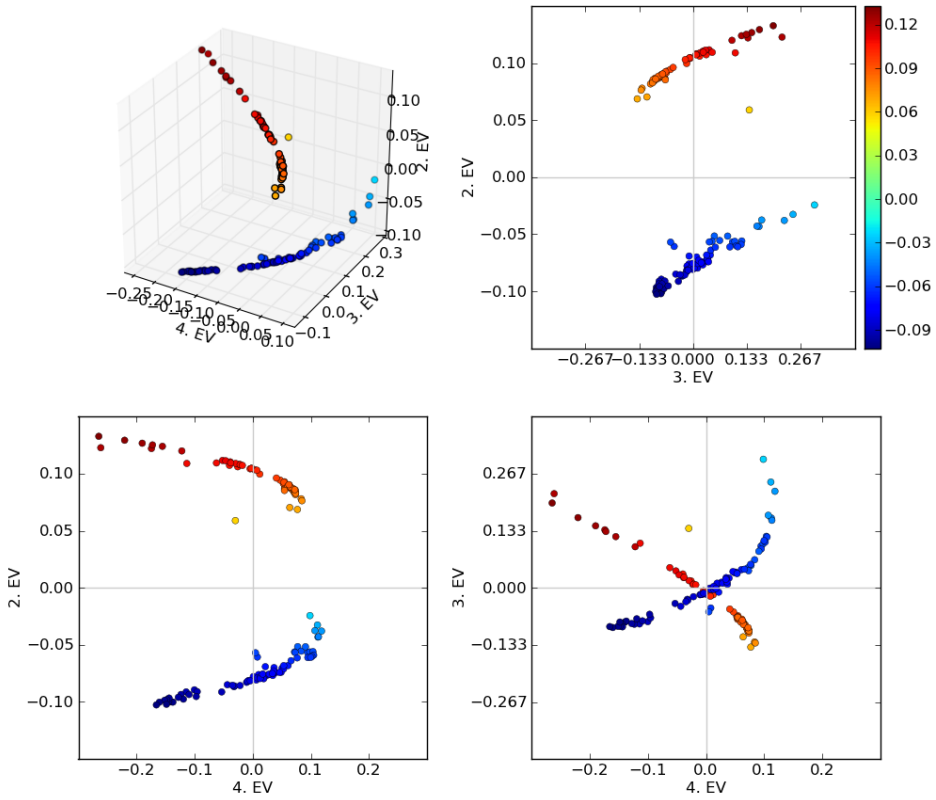

Abbildung 5.16: 3D-Darstellung der Einbettung des Datensatzes *TRUCK-Beam* in den $\mathbb{R}^3$ mittels Diffusion Maps und χ^2-Histogrammabstand bei $n_{\text{bins}} = 10$ Klassen. Wie bei Verwendung von EMD ist die Bifurkation klar zu erkennen und auch die Achsen verhalten sich ähnlich. (Weiterer Parameter: $\gamma = 32$, andere Wahlen verhielten sich analog)

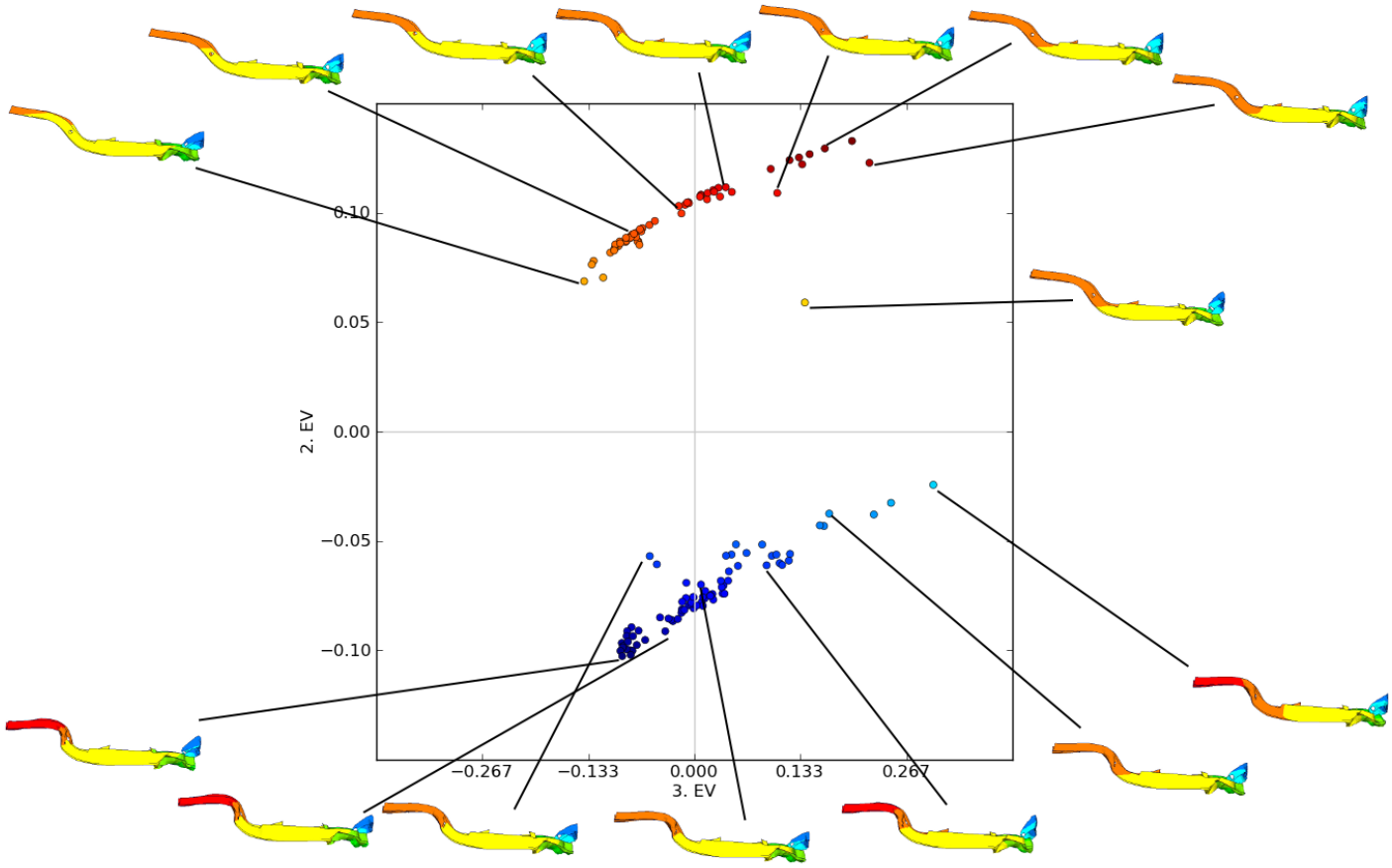

Abbildung 5.17: Die Einbettung des Datensatzes *TRUCK-Beam* mit Diffusion Maps und χ^2-Histogrammabstand bei $n_{\text{bins}} = 10$ Klassen in den $\mathbb{R}^2$ im Detail. Wie bei Abbildung 5.4 ist die Anordnung der Punkte sehr passend und aussagekräftig: Die Bifurkationsmodi werden durch das Vorzeichen des zweiten Eigenvektors strikt getrennt, die Koordinate des dritten Eigenvektors entspricht der gesamten Verformung. (Parameter: $\gamma = 32$, andere Wahlen verhielten sich analog)

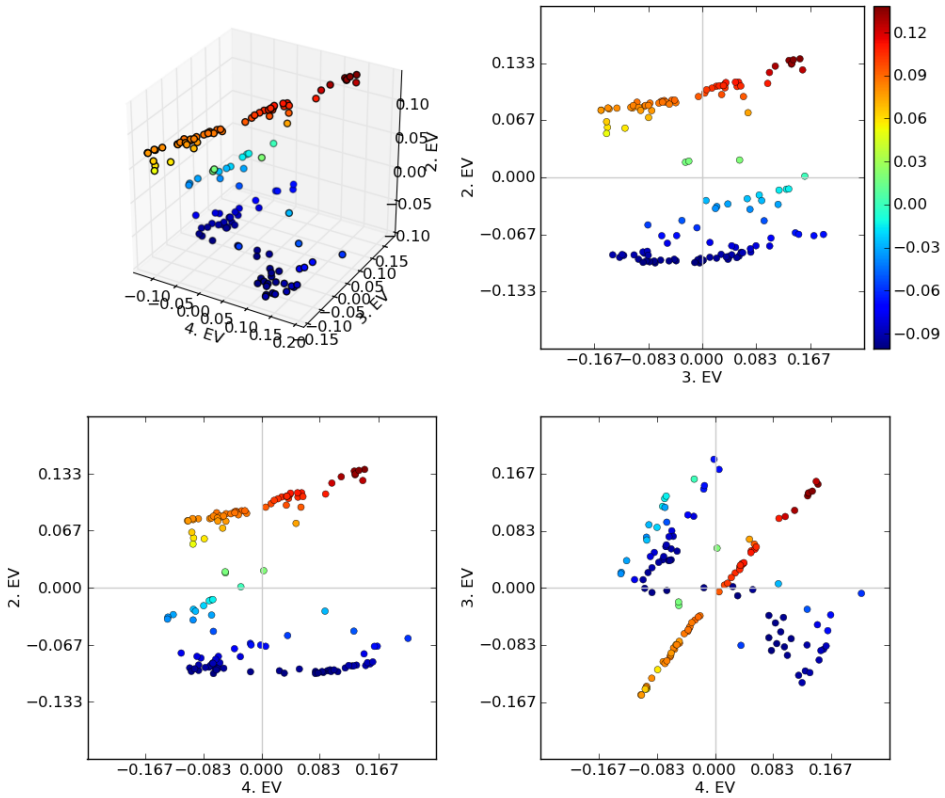

Abbildung 5.18: 3D-Darstellung der Einbettung des Datensatzes *TRUCK-Beam* in den $\mathbb{R}^3$ mittels Diffusion Maps und χ^2-Histogrammabstand bei $n_{\text{bins}} = 50$ Klassen. Interessant ist insbesondere die Ähnlichkeit zur Einbettung bei Diffusion Maps mit euklidischem Abstand, Abbildung 5.14. (Weiterer Parameter: $\gamma = 32$, andere Wahlen verhielten sich ähnlich)

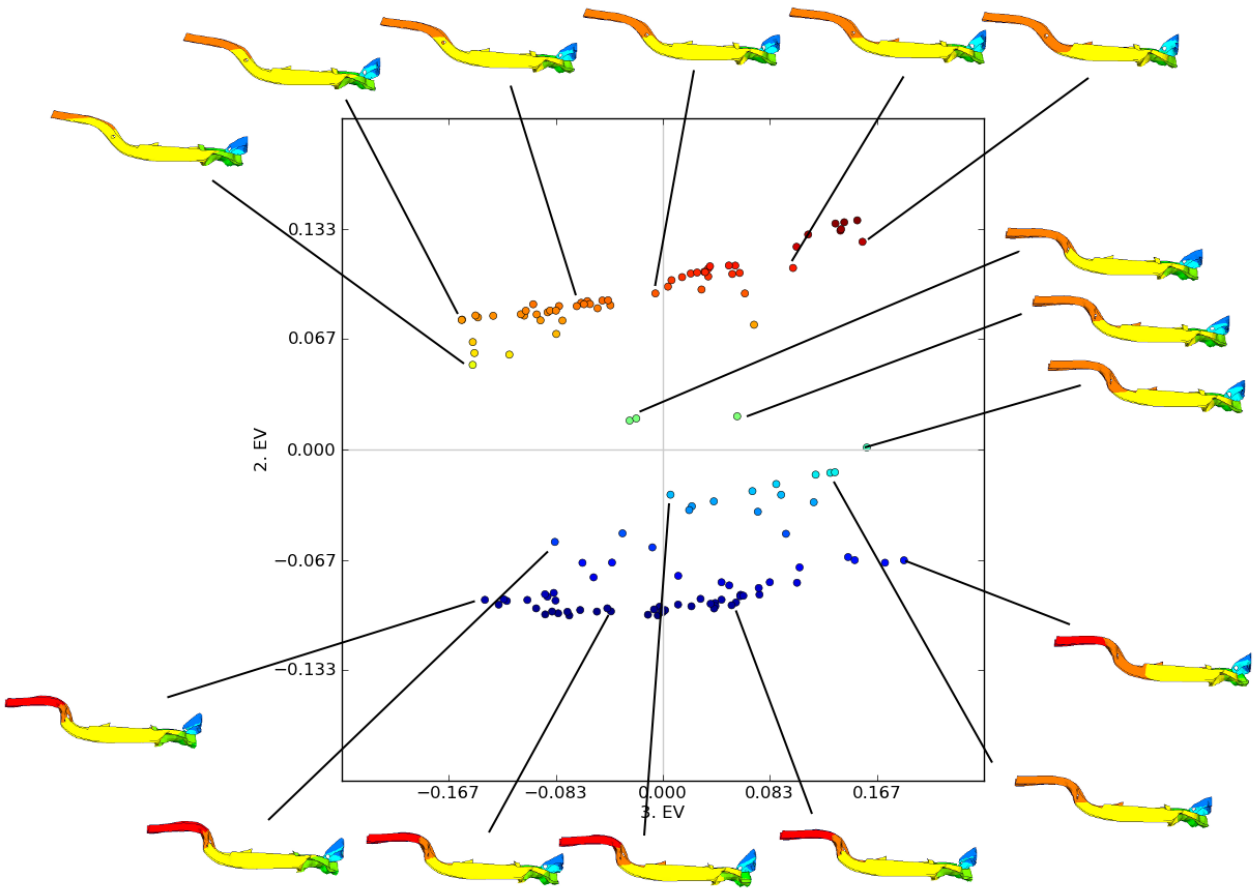

Abbildung 5.19: Die Einbettung des Datensatzes *TRUCK-Beam* mit Diffusion Maps und χ^2-Histogrammabstand bei $n_{\text{bins}} = 50$ Klassen in den $\mathbb{R}^2$ im Detail. Auf dieser Ebene ist das Verhalten nahezu analog zu Abbildung 5.15: Der zweite Eigenvektor korreliert zwar noch mit der Bifurkation, die Trennung ist aber weniger deutlich als bei $n_{\text{bins}} = 10$; die Datenpunkte in der Mitte (hellgrün) wären passender auf die negative Halbebene (starkes Ausbeulen) sortiert worden. (Parameter: $\gamma = 32$, andere Wahlen verhielten sich analog)

5.4 Diskussion der Ergebnisse

Die Ergebnisse der Experimente zur Verwendung von Diffusion Maps mit Histogrammabständen sind erstaunlich gut. Die Bifurkation im überprüften *TRUCK-Beam*-Datensatz wird erkannt und scharf abgegrenzt und die zweidimensionale Einbettung weist entlang der Achsen auffällige Korrelationen mit dem gesamten Verformungsverhalten des Trägers auf. Damit übertrifft EMD für dieses Beispiel sogar den detaillierter bestimmten euklidischen Abstand. Selbst bei ungünstiger Klassengröße ist auch der χ^2-Abstand nicht schlechter als der euklidische.
Für den deutlich komplexeren *PKW-Seite*-Datensatz entsprachen die Einbettungen ebenfalls den Beobachtungen anhand der Visualisierungen, wobei hier weniger deutliche Verläufe entlang der Achsen erkennbar waren. Es ist hervorzuheben, dass der klassische euklidische Abstand wegen der variablen Geometrie auf diesen Datensatz nicht anwendbar ist und die in der Arbeit vorgeschlagene Methodik somit ihr Ziel erfüllt.
Für die geeignete Wahl von Parametern konnten keine unabhängigen, fest quantifizierbaren Regeln entdeckt werden. Eine Analyse der Eigenvektoren abhängig von γ ergab jedoch, dass bei Verwendung von EMD und niedriger Klassenanzahl zumindest die ersten Eigenvektoren recht stabil auf Änderungen reagieren. Auch unter Variation der Klassenanzahl n_{bins} des Histogramms (in Größen zwischen 8 und 200) blieben die wichtigsten Charakteristika der Einbettungen erhalten. Höhere Klassenanzahlen sind also nicht erforderlich und benötigen insbesondere auch etwas mehr Fingerspitzengefühl, was die Wahl eines geeigneten γ betrifft.
Dass bei der Darstellung der Daten durch eindimensionale Histogramme Informationen zu Position und Richtung der Verschiebungen verloren gehen, machte sich bei den untersuchten Datensätzen nicht bemerkbar, könnte aber ein Ansatzpunkt für zukünftige Untersuchungen mit komplexeren Datensätzen sein.

6 Zusammenfassung und Ausblick

Zusammenfassung

In dieser Bachelorarbeit wurde untersucht, wie *Diffusion Maps* als Methode der Nichtlinearen Dimensionsreduktion für Automobildaten aus Crashtestsimulationen eingesetzt wird. Darauf aufbauend wurde ein neues Verfahren vorgestellt und analysiert, welches in der Lage ist, auch Simulationsdaten mit unterschiedlichen Geometrien adäquat zu vergleichen.
Hierfür werden in einem Vorverarbeitungsschritt die bei der Crashsimulation errechneten Verschiebungen als *Histogramme* dargestellt und deren paarweise Histogrammabstände als Grundlage für die Kernfunktion von Diffusion Maps verwendet. Besonderes Gewicht wurde auf die Betrachtung der *Earth Mover's Distance* gelegt.
In numerischen Experimenten zeigte sich, dass dieses Verfahren der Dimensionsreduktion auf variable Geometrien angewandt tatsächlich die gewünschten Ergebnisse liefert. Bei Scharen von Simulationsläufen gleicher Geometrie war festzustellen, dass das untersuchte Verfahren mit Histogrammabständen – zumindest auf den betrachteten Datensätzen – sogar einige Phänomene deutlicher zum Vorschein bringt als das klassische Verfahren auf Basis des euklidischen Abstands. Insbesondere die Earth Mover's Distance zeigte sich als stabil bezüglich verschiedener Parametervariationen. Andere Abstände, wie der χ^2-Abstand, bieten im Idealfall vergleichbare Lösungen bei besseren Laufzeiten, sind aber empfindlich was die Aufteilung der Histogramme angeht.

Ausblick

Sollten sich die Ergebnisse dieser Arbeit mit weiteren Datensätzen aus der Industrie bestätigen lassen, kann das vorgestellte Verfahren tatsächlich für die Analyse von Automobildaten oder auch generelleren FEM-Simulationsdaten eingesetzt werden. Sowohl bei variabler Geometrie als auch bei unterschiedlich feiner Aufteilung der Netze bietet es eine Alternative zu der aufwändigen Projektion auf (ggf. ungenaue) Hilfsnetze.
Die Übertragung auf andere Anwendungsbereiche der Dimensionsreduktion wird nur in spezifischen Einzelfällen möglich sein, bei denen die Darstellung der Daten als Histogramme zur ursprünglichen Struktur passt. Bei Verschiebungen aus FEM-Simulationsdaten ist dies durch die Verwendung des FE-Gitters relativ gut möglich, da sich durch die engen Verknüpfungen benachbarter FE-Knoten recht glatte Histogramme bilden lassen. Für diese Arbeit wurden nur eindimensionale Histogramme betrachtet. Interessant könnte auch die Darstellung der Automobildaten als mehrdimensionale Histogramme oder Signaturen sein. Dadurch könnte nicht nur die absolute Verschiebung der Gitterpunkte, sondern das Verhältnis der

einzelnen Verschiebungen in x, y und z-Richtung erfasst werden. Auch auf diesen Strukturen kann EMD als Abstandsmaß verwendet werden.
Des weiteren können Methoden zur Beschleunigung von EMD wie die Verwendung beschränkter Grundabstände [25] untersucht werden. Neben den für diese Arbeit betrachteten Abständen EMD und χ^2 gibt es eine Vielzahl weiterer Histogrammabstände, welche mit der hier vorgeschlagenen Methode kombiniert und geprüft werden können.

A Anhang

Bei allen hier abgebildeten Plots wurde die Färbung der Punkte entsprechend des zweiten Eigenvektors vorgenommen. Falls nichts anderes erwähnt ist, wurde der *TRUCK-Beam*-Datensatz und der Abstand *EMD* verwendet.

A.1 *TRUCK-Beam*: Variation von γ

Alle hier dargestellten Varianten weisen die in Kapitel 5 aufgeführten, aussagekräftigen Korrelationen auf. Nur die Belegung der Koordinaten ändert sich. Während für niedrige γ auch Informationen über die Gesamtverformung in der Koordinate entsprechend des zweiten Eigenvektors vorliegen, finden diese sich bei wachsendem γ in der Koordinate entsprechend des dritten Eigenvektors.

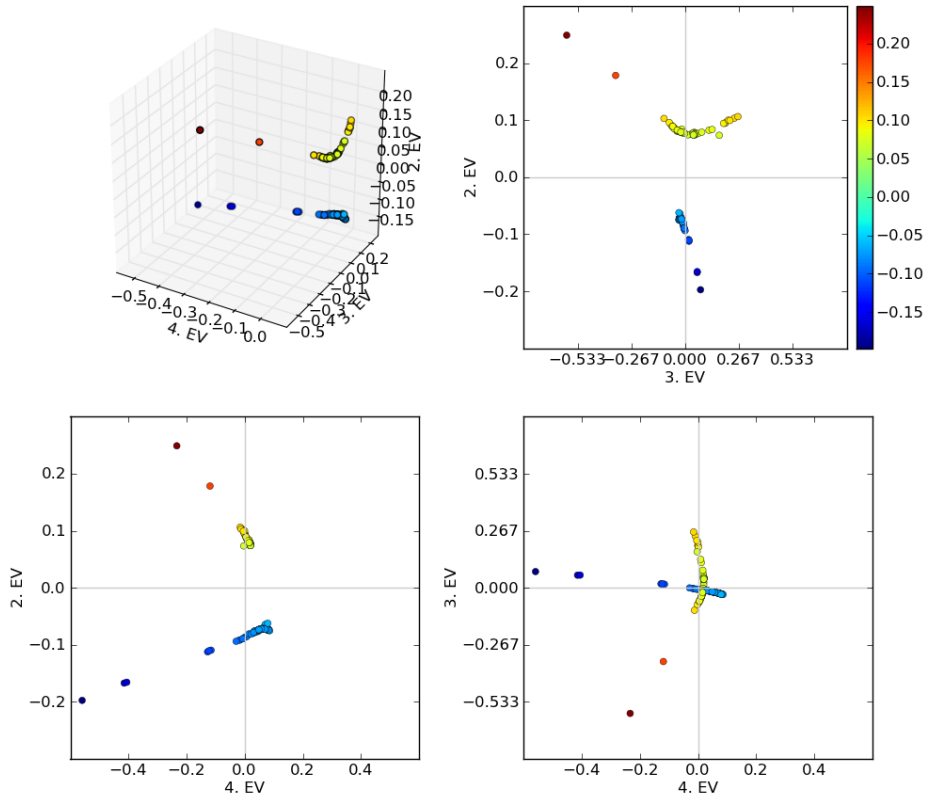

Abbildung A.1: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 10$, $\gamma = 2$

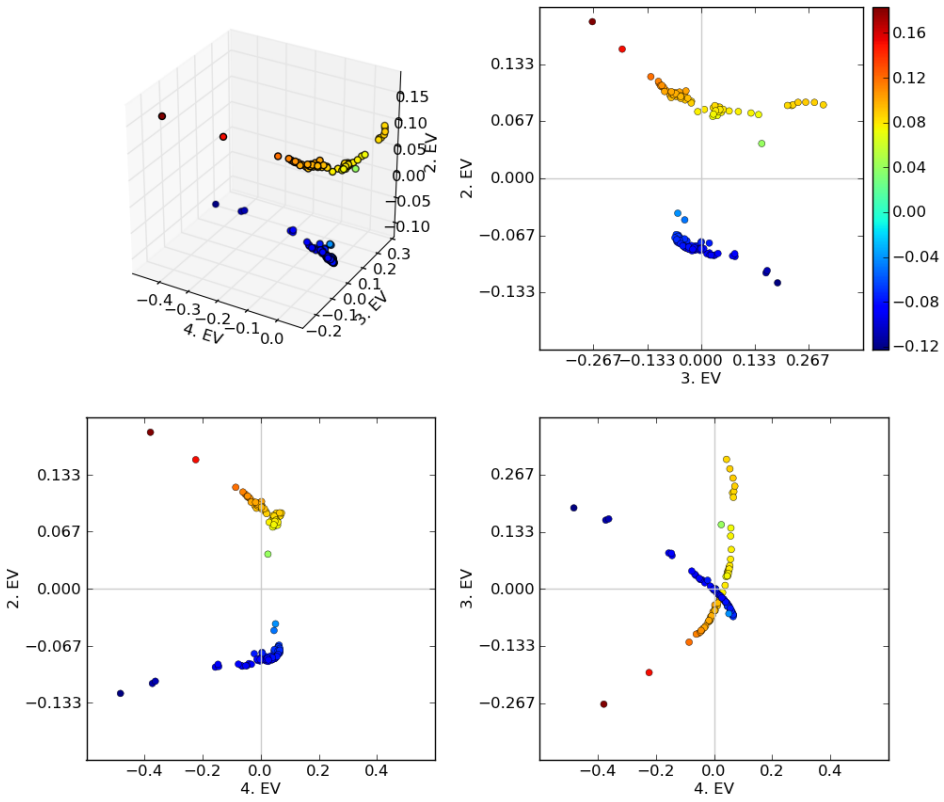

Abbildung A.2: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 10$, $\gamma = 4$

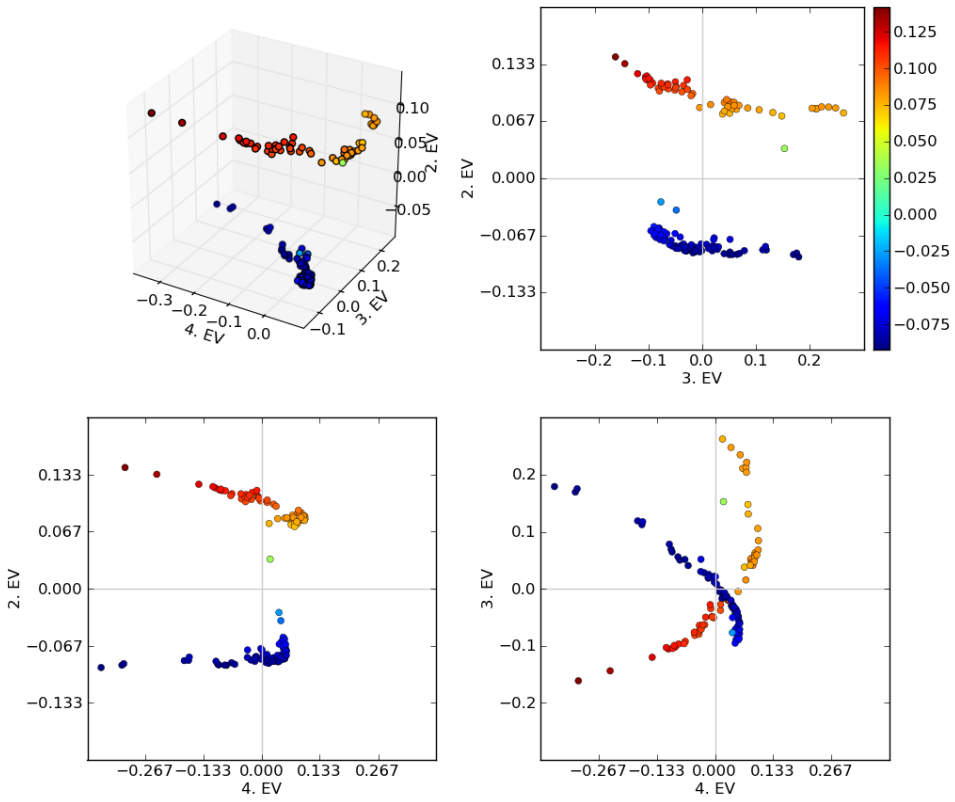

Abbildung A.3: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 10$, $\gamma = 8$

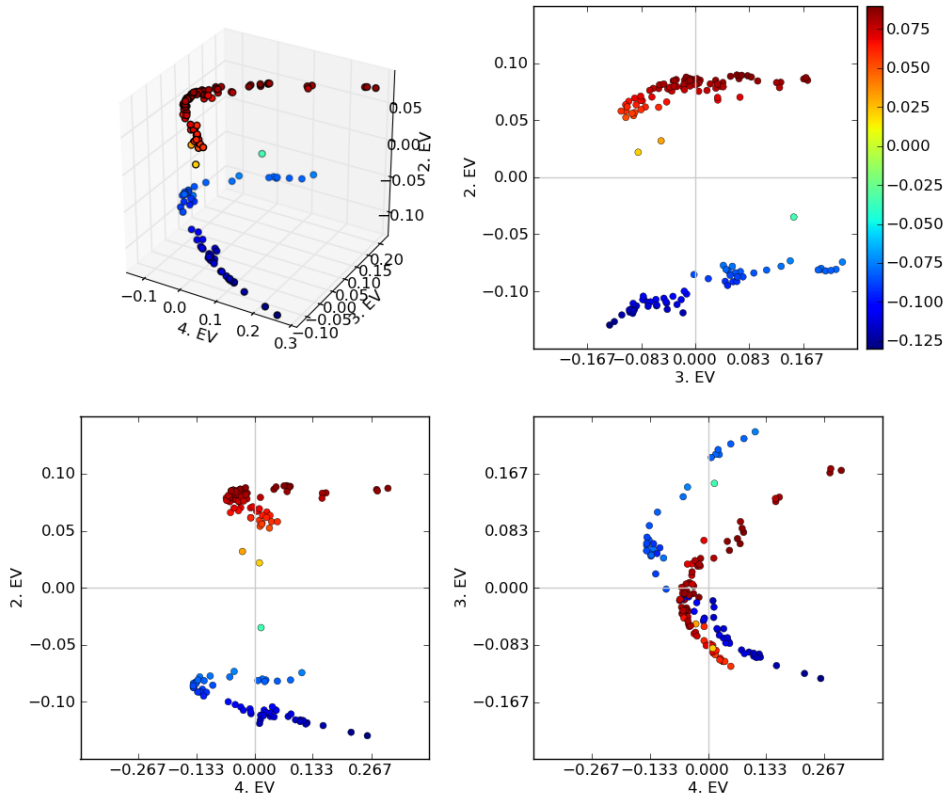

Abbildung A.4: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 10$, $\gamma = 16$. Ab dieser Wahl von γ bleibt die Struktur der Einbettung beibehalten, es ändert sich nur noch die Skalierung.

A.2 *TRUCK-Beam*: Variation von n_{bins}

Die Variation von n_{bins} wurde in Abschnitt 5.1.2 diskutiert. Für den *TRUCK-Beam*-Datensatz wurden mit $n_{\text{bins}} = 10$ gute Ergebnisse erzielt. Sinnvolle Einbettungen ergaben sich erst ab $n_{\text{bins}} = 8$. Für $n_{\text{bins}} \geq 15$ ist die Bifurkation wiederum weniger deutlich abzulesen, die innere Struktur der Einbettung bleibt jedoch für alle überprüften Werte erhalten.

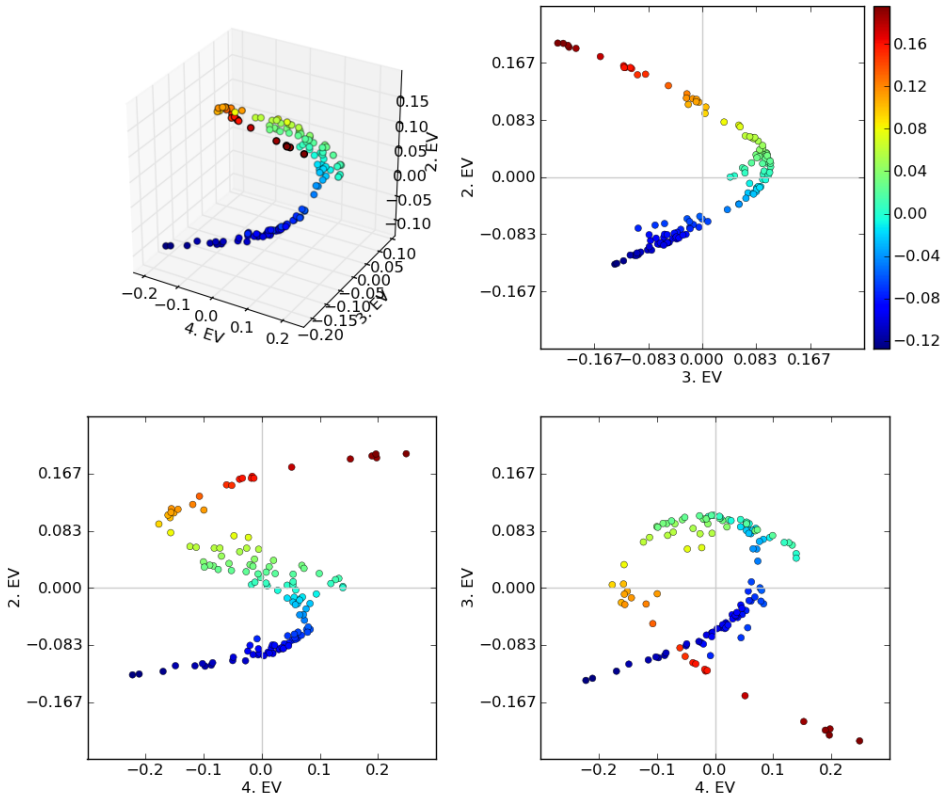

Abbildung A.5: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 5$, $\gamma = 32$. Verglichen mit den Visualisierungen sind keine Korrelationen feststellbar.

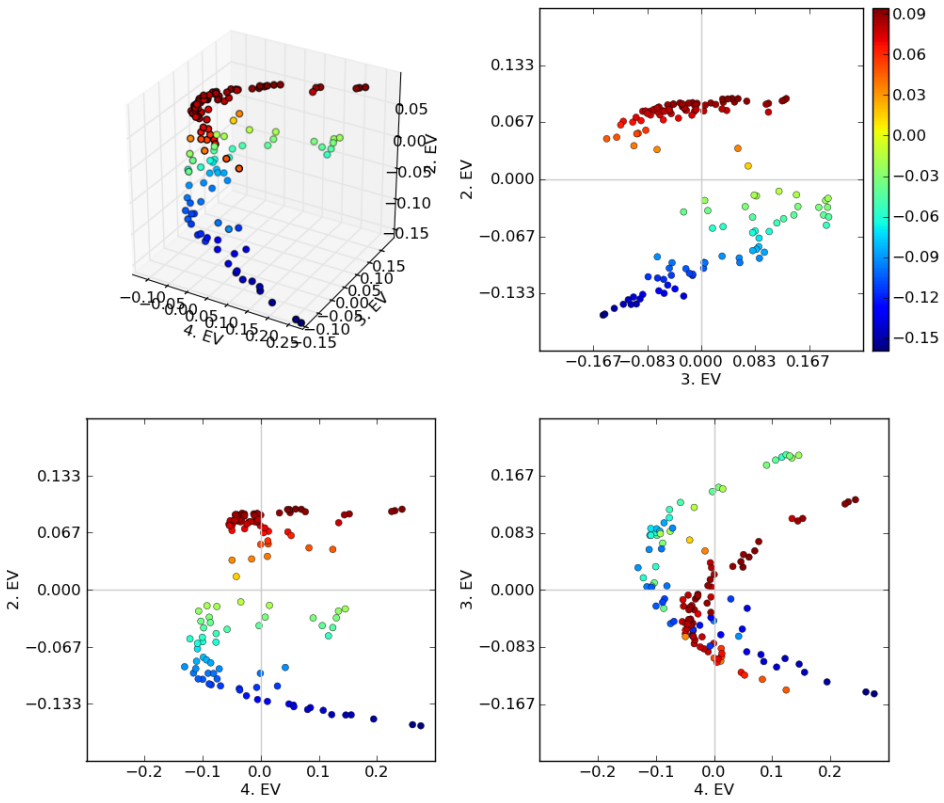

Abbildung A.6: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 14$, $\gamma = 32$. Hier wird erkennbar, dass die bei $n_{\text{bins}} = 10$ deutliche Trennung bereits verschwimmt.

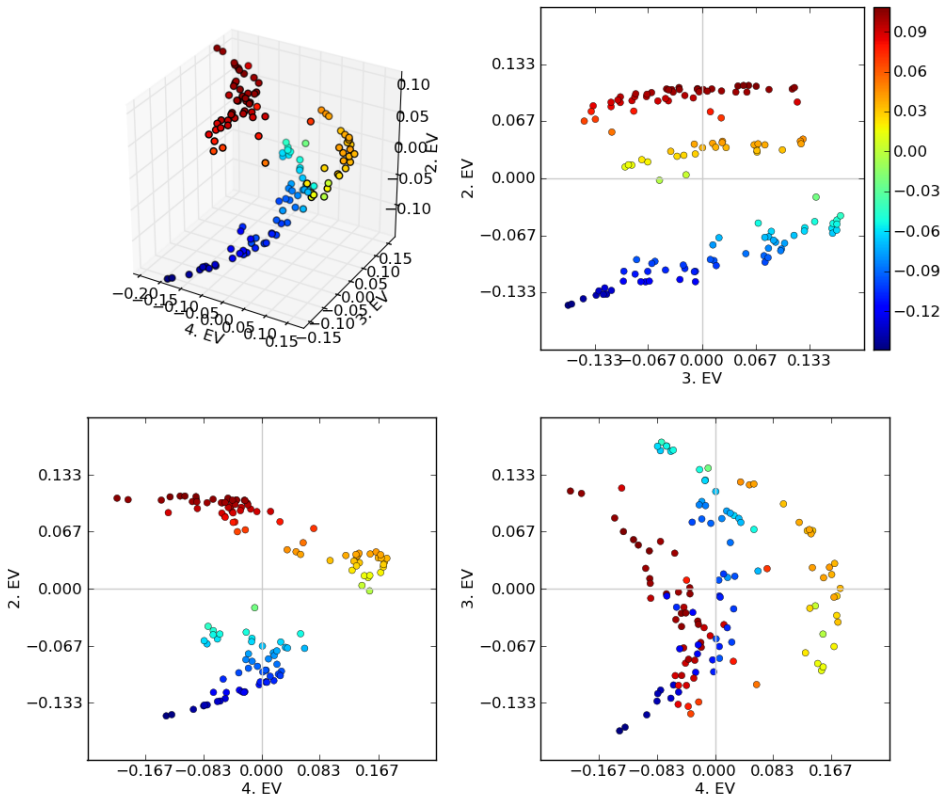

Abbildung A.7: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 20$, $\gamma = 32$. Es hat sich eine dritte Häufung gebildet, welche bei Betrachten der visualisierten Daten offensichtlich nicht mehr der Bifurkation entspricht. Die anderen Charakteristika sind jedoch noch immer enthalten, insbesondere korreliert die Einbettung weiterhin mit dem Verlauf der Gesamtverfomung.

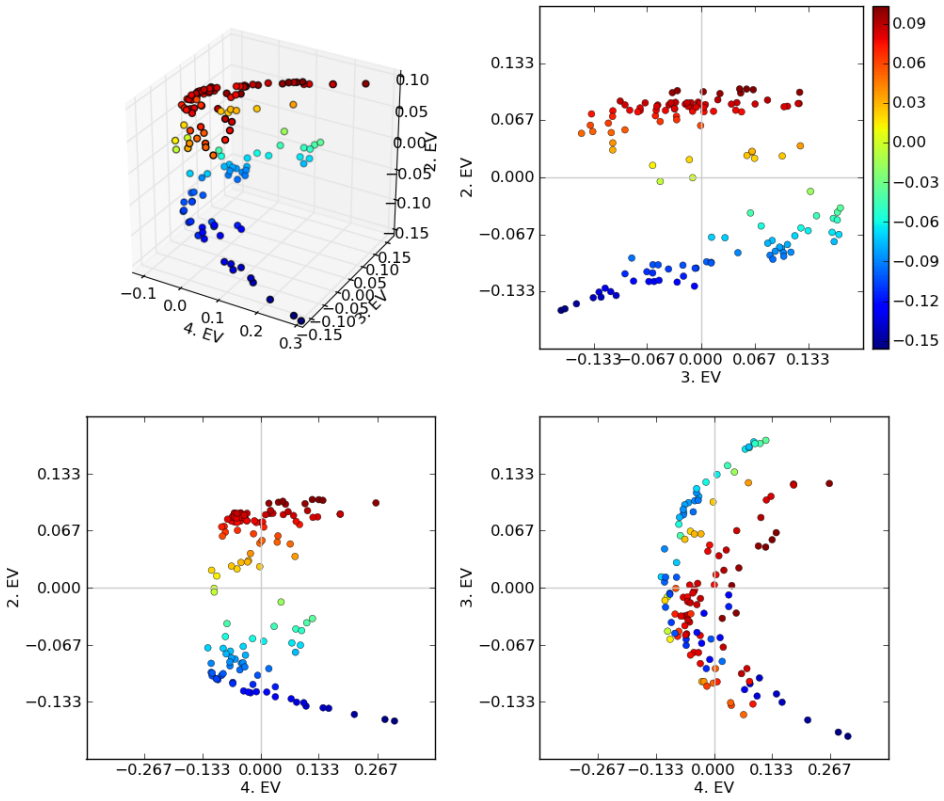

Abbildung A.8: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 30$, $\gamma = 32$. Die dritte Häufung, die bei $n_{\text{bins}} = 20$ erkennbar war, ist wieder verschwunden.

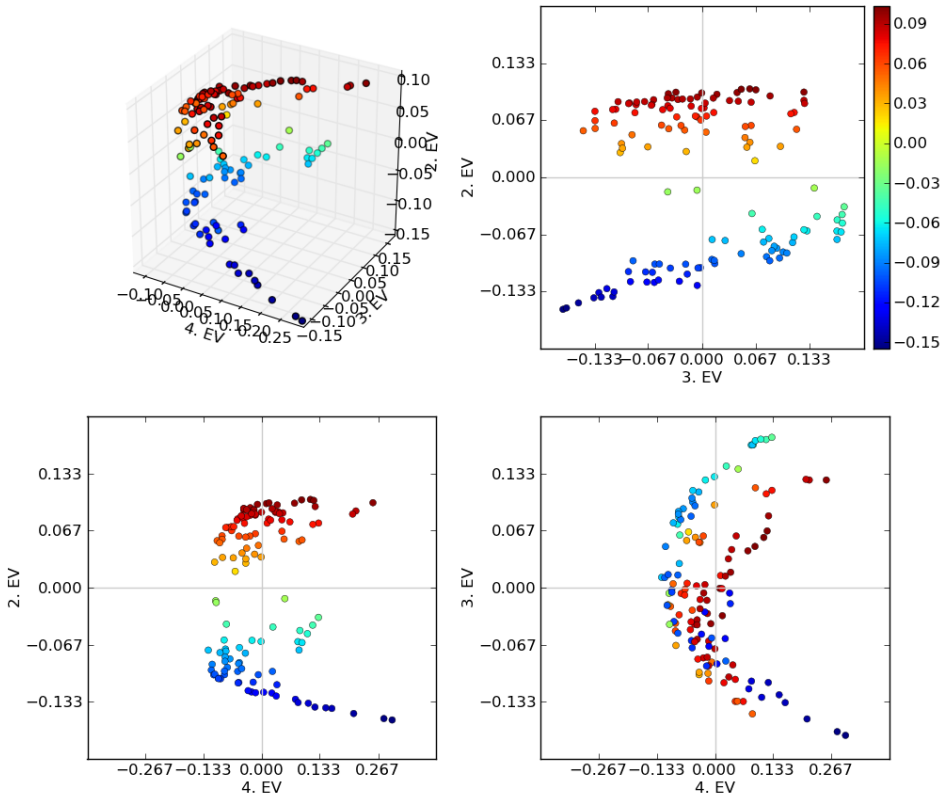

Abbildung A.9: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$. Parameter: $n_{\text{bins}} = 100$, $\gamma = 32$. Die Abgrenzung zwischen den Klassen ist deutlich verschwommen und zwei der hellgrünen Punkte in der Mitte wurden durch den zweiten Eigenvektor unpassend zugeordnet. Dieses Verhalten ähnelt dem von Diffusion Maps mit euklidischem Abstand.

A.3 Diffusion Maps mit χ^2-Abstand

Die folgenden Abbildungen zeigen Einbettungen der zuvor betrachteten Datensätze unter Verwendung von Diffusion Maps mit χ^2-Abstand, welche vergleichbare Strukturen aufweisen wie die unter Verwendung von EMD berechneten Einbettungen. Details zum Vergleich der Methoden finden sich in *Abschnitt 5.3.2*.

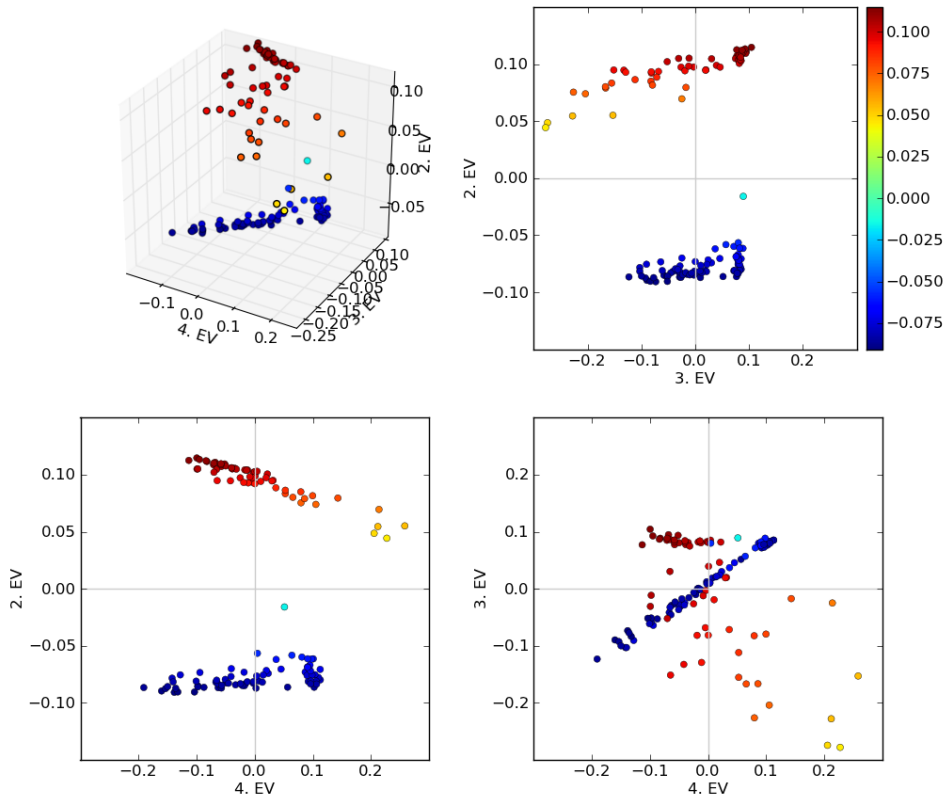

Abbildung A.10: Plot der Einbettung der Simulationsläufe von *TRUCK-Beam* in den $\mathbb{R}^3$ mit Diffusion Maps und χ^2. Parameter: $n_{\text{bins}} = 12$, $\gamma = 32$. Die für $n_{\text{bins}} = 10$ vorhandenen Charakteristika sind hier ebenfalls erkennbar und die Bifurkation entspricht den Daten. Die Form der Punktwolke unterscheidet sich dennoch sichtbar.

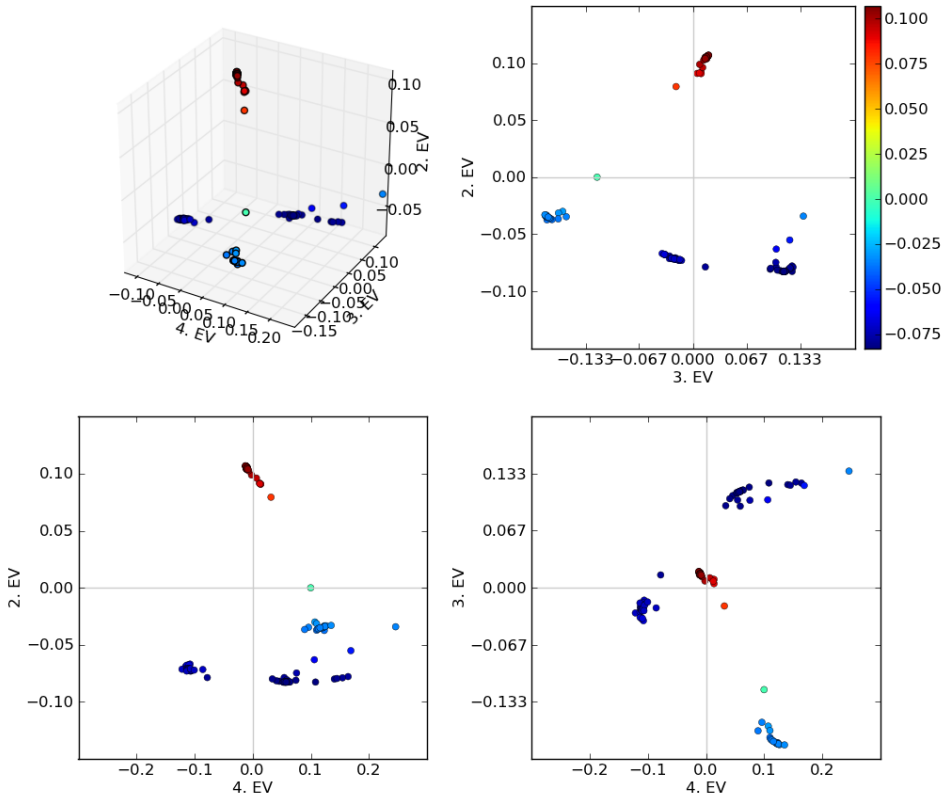

Abbildung A.11: Plot der Einbettung der Simulationsläufe von *PKW-Seite* in den $\mathbb{R}^3$ mit Diffusion Maps und χ^2. Parameter: $n_{\text{bins}} = 16$, $\gamma = 32$. Beeindruckend ist der Vergleich mit Abbildung 5.13, in der derselbe Datensatz mit EMD eingebettet dargestellt ist: Auch für die komplexeren Daten liefert der χ^2-Abstand vergleichbar gute Ergebnisse.

Abbildungsverzeichnis

Literaturverzeichnis

[1] Belkin, M. und Niyogi, P.: *Laplacian Eigenmaps for dimensionality reduction and data representation.* Neural Comput., 15(6):1373–1396, 6.

[2] Bohn, B., Garcke, J., Iza-Teran, R., Paprotny, A., Peherstorfer, B., Schepsmeier, U. und Thole, C. A.: *Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods.* Procedia Computer Science, 18:621 – 630, 2013. 2013 International Conference on Computational Science.

[3] Braess, D.: *Finite Elemente.* Springer, Berlin, 4. Aufl., 2007.

[4] Coifman, R. R. und Lafon, S.: *Diffusion maps.* Applied and Computational Harmonic Analysis, 21(1):5–30, 2006.

[5] Deuflhard, P. und Bornemann, F.: *Numerische Mathematik 2. Gewöhnliche Differentialgleichungen.* de Gruyter, Berlin, 3. Aufl., 2008.

[6] ESI Group: *PAM-CRASH.* `http://virtualperformance.esi-group.com`. Zuletzt besucht: August 2013.

[7] Freedman, D. und Diaconis, P.: *On the histogram as a density estimator: L2 theory.* Probability Theory and Related Fields, 57(4):453–476, 1981.

[8] Georgii, H.: *Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik.* de Gruyter, 2007.

[9] GNS: *GNS mbH.* `http://www.gns-mbh.com`. Zuletzt besucht: August 2013.

[10] Haug, E., Scharnhorst, T. und DuBois, P.: *FEM-Crash. Berechnung eines Fahrzeugfrontalaufpralls.* VDI Berichte 613, S. 479 – 505, 1986.

[11] Hunter, J. D.: *Matplotlib: A 2D graphics environment.* Computing In Science & Engineering, 9(3):90–95, 2007.

[12] Iza-Teran, R.: *Enabling the Analysis of Finite Element Simulation Bundles.* International Journal for Uncertainty Quantification, 2013.

[13] Jähne, B.: *Digitale Bildverarbeitung.* Springer, 2005.

[14] Korte, B. und Vygen, J.: *Kombinatorische Optimierung - Theorie und Algorithmen.* Springer-Verlag, Berlin, 2008.

[15] Lafon, S.: *Diffusion Maps and Geometric Harmonics.* Dissertation, Yale University, 2004.

[16] Lee, J. A. und Verleysen, M.: *Nonlinear Dimensionality Reduction.* Springer Science and Business Media, New York, 2007.

[17] Livermore Software Technology Corporation: *LS-DYNA Keyword User's Manual*, 2004.

[18] Luxburg, U.: *A tutorial on spectral clustering.* Statistics and Computing, 17(4):395–416, 2007.

[19] Meila, M. und Shi, J.: *A Random Walks View of Spectral Segmentation.* In: *8th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.

[20] Meywerk, M.: *CAE-Methoden in der Fahrzeugtechnik.* Springer, Berlin, 2007.

[21] Mohar, B.: *The Laplacian spectrum of graphs.* In: *Graph Theory, Combinatorics, and Applications*, S. 871–898. Wiley, 1991.

[22] Nadler, B., Lafon, S., Coifman, R. und Kevrekidis, I. G.: *Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms.* In: *Principal manifolds for data visualization and dimension reduction*, S. 238–260. Springer, 2008.

[23] Ng, A. Y., Jordan, M. I. und Weiss, Y.: *On Spectral Clustering: Analysis and an algorithm.* In: *Advances in Neural Information Processing Systems*, S. 849–856. MIT Press, 2001.

[24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. und Duchesnay, E.: *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[25] Pele, O. und Werman, M.: *The Quadratic-Chi Histogram Distance Family.* In: *ECCV*, 2010.

[26] Rubner, Y., Tomasi, C. und Guibas, L. J.: *The earth mover's distance as a metric for image retrieval.* International Journal of Computer Vision, 40:99–121, 2000.

[27] Schöne, C., Iza-Teran, R. und Garcke, J.: *A Framework for Simulation Process Management and Data Mining.* In: *1st International Simulation Data and Process Management Conference, Salzburg, Jun 9-12*, 2013.

[28] Scott, D. W.: *On optimal and data-based histograms.* Biometrika, 66(3):605–610, Dez. 1979.

[29] Shawe-Taylor, J. und Cristianini, N.: *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[30] Shi, J. und Malik, J.: *Normalized Cuts and Image Segmentation.* IEEE Trans. Pattern Anal. Mach. Intell., 22(8):888–905, 2000.

[31] Sturges, H. A.: *The choice of a class interval.* American Statistical Association, 21:65–66, 1926.

[32] Tenenbaum, J. B., Silva, V. und Langford, J. C.: *A Global Geometric Framework for Nonlinear Dimensionality Reduction.* Science, 290(5500):2319–2323, 2000.

[33] The HDF Group: *HDF5.* `http://www.hdfgroup.org/HDF5/`. Zuletzt besucht: August 2013.

[34] Van Der Walt, S., Colbert, S. C. und Varoquaux, G.: *The NumPy array: a structure for efficient numerical computation.* Computing in Science & Engineering, 13(2):22–30, 2011.