SIMDATA-NL
Nichtlineare Charakterisierung und Analyse von FEM-Simulationsergebnissen für Autobauteile und Crash-Tests

Michael Griebel, Hans-Joachim Bungartz, Claudia Czado, Jochen Garcke, Ulrich Trottenberg, Clemens-August Thole, Bastian Bohn, Rodrigo Iza-Teran, Alexander Paprotny, Benjamin Peherstorfer und Ulf Schepsmeier

Zusammenfassung

Die Arbeiten am Verbundprojekt SIMDATA-NL haben sich mit der computerbasierten Unterstützung bei der Analyse und ingenieursmäßigen Interpretation von Scharen von Simulationsdaten aus dem Automobilbereich befasst. In diesem Kontext waren vor Projektbeginn Arbeiten zur Analyse mittels linearer Dimensionsreduktionsverfahren bekannt. Da die im Crash-Bereich oftmals auftretenden nichtlinearen Effekte und Bucklingmoden mit diesen Verfahren jedoch nicht adäquat zu beschreiben sind, haben sich die Arbeiten im Projekt insbesondere auf die Neu- und Weiterentwicklung geeigneter nichtlinearer Methoden konzentriert. Durch regelmäßiges Feedback aus der Automobilindustrie wurden diese verfeinert und an die konkreten Bedürfnisse angepasst.
Während der Projektlaufzeit hat sich die Anwendung eines dreistufigen Workflows etabliert, welcher die folgenden Punkte umfasst:

1. Ein Clustering zur groben Separation auftretender Effekte sowie als initiale Dimensionsreduktion.

2. Eine nichtlineare Dimensionsreduktion der geclusterten Daten zum Detektieren der intrinsischen Dimension der Daten sowie der endgültigen Separation der auftretenden Effekte.

3. Die Analyse der resultierenden Daten zur Identifikation von relevanten Designveränderungen und der Bewertung der Güte der erstellten Einbettung.

Im ersten Schritt (Clustering) hat sich gezeigt, dass sowohl Standardverfahren, wie das k-means Clustering, als auch neue Verfahren mit nichtkonvexen Clustergrenzen, so z.B. das spektrale Clustering mit dünnen Gittern, geeignet sind, um die Automobildaten zu verarbeiten und bereits erste Interpretationen über das Crashverhalten zuzulassen. Insbesondere letztere Verfahren sind im Stande, Ergebnisse zu liefern, die den physikalischen Crashprozess präzise widerspiegeln.
Im zweiten Schritt (Dimensionsreduktion) hat die (Weiter-)Entwicklung diverser Verfahren mit unterschiedlichen Charakteristiken gezeigt, dass es insbesondere vom konkreten Analyseziel im dritten Schritt abhängt, welche Art von Verfahren sich optimal eignet. Ist man lediglich an guten Rekonstruktionsfehlern interessiert, können lineare Verfahren bereits zufriedenstellende Ergebnisse liefern, sofern genügend viele Dimensionen genutzt werden. Liegt das Hauptinteresse hingegen auf dem Detektieren der minimalen Einbettungsdimension der Daten und somit der korrekten Anzahl der auftretenden Effekte, ist das generative Principal Manifold Learning Verfahren mit adaptiven dünnen Gittern gut geeignet. Eine valide Identifikation der Input-Parameterabhängigkeiten ist mit kernbasierten Verfahren, wie Diffusion Maps, erzielbar.
Durch den entwickelten Workflow konnten nun erstmalig industrierelevante Beispiele von Auto-Crashsimulationen mit nichtlinearen Data-Mining-Methoden untersucht werden. Die Beispiele haben gezeigt, dass komplexe Effekte mit nichtlinearen Verfahren mit weniger Dimensionen dargestellt werden können, als dies bei den bisher verwendeten linearen Verfahren möglich war. Dies wurde sowohl anhand quantitativer Größen (kleinere Rekonstruktionsfehler) als auch durch qualitative Vorteile, wie einer einfacheren und übersichtlicheren Darstellung der Daten für Ingenieure, belegt.

Inhaltsverzeichnis

Bericht

Das Verbundprojekt SIMDATA-NL wurde im Rahmen der sechsten Förderperiode des Programms “Mathematik für Innovationen in Industrie und Dienstleistungen” durch das BMBF in den Jahren 2010 bis 2013 gefördert. An diesem Projekt waren folgende Universitäten und Forschungseinrichtungen beteiligt

- die Rheinische-Friedrich-Wilhelms-Universität Bonn,
- die Technische Universität München,
- die Technische Universität Berlin und
- die Fraunhofer Gesellschaft.

1 Aufgabenstellung

In der Automobilindustrie ist die Simulation von Crashtests neuer KFZ-Modelle am Rechner heutzutage eine günstige Alternative zu kostspieligen Real-Crashtests. Mittlerweile stellt sich allerdings während des Forschungs- und Entwicklungsprozesses regelmäßig das Problem riesige Archive von hochdimensionalen Finite Element Crashsimulationen explorieren zu müssen, um diese im Hinblick auf unterschiedliche Crashverhalten zu analysieren. Erst Vergleiche zwischen verschiedenen Simulationsergebnissen lassen Schlussfolgerungen für eine verbesserte Modellierung zu. Der aktuelle Produktentwicklungsprozess besteht größtenteils aus einer manuellen, iterativen Analyse der Zusammenhänge zwischen Input-Parameter-Änderungen (z.B. Blechdickenvariationen) und deren Auswirkungen auf die Zielgrößen, die aus der Simulation des physikalischen Prozesses (z.B. eines Aufpralls) abgeleitet wurden. Die anschließenden Modelländerungen werden dann auf Basis des Know-how des Ingenieurs sowie einer Reihe funktioneller und gesetzlicher Anforderungen (z.B. Fahrverhalten, Sicherheit, etc.) durchgeführt. Dies ist für den Ingenieur ein sehr zeitintensiver Prozess, der durch eine computergestützte Analyse der vorliegenden Daten deutlich beschleunigt werden kann.
Fortschritte bezüglich der Datenaufbereitung und des Datenzugriffs wurden in den letzten Jahren u.A. durch die Einführung sogenannter *Simulations-Daten-Management (SDM)*-Systeme erzielt, die der Verwaltung und Analyse der Informationen aus Archiven verschiedener Simulationsläufe dienen. Allerdings verlangen diese Systeme eine Standardisierung des Produktionsprozesses, was dazu führt, dass nicht-standardisierte Datenbestände damit nicht mehr analysiert werden können. Des Weiteren beschränkt sich die Analyse innerhalb solcher SDM-Systeme, aufgrund der hohen Dimensionalität der Simulationsdatensätze, hauptsächlich auf den Vergleich von aus der Simulation abgeleiteten Zielgrößen (Skalare bzw. Vektoren niedriger Dimensionen). Wie zuverlässig diese Zielgrößen das generelle Crashverhalten darstellen, auch im Vergleich zu den vollen hochdimensionalen Datensätzen, ist ein offenes Problem.

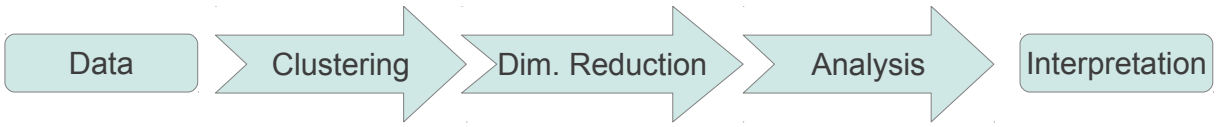


Abbildung 1: Der grobe Workflow des Projekts.

Das Ziel unserer Arbeiten war somit die Untersuchung und die Bereitstellung von neuartigen Verfahren, welche das Auffinden der Designveränderungen mit relevanten Auswirkungen auf das generelle Crashverhalten (z.B. Bucklingmoden und sicherheitsrelevante Kenngrößen wie HIC-Index (Head Injury Criterion) und Stirnwandintrusion) erleichtern. Dabei sollten die verwendeten mathematischen Methoden - im Unterschied zu bisherigen Analyseverfahren - direkt auf den hochdimensionalen Simulationsdatensätzen arbeiten. Ein grundlegendes Problem in diesem Kontext ist der sogenannte Fluch der Dimension. Hiermit ist vor allem das immense Wachstum der Laufzeit von numerischen Algorithmen gemeint, welche vielmals exponentiell mit der Dimension der Eingabedaten skalieren. In nahezu sämtlichen praxisbezogenen Szenarien hat sich allerdings in der Vergangenheit gezeigt, dass die nominell hochdimensionalen Daten näherungsweise auf einer niederdimensionalen, eventuell nichtlinearen *intrinsischen Mannigfaltigkeit* liegen. Ist diese Mannigfaltigkeit gefunden, lässt sich der Fluch der Dimension dadurch umgehen, dass die Algorithmen in diesem niederdimensionalen Koordinatensystem arbeiten.
Eine besondere Herausforderungen für die verwendeten nichtlinearen Dimensionsreduktionsmethoden zur Detektion der intrinsischen Mannigfaltigkeit stellte in diesem Projekt die, im Vergleich zu ihrer Datendimension, sehr geringe Anzahl an Simulationsergebnissen dar.
Des Weiteren ist ein intrinsisches Koordinatensystem für das gesamte KFZ-Modell hier nicht zielführend, da dann nicht mehr garantiert werden kann, dass verschiedene Bucklingmoden und Biegeverhalten korrekt getrennt werden. Außerdem kann die Dimension der entsprechenden Mannigfaltigkeit größer ausfallen, als es bei einer lokalen Analyse der Fall wäre. Aus diesem Grund wurde das Gesamtmodell zunächst mittels Clustering-Verfahren unterteilt. Dies kann als initiale Dimensionsreduktion verstanden werden und soll das Auffinden von auftretenden Effekten (wie z.B. Buckling) erleichtern. Zur Verarbeitung der Daten und der Konstruktion der intrinsischen Mannigfaltigkeit sowie der Separation der auftretenden Effekte orientierten wir uns während des Projekts an der Pipeline in Abbildung 1, welche in Abschnitt 3 im Detail vorgestellt wird.

2 Wissenschaftlicher und technischer Stand, an den angeknüpft wurde

Erste Algorithmen zur Dimensionsreduktion von Crashtestdatensätzen wurden bereits vor Projektbeginn am Fraunhofer SCAI entwickelt. Diese basieren jedoch auf einer

linearen Dimensionsreduktion mittels Hauptachsenanalyse und sind nicht in der Lage nichtlineare Effekte herauszufiltern, siehe z.B. [16]. Nichtlineare Algorithmen, die direkt auf den hochdimensionalen (verformten) Geometrien arbeiten und so eine Analyse von großen Crashsimulationsdatensätzen durchführen, existierten bisher nicht.
Initiale Untersuchungen der Eignung des nichtlinearen *Diffusion Maps* Algorithmus zur Analyse von Crashtestdaten begannen bereits unmittelbar vor Projektbeginn am SCAI. Im Bereich des Manifold Learning und Data-Mining existierten bereits eine Vielzahl erfolgreicher Dimensionsreduktionsalgorithmen, siehe [10]. Obwohl bereits verschiedene Dünngitter-Data-Mining Algorithmen existierten, blieb die Übertragung der bekannten Konzepte auf ein Clustering-Problem bisher aus. Des Weiteren sind die meisten Data-Mining-Verfahren nicht generativ und erlauben somit keine direkte Rekonstruktion (und damit keine direkte Kontrolle des Einbettungsfehlers) der Daten. Eine Dünngitter-Variante des generativen *Principal Manifold Learning* Verfahrens wurde bereits in [6] behandelt. Diese nichtadaptive Variante ist allerdings nicht in der Lage die korrekte Einbettungsdimension zu ermitteln.

3 Erzielte Ergebnisse

Während der Projektlaufzeit wurde ein Arbeitsablauf nach Abbildung 1 etabliert, welcher es erlaubt die notwendigen Schritte weitestgehend unabhängig voneinander zu betrachten und sich letztlich durch die Kombination der einzelnen Arbeitspakete auch als zielführend erwies. Im Folgenden gehen wir näher auf die einzelnen Komponenten des Workflows ein:

- **Stufe 0 (Daten)**:

  Aus einem gegebenen Archiv von Crashsimulationen zu einem spezifischen Modell wurden zunächst die Verschiebungsvektoren der Finite-Element-Knoten zwischen ausgewählten Zeitpunkten extrahiert. Diese dienten als Grundlage für unsere Analysen.

- **Stufe 1**: Ziel war es eine Gruppierung der FEM-Knoten zu finden, so dass sich alle Knotenverschiebungen in einem Cluster ähnlich verhalten. Hierfür wurden zwei unterschiedliche Ansätze verfolgt: Erstens, ein statistischer Ansatz mit klassischen Clusteringmethoden wie k-means und hierarchischem Clustering und zweitens ein Clustering mit raumadaptiven dünnen Gittern. Die Motivation zur Entwicklung des zweiten Ansatzes war die Tatsache, dass sich Dünngitter Methoden erfahrungsgemäß gut für große Datenmengen eignen. Eine besondere Herausforderung in beiden Ansätzen war die Schätzung oder Bestimmung der Anzahl an Gruppen/Clustern und des geeigneten Distanzmaßes.

- **Stufe 2**: Nach dem initialen Clustering wurden die Verschiebungsvektoren, die den Knoten eines Clusters zugeordnet sind, mittels Dimensionsreduktionsverfahren verarbeitet. Hierbei stand insbesondere das Erkennen niedrigdimensionaler

Strukturen mittels nichtlinearen Methoden im Vordergrund. Diese niederdimensionale Struktur charakterisiert das Verhalten der hochdimensionalen Verschiebungsvektoren vollständig und erlaubt es verschiedene Simulationen zu analysieren, ohne dass die dabei verwendeten mathematische Methoden dem Fluch der Dimension unterliegen. Hierzu wurden diverse nichtlineare Dimensionsreduktionsverfahren entwickelt bzw. erweitert, sodass diese im Rahmen der Crashsimulationsanalyse verwendet werden konnten.

Im Rahmen des Projekts wurden insbesondere eine lokal lineare Approximation, ein kernbasiertes Verfahren (Diffusion Maps) und ein auf dimensionsadaptiven dünnen Gittern aufbauendes Verfahren zum Lernen von Mannigfaltigkeiten (Principal Manifold Learning) angewandt.

- **Stufe 3**: Zur Analyse der eingebetteten Daten und der Qualität der mit den Dimensionsreduktionsverfahren gefundenen Darstellung war insbesondere die Detektion der Korrelationen zwischen generellem Crash-Verhalten (z.B. sichtbare Auswirkungen oder Verhalten gewisser Kenngrößen) und den Input-Parametern (z.B. Blechdicken) relevant. Eine Einbettung der Daten in einen zwei- oder dreidimensionalen Raum lieferte hierbei oftmals allein durch eine visuelle Analyse des Ergebnisses neue ingenieursrelevante Erkenntnisse.

  Ein weiteres Gütekriterium war die Größe des Rekonstruktionsfehlers, den wir beim Principal Manifold Learning direkt messen und mit der PCA vergleichen konnten. Beim lokal linearen Ansatz wurde eine stückweise konstante Rekonstruktion der Daten zugrundegelegt um den Effekt einer lokalisierten Analyse im Simulationsparameterraum zu studieren. Eine fortgeschrittenere Methode zum Vergleich der Beziehungen zwischen originalen und rekonstruierten Simulationsläufen anhand relevanter Crashsimulationskenngrößen, wie z.B. der Intrusion spezieller Punkte auf der Stirnwand, wurde auf Basis von Vine-Copulas entwickelt.

- **Stufe 4**: Gemeinsam mit den Partnern aus der Automobilindustrie wurden die Ergebnisse anschließend interpretiert, um den Einfluss der Input-Parameter auf das Crash Verhalten zu untersuchen.

Im weiteren Verlauf des Berichts gehen wir nun auf die einzelnen Punkte des Workflows ein und beschreiben die in diesen Bereichen erzielten Ergebnisse.

3.1 Pre-Processing der Crashsimulationsdaten (Stufe 0)

Für ein gegebenes KFZ-Modell betrachteten wir R Simulationen mit T Zeitschritten, die anhand eines Finite-Element-Modells mit M Knoten berechnet wurden. Im Folgenden bezeichnen wir mit $n_1, \ldots, n_M \in \mathbb{N}$ die Knoten, die im Simulationslauf $r \in \{1, \ldots, R\}$ und zum Zeitschritt $t_i \in \{1, \ldots, T\}$ die Positionen ${}^r x_1^{t_i}, \ldots, {}^r x_M^{t_i} \in \mathbb{R}^3$ haben. Wobei M Knoten im Weiteren nicht zwingend das gesamte Modell beschreiben, sondern z.B. auch für die Knoten eines mechanisches Teils, oder einer Teilmenge der gesamten Knoten (Cluster) stehen. Die Verschiebung (Displacement) zwischen zwei Zeitschritten

t_i und t_j für einen finite Element Knoten n_l bezeichnen wir mit ${}^r d_l^{t_i,t_j}$ und den gesamten Verschiebungsvektor der Dimension $D = 3M$ mit ${}^r \boldsymbol{d}^{t_i,t_j}$. Es gilt also

$$ {}^r d_l^{t_i,t_j} := {}^r x_l^{t_i} - {}^r x_l^{t_j} \in \mathbb{R}^3 \quad \text{und} \quad {}^r \boldsymbol{d}^{t_i,t_j} := \left(\left({}^r d_1^{t_i,t_j} \right)^T, \ldots, \left({}^r d_M^{t_i,t_j} \right)^T \right)^T \in \mathbb{R}^D. \quad (1) $$

Diese Vektoren dienten uns als Grundlage für die weiteren Schritte des Workflows. Detaillierte Betrachtungen führten wir während der Projektlaufzeit insbesondere an den folgenden zwei KFZ-Modellen durch:

- Direkt zu Beginn des Projekts Ende 2010 wurde die erste gemeinsame Datenbasis spezifiziert. Dazu wurden am Fraunhofer SCAI 126 Frontalcrash-Simulationen mit verschiedenen Material-Blechdicken für das vom NCAC (National Crash Analysis Center) öffentlich zur Verfügung gestellte Chevrolet C2500 Pickup Truck Modell berechnet (siehe auch http://www.ncac.gwu.edu/vml/models.html). Dieser Datensatz wurde anschließend allen Projektpartnern in Form von komprimierten FEMZIP-Dateien zur Verfügung gestellt und diente in der ersten Projektphase als Benchmark-Test für die entwickelten Algorithmen. Die Problemgröße bei diesem Beispiel ist mit ca. 66.000 Finite-Element-Knoten und 17 Zeitschritten noch sehr moderat. Von besonderem Interesse in diesem Datensatz waren die qualitativ unterschiedlichen Knickverhalten der Längsträger (siehe Abbildung 3) bei Änderung der Blechdicken diverser Bauteile (siehe Abbildung 2). Gesucht wurde eine geeignete Darstellung der Simulationsdaten unter dem Gesichtspunkt der korrekten Klassifizierung dieses Knickverhaltens. Des Weiteren sollte die Darstellung in geeigneter Art und Weise Rückschlüsse über die Originaldaten zulassen, hierzu wurden in Stufe 3 verschiedene Fehlermaße eingeführt.

- Nach der Entwicklung und Implementierung der Clustering- und Dimensionsreduktionsalgorithmen stellte sich als nächste Herausforderung die Anwendung auf ein Beispiel mit industrierelevanter Größe. Hierzu dienten uns zwei vom SCAI zur Verfügung gestellte Datensätze des Ford Taurus Modells. Die Grundlage für den ersten Datensatz war eine minimale Variation der Blechdicken aller Teile, um so numerische Ungenauigkeiten (z.B. durch Parallelisierung) zu erzeugen. Für den zweiten Datensatz wurden größere Blechdicken-Veränderungen (5 $\%$) an 19 Bauteile zugelassen (siehe Abbildung 5), wie sie auch im Fertigungsprozess zu finden sind. Die beiden Datensätze umfassten 251 bzw. 289 Simulationen mit jeweils etwa 900.000 Knoten und 300 Zeitschritten.

3.2 Initiales Clustering der Verschiebungsvektoren (Stufe 1)

Im ersten Schritt unseres Ansatzes teilten wir die FEM-Knoten aller Simulationen in Gruppen, sogenannten *Clustern*, ein, sodass alle Knoten in einem Cluster ein ähnliches Verschiebungsverhalten zeigen. Eine solche Vorgruppierung der Knoten vor dem eigentlichen Dimensionsreduktionsschritt erwies sich als sehr nützlich und wird in Abbildung 2 exemplarisch anhand von 10 Längsträgerbauteilen demonstriert.

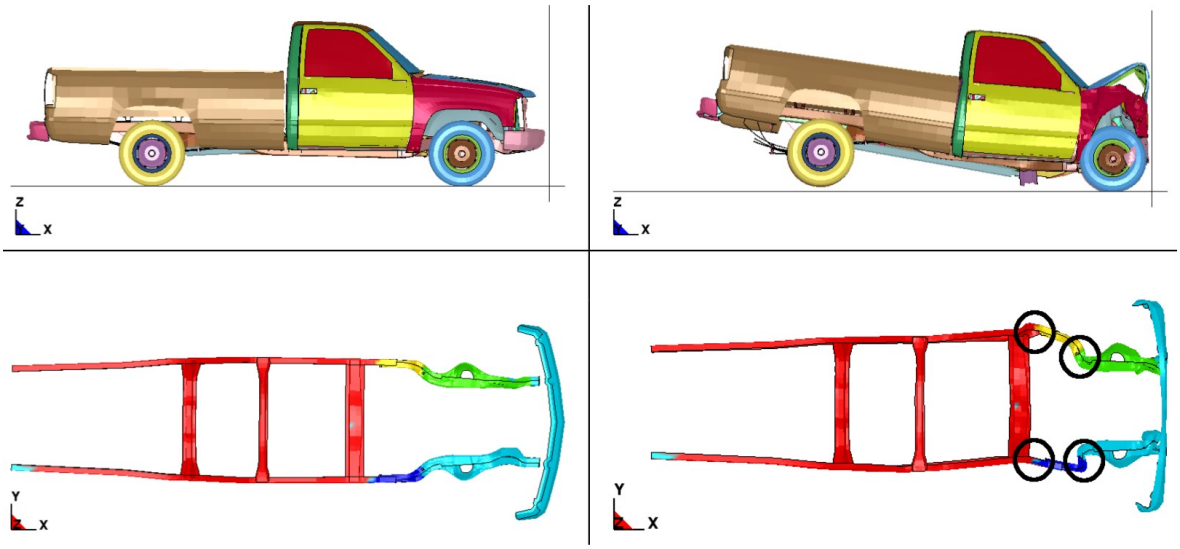
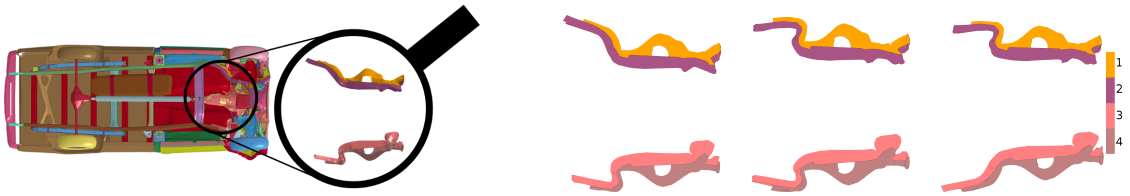


Abbildung 2: Der Chevrolet C2500 Pickup Truck. Die Bilder zeigen eine Simulation eines Frontalaufpralls vor dem Crash (links) und nach dem Crash (rechts). Für den Benchmark-Datensatz wurden die Blechdicken der 10 unten dargestellten (Draufsicht) Bauteile variiert. Rechts unten sind die charakteristischen Knickstellen der Längsträger markiert. Die Farbkodierung stellt hierbei eine berechnete Clusterzuordnung dar.

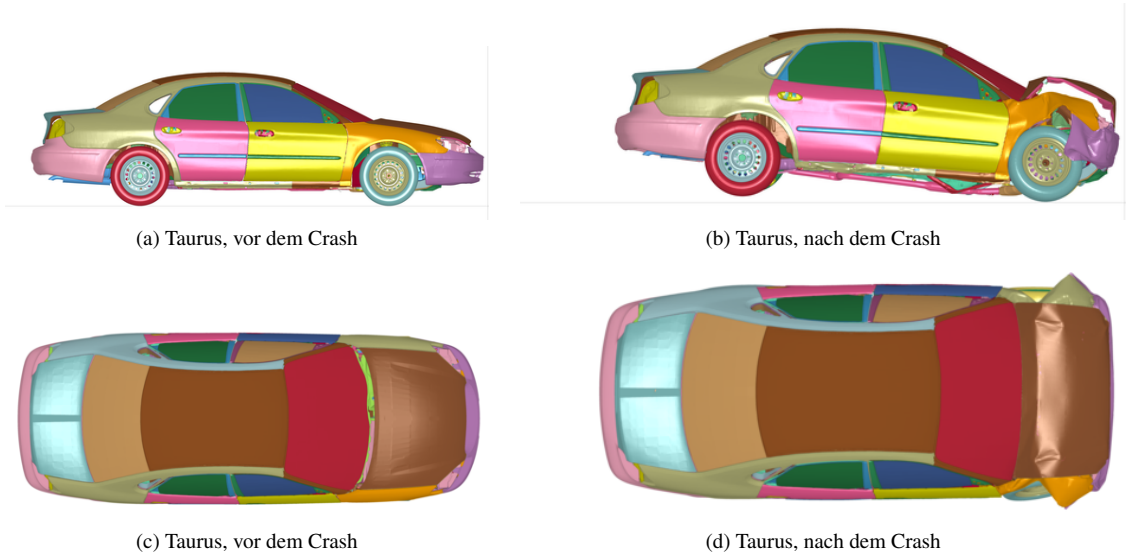


(a) Chevrolet Truck von unten

(b) Verschiedene Ausbeulungen der Längsträger durch Blechdickenvariation

Abbildung 3: (a) zeigt den Truck nach einer Simulation eines Frontalaufpralls. Die Längsträger sind vergrößert dargestellt. Drei verschiedene Biegeverhalten der Längsträger nach der Aufprallsimulation sind in (b) zu sehen. Die Farbskala dient zur Unterscheidung verschiedener physikalischer Teile. Es kann ein grundsätzlich unterschiedliches Crashverhalten beobachtet werden.

Vergleicht man verschiedene Simulationsläufe, so kann ein unterschiedliches Biegeverhalten der Längsträger festgestellt werden, siehe Abbildung 3 (b). Die ursprüngliche Einteilung in Bauteile, wie sie durch den Fertigungsprozess vorgegeben ist, kann das lokale Biegeverhalten und dessen Variabilität über die verschiedenen Simulationsläufe nur schlecht widerspiegeln.
Für die zwei folgenden Clusteringverfahren wird jeweils die Anzahl der Cluster sowie eine Distanzmatrix vorgegeben. Nach der Untersuchung der Eignung und Aussagekraft verschiedener Abstandsmaße konzentrierten wir uns auf Euklidische Distanzmatrizen. Auf die Problematik der Bestimmung der Anzahl der Cluster gehen wir nach Vorstellung der verwendeten Clusteringverfahren ein.



(a) Taurus, vor dem Crash

(b) Taurus, nach dem Crash

(c) Taurus, vor dem Crash

(d) Taurus, nach dem Crash

Abbildung 4: Abbildungen (a) und (c) zeigen den Taurus vor dem Crash zum Zeitpunkt $t = 0$. In (b) und (d) ist das durch den Crash verformte Modell zu sehen, siehe [Peh13].

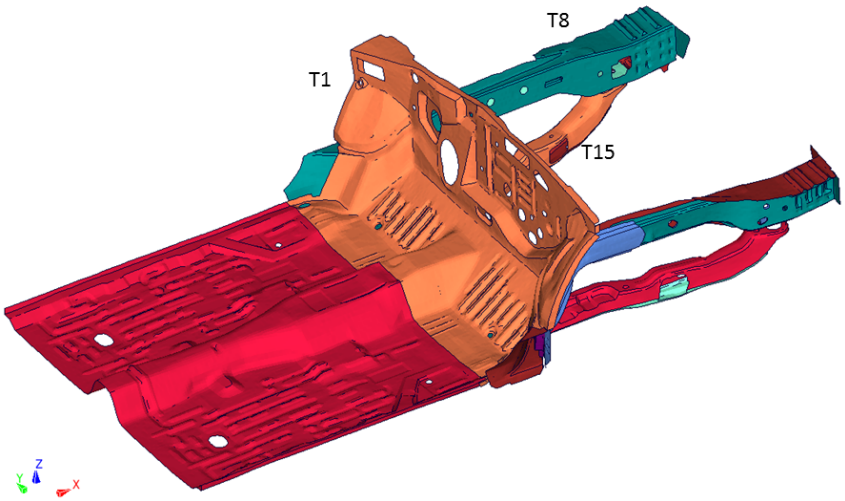


Abbildung 5: Ausschnitt des Taurus Modells. Für den Benchmark-Datensatz wurden die Blechdicken der 19 dargestellten Bauteile variiert.

3.2.1 Statistische Clustering Verfahren

Die große Herausforderung für klassische partitionierende oder hierarische Clusteringansätze in diesem Projekt war die hohe Dimension des Datensatzes. Aus diesem Grund wurde ein dreigliedriger Ansatz gewählt, der das Problem partitioniert und die Stärken der unterschiedlichen Methoden vorteilhaft einsetzen kann.
Die Hauptidee des partionierenden **k-means** Algorithmus ist die Minimierung des mittleren quadrierten Abstandes jedes Punktes zum nächsten Clustermittelpunkt. Sei hierfür $\mathcal{Y} := \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$, $\boldsymbol{e}_j \in \mathbb{R}^m$ ein Satz von n Datenpunkten und $k \leq n$ die Anzahl an

Clustern, so wird

$$\arg\min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k} \sum_{j=1}^{n} \min_{i\in\{1,\ldots,k\}} \|\boldsymbol{e}_j - \boldsymbol{\mu}_i\|_2^2,$$

minimiert, wobei $\boldsymbol{\mu}_i$ die Clustermittelpunkte sind. Die entsprechende Einteilung der Datenpunkte in k Cluster $C_1, \ldots, C_k$, geschieht dann über die sogenannten *Voronoi* Zellen $V_{\boldsymbol{\mu}_i}$ der Clustermittelpunkte. Folglich ist das Cluster C_i durch die Menge der Datenpunkte in $V_{\boldsymbol{\mu}_i}$ definiert, d.h. $C_i := \{\boldsymbol{e}_j \in V_{\boldsymbol{\mu}_i}\}$, mit $V_{\boldsymbol{\mu}_i} := \{z \in \mathbb{R}^m \mid \|z - \boldsymbol{\mu}_i\| \leq \|z - \boldsymbol{\mu}_j\| \; \forall j \in \{1, \ldots, k\}\}$. Um dieses Minimierungsproblem zu lösen existieren mehrere Ansätze. Eine der bekanntesten Heuristiken ist die Lösung mittels eines iterativen Schemas zum Finden von lokalen Minima. Im Rahmen dieses Projekts wurde hierzu ein effizienter Algorithmus verwendet, der auf der Filtermethode von Lloyd basiert [11, 9]. Dieses Vorgehen wird in der Literatur oft als “k-means Algorithmus” bezeichnet.
Obwohl die Prozedur aus [9] zu einer Lösung konvergiert, muss diese Lösung nicht das globale Minimum darstellen, sondern kann ein lediglich lokales Minimum sein. Da die Startwerte für die Clusterzentren im Algorithmus zufällig gewählt werden, können bei wiederholter Anwendung der Prozedur von [9] unterschiedliche Lösungen gefunden werden. Folglich ließen wir den Algorithmus mehrmals laufen und wählten die Lösung mit dem kleinsten Wert des Minimierungsterms. In unserer Konfiguration betrug die Wiederholungsrate 10. Der Algorithmus liefert insbesondere für den Fall, dass die Anzahl der Cluster signifikant kleiner als die Anzahl der Datenpunkte ist, ein schnelles und effizientes Clusteringverfahren.
Die zweite große Gruppe der Clusteringalgorithmen sind die **hierarischen** Methoden. Diese Verfahren basieren auf einem levelweisen Vorgehen und geben eine hierarische Repräsentation wieder, in der Gruppen oder einzelne Beobachtungen aus dem niedrigeren Level zusammengefügt werden. Basierend auf einem Distanz- oder Ähnlichkeitsmaß werden die zwei nächsten Cluster in ein gemeinsames Cluster zusammengeführt. Im Extremfall bestehen die Cluster im untersten Level aus nur jeweils einem Datenpunkt und im höchsten Level existiert lediglich ein Cluster, der alle Datenpunkte vereinigt.
Im Gegensatz zum k-means Algorithmus wählt man die Anzahl der Cluster nicht direkt, sondern legt stattdessen das Level fest, bei welchem der Algorithmus abgebrochen werden soll. Dieses Level korrespondiert aber (offensichtlich) eins-zu-eins zu einer Anzahl k an Clustern [8]. Zur Bestimmung von k kann ein Dedogramm, eine graphische Darstellung als Baum, hilfreich sein.
Ein großer Vorteil der hierarischen Clusteringalgorithmen ist die Tatsache, dass in der untersten Ebene nicht notwendigerweise einzelne Datenpunkte liegen müssen, sondern auch Gruppen von Daten zugelassen sind. Folglich ist es naheliegend die Idee auf mehrere Datensätze zu erweitern. Jeder Datensatz kann dann von einem separaten Clusteringalgorithmus, z.B. k-means, vorverarbeitet werden und mit Hilfe des hierarischen Clustering können dann Cluster unterschiedlicher Datensätze zusammengefügt werden. Diese neue Idee wurde im folgenden **kombinierten Ansatz** verfolgt.

1. Clustering der Positionsdaten ${}^r x_l^0$ (von r unabhängig, da ${}^1 x_l^0 = {}^2 x_l^0 = \ldots = {}^R x_l^0$) mit Hilfe von k-means. Damit bekommen wir ein von den physikalischen Bauteilen unabhängiges Clustering der unterliegenden Struktur. Die Anzahl der Cluster

$C_i^{Pos}, i = 1, \ldots, k_{Pos}$ ist hierbei in der Regel kleiner als die Anzahl der Bauteile (z.B. $k_{Pos} = 100$).

2. Clustering der Verschiebungsvektoren ${}^r d_l^{t_i,t_j}$ mit Hilfe von k-means in jedem Cluster C_i^{Pos}. Die daraus resultierenden Cluster bezeichnen wir mit $C_j^{Dis,(i)}, j = 1, \ldots, k_{Dis}^{(i)}$. Die Anzahl der Cluster $k_{Dis}^{(i)}$ variiert hierbei über die Cluster C_i^{Pos} und wird mit einer Faustregel festgesetzt (siehe unten zur Problematik der Bestimmung von k).

3. Zusammenführung der Cluster $C_j^{Dis,(i)}$ über Positionscluster hinweg mit Hilfe von hierarischem Clustering. Das finale Clustering bezeichnen wir mit $C_m, m = 1, \ldots, k$. Hier muss die Anzahl der finalen Cluster k erneut festgelegt, bzw. geschätzt werden.

3.2.2 Dünngitter-basierte Clustering Verfahren

Wir setzten Dünne Gitter für zwei konzeptionell unterschiedliche Clustering Verfahren ein. Zuerst entwickelten wir eine Out-of-Sample Extension für spektrales Clustering. Das entstandene Verfahren eignet sich inbesondere für die detailliertere Einteilung in Gruppen unter Berücksichtigung einer angemessenen Anzahl an Clustern. Dieses Verfahren ist jedoch zu langsam für große Datensätze mit mehreren Millionen Punkten. Für solche Datensätze haben wir ein Verfahren basierend auf Dichteschätzung entwickelt, welches zusätzlich einen Indikator für die Anzahl der Cluster liefert.
Beide Verfahren sind nicht-lineare Clustering Methoden, d.h. sie finden Cluster mit nicht-konvexer Form. Dass dies essentiell sein kann, zeigt exemplarisch Abbildung 6.

Dünne Gitter Im Folgenden werden Dünne Gitter mehrmals verwendet, daher werden in diesem Paragraphen kurz die dafür nötigen Konzepte und Notationen eingeführt. Details können z.B. in [2] und [Gar13] nachgeschlagen werden.
Wir definieren die i-te eindimensionale Hutfunktion des Levels l durch

$$\phi_{l,i}(x) := \phi(2^l \cdot x - i)|_{[0,1]} \quad \text{mit } \phi(x) := \max(1 - |x|, 0).$$

Zur Definition der multivariaten, s-dimensionalen, Hutfunktionen bilden wir das Tensorprodukt

$$\phi_{\mathbf{l},\mathbf{i}}(x_1, \ldots, x_s) := \prod_{k=1}^{s} \phi_{l_k,i_k}(x_k).$$

für Multiindizes $\mathbf{l}, \mathbf{i} \in \mathbb{N}^s$. Wir nennen den Punkt $x_{\mathbf{l},\mathbf{i}} := (2^{-l_1} i_1, \ldots, 2^{-l_s} i_s)$ Gitterpunkt zur Funktion $\phi_{\mathbf{l},\mathbf{i}}$. Mit den Indexmengen

$$I_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^s \;\middle|\; \begin{array}{ll} 0 \le i_j \le 1, & \text{falls } l_j = 0 \\ 1 \le i_j \le 2^{l_j} - 1,\ i_j \text{ ungerade} & \text{falls } l_j > 0 \end{array} \text{ für alle } 1 \le j \le s \right\} .$$

können wir nun die sogenannten hierarchischen Überschussräume durch

$$W_{\mathbf{l}} := \text{ span} \{\phi_{\mathbf{l},\mathbf{i}} \mid \mathbf{i} \in I_{\mathbf{l}}\}$$

definieren. Die Menge der Gitterpunkte, die dem Ansatzraum

$$V_t := \bigoplus_{|\mathbf{k}|_1 \leq t} W_{\mathbf{k}}$$

zugeordnet werden können, nennt sich (reguläres) dünnes Gitter des Levels t. Die in V_t enthaltenen Funktionen $\phi_{\mathbf{l},\mathbf{i}}$ bilden eine Basis dieses Funktionenraums.
Die Anzahl der Basisfunktionen im dünnen Gitter beträgt nur $\mathcal{O}(2^t \cdot t^{s-1})$. In vielen Anwendungen profitiert man dennoch von einer Approximationsrate ähnlich der der Vollgitterräume, siehe [2], [Gar13].
Durch ein einfaches Ersetzen der eindimensionalen Basisfunktion $\phi_{0,0}$ durch die konstante Funktion 1 und das Ersetzen des (eindimensionalen) Überschussraums W_0 durch die Räume

$$W_{-1} := \text{span}\{1\} \quad \text{und} \quad W_0 := \text{span}\{\phi_{0,1}\}$$

erhält man eine modifizierte Basis, die insbesondere für den bei der Dimensionsreduktion verwendeten dimensionsadaptiven Algorithmus geeignet ist. Der Levelindex -1 wird hierbei bei Normbildungen als 0 behandelt und dient lediglich der Unterscheidung der konstanten von den nicht-konstanten Funktionen.

Spectral Clustering mit Dünngitter-basierter Out-of-Sample Extension Spectral Clustering hat sich in der Vergangenheit als besonders erfolgreiches nicht-lineares Clustering Verfahren herausgestellt [17]. Es können jedoch große Datenmengen nur mit viel Aufwand bearbeitet werden, da ein Eigenproblem

$$\boldsymbol{L}\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y} \tag{2}$$

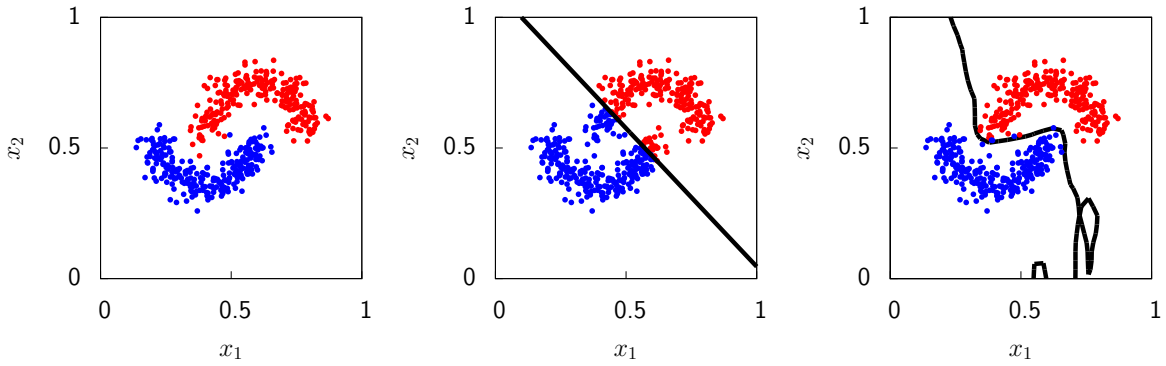
der Größe $n \times n$ gelöst werden muss, wobei n die Anzahl der Datenpunkte ist. Die Matrix $\boldsymbol{L}$ ist der Graph-Laplacian und die Matrix $\boldsymbol{D}$ beinhaltet auf der Diagonale die Ordnung der Datenpunkte im zugehörigen Ähnlichkeitsgraph. Damit ist die Komplexität dieses Verfahrens in $\mathcal{O}(n^3)$, was bereits für Datenmengen mit $n \approx 10,000$ zu aufwändig für die Praxis ist.
Unser Dünngitter-Ansatz lernt nicht direkt einen Vektor $\boldsymbol{y} \in \{0,1\}^n$, welcher angibt, zu welchem Cluster die Datenpunkte gehören, sondern eine Funktion $f : \mathbb{R}^d \to \{0,1\}$ mit welcher man die Komponenten von $\boldsymbol{y}$ approximieren kann. Die Dimension d ist hier ein freier Parameter, der a-priori zu wählen ist. Die Funktion f wird auf einem Dünnen Gitter diskretisiert, d.h.,

$$f(\boldsymbol{x}) = \sum_{|\mathbf{l}|_1 \leq t} \sum_{\mathbf{i} \in I_{\mathbf{l}}} \alpha_{\mathbf{l},\mathbf{i}} \phi_{\mathbf{l},\mathbf{i}}(\boldsymbol{x}) \,,$$

und benötigt dafür lediglich $\hat{n} = \mathcal{O}(2^t \cdot t^{d-1})$ Basisfunktionen [PPB11]. Das verallgemeinerte Eigenproblem (2) wird dann geändert zu

$$\boldsymbol{B}^T \boldsymbol{L} \boldsymbol{B} \boldsymbol{\alpha} = \lambda \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \boldsymbol{\alpha}$$

wobei $\boldsymbol{B}$ eine $n \times \hat{n}$ Matrix ist und damit das Eigenproblem nur noch von der Größe $\hat{n} \times \hat{n}$ ist. Da üblicherweise $\hat{n} \ll n$ gilt, insbesondere für große Datenmenge mit n im



(a) Daten (b) lineares Clustering (c) nicht-lineares Clustering

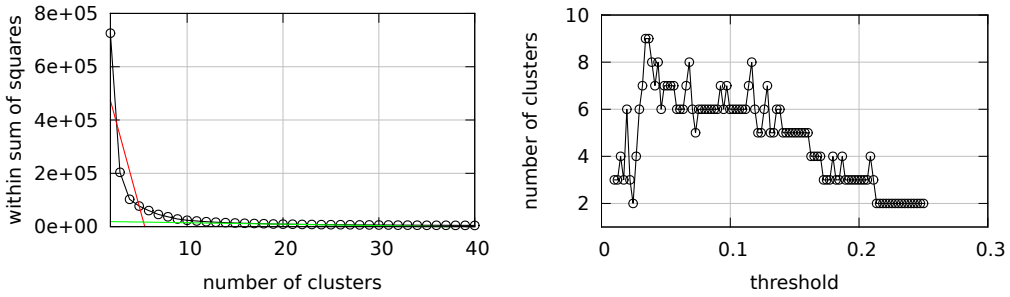
Abbildung 6: Lineare Clustering Verfahren haben Probleme den Two-Moons-Datensatz in (a) zu clustern [Peh13]. In Abbildung (c) sieht man auch, dass die Methode keine Aussagen treffen kann, wo keine Datenpunkte vorhanden sind (rechts unten).

Bereich von mehreren 100.000, verringert sich die Komplexität der Berechnung eines Spectral Clusterings daher erheblich von $\mathcal{O}(n^3)$ auf nur noch $\mathcal{O}(\hat{n}^3)$.
Mit den hier verwendeten adaptiven Dünnen Gittern kann die Anzahl der Gitterpunkte $\hat{n}$ gering gehalten werden, und trotzdem noch eine ausreichende Approximationsgenauigkeit mit der Funktion f erzielt werden. Neben den hier besprochenen Anwendungen im Bereich der Crash Test Simulationsdaten, hat sich dieses Verfahren auch als besonders erfolgreich im Bereich Image Segmentation gezeigt, wo ebenfalls große Datenmenge (Pixel) geclustert werden müssen, siehe, z.B., [PAPB13]. In sämtlichen Fällen hat sich eine Surplus-basierte Verfeinerungsstrategie als erfolgreich herausgestellt.

Dichte-basiertes Clustering mit adaptiven Dünnen Gittern Dichte-basierte Methoden folgen der Anschauung, dass ein Cluster eine sehr dichte Region (mit vielen Datenpunkten) ist, die von einer weniger dichten Region (mit wenigen Datenpunkten) umgeben ist. Die dichten Regionen werden als Cluster bezeichnet und die Datenpunkte außerhalb der Cluster als Rauschen. Um diese dichten Regionen zu finden, wird die Wahrscheinlichkeitsdichte aus den Daten bestimmt. Hierzu verwenden wir einen Dünngitter-basierten Ansatz, welcher, ausgehend von einer Startlösung f_ϵ, die Funktion

$$\hat{f} = \underset{u \in V}{\operatorname{argmin}} \int_\Omega \left(u(\boldsymbol{x}) - f_\epsilon(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x} + \lambda \|\Lambda u\|_{L^2}^2. \tag{3}$$

für ein Glattheitsfunktional Λ und Funktionen u aus einem Dünngitterraum findet. Es sei an dieser Stelle auch angemerkt, dass eine Mehrgitter-Methode für Dünne Gitter und Finite Element Ansätze untersucht wurde, welche z.B. bei der Behandlung des Regularisierungsterms in (3) eingesetzt werden kann [Peh11a, Peh12b]. Es handelt sich hierbei um ein konvexes Minimierungsproblem, welches einfach gelöst werden kann. Die Kosten der Dichteschätzung steigen nur linear mit der Anzahl der Datenpunkte [PPB12, PFPB13].
Die Dichtefunktion gibt uns zwar an, wo die dichten Regionen sind, was wir jedoch bereits als “dicht” anerkennen, wird hierbei durch einen Schwellwert ϵ bestimmt. Die-



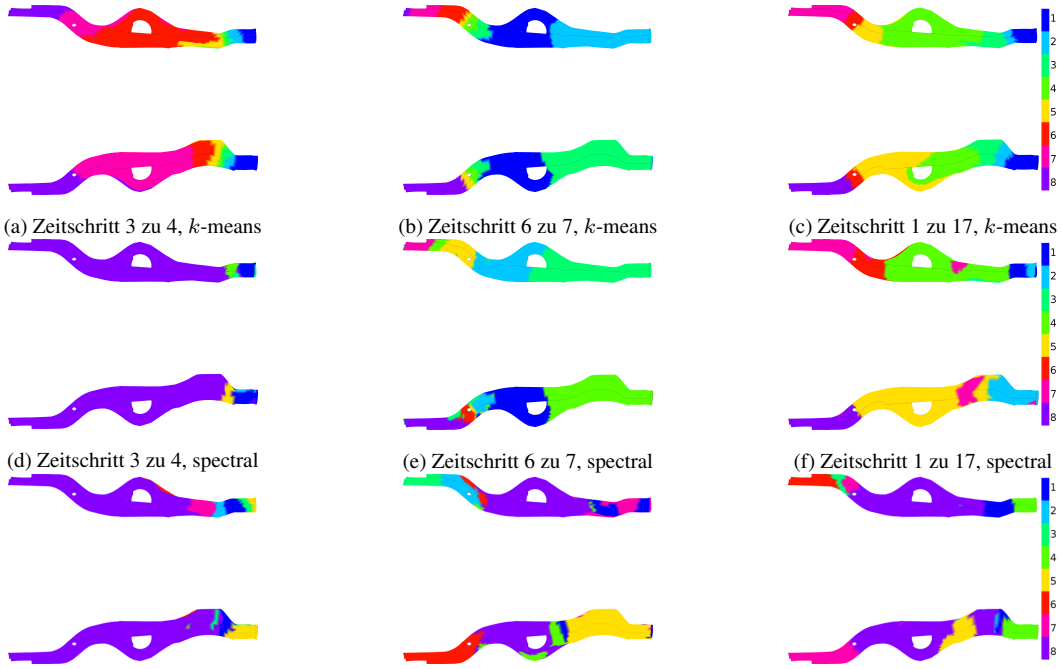
(a) L-Methode (b) Dichte-basierte Methode

Abbildung 7: Vergleich der geschätzen Anzahl von Clustern im Chevrolet Truck Datensatz mit der statistisch-motivierten L-Methode (a) und der Dichte-basierten Indikatormethode (b).

ser Schwellwert ist ein Parameter unseres Dichte-basierten Clustering-Verfahrens und muss entsprechend gewählt werden. Hierzu gibt es zwei grundsätzliche Vorgehensweisen. Erstens, falls die Anzahl der Cluster bereits bekannt sind, kann der Schwellwert so gewählt werden, dass entsprechend viele Cluster gefunden werden (als “dicht” angesehen werden), oder, zweitens, falls die Anzahl der Cluster nicht bekannt ist, wird in einem Komponenten-Schwellwert-Graphen nach Regionen mit gleicher Steigung gesucht. Eine detailliertes Beispiel folgt in Kapitel 3.2.3.

3.2.3 Indikator für die Anzahl der Cluster

Wie bereits oben erwähnt, ist ein wichtiger Schritt im Clustering-Prozess das Bestimmen der Anzahl der Cluster. Die Anzahl, sowie das Clustering selbst, hängt immer stark vom aktuellen Anwendungsfall ab, daher sind auch sämtliche Verfahren Heuristiken. Anhand von vier Längsträgern der Daten des Chevrolet Trucks zeigen wir in Abbildung 7 zwei Methoden, welche sich in diesem Projekt als besonders hilfreich herausgestellt haben.
In Abbildung 7 (a) findet sich der Graph der so-genannten L-Method [13]. Hier wird der mittlere quadrierte Abstand jedes Punktes zum nächsten Clustermittelpunkt gegen die Anzahl der Cluster geplottet. Die Heuristik besagt nun, dass ein auftretender Knick die Anzahl der Cluster festlegt. In unserem Beispiel ist ein solcher Knick etwa bei 6–8 Clustern zu finden.
Auf ein ganz ähnliches Ergebnis kommt der Dichte-basierte Indikator, wie in Kapitel 3.2.2 besprochen. Hierzu wird der Schwellwert ϵ gegen die Anzahl der auftretenden Clusterzentren (Peaks in der Dichtefunktion) geplottet, siehe Abbildung 7 (b). Bleibt die Anzahl der Cluster für einen gewissen Bereich von Schwellwerten konstant, ist dies ein Indikator dafür, dass es sich um ein “stabiles” Clustering handelt. Ebenso wie bei der L-Methode findet man im Bereich von 6 Cluster solch ein Verhalten. Im kombinierten Ansatz in Abschnitt 3.2.1 wurde noch die sehr einfache “Faustregel” $k \approx \sqrt{n/2}$ [12] angewendet, wobei n die Anzahl der Datenpunkte ist.



(a) Zeitschritt 3 zu 4, k-means (b) Zeitschritt 6 zu 7, k-means (c) Zeitschritt 1 zu 17, k-means

(d) Zeitschritt 3 zu 4, spectral (e) Zeitschritt 6 zu 7, spectral (f) Zeitschritt 1 zu 17, spectral

(g) Zeitschritt 3 zu 4, dichte (h) Zeitschritt 6 zu 7, dichte (i) Zeitschritt 1 zu 17, dichte

Abbildung 8: Clustering mit k-means (oben), Spectral Clustering mit Dünngitter-basierter Out-of-Sample Extension (mitte), und Dünngitter-basiertes Clustering mit Dichteschätzung (unten).

3.2.4 Ergebnisse auf den Fahrzeug-Daten

Wie oben bereits erwähnt hat das Clustering in unserem Workflow zwei Bedeutungen. Zum einen dienen die geclusterten Daten als Input für die nichtlineare Dimensionsreduktion, welche im nächsten Kapitel beschrieben wird, und zum anderen können aber bereits durch Visualisierung der Cluster Muster in den Daten sichtbar werden. Dies soll in diesem Kapitel anhand des Chevrolet Trucks und Ford Taurus dargestellt werden.

Chevrolet Truck Zunächst betrachten wir vier Längsträger des Chevrolet Trucks, vgl. Abbildung 3. Ein entsprechendes Clustering dieser vier Bauteile ist in Abbildung 8 gezeigt. Wir betrachten die Verschiebungen von Zeitschritt 3 auf 4, von 6 auf 7, und von 1 auf 17. Das Clustering wurde jeweils mit k-means, Spectral Clustering, und der Dichte-basierten Dünngitter-Methode durchgeführt. Hierbei haben wir die L-Methode und den Dichte-basierten Indikator zur Schätzung der Anzahl der Cluster verwendet. Details zur Parameterfindung können in [BGIT+13] nachgeschlagen werden.
Die Abbildung 8 zeigt deutlich, dass sich das Clustering über die Zeit verändert. So ergeben sich von Zeitschritt 3 auf 4 (erste Spalte) im vorderen (rechts) Bereich der Träger viele kleine Cluster, welche in den Verschiebungen von 6 auf 7 (mittlere Spalte) völlig verschwinden. Eine Mischung zwischen beiden erhält man wenn man sämtliche Verschiebungen von Zeitpunkt 1 bis zum Endzeitpunkt 17 betrachtet (letzte Spalte).
Des Weiteren gibt es Unterschiede zwischen den Clustering Verfahren. So finden sämtliche Verfahren für Zeitschritt 1 auf 17 ein sehr ähnliches Clustering, jedoch zeigt k-means ein unterschiedliches Verhalten zu Spectral und Dichte-basiertem Clustering

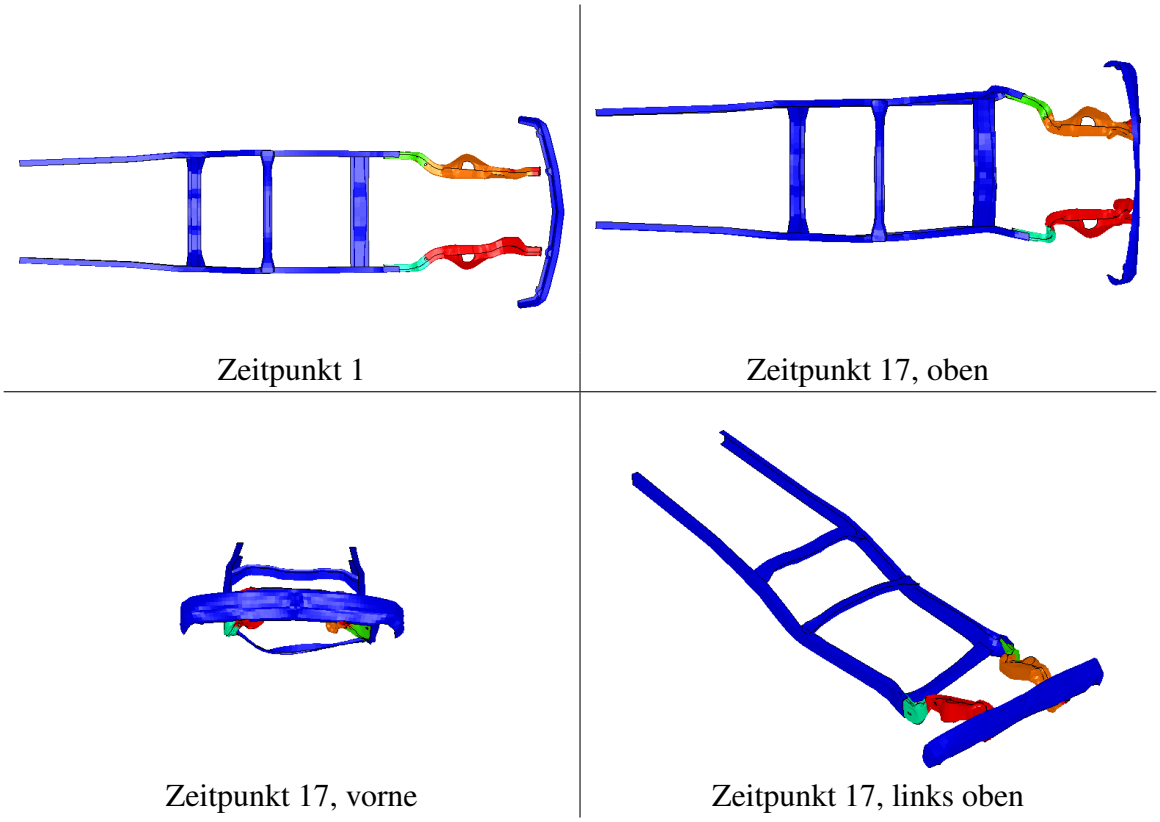
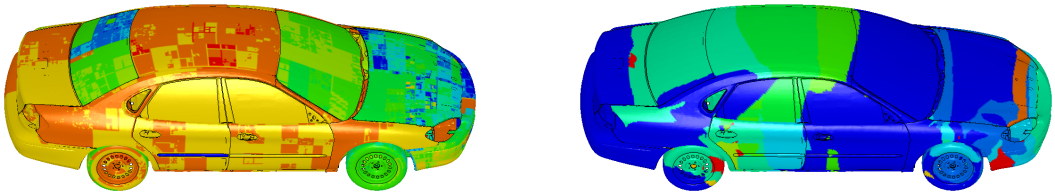


Abbildung 9: Spectral Clustering des Grundgerüsts des Chevrolet Trucks. ($k = 8$)

in den Zeitschritt 3 auf 4. Während k-means hier auch die hinteren Teile der Träger weiter unterteilt, geschieht dies bei Spectral Clustering und Dichte-basiertem Clustering nicht. Ähnliche Unterschiede können von Zeitschritt 6 auf 7 erkannt werden.
Mit solchen und ähnlichen Analysen, ist es für den Ingenieur schnell möglich, einen Überblick über das Crashverhalten zu erhalten. So wird z.B. hier durch das Clustering klar, dass sich von Zeitschritt 3 auf 4 nur die vorderen Teile verschieben, während von Zeitpunkt 6 auf 7 eher die hinteren Bereiche verbogen werden. Eine direkte Visualisierung sämtlicher Simulationsläufe wäre deutlich zeit- und kostenintensiver.
Ein weiteres Clustering haben wir für die ersten 10 Bauteile, und damit die Grundstruktur des Trucks, visualisiert, siehe Abbildung 9. Dort ist sehr gut sichtbar, dass die Clustergrenzen genau dort gewählt worden sind, wo verschiedene Knickverhalten auftreten.

Taurus Für den Ford Taurus wurden ähnliche Ergebnisse wie für den Chevrolet Truck erzielt. Durch die Größe und die damit einhergehende große Anzahl der Cluster ($k = 150$), ist jedoch eine Visualisierung des Modells nicht mehr so einfach möglich. In Abbildung 10 sind je ein Clustering der spektralen Methode und eines des k-means Algorithmus für den gesamten Taurus abgebildet. Eine genauere Analyse und Auswirkung der Clusterings auf die nichtlineare Dimensionsreduktion wird in den nächsten Abschnitten durchgeführt.



(a) k-means (b) Dünne Gitter

Abbildung 10: Visualisierung des Clusterings für die Taurus Daten. (k-means: $k = 150$, spectral: $k = 129$)

3.3 Nichtlineare Dimensionsreduktion (Stufe 2)

Die Verschiebungsvektoren eines Clusters führen nun zu C (Anzahl der Cluster) unabhängigen Dimensionsreduktionsproblemen mit R Input-Vektoren. Die Länge $D_c = 3M_c$ der Vektoren eines einzelnen Problems ergibt sich hierbei durch die Anzahl der FE-Knoten M_c im aktuellen Cluster c.
Unabhängig von der Dimension der Verschiebungsvektoren eines Clusters wurde zunächst eine lineare Dimensionsreduktion mittels einer Hauptachsenzerlegung (auch *Principal Component Analysis (PCA)* genannt) durchgeführt. Durch das clusterweise Vorgehen sowie die Anwendung der PCA waren wir, selbst mit einem Varianzerhalt von über 99%, in der Lage bereits moderate Problemdimensionen von ca. $10 - 50$ zu erzielen.
Die aus der PCA resultierenden R Vektoren $y_r, r = 1, \ldots, R$ der Dimension $d_c < D_c$ dienten nun als Input-Daten für die nichtlinearen Dimensionsreduktionsverfahren, welche die Dimension der Vektoren bis auf $s < d_c$ reduzieren.

3.3.1 Lokal lineare Approximation

Bei der lokal linearen Approximation (LLA) wird die zugrunde liegende s-dimensionale Mannigfaltigkeit $\mathcal{M}$ durch die Vereinigung der linearen Tangentialräume $T_{y_r}\mathcal{M}$ an den Vektoren $y_r, r = 1, \ldots, R$ approximiert. Insbesondere nutzt LLA dabei lokalisierte Informationen.
Dafür werden jeweils die geodätischen Abstände, das heißt die Abstände auf der Mannigfaltigkeit, zwischen zwei Punkten y_{r_1} und $y_{r_2}, r_1,\ r_2 = 1, \ldots, R$ benötigt. Da die tatsächliche Mannigfaltigkeit unbekannt ist, können diese Abstände nicht genau berechnet werden und müssen angenähert werden. Sei $Y = [y_1, \ldots, y_R] \in \mathbb{R}^{d_c \times R}$ die Matrix der dimensionsreduzierten Verschiebungsvektoren. Der Adaptive Nearest Neighbor Algorithmus [7] bildet aus den quadrierten paarweisen euklidischen Abständen der Punkte einen Nachbarschaftsgraphen. In diesem Graphen werden kürzeste Wege zwischen den Knoten gesucht. Unter geeigneten Annahmen an die Mannigfaltigkeit ([15]) approximiert dabei solch ein kürzester Pfad den Weg zwischen zwei Punkten auf der Mannigfaltigkeit. Somit ist die Länge eines kürzesten Weges eine Näherung an den geodätischen Abstand zwischen diesen Punkten.
Nun kann der Tangentialraum in jedem Punkt y_r mit dem gewichteten Hauptkompo-

nentenraum der (bezüglich kürzester Wege im Graphen) k-nächsten, in y_r zentrierten Nachbarn berechnet werden. Sei Y_r die Matrix, die aus den k Spalten von Y besteht, die den k-nächsten Nachbarn von y_r entsprechen. Dann wird die Singulärwertzerlegung

$$U_r \Sigma_r V_r^T = (Y_r - [y_r, \ldots, y_r]) \operatorname{diag}(w_r)$$

berechnet, wobei w_r ein Gewichtungsvektor ist, dessen j-ter Eintrag dem Graphenabstand zwischen y_r und der j-ten Spalte von Y_r entspricht, und $\operatorname{diag}(w_r)$ eine Diagonalmatrix mit w_r auf der Diagonalen ist. Nun spannen die s ersten Spalten von U_r eine Approximation $\tilde{T}_{y_r}\mathcal{M}$ an den Tangentialraum $T_{y_r}\mathcal{M}$ auf. Damit wird die Mannigfaltigkeit durch

$$\tilde{\mathcal{M}} = \bigcup_{r=1}^{R} \rho_r(\mathcal{V}_r)$$

angenähert, wobei $\mathcal{V}_r = \{z \in \mathbb{R}^{d_c} : \|z - y_r\| \leq \|z - y_{r'}\| \quad \forall r' = 1, \ldots, R\}$ die Voronoi-Zelle von y_r ist. In ihr sind alle Punkte z enthalten, die näher an y_r liegen als an allen anderen Punkten $y_{r'}$. Für die Abbildung ρ_r gilt

$$\begin{aligned} \rho_r : \mathbb{R}^{d_c} &\to y_r + \tilde{T}_{y_r}\mathcal{M} \\ z &\mapsto \operatorname*{arg\,min}_{q \in y_r + \tilde{T}_{y_r}\mathcal{M}} \|q - z\|_2, \end{aligned}$$

das heißt ein Punkt z aus der Voronoi-Zelle wird auf den nächsten Punkt auf $y_r + \tilde{T}_{y_r}\mathcal{M}$ abgebildet. Für die Darstellung eines neuen Testdatenpunkts aus $\mathbb{R}^{d_c}$ wird die Voronoi-Zelle des nächstgelegenen Datenpunkts verwendet.
Wir nutzen LLA als einfaches, heuristisches Verfahren, das zwar keine echte Parametrisierung und damit keine echte Mannigfaltigkeit liefert, aber eine einfache Möglichkeit darstellt, die Rekonstruktionseigenschaften von neuen Datensätzen mit Hilfe von bestehenden zu untersuchen.

3.3.2 Principal Manifold Learning mit dimensionsadaptiven dünnen Gittern

Die von uns verwendete Methode des Principal Manifold Learnings (PML) basiert auf der Minimierung des regularisierten Kostenfunktionals

$$\mathcal{A}(f) = \frac{1}{R} \sum_{r=1}^{R} \|y_r - \min_{t_r \in [0,1]^s} f(t_r)\|^2 + \lambda \sum_{j=1}^{d_c} |f^{(j)}|_{H^1_{\mathsf{mix}}([0,1]^s)}. \tag{4}$$

Hierbei ist $f : [0,1]^s \to \mathbb{R}^{d_c}$ eine vektorwertige Funktion aus einem zu spezifizierenden Ansatzraum V mit Komponenten $f^{(j)}, j = 1, \ldots, d_c$. Die Einbettungsdimension s ist apriori zu wählen. Der regularisierende Strafterm entspricht hierbei der Summe der H^1_{mix} Sobolev-Seminormen der reellwertigen Funktionen $f^{(j)}$

$$|f^{(j)}|_{H^1_{\mathsf{mix}}([0,1]^s)} := \sum_{|(a_1,\ldots,a_s)|_\infty = 1} \| \frac{\partial^{a_1}}{\partial x_1} \cdots \frac{\partial^{a_s}}{\partial x_s} f^{(j)} \|_{L_2([0,1]^s)}.$$

Diese Seminorm korrespondiert zum Sobolevraum mit gemischten Ableitungen erster Ordnung $H^1_{\mathsf{mix}}([0,1]^s) = H^1([0,1]) \otimes \ldots \otimes H^1([0,1])$. Die Gewichtung der Regularisierung findet über den Parameter $\lambda > 0$ statt. Einer der großen Vorteile der Principal Manifold Learning Methode ist der inherent generative Ansatz, d.h. es wird eine Parametrisierung der den Input-Daten zugrundeliegenden Mannigfaltigkeit erstellt. Dadurch ist es nativ möglich, jedem niederdimensionalen Punkt $t \in [0,1]^s$ - auch wenn dieser nicht zum ursprünglichen Datensatz gehört - einen hochdimensionalen Punkt $f(t)$ zuzuordnen.
Zur Darstellung einer vektorwertigen Funktion wurde jede Bildkomponente einzeln behandelt. Im Rahmen dieses Projekts wurde eine Variante der PML-Methode entwickelt, die als Ansatzraum V einen Vektor aus Komponentenfunktionen aus einem dimensionsadaptiven Dünngitterraum verwendet.

Dimensionsadaptive dünne Gitter Aufbauend auf Abschnitt 3.2.2 werden wir im Folgenden kurz den dimensionsadaptiven Algorithmus beschreiben. Er basiert auf uns zur Verfügung stehender Software Software für die Repräsentation, Speicherung und Verarbeitung von Funktionen und diskreten Simulationsdaten mittels der Technik der adaptiven dünnen Gitter. Hier wird die optimale Diskretisierung für die jeweilige Komponentenfunktion durch abwechselnde Verfeinerungs- und Kompressionsschritte a-posteriori bestimmt. Dazu werden konkret folgende Schritte ausgeführt:

- Minimiere (4) auf dem aktuellen Gitter.
- Iteriere über alle $W_{\mathbf{l}}$ im aktuellen Ansatzraum: Lösche $W_{\mathbf{l}}$ aus dem aktuellen Ansatzraum, falls sämtlichen Basisfunktionen in $W_{\mathbf{l}}$ Koeffizienten zugewiesen sind, deren Betrag kleiner oder gleich einer Schranke ε ist und sofern sämtliche feineren Räume $W_{\mathbf{l}+\mathbf{k}}$ aus dem aktuellen Ansatzraum für einen belibigen Multiindex $\mathbf{k} \neq \mathbf{0}$ ebenfalls diese Bedingung erfüllen.
- Minimiere (4) auf dem neuen Gitter.
- Iteriere über alle $W_{\mathbf{l}}$ im neuen Ansatzraum: Falls mindestens ein Koeffizient einer Basisfunktion aus $W_{\mathbf{l}}$ einen größeren Betrag als ε aufweist, füge die Teilräume $W_{\mathbf{l}} + \mathbf{e}_j$ für sämtliche $j = 1, \ldots, s$ zum Ansatzraum hinzu, für die $l_j \neq -1$ ist. Hierbei bezeichnet $\mathbf{e}_j = (0, 0, \ldots, 0, 1, 0, \ldots, 0, 0)$ den Einheitsmultiindex in Koordinatenrichtung j.

Dieses Vorgehen wird bis zu einem maximalen Verfeinerungslevel iteriert. Falls Kompressions- und Verfeinerungsschritt das dünne Gitter nicht mehr verändern, wird die Adaptionsschleife ebenfalls abgebrochen. Die Bedingung $l_j \neq -1$ im letzten Schritt der Adaptionsschleife charakterisiert genau die Richtungen, in welchen die Ansatzfunktionen aus $W_{\mathbf{l}}$ konstant sind. In [5] wurde gezeigt, dass die hier verwendeten Dünngitterräume eine native Zerlegung in konstante und nicht-konstante Richtungen einer Funktion erlauben, die mit einer diskretisierten sogenannten *Anker-ANOVA-Zerlegung* mit Anker 0 übereinstimmt. Aus diesem Grund führt die oben erläuterte Adaptionsstrategie zur (approximativen) Anker-ANOVA-Zerlegung der Komponentenfunktionen und

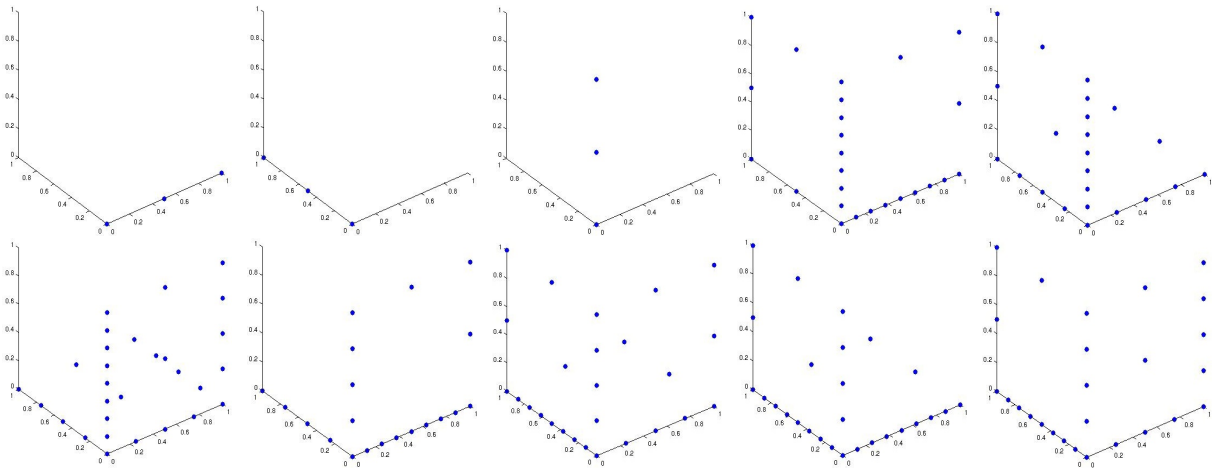


Abbildung 11: Die resultierenden dreidimensionalen dünnen Gitter zu den (nach initialer PCA) 10 Komponentenfunktionen zur Beschreibung der Verschiebungsvektoren eines Längsträgers des Chevrolet Trucks zwischen Anfangs- und Endzeitpunkt ($t_i = 1, t_j = 17$).

wir sind in der Lage die sogenannte *Superpositionsdimension* a-posteriori zu bestimmen. Dies ist die maximale Anzahl von Variablen, deren Interaktion für die Darstellung einer Funktion relevant ist. Für eine detaillierte Erklärung der Anker-ANOVA-Zerlegung und der Zusammenhänge zum dimensionsadaptiven Algorithmus sei auf [BG12] verwiesen.
Ein Beispiel zur Detektion der Superpositionsdimension der Mannigfaltigkeit der Input-Daten ist in Abbildung 11 dargestellt. Hierzu wurde zu den Verschiebungsvektoren des Längsträgers 1 des Chevrolet Trucks (siehe Abbildung 3 (b)) mittels des dimensionsadaptiven PML-Algorithmus die zugrundeliegende Mannigfaltigkeit rekonstruiert. Die initiale Dimension wurde hierzu als $s = 3$ gewählt. Die dargestellten Gitter der Komponentenfunktionen weisen ausschließlich Punkte auf, bei denen mindestens eine Koordinate 0 ist. Dies ist bei unseren Ansatzräumen äquivalent dazu, dass die Komponenten als Summe von Funktionen von zwei statt aller drei Variablen dargestellt werden können und die Superpositionsdimension somit 2 ist.

Numerische Minimierung des Zielfunktionals Zur Evaluation des Funktionals $\mathcal{A}$ ist zu jedem Datenpunkt y_r das Minimum der Norm über alle Urbildpunkte zu bilden. Daher wurde bei der numerischen Minimierung von $\mathcal{A}$ ein alternierender Ansatz verwendet, welcher abwechselnd für feste Urbildpunkte t_r mittels eines schnellen Lösers für lineare Gleichungssysteme auf dünnen Gittern nach f minimiert und anschließend für festes f mittels eines modifizierten Downhill-Simplex-Verfahrens nach den Punkten t_r minimiert. Diese Herangehensweise ist eng verwandt mit dem aus der Statistik bekannten expectation-maximization Schema.

3.3.3 Diffusion Maps

Der Begriff “Diffusion Maps” bezeichnet eine Klasse von Dimensionsreduktionsverfahren, welche eine niedrigdimensionale Darstellung des Datensatzes unter Beibehaltung lokaler Beziehungen der Datenpunkte zueinander suchen. Ziel dieser Verfahren ist es, die lokale Struktur der Originalinformation bei der Einbettung möglichst zu erhalten.
Sei (X, A, μ) ein Maßraum, wobei X hier den Datensatz bezeichnet und μ die Datenverteilung repräsentiert. Um die lokalen Nachbarschaftsbeziehungen des Datensatzes oder die sogenannte Geometrie der Daten darzustellen wird eine Kernfunktion $k : X \times X \to \mathbb{R}$ definiert, welche symmetrisch $(k(x, y) = k(y, x))$ und positiv $(k(x, y) \geq 0)$ ist. Ein bestimmter Kern repräsentiert dann ein konkretes Ähnlichkeitmaß für die Daten. So wird beispielsweise der Gauss-Kern $k(x, y) = \exp(-\|x - y\|^2/\varepsilon)$ in der Bildverabeitung und der Graphentheorie verwendet.
Einem Kern auf X kann ein Graph zugeordnet werden, welcher als eine Markov-Kette (eine spezielle Klasse von stochastischen Prozessen) betrachtet werden kann. Hierzu definieren wir $d(x) = \int_X k(x, y)d\mu(y)$ und $p(x, y) = k(x, y)/d(x)$. p stellt hierbei den Übergangskern der Markov-Kette dar. Es folgt, dass der Operator $Pf(x) = \int_X p(x, y)f(y)d\mu(y)$ ein Diffusionsoperator ist.
Die Informationen über die Geometrie des Datensatzes (definiert durch die Nachbarschaftsbeziehungen), die P enthält, können über eine Eigendekomposition extrahiert werden. Hierzu sei die diskrete Folge von Eigenwerten $\lambda_l(l \geq 0)$ und Eigenfunktionen $\phi_l(l \geq 0)$ so geordnet, dass $1 = \lambda_0 > |\lambda_1| \geq |\lambda_2| \ldots \geq 0$ mit $P\phi_l = \lambda_l\phi_l$ gilt.
Die Veränderungen des Diffusionsprozesses und somit die Entwicklung der oben dargestellten Markov-Kette über die Zeit korrespondiert direkt mit den Potenzen des Operators P. Somit lässt sich der Kern $p_t(x, y)$, welcher die t-te Potenz P^t des Operators P beschreibt, auch als die Übergangswahrscheinlichkeit von x nach y im Intervall $[0, t]$ interpretieren.
Durch diese Betrachtung können Informationen über relevante geometrische Strukturen des Datensatzes auf unterschiedlichen Skalen extrahiert werden. Diese Interpretation beruht auf dem Verhalten des Spektrums der verschiedenen Potenzen von P. Da sich der Rang von P^t mit wachsendem t reduziert, entstehen dadurch, dass man unterschiedlich detaillierte Sequenzen erhält, mehrere Skalen.

Die Diffusionsmetrik Eine natürliche Konsequenz der Betrachtung von $p_t(x, y)$ ist die Relevanz des Diffusionsabstandes, der die Distanz zwischen zwei Übergangsdistributionen ausdrückt:

$$D_t(x, y) \triangleq \|p_t(x, .) - p_t(y, .)\|^2_{L^2(X, d\mu/\pi)} = \int_X (p_t(x, u) - p_t(y, u))^2 \frac{d\mu(u)}{\pi(u)}.$$

Dabei ist $\pi(y) = \frac{d(y)}{\sum_{z \in X} d(z)}$ die stationäre Verteilung der Markov-Kette.
Der Abstand $D_t(x, y)$ kann als Summe über alle Verbindungen zwischen x und y betrachtet werden. Er gibt daher die Konnektivität der Daten wieder und verhält sich robust gegenüber Rauschen. Die Metrik ist somit geeignet, um Strukturinformationen über den Datensatz zu gewinnen, da sämtliche Beziehungen zwischen x und y einbezogen werden.

Ein bemerkenswertes Ergebnis (siehe [3]) für einen Beweis) ist die Tatsache, dass dieser Diffusionsabstand aus den Eigenfunktionen ϕ_l und Eigenwerten λ_l von P^t berechnet werden kann:

$$D_t(x,y) = (\sum_{l\geq 1} \lambda_l^{2t}(\phi_l(x) - \phi_l(y))^2)^{1/2}$$

Hierbei gilt es zu beachten, dass die Eigenwerte der Größe nach sortiert sind. Deshalb kann diese Summe abhängig von der gewünschten Präzision vorzeitig trunkiert werden.
Um nun, unter Berücksichtigung der Struktur der Ursprungsdaten, eine Einbettung in $\mathbb{R}^s$ (mit $s < n$) zu erhalten, definiert man die Abbildung $\Phi : \mathbb{R}^n \to \mathbb{R}^s$ durch

$$\Phi(x) = (\lambda_1\phi_1(x), \lambda_2\phi_2(x), ..., \lambda_s\phi_s(x)).$$

Komponenten mit kleinen Einträgen (bzw. Eigenwerten) werden vernachlässigt. Abhängig von der Wahl des Kerns genügen für viele Probleme nur wenige Eigenwerte, um mit dieser Einbettung bereits wesentliche Informationen aus dem Datensatz zu extrahieren. Dies erklärt die erfolgreiche Anwendung solcher Ansätze in vielen Disziplinen.

3.4 Analyse der eingebetteten Daten und ingenieursmäßige Interpretation der Ergebnisse (Stufen 3,4)

3.4.1 Rekonstruktionsfehler

Für die mit lokal linearer Approximation und Principal Manifold Learning durchgeführten Rekonstruktionsexperimente, wählten wir zu Beginn randomisiert eine Menge von Testindizes $I_{\text{test}} \subset \{1, \ldots, R\}$. Die korrespondierenden Simulationsläufe wurden nicht als Input für den jeweiligen Algorithmus, sondern lediglich zur Evaluation des Fehlers nach Abschluss der Berechnungen verwendet. Als Input dienten somit die Verschiebungsvektoren $\mathcal{Y} := \{{}^r\mathbf{d}^{t_i,t_j} \mid r \in I_{\text{train}}\}$ mit $I_{\text{train}} := \{1, \ldots, 126\} \setminus I_{\text{test}}$ für zwei fixierte Zeitschritte t_i, t_j.
Zur Rekonstruktion der Testdatensätze können beim PML-Verfahren die auf die Mannigfaltigkeit projizierten Testdaten mittels der Funktion f wieder in den hochdimensionalen Raum zurücktransformiert werden, siehe z.B. [6]. Bei der lokal linearen Approximation (LLA) geschieht dies durch die Bestimmung des zu einem Testdatums nächstgelegenen Nachbarpunkts der Trainingsdatenmenge und anschließender linearer Approximation in dessen Voronoi-Zelle.
Als Maß für den Rekonstruktionsfehler pro Knoten n_l wählten wir

$$\frac{1}{|I_{\text{test}}|} \sum_{r \in I_{\text{test}}} \left|\left| \overline{{}^r d_l^{t_i,t_j}} - {}^r d_l^{t_i,t_j} \right|\right|^2,$$

wobei $\overline{{}^r d_l^{t_i,t_j}}$ die rekonstruierte Verschiebung des Knotens l zwischen den Zeitschritten t_i und t_j bezeichnet. Dieser absolute Fehler dient dem Ingenieur zur Beurteilung der Güte der Rekonstruktion, indem er einen Verschiebungsbereich (z.B. 1 Millimeter) bestimmt, der tolerabel ist.

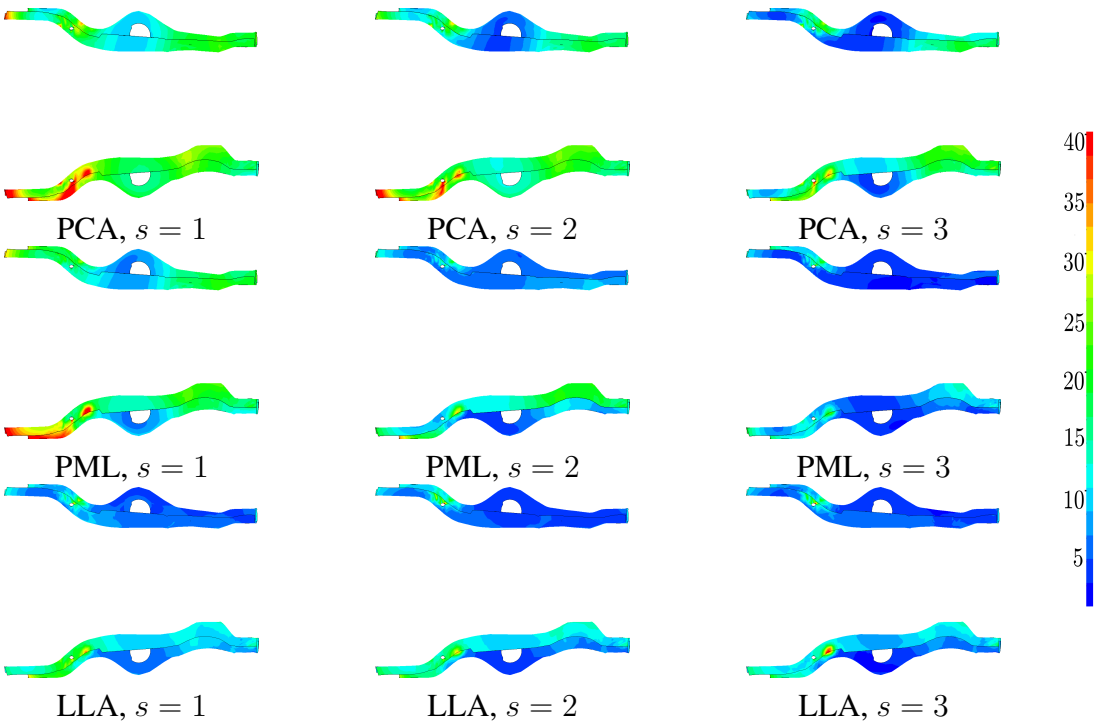


Abbildung 12: Mittlerer quadrierter Rekonstruktionsfehler (bis max. 40mm^2) für die Längsträger des Chevrolet Truck. Die Berechnungen wurden auf Basis der Bauteile (ohne Clustering) für $t_i = 7$ und $t_j = 6$ durchgeführt. Die Evaluation des Fehlers fand auf 10 zufälligen Testsimulationen statt.

Chevrolet Truck In Abbildung 12 sind exemplarisch die Ergebnisse der PML- und PCA-Methoden für $s = 1, 2, 3$ und $t_i = 6, t_j = 7$ dargestellt, welche ohne vorheriges Clustering berechnet wurden. Abbidlung 13 zeigt die Ergebnisse der LLA-, PML- und PCA-Methode für das k-means und das spektrale Dünngitter-Clustering. Für die nichtlinearen Verfahren ist deutlich erkennbar, dass der Fehler für $s = 2$ bereits so gering ist, dass er sich durch eine höhere Einbettungsdimension nicht wesentlich verändert. Der punktweise Fehler für die PCA ist hingegen für $s = 2$ im hinteren Bereich der Längsträger noch deutlich größer. Da für die PML-Methode eine echte Parametrisierung berechnet wird, kann somit gefolgert werden, dass den Daten eine zweidimensionale nichtlineare Mannigfaltigkeit zugrundeliegt, die das lineare Verfahren erst in drei Dimensionen darstellen kann.
Zwischen den Resultaten für die beiden Clustering-Verfahren sieht man lediglich geringe Unterschiede. So schneidet das spektrale Clustering beispielsweise für die PML Methode etwas besser ab als das k-means Clustering. Im Vergleich zu den Resultaten, die ohne vorheriges Clustering berechnet wurden, ist jedoch bei beiden Verfahren eine deutlich Verkleinerung des Fehlers, insbesondere im vorderen Bereich der Längsträger für $s = 2, 3$, zu erkennen.

Ford Taurus In Abbildung 14 sind die Ergebnisse für den Simulationssatz des Ford Taurus (mit großen Input-Parameteränderungen) zu sehen. Insbesondere im hinteren Bereich sowie für $s = 2, 3$ auch auf der Stirnwand wird der geringere Rekonstruktions-

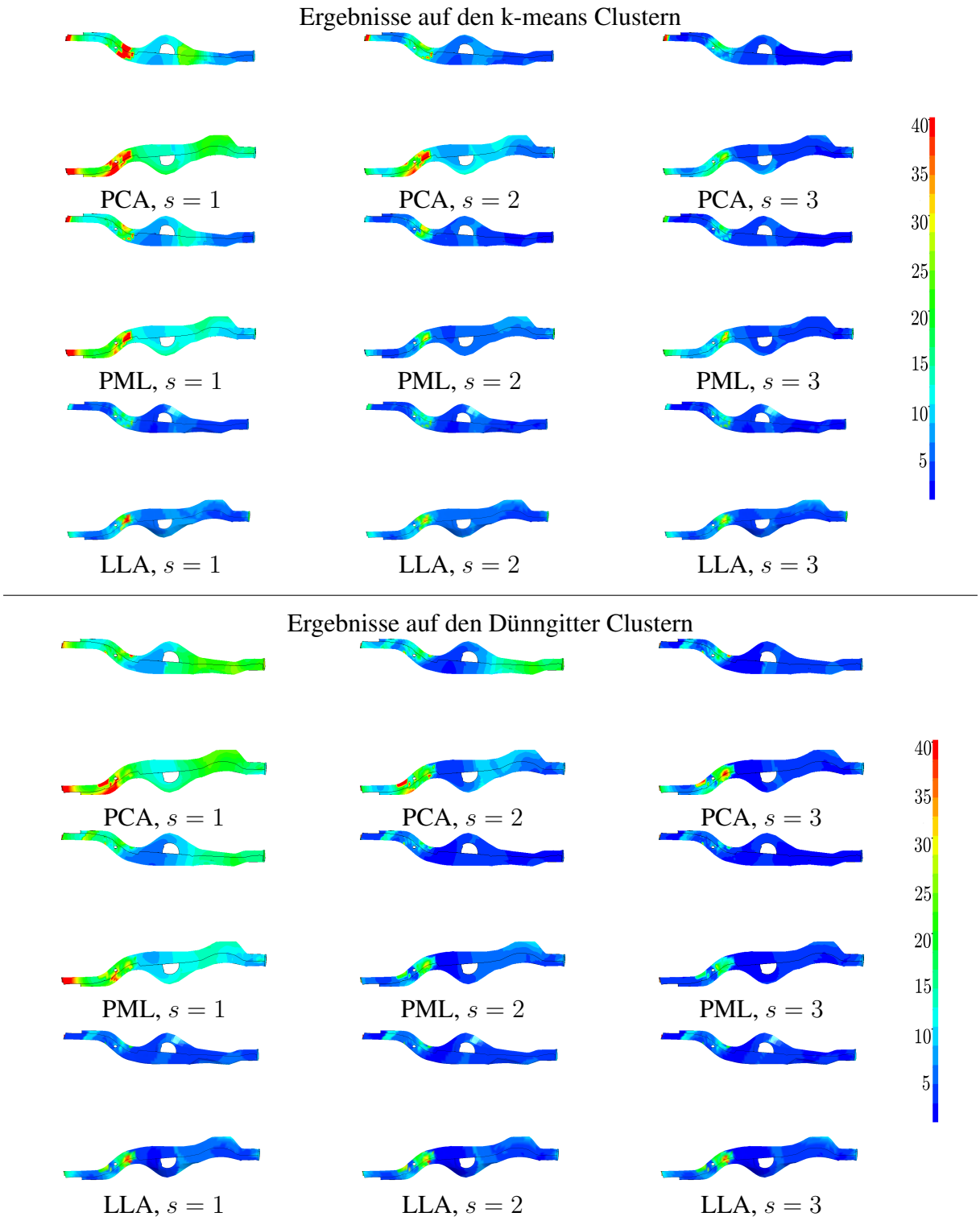
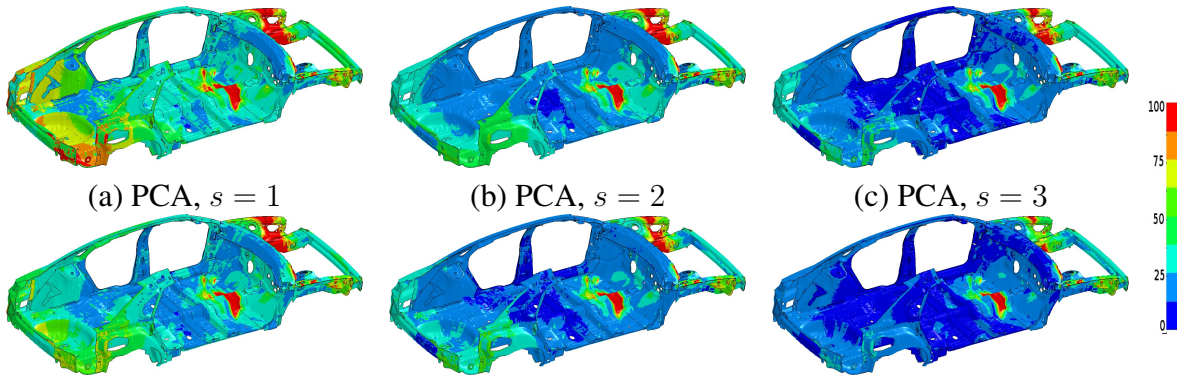


Abbildung 13: Mittlerer quadrierter Rekonstruktionsfehler (bis max. 40mm^2) für die Längsträger des Chevrolet Truck. Die Berechnungen wurden auf Basis von k-means und dem spektralen Dünngitter-Clustering (jeweils 8 Cluster) für $t_i = 7$ und $t_j = 6$ durchgeführt. Die Evaluation des Fehlers fand auf 10 zufälligen Testsimulationen statt.

fehler der nichtlinearen Methode im Vergleich zur PCA deutlich. Lediglich im vorderen Bereich der Bodenplatte ist ein größerer Bereich, in welchem der Rekonstruktionsfehler auch mit dem nichtlinearen Verfahren für $s = 3$ noch deutlich größer als 10 cm. Dies ist insbesondere auf das dortige, punktweise sehr unterschiedliche Biegeverhalten zurückzuführen, das auch von unseren Clusteringmethoden nicht adäquat aufgelöst werden kann.



(a) PCA, $s = 1$ (b) PCA, $s = 2$ (c) PCA, $s = 3$

(d) PML, $s = 1$ (e) PML, $s = 2$ (f) PML, $s = 3$

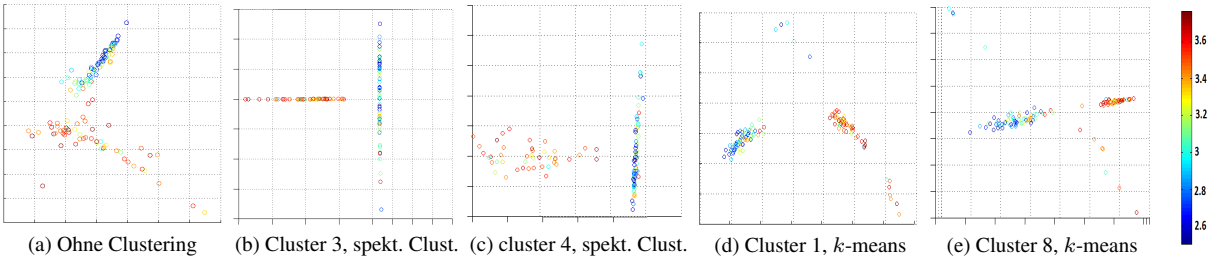
Abbildung 14: Mittlerer quadrierter Rekonstruktionsfehler (bis max. 100mm^2) für das Grundgerüst des Ford Taurus. Die Berechnungen wurden auf Basis des k-means Clusterings und $t_i = 300$ und $t_j = 1$ durchgeführt. Die Evaluation des Fehlers fand auf 20 zufälligen Testsimulationen statt.

Insgesamt kann also festgestellt werden, dass sowohl das initiale Clustering als auch die nichtlineare Dimensionsreduktion wichtige Schritte zum Auffinden der intrinsischen Dimension s der Daten sind. Außerdem können im entsprechenden niederdimensionalen Raum kleinere Rekonstruktionsfehler erzielt werden als mit der PCA. Somit kann der Fluch der Dimension umgangen werden. Die bessere Schätzung für s als es mit linearen Methoden möglich ist, ist ein wichtiger Schritt zum Identifizieren der Haupteffekte. Dies wird im folgenden Unterabschnitt thematisiert.

3.4.2 Abhängigkeit von den Input-Parametern

Um in der Praxis eine gewünschte Zielkonfiguration zu erreichen, ist es sehr wichtig zu identifizieren, wie spezifische Änderungen in den Eingangsparametern die Ausgangsgrößen beeinflussen. Die Berechnung solcher Abhängigkeiten ist eine der großen Herausforderungen und ein wichtiges Ziel der Dimensionsreduktion. Wir illustrieren im Folgenden anhand des Diffusion Maps Algorithmus, dass wir in der Lage sind, diese Abhängigkeiten zu identifizieren.

Chevrolet Truck Abbildung 15 zeigt die 2D-Einbettung der 126 Simulationsergebnisse des Frontalaufpralls. Die Analyse dieses Datensatzes ergab eine dominante Abhängigkeit des Crashverhaltens von der Blechdicke des Längsträgers 1 (siehe Abbildung 3 (b)). Den dargestellten Abbildungen ist diese Abhängigkeit direkt zu entnehmen. Jede Simulation wird hierbei durch einen Punkt dargestellt. Variationen der Blechdicke in



(a) Ohne Clustering (b) Cluster 3, spekt. Clust. (c) cluster 4, spekt. Clust. (d) Cluster 1, k-means (e) Cluster 8, k-means

Abbildung 15: 2D-Einbettung des Truck-Datensatzes basierend auf spektralem und k-means Clustering. Jeder Punkt entspricht einem Simulationslauf. Die Farbe kodiert die Blechdicke des ersten Längsträgers ($t_i = 7$, $t_j = 6$).

einem bestimmten Bereich veursachten bei diesem Datensatz sogar eine Bifurkation der Verformungen.

Ford Taurus Im Unterschied zum Truck existiert beim umfangreicheren Taurus-Datensatz keine auffällige Bifurkation, die durch die Blechdicken-Variation entsteht. Dennoch kann das Diffusion Maps Verfahren eine Tendenz im Crashverhalten des Modells aufzeigen (siehe Abbildung 16).

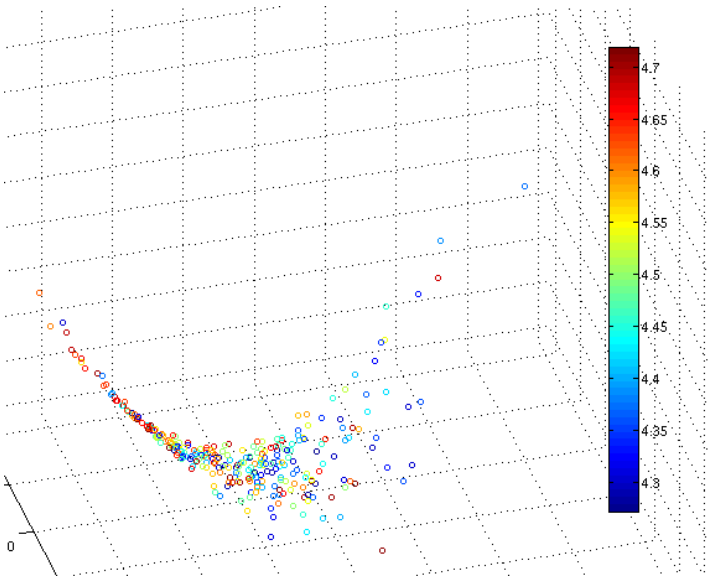


Abbildung 16: Einbettung der Verformungen in x-Richtung des Bauteils $T1$ (Stirnwand) mit Diffusion Maps. Die Farbkodierung bezieht sich auf die Blechdicke des Bauteils $T15$ (siehe Abbildung 5).

3.4.3 Multiskalen-Analyse von Simulationen

Im Teil-Projekt “Ingenieursmäßige Interpretation der Ergebnisse” wurde eine Verbesserung der Identifizierung der Eingangsparameter-Abhängigkeiten angestrebt. Ein Multiskalen-Ansatz wurde implementiert, der ein neues Abstandsmaß für den Vergleich

von Simulationen verwendet. Hierzu wird jede Simulation auf eine orthogonale Basis projiziert, welche geometriebasiert ist. Ein Beispiel hierfür sind die Eigenvektoren des Laplace-Beltrami Operators für die verformte Geometrie. Die ersten Komponenten stellen die gröberen Anteile der Verformungen, die weiteren Komponenten die Details dar. Abbildung 17 zeigt ein Beispiel, einer solchen Darstellung für die Stirnwand-Verformungen.

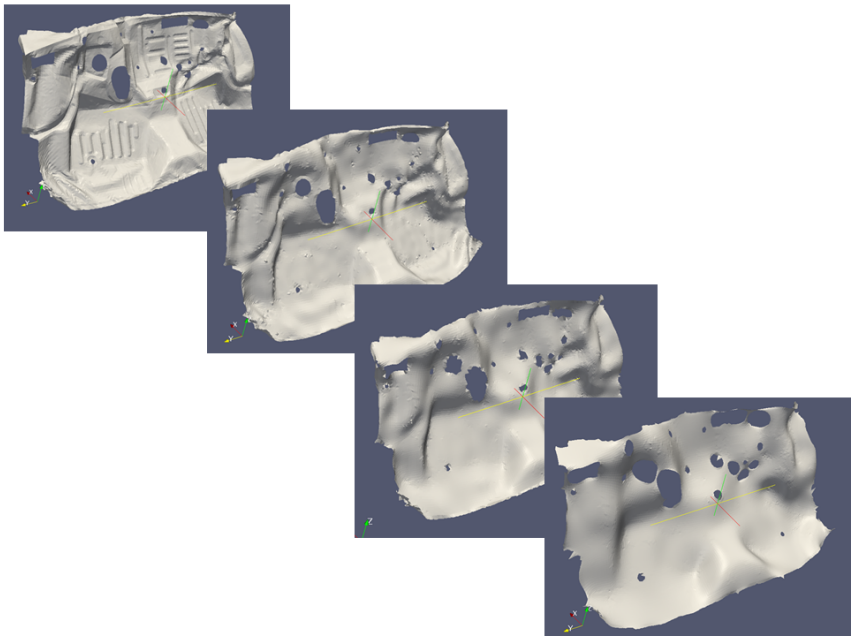


Abbildung 17: Darstellung der Verformungen auf unterschiedlichen Skalen.

Nach dieser Transformation wird ein Dimensionsreduktionverfahren (wie z.B. Diffusion Maps) verwendet (siehe Abbildung 18). Die 3D-Einbettung zeigt nun eine deutliche Verbesserung bei der Identifizierung der Eingangsparameter-Abhängigkeiten. Zu den Details des Verfahrens ist eine Veröffentlichung in Vorbereitung [ITG].
Die Simulationen können entsprechend der Lage der Punkte in der Einbettung angeordnet werden. In Abbildung 18 werden Simulationen, die den roten Punkten in der Einbettung zugeordnet sind, zusammengefasst und mit gleicher Farbe dargestellt (Grün). Gleichermaßen werden Simulationen, die zu den blauen Punkten in der Einbettung korrespondieren, mit einer zweiten Farbe dargestellt (Blau).
Abbildung 18 zeigt deutlich, dass die Anordnung bezüglich der Lage der Punkte in der Einbettung eine Trennung der Verformungen bezüglich der Blechdicke (Schichtung) ermöglicht.
Eine weitere Untersuchung wurde mit den Verformungen des Bauteils $T8$ durchgeführt. Auch für dieses Bauteil konnte eine Korrelation zur Blechdicke des Bauteils $T15$ (wie zuvor) identifiziert werden. Des Weiteren zeigte sich auch eine Korrelation zu den Blechdicken des Bauteils $T8$ selbst (siehe Abbildung 19 für die Korrelationen und 5 für die Lage der Bauteile).
Abbildung 20 zeigt die Verformungen des Bauteils, welche entsprechend der Informationen aus der Einbettung aus Abbildung 19 (b) angeordnet wurden. Auch hier zeigt

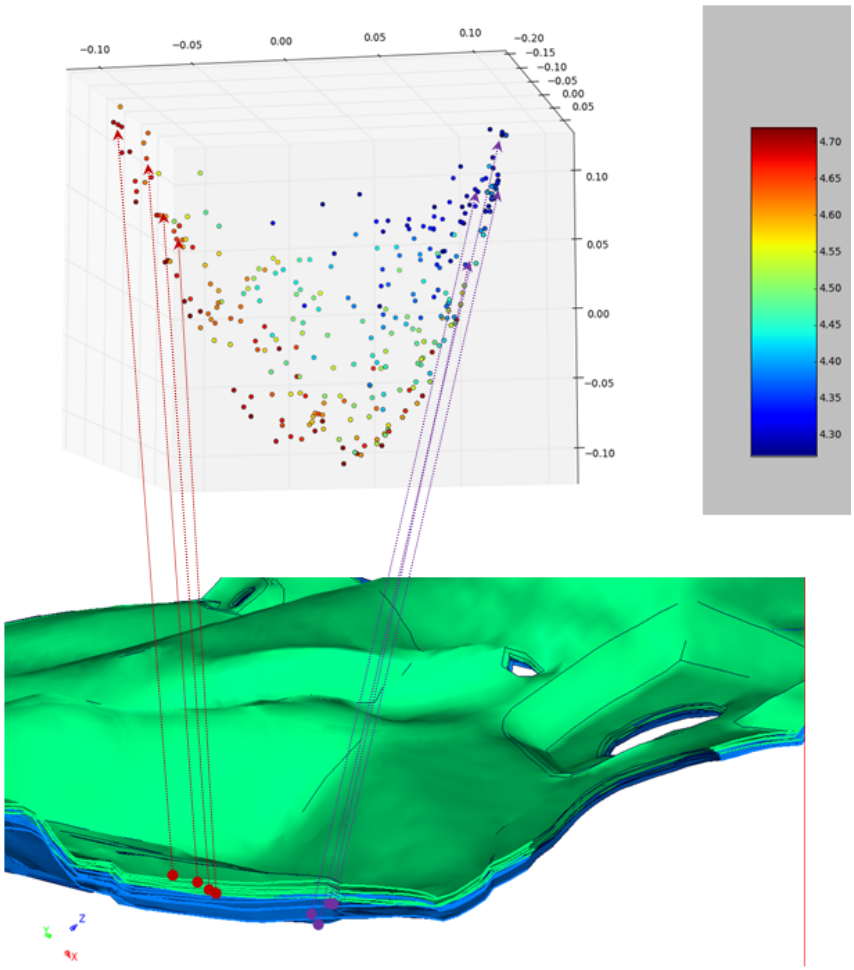
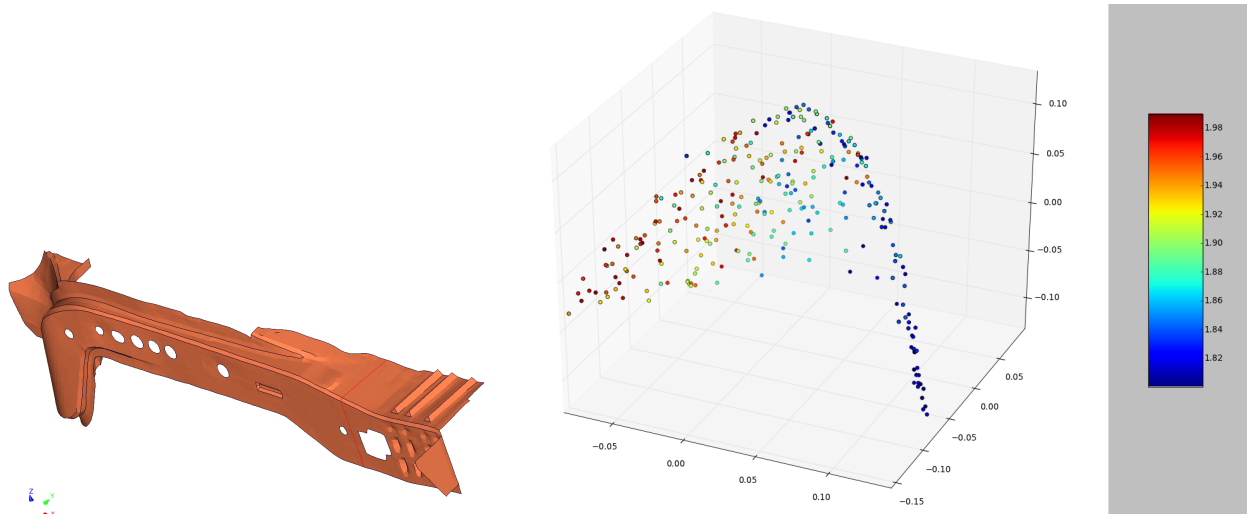


Abbildung 18: Analyse von Auswirkungen der Blechdicken-Variation an Bauteil $T15$ ($t_i = 120$, $t_j = 1$). Die 3D-Einbettung wurde mittels Diffusion Maps generiert, angewandt auf die verschiedenen Verformungen der Stirnwand ($T1$) in Aufprall-Richtung. Ein Punkt entspricht dabei einer bestimmten Verformung. Diese sind bezüglich der Lage der entsprechenden Punkte in der Einbettung angeordnet. Die Farbkodierung entspricht der Blechdicke des Bauteils $T15$ (siehe Abbildung 5 für die Lage der jeweiligen Bauteile).



(a) Bauteil T8 (b) 3D-Einbettung mit Korrelation

Abbildung 19: Analyse der Auswirkungen der Blechdicken-Variation an Bauteil $T8$ ($t_i = 120$, $t_j = 1$). Die 3D-Einbettung wurde mittels Diffusion Maps, angewandt auf die Verformungen des Bauteils $T8$ in Aufprall-Richtung, generiert. Die Farbkodierung entspricht der Blechdicke des Bauteils $T8$, siehe 5 für die Lage des Bauteils.

sich der Trennungseffekt bezüglich der Blechdicke (Schichtung).

3.4.4 Analyse von Output-Variablen mit Vine-Copula Modellen

Die Güte der Rekonstruktion kann nicht nur durch einen (absoluten) Rekonstruktionsfehler bestimmt werden, sondern auch durch dessen Einfluss auf die sogenannten Output-Variablen. Beispiele für solche Variablen sind der Head Injury Criterion (HIC) Index, die Stirnwandintrusion oder die Türspaltbreite. In unserer Analyse konzentrierten wir uns auf die Stirnwandintrusion. Ziel war es die Verteilung der Stirnwandintrusion mit Hilfe von Vine-Copulas jeweils auf den Originaldaten wie auch auf den rekonstruierten Daten zu modellieren. Dies sollte Aufschluss darüber geben, ob bestimmte, sicherheitsrelevante Outputgrößen korrekt rekonstruiert wurden.

Strinwandintrusionsdaten auf dem Ford Taurus Die Stirnwand trennt den Motorraum vom Passagierraum und folglich ist ihr Verhalten im Falle eines (Frontal-)Unfalls von besonderem Interesse für den Ingenieur, um die Sicherheit der Fahrgäste zu bewerten. Herr Taeschner (Volkswagen AG) nannte uns hierzu 23 markante Punkte auf der Stirnwand, die zur Bewertung herangezogen werden sollten (siehe Abbildung 21).
Die Stirnwandintrusion ist für jeden Zeitschritt $t_i = 1, \ldots, T$ als die Differenz zwischen der Verschiebung eines Knotens selbst und der Verschiebung eines festen Referenzknotens im Heck des Automobils definiert, jeweils bezüglich des Zeitschritts t_j, d.h.

$$ {}^r s_l^{t_i,t_j} = {}^r x_l^{t_i} - {}^r x_l^{t_j} - \left({}^r x_{fix}^{t_i} - {}^r x_{fix}^{t_j} \right) \quad l = 1, \ldots, 23. $$

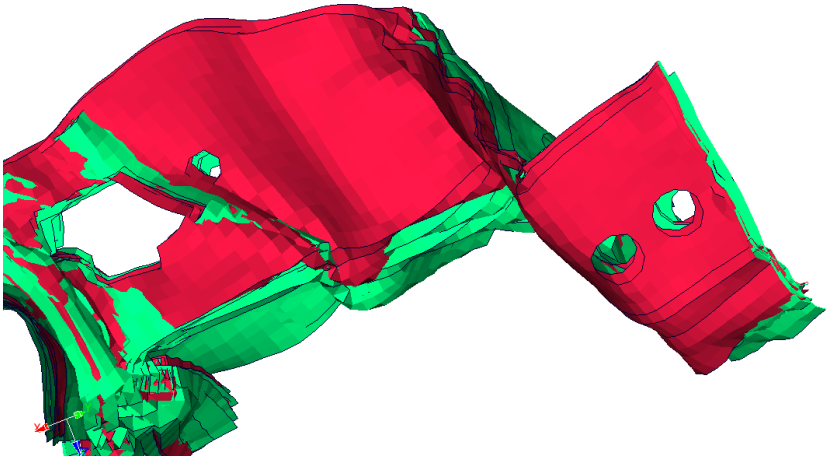


Abbildung 20: Sortierte Verformungen (bezüglich der Information der Einbettung aus Abbildung 19 (b) ($t_i = 120$, $t_j = 1$)

Der Referenzknoten wird hier verwendet, um Festkörperbewegungen des Autos bei der Analyse der Stirnwandintrusion nicht zu berücksichtigen. Analog definieren wir ${}^r\overline{s_l^{t_i,t_j}}$ für rekonstruierte Testdaten.
Speichern wir nun alle Euklidischen Normen dieser Verschiebungen in einer Matrix ab, so erhalten wir die finale Datenbasis zur Analyse der Stirnwand

$$S^{t_i,t_j} = \begin{pmatrix} \|{}^1s_1^{t_i,t_j}\|_2 & \|{}^2s_1^{t_i,t_j}\|_2 & \dots & \|{}^Rs_1^{t_i,t_j}\|_2 \\ \|{}^1s_2^{t_i,t_j}\|_2 & \|{}^2s_2^{t_i,t_j}\|_2 & \dots & \|{}^Rs_2^{t_i,t_j}\|_2 \\ \vdots & \vdots & & \vdots \\ \|{}^1s_M^{t_i,t_j}\|_2 & \|{}^2s_M^{t_i,t_j}\|_2 & \dots & \|{}^Rs_M^{t_i,t_j}\|_2 \end{pmatrix} \in \mathbb{R}^{23\times 289} \tag{5}$$

für $M = 23$ und $R = 289$. In unserem Fall berechneten wir die Stirnwandintrusion bezüglich $t_j = 1$. Voruntersuchungen erlaubten uns die zu untersuchende Zeitspanne auf $t_i = 150, \dots, 300$ einzuschränken, da erst diese Zeiträume das eigentliche Crashverhalten widerspiegelten.

Reguläre Vine Copula Modelle Betrachten wir die Zufallsvariablen $X_1, \dots, X_d$ mit den Randverteilungen $F_1(x_1), \dots, F_d(x_d)$, so kann die Bestimmung der gemeinsamen Verteilung (cdf) $F(x_1, \dots, x_d)$ von $X_1, \dots, X_d$ eine schwierige Aufgabe sein. Copulas erlauben uns, dieses Problem in zwei Teilprobleme zu zerlegen. Das Theorem von Sklar [14] trennt die Modellierung der Randverteilungen von der Modellierung der gemeinsamen Abhängigkeitsstruktur. Im zweidimensionalen Fall haben wir somit $F(x_1, x_2) = C(F_1(x_1), F_2(x_2))$, wobei $C : [0,1]^2 \to [0,1]$ eine Verteilungsfunktion mit gleichmäßigen Randverteilungen, die sogenannte Copula ist.
Im Zweidimensionalen sind Copulas sehr flexibel und könnnen unterschiedlichste Abhängigkeitsstrukturen und tail-Abhängigkeiten modellieren. Multivariate Copulas hingegen verlieren ihre Flexibilität in höheren Dimensionen, insbesondere die Fähigkeit

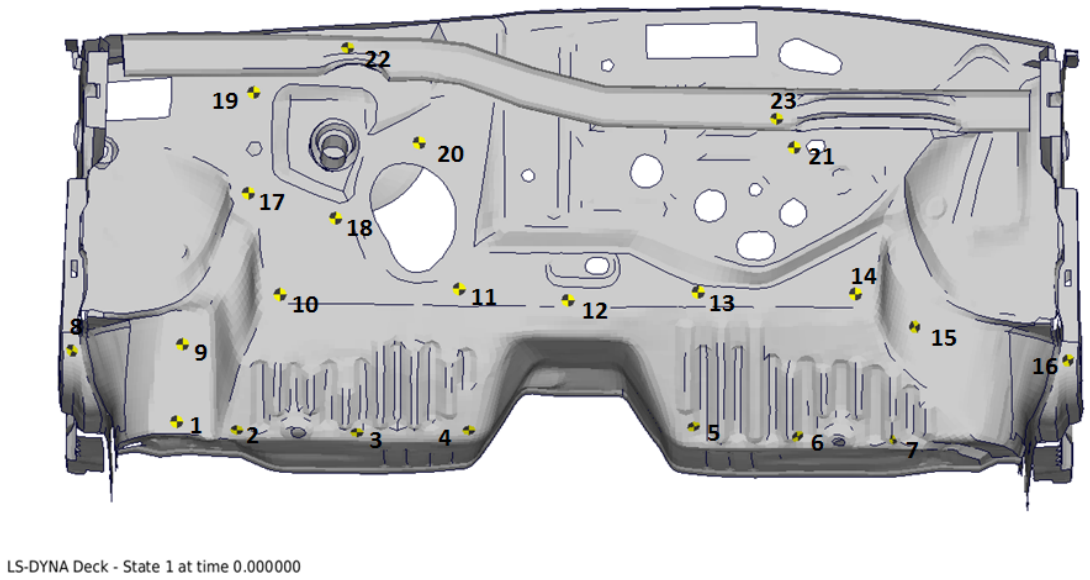


Abbildung 21: Stirnwand des Ford Taurus mit 23 markierten Intrusionspunkten

unterschiedliche tail-Abhängigkeiten darzustellen. Die multivariaten Vine Copula Modelle überwinden diese Probleme durch Zerlegung der multivariaten Dichte in bivariate (bedingte) Copulas und Randverteilungen. Dies wird Paar-Copula-Konstruktion genannt, wobei die graphischen Darstellungen dieser Zerlegungen Vines genannt werden. Durch die hohe Flexibilität der bivariaten Bausteine kann somit auch im hochdimensionalen Fall eine hohe Flexibilität und einfache Berechnung erreicht werden [Sch],[SS13b].

Ford Taurus Betrachtet man nun ähnlich wie in Abschnitt 3.4.1 den Fehler $|\|{}^r s_l^{t_i,t_j}\|_2 - \overline{\|{}^r s_l^{t_i,t_j}\|_2}|$ für jede einzelne der 20 Testsimulationen und jeden Zeitschritt t_i mit $i = 150, \ldots, 300$ und $l = 1, \ldots, 23$ separat, so kann man einzelne Simulationsläufe und Zeitschritte erkennen, in denen die Rekonstruktion suboptimal durchgefürt wurde (s. Abbildung 22). Hierbei ist $\overline{\|{}^r s_r^{t_i,t_j}\|_2}$ die mit PML oder PCA rekonstruierte Stirnwandintrusion.
Um die im Copulaansatz benötigten gleichverteilten und unabhängigen Beobachtungen zu erhalten, wurde der Trend mit Hilfe von B-splines entfernt. Eine anschliessende Bearbeitung der Residuen mit einem Zeitreihenmodell entfernt die zeitliche Abhängigkeit. Die nun unabhängigen Residuen lassen sich mit der probability integral Transformation auf gleichverteilte unabhängige Daten, sogenante Copula-Daten, umwandeln. Wir bezeichnen diese mit ${}^r\boldsymbol{u}_l, l = 1, \ldots, 23$ und $r \in I_{test}$.
Wir beginnen die Analyse der Abhängigkeitsstruktur der Stirnwandintrusion mit einfachen Vergleichen der empirischen Rank-Korrelation Kendall's τ. Hierzu berechnen wir ${}^r\tau$, das empirische Kendall's τ der Originalsimulationen, und $\overline{{}^r\tau}$, jenes der rekonstruierten Läufe, auf Basis der zeit-bereinigten, standartisierten und transformierten Stirnwandintrusionen ${}^r\boldsymbol{u}_l, l = 1, \ldots, 23$. Wie beim absoluten Fehler vergleichen wir wieder

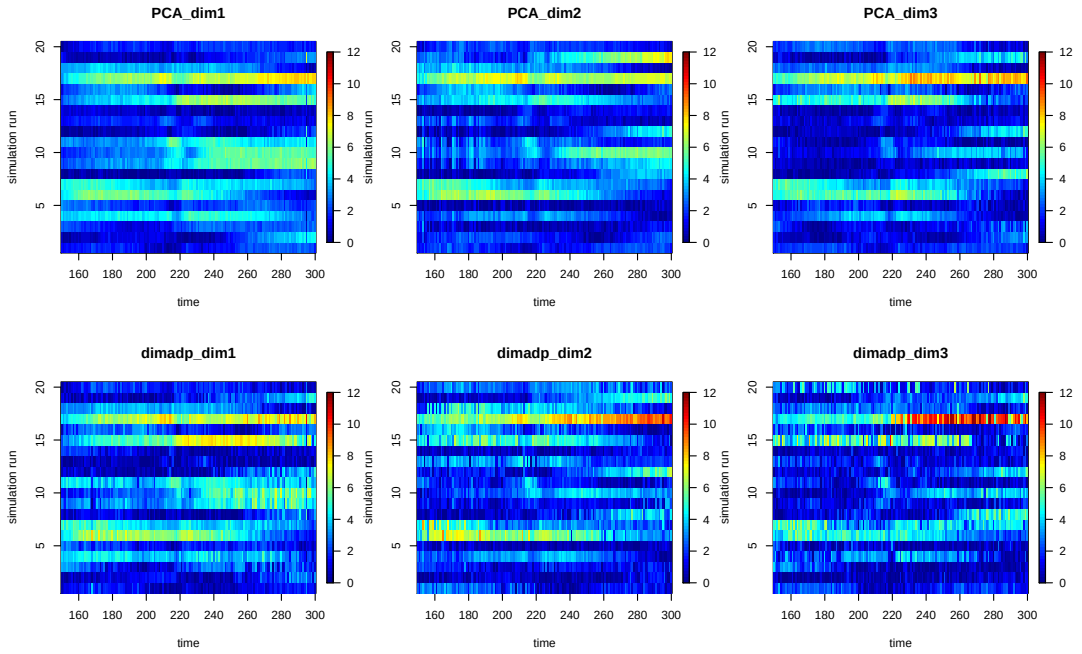


Abbildung 22: Absoluter Fehler der PCA und PML Rekonstruktion bezüglich dem Original für jeden Zeitschritt $t_j = 150, \ldots, 300$ und jede Simulation aus I_{test} für den ersten Stirnwandintrusionspunkt.

die Ergebnisse, d.h.

$$ {}^{r}\tau_{kl}^{diff} = |{}^{r}\tau_{kl} - \overline{{}^{r}\tau_{kl}}| \qquad k, l = 1, \ldots, 23. $$

In Abbildung 23 ist exemplarisch der absolute Fehler der PML Rekonstruktion für $s = 3$ Dimensionen dargestellt. In niedrigeren Einbettungsdimensionen vergrössern sich die Fehler leicht, da weniger Infomationen zur Rekonstruktion verwendet werden und demenstsprechend die Abweichungen von den ursprünglichen Simulationen zunehmen. Ähnliche, leicht schlechtere, Kendall's τ Fehler lassen sich auch für PCA Rekonstruktionen berechnen. Im Allgemeinen können wir erkennen, dass der absolute Fehler der Kendall's τ Werte für die meisten Simulationen nicht groß ist. Wie wir im Folgenden sehen werden, hat dieser allerdings bereits signifikante Auswirkungen für die unterschiedliche Vine-copula Modellierung.
Eine weit verbreitete Heurisik zum Auffinden der besten Vine-copula Struktur, d.h. der besten Zerlegung der d-dimensionalen Dichte in (bedingte) bivariate Copulas, ist der Algorithms von [4], welcher auf einem Maximum-Spanning-Tree Algorithmus basiert. Die Datenbasis bilden wiederum die gleichen 23 zeit-bereinigten Stirnwandintrusionspunkte der 151 betrachteten Zeitschritte. Die hieraus auf dem Original und den rekonstruierten Daten resultierenden ersten Bäume können exemplarisch für die Simulation 195 der Abbildung 24 entnommen werden. Im ersten Baum stimmen lediglich 8 Kanten, d.h. bivariate Copulas, überein. In anderen Simulationen und anderen intrinsischen Dimensionen sind es zum Teil noch weniger Übereinstimmungen. Hierbei gibt es auch keine signifikante Verbesserung von PCA zu PML.

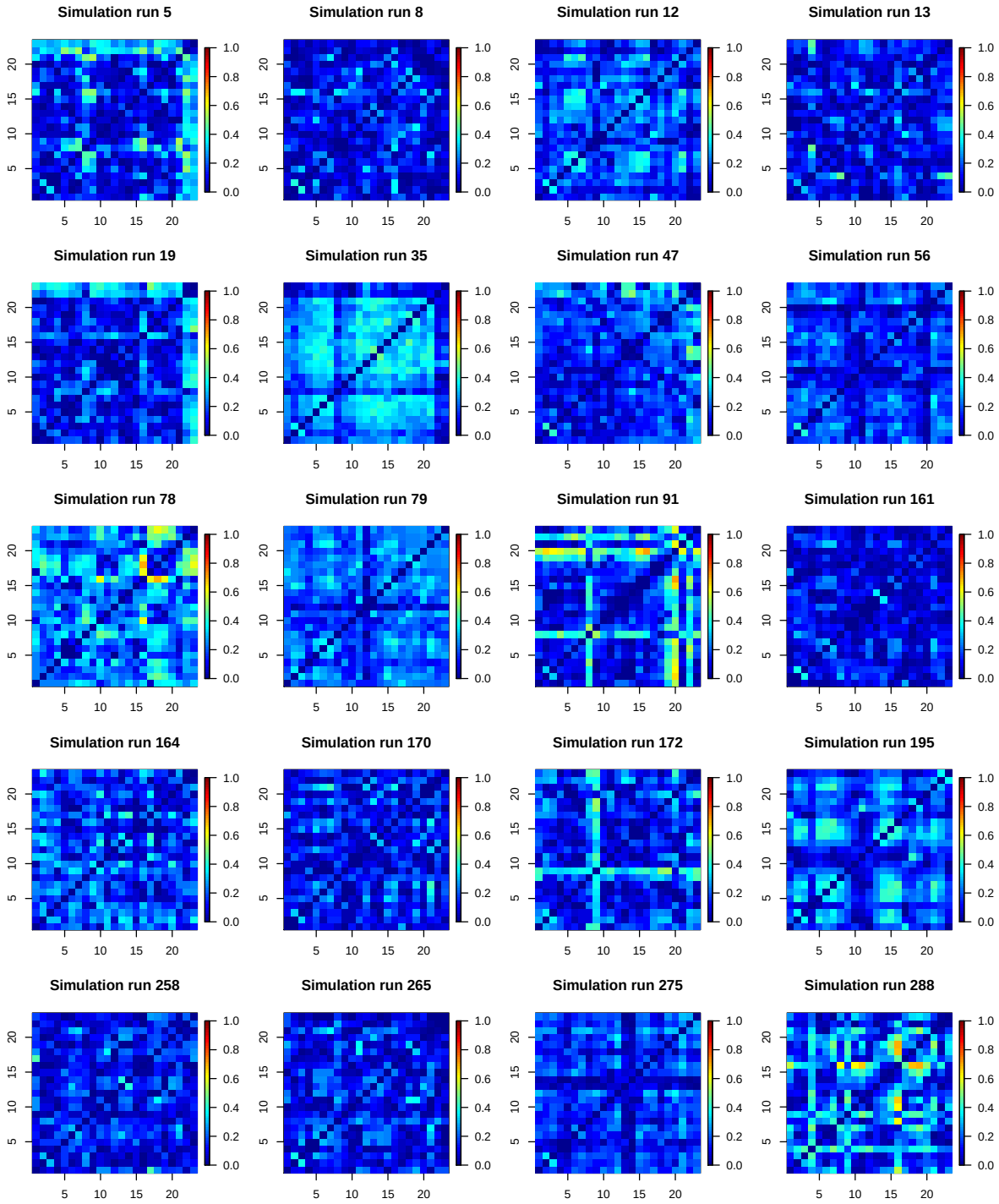


Abbildung 23: Absoluter Fehler der PML Rekonstruktion bezüglich dem Original auf den 23 Stirnwandintrusionspunkten bei Verwendung von Kendall's τ als Maß

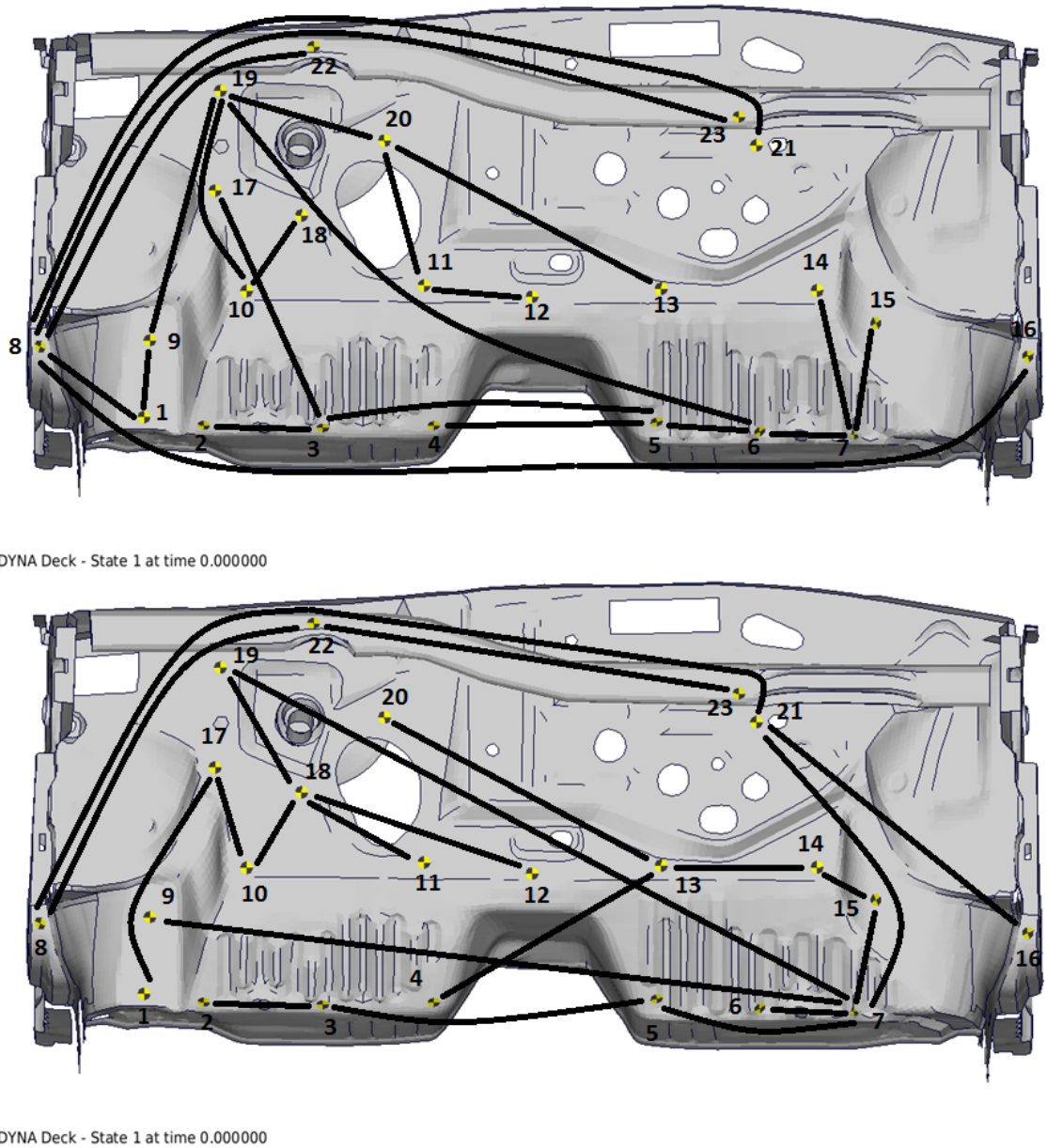


Abbildung 24: Erster Baum des Vine-copula Modells für den Simulationslauf 195. Oben: Original, unten: PML ($s = 3$)

Betrachtet man aber hingegen den Log-Likelihood oder Akaikes Informationskriterium (AIC) dieser Modelle und vergleicht sie mit den Ergebnissen, die man erhält, wenn man die fixe Struktur, die aus der Modellierung der Originaldaten resultiert, auf die rekonstruierten Daten anwendet, so kann man in vielen Fällen nur sehr geringe Unterschiede erkennen (siehe Tabelle 1). Anmerkung: Nur die Zerlegung der 23-dimensionalen Dichte wurde fix gewählt, nicht der komplette Vine, d.h. die Copulafamilien und deren Parameter der Paar-Copulas im Vine wurden frei selektiert und geschätzt. Die geringen Unterschiede im Log-likelihood und AIC sprechen hierbei für eine ähnliche Anpassungsgüte der Modelle. Ferner zeigen sie uns auf, dass die Vine-Copula Struktur, d.h. die Struktur der Bäume, in diesen Fällen wenig Einfluss auf die Anpassungsgüte der Modelle hat und somit auch auf die gemeinsame Verteilung der Stirnwandintrusionspunkte.
In einigen Simulationen stellen wir auch größere relative Abweichungen zwischen den Modellen fest. Dies spricht dafür, dass signifikante Merkmale dieser Simulationen wie beispielsweise Tail-Abhängigkeiten nicht korrekt rekonstruiert wurden und sich somit die Struktur und die gewählten Paar-Copulas in größerem Ausmaß geändert haben. Zu berücksichtigen ist hier allerdings auch die Modellunsicherheit in der Wahl der Vinecopulas. Die Selektierung der Vine-Copula Struktur sowie auch die Wahl der Paar-Copula Familien unterliegen statistischen Heuristiken, deren Optimalität nicht in allen Fällen gegeben ist.
Diese beiden Punkte sind unter Anderem bei der Beurteilung größerer relativer Differenzen der Log-Likelihood Werte oder AICs, wie z.B. in Smulation 288, zu berücksichtigen. Für diesen Simulationslauf zeigten sich bereits bei der Betrachtung der Kendall's τ Werte teils größere Abweichungen für die verschiedenen Rekonstruktionen (letzter Plot in Abbildung 23; insbesondere Stirnwandpunkt 16).

	Log-Likelihood						AIC					
		PCA			PML			PCA			PML	
Sim	$s=1$	$s=2$	$s=3$	$s=1$	$s=2$	$s=3$	$s=1$	$s=2$	$s=3$	$s=1$	$s=2$	$s=3$
5	2.47	1.70	6.72	0.60	2.28	1.65	-7.14	-5.45	-14.21	-1.94	-5.45	-5.40
8	2.87	1.93	0.81	0.14	3.27	2.58	-6.84	-5.19	-2.86	-1.75	-7.39	-7.22
12	5.38	4.33	2.83	7.34	6.74	1.79	-13.13	-10.52	-6.99	-16.40	-15.74	-5.52
13	4.05	-0.26	2.18	-0.11	0.42	0.59	-8.28	-1.36	-4.76	-1.50	-2.60	-3.00
19	3.54	4.59	1.96	0.43	3.34	0.96	-9.31	-11.11	-6.80	-3.93	-8.01	-3.60
35	2.25	2.11	2.15	2.01	1.65	2.64	-5.02	-4.80	-6.36	-4.95	-4.72	-6.93
47	4.47	1.53	5.80	1.84	5.77	6.95	-10.10	-5.27	-13.27	-5.61	-12.91	-15.41
56	4.40	-0.94	0.84	3.17	2.16	4.25	-8.94	0.75	-2.80	-6.13	-5.09	-9.40
78	2.03	3.39	-0.07	1.04	9.06	1.48	-5.09	-6.77	-2.24	-2.98	-17.35	-4.16
79	3.79	1.54	0.75	1.70	3.11	-0.22	-8.26	-5.07	-2.62	-5.31	-7.66	-1.18
91	2.41	5.27	1.40	2.97	2.30	3.64	-5.41	-12.81	-4.41	-8.89	-7.99	-9.58
161	3.26	1.67	1.58	1.86	2.65	3.92	-7.73	-4.73	-5.27	-5.33	-6.51	-9.11
164	2.79	1.75	3.62	3.37	1.39	1.90	-7.49	-4.80	-10.36	-8.14	-4.85	-6.84
170	3.64	3.63	1.50	2.97	2.19	3.38	-7.92	-9.01	-4.68	-7.64	-8.04	-8.56
172	2.99	2.59	2.13	5.28	1.19	3.65	-6.52	-6.68	-5.26	-12.09	-5.59	-10.54
195	1.91	3.14	4.10	2.12	2.72	1.47	-4.32	-6.45	-9.53	-4.56	-6.87	-4.43
258	6.83	7.07	4.44	4.62	3.05	2.90	-14.67	-16.10	-10.18	-10.90	-7.80	-8.36
265	3.85	7.60	3.84	6.75	4.16	1.28	-7.96	-15.62	-9.53	-14.82	-10.61	-5.11
275	3.14	0.01	0.01	-0.82	0.91	0.76	-7.61	-1.91	-1.69	-0.81	-2.03	-3.03
288	-2.02	0.13	3.10	1.80	8.62	7.15	2.74	-1.48	-6.05	-4.82	-19.42	-17.86

Tabelle 1: Relative Differenz der Log-Likelihood Werte und des Akaike Informationskriterium (AIC) der R-Vine Modelle basierend auf den mittels PCA und PML rekonstruierten Daten in Prozent ($\frac{ll-ll_{fix}}{ll}$). Wobei ll die log-likelihood bezeichnet, welche entstand, wenn die Struktur der R-Vine Modelle mit Hilfe des Algorithmus von [4] frei angepasst wurde und und ll_{fix} die log-likelihood für den Fall bezeichnet, in welchem die Struktur der R-vine Modelle der Originaldaten verwendet wurde, d.h. die Zerlegung der multidimensionalen Dichte wurde fixiert, aber die Copulafamilien und deren Parameter der beteiligten Paar-Copulas wurden frei selektiert und geschätzt.

3.5 Wichtigste Ergebnisse

Mit dem vorgestellten Workflow war es erstmals möglich industrierelevante Beispiele mit nichtlinearen Data Mining Methoden zu analysieren und untersuchen. Die hieraus resultierenden Vorteile, wie z.B. die korrekte Darstellung nichtlinearer Effekte auf niedrig-dimensionalen Strukturen, wurden anhand eines Chevrolet Truck und eines Ford Taurus Modells analysiert und illustriert. Dabei wurden sowohl quantitative Größen (Rekonstruktionsfehler) als auch qualitative Vorteile (Analyse anhand einer Visualisierung) berücksichtigt.

Dem Workflow folgend, seien hier nochmal die wichtigsten Ergebnisse im Hinblick auf die praktischen Aufgabenstellungen zusammengefasst:

- Ein nichtlineares Clustering basierend auf statistischen Methoden und Dünnen Gittern kann sowohl konvexe als auch nicht-konvexe Cluster erkennen. Dies führt zu einer genaueren Aufteilung des Modells. Aus dieser kann man nicht nur eine genauere Interpretation (qualitativ) ableiten, sondern es werden auch im anschließenden Dimensionsreduktionschritt kleinere Rekonstruktionsfehler (quantitativ) beobachtet.

- Mit den im Projekt entwickelten und anschließend auf die Crash-Modelle angewendeten nichtlinearen Dimensionsreduktionmethoden können nichtlineare Effekte mit weniger Dimensionen dargestellt werden, als dies mit herkömmlichen linearen Methoden bis jetzt möglich war. Je nach Analyseziel haben sich hierzu sowohl dünngitter-basierte (insbesondere zur Detektion der intrinsischen Dimension) als auch kern-basierte Methoden (insbesondere zur Visualisierung der niederdimensionalen Einbettung) als besonders geeignet herausgestellt.

- Des Weiteren konnte gezeigt werden, dass es mit nichtlinearen Verfahren (Diffusion Maps, Multiskalenansatz) möglich ist, den Einfluss der Eingangsparameter auf die Ausgangsdaten zu bestimmen und zu erklären. Dazu wurde unter anderem eine neue Multiskalen-Analyse entwickelt, welche für das Ford Taurus Modell eine deutlich bessere 3D-Einbettung erlaubt.

- In diesem Projekt wurden erstmals Vine-Copula Modelle zur Analyse der Output-Variablen verwendet. Copulas werden in anderen daten-getriebenen Anwendungen bereits seit Langem eingesetzt, eine Erweiterung zur Analyse von Simulationsdaten blieb jedoch bis jetzt aus. Die in diesem Projekt vorgenommene Analyse der Output Variablen der Crash-Modelle (z.B. Head Injury Criterion) führt zu dem Schluß, dass die Ergebnisse nach der nichtlinearen Dimensionsreduktion ein ähnliches Verhalten aufweisen, wie die hoch-dimensionalen Originaldaten. Dies bestätigt wieder die Vorteile der nichtlinearen Reduktion.

Insgesamt konnte klar gezeigt werden, dass für die Analyse von aktuellen, in der Industrie eingesetzten Crash-Modellen, herkömmliche lineare Verfahren nicht mehr ausreichen, sondern ein Workflow basierend auf nichtlinearen Methoden nötig ist.

3.6 Erfolgte und geplante Veröffentlichungen

Eine Übersicht über die im Projektumfeld erfolgten und geplanten Veröffentlichungen in Fachzeitschriften und -büchern sowie von Konferenzvorträgen ist am Ende des Berichts als Publikationsliste zu finden. Zusätzlich zu den Veröffentlichungen, die sich unmittelbar mit dem dargestellten Workflow zur Analyse von Simulationsdaten aus der Automobilindustrie beschäftigen und somit im Bericht bereits an entsprechender Stelle aufgeführt wurden, sind folgende in referierten Zeitschriften oder Proceedingsbänden erschienene (bzw. zu erscheinende) Publikationen zu erwähnen.

- In [PG12] wurde eine vereinheitlichte Beschreibung der Verfahren *Maximum Variance Unfolding* (MVU) [18, 19], *IsoMap* [15], und *Laplacian Eigenmaps* [1] entwickelt und zum ersten Mal eine Konvergenzaussage für MVU bewiesen.
- Die Anwendung des kernbasierten Verfahrens *Diffusion Maps* [3] auf Scharen von Crash-, NVH- und Umform-Simulationensdaten wurde in [IT13] ausführlich untersucht.
- Der Einsatz dünner Gitter zur Dimensionreduktion resultiert für Verfahren, die grundsätzlich kernbasiert arbeiten in einem zusätzlichen Diskretisierungsfehler. Dieser kann für spezielle Räume auch als eine Trunkation der Reihenentwicklung des Kerns angesehen werden [GRZ13].
- Eine Veröffentlichung zur Abschätzung des auftretenden Diskretisierungsfehler für Regression mit dünnen Gittern sowie für das konkrete Beispiel des Principal Manifold Learnings ist in Vorbereitung [BG].
- Die im Rahmen des Projekts durchgeführten Untersuchungen zur Einsetzbarkeit adaptiver dünner Gitter zur generativen Dimensionsreduktion resultierten in weiteren Veröffentlichungen zum eng verwandten Generative Topographic Mapping Verfahren [GH12],[GH13].
- In [BPPB12] wurden Dünngitter-Lernverfahren für große Datenmengen untersucht und zur Visualisierung von Simulationsdaten angewendet. Durch den gitterbasierten Ansatz konnten auch Real-World-Probleme aus dem Bauingenieurswesen untersucht werden. In [HPPS12] konnte eine bereits vorliegende Dünngitter-Methode zur Klassifizierung mit AdaBoost verwendet werden. Dazu wurden mehrere Dünngitter-Klassifikatoren in einer gewichteten Summe kombiniert. Da aktuelle High-Performance Systeme (heterogene CPU/GPU Systeme) besser mit vielen, regulären als mit wenigen aber unstrukturierten (adaptiven) Datenstrukturen umgehen können, konnten Klassifikationsprobleme deutlich schneller gelöst werden als mit den bisherigen Dünngitter-Klassifikationsmethoden.
- Neben dem im Bericht zitierten Paper [SS13b] und [Sch] sind noch die folgenden Arbeiten im Rahmen der Vine Copulas entstanden: [SS13a],[Sch13],[Sch13]. Sie alle beschäftigen sich mit der Unsicherheit in Vine Copula Modellen. Wärend [SS13b] und [SS13a] die Parameterunsicherheit behandelen, so wird die Modellunsicherheit mit Goodness-of-fit Tests in den Arbeiten [Sch13] und [Sch13]

betrachtet. Diese Themen und Manuskripte finden sich auch in der Dissertation von Herrn Schepsmeier wieder [Sch14].

- Im Rahmen der entstandenen Software `VineCopula` [SSB13] für Vine Copula Modelle ist noch die Vignette [BS13] zu erwähnen. Sie gibt eine Einführung in das Thema Vine Copulas und erläutert die Funktionalität des Paketes mit Beispielen.

4 Referenzen

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15:1373–1396, 2003.

[2] H.-J. Bungartz and M. Griebel. Sparse grids. Acta Numerica, 13:1–123, 2004.

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

[4] J. Dißmann, E. Brechmann, C. Czado, and D. Kurowicka. Selecting and estimating regular vine copulae and application to financial returns. Computational Statistics and Data Analysis, 59(1):52 – 69, 2013.

[5] C. Feuersänger. Sparse grid methods for higher dimensional approximation. PhD thesis, University of Bonn, 2010.

[6] C. Feuersänger and M. Griebel. Principal manifold learning by sparse grids. Computing, 85(4):267–299, 2009.

[7] J. Giesen and U. Wagner. Shape dimension and intrinsic metric from samples of manifolds with high co-dimension. Discrete and Computational Geometry, 32:245–267, 2004.

[8] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics. Springer, 2nd ed. 2009. corr. 3rd printing 5th printing. edition, Sept. 2009.

[9] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans. Pattern Analysis and Machine Intelligence, 24(7):881–892, 2002.

[10] J. Lee and M. Verleysen. Nonlinear dimensionality reduction. Springer, 2007.

[11] S. Lloyd. Least squares quantization in pcm. IEEE Trans. Information Theory, 28(2):129–137, 1982.

[12] K. V. Mardia, J. T. Kent, and J. M. Bibby. Multivariate Analysis. Academic Press, 1979.

[13] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. IEEE International Conference on Tools with Artificial Intelligence, pages 576–584, 2004.

[14] M. Sklar. Fonctions de répartition á n dimensions et leurs marges. Publ. Inst. Statist. Univ. Paris, 8:229–231, 1959.

[15] J. B. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 22 December 2000.

[16] C.-A. Thole, L. Nikitina, I. Nikitin, and T. Clees. Advanced mode analysis for crash simulation results. In Proceedings of the 9th LS-DYNA Forum, 2010.

[17] U. von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17:395–416, 2007.

[18] K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In Unfolding, Proceedings of the 21st National Conference on Artificial Intelligence. AAAI, 2006.

[19] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In ICML, 2004.

Eigene projektbezogene Publikationen

Artikel in referierten Fachzeitschriften und -büchern

[BG12] B. Bohn and M. Griebel. An adaptive sparse grid approach for time series predictions. In Sparse grids and applications, volume 88 of Lecture Notes in Computational Science and Engineering, pages 1–30. Springer, 2012.

[BGIT+13] B. Bohn, J Garcke, R. Iza Teran, A. Paprotny, B. Peherstorfer, U. Schepsmeier, and C.-A. Thole. Analysis of car crash simulation data with nonlinear machine learning methods. In Proceedings of the ICCS 2013, volume 18 of Procedia Computer Science, pages 621–630. Elsevier, 2013.

[BPPB12] Daniel Butnaru, Benjamin Peherstorfer, Dirk Pflüger, and Hans-Joachim Bungartz. Fast insight into high-dimensional parametrized simulation data. In 11th International Conference on Machine Learning and Applications (ICMLA), Boca Raton, Florida, December 2012. IEEE.

[BS13] Eike Christian Brechmann and Ulf Schepsmeier. Dependence modeling with C- and D-vine copulas: The R-package CDVine. Journal of Statistical Software, 52(3):1–27, 2013.

[Gar13] J. Garcke. Sparse grids in a nutshell. In J. Garcke and M. Griebel, editors, Sparse grids and applications, volume 88 of Lecture Notes in Computational Science and Engineering, pages 57–80. Springer, 2013.

[GH12] M. Griebel and A. Hullmann. A sparse grid based generative topographic mapping for the dimensionality reduction of high-dimensional data. In Proceedings of the 5th International Conference on High Performance Scientific Computing - Modeling, Simulation and Optimization of Complex Processes, 2012. submitted.

[GH13] M. Griebel and A. Hullmann. Dimensionality reduction of high-dimensional data with a non-linear principal component aligned generative topographic mapping. submitted. Also available as INS Preprint No. 1311, 2013.

[GRZ13] M. Griebel, C. Rieger, and B. Zwicknagl. Multiscale approximation and reproducing kernel hilbert space methods. submitted. Also available as INS Preprint No. 1312, 2013.

[HPPS12] Alexander Heinecke, Benjamin Peherstorfer, Dirk Pflüger, and Zhongwen Song. Sparse grid classifiers as base learners for adaboost. In 2012 International Conference on High Performance Computing and Simulation (HPCS),, pages 161–166, Madrid, July 2012. IEEE.

[IT13] R. Iza Teran. Enabling the analysis of finite element simulation-bundles. International Journal for Uncertainty Quantification, 2013. Accepted, doi:10.1615/Int.J.UncertaintyQuantification.2013005436.

[PAPB13] Benjamin Peherstorfer, Julius Adorf, Dirk Pflüger, and Hans-Joachim Bungartz. Image segmentation with adaptive sparse grids. In Proceedings of 26th Australasian Joint Conference on Artificial Intelligence. Springer, September 2013.

[PFPB13] Benjamin Peherstorfer, Fabian Franzelin, Dirk Pflüger, and Hans-Joachim Bungartz. Classification with probability density estimation on sparse grids. In Sparse Grids and Applications, 2013. accepted.

[PG12] A. Paprotny and J. Garcke. On a connection between maximum variance unfolding, shortest path problems and isomap. In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS 2012), pages 859–867, 2012.

[PPB11] Benjamin Peherstorfer, Dirk Pflüger, and Hans-Joachim Bungartz. A sparse-grid-based out-of-sample extension for dimensionality reduction and clustering with laplacian eigenmaps. In Dianhui Wang and Mark Reynolds, editors, AI 2011: Advances in Artificial Intelligence, volume 7106 of Lecture Notes in Computer Science, pages 112–121, Berlin Heidelberg, December 2011. Murdoch University, Western Australia, Springer.

[PPB12] Benjamin Peherstorfer, Dirk Pflüger, and Hans-Joachim Bungartz. Clustering based on density estimation with sparse grids. In KI 2012: Advances in Artificial Intelligence, volume 7526 of Lecture Notes in Computer Science. Springer, October 2012.

[Sch13] U. Schepsmeier. A goodness-of-fit test for regular vine copula models. under review, http://arxiv.org/abs/1306.0818, 2013.

[SS13a] U. Schepsmeier and J. Stöber. Derivatives and fisher information of bivariate copulas. Statistical Papers, online first: http://link.springer.com/article/10.1007/s00362-013-0498-x., 2013.

[SS13b] J. Stoeber and U. Schepsmeier. Estimating standard errors in regular vine copula models. Computational Statistics, online first: http://link.springer.com/article/10.1007/s00180-013-0423-8, 2013.

Vorträge bei wissenschaftlichen Konferenzen

[BG12a] B. Bohn and M. Griebel. Adaptive sparse grid discretizations for high-dimensional manifold learning tasks. In DWCAA 2012 - 3rd Dolomites Workshop on Constructive Approximation and Applications, Alba di Canazei, Italy, 2012.

[BG12b] B. Bohn and M. Griebel. Principal manifold learning with a dimension-adaptive sparse grid discretization. In SGA2012 - 2nd Workshop on Sparse Grids and Applications, Munich, Germany, 2012.

[BG13] B. Bohn and M. Griebel. On dimensionality reduction for high-dimensional machine learning with sparse grids. In HDA2013 - 5th Workshop on High-Dimensional Approximation, Canberra, Australia, 2013.

[Cza13a] C. Czado. Pair-copula constructions - even more flexible than copulas. In Workshop Non-Gaussian Multivariate Statistical Models and their Applications, Calgary, Canada, 2013. Banff International Research Station for Mathematical Innovation and Discovery.

[Cza13b] C. Czado. Pair copula copula construction for mixed outcomes with application to comorbidity among the elderly. In DAGStat 2013, Freiburg, Germany, 2013.

[Gar12] J. Garcke. Analysis of simulation data with nonlinear machine learning methods. In Mathematical Methods in Fluid Dynamics and Simulation of Giant Oil and Gas Reservoirs, Istanbul, Turkey, 2012.

[GP11] J. Garcke and A. Paprotny. An asymptotic convergence result for maximum variance unfolding based on an interpretation as a regularized shortest path problem. In HIM-Workshop on Manifold Learning, Bonn, Germany, 2011.

[GP12] J. Garcke and A. Paprotny. Graph-based spectral approaches to manifold learning - a unifying theoretical analysis based on distance matrices and graph approximations to the geodesic metric. In DMV Jahrestagung, Saarbrücken, Germany, 2012.

[IT11a] R. Iza-Teran. Diffusion maps for finite-element simulation data. In HIM - Workshop on Manifold Learning, Bonn, Germany, 2011.

[IT11b] R. Iza-Teran. Spectral kernel methods for simulation data and the eigenvector switching problem. In HDA2011 - 4th Workshop on High-Dimensional Approximation, Bonn, Germany, 2011.

[Pap11] A. Paprotny. Algebraic multigrid methods for discrete stochastic optimal control. In ILAS, Braunschweig, Germany, 2011.

[Peh11a] Benjamin Peherstorfer. A multigrid method for PDEs on spatially adaptive sparse grids. In HDA2011 - 4th Workshop on High-Dimensional Approximation, Bonn, Germany, 2011.

[Peh11b] Benjamin Peherstorfer. Reduced basis methods and sparse grids. In HIM - Workshop on Sparse Grids and Applications, Bonn, Germany, 2011.

[Peh12a] Benjamin Peherstorfer. Clustering based on density estimation with sparse grids. In KI 2012: Advances in Artificial Intelligence, Saarbrücken, Germany, 2012.

[Peh12b] Benjamin Peherstorfer. A multigrid method for PDEs on spatially adaptive sparse grids. In 28th GAMM-Seminar on Analysis and Numerical Methods in Higher Dimensions, Leipzig, Germany, 2012.

[Peh12c] Benjamin Peherstorfer. A sparse-grid-based out-of-sample extension for dimensionality reduction and clustering with Laplacian eigenmaps. In SGA 2012, Munich, Germany, 2012.

[Sch12] U. Schepsmeier. Efficient inference and fisher information for regular vine copula distributions. In 8th World Congress of Probability and Statistics, Istanbul, Turkey, 2012.

[Sch13] U. Schepsmeier. A goodness-of-fit test for regular vine copula models. In DAGStat 2013, Freiburg, Germany, 2013.

Abschlussarbeiten

[Bet11] Sebastian Bethke. Zwei Methoden zur gitterunabhängigen Klassifikation von Bauteilen unter Berücksichtigung geometrischer Eigenschaften. Diplomarbeit, Universität zu Köln, 2011.

[Fra11] Fabian Franzelin. Classification with estimated densities on sparse grids. Master's thesis, Institut für Informatik, Technische Universität München, November 2011.

[Hör13] Thomas Hörmann. Parallel algorithms for sparse grids in X10. Bachelor's thesis, Institut für Informatik, Technische Universität München, June 2013.

[Kol13] Torsten Koller. Employing policy search techniques for learning throttle valve control. Bachelor's thesis, Institut für Mathematik, Technische Universität München, June 2013.

[MH13] Ao Mo-Hellenbrand. Sparse grid interpolants as surrogate models in statistical inverse problems. Master's thesis, Institut für Informatik, Technische Universität München, September 2013.

[Oet11] Jens Oettershagen. Reduktion der effektiven Dimension und ihre Anwendung auf hochdimensionale Probleme. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2011.

[Peh13] Benjamin Peherstorfer. Model Order Reduction of Parametrized Systems with Sparse Grid Learning Techniques. Dissertation, Department of Informatics, Technische Universität München, October 2013.

[Röh11] Kilian Röhner. Polynomial basis functions on adaptive sparse grids. Bachelor's thesis, Institut für Informatik, Technische Universität München, August 2011.

[Sch13] Anna-Luisa Schwartz. Diffusion Maps und ihre Anwendung bei der Analyse von Automobildaten. Bachelorarbeit, Universität Bonn, 2013. Ada-Lovelace Preis 2013.

[Sch14] U. Schepsmeier. Estimating standard errors and efficient goodness-of-fit tests for regular vine copula models. Dissertation, Technische Universität München, 2014.

[Vor12] Thilo Vorderbrück. Diffusionsbasierte Clustering-Verfahren. Bachelorarbeit, Universität zu Köln, 2012.

[Zac13] Florian Zacherl. New loss functions for sparse grid classifiers. Master's thesis, Institut für Informatik, Technische Universität München, May 2013.

Interne Berichte und unveröffentlichte Arbeiten

[BG] B. Bohn and M. Griebel. Error estimates for multivariate regression and principal manifold learning algorithms on discretized function spaces. In preparation.

[ITG] R. Iza-Teran and J. Garcke. Operator based multiscale analysis of engineering data. In preparation.

[Sch] U. Schepsmeier. Dependence modeling with r-vine copula model: A case study for car crash simulation data. In preparation.

[Sch13] U. Schepsmeier. Efficient goodness-of-fit tests in multi-dimensional vine copula models. preprint, available at http://arxiv.org/abs/1309.5808, 2013.

[SSB13] Ulf Schepsmeier, Jakob Stoeber, and Eike Christian Brechmann. VineCopula: Statistical inference of vine copulas, 2013. R package version 1.2.