

ALBERT-LUDWIGS-UNIVERSITÄT  
FREIBURG  
INSTITUT FÜR INFORMATIK

Lehrstuhl für Mustererkennung und Bildverarbeitung  
Prof. Dr. Hans Burkhardt



Entwicklung von Algorithmen zur  
lageinvarianten Merkmalsgewinnung  
für Drahtgittermodelle

Diplomarbeit

Marco Reisert

1. Juli 2003 – 1. Januar 2004



# Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit von mir selbständig und nur unter Verwendung der aufgeführten Hilfsmittel erstellt wurde.

Freiburg, den 28. Dezember 2003

# Danksagung

Zuerst möchte ich Bernard Haasdonk, meinem Betreuer, danken, der mir immer mit Rat und Tat zur Seite stand. Besonderer Dank gebührt Prof. Dr. Hans Burkhardt, der mir die hier vorliegende Arbeit erst in dieser Form ermöglicht hat und wertvolle Hinweise zur 3Distanzkodierung in Kapitel 4 gegeben hat.

Natürlich möchte ich noch meinem Schnuck danken, der mich immer unterstützt und aufgebaut hat und oft viel Geduld mit mir bewiesen hat, und meinen lieben Eltern, die ja erst dafür verantwortlich sind, dass ich da bin.

Freiburg, Baden-Württemberg  
28. Dezember 2003

Marco Reisert

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Zielsetzung . . . . .	7
1.2	Aufbau dieser Arbeit . . . . .	8
<b>2</b>	<b>Übersicht über bestehende Ansätze</b>	<b>9</b>
2.1	ShapeDistribution . . . . .	10
2.2	Fourier Deskriptoren und Kugelfunktionen . . . . .	11
2.3	Gruppenmittelung . . . . .	13
<b>3</b>	<b>Vorstellung des Rahmenwerks</b>	<b>15</b>
3.1	Grundlagen . . . . .	15
3.2	Motivation . . . . .	17
3.2.1	Physikalische Sicht . . . . .	17
3.3	Kodierungen . . . . .	18
3.4	Merkmalsgewinnung . . . . .	20
3.4.1	Merkmalsgewinnung durch Mittelung . . . . .	20
3.4.2	Merkmalsgewinnung durch Histogramme . . . . .	21
3.4.3	Die Exponential-Abbildung . . . . .	22
3.4.4	Numerische Berechnung und Komplexität . . . . .	23
3.5	Vollständigkeitsbetrachtungen . . . . .	25
3.5.1	Die Zurückgewinnung der Vektoren . . . . .	26
3.5.2	Von den Vektoren zur Kodierung . . . . .	28
3.5.3	Von der Kodierung zum Gitter . . . . .	29
3.6	Ein einfaches Beispiel . . . . .	30
<b>4</b>	<b>Stückweise lineare Polygonzüge in der Ebene</b>	<b>31</b>
4.1	Allgemeine Betrachtungen . . . . .	31
4.1.1	CT-Transformation . . . . .	33
4.1.2	Diskrete Fourierdeskriptoren . . . . .	35
4.2	Kodierungen . . . . .	35

4.2.1	Distanzkodierungen . . . . .	36
4.3	Experimente . . . . .	39
4.3.1	Störungsuntersuchungen . . . . .	42
4.4	Vollständigkeitsbetrachtungen . . . . .	47
4.5	Kommentare . . . . .	49
<b>5</b>	<b>Allgemeine 3D-Drahtgittermodelle</b>	<b>51</b>
5.1	Kodierung . . . . .	51
5.2	Implementation und Schnittstellen . . . . .	55
5.3	Experimente . . . . .	56
5.3.1	Moleküle . . . . .	57
5.3.2	Fische . . . . .	64
5.4	Kommentare . . . . .	65
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>68</b>
<b>A</b>	<b>Symbole</b>	<b>72</b>
<b>B</b>	<b>Beschreibung der Implementation</b>	<b>75</b>
B.1	Benutzte Pakete und Umgebungen . . . . .	75
<b>C</b>	<b>Moleküle</b>	<b>77</b>

# Kapitel 1

## Einleitung

Das jedermann weltweit zugängliche und stetig wachsende Angebot von multimedialen Daten erfordert sinnvolle Methoden zur effektiven Suche in multimedialen Datenbanken. Man denke hier an die Suche nach Bildern, Fotos, 3D-Modellen oder Audiodaten im Internet. Eine solche Suchanfrage könnte zum Beispiel lauten: „Suche alle Bilder, die einen Baum enthalten!“ oder „Gib mir alle 3D-Modelle, die das Raumschiff *Enterprise* als Motiv enthalten!“. Die einfachste Methode ist sicher die Suche nach den Dateinamen solcher Objekte, da diese häufig in semantischem Zusammenhang mit dem Inhalt der Datei stehen. Jedoch reichen dem Anwender oft diese Informationen nicht aus oder sind sogar falsch. Wenn jedoch der Benutzer bereits eine konkretere Vorstellung seines Objekts, zum Beispiel in Form einer Bilddatei, die einen Baum enthält, besitzt, kann anhand dieses Objekts in einer Datenbank nach ähnlichen Objekten gesucht werden. Solche Suchanfragen setzen aber voraus, dass die Suchmaschine auf sinnvolle Weise Vergleiche zwischen den entsprechenden Bildern durchführen kann. Das heißt, dass der Vergleich der Bilder unabhängig von der Lage des Objektes erfolgen muss, denn der Baum muss sich nicht an einer bestimmten Stelle des Bildes befinden, sondern könnte sich irgendwo im Bild befinden oder in leicht gedrehter Version vorliegen.

Die naheliegendste Methode ist ein direkter Vergleich der Objekte in allen möglichen Lagen, jedoch ist dies sehr zeitaufwendig und rechenintensiv und deshalb praktisch kaum anwendbar. Um den Vergleich effizienter auszuführen, bietet es sich an, die Objekte vor dem Vergleich zu normalisieren, also in eine Standardlage zu bringen, um dann einen direkten Vergleich zu ermöglichen. Jedoch ist es schwierig, Algorithmen zu finden, die solche Normalisierungen auf eine kanonische Weise durchführen.

Eine weit verbreitete und sehr allgemeine Methode ist die Extraktion eines Merkmalsvektors, der eine möglichst vollständige, kompakte und natürlich lageunabhängige Beschreibung des vorliegenden Objektes liefert. Der Vergleich reduziert sich anschließend auf den einfachen Vergleich der korrespondierenden Merkmalsvektoren. Die Art und Weise der Gewinnung der Merkmalsvektoren zusammen mit einem Distanzmaß

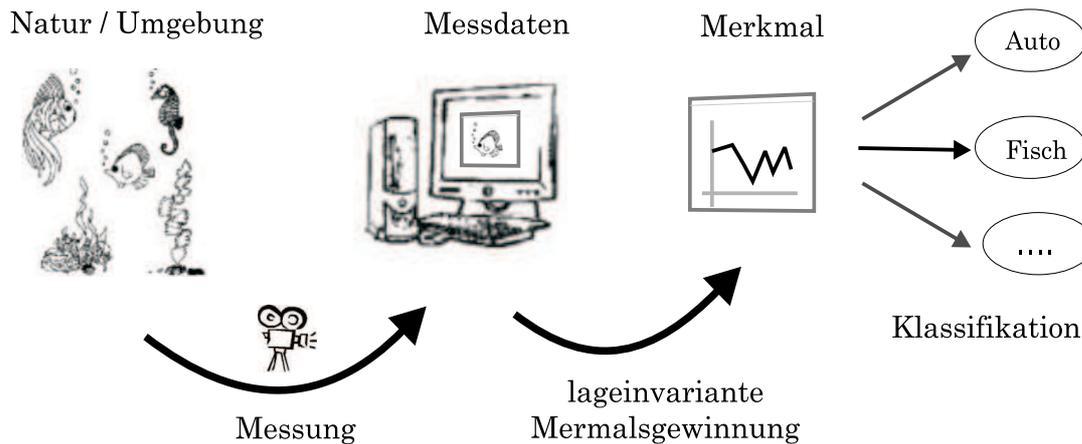


Abbildung 1.1: **Der Erkennungsprozess.**

auf diesem Vektorraum induziert dann ein Ähnlichkeitsmaß auf dem Musterraum. Diese Methode ist natürlich nicht nur auf Suchanfragen in multimedialen Datenbanken beschränkt, sondern findet auch in anderen Bereichen reichlich Anwendung. Zum Beispiel in der automatischen Defekterkennung industrieller Bauteile oder als Vorverarbeitungsschritt in den verschiedensten Mustererkennungsproblemen. In Abbildung 1.1 sind die Schritte des allgemeinen Mustererkennungsprozesses anschaulich dargestellt. Zu Beginn werden die zu erkennenden Objekten aufgenommen, die dann in digitaler Form auf dem Rechner landen. Dort werden dann die gemessenen Daten, nach eventueller Nachbearbeitung und Segmentierung, auf Merkmale abgebildet, die möglichst charakteristisch für das Objekt sein sollen. Diese werden schließlich einem Klassifikationsprozess unterworfen, der das Objekt seiner Bedeutungsklasse zuordnet.

## 1.1 Zielsetzung

Im Mittelpunkt dieser Arbeit steht die Entwicklung eines neuen Rahmenwerks zur Gewinnung lageunabhängiger Merkmale von Drahtgittermodellen. Unter einem Drahtgittermodell verstehe man einen in den Raum eingebetteten Graphen. Dieses Rahmenwerk soll sowohl aus theoretischer als auch aus praktischer Sicht betrachtet werden. Die theoretischen Überlegungen sollen untersuchen, inwieweit der Merkmalsbildungsprozess eindeutig umkehrbar ist, das heißt, ob die Objekte vollständig aus ihren Merkmalen zurückgewinnbar sind. Aus praktischer Sicht stehen Dinge wie Berechenbarkeit und Störsensibilität im Vordergrund. Schließlich sollen die neu vorgestellten Merkmale im Vergleich zur bereits existierenden Methode, der *ShapeDistribution*, betrachtet werden.

## 1.2 Aufbau dieser Arbeit

Die vorliegende Schrift ist in vier Teile gegliedert:

- Eine Übersicht über bereits bestehende und etablierte Ansätze der invarianten Merkmalsgewinnung wird in Kapitel 2 gegeben.
- In Kapitel 3 wird das neue Rahmenwerk zur Gewinnung lageunabhängiger Merkmale eingeführt, wobei zuerst die mathematischen Grundlagen erarbeitet werden und dann die allgemeine Herangehensweise erklärt wird. Soweit wie möglich werden dort bereits auch die Voraussetzungen für Vollständigkeit formuliert und diskutiert.
- In den Kapiteln 4 und 5 werden dann die vorgestellten Methoden auf Polygonzüge in der Ebene und allgemeine Drahtgittermodelle im dreidimensionalen Raum angewandt, wobei bei ersteren die in Kapitel 3 gemachten Vollständigkeitsbetrachtungen genauer ausgeführt werden sollen.
- Kapitel 6 liefert eine Zusammenfassung der Ergebnisse und einen Ausblick auf weitere Fragestellungen.

# Kapitel 2

## Übersicht über bestehende Ansätze

In der Praxis treten komplexe 3-dimensionale Objekte und Strukturen in den verschiedensten Repräsentationen auf. Grundlegend kann man zwischen Volumendaten<sup>1</sup> (oder auch Voxelmodelle) und Drahtgittermodellen unterscheiden. Volumendaten tauchen vor allem in medizinischen und biologischen Datensätzen auf. In der Mustererkennung bestehen bereits eine Vielzahl von Ansätzen zur Gewinnung lageinvarianter Merkmale aus Volumendaten [2, 5, 15, 22, 26], die auch bereits erfolgreich Anwendung gefunden haben. Typische Datensätze stammen von Kernspintomographen oder werden durch Mikroskopie gewonnen. Bei Volumendatensätzen handelt es sich häufig um direkten oder leicht aufbereiteten sensorischen Input, bei Drahtgittermodellen hingegen geht es meist um schon interpretierte Sensordaten oder Daten, die auf einem Computer erstellt worden sind. Als Beispiel sind hier CAD-Modelle, Oberflächenmodelle in der Computergrafik oder molekulare Daten aus der Chemie zu nennen. Die CAD-Modelle weisen die komplexeste Struktur auf. Ein CAD-Modell besteht in der Regel aus einer Menge primitiver Grundbausteine, wie Kreis- und Kugelsegmente, Tori, Polyedern und stetige Oberflächenparametrisierungen z.B. mit Hilfe von Splines oder Bezier-Kurven.

Oberflächenmodelle werden in der Computergrafik sehr einheitlich mittels triangulierten Oberflächen beschrieben, was die Darstellung auf dem Computer vereinfacht und standardisiert. Ein einzelnes Modell besteht in der Regel aus unzusammenhängenden Einzelbausteinen, wobei jeder Baustein aus einer triangulierten Oberfläche besteht. Diese Oberflächen müssen weder geschlossen noch auf irgendeine Weise orientiert sein. Die Mustererkennungsgemeinde hat auch hier schon eine Vielzahl von Ansätzen hervorgebracht [1, 3, 4, 28], auf welche wir später noch genauer eingehen werden.

Molekulare Daten spielen vor allem in der Biochemie eine wichtige Rolle. In der Biotechnologie und Pharmazie wird vermehrt computergestützt nach biologisch aktiven Wirkstoffen gesucht. Es wird in großen Datenbanken anhand bereits bekannter,

---

<sup>1</sup>Unter Volumendaten sind dreidimensionale diskrete Felder zu verstehen. Der Raum wird in Quader (Voxel) unterteilt und jedem dieser Quader ein Wert zugeordnet, zum Beispiel Farbe oder Helligkeit.

medizinisch wirksamer Regionen nach ähnlichen Bereichen gesucht, die sich dann hoffentlich als biologisch aktiv herausstellen werden [10, 21, 27]. Die Repräsentation eines Moleküls besteht im allgemeinen aus den Positionen seiner Atome, den Namen dieser Atome und gegebenenfalls den Bindungsinformationen zwischen den Atomen; diese sind jedoch meist auch aus den Positionen der Atome zu gewinnen.

In den folgenden Abschnitten sollen nun verschiedene Methoden der lageinvarianten Merkmalsgewinnung beschrieben werden. Insbesondere werden wir die Merkmalsgewinnung durch die *ShapeDistribution* [4] als Referenzverfahren genauer betrachten, die auch später in den Experimenten Anwendung finden wird. Weiter werden noch die Methoden der Gruppenmittelung, der Fourier Deskriptoren und die Anwendung von Kugelfunktionen betrachtet.

## 2.1 ShapeDistribution

Bei der Methode der *ShapeDistribution* wird versucht, mittels einer Art von Histogramm die geometrische Form eines Objekts darzustellen. Anschaulich stelle man sich vor, man messe mehrmals den euklidischen Abstand zwischen jeweils zwei zufällig gewählten Punkten auf der Oberfläche des geometrischen Objekts und zähle, wie häufig die verschiedenen Abstände auftreten. Die Messung sämtlicher Häufigkeiten ergibt die Verteilungsfunktion des euklidischen Abstandes, die *ShapeDistribution* genannt wird. Statt des euklidischen Abstandes können auch andere Eigenschaften gemessen werden. Diese Eigenschaften bzw. Funktionen werden von den Autoren *ShapeFunction* genannt. In [4] wurde eine Reihe solcher *ShapeFunction* vorgestellt und untersucht, z.B.

- **D1** misst den Abstand von einem festen Punkt  $\vec{p}$  und einem zufällig gewählten Punkt auf der Oberfläche des Objekts. Als Punkt  $\vec{p}$  wählt man den Schwerpunkt des Objekts.
- **D2** misst den Abstand zwischen zwei zufällig gewählten Punkten.
- **D3** misst die Fläche eines Dreiecks, das durch drei zufällig gewählte Punkte auf der Oberfläche des Objekts erzeugt wird.
- **D4** misst das Volumen eines Tetraeders, das durch vier zufällig gewählte Punkte auf der Oberfläche des Objekts erzeugt wird.

Für die Implementation wird von einer triangulierten Oberfläche ausgegangen. Um gleichverteilt einen zufälligen Punkt auf der Oberfläche des Objekts wählen zu können, wird zunächst ein zufälliges Dreieck ausgewählt, wobei die Wahrscheinlichkeit, dass dieses Dreieck gewählt wird, proportional zu dessen Fläche ist. Dann wird auf der

Oberfläche des Dreiecks ein Punkt zufällig ausgewählt. Die Auswahl des Dreiecks ist der rechenintensive Teil. Zu Beginn werden die Flächen  $a_i$  der Dreiecke berechnet und daraus die Verteilungsfunktion  $f_i = \sum_{j=1}^i a_j$ . Man wähle nun gleichverteilt zufällig eine Zahl  $r \in (0, f_n)$  und suche das Dreieck  $j$ , so dass  $f_j < r < f_{j+1}$ . Die Suche nach dem Dreieck kann schnell mit einer binären Suche durchgeführt werden. Ein zufälliger Punkt  $\vec{p}$  auf dem Dreieck  $(\vec{a}, \vec{b}, \vec{c})$  kann durch folgende Formel gewonnen werden:

$$\vec{p} = (1 - \sqrt{r_1})\vec{a} + \sqrt{r_1}(1 - r_2)\vec{b} + \sqrt{r_1}r_2\vec{c},$$

wobei  $r_1$  und  $r_2$  zwei uniform verteilte zufällige Zahlen aus dem Intervall  $(0, 1)$  sind. Zur Herleitung der Formel siehe man [4]. Bei den Versuchen in [4] werden 1024 Bins bzw. Urnen zur Gewinnung einer *ShapeDistribution* benutzt und es werden  $S = 1024^2$  Messungen unternommen. Die Parameter wurden empirisch gewonnen, mit Hinsicht auf ein vernünftiges Verhältnis zwischen Berechnungszeit und Genauigkeit. Der Aufwand der binären Suche ist logarithmisch in der Größe  $N$  der zu untersuchenden Menge, in diesem Fall also logarithmisch in der Anzahl der Dreiecke. Es ergibt sich für den Gesamtaufwand des Verfahrens  $\mathcal{O}(S \log(N))$ . Auf einem Pentium II 400Mhz, 256MB RAM benötigt die Berechnung einer Verteilung für typische Modelle (weniger als  $10^4$  Dreiecke) etwa 10 Sekunden.

## 2.2 Fourier Deskriptoren und Kugelfunktionen

Beiden Methoden liegt die gleiche Idee zugrunde. Es wird ein Satz orthogonaler, normierter Vektoren  $\{\mathbf{e}_k\}$  gewählt, die den Musterraum vollständig beschreiben, das heißt eine Basis für ihn bilden. Diese seien so gewählt, dass sich die Transformation  $\epsilon$ , unter denen die Merkmale invariant sein sollen, durch eine punktweise Multiplikation auf die Darstellung des Musters  $\mathbf{x}$  zur Basis  $\{\mathbf{e}_k\}$  auswirkt, das heißt

$$\mathbf{e}_k^\dagger \epsilon(\mathbf{x}) = T_k(\epsilon) \mathbf{e}_k^\dagger \mathbf{x}. \quad (2.1)$$

Diese Eigenschaft bietet die Möglichkeit, durch geeignete Normierungen eine invariante Darstellung zu erhalten.

Die Fourier Deskriptoren behandeln als Musterraum die geschlossenen Konturen in der Ebene. Diese werden durch eine stetige  $2\pi$ -periodische Funktion  $x(t) : \mathbb{R} \mapsto \mathbb{C}$  beschrieben. Diese Funktion muss hinsichtlich der Äquivalenztransformation einheitlich parametrisiert sein. Für Drehungen wählt man zum Beispiel das Bogenmaß als

Parametrisierung. Seien die  $\{\mathbf{e}_k\}$  die Fourier Basis<sup>2</sup>

$$e_k(t) = \frac{1}{\sqrt{2\pi}} e^{-ikt}.$$

Die euklidische Bewegung drückt sich durch eine Verschiebung  $a \in \mathbb{C}$  und einer Drehung um den Winkel  $\phi$  aus. Zudem ist die Wahl des Aufpunkts  $t_0$  beliebig, also ergibt sich für die gesamte Äquivalenztransformation  $\epsilon(x(t)) = e^{i\phi}x(t - t_0) + a$ . Diese wirkt sich auf die Fourierdarstellung  $\tilde{x}_k = \mathbf{e}_k^\dagger \mathbf{x}$  des Polygonzuges wie folgt aus

$$\mathbf{e}_k^\dagger \epsilon(\mathbf{x}) = \begin{cases} e^{i\phi}\tilde{x}_0 + \sqrt{2\pi}a & \text{falls } k = 0 \\ e^{i(\phi+kt_0)}\tilde{x}_k & \text{sonst.} \end{cases}$$

Die Äquivalenztransformation wirkt sich also für die Koeffizienten mit  $k \neq 0$  wie in (2.1) gesagt aus. Die Verschiebung des Objekts wirkt sich nur auf den nullten Koeffizienten aus, der auch als Linienschwerpunkt des Polygonzuges interpretiert werden kann. Der nullte Koeffizient enthält also gerade die Translationsinformation und kann für eine invariante Beschreibung weggelassen werden. Sehr naheliegende invariante Merkmale sind die Quadrate der komplexen Beträge der Fourierkoeffizienten  $\tilde{x}_k\tilde{x}_k^*$ . Diese Darstellung ist jedoch nicht vollständig; Polygonzüge mit gleicher Autokorrelierten<sup>3</sup> werden gleiche Merkmale zugeordnet. Eine Möglichkeit, vollständige Merkmale anzugeben, ist zum Beispiel folgende

$$I_k(\mathbf{x}) = \frac{\tilde{x}_k}{\tilde{x}_1} e^{-i(k-1)(\arg(\tilde{x}_r) - \arg(\tilde{x}_{r-1}))},$$

wobei  $r$  ein für alle Muster fest zu wählender Parameter ist. Die Verhältnisbildung  $\frac{\tilde{x}_k}{\tilde{x}_1}$  macht das Merkmal unabhängig von einer Drehung (und auch einer Skalierung). Der Parameter  $r$  muss so gewählt werden, dass die Koeffizienten  $\tilde{x}_r$  und  $\tilde{x}_{r-1}$  betragsmäßig möglichst groß sind, dann lassen sich die Argumente stabil berechnen. Ist dies aus Symmetriegründen nicht möglich, so gibt es auch allgemeinere Methoden Merkmale zu bilden; ausführliche Darstellungen findet man in [35]. Praktisch ist die Vollständigkeit nur eingeschränkt zu erreichen, da nur endlich viele  $I_k$  tatsächlich berechnet werden können. Es muss sich also, um Vollständigkeit zu erreichen, auf 'frequenzbegrenzte' Konturen beschränkt werden. Der Aufwand ergibt sich zu  $\mathcal{O}(nk)$ , wobei  $k$  die Grenzfrequenzzahl ist.

---

<sup>2</sup>Die Orthogonalität ist im Sinne des Standardskalarprodukts  $\mathbf{x}^\dagger \mathbf{y} = \int_0^{2\pi} x(t)^* y(t) dt$  zu verstehen, wobei  $*$  die komplexe Konjugation ist.

<sup>3</sup>Die Korrelierte ist definiert als  $\mathbf{x}\#\mathbf{y} = \int_0^{2\pi} x(t'-t)y(t')dt'$  und es gilt gerade  $\mathbf{e}_k^\dagger(\mathbf{x}\#\mathbf{x}) = \tilde{x}_k\tilde{x}_k^* = |\tilde{x}_k|^2$ . Man kann also genau die Autokorrelierte aus dem Betragsspektrum zurückgewinnen, es gibt aber verschiedene Polygonzüge mit gleicher Autokorrelierten.

Kugelfunktionen (oder in der Literatur auch *Spherical Harmonics*) können verwendet werden, um Merkmale für sphärische 3D-Modelle und Volumendaten zu gewinnen. Ein sphärisches Modell  $\mathbf{x}$  wird durch eine Funktion  $x(\varphi, \theta) : [0, 2\pi] \times [0, \pi] \mapsto \mathbb{R}^+$  beschrieben. Jedem Punkt auf einer Sphäre wird ein positiver Wert zugeordnet, der als Abstand vom Schwerpunkt des Objekts zu dessen Oberfläche interpretiert wird. Die Kugelfunktionen<sup>4</sup>

$$\mathbf{e}_l^m = \{Y_l^m(\varphi, \theta)\} \quad \text{mit } m \in \{-l, \dots, l\}$$

bilden eine orthonormale Basis. Die Unterräume  $U_l = \text{span}(\mathbf{e}_l^{-l}, \mathbf{e}_l^{-l+1}, \dots, \mathbf{e}_l^l)$  sind unter Drehungen invariant. Die Projektion des Objekt  $\mathbf{x}$  auf diesen  $2l+1$ -dimensionalen Unterraum transformiert bei einer Drehung des dreidimensionalen Koordinatensystems orthogonal

$$\mathbf{e}_l^\dagger \epsilon(\mathbf{x}) = T_l(\epsilon) \mathbf{e}_l^\dagger \mathbf{x},$$

wobei  $\mathbf{e}_l^\dagger \mathbf{x}$  die orthogonale Projektion von  $\mathbf{x}$  auf  $U_l$  ist und  $T_l(\epsilon)$  die orthogonale Transformation. Somit bleibt der Betrag des  $2l+1$  dimensionalen Vektors  $\mathbf{e}_l^\dagger \mathbf{x}$  bei Drehungen erhalten und man kann invariante Merkmale durch Betragsbildung angeben

$$I_l = \|\mathbf{e}_l^\dagger \mathbf{x}\|^2.$$

Die Merkmale  $I_l$  kann man, ähnlich wie bei den Fourier Deskriptoren, als Frequenzspektrum des Objekts interpretieren. Somit ist von dieser Beschreibung keine Vollständigkeit zu erwarten. Für Beschreibungen, die Vollständigkeit garantieren, siehe man [22]. Neuere Experimente und Vergleiche wurden in [1] durchgeführt. Für Anwendungen der Kugelfunktionen auf Volumendaten siehe man [15].

## 2.3 Gruppenmittelung

Eine sehr allgemeine Methode zur Gewinnung invarianter Merkmale ist die Gruppenmittelung für kompakte Gruppen. Die Idee besteht darin, eine Eigenschaft  $f$  eines Objektes  $\mathbf{x}$ , welche nicht invariant unter Gruppenwirkung  $g \in G$  ist, durch eine Mittelung über alle Ausprägungen der Eigenschaft  $f$  unter der Gruppenwirkung  $g$  auf  $\mathbf{x}$  invariant zu machen

$$I_f(\mathbf{x}) = \frac{1}{V(G)} \int_G f(g\mathbf{x}) dg.$$

Diese Methode hat zum Beispiel in [2] für dreidimensionale Volumendaten erfolgreich Anwendung gefunden. Es werden 'lokale' Eigenschaften  $f$  eines Volumenmodells  $\mathbf{x}$ ,

---

<sup>4</sup>Die Kugelfunktionen sind definiert als  $Y_l^m(\varphi, \theta) = P_l^m(\cos \theta) e^{im\varphi}$ , wobei die  $P_l^m$  die Legendrepolynome sind,  $P_l^m(u) \sim (i\sqrt{1-u^2})^m \frac{d^{l+m}}{du^{l+m}}(u^2-1)^l$ . Für eine genauere Darstellung siehe [13].

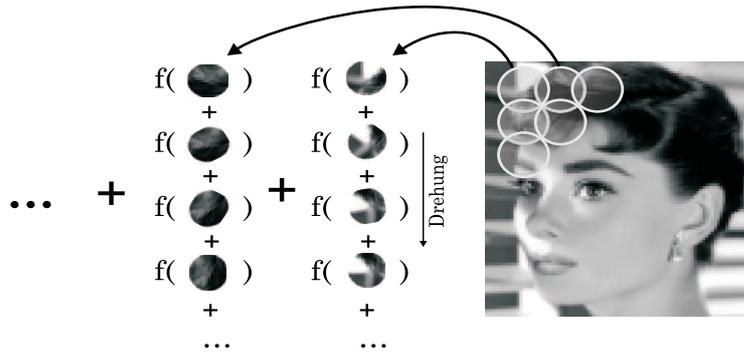


Abbildung 2.1: **Integralinvarianten für Bilder**

welches durch ein skalares Feld  $x(\vec{r})$  (bei Graubilddaten) dargestellt wird, benutzt, um lageinvariante Merkmale zu gewinnen. Für  $f$  wählt man zum Beispiel Monome in  $x$  um einen Punkt  $\vec{a}$ , die sich nur auf einen kleinen Nachbarschaftsbereich erstrecken. Das invariante Integral für die Translations- und Rotationsgruppe lässt sich wie folgt schreiben:

$$I_f(\mathbf{x}) = \int_{\vec{a}} \int_R f(x(R^T(\vec{r} - \vec{a}))) dR d\vec{a}.$$

Das Integral kann so interpretiert werden: man betrachte das Modell  $\mathbf{x}$  an einem Punkt  $\vec{a}$  und werte die Funktion  $f$  an diesem Punkt des Volumenbildes für alle Rotationen um den Punkt  $\vec{a}$  aus und summiere darüber. Dies geschehe für jeden Punkt des Volumens, wobei die Ergebnisse wiederum summiert werden. Bei der Summation über die Rotationen muss beachtet werden, dass jedes Raumwinkelement mit gleichem Anteil in die Summe einfließt. Da die Auswertung von  $f$  nur eine kleine Nachbarschaft der Punkte um  $\vec{a}$  benötigt, wird bei der praktischen Berechnung die Rotation auch nur in geringer Nachbarschaft um diesen Punkt herum ausgeführt. In Abbildung 2.1 ist das Vorgehen für den zweidimensionalen Fall veranschaulicht.

Betrachten wir noch kurz endliche Matrixgruppen etwas genauer. Hier ist es möglich, vollständige Merkmalsmengen anzugeben. Betrachtet man eine endliche Matrixgruppe  $G \subset GL(\mathbb{C}, n)$  und einen Vektor  $\mathbf{x} \in \mathbb{C}^n$ , dann sagt ein Satz von Emmy Noether (siehe [26]), dass die Gruppenmittel über die Monome<sup>5</sup> vom Grad kleiner gleich  $|G|$  eine vollständige Merkmalsmenge liefern. Es existieren

$$T_n^{|G|} = \binom{n + |G|}{n}$$

Monome vom Grad kleiner  $|G|$ . Es sind jedoch nicht immer alle  $T_n^{|G|}$  Merkmale nötig, um Vollständigkeit zu erreichen, vor allem, wenn es einschränkende Bedingungen im Objektraum  $\mathbb{C}^n$  gibt. Es existieren aber auch Gruppen, für die alle  $T_n^{|G|}$  Merkmale für die Vollständigkeit nötig sind, das heißt  $T_n^{|G|}$  ist als Obergrenze optimal.

<sup>5</sup>Ein Monom  $f : \mathbb{C}^n \mapsto \mathbb{C}$  vom Grad  $d$  hat die Form  $f(\mathbf{x}) = \prod_{i=1}^n x_i^{d_i}$  mit  $\sum_{i=1}^n d_i = d$  und  $d_i \in \mathbb{N}_0$ .

# Kapitel 3

## Vorstellung des Rahmenwerks

In diesem Kapitel soll eine allgemeine Art der Merkmalsgewinnung für Drahtgittermodelle vorgestellt werden. Zu Beginn werden die mathematischen Grundlagen gelegt und die Terminologie festgelegt. Begriffe mit strenger Bedeutung werden beim ersten Auftauchen *kursiv* gedruckt und im Kontext definiert.

In Abschnitt 3.2 soll am Beispiel einer physikalischen Feldgleichung das Vorgehen motiviert werden. Die daraus resultierenden Anforderungen werden im folgenden Abschnitt präzisiert, jedoch weiter in einem möglichst allgemeinen Rahmen gehalten. Es werden zwei sich daraus ergebende Methoden der Merkmalsgewinnung vorgestellt, und diese bezüglich ihrem Laufzeitverhalten untersucht. Im letzten Abschnitt werden, noch einmal in einem möglichst allgemeinen Rahmen, die Eigenschaften der Merkmale in Bezug auf Vollständigkeit untersucht. Es werden Bedingungen an den Merkmalsgewinnungsprozess gestellt, die zur Vollständigkeit notwendig sind.

### 3.1 Grundlagen

**Graphen** Ein *ungerichteter Graph*  $g = (V, E)$  besteht aus einer endlichen Knotenmenge  $V \simeq \{1, \dots, n\}$  und einer Kantenmenge  $E \subset V \times V$  mit  $(i, j) \in E \Rightarrow (j, i) \in E$ . Im folgenden werden nur *zusammenhängende Graphen* betrachtet, das heißt für je zwei Knoten existiert immer ein, die Knoten verbindender, Pfad. Zwei Graphen  $g, g'$  sind isomorph  $g = g'$ , falls eine eindeutige Abbildung  $\sigma : V \mapsto V'$  existiert, so dass  $E' = \sigma(E)$ .

**Gitter** Ein *knoten-attributierter Graph* besitzt zusätzlich eine Menge von Abbildungen der Form  $a : V \mapsto \mathbb{K}$  (bzw. eine Menge von Vektoren  $\mathbf{a} \in \mathbb{K}^n$ ), die jedem Knoten ein Attribut  $a$  zuordnet. Im weiteren ist das Attribut  $\vec{x} : V \mapsto \mathbb{R}^d$  der Ort eines Knotens und kurz  $\vec{x}_i = \vec{x}(i)$ . Ein *orts-attributierter Graph*  $m = (V, E, \{\vec{x}_i\})$  nennen wir auch *d-Gitter*. Die Menge aller d-dimensionalen, zusammenhängenden Gitter wird

im folgenden  $\mathbb{G}_d$  genannt. Zwei Gitter  $m, m'$  sind gleich,  $m \simeq m'$ , falls ein Graphenisomorphismus  $\sigma$  zwischen den zugrundeliegenden Graphen existiert mit  $\vec{x}'_i = \vec{x}_{\sigma(i)}$ . Zwei Gitter  $m, m'$  sind unter der euklidischen Bewegung äquivalent,  $m \sim m'$ , falls ein Graphenisomorphismus  $\sigma$  und eine euklidische Bewegung  $\vec{x}' = R\vec{x} + \vec{a}$  existiert mit  $\vec{x}'_i = R\vec{x}_{\sigma(i)} + \vec{a}$ .

**Merkmale** Ein *Merkmal*  $I(m)$  ist eine Abbildung, die unabhängig von der *Repräsentation* von  $m$  ist, das heißt  $m \simeq m' \Rightarrow I(m) = I(m')$ . Ein *lageinvariantes Merkmal*  $I(m)$  eines Gitters ist invariant gegenüber der euklidischen Bewegung  $\vec{x}' = R\vec{x} + \vec{a}$ , das heißt  $m \sim m' \Rightarrow I(m) = I(m')$ . Für ein *vollständiges Merkmal* gilt  $I(m) = I(m') \Rightarrow m \sim m'$ , das heißt  $I(m)$  induziert eine eindeutig umkehrbare Abbildung zwischen den Äquivalenzklassen  $[m]_{\sim} = \{m' \in \mathbb{G} \mid m \sim m'\}$  und dem Merkmalsraum. Für allgemeinere Betrachtungen und ausführliche Definitionen siehe [26].

**Lineare Algebra** Ein Hilbertraum  $\mathbb{H}$  ist ein  $\mathbb{C}$ -Vektorraum versehen mit einem Skalarprodukt  $\mathbf{v}^\dagger \mathbf{w}$ , wobei  $\mathbf{v}^\dagger \mathbf{w} = (\mathbf{w}^\dagger \mathbf{v})^*$  gilt (hierbei ist  $*$  die komplexe Konjugation). Wir werden uns nur mit endlichdimensionalen Hilberträumen beschäftigen, deren Vektoren  $\mathbf{v} \in \mathbb{H}$  daher Darstellungen in  $\mathbb{C}^n$  haben. Genauso haben die linearen Abbildungen, die  $\mathbb{H}$  in sich selbst überführen, Darstellungen in  $\mathbb{C}^{n \times n} = \text{Mat}(n, \mathbb{C})$ , wobei sich die Anwendung der linearen Abbildung durch die Matrizenmultiplikation ergibt. Jedes  $\mathbf{A} \in \text{Mat}(n, \mathbb{C})$  besitzt ein adjungiertes  $\mathbf{A}^\dagger \in \text{Mat}(n, \mathbb{C})$  definiert durch  $\mathbf{w}^\dagger (\mathbf{A}\mathbf{v}) = (\mathbf{A}^\dagger \mathbf{w})^\dagger \mathbf{v}$ , wobei sich das Adjungieren auf die Matrix durch Transponieren und komplex Konjugieren ergibt. Spezielle Matrizen, für die  $\mathbf{A} = \mathbf{A}^\dagger$  ( $\mathbf{A} = -\mathbf{A}^\dagger$ ) gilt, heißen hermitesch (antihermitesch). Hermitesche und antihermitesche Matrizen unterscheiden sich nur um die imaginäre Einheit  $\mathbf{i}$ , das heißt jede antihermitesche Matrix ist darstellbar als eine hermitesche Matrix multipliziert mit  $\mathbf{i}$ . Die  $n$ -dimensionale Identitätsmatrix wird als  $\mathbf{I}_n$  geschrieben. Folgende Teilmengen von  $\text{Mat}(n, \mathbb{C})$  werden später von Bedeutung sein

$$O(n) := \{\mathbf{U} \in \text{Mat}(n, \mathbb{R}) \mid \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}_n\} \quad (\text{orthogonale Transformation})$$

$$U(n) := \{\mathbf{U} \in \text{Mat}(n, \mathbb{C}) \mid \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}_n\} \quad (\text{unitäre Transformation}).$$

Die folgenden Definitionen werden wir im speziellen benutzen

$$P(n) := \{\mathbf{P} \in O(n) \mid \mathbf{P} \text{ ist Permutationsmatrix}\}$$

$$T(n) := \{\mathbf{T} \in O(n) \mid \mathbf{T} \text{ ist zykl. Translation- oder Spiegelungsmatrix}\}$$

$$U_{\mathbf{v}}(n) := \{\tilde{\mathbf{U}} \in U(n) \mid \tilde{\mathbf{U}}\mathbf{v} = \mathbf{v}\} \quad (\text{den Vektor } \mathbf{v} \text{ erhaltende Unitäre}).$$

Der Produktraum  $\mathbb{H}$  zweier Hilberträume  $\mathbb{H}'$  und  $\mathbb{H}''$  schreiben wir als  $\mathbb{H} = \mathbb{H}' \otimes \mathbb{H}''$ , wobei sich die Dimension des Produktraum als Produkt der beiden anderen Dimensionen

ergibt. Faktorisierte Operatoren werden mit dem Kroneckerprodukt  $\otimes$  geschrieben. Also wird zum Beispiel ein Operator  $\mathbf{A}$ , der nur auf  $\mathbb{H}'$  wirkt als  $\mathbf{A} \otimes \mathbf{I}_{\mathbb{H}'}$  geschrieben.

Für Vektoren  $\mathbf{v}$  in Komponentenschreibweise  $\mathbf{v} = \{v_i\}$  sind die Indizes im folgenden immer modulo  $n$  zu verstehen, das heißt  $v_{i \bmod n} = v_i$ .

Für genauere Beschreibungen wird auf Standardlehrbücher der Linearen Algebra verwiesen oder im speziellen für Matrizen siehe [33, 34].

## 3.2 Motivation

Die grundlegende Idee der in dieser Arbeit entwickelten Invarianten ist eine Abbildung, die zwischen einem Gitter  $m$  und einer Matrix  $\mathbf{H}$  vermittelt. Die Abbildung muss so beschaffen sein, dass sich aus  $\mathbf{H}(m)$  wieder ein zu  $m$  äquivalentes Gitter zurückgewinnen lässt. Diese Abbildung wird später auch als *Kodierung* des Gitters bezeichnet. Nachdem die Matrix  $\mathbf{H}$  aus einem Gitter gewonnen wurde, wird mit Mitteln der Linearen Algebra versucht, Eigenschaften aus der Matrix herauszuarbeiten, die nicht von der Repräsentation und von der Lage des Gitters abhängen. Im folgenden soll aus der physikalischen Perspektive die Vorgehensweise plausibel gemacht werden.

### 3.2.1 Physikalische Sicht

Die Homogenität und Isotropie des Raumes zwingen physikalische Feldgleichungen invariant unter der euklidischen Bewegung zu sein. Bei der Lösung solcher Gleichungen beschränkt man sich meist nur auf ein zusammenhängendes, beschränktes Gebiet des Raumes und fordert, dass die Lösung außerhalb des Gebietes null wird. Bei der numerischen Lösung partieller Differentialgleichungen kann man das zu simulierende Gebiet durch ein Gitter repräsentieren. Es ist durchaus zu erwarten, dass die Lösungen einer Differentialgleichung auf eine charakteristische Weise von der Form des Simulationsgebietes abhängen und somit von dem repräsentierenden Gitter. Betrachten wir zum Beispiel die zeitabhängige Schrödingergleichung eines freien Teilchens ([30]) in einem Potentialtopf  $\Omega$ ,

$$\Delta \mathbf{v}(\mathbf{t}) = \mathbf{i} \frac{d}{dt} \mathbf{v}(\mathbf{t}).$$

Für genügend hohe Potentiale kann gefordert werden, dass die Wellenfunktion  $\mathbf{v}$  des Teilchens auf dem Rand  $\partial\Omega$  des Potentialtopfes verschwindet. Diskretisiert man nun den kontinuierlichen, hermiteschen Differentialoperator  $\Delta$  mittels einem aus der Numerik bekannten Verfahren (Finite Differenzen, Finite Elemente), erhält man für den Differentialoperator  $\Delta$  eine korrespondierende Matrix  $\mathbf{H}$ , die auf charakteristische Weise von diesen Randbedingungen abhängt. Jede einzelne Komponente des jetzt endlichdimensionalen Vektors  $\mathbf{v}$  ist mit einem Knoten bzw. Element des Diskretisierungsgitters

$m$  zu assoziieren. Die Lösung der Differentialgleichung zu einem Anfangsvektor  $\mathbf{v}_0$  kann nun sofort angegeben werden

$$\mathbf{v}(\mathbf{t}) = e^{-i\mathbf{H}t}\mathbf{v}_0.$$

Bekanntlich werden in der Quantenmechanik die Betragsquadrate der Komponenten  $v_i(t) = \mathbf{e}_i^\dagger \mathbf{v}(\mathbf{t})$  als Wahrscheinlichkeit interpretiert, ein Teilchen im Zustand  $\mathbf{v}$  am Ort  $\mathbf{e}_i$  anzutreffen. Allgemein kann das Skalarprodukt  $\mathbf{w}^\dagger \mathbf{v}$  als Übergangswahrscheinlichkeit von Zustand  $\mathbf{v}$  in den Zustand  $\mathbf{w}$  interpretiert werden. Anhand dieser Wahrscheinlichkeiten wollen wir nun invariante Merkmale angeben. Wir suchen einen Startvektor  $\mathbf{v}_0$ , der unabhängig von der Repräsentation des Gitters ist, zum Beispiel eine Gleichverteilung  $\mathbf{e}_i^\dagger \mathbf{v}_0 = 1$ . Jetzt bestimmen wir die Wahrscheinlichkeit, ausgehend von diesem Zustand nach einem Zeitraum  $t$ , wieder im Zustand  $\mathbf{v}_0$  zu landen, formal

$$I_t = \mathbf{v}_0^\dagger \mathbf{v}(\mathbf{t}) = \mathbf{v}_0^\dagger e^{-i\mathbf{H}t} \mathbf{v}_0.$$

Anschaulich sollten die Skalarprodukte  $\mathbf{v}_0^\dagger \mathbf{v}(\mathbf{t})$  lageinvariante Merkmale bilden, da die Wahrscheinlichkeit nur im Bezug auf einen Zustand gebildet wurde, der unabhängig von der Repräsentation des Gitters ist. Betrachten wir die Invarianzeigenschaften etwas genauer. Hat man  $\mathbf{H}$  durch Finite Differenzen approximiert, dann wirkt sich ein Wechsel der Repräsentation des Gitters als Permutation  $\mathbf{P}$  auf die Matrix  $\mathbf{H}$  aus,

$$\mathbf{H}' = \mathbf{P}^\dagger \mathbf{H} \mathbf{P}.$$

Auf den speziell gewählten Startvektor  $\mathbf{v}_0 = \mathbf{1}$  wirkt sich eine Permutation nicht aus, das heißt  $\mathbf{P}_\sigma \mathbf{v}_0 = \mathbf{v}_0$ . Somit ist offensichtlich, dass  $I_t$  von einem Wechsel der Repräsentation des Gitters unberührt bleibt

$$\mathbf{v}_0^\dagger e^{-i\mathbf{H}'t} \mathbf{v}_0 = \mathbf{v}_0^\dagger \mathbf{P}^\dagger e^{-i\mathbf{H}t} \mathbf{P} \mathbf{v}_0 = \mathbf{v}_0^\dagger e^{-i\mathbf{H}t} \mathbf{v}_0.$$

Im folgenden Abschnitt soll nun die im Vorangegangenen angedeutete Vorgehensweise in einem etwas allgemeinerem Rahmen formuliert werden.

### 3.3 Kodierungen

Wie schon in Abschnitt 3.2 angedeutet, haben für Konstruktionsvorschriften, die für die lageinvariante Merkmalsgewinnung dienen sollen, gewisse Gesetze zu gelten. Eine Konstruktionsvorschrift ist eine Abbildung

$$\mathbf{H}(m) : \mathbb{G} \mapsto \text{Mat}(n, \mathbb{K}).$$

Folgende Regeln haben zu gelten:

1. Eine Permutation der Repräsentation eines Gitters  $m$  muss sich auf die Konstruktion über eine Permutationsmatrix  $\mathbf{P}_\sigma \in P(n)$  auswirken:

$$\mathbf{H}(\sigma(m)) = \mathbf{P}_\sigma^\dagger \mathbf{H}(m) \mathbf{P}_\sigma. \quad (3.1)$$

2. Die euklidische Bewegung  $\varepsilon$  hat sich durch eine unitäre, also eine Skalarprodukt erhaltende, Transformation  $\mathbf{U} \in U(n)$  auf die Konstruktion auszuwirken:

$$\mathbf{H}(\varepsilon(m)) = \mathbf{U}_\varepsilon^\dagger \mathbf{H}(m) \mathbf{U}_\varepsilon. \quad (3.2)$$

Gelten diese Regeln für eine Konstruktion, dann wird sie *Kodierung* genannt. Absichtlich werden hier noch nicht die Zusammenhänge  $\sigma \mapsto \mathbf{P}_\sigma$  und  $\varepsilon \mapsto \mathbf{U}_\varepsilon$  explizit angegeben, da sie immer von der konkret vorliegenden Konstruktionsvorschrift abhängen. Folgende Abbildung soll die Beziehung verdeutlichen:

$$\begin{array}{ccc} m & \longrightarrow & \mathbf{H} \\ \sigma, \varepsilon \downarrow & & \downarrow \mathbf{U}_\varepsilon, \mathbf{P}_\sigma \\ m' & \longrightarrow & \mathbf{H}' \end{array}$$

Als Beispiel für eine einfache Distanz-Kodierung eines Gitters sei folgende Konstruktion  $\{h_{ij}\} = \mathbf{H}(m) \in \text{Mat}(n, \mathbb{R})$  gegeben:

$$h_{ij} = \begin{cases} f(\|\vec{x}_i - \vec{x}_j\|) & \text{falls } (i, j) \in E \\ 0 & \text{sonst.} \end{cases} \quad (3.3)$$

Da die Kodierung nur von den Abständen zwischen den Punkten abhängt, wirkt sich eine euklidische Bewegung  $\varepsilon$  nicht auf die Kodierung aus, also  $\mathbf{U}_\varepsilon = 1$ . Weiter ist zu bemerken, dass es sich bei Kodierung (3.3) um keine eindeutige Abbildung hinsichtlich Gitteräquivalenz handelt. Es ist nicht zu erwarten, dass beliebige Gitter wieder aus ihrer Kodierung zurückgewonnen werden können. Falls dies jedoch der Fall ist, also die Kodierung umkehrbar ist, soll sie *vollständig* heißen.

Im oberen Beispiel gilt im speziellen  $\mathbf{H}^\dagger(m) = \mathbf{H}(m)$ , das heißt die Matrix ist symmetrisch (hermitesch); wir sagen dann, die Kodierung ist symmetrisch (hermitesch). Genauso gibt es antisymmetrische (antihermitesche) Kodierungen. Aus der Lie-Theorie [9] wissen wir, dass die Matrixabbildung  $\mathbf{A} \mapsto e^{\mathbf{A}} = \sum_{i=0}^{\infty} \frac{\mathbf{A}^i}{i!}$  zwischen den antisymmetrischen und orthogonalen Matrizen vermittelt (antihermitesch und unitär). Ist zum Beispiel eine hermitesche Kodierung gegeben, dann bildet  $e^{\mathbf{A}}$  die hermitesche Kodierung auf eine unitäre Kodierung ab, weil die Abbildung  $e^{\mathbf{A}}$  die Bedingungen (3.1) und (3.2) erhält.

## 3.4 Merkmalsgewinnung

Hat man nun eine Kodierung  $\mathbf{H}(m)$  eines Gitters gegeben, muss versucht werden, Merkmale  $I(m) = I(\mathbf{H}(m))$  aus der Matrix  $\mathbf{H}$  herauszuarbeiten, die invariant unter Permutation  $\mathbf{P}_\sigma$  und unter euklidischer Bewegung  $\mathbf{U}_\varepsilon$  sind

$$I(\mathbf{P}_\sigma^\dagger \mathbf{H} \mathbf{P}_\sigma) = I(\mathbf{H}) \quad (3.4)$$

$$I(\mathbf{U}_\varepsilon^\dagger \mathbf{H} \mathbf{U}_\varepsilon) = I(\mathbf{H}). \quad (3.5)$$

Die wohl bekannteste Abbildung, die diese Bedingungen erfüllt, ist die Determinante  $\det(\mathbf{H}) : \text{Mat}(n, \mathbb{K}) \mapsto \mathbb{K}$ . Die Merkmalsmenge  $I_\lambda = \det(\mathbf{H} - \lambda)$  hängt gerade vom Spektrum der Matrix  $\mathbf{H}$  ab und scheint sich als Merkmal zu eignen, jedoch wird durch die Determinantenanwendung eine Reihe weiterer Invarianztransformationen zugelassen, nämlich beliebige unitäre Transformationen, die eventuell Mehrdeutigkeiten erzeugen könnten. Das heißt es könnte ein unitäres  $\mathbf{U}$  existieren, welches die Kodierung  $\mathbf{H}(m)$  eines Gitters  $m$  in eine Kodierung  $\mathbf{H}(m')$  transformiert, wobei  $m'$  ein zu  $m$  nicht äquivalentes Gitter ist.

Im folgenden sollen nun zwei Methoden der Merkmalsgewinnung vorgestellt werden, die diese Menge der zusätzlichen Invarianztransformationen verkleinert bzw. komplett verschwinden lässt.

### 3.4.1 Merkmalsgewinnung durch Mittelung

Angenommen, es sei ein Vektor  $\mathbf{v}_0$  gegeben, der den folgenden beiden Gleichungen genügt:

$$\mathbf{v}_0 = \mathbf{U}_\varepsilon \mathbf{v}_0 \quad \mathbf{v}_0 = \mathbf{P}_\sigma \mathbf{v}_0, \quad (3.6)$$

dann lassen sich aus diesem, in Anlehnung an die physikalische Motivation in Abschnitt (3.2), die Vektoren  $\mathbf{v}_k$  konstruieren

$$\mathbf{v}_k(m) = (\mathbf{H}(m))^k \mathbf{v}_0.$$

Die paarweise gebildeten Skalarprodukte dieser Vektoren ergeben dann die lageinvariante Merkmalsmatrix  $\mathbf{C}(m) = \{c_{jk}(m)\} \in \text{Mat}(n, \mathbb{K})$

$$\boxed{c_{jk}(m) = (\mathbf{v}_j(m))^\dagger \mathbf{v}_k(m)}. \quad (3.7)$$

Die Invarianz kann sofort durch Einsetzen überprüft werden:

$$\begin{aligned}
(\mathbf{v}_j(m'))^\dagger \mathbf{v}_k(m') &= \mathbf{v}_0^\dagger (\mathbf{H}^\dagger(m'))^j (\mathbf{H}(m'))^k \mathbf{v}_0 \\
&= \mathbf{v}_0^\dagger \tilde{\mathbf{U}}^\dagger (\mathbf{H}^\dagger(m))^j \tilde{\mathbf{U}} \tilde{\mathbf{U}}^\dagger (\mathbf{H}(m))^k \tilde{\mathbf{U}} \mathbf{v}_0 \\
&= \underbrace{(\tilde{\mathbf{U}} \mathbf{v}_0)^\dagger}_{\mathbf{v}_0^\dagger} (\mathbf{H}^\dagger(m))^j (\mathbf{H}(m))^k \underbrace{(\tilde{\mathbf{U}} \mathbf{v}_0)}_{\mathbf{v}_0} \\
&= \mathbf{v}_0^\dagger (\mathbf{H}^\dagger(m))^j (\mathbf{H}(m))^k \mathbf{v}_0 = (\mathbf{v}_j(m))^\dagger \mathbf{v}_k(m), \tag{3.8}
\end{aligned}$$

wobei die Invarianz genau für die unitären Transformationen  $\tilde{\mathbf{U}} \in U_{\mathbf{v}_0}(n)$  gilt, die den Vektor  $\mathbf{v}_0$  fest lassen. Dies wurde ja unter anderem in (3.6) gefordert. Somit ist die Menge der zusätzlichen Invarianztransformationen etwas eingeschränkter.

Für allgemeine Kodierungen besteht die Merkmalsmatrix aus  $n^2$  auf nicht triviale Weise zusammenhängenden Zahlen. Jedoch ergibt sich für (anti-)hermitesche und unitäre Kodierungen eine besonders einfache Struktur für die Merkmalsmatrix, welche sich nur noch aus  $\mathcal{O}(n)$  unabhängigen Zahlen zusammensetzt. Für hermitesche Kodierungen ergibt sich zum Beispiel

$$c_{jk}(m) = \mathbf{v}_0^\dagger (\mathbf{H}(m))^{j+k} \mathbf{v}_0 = I_{j+k}(m),$$

das heißt die komplette Merkmalsmatrix ist durch die  $2n - 1$  Zahlen  $I_k$  beschrieben.

$$\boxed{I_k(m) = \mathbf{v}_0^\dagger (\mathbf{H}(m))^k \mathbf{v}_0} \tag{3.9}$$

Es besteht natürlich auch die Möglichkeit, mehr als  $2n - 1$  Merkmale zu berechnen, jedoch werden wir bei den späteren Vollständigkeitsbetrachtungen sehen, dass mehr als  $n$  Vektoren aus den paarweise gebildeten Skalarprodukten nicht zurückgewonnen werden können und somit nur die  $(2n - 1)$  Werte von Nutzen sind.

### 3.4.2 Merkmalsgewinnung durch Histogramme

Wir wollen nun versuchen, die Menge der zusätzlichen Invarianztransformationen noch weiter einzuschränken. Im vorigen Abschnitt haben wir das Merkmal durch (gewichtete) Summation über die Matrixelemente von  $\mathbf{H}^k$  gewonnen, was gleichbedeutend ist mit der Summation über die  $\mathbb{K}$ -Vektoreinträge von  $\mathbf{v}_k = \mathbf{H}^k \mathbf{v}_0$ . Wir wollen nun eine gut in unser Konzept passende Methode zur Histogrammgewinnung angeben, die insbesondere Bezug auf den Startvektor  $\mathbf{v}_0$  nimmt. Die Bildung eines Histogramms über den Wertebereich  $\mathbb{K}$  des Vektors  $\mathbf{v}_k$  kann als punktweise Anwendung einer Urnenfunktion  $f : \mathbb{K} \mapsto \mathbb{K}$  auf den Vektor  $\mathbf{v}_k - \lambda \mathbf{v}_0$  und anschließendem Skalarprodukt mit  $\mathbf{v}_0$  verstanden werden, also

$$\boxed{I_{k\lambda}(m) := \mathbf{v}_0^\dagger f(\mathbf{v}_k - \lambda \mathbf{v}_0)}. \tag{3.10}$$

Häufig wird als Urnenfunktion eine Rechteckfunktion<sup>1</sup>  $r(\lambda)$  verwendet. Falls man für  $\mathbf{v}_0$  den Einsvektor  $\mathbf{1}$  wählt, führt die Form (3.10) unter Benutzung der Rechteckfunktion zu einem gewöhnlichen Histogramm

$$\begin{aligned} I_{k\lambda}(m) &= \mathbf{v}_0^\dagger r(\mathbf{v}_k - \lambda \mathbf{v}_0) = \sum_{i=1}^n r((\mathbf{v}_k - \lambda \mathbf{v}_0)_i) \\ &= \sum_{i=1}^n \begin{cases} 1 & \text{falls } \lambda - \frac{1}{2} < (\mathbf{v}_k)_i < \lambda + \frac{1}{2} \\ 0 & \text{sonst} \end{cases} . \end{aligned}$$

Diese Art der Merkmalsgewinnung ist nun nicht mehr invariant unter beliebigen unitären Transformationen, es bleibt uns sogar nur noch die Permutationsgruppe als Invarianztransformation, da sich eine Permutation nur auf die Ordnung der Komponenten der Vektoren auswirkt, formal

$$\mathbf{v}_0^\dagger f(\mathbf{P}\mathbf{v}_k - \lambda \mathbf{v}_0) = \mathbf{v}_0^\dagger \mathbf{P}f(\mathbf{v}_k - \lambda \mathbf{v}_0) = \mathbf{v}_0^\dagger f(\mathbf{v}_k - \lambda \mathbf{v}_0). \quad (3.11)$$

Das heißt jedoch für lageinvariante Merkmale, dass  $\mathbf{U}_\varepsilon = \mathbf{1}$  gelten muss. In Kapitel 5 werden wir eine Kodierung vorstellen, die dieses Problem löst, indem  $\mathbf{H}$  im Produkt- raum aus Orts- und Knotenraum aufgestellt wird.

### 3.4.3 Die Exponential-Abbildung

Wie schon oben erwähnt, kann mit dem Matrixexponential aus einer bestehenden Ko- dierung eine neue gewonnen werden

$$\mathbf{H} \mapsto e^{\mathbf{H}t}.$$

Betrachtet man  $t \in [0, t_0]$  als kontinuierlich, dann ist durch

$$\boxed{I_t(m) = \mathbf{v}_0^\dagger \mathbf{v}(t) = \mathbf{v}_0^\dagger e^{\sigma_n \mathbf{H}t} \mathbf{v}_0} \quad (3.12)$$

eine Merkmalsfunktion gegeben. Im ersten Moment scheint dies keinen Vorteil zu brin- gen, da in der Praxis der Merkmalsgewinnung  $t$  diskrete Werte annimmt, jedoch bietet es mit Hilfe des Parameters  $\sigma_n$  die Möglichkeit, Gitter mit verschiedener Anzahl von Knoten auf eine einheitliche Weise zu behandeln. Der Parameter sollte so gewählt wer-

---

<sup>1</sup>Die Rechteckfunktion sei wie folgt definiert:

$$r(\lambda) := \begin{cases} 1 & \text{falls } -\frac{1}{2} < \lambda < \frac{1}{2} \\ 0 & \text{sonst} \end{cases} .$$

den, dass die Bogenlänge<sup>2</sup>  $L$  des Merkmals für Objekte mit verschiedener Anzahl von Knoten gleich ist. Es wird in den Experimenten noch darauf eingegangen.

Insbesondere sind die Exponentiale, die durch antihermitesche Matrizen erzeugt werden, unitär. Unitäre Matrizen besitzen nur Eigenwerte mit Betrag 1.

Praktisch kann das Exponential natürlich nicht genau ausgerechnet werden, es muss approximiert werden. Eine Möglichkeit ist, die Reihenentwicklung an bestimmter Stelle abzuberechnen

$$\mathbf{v}_0^\dagger \mathbf{v}(\mathbf{t}) = \sum_{k=0}^m \frac{(\sigma_n t)^k}{k!} \mathbf{v}_0^\dagger \mathbf{H}^k \mathbf{v}_0 = \sum_{k=0}^m \frac{(\sigma_n t)^k}{k!} I_k.$$

Offenbar läuft die praktische Berechnung auf die Bestimmung der  $I_k$  (3.9) hinaus. Es ist also qualitativ nichts Neues hinzugekommen, es werden lediglich die  $I_k$  als Entwicklungskoeffizienten einer kontinuierlichen Merkmalsfunktion interpretiert.

### 3.4.4 Numerische Berechnung und Komplexität

Die praktische Berechnung der Merkmale sollte vor allem unter den Gesichtspunkten der Laufzeitkomplexität und der stabilen numerischen Berechnung betrachtet werden. Im folgenden sollen nun die oben vorgestellten Methoden der Merkmalsgewinnung daraufhin untersucht werden.

Betrachten wir als erstes die Merkmalsgewinnung durch Mittelung aus Abschnitt 3.4.1. Die Berechnung der Vektoren  $\mathbf{v}_k$  kann natürlich iterativ erfolgen

$$\mathbf{v}_k := \mathbf{H} \mathbf{v}_{k-1}. \quad (3.13)$$

Existiert nun ein betragsmäßig größter Eigenwert  $\lambda_{max}$ , dessen Eigenvektor einen Anteil an  $\mathbf{v}_0$  hat, dann hängt das asymptotische Verhalten der Vektoren  $\mathbf{v}_k$  von diesem ab. Es gilt asymptotisch  $\|\mathbf{v}_k\| \approx \|\mathbf{v}_0\| \lambda_{max}^k$ . Für  $|\lambda_{max}| > 1$  wird  $\|\mathbf{v}_k\| \rightarrow \infty$  für große  $k$  gehen, also divergieren. Für  $|\lambda_{max}| < 1$  wird  $\|\mathbf{v}_k\| \rightarrow 0$  und konvergiert somit. Für große  $k$  erscheint dies für eine Implementation auf einem Rechner nicht geeignet, da man mit *Over-* bzw. *Underflows* zu rechnen hat. Nur für unitäre Transformationen bestehen diese Probleme nicht. Für allgemeine Matrizen erscheint folgende Iterationsvorschrift viel sinnvoller:

$$\begin{aligned} \mathbf{v}_k' &:= \frac{1}{w_k} \mathbf{H} \mathbf{v}_{k-1}' \\ w_k &:= \|\mathbf{H} \mathbf{v}_{k-1}'\|. \end{aligned} \quad (3.14)$$

---

<sup>2</sup>Unter der 'Länge' des Merkmals ist die Bogenlänge der Merkmalsfunktion zu verstehen

$$L = \int_0^{t_0} \left\| \frac{dI}{dt} \right\| dt.$$

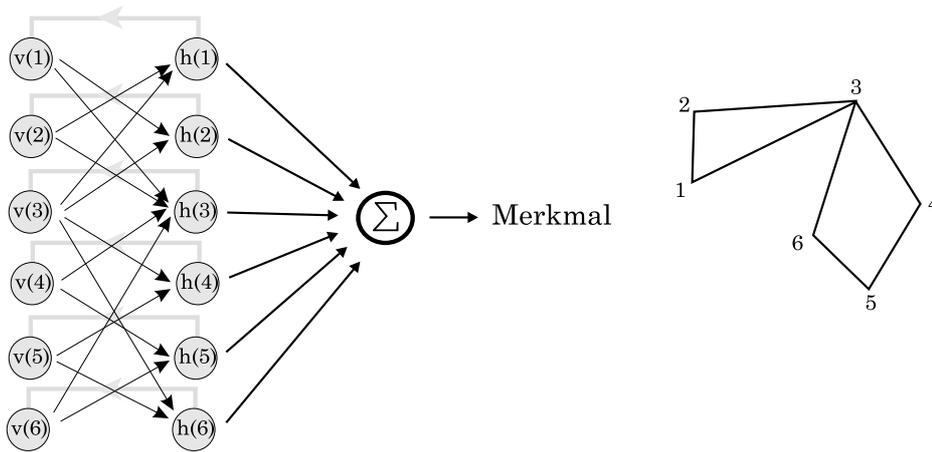


Abbildung 3.1: **Berechnung der Merkmale.** Aus dem Vektor  $\mathbf{v}$  wird in Abhängigkeit der Verbindungsverhältnisse des Gitters der neue Vektor gewonnen. Dieser wird summiert und bildet das Merkmal. Das nächste Merkmal wird durch Ersetzen des alten Vektors mit dem neuen und Wiederholung der Prozedur gewonnen.

Sie ist aber theoretisch gleichbedeutend mit der vorigen, da sich die  $\mathbf{v}_{\mathbf{k}}$  aus den  $\mathbf{v}_{\mathbf{k}'}$  und  $w_k$  berechnen lassen,  $\mathbf{v}_{\mathbf{k}} = \mathbf{v}_{\mathbf{k}'} \prod_{i=0}^k w_i$ . Es werden sozusagen die Richtungsanteile  $\mathbf{v}_{\mathbf{k}'}$  und die Streckungsanteile  $w_k$  des Vektor  $\mathbf{v}_{\mathbf{k}}$  getrennt behandelt; somit können auch die Merkmale  $I_k$  in einen Winkelanteil  $I'_k = \mathbf{v}_0^\dagger \mathbf{v}_{\mathbf{k}'}$  und einen Streckungsanteil  $w_k$  aufgespalten werden.

Von der Laufzeitkomplexität besteht zwischen Verfahren (3.13) und (3.14) bis auf die Berechnung der Norm kein Unterschied. Bei dünn besetzten Matrizen  $\mathbf{H}$ , das heißt die Anzahl der von Null verschiedenen Elemente einer Zeile von  $\mathbf{H}$  ist durch eine Konstante nach oben hin beschränkt, ist der Aufwand einer Anwendung von  $\mathbf{H}$  auf einen Vektor der Ordnung  $\mathcal{O}(n)$ .

In Abbildung 3.1 ist schematisch die Berechnung der Merkmale eines 6-knotigen Gitters dargestellt. Die Verbindungsstruktur zwischen den Speicherknöten  $v(i)$  auf  $h(k)$  spiegelt die Gestalt des Gitters wieder. Beispielhaft wurde die einfache Kodierung (3.3) gewählt, die nur einen Eintrag  $h_{ij}$  ungleich null besitzt, falls es eine Verbindung zwischen Knoten  $i$  und  $j$  gibt. Es besteht also genau dann ein Einfluss von Speicherknöten  $v(i)$  auf  $h(k)$ , falls eine Verbindung zwischen den jeweiligen Gitterknöten  $i$  und  $k$  besteht. Nachdem die  $h(k)$  aus den  $v(k)$  gewonnen worden sind, werden die Merkmale durch Mittelung mit dem  $\mathbf{1}$ -Vektor gewonnen, das heißt die Summe über alle  $h(k)$  Speicherknöten bilden ein Merkmal. Nach der Gewinnung des Merkmals werden die alten Inhalte der Speicherknöten  $v(k)$  mit den neu gewonnen Inhalten in  $h(k)$  ersetzt und die Prozedur beginnt von neuem. Offenbar genügen  $2n$  Speicherzellen zur Gewinnung der Merkmale. Die Berechnung ist zudem parallelisierbar.

Die Gewinnung der kompletten Merkmalsmatrix (3.7) für nicht symmetrische Ma-

trizen erfordert  $n - 1$  Anwendungen von  $\mathbf{H}$ , somit ergibt sich ein Gesamtaufwand von  $\mathcal{O}(n^2)$ . Der Speicherbedarf steigt bei nicht symmetrischen Matrizen auf  $\mathcal{O}(n^2)$ , da sämtliche Vektoren für die Skalarproduktbildung zwischengespeichert werden müssen.

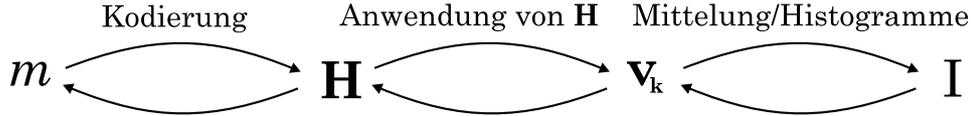
Die Merkmalsgewinnung mittels Histogrammen mit allgemeiner Urnenfunktion ist von weit höherer Komplexität. Wählen wir jedoch eine Rechteckfunktion als Urnenfunktion, so hängt die Laufzeit zur Gewinnung eines Merkmalshistogramms  $\{I_{k\lambda}|\lambda\}$  aus einem Vektor  $\mathbf{v}_k$  nicht von der Unterteilung von  $\lambda$  ab, da für jede Komponente des Vektors  $\mathbf{v}_k$  nur an einer Stelle im Histogramm ein Wert addiert werden muss. Dies kann mit konstantem Aufwand geschehen, da nur festgestellt werden muss, in welche Urne der Wert fällt, was durch eine Rundung geschehen kann. Es ergibt sich somit gerade die Laufzeit  $\mathcal{O}(n)$  für die Gewinnung eines Histogramms mit Rechteckfunktion. Für eine allgemeine Urnenfunktion beträgt die Laufzeit zur direkten Gewinnung des Merkmalshistogramms  $\mathcal{O}(nN)$ , wobei  $N$  die Anzahl der diskreten  $\lambda$ -Abtastwerte ist. Dies liegt daran, dass eine Komponente des Vektors  $\mathbf{v}_k$  Einfluss auf alle Werte des Histogramms haben kann. Histogramme mit allgemeiner Urnenfunktion können auch über einen kleinen Umweg gewonnen werden. Zuerst wird das Histogramm mit Rechteckfunktion gewonnen, dann wird aus diesem Histogramm durch Faltung mit einer allgemeinen Urnenfunktion das neue Histogramm. Falls man die Faltung mittels einer FFT ausführt, ergibt sich ein Aufwand von  $\mathcal{O}(n) + \mathcal{O}(N \log(N))$ .

### 3.5 Vollständigkeitsbetrachtungen

Im folgenden soll in einem möglichst allgemeinen Rahmen versucht werden, geeignete Bedingungen an Kodierung und Merkmalsgewinnung zu formulieren, die die Separationseigenschaften der Merkmale verbessern und im besten Fall Vollständigkeit garantieren. Sowohl in Abschnitt 3.4.1 als auch in 3.4.2 haben die Vektoren  $\mathbf{v}_k = \mathbf{H}^k \mathbf{v}_0$  eine zentrale Rolle gespielt. In beiden Fällen hängen die Merkmale von diesen Vektoren ab, also

$$I_k(m) = I_k(\mathbf{H}(m)) = I(\mathbf{v}_k(\mathbf{H}(m))) = I(\mathbf{H}^k(m)\mathbf{v}_0).$$

Die Mittelungs- und Histogrammmethode unterscheiden sich nur dahingehend, dass sie auf unterschiedliche Art und Weise versuchen, eine gegenüber  $\mathbf{U}_\varepsilon$  und  $\mathbf{P}_\sigma$  invariante Darstellung der Vektoren  $\mathbf{v}_k$  zu geben. Die folgende Abbildung veranschaulicht die Abhängigkeiten in den verschiedenen Stufen des Merkmalsgewinnungsprozesses.



Betrachten wir diese Abhängigkeitskette. An erster Stelle steht die Kodierung des Gitters in einer Matrix  $\mathbf{H}$ , daraufhin wird versucht,  $\mathbf{H}$  vollständig durch die Vektoren  $\mathbf{v}_k$  darzustellen und im letzten Schritt werden schließlich die Vektoren permutationsinvariant und  $\mathbf{U}_\varepsilon$ -invariant auf die endgültigen Merkmale abgebildet. Jeder dieser einzelnen Schritte muss bis auf seine Invarianztransformation eindeutig umkehrbar sein, damit das gesamte Verfahren umkehrbar ist und somit vollständig.

In den folgenden drei Abschnitten werden die einzelnen Stufen des Merkmalsbildungsprozesses auf ihre Invertierbarkeit überprüft. Wir beginnen mit dem letzten Schritt, wobei sich hier vor allem auf die Mittelungsmethode aus Abschnitt 3.4.1 konzentriert werden soll.

### 3.5.1 Die Zurückgewinnung der Vektoren

Betrachten wir also das Mittelungsverfahren aus Abschnitt 3.4.1. Durch die Skalarprodukte  $\mathbf{v}_j^\dagger \mathbf{v}_k$  sind uns in gewissem Sinne alle Winkel zwischen den Vektoren  $\mathbf{v}_k$  bekannt. Es ist plausibel zu vermuten, dass dann die relative Stellung der  $n$  Vektoren im Raum durch die  $n^2$  Skalarprodukte  $\mathbf{v}_j^\dagger \mathbf{v}_k$  festgelegt sind, vorausgesetzt die  $\mathbf{v}_k$  sind linear unabhängig. 'Relativ' soll hier bedeuten, dass die Vektoren nur bis auf unitäre Transformationen festgelegt sind. Formulieren wir das nun genauer in folgendem

**Satz 1 (Eindeutigkeit der Faktorisierung).** *Ist  $\mathbf{C}$  eine reguläre Matrix, dann hat  $\mathbf{C}$  eine, bis auf unitäre Transformation, eindeutige Wurzel  $\mathbf{B}$ , so dass  $\mathbf{C} = \mathbf{B}^\dagger \mathbf{B}$ .*

*Beweis.* Die Existenz ergibt sich sofort aus der Eigenwertzerlegung von  $\mathbf{C}$  (siehe [34]). Um die Eindeutigkeit zu zeigen, nehmen wir an, es gibt eine zweite Matrix  $\mathbf{G}$ , so dass  $\mathbf{B}^\dagger \mathbf{B} = \mathbf{G}^\dagger \mathbf{G}$ . Da  $\mathbf{G}$  regulär und somit invertierbar, gilt

$$\underbrace{(\mathbf{G}^\dagger)^{-1} \mathbf{B}^\dagger \mathbf{B}}_{\mathbf{U}} = \mathbf{G}.$$

Die Matrix  $\mathbf{U}$  ist unitär, da

$$\begin{aligned} \mathbf{U}^\dagger \mathbf{U} &= ((\mathbf{G}^\dagger)^{-1} \mathbf{B}^\dagger)^\dagger (\mathbf{G}^\dagger)^{-1} \mathbf{B}^\dagger = \mathbf{B} (\mathbf{G})^{-1} (\mathbf{G}^\dagger)^{-1} \mathbf{B}^\dagger \\ &= \mathbf{B} (\mathbf{G}^\dagger \mathbf{G})^{-1} \mathbf{B}^\dagger = \mathbf{B} (\mathbf{B}^\dagger \mathbf{B})^{-1} \mathbf{B}^\dagger = \mathbf{I}. \end{aligned}$$

Also existiert ein unitäres  $\mathbf{U}$ , so dass  $\mathbf{U} \mathbf{B} = \mathbf{G}$ . □

Die Spalten der Matrix  $\mathbf{B}$  sind mit den Vektoren  $\mathbf{v}_k$  zu identifizieren und die Matrixelemente von  $\mathbf{C}$  mit den Produkten  $\mathbf{v}_j^\dagger \mathbf{v}_k$ . Setzt man also voraus, dass  $\mathbf{H}$  einen Satz von linear unabhängigen Vektoren  $\mathbf{v}_k$  erzeugt, dann sind sie aus den Produkten  $\mathbf{v}_j^\dagger \mathbf{v}_k$  bis auf gemeinsame unitäre Transformationen zurückgewinnbar. Da man die Form von  $\mathbf{v}_0$  kennt, kann man die Menge der unitären Transformationen bis auf diejenige eingrenzen, die  $\mathbf{v}_0$  fest lassen. Im nächsten Abschnitt soll hierauf noch genauer eingegangen werden.

Woran aber kann man erkennen, dass  $\mathbf{H}$  linear unabhängige Vektoren erzeugt? Zum einen muss  $\mathbf{v}_0$  an jedem Eigenvektor  $\mathbf{h}_i$  von  $\mathbf{H}$  einen Anteil haben, es muss also für alle  $\mathbf{h}_i$  gelten,  $\mathbf{h}_i^\dagger \mathbf{v}_0 \neq 0$ . Weiter müssen die Eigenwerte  $h_i$  zu den Vektoren  $\mathbf{h}_i$  paarweise verschieden sein, sonst könnte es passieren, dass der Anteil von  $\mathbf{v}_0$  am Eigenraum der mehrfachen Eigenwerte auf dessen Eigenraumdiagonalen gefangen ist. Schließlich muss der Rang von  $\mathbf{H}$  möglichst hoch sein, da die Vektoren im Kern von  $\mathbf{H}$  nicht erreichbar sind. Tatsächlich muss der Rang größer gleich  $n - 1$  sein, wie der folgende Satz zeigt:

**Satz 2 (Lineare Unabhängigkeit).** *Seien  $\mathbf{h}_j$  die Eigenvektoren zu einer Matrix  $\mathbf{H}$ , also  $\mathbf{H}\mathbf{h}_j = h_j\mathbf{h}_j$ . Falls die Eigenwerte  $\{h_i\}$  paarweise verschieden und  $\mathbf{v}_0 = \sum \alpha_j \mathbf{h}_j$  mit  $\alpha_j \neq 0$ , dann sind die Vektoren*

$$\mathbf{v}_k = \mathbf{H}^k \mathbf{v}_0 \quad k \in \{0, \dots, n-1\}$$

*linear unabhängig.*

*Beweis.* Wir betrachten die Determinante, die durch die Vektoren  $\mathbf{v}_k$  gebildet wird. In der Eigendarstellung von  $\mathbf{H}$  lassen sich die  $\mathbf{v}_k$  wie folgt schreiben

$$\mathbf{v}_k = \mathbf{H}^k \mathbf{v}_0 = \sum_{j=0}^{n-1} \alpha_j \mathbf{H}^k \mathbf{h}_j = \sum_{j=0}^{n-1} \alpha_j h_j^k \mathbf{h}_j.$$

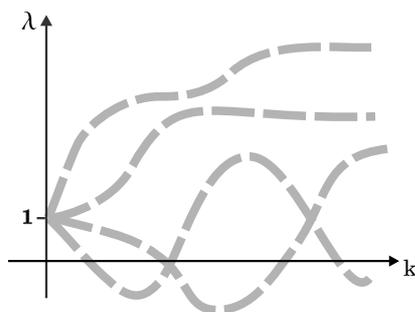
Für die Determinante ergibt sich

$$\begin{aligned} \det(\mathbf{v}_0, \dots, \mathbf{v}_{n-1}) &= \left( \prod_{j=0}^{n-1} \alpha_j \right) \cdot \det \underbrace{\begin{bmatrix} \begin{pmatrix} 1 & h_0 & \dots & h_0^{n-1} \\ 1 & h_1 & \dots & h_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & h_{n-1} & \dots & h_{n-1}^{n-1} \end{pmatrix} \\ \text{Vandermonde-Determinante} \end{bmatrix}} \\ &= \left( \prod_{j=0}^{n-1} \alpha_j \right) \cdot \left( \prod_{1 \leq j < l \leq n-1} (h_l - h_j) \right). \end{aligned}$$

Zur Berechnung der *Vandermonde*-Determinante siehe [20]. Bei den gegebenen Voraussetzungen ergibt sich, dass die Determinante ungleich null ist, also sind die  $\mathbf{v}_k$  linear unabhängig.  $\square$

Sind also die Vorbedingungen des obigen Satzes erfüllt, ist die Matrix  $\mathbf{B}$  regulär und somit auch die Merkmalsmatrix  $\mathbf{C}$ . Es können also die  $n$  Vektoren  $\mathbf{v}_k$  bis auf unitäre Transformationen aus den  $n^2$  Skalarprodukten  $\mathbf{v}_k^\dagger \mathbf{v}_i$  unter den gegebenen Voraussetzungen zurückgewonnen werden.

Betrachten wir nun noch kurz das Histogrammverfahren aus Abschnitt 3.4.2. Falls die Komponenten eines Vektors paarweise verschieden sind, ist es natürlich möglich, die Komponenten des Vektors aus einem Histogramm bis auf seine Reihenfolge zurückzugewinnen. Jedoch bestehen zwischen unseren Vektoren  $\mathbf{v}_k$  Beziehungen, die verloren gehen würden, wenn punktweise, für jeden Vektor einzeln, die Histogrammbildung angewendet werden würde. Es muss also irgendwie versucht werden, die Reihenfolge vom  $k$ -ten Vektor auf den  $k + 1$ -Vektor herüber zu retten. Wenn man sich  $k$  als kontinuierlich vorstellt und  $\mathbf{v}_k$  als stetige Funktion, abhängig von  $k$ , dann kann man sich das Histogramm als übereinander gelegte Funktionsgraphen der einzelnen Komponenten  $(\mathbf{v}_k)_i$  vorstellen:



Man sieht, wie jeder Zweig in 1 beginnt. Dies passiert, wenn  $\mathbf{v}_0 = (1, \dots, 1)^T$  als Startvektor gewählt wird. Geht man davon aus, die Zweige bzw. Funktionen seien genügend glatt und schneiden sich nur in einzelnen Punkten, so sollte es kein Problem darstellen, jede einzelne Komponente zu verfolgen und sie rekonstruieren zu können. Somit haben wir ein Verfahren, mit welchem wir die Vektoren  $\mathbf{v}_k$  zurückgewinnen können.

### 3.5.2 Von den Vektoren zur Kodierung

Nimmt man nun an, die  $n$  Vektoren  $\mathbf{v}_0, \mathbf{v}'_1, \dots, \mathbf{v}'_{n-1}$  seien bekannt, wobei diese mit den Originalvektoren über eine unitäre Transformation  $\tilde{\mathbf{U}} \in U_{\mathbf{v}_0}(n)$ , die  $\mathbf{v}_0$  fest lässt, zusammenhängen  $\tilde{\mathbf{U}}\mathbf{v}_k = \mathbf{v}'_k$ , dann bilden die  $\mathbf{v}'_k$  für die Matrix  $\mathbf{H}' = \tilde{\mathbf{U}}\mathbf{H}\tilde{\mathbf{U}}^\dagger$  ein Gleichungssystem. Setzen wir, wie im Abschnitt zuvor, voraus, die Vektoren  $\mathbf{v}_k$  seien

linear unabhängig, dann werden durch die  $n - 1$  Gleichungen

$$\begin{aligned} \mathbf{v}'_1 &= \mathbf{H}'\mathbf{v}'_0 \\ &\vdots \\ \mathbf{v}'_{n-1} &= \mathbf{H}'\mathbf{v}'_{n-2} \end{aligned}$$

genau  $n - 1$  Bilder der Abbildung  $\mathbf{H}'$  festgelegt. Eine allgemeine  $n \times n$ -Matrix ist jedoch durch  $n - 1$  Bilder leider noch nicht festgelegt.

Nimmt man an,  $\mathbf{H}'$  sei eine reelle, antisymmetrische Matrix ungerader Dimension  $n$ , so weiß man [20], dass  $\text{rang } \mathbf{H}' = (n - 1)$  ( $\mathbf{H}'$  erfüllt trotzdem die Voraussetzungen von Satz 2). Also ist das Bild von  $\mathbf{H}'$   $(n - 1)$ -dimensional und ist somit durch die obigen Gleichungen festgelegt (falls kein Vektor  $\mathbf{v}_k$  im Kern von  $\mathbf{H}$  liegt).

Für andere Fälle muss entsprechend auf die spezielle Struktur der Kodierungen eingegangen werden, um zu sichern, dass  $(n - 1)$  Bilder die Kodierung festlegen.

### 3.5.3 Von der Kodierung zum Gitter

Hat man die Matrix  $\mathbf{H}'$  aus den Invarianten zurückgewonnen, so muss man aus dieser Matrix das Gitter zurückgewinnen. Wir haben gesehen, dass  $\mathbf{H}'$  mit der ursprünglichen Kodierung  $\mathbf{H}(m)$  über eine unitäre Transformation  $\tilde{\mathbf{U}} \in U_{\mathbf{v}_0}(n)$ , die den Startvektor fest lässt, zusammenhängt

$$\tilde{\mathbf{U}}^\dagger \mathbf{H}(m) \tilde{\mathbf{U}} = \mathbf{H}'.$$

Wir haben jedoch kein Wissen über die Transformation  $\tilde{\mathbf{U}}$ . Dennoch wissen wir, wie  $\mathbf{H}(m)$  nach einem gewissen Schema, durch die Kodierung vorgegeben, konstruiert wurde. Es muss also eine unitäre Transformation gesucht werden, zu der ein Gitter  $m$  existiert, so dass obige Gleichung erfüllt ist. Jedoch darf es der Vollständigkeit wegen kein  $\tilde{\mathbf{U}}$  geben, welches die Kodierungen zweier unterschiedlicher, nicht ähnlicher Gitter ineinander überführt, formal

$$\tilde{\mathbf{U}}^\dagger \mathbf{H}(m) \tilde{\mathbf{U}} = \mathbf{H}(m') \Rightarrow m \sim m'. \quad (3.15)$$

Kodierungen, die (3.15) erfüllen, nennen wir im weiteren *eindeutig* bezüglich  $U_{\mathbf{v}_0}(n)$ . Die Eindeutigkeit ist also immer im Bezug auf eine Teilmenge von unitären Transformationen zu verstehen. Jetzt ist auch zu verstehen, warum diese Teilmenge möglichst klein sein sollte. Umso kleiner die Teilmenge, desto größer ist die Chance, dass die von der Kodierung erzeugten Matrizen eine Art Normalform unter diesen Transformationen bilden. Für die Menge der durch eine bestimmte Kodierung erzeugten Matrizen gilt dies genauso. Umso eingeschränkter die Menge der erzeugbaren Matrizen ist, desto größer ist die Chance, dass diese eine Normalform bilden. Es ist also nicht zu erwarten, dass

es  $n \times n$ -Kodierungen gibt, die für  $n$ -knotige Gitter mit beliebigen Verbindungsgrad eindeutig sind.

Besitzen wir also eine eindeutige und vollständige Kodierung, so sollte es möglich sein, das Gitter aus den Invarianten zu rekonstruieren, vorausgesetzt die Vektoren  $\mathbf{v}_k$  sind linear unabhängig und somit zurückgewinnbar.

## 3.6 Ein einfaches Beispiel

Im folgenden soll das wohl einfachste Szenario, welches in den gerade eingeführten Rahmen passt, vorgestellt werden. Das Ziel ist es, einen Vektor  $\mathbf{x} \in \mathbb{R}^n$  permutationsinvariant zu repräsentieren. Als Kodierung des Vektors  $\mathbf{x}$  wird gerade seine zugehörige Diagonalmatrix gewählt, also

$$\mathbf{H}(\mathbf{x}) = \mathbf{Diag}(\mathbf{x}).$$

Die Merkmale  $I_k$ , mit dem  $\mathbf{1}$ -Vektor als Startvektor, ergeben sich zu

$$I_k = \mathbf{1}^\dagger \mathbf{v}_k = \mathbf{1}^\dagger \mathbf{H}^k \mathbf{1} = \sum_{i=0}^n x_i^k \quad \text{mit } k = 1 \dots (2n - 1).$$

Setzen wir voraus, alle Einträge des Vektors  $\mathbf{x}$  seien paarweise ungleich, dann sind diese Merkmale sogar vollständig. Um dies zu zeigen, müssen die folgenden drei Punkte betrachtet werden:

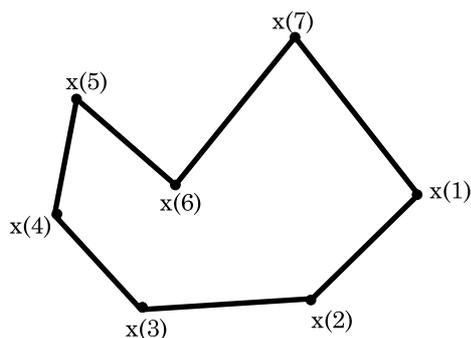
1. **Die Merkmalsmatrix  $\mathbf{C}$  ist regulär (Abschnitt 3.5.1).** Die Merkmalsmatrix  $c_{ij} = \mathbf{v}_i^\dagger \mathbf{v}_j$  ist regulär, wenn die Vektoren  $\mathbf{v}_k$  mit  $k = 0 \dots (n - 1)$  linear unabhängig sind. Die lineare Unabhängigkeit für die oben definierten Vektoren folgt sofort aus Satz 2.
2. **Die Kodierung ist durch das Bild des Vektors  $\mathbf{1}$  bestimmt (Abschnitt 3.5.2).** Es gilt gerade  $\mathbf{x} = \mathbf{H}(\mathbf{x})\mathbf{1}$ .
3. **Die Kodierung ist eindeutig (Abschnitt 3.5.3).** Die reelle Diagonalform einer hermiteschen Matrix ist bis auf Permutation eindeutig bestimmt.

Im letzten Punkt wird kein Gebrauch davon gemacht, dass die Eindeutigkeit nur in Bezug auf unitäre Transformationen, die den  $\mathbf{1}$ -Vektor fest lassen, zu zeigen war. Diese Tatsache ist jedoch von Bedeutung, wenn man obige Merkmale auf komplexe Vektoren  $\mathbf{x}$  erweitern will.

# Kapitel 4

## Stückweise lineare Polygonzüge in der Ebene

In diesem Kapitel wollen wir uns mit ganz speziellen Gittern im 2D beschäftigen. Stückweise lineare, geschlossene Polygonzüge sind durch eine endliche, zyklisch geordnete Menge von Punkten  $\vec{x}_i \in \mathbb{R}^2$  gegeben. Durch die Ordnung ist die Kantenrelation  $E$  des Gitters vorgegeben.



Zuerst werden einige allgemeine Betrachtungen gemacht. Es wird sich herausstellen, dass die translationsinvariante Darstellung von Vektoren von zentraler Bedeutung ist. Es werden zwei schon bestehende Ansätze für die Gewinnung translationsinvarianter Merkmale vorgestellt (CT-Transformation und diskrete Fouriertransformation). Danach werden, mittels den in Kapitel 3 eingeführten Methoden, lageinvariante Merkmale vorgestellt und es werden diese bezüglich Vollständigkeit und dem Verhalten unter Störungen untersucht. Die Merkmale werden durchgehend durch Mittelung gewonnen (siehe Abschnitt 3.4.1).

### 4.1 Allgemeine Betrachtungen

Die euklidische Bewegung  $\vec{x}' = R\vec{x} + \vec{a}$  setzt sich aus einer Dreh-(Spiegelung)  $R$  und einer Verschiebung  $\vec{a}$  zusammen. Nun ist es ein leichtes, durch Differenzenbildung zwi-

schen den Nachbarknoten eine verschiebungsinvariante Darstellung des Polygonzuges zu finden, bei der keine Information verloren geht. Es würden sogar  $n - 1$  Differenzvektoren ausreichen, um den Polygonzug vollständig zu beschreiben, genau die zwei Freiheitsgrade weniger, die durch die Redundanz bzgl. Translation der ursprünglichen Darstellung dazugekommen sind.

Die Darstellung hängt nun immer noch von Drehung und Indizierung der Knoten ab. Eine Drehung ist durch die Erhaltung der Winkel bzw. der Abstände charakterisiert. Es liegt also nahe, diese als Repräsentanten eines Polygonzuges zu benutzen. Später werden wir eine Darstellung zu sehen bekommen, die nur auf den Abständen  $d_{ij} = \|\vec{x}_i - \vec{x}_j\|$  basieren wird. Es wird sich zeigen, dass die Abstände zwischen nächsten, über- und über-über-nächsten Nachbarn

$$( d_{i(i+1)}, d_{i(i+2)}, d_{i(i+3)} ) \quad (4.1)$$

ausreichen werden, um einen Polygonzug zu beschreiben. Jedoch muss man sich für Vollständigkeit auf Polygonzüge beschränken, die keine drei benachbarte, kollineare Punkte besitzen, das heißt es darf kein Knoten  $i$  und ein Faktor  $\alpha$  existieren mit  $\vec{x}_i - \vec{x}_{i-1} = \alpha(\vec{x}_{i+1} - \vec{x}_i)$ . Eine weitere Möglichkeit besteht darin, sich pro Knoten nur die Abstände  $d_{i(i+1)}, d_{i(i+2)}$  und zusätzlich eine Vorzeicheninformation  $w_i$  zu merken, die die Orientierung der Knoten  $x_i, x_{i+1}, x_{i-1}$  beschreibt. Jedoch wird sich diese Darstellung als unstetig bei stetigen Deformationen des Polygonzuges erweisen.

Es stehen zusätzlich noch die Winkel für die Darstellung zur Verfügung. Ein Polygonzug ist zum Beispiel durch die Tupel  $(d_{i(i+1)}, (\varphi_i - \varphi_{i+1}))$  lageinvariant beschrieben, wobei  $\varphi_i$  den Winkel zwischen der Kante  $(i, i + 1)$  und der x-Achse angibt. Es gibt sicher noch andere Möglichkeiten, mit Kombinationen von Abständen und Winkeln Polygonzüge zu beschreiben, jedoch ist bei der Verwendung von Winkeln immer zu beachten, dass ein gewisser Umlaufsinn des Polygonzuges vorausgesetzt ist. Die folgende Darstellung  $\{z_i\} = \mathbf{z} \in \mathbb{C}^n$  eines Polygonzuges wäre möglich:

$$z_i := \left(1 + \frac{d_{i(i+1)}}{d_{(i-1)i}}\right) e^{i(\varphi_i - \varphi_{i-1})}. \quad (4.2)$$

Diese Darstellung ist insbesondere skalierungsinvariant, da nur Abstandsverhältnisse einfließen.

Betrachtet man solche vektorartigen Darstellungen, bei der die Komponente des Vektors mit einer Kante bzw. einem Knoten des Polygonzuges korrespondiert, dann hängen diese Darstellungen natürlich immer noch von der Indizierung ab. Eine Umin-dizierung wirkt sich auf die Vektorrepräsentation durch zyklische Translation aus. Wir suchen also eine Darstellung der Vektoren, die invariant gegenüber einer zyklischen

Translation  $\mathbf{P}_\tau \in T(n)$  eines Vektors ist. Im folgenden sollen nun zwei in der Bildverarbeitung bekannte Methoden vorgestellt werden, die translationsinvariante Merkmale für Vektoren liefern.

### 4.1.1 CT-Transformation

Die CT-Transformation [35] basiert auf einem Rekursionsschema. Gehen wir davon aus, unser Objekt sei durch einen  $2^N = n$  dimensionalen Vektor  $\mathbf{x} \in \mathbb{K}^n$  repräsentiert. Die Transformation  $T_n$  bildet den Vektor  $\mathbf{x} \in \mathbb{K}^n$  auf einen Vektor  $\tilde{\mathbf{x}} \in \mathbb{K}^n$  ab, der invariant unter zyklischer Translation des Vektors  $\mathbf{x}$  ist. Die Transformation ist wie folgt rekursiv definiert:

$$\begin{aligned}
 T_1 &= 1 \\
 T_{2n}(\mathbf{x}) &= (T_n(f(\mathbf{x}^{(l)}, \mathbf{x}^{(h)})), T_n(g(\mathbf{x}^{(l)}, \mathbf{x}^{(h)}))) \quad (4.3) \\
 &\text{wobei } \mathbf{x} = (\mathbf{x}^{(l)} \mathbf{x}^{(h)}),
 \end{aligned}$$

hierbei seien  $f, g : (\mathbb{K} \times \mathbb{K}) \mapsto \mathbb{K}$  zwei symmetrische Funktionen, das heißt  $f(a, b) = f(b, a)$  und  $g(a, b) = g(b, a)$ . Der Vektor  $\mathbf{x}^{(l)}$  bezeichnet die  $n$  unteren Komponenten  $x_1, \dots, x_n$  des Vektors  $\mathbf{x} \in \mathbb{K}^{2n}$ , und  $\mathbf{x}^{(h)}$  die oberen Komponenten  $x_{n+1}, \dots, x_{2n}$ . Die Anwendung der Funktionen  $f$  und  $g$  erfolgt punktweise auf die  $n$ -komponentigen Vektoren  $\mathbf{x}^{(l)}$  und  $\mathbf{x}^{(h)}$ . In Abbildung 4.1 wird die Rekursion für einen 8 dimensionalen Vektor veranschaulicht. Durch die rekursive Definition lässt sich leicht der Grund für

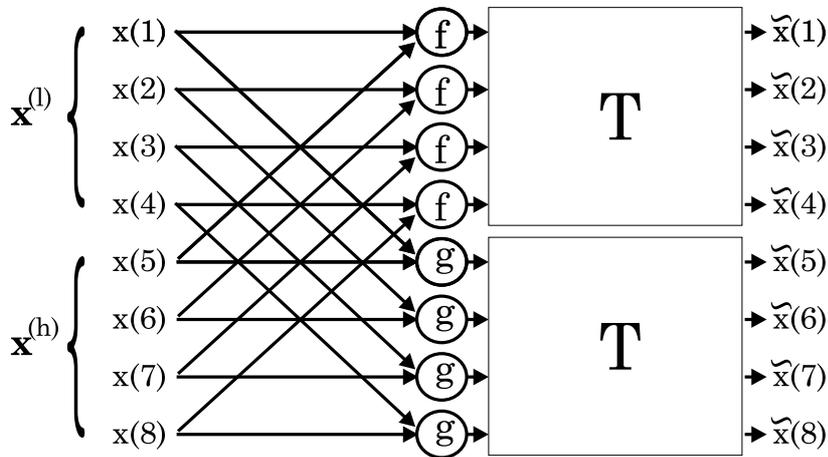


Abbildung 4.1: **Rekursive Definition der Transformation.** Schematisch

die Invarianz der Transformation verstehen. Geht man davon aus, die Transformation  $T_n$  sei zyklisch invariant, dann kann man mittels der Definition (4.3) eine zyklisch invariante Transformation  $T_{2n}$  konstruieren. Dies kann man daran erkennen, dass sich eine zyklische Translation des Vektors  $\mathbf{x} \in \mathbb{K}^{2n}$  ebenfalls als zyklische Translation auf

den Vektor  $f(\mathbf{x}^{(h)}, \mathbf{x}^{(l)}) \in \mathbb{K}^n$  auswirkt, vorausgesetzt man kann die Argumente von  $f$  vertauschen (in Abbildung 4.2 ist dies beispielhaft für einen 8er Vektor abgebildet). Das heißt  $T_n(f(\mathbf{x}^{(h)}, \mathbf{x}^{(l)}))$  ist invariant unter zyklischer Translation des Vektors  $\mathbf{x}$ . Dies gilt genauso für den zweiten Teil  $T_n(g(\mathbf{x}^{(h)}, \mathbf{x}^{(l)}))$  und somit ist die Transformation  $T_{2n}$  invariant.

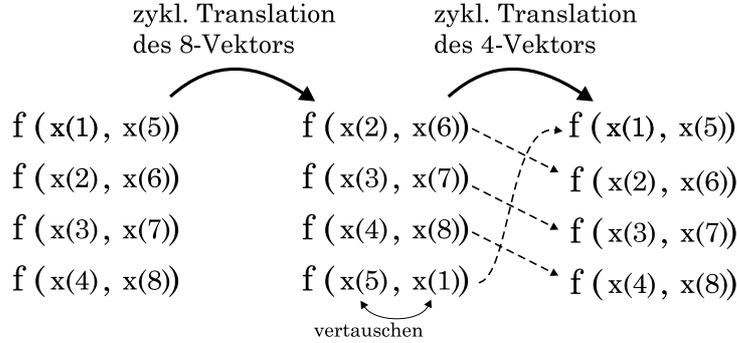


Abbildung 4.2: **Invarianz der CT.** Die zyklische Translation eines 8 komponentigen Vektors  $\mathbf{x}$  führt zu einer zyklischen Translation des 4 komponentigen Vektors  $f(\mathbf{x}^{(h)}, \mathbf{x}^{(l)})$ .

Für die Wahl der Funktionen  $f$  und  $g$  existieren eine Reihe von Möglichkeiten [35]. Wendet man zum Beispiel die CT-Transformation auf die Polygonrepräsentation (4.2) an, dann wäre etwa

$$f(z, z') = |z + z'|^2 \text{ und } g(z, z') = |z - z'|^2$$

eine Wahl. Für die Distanzrepräsentation (4.1) ist es schon schwieriger: jede Vektor-komponente besteht aus 3 reellen, positiven Zahlen  $\vec{q}_i = (d_{i(i+1)}, d_{i(i+2)}, d_{i(i+3)})$ . Die Funktionen  $f, g$  sind also vom Typ  $(\mathbb{R}^3 \times \mathbb{R}^3) \mapsto \mathbb{R}^3$ . Natürlich wäre es eine einfache Möglichkeit, die Funktionen von oben punktweise auf die 3er Vektoren anzuwenden, jedoch gehen so die Zusammenhänge zwischen den Abständen verloren, das heißt der Vektor  $\{d_{i(i+1)}\}$  transformiert unabhängig von den zwei anderen und kann somit frei verschoben werden, ohne die Merkmale zu ändern, was der Separationsfähigkeit nicht unbedingt zuträglich ist. Das heißt wir benötigen ein symmetrisches 3er-Vektorprodukt, wobei die drei Komponenten untereinander auf irgendeine Weise koppeln müssen, z.B.

$$f((d_1, d_2, d_3), (d'_1, d'_2, d'_3)) = \begin{pmatrix} (d_2 - d'_2)^2 + (d_3 - d'_3)^2 \\ (d_3 - d'_3)^2 + (d_1 - d'_1)^2 \\ (d_1 - d'_1)^2 + (d_2 - d'_2)^2 \end{pmatrix}.$$

Eine ähnliche Wahl wäre für  $g$  möglich.

Betrachten wir noch kurz den Aufwand zur Berechnung der Transformation. Es sind  $\mathcal{O}(n \log_2(n))$  Anwendungen der zweistelligen Verknüpfungen  $f, g$  nötig. Dies ist sofort

daran zu erkennen, dass die Rekursionstiefe des Verfahrens  $N = \log_2(n)$  ist und für jeden Rekursionsschritt  $n$  Verknüpfungen berechnet werden müssen. Der Speicherbedarf ist gerade  $\mathcal{O}(n)$ , da jeder Rekursionsschritt nur auf Ergebnisse des letzten Schritts zurückgreifen muss, somit muss nicht für jeden Schritt neuer Speicher alloziert werden und es genügen  $2n$  Speicherzellen vom Typ  $\mathbb{K}$ .

## 4.1.2 Diskrete Fourierdeskriptoren

Die Diskrete Fourier Transformierte  $\tilde{\mathbf{x}} \in \mathbb{C}^n$  eines Vektors  $\mathbf{x} \in \mathbb{C}^n$  ist gegeben durch

$$\tilde{x}_k = \mathbf{e}_k^\dagger \mathbf{x} = \sum_{i=0}^{n-1} x_i e^{-i\frac{2\pi}{n}ik},$$

wobei  $\mathbf{e}_k^\dagger$  die Fourier-Basisvektoren sind. Eine zyklische Translation  $\mathbf{T}_t \in T(n)$  des Vektors  $\mathbf{x}$  um  $t$  Komponenten wirkt sich auf die Transformierte mit einer punktweisen Multiplikation mit einem Phasenfaktor  $e^{i\frac{2\pi}{n}tk}$  aus

$$\mathbf{e}_k^\dagger \mathbf{T}_t \mathbf{x} = \mathbf{e}_k^\dagger \mathbf{x} e^{i\frac{2\pi}{n}tk}.$$

Um Invarianz zu erreichen, muss dieser Faktor durch Normierung zum Verschwinden gebracht werden. Dies kann, ähnlich wie bei den kontinuierlichen Fourierdeskriptoren, wie folgt geschehen:

$$I_k(\mathbf{x}) = \tilde{x}_k e^{-ik(\arg(\tilde{x}_r) - \arg(\tilde{x}_{r-1}))},$$

wobei  $r$  wieder eine für alle Objekte feste ganze Zahl ist. Die Invarianz kann leicht durch Einsetzen<sup>1</sup> überprüft werden. Der Parameter  $r$  sollte wieder so gewählt werden, dass die entsprechenden Fourierkoeffizienten  $\tilde{x}_r, \tilde{x}_{r-1}$  vom Betrag her möglichst groß sind, damit die Winkel numerisch sicher zu berechnen sind. Die Berechnung der Merkmale  $I_k(\mathbf{x})$  kann mittels FFT [31] in  $\mathcal{O}(n \log(n))$  geschehen. Die Merkmale sind sogar vollständig, d.h wir können den Vektor  $\mathbf{x}$  bis auf zyklische Translationen aus den Merkmalen zurückgewinnen. Es liegt nahe, diese Methode auf die Polygonrepräsentation (4.2) anzuwenden.

## 4.2 Kodierungen

Es sollen, wie schon oben erwähnt, zwei verschiedene Kodierungen betrachtet werden. Die Eine wird allein nur auf 3 Distanzen pro Knoten beruhen. Die Zweite auf zwei

<sup>1</sup>Die Invarianz kann durch Einsetzen leicht gezeigt werden

$$\begin{aligned} I_k(\mathbf{T}_t \mathbf{x}) &= \tilde{x}_k e^{i\frac{2\pi}{n}tk} e^{-ik(\arg(\tilde{x}_r) - \arg(\tilde{x}_{r-1}) + \frac{2\pi}{n}(tr - t(r-1)))} \\ &= \tilde{x}_k e^{i\frac{2\pi}{n}tk} e^{-i\frac{2\pi}{n}kt} e^{-ik(\arg(\tilde{x}_r) - \arg(\tilde{x}_{r-1}))} = I_k(\mathbf{x}). \end{aligned}$$

Distanzen pro Knoten und einer Vorzeicheninformation.

Da durch die Nachbarschaftsverhältnisse der Knoten eine zyklische Ordnung auf der Knotenmenge vorgegeben ist, unterscheiden sich verschiedene Repräsentationen des Polygonzuges nur dahingehend, mit welchem Knoten man die Aufzählung beginnt und in welche Richtung gezählt wird. Die Invarianz der Repräsentation erstreckt sich somit, nicht wie im allgemeinen Fall, auf die Permutationsgruppe, sondern auf die Gruppe der zyklischen Translationen und Spiegelungen. In der Forderung (3.1) aus Abschnitt 3.3 kann man sich somit auf diese beiden beschränken. Fassen wir die in Kapitel 3 formulierten Anforderungen an eine Kodierung im speziellen Fall der Polygonzüge zusammen:

1. **Vollständigkeit der Kodierung**

Aus der Matrix  $\mathbf{H}(m)$  muss der Polygonzug  $m$  zurückgewinnbar sein.

2. **Aufpunktinvarianz**

Wie schon oben erwähnt, hat für eine Kodierung  $\mathbf{H}(m)$  eines Polygonzuges  $m$  zu gelten:  $\mathbf{H}(\tau(m)) = \mathbf{P}_\tau^\dagger \mathbf{H}(m) \mathbf{P}_\tau$ , wobei  $\mathbf{P}_\tau \in T(n)$ .

3. **Regularität**

Die Matrix  $\mathbf{H}(m)$  sollte, wie in Abschnitt 3.5.1 erläutert, möglichst regulär sein.

4. **Rückgewinnbarkeit**

Die Matrix  $\mathbf{H}(m)$  muss durch  $(n - 1)$  linear unabhängige Bilder vollständig bestimmt sein.

5. **Eindeutigkeit**

Die Eindeutigkeitsbedingung (3.15) aus Abschnitt 3.5.3 kann für Polygonzüge wie folgt formuliert werden:

$$\tilde{\mathbf{U}}^\dagger \mathbf{H}(m) \tilde{\mathbf{U}} = \mathbf{H}(m') \Rightarrow \tilde{\mathbf{U}} \in T(n).$$

Das heißt, falls eine unitäre Transformation  $\tilde{\mathbf{U}} \in U_{\mathbf{v}_0}(n)$  existiert, die zwei verschiedene Matrizen  $\mathbf{H}(m)$  und  $\mathbf{H}(m')$  ineinander überführt, dann muss diese Transformation eine zyklische Translation oder eine Spiegelung sein.

Mit diesen Bedingungen vor Augen sollen die zwei Kodierungen angegeben werden.

### 4.2.1 Distanzkodierungen

Bezeichnen wir mit  $d_{ij}$  den Abstand zwischen dem  $i$ -ten und  $j$ -ten Knoten  $d_{ij} = \|\vec{x}_i - \vec{x}_j\|$ . Ein Polygonzug ist bis auf euklidische Bewegung und Spiegelung durch die

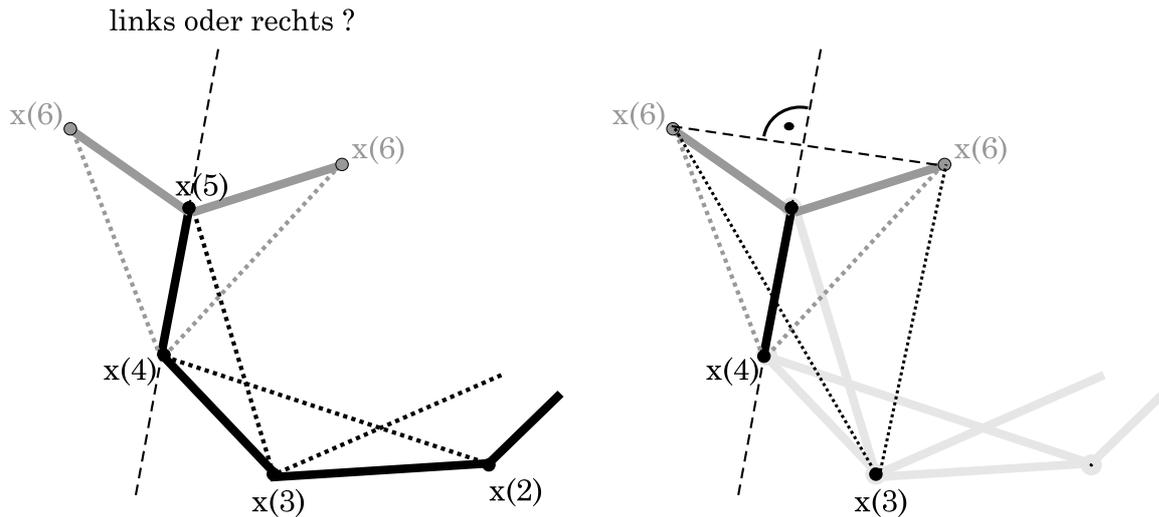


Abbildung 4.3: **Wie ein Polygonzug konstruiert wird.** Angenommen, der Polygonzug ist bis Knoten  $\vec{x}_5$  konstruiert. Durch die Abstände  $d_{56}$  und  $d_{46}$  ist der Knoten  $\vec{x}_6$  bis auf Spiegelung um die Kante  $(\vec{x}_4, \vec{x}_5)$  festgelegt. Setzt man voraus,  $\vec{x}_3$  liege nicht auf der Spiegelungsachse, dann ist  $\vec{x}_6$  durch den Abstand  $d_{36}$  festgelegt, da alle Punkte, die nicht auf der Spiegelachse liegen, unterschiedliche Abstände zu den zwei möglichen neuen Punkten haben.

Abstände  $d_{i(i+1)}$ ,  $d_{i(i+2)}$ ,  $d_{i(i+3)}$  bestimmt. Zur Veranschaulichung stelle man sich vor, die Abstände  $d_{i(i+1)}$ ,  $d_{(i-1)i}$  und  $d_{(i-1)(i+1)}$  bilden ein Dreieck, dann kann man mittels Zusammenstecken dieser Dreiecke einen Polygonzug konstruieren. Für jedes Dreieck muss zusätzlich die Entscheidung getroffen werden, ob der neu entstehende Punkt  $\vec{x}_{i+1}$  links oder rechts von der Kante  $(\vec{x}_i, \vec{x}_{i-1})$  liegt (siehe Abb. 4.3). Dies kann mit dem Abstand zum vorhergehenden Punkt  $\vec{x}_{i-2}$  entschieden werden. Lediglich zu Beginn, wenn kein vorhergehender Punkt existiert, weiß man nicht, ob der neue Punkt  $\vec{x}_3$  links oder rechts von der Spiegelungsachse  $(\vec{x}_1, \vec{x}_2)$  liegt, das heißt der Polygonzug ist nur bis auf Spiegelung zurückgewinnbar. Die zweite Möglichkeit die Entscheidung zu treffen, ob der Punkt  $\vec{x}_{i+1}$  links oder rechts von der Kante  $(\vec{x}_i, \vec{x}_{i-1})$  liegt, besteht darin, sich die Orientierung

$$\omega_i = \text{sign} \left( \det \begin{pmatrix} \vec{x}_{i+1} & \vec{x}_i & \vec{x}_{i-1} \\ 1 & 1 & 1 \end{pmatrix} \right)$$

zu merken. Jedoch führt die Anwendung der Signumfunktion zu Unstetigkeiten, die gerade dann entstehen, wenn es drei benachbarte kollineare Punkte gibt, eben genau dann, wenn obige Determinante zu Null wird.

Hat man nun solch eine Darstellung des Polygonzuges mittels Abständen gegeben, dann ist diese invariant gegenüber der euklidischen Bewegung. Also wird sich eine Lageänderung  $\varepsilon$  durch  $\mathbf{U}_\varepsilon = 1$  auf eine Kodierung, die nur von diesen Abständen abhängt, auswirken. Betrachten wir als erstes die Distanzkodierung, die auf 3 Distanzen

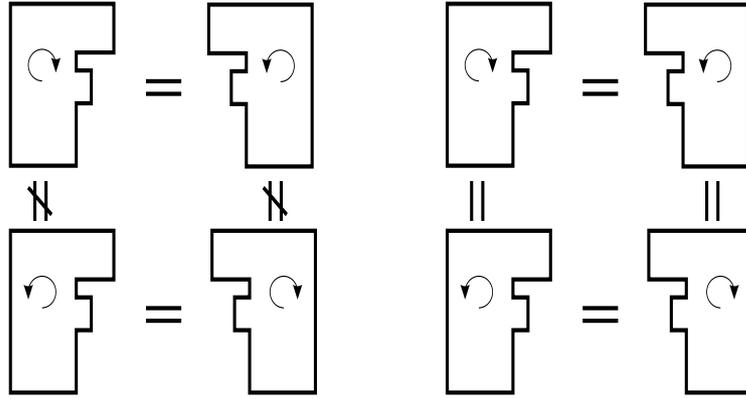


Abbildung 4.4: **Spiegelungsäquivalenz.** Links: hermitesche 3Distanzkodierung. Es wird zwischen verschiedenen Umlaufrichtungen unterschieden. Rechts: asymmetrische 2Distanzkodierung.

pro Knoten basiert. Sei  $\mathbf{D}_1$  die Matrix, die auf der Diagonalen die Einträge  $d_{i(i+1)}$  besitzt. Entsprechend seien  $\mathbf{D}_2$  und  $\mathbf{D}_3$  die Diagonalmatrizen mit den Einträgen  $d_{i(i+2)}$  bzw.  $d_{i(i+3)}$  auf der Diagonalen. Dann ist die Kodierung wie folgt definiert:

$$\mathbf{H}(m) := i\mathbf{D}_1 + (\mathbf{T}\mathbf{D}_2 - \mathbf{D}_2\mathbf{T}^\dagger) + (\mathbf{T}^2\mathbf{D}_3 - \mathbf{D}_3(\mathbf{T}^\dagger)^2), \quad (4.4)$$

wobei  $\mathbf{T}$  eine zyklische Translationmatrix<sup>2</sup> ist. Die Kodierung ist offensichtlich antihermitesch, sie setzt sich aus einem imaginären Diagonalteil und einem antisymmetrischen, reellen Teil zusammen. Die Merkmalsmatrix  $\mathbf{C}(m)$  (siehe (3.7)) ist also durch  $2n - 1$  komplexe Zahlen vollständig beschrieben. Wir nennen diese Kodierung im folgenden 3Distanzkodierung. Durch den antisymmetrischen, reellen Teil wird eine Umlaufrichtung des Polygonzuges induziert, das heißt die Kodierung 'erkennt' verschiedene Umlaufrichtungen des Polygonzuges (siehe Abbildung 4.4).

Als zweites führen wir die 2Distanzkodierung ein. Sie basiert, wie schon oben erklärt, auf nur 2 Distanzen und einem Vorzeichen pro Knoten. Die Matrix  $\mathbf{W}$  sei die Diagonalmatrix mit den Vorzeichen  $\omega_i$  auf der Diagonalen. Die 2Distanzkodierung sei dann wie folgt gegeben:

$$\mathbf{H}(m) := i\mathbf{W}\mathbf{D}_1 + (\mathbf{T}\mathbf{D}_2 + \mathbf{D}_2\mathbf{T}^\dagger). \quad (4.5)$$

Offenbar ist sowohl die 2Distanzkodierung als auch 3Distanzkodierung vollständig, das heißt die korrespondierenden Polygonzüge sind aus der Kodierung rekonstruierbar. Tatsächlich wird sich später zeigen, dass die 2Distanzkodierung unter den in Kapitel 3 erläuterten Bedingungen zu vollständigen Merkmalsmatrizen  $\mathbf{C}(m)$  führen wird. Jedoch ist sie weder (anti-)hermitesch noch unitär und somit hängen die Einträge von

<sup>2</sup>Die Matrixelemente von  $\mathbf{T}$  sind durch  $t_{ij} = \delta_{i(j+1)}$  gegeben.

$\mathbf{C}(m)$  nicht auf einfache Weise voneinander ab, was zu einem  $n^2$  dimensionalen Merkmal führt. Dies ist natürlich nicht wünschenswert, da man schon in  $\mathcal{O}(n^2)$  zwei Vektoren auf Gleichheit unter zyklischer Translation überprüfen kann.

### 4.3 Experimente

Im folgenden werden einige praktische Versuche zu der eben vorgestellten Distanzkodierung (4.4) vorgenommen. Wir beschränken uns dabei auf die 3Distanzkodierung, da sie dank ihrer Stetigkeit von eher praktischer Bedeutung ist. Die Merkmale  $I_k$  werden durch Mittelung gewonnen (siehe Abschnitt 3.4.1), wobei  $k$  wieder  $2n - 1$  Werte annimmt. Für antisymmetrische Kodierungen reichen diese bekanntlich aus, um die komplette Merkmalsmatrix  $\mathbf{C}(m)$  aufzustellen. Neben der reinen 3Distanzkodierung betrachten wir auch die, durch die Exponentialabbildung induzierte, unitäre 3Distanzkodierung (siehe Abschnitt 3.4.3). Die unsymmetrische 2Distanzkodierung (4.5) wird nicht betrachtet.

Name	Kodierung	It.vorschrift	Merkmale
3Distanz	(4.4)	(3.14)	$\mathbf{v}_0^\dagger \mathbf{H}^k \mathbf{v}_0 / \ \mathbf{H}^k \mathbf{v}_0\ $
unitäre 3Distanz	(4.4)	(3.13)	$\mathbf{v}_0^\dagger \exp(\sigma_n \mathbf{H}t) \mathbf{v}_0$

Für die unitären Merkmale bietet es sich an, Iterationsvorschrift (3.13) zu verwenden, da sie auch ohne Normierung in einem vernünftigen Zahlenbereich bleiben. Der Parameter  $\sigma_n$  wurde durch Ausprobieren<sup>3</sup> wie folgt gewählt:

$$\sigma_n = \frac{n}{n + t_0}, \quad (4.6)$$

wobei die Merkmalsfunktion zwischen 0 und  $t_0$  ausgewertet wird. Bei den reinen 3Distanzmerkmalen wird Iterationsvorschrift (3.14) eingesetzt. Es werden nur die Winkelanteile  $I'_k$  (siehe Abschnitt 3.4.4) als Merkmal benutzt. Die Winkelanteile  $I'_k$  sind insbesondere invariant unter Skalierungen<sup>4</sup> des Polygonzuges.

Als Testobjekte wollen wir zu Beginn sechs 4-knotige Polygonzüge benutzen, wobei jedes Objekt zweimal in leicht unterschiedlichen Versionen im Datensatz vorkommt. Die Punkte der Objekte liegen jeweils im Gebiet  $[-1, 1] \times [-1, 1]$ . In Abbildung 4.5

<sup>3</sup>Es ist zu beobachten, dass bei konstantem  $\sigma_n = \sigma$  die Bogenlänge  $L$  der Merkmalsfunktion mit wachsender Knotenzahl abnimmt. Also sollte  $\sigma_n$  klein für kleines  $n$  werden, damit die Merkmalsfunktion langsamer durchlaufen wird.

<sup>4</sup>Eine Skalierung des Polygonzuges mit dem Faktor  $\alpha$  wirkt sich auf die Kodierung durch Multiplikation mit  $\alpha$  aus

$$I'_k(\alpha m) = \mathbf{v}_0^\dagger (\alpha \mathbf{H})^k \mathbf{v}_0 / \|(\alpha \mathbf{H})^k \mathbf{v}_0\| = \mathbf{v}_0^\dagger \mathbf{H}^k \mathbf{v}_0 / \|\mathbf{H}^k \mathbf{v}_0\| = I'_k(m).$$

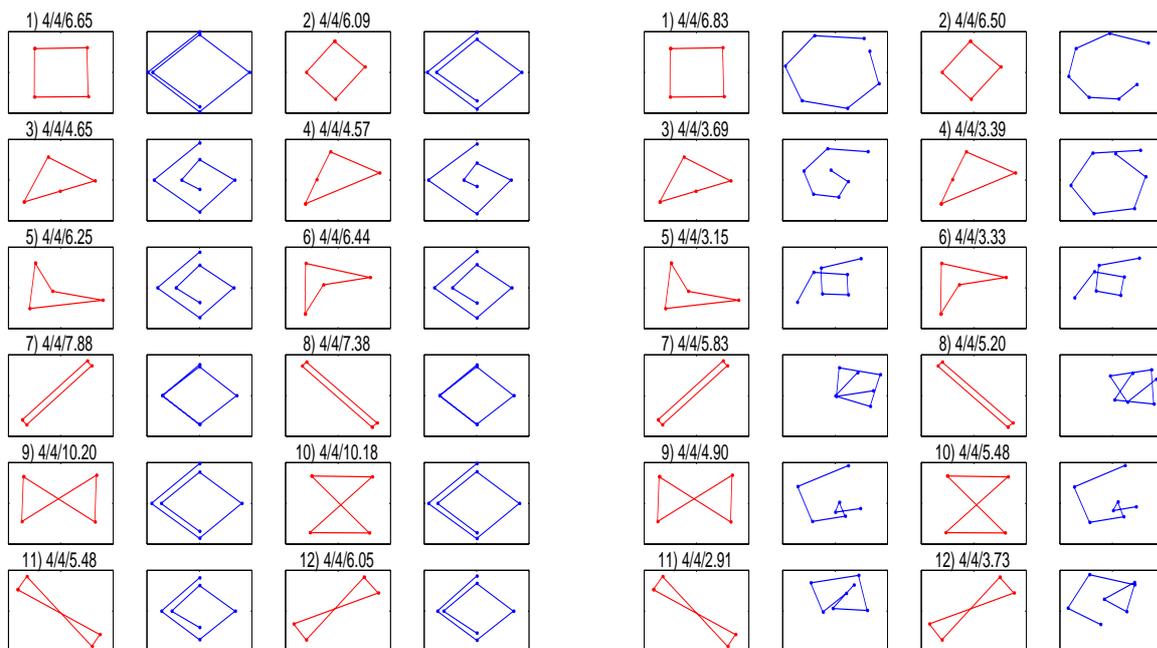


Abbildung 4.5: **4-knotige Polygonzüge 3Distanzkodierung.** Das Merkmal wird in der komplexen Ebene dargestellt. Links: 3Distanzkodierung (4.4) mit Iterationsvorschrift (3.14). Rechts: unitäre 3Distanzkodierung mit Iterationsvorschrift (3.13). Die Achsenbereiche der Merkmale sind alle identisch gewählt.

sind die Objekte rot dargestellt (unter der Beschriftung). Rechts davon, blau, sind jeweils die Merkmale in der komplexen Ebene parametrisch aufgetragen. Die Beschriftung oberhalb des Objekts setzt sich wie folgt zusammen: an erster Stelle steht die Objekt-nummer und danach die Anzahl der Knoten des Objektes. Die dritte und vierte Zahl sind für die Vollständigkeit von Bedeutung. Die Dritte ist der Rang der Merkmalsmatrix  $\mathbf{C}(m)$  und die Vierte der Logarithmus der Konditionierungszahl der Matrix  $\mathbf{C}(m)$ . Die Konditionierungszahl ist das Verhältnis vom betragsmäßig größtem zum kleinsten Eigenwert der Matrix. Erinnern wir uns an die Vollständigkeitsbetrachtungen aus Abschnitt 3.5.1, dann muss die Merkmalsmatrix  $\mathbf{C}(m)$  invertierbar sein. Große Konditionierungszahlen bedeuten also nahezu singuläre, nicht invertierbare Merkmalsmatrizen. Man kann sie als Maß für die Invertierbarkeit des Merkmalsbildungsprozesses ansehen.

Betrachten wir also Abbildung 4.5. Die reine 3Distanzkodierung ordnet jedem Objekt  $2n - 1$  komplexe Zahlen als Merkmal zu. Für die unitäre 3Distanzkodierung wurde die Merkmalsfunktion an den Stellen  $t = 1, 2, \dots, 2n - 1$  ausgewertet. Tatsächlich kann man erkennen, dass die Merkmale auf bestimmte Art charakteristisch für die Form der Objekte sind. Die Merkmalsmatrix  $\mathbf{C}(m)$  ist scheinbar für alle dargestellten Objekte invertierbar, jedoch ist die Konditionierungszahl relativ groß. Im Vergleich haben die

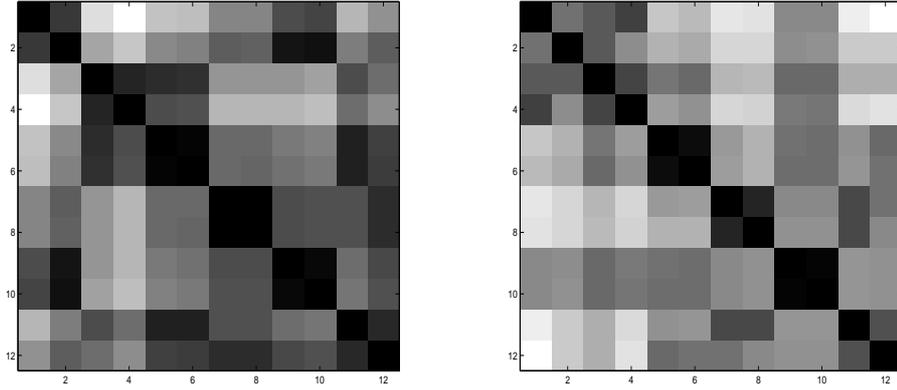


Abbildung 4.6: **Ähnlichkeitsmatrizen.** Links: 3Distanzkodierung. Rechts: unitäre 3Distanzkodierung.

unitären Merkmale leicht geringere Konditionierungszahlen. Eigentlich ist die Merkmalsmatrix eines exakten Quadrats singular, die Invertierbarkeit der Merkmalsmatrix  $C(m)$  von Objekt 1 rührt also nur von den kleinen Störungen her, die es vom perfekten Quadrat unterscheiden.

In Abbildung 4.6 sind Ähnlichkeitsmatrizen zu den 2 verschiedenen Versuchsreihen angegeben. Zum Vergleich wurde der euklidische Abstand genommen

$$D(m, m') = \sqrt{\sum_{k=1}^{2n-1} |I_k(m) - I_k(m')|^2}. \quad (4.7)$$

Die Ähnlichkeitsmatrizen zeigen schön, wie die Distanzen zwischen ähnlichen Objekten klein werden. Es treten trotzdem ein paar Verwechslungen auf. Für die reine 3Distanzkodierung ist das Quadrat 2) den Objekten 9),10) ähnlicher als dem Quadrat 1). Die unitäre 3Distanzkodierung hat Probleme, zwischen den Quadraten 1),2) und den Dreiecken 3),4) zu unterscheiden.

Neben den 4-knotigen Polygonzügen wurde auch mit vielknotigen Objekten experimentiert. Bis jetzt wurden  $2n - 1$  Merkmale einem  $n$ -knotigen Objekt zugeordnet. Um Objekte mit verschiedener Anzahl von Knoten vergleichbar zu machen, werden nun immer eine feste Anzahl  $M$  an Exponentationen durchgeführt, somit werden jedem Objekt  $M$  komplexe Merkmale zugeordnet. In den folgenden Versuchen wird  $M = 50$  gewählt. Für die unitären Merkmale wird die Merkmalsfunktion an den Stützstellen  $t = 1, 2, \dots, 50$  ausgewertet.

Im Abbildung 4.7 sind für einige Beispielobjekte die Merkmale berechnet worden. Für die reine Distanzkodierung ist zu erkennen, dass der komplexe Betrag der Merkmalskomponenten mit zunehmender Knotenzahl immer schneller konvergiert. Dies schränkt die Reichhaltigkeit der Merkmale deutlich ein, das heißt für große  $k$  sind be-

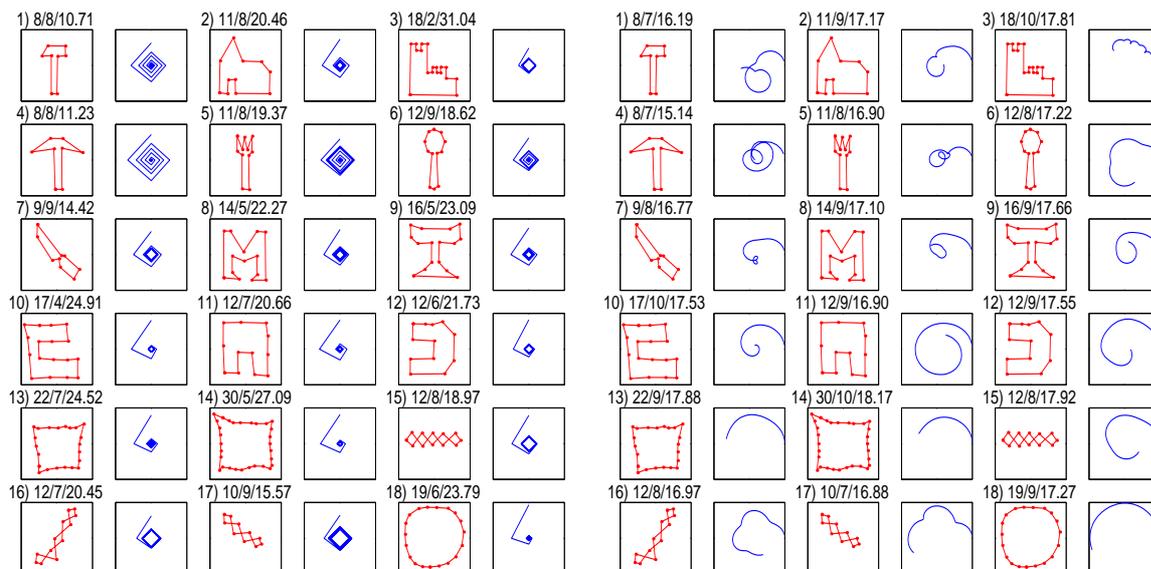


Abbildung 4.7: **n-knotige Polygonzüge 3Distanzkodierung.** Das Merkmal wird in der komplexen Ebene dargestellt. Links: 3Distanzkodierung (4.4) mit Iterationsvorschrift (3.14). Rechts: unitäre 3Distanzkodierung mit Iterationsvorschrift (3.13). Die Achsenbereiche der Merkmale sind alle identisch gewählt.

nachbarte Merkmalskomponenten immer stärker korreliert. Der Grenzwert des Betrages ist in der Regel<sup>5</sup> durch den Betrag des Skalarprodukts des Startvektors  $\mathbf{v}_0$  mit dem Eigenvektor  $\mathbf{e}_\infty$  zum betragsmäßig größten Eigenwert der Kodierung gegeben. Genau so ist die Invertierbarkeit der Merkmalsmatrix für Objekte mit großen Knotenzahlen nicht mehr gegeben. Dies ist auch durch den schellen Konvergenzprozess zu erklären; die Vektoren  $\mathbf{v}_k$  spannen nur einen kleinen Teilraum des  $\mathbb{C}^n$  auf, da die  $\mathbf{v}_k$  sehr schnell gegen  $\mathbf{e}_\infty$  konvergieren und sich ihr Konvergenzpfad nur in einem kleinem Teilraum befindet.

Die unitären zeigen im Gegensatz zu den reinen Merkmalen mehr Vielfalt, jedoch, wenn wir uns an Abschnitt 3.4.3 erinnern, wissen wir, dass sie in gewisser Weise die gleiche Information wie die reine Distanzkodierung beinhalten, es wird also nur ein anderes Vergleichsmaß angelegt.

### 4.3.1 Störungsuntersuchungen

Die vorgestellten Merkmale sollen nun daraufhin untersucht werden, wie sie auf kleine, unabhängige Verschiebung der einzelnen Knoten reagieren. Die ursprünglichen Koor-

<sup>5</sup>Die Merkmalsvektoren  $\mathbf{v}_k = \mathbf{H}^k \mathbf{v}_0 / \mathbf{v}_0^\dagger \mathbf{H}^k \mathbf{v}_0$  konvergieren gegen den Eigenvektor  $\mathbf{e}$ , der den betragsmäßig größten Eigenwert besitzt und gleichzeitig Anteil an dem Startvektor  $\mathbf{v}_0$  hat, also  $\mathbf{v}_0^\dagger \mathbf{e} \neq 0$ . Der Begriff der Konvergenz ist hier nicht ganz sauber benutzt: Die Richtung des Vektors  $\mathbf{v}_k$  konvergiert, die komplexe Phase des Vektors oszilliert.

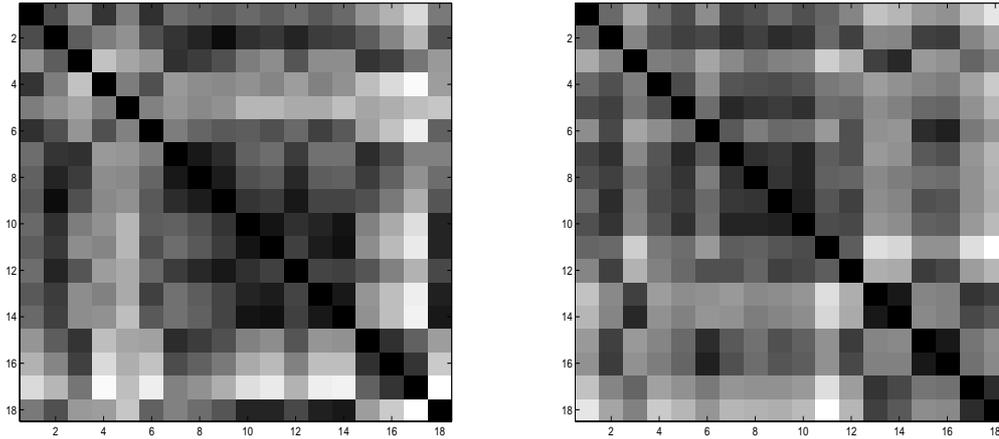


Abbildung 4.8: **Ähnlichkeitsmatrizen.** Links: 3Distanzkodierung. Rechts: unitäre 3Distanzkodierung

koordinaten  $\{\vec{x}_i\}$  werden durch Verschiebungen  $\{\Delta\vec{x}_i\}$  auf die Koordinaten  $\{\vec{x}'_i\}$  abgebildet

$$\vec{x}'_i = \vec{x}_i + \epsilon\Delta\vec{x}_i.$$

Bei den folgenden Versuchen werden Verschiebungen betrachtet, die durch  $n$  unabhängige, normalverteilte, mittelwertfreie Zufallsprozesse mit Standardabweichung  $\sigma = 1$  gewonnen wurden. Der Parameter  $\epsilon$  kann als ein Maß für das Rausch/Signal-Verhältnis interpretiert werden. Die Testobjekte bzw. Signale befinden sich in dem Gebiet  $[-1, 1] \times [-1, 1]$ , und deren Ausdehnungen sind im Bereich 1. Bei einer Störung der Breite  $\sigma = 1$  haben wir also ein Rausch/Signalverhältnis von 1, das Objekt wird bis zur 'Unkenntlichkeit' zerstört. Für die folgenden Untersuchungen wurden Rausch/Signal-Verhältnisse zwischen 0 und 0.2 gewählt.

In Abbildung 4.9 wird die Umgebung eines Quadrats betrachtet. In den beiden oberen Diagrammen sind für verschiedene, zufällig gewählte Störrichtungen  $\Delta m = \{\Delta\vec{x}_i\}$  die Abstände  $D(m, m + \epsilon\Delta m)$  zwischen dem ursprünglichen Quadrat und dem gestörten Quadrat für variierendes  $\epsilon$  aufgetragen. Es ist zu erkennen, dass für die reine 3Distanzkodierung der Abstand für kleine  $\epsilon$  etwas stärker anwächst als bei der unitären Kodierung. Dafür flacht er für größere  $\epsilon$  bei der reinen Kodierung stärker ab, was bei dem linearen Verhalten der unitären Kodierung nicht zu beobachten ist. In den beiden unteren Histogrammen sind die Verteilungsformen für das Auftreten dieser Abstände bei festem  $\epsilon = 0.2$  dargestellt. Betrachtet man diese, so kann man erkennen, dass der Abstand bei der reinen Kodierung nach oben hin beschränkt ist, was bei der unitären nicht zu beobachten ist. Es gibt also bei der unitären Kodierung Störrichtungen  $\Delta m$ , die sich besonders stark auf die Merkmale auswirken.

In Abbildung 4.10 werden die sechs 4-knotigen Objekte aus Abbildung 4.5 betrachtet. Es wird ermittelt, wie stark eine Störung sein darf, dass die sechs Objekte noch

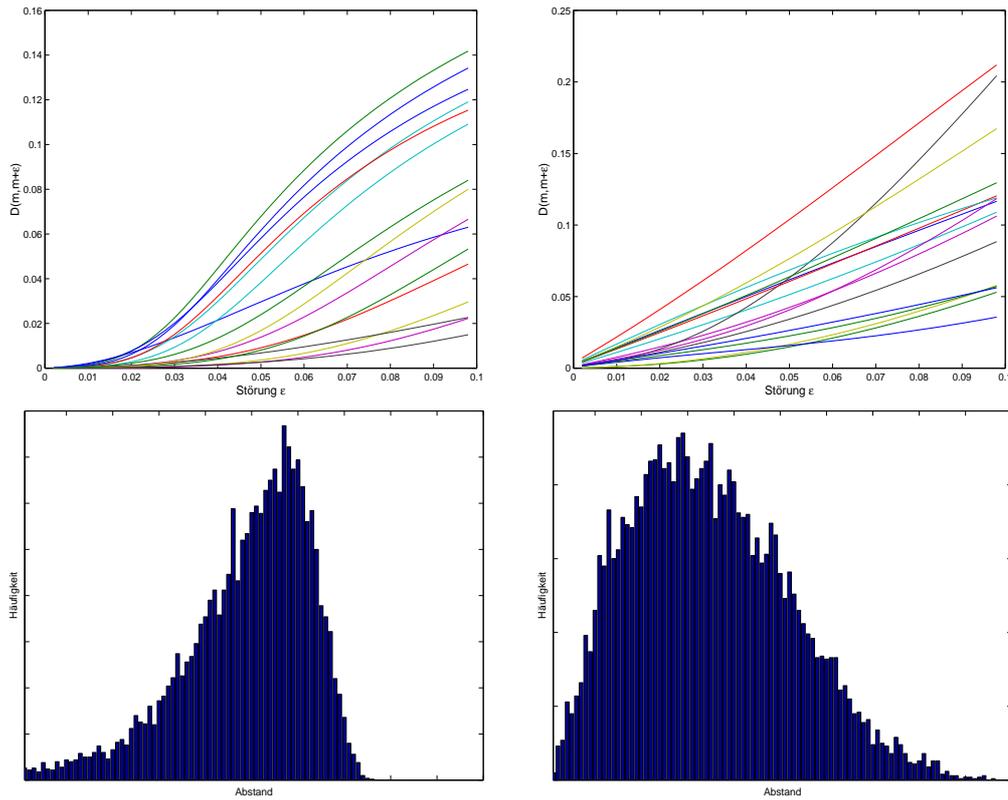
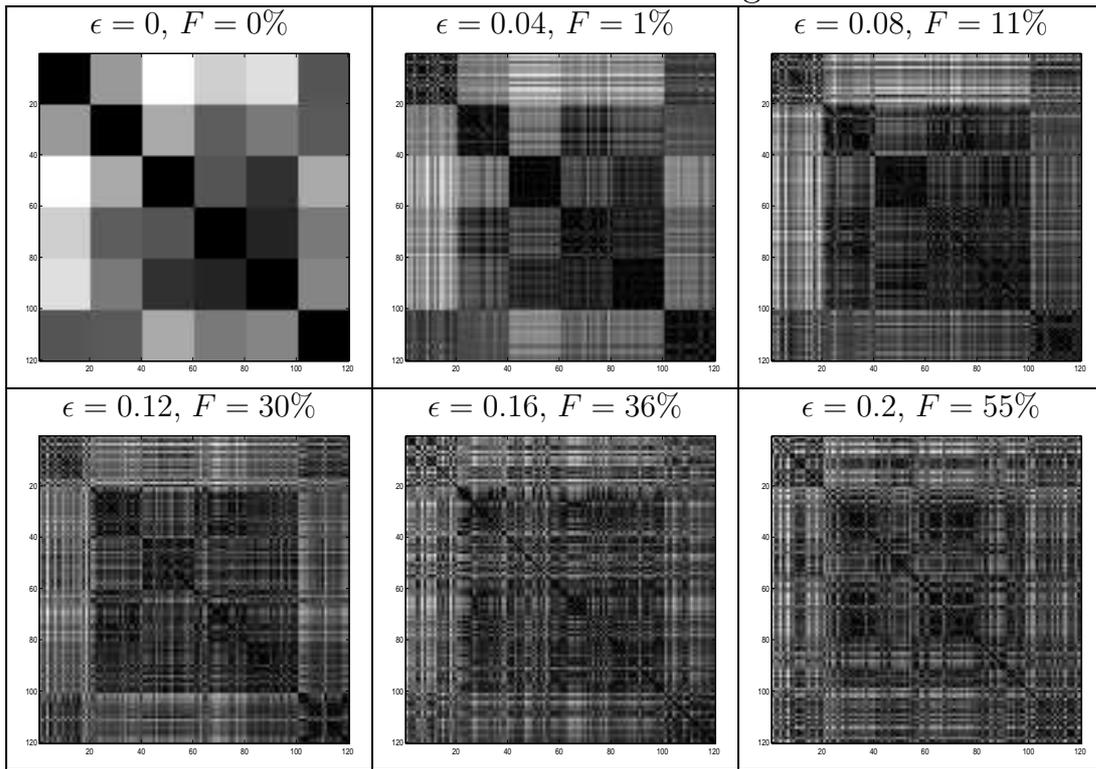


Abbildung 4.9: **Störungsuntersuchung für ein Quadrat.** Links: 3Distanzkodierung. Rechts: Unitäre 3Distanzkodierung. Es sind für verschiedene Störungen  $\{\Delta\vec{x}_i\}$  die Abstände  $D(m, m + \epsilon)$  vom gestörten Quadrat  $\vec{x}'_i = \vec{x}_i + \epsilon\Delta\vec{x}_i$  zum ursprünglichen Quadrat aufgetragen.

zu unterscheiden sind. Für jedes der sechs Objekte wurden zehn zufällige, gestörte Versionen erzeugt und diese insgesamt 60 Objekte in Ähnlichkeitsmatrizen gegenübergestellt. Dies wurde für fünf verschiedene Störintensitäten  $\epsilon$  gemacht. Weiter wurde der prozentuale Zuordnungsfehler  $F$  berechnet, der in der Literatur [8] auch als 'leave one out'-Fehler bekannt ist. Ein Objekt  $m$  wird falsch zugeordnet, falls es einen Abstand  $D(m, m')$  gibt, der minimal für ein Objekt  $m'$  wird, welches nicht zu der Objektklasse des Objekts  $m$  gehört. Dieser primitive Klassifikatoransatz ist auch als 'Nächste Nachbar'-Klassifikator bekannt. Der Zuordnungsfehler  $F$  ist dann der Anteil der falsch zugeordneten Objekte. Er schwankt für verschiedene Stichproben um die  $\pm 1\%$ . Es wurden wieder Versuche zur reinen und unitären 3Distanzkodierung gemacht. Für die 4-knotigen Objekte schneidet die unitäre Kodierung vom Zuordnungsfehler leicht besser ab.

Es wurden noch zwei weitere Versuchsreihen mit mehrknotigen Objekten vorgenommen. In Tabelle 4.1 sind die Ergebnisse dargestellt. Es ist zu beobachten, dass der Fehler für die vielknotigen Objekte, unabhängig von der Kodierung, weit größer

### Reine 3Distanzkodierung



### Unitäre 3Distanzkodierung

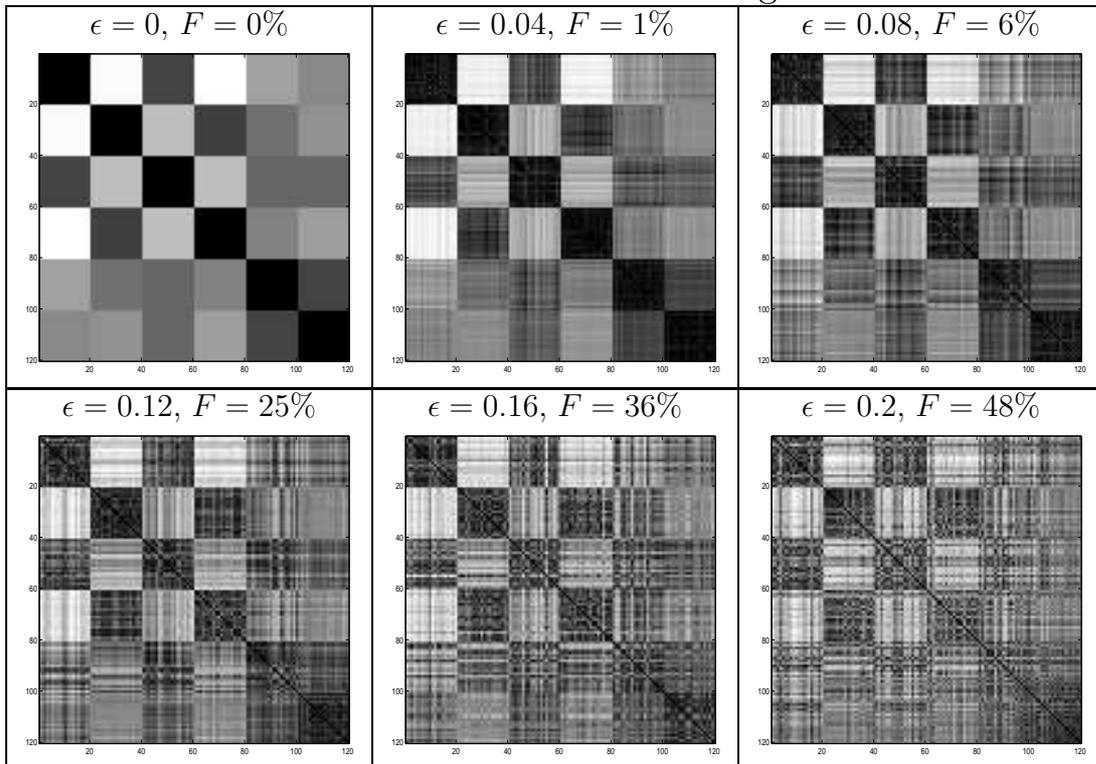
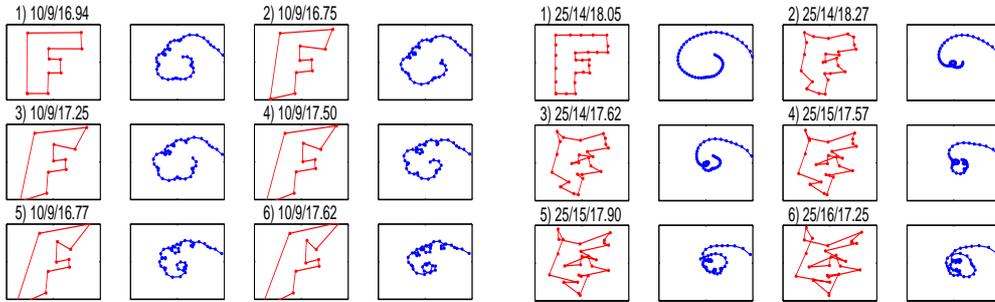


Abbildung 4.10: Störungsuntersuchung für 4-knotige Objekte mit 3Distanzkodierung

reine 3Distanz				unitäre 3Distanz			
$\epsilon$	4er	poly1	poly2	$\epsilon$	4er	poly1	poly2
0.01	0%	2%	3%	0.01	1%	2%	2%
0.04	1%	14%	10%	0.04	1%	30%	10%
0.08	11%	40%	38%	0.08	6%	65%	40%
0.12	30%	50%	60%	0.12	25%	79%	70%

Tabelle 4.1: **Klassifikationsfehler.**

ist als bei den 4-knotigen Objekten. Dies ist damit zu erklären, dass in der Regel für Objekte bei konstanter Ausdehnung und steigender Knotenzahl die Abstände benachbarter Knoten kleiner werden. Die Störungen wirken sich also bei vielen Knoten lokal unkorrelierter aus. Kleine Abstände ändern sich relativ stärker bei gleich bleibender Störintensität als größere Abstände. Im folgenden Versuch wird dieses Verhalten demonstriert:



Links ist das Objekt mit möglichst wenig Knoten konstruiert, rechts mit relativ vielen Knoten. Die Störung bewegt sich wie in den vorigen Versuchen zwischen  $\epsilon = 0 \dots 0.2$ . Für den Betrachter erscheint die Störung des vielknotigen Objekts stärker als die des wenigknotigen Objekts. Für das vielknotige Objekt kann schon eine Störintensität von  $\epsilon = 0.04$  als stark bezeichnet werden.

In Abbildung 4.11 wird die Störempfindlichkeit der einzelnen Merkmalskomponenten  $I_k$  für verschiedene Objekte genauer untersucht. Für jedes  $k$  wird der Erwartungswert der Abweichung der Merkmalskomponenten

$$\Delta I_k(m) = \sqrt{\frac{1}{N} \sum_{\epsilon} (I_k(m) - I_k(m + \epsilon))^2}$$

gemessen, wobei  $N$  die Anzahl der Messungen ist. Offenbar sind die Merkmalskomponenten  $I_k$  mit großem  $k$  sehr sensibel gegenüber Störungen. Dieses Verhalten war auch zu erwarten, da sich anfänglich kleine Änderung mit fortlaufender Exponentiation von  $\mathbf{H}$  immer weiter vergrößern. Das Verhalten ändert sich nicht wesentlich für die verschiedenen Objekte.

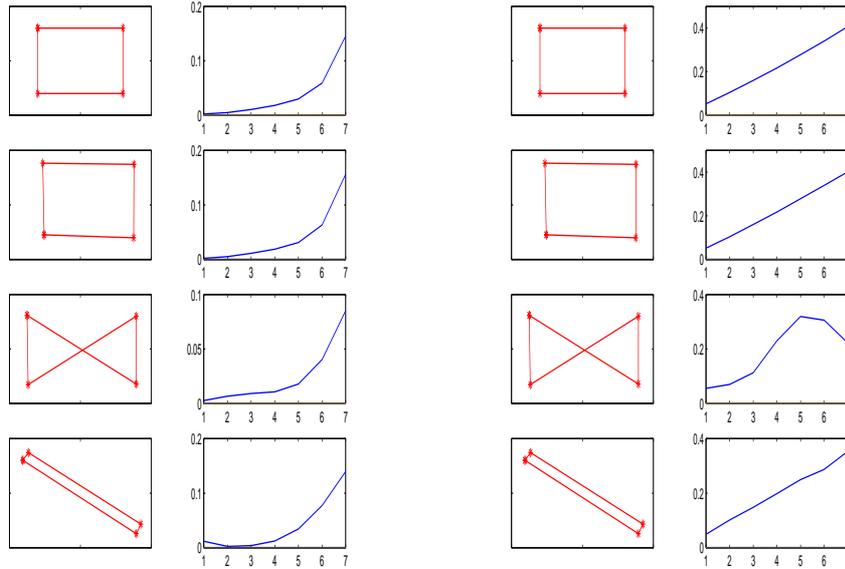


Abbildung 4.11: **Störungssensibilität der Merkmalskomponenten.** Links: 3Distanzkodierung. Rechts: Unitäre 3Distanzkodierung. Es ist mittlere Abweichung der  $2n - 1$  Merkmalskomponenten  $I_k$  für Störungen mit  $\epsilon = 0.2$  aufgetragen.

## 4.4 Vollständigkeitsbetrachtungen

In den Experimenten haben wir gesehen, dass die Merkmalsmatrizen  $\mathbf{C}(m)$  für Polygonzüge mit relativ wenig Knoten (weniger als 10 Knoten) regulär sind. In diesem Abschnitt wollen wir am Beispiel der asymmetrischen Distanzkodierung (4.5) zeigen, dass es unter dieser Voraussetzung möglich ist, mit 'einfachen' Mitteln der Linearen Algebra einen Vollständigkeitsbeweis zu führen. Die folgenden Ausführungen sollen mehr als Demonstration dienen, dass ein einfacher mathematischer Zugang für eine Vollständigkeitsanalyse der in dieser Arbeit vorgestellten Merkmale existiert. Das eigentliche Ergebnis, die Vollständigkeit der asymmetrischen 2Distanz-Merkmale, ist von geringerer Bedeutung, da sie von geringerem praktischen Interesse sind, wegen ihrer Unstetigkeit und Größe ( $n^2$ -Merkmale).

Setzen wir also voraus, dass die Merkmalsmatrix  $\mathbf{C}(m)$  regulär ist. Das heißt die Vektoren  $\mathbf{v}_k$  sind in diesem Fall bis auf unitäre Transformationen  $\tilde{\mathbf{U}}$ , die den Startvektor  $\mathbf{v}_0$  fest lassen, zurückgewinnbar. Nun muss zum einen die Kodierung  $\mathbf{H}(m)$  durch  $(n-1)$  Bilder der Vektoren  $\mathbf{v}_k$  festgelegt sein (siehe Abschnitt 3.5.2) und zum anderen muss die Form von  $\mathbf{H}(m)$  eindeutig unter  $\tilde{\mathbf{U}} \in U_{\mathbf{v}_0}(n)$  sein (siehe Abschnitt 3.5.3). Diese beiden Dinge wollen wir im folgenden zeigen. Beginnen wir mit dem zweiten Teil.

Erinnern wir uns kurz an die Distanzkodierung (4.4), sie setzt sich aus einem diagonalen, imaginären Teil  $\mathbf{H}_a = \mathbf{iWD}_1 = \mathbf{iD}$  und einem symmetrischen, reellen Teil  $\mathbf{H}_h$  zusammen, wobei ersterer antihermitesch und der zweite hermitesch ist. Diese Tatsache

benutzen wir im folgenden

**Satz 3 (Eindeutigkeit).** *Sei  $\mathbf{H}(m)$  die Kodierung aus (4.4) und sei  $n > 4$ . Setzen wir voraus, dass alle  $d_{i(i+1)}$  paarweise verschieden und alle  $d_{i(i+2)}, d_{i(i+3)}$  ungleich null sind, dann gilt für beliebiges  $\mathbf{U} \in U(n)$ ,*

$$\mathbf{U}^\dagger \mathbf{H}(m) \mathbf{U} = \mathbf{H}(m') \Rightarrow \mathbf{U} \in T(n).$$

*Beweis.* Die Matrix  $\mathbf{U}^\dagger \mathbf{H}(m) \mathbf{U}$  hat also die Form einer Kodierung. Die Kodierung (4.4) setzt sich aus einem hermiteschen und antihermiteschen Teil zusammen. Also kann man schreiben

$$\begin{aligned} \mathbf{U}^\dagger \mathbf{H}(m) \mathbf{U} &= \mathbf{H}_a(m') + \mathbf{H}_h(m') \\ &= \mathbf{U}^\dagger \mathbf{H}_a(m) \mathbf{U} + \mathbf{U}^\dagger \mathbf{H}_h(m) \mathbf{U}. \end{aligned}$$

Da eine Matrix eindeutig in einen hermiteschen und antihermiteschen Teil zerlegt werden kann und diese (anti)-hermiteschen Anteile unter unitären Transformationen invariante Unterräume bilden, muss

$$\begin{aligned} \mathbf{H}_a(m') &= \mathbf{U}^\dagger \mathbf{H}_a(m) \mathbf{U} \\ \wedge \mathbf{H}_h(m') &= \mathbf{U}^\dagger \mathbf{H}_h(m) \mathbf{U} \end{aligned}$$

gelten. Somit wissen wir, dass  $\mathbf{U}^\dagger \mathbf{H}_a(m) \mathbf{U}$  diagonal ist. Da vorausgesetzt wurde, dass diese Diagonalelemente alle ungleich sind, wissen wir [20] von der Eindeutigkeit des Spektrums eines hermiteschen Operators, dass es bis auf Permutation nur eine unitäre Transformation gibt, für die eine hermitesche Matrix auf reelle Diagonalform gebracht werden kann. Also muss  $\mathbf{U} \in P(n)$  gelten. Unter den Permutationen erhalten jedoch nur die zyklischen Translationen und Spiegelungen die Form des symmetrischen Teils  $\mathbf{H}_h(m)$  (vorausgesetzt die ersten Nebendiagonalelemente sind alle ungleich null). Also folgt  $\mathbf{U} \in T(n)$ .  $\square$

Es ist zu bemerken, dass die Eindeutigkeit eigentlich nur in Bezug auf unitäre Transformationen, die einen bestimmten Startvektor  $\mathbf{v}_0$  fest lassen, zu zeigen war. Von dieser Tatsache wurde jedoch kein Gebrauch gemacht. Die Eindeutigkeit wurde durch die Asymmetrie der Kodierung erkauft. Für symmetrische Kodierungen, die nur  $\mathcal{O}(n)$  unabhängige Merkmale erzeugen, ist jedoch die Einschränkung auf nur eine Teilmenge von unitären Transformationen von zentraler Bedeutung.

Nun muss noch gezeigt werden, dass die Kodierung durch die Bilder der  $\mathbf{v}_k$  festgelegt ist. Tatsächlich reichen die Bilder der Vektoren  $\mathbf{v}_0$  und  $\mathbf{v}_1$  aus.

**Satz 4.** *Gelten die Voraussetzungen von Satz 3, dann legen die zwei Vektoren*

$$\mathbf{v}_k = \mathbf{H}^k \mathbf{v}_0 \quad k \in \{1, 2\}$$

*die Matrixeinträge der Kodierung  $\mathbf{H}$  eindeutig fest.*

*Beweis.* Die Diagonaleinträge sind durch die Imaginärteile der Komponenten des Vektors  $\mathbf{v}_1$  sofort festgelegt, also  $h_{ii} = \text{Im}\{\mathbf{e}_i^\dagger \mathbf{v}_1\}$ , wobei  $\mathbf{e}_i$  die Standardbasis ist. Für die Nebendiagonalelemente können wir aus der Gleichung für Vektor  $\mathbf{v}_1$  nur schließen, dass  $h_{i(i+1)} + h_{(i-1)i} = \text{Re}\{\mathbf{e}_i^\dagger \mathbf{v}_1\}$  gilt. Betrachten wir also  $\mathbf{v}_2$

$$\begin{aligned} \text{Im}\{\mathbf{v}_2\} &= \text{Im}\{(\mathbf{iD} + \mathbf{H}_s)^2 \mathbf{v}_0\} \\ &= \text{Im}\{-\mathbf{D}^2 + \mathbf{i}(\mathbf{H}_s \mathbf{D} + \mathbf{D} \mathbf{H}_s) + \mathbf{H}_s^2 \mathbf{v}_0\} \\ &= (\mathbf{H}_s \mathbf{D} + \mathbf{D} \mathbf{H}_s) \mathbf{v}_0. \end{aligned}$$

Dann ergibt sich für die Komponenten

$$\text{Im}\{\mathbf{e}_i^\dagger \mathbf{v}_2\} = \underbrace{(h_{ii} + h_{(i+1)(i+1)})}_{a} h_{i(i+1)} + \underbrace{(h_{(i-1)(i-1)} + h_{ii})}_{b} h_{(i-1)i}.$$

Somit haben wir ein lineares Gleichungssystem für  $h_{i(i+1)}$  und  $h_{(i-1)i}$

$$\begin{pmatrix} 1 & 1 \\ a & b \end{pmatrix} \begin{pmatrix} h_{i(i+1)} \\ h_{(i-1)i} \end{pmatrix} = \begin{pmatrix} \text{Re}\{\mathbf{e}_i^\dagger \mathbf{v}_1\} \\ \text{Im}\{\mathbf{e}_i^\dagger \mathbf{v}_2\} \end{pmatrix}, \quad (4.8)$$

wobei  $a$  und  $b$  nur von den schon bekannten Diagonalelementen abhängt. Aus den Voraussetzungen folgt zusätzlich, dass  $a$  und  $b$  ungleich sind, somit ist (4.8) eindeutig lösbar. Also sind auch die Nebendiagonalelemente eindeutig bestimmt.  $\square$

Der Beweis des letzten Satzes ist der Grund für die Wahl der 2Distanzkodierung. Leider ist es dem Autor nicht gelungen, Satz 4 auch für 3Distanzkodierungen zu formulieren, was nicht heißen soll, dass dies nicht möglich ist. Im Gegensatz dazu ist Satz 3 nahezu direkt auf die 3Distanzkodierung übertragbar.

## 4.5 Kommentare

In diesem Kapitel wurde eine Anwendung der in Kapitel 3 vorgestellten Methodik präsentiert. Es wurden distanzbasierte Darstellungen von Polygonzügen angegeben und diese als Kodierungen zur Merkmalsgewinnung eingesetzt. Zudem haben Experimente die praktische Anwendbarkeit der so entstandenen Merkmale gezeigt. In weiteren Störungsuntersuchungen konnte auch die Gutmütigkeit der Merkmale gegenüber kleinen Störungen nachgewiesen werden. Die Vollständigkeitsbetrachtungen haben gezeigt,

dass es in diesem Rahmen möglich ist, mit einfachen Mitteln der Linearen Algebra die Merkmale auf deren Separationsfähigkeit zu untersuchen. Die Konditionierungszahl der Merkmalsmatrix hat sich als sinnvolles Maß für die Qualität der Merkmale herausgestellt. Jedoch hat sich in der Praxis gezeigt, dass die Konditionierungszahl in der Regel relativ groß ist, das heißt die Merkmalsmatrix ist nahezu singulär. Dies ist vor allem durch die schnelle Konvergenz des Merkmalsbildungsprozesses begründet.

Im Vergleich zu bereits existierenden Ansätzen, wie diskrete (Abschnitt 4.1.2) und kontinuierliche (Abschnitt 2.2) Fourierdeskriptoren, schneiden die vorgestellten Merkmale nicht ganz so gut ab. Die Berechnung der kompletten Merkmalsmatrix ist nur in  $\mathcal{O}(n^2)$  möglich. Die diskreten FDs dagegen liefern vollständige Merkmale schon in  $\mathcal{O}(n \log(n))$ , sind jedoch nicht zum Vergleich von Polygonzügen mit verschiedenen Knotenzahlen geeignet. Hier bieten die kontinuierlichen FDs den besseren Ansatz. Die einheitliche Parametrisierung des Polygonzuges durch eine stetige Funktion macht die kontinuierlichen FDs unsensibel gegenüber der eigentlichen Darstellung des Polygonzuges durch seine Knoten. Allerdings kann hier keine FFT zur schnellen Berechnung der Merkmale benutzt werden; somit ergibt sich der Aufwand zu  $\mathcal{O}(nk)$ . Die Vorteile, der in diesem Kapitel vorgestellten Vorgehensweise, liegen sicher in der Allgemeinheit und Adaptivität der Invarianten. Es gibt eine Vielzahl von Möglichkeiten, Kodierungen zu konstruieren. Die Kodierungen können mit Parametern versehen werden. Diese können benutzt werden, um verschiedene Eigenschaften der Objekte zu betonen oder zu vernachlässigen. Auf diese Weise kann auch das Problem der schnellen Konvergenz umgangen werden. Statt alle  $n$  Vektoren aus einer Kodierung zu gewinnen, könnten jeweils für verschiedene Parameter einer Kodierung nur einige wenige Vektoren bestimmt werden. Damit ist der Konvergenzprozess noch nicht so weit fortgeschritten und die Merkmale sind reichhaltiger.

# Kapitel 5

## Allgemeine 3D-Drahtgittermodelle

Im diesem Kapitel soll das vorgestellte Konzept auf Gitter mit beliebigem Verbindungsgrad im Dreidimensionalen angewendet werden. Im vorangegangenen Kapitel wurden nur Gitter mit Verbindungsgrad 2 in der Ebene betrachtet, das heißt wir hatten viel 'Platz' in der Matrix um eine Verbindung zu beschreiben, nämlich eine ganze Zeile für jede Verbindung. Bei beliebigem Verbindungsgrad wird es schon schwieriger, die Information zurückgewinnbar in einer  $n \times n$  Matrix zu speichern. Um das Mehr an Information unterzubringen, werden einfach die Matricelemente selbst zu kleinen Matrizen. Die weitere Vorgehensweise soll durch eine physikalische Feldgleichung motiviert werden.

Im folgenden wird zuerst die neue Kodierung für 3D-Drahtgittermodelle definiert und daraufhin die GUI-Anwendung, die im Rahmen dieser Arbeit entwickelt wurde, kurz vorgestellt. Diese wird dann in den darauf folgenden Experimenten als Analysewerkzeug benutzt und kann auch für zukünftige Untersuchung von invarianten Merkmalen geometrischer Objekte dienen.

### 5.1 Kodierung

In der relativistischen Quantentheorie beschreibt die Dirac-Gleichung

$$\mathbf{i} \frac{d}{dt} \vec{\mathbf{v}}(t) = \left( \sum_{j=1}^3 \alpha_j P_j + mc\beta + V \right) \vec{\mathbf{v}}(t) \quad (5.1)$$

das Verhalten eines Teilchens mit halbzahligen Spin, zum Beispiel das eines Elektrons. Eine schöne Darstellung dieser Gleichung gibt es in [18]. Der beschreibende Zustandsvektor  $\vec{\mathbf{v}}(t)$  ordnet nun nicht mehr jedem Punkt im Raum (in unserem Bild jedem Knoten) einen Skalar zu, sondern einen 4-komponentigen Vektor. Der Vektor  $\vec{\mathbf{v}}(t)$  ist also aus dem Produktraum  $\mathbb{H} \otimes \mathbb{C}^4$ . Die Impulsoperatorkomponenten  $P_i$  sind definiert durch  $\mathbf{i} \frac{d}{dx_j}$  (für Physiker: in Ortsdarstellung). Die  $\alpha_j$  und  $\beta$  sind spezielle antihermitesche Matrizen  $\mathbb{C}^{4 \times 4}$ , die bestimmte Kommutatorrelationen erfüllen. Der Faktor  $m$

entspricht der Masse des Teilchens und  $c$  ist die Lichtgeschwindigkeit. Das Potential  $V$  soll ein in der Zeit konstantes und ortsunabhängiges Potential sein.

Wir versuchen nun, die Gleichung in unserem diskreten Gitterraum zu interpretieren. Der Impulsoperator  $P_j$  kann diskret als Differenzenbildung zwischen den Nachbarn interpretiert werden. Die Summe über  $j$  erstreckt sich über alle Nachbarn eines Knotens  $i$  und die  $\alpha_j$  beschreiben die geometrischen Nachbarschaftsverhältnisse, wobei diese aber nun von Knoten zu Knoten unterschiedlich sind, das heißt die  $\alpha_j$  hängen zusätzlich von dem entsprechenden Knoten  $i$  ab, also schreiben wir  $\alpha_{ij}$ . Genauso hängen nun die  $\beta_i$  vom jeweiligen Knoten  $i$  ab. Die Komponenten  $\vec{v}_i = (\vec{\mathbf{v}})_i \in \mathbb{R}^4$  des Vektors  $\vec{\mathbf{v}}$  sind mit den einzelnen Knoten des Gitters zu assoziieren. Wir wollen nun eine Konstruktionsvorschrift vorstellen, die zu reellen  $4n \times 4n$ -Matrizen führt und sich als Kodierung erweisen wird.

Die Geometrie wird durch die  $4 \times 4$ -Matrizen  $\alpha_{ij}$  und  $\beta_{ij}$  beschrieben

$$\begin{aligned}\alpha_{ij} &:= \begin{pmatrix} [\vec{x}_i - \vec{x}_j]_{\times} & 0 \\ 0^T & 0 \end{pmatrix} = \begin{pmatrix} [\vec{d}_{ij}]_{\times} & 0 \\ 0^T & 0 \end{pmatrix} \\ \beta_{ij} &:= \begin{pmatrix} 0 & (\vec{x}_i - \vec{x}_j) \\ (\vec{x}_j - \vec{x}_i)^T & 0 \end{pmatrix} = \begin{pmatrix} 0 & \vec{d}_{ij} \\ -\vec{d}_{ij}^T & 0 \end{pmatrix},\end{aligned}\quad (5.2)$$

wobei  $[\ ]_{\times}$  die Kreuzproduktmatrix ist, das heißt  $\vec{a} \times \vec{b} = [\vec{a}]_{\times} \vec{b}$ . Für diese Matrizen gilt bei einer Rotation  $R$  des dreidimensionalen Koordinatensystems

$$\begin{aligned}\alpha'_{ij} &= \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \alpha_{ij} \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}^T \\ \beta'_{ij} &= \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \beta_{ij} \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}^T.\end{aligned}\quad (5.3)$$

Die in (5.2) gewählten Matrizen sind nur eine Möglichkeit. Es haben lediglich die Bedingungen (5.3) zu gelten, das heißt die Geometriematrizen müssen wie ein Tensor 2. Stufe transformieren. Wählt man die Matrizen wie in (5.2), dann haben Punkte, die weit entfernt und verbunden sind, einen größeren Einfluss auf die Merkmale als verbundene Punkte mit geringem Abstand. Es könnte auch wünschenswert sein, dass dies genau andersherum gilt, also dass der Einfluss umgekehrt proportional zum Abstand ist. Wir werden uns jedoch auf die Geometriematrizen in (5.2) bei allen folgenden Betrachtungen beschränken. Jetzt können wir die uminterpretierte Version der Gleichung (5.1) in Komponentenschreibweise aufschreiben

$$\frac{d}{dt} \vec{v}_i(t) = \sum_{(i,j) \in E} \alpha_{ij} (\vec{v}_j(t) - \vec{v}_i(t)) + m \beta_i \vec{v}_i(t) + V \vec{v}_i(t),\quad (5.4)$$

wobei  $\beta_i = \sum_{(i,j) \in E} \beta_{ij}$ . In kompakter Schreibweise

$$\frac{d}{dt} \vec{\mathbf{v}} = \mathbf{H} \vec{\mathbf{v}}.$$

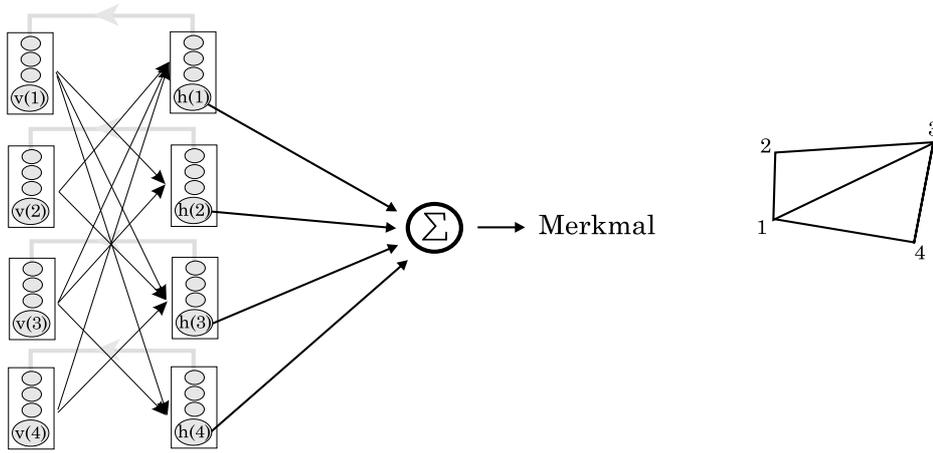


Abbildung 5.1: **Berechnung der Merkmale.** Aus dem Vektor  $\mathbf{v}$  wird in Abhängigkeit der Verbindungsverhältnisse des Gitters der neue Vektor gewonnen. Zur Merkmalsgewinnung dient nur die vierte 'isotrope', richtungsunabhängige, Komponente des Vektors.

Nun ergibt sich für die Matrixelemente der Matrix  $\mathbf{H}$  ausgehend von (5.4) in Blockschreibweise (das heißt die Matrixelemente sind  $4 \times 4$ -Matrizen):

$$h_{ij} := \begin{cases} -\sum_{(i,j) \in E} (\alpha_{ij} - m\beta_{ij}) + V & \text{falls } i = j \\ \alpha_{ij} & \text{falls } (i, j) \in E \\ 0 & \text{sonst.} \end{cases}$$

Die so entstandene Kodierung ist für  $V = 0$  antisymmetrisch. Die Wahl eines Potentials  $V$  ungleich Null bietet den Vorteil, das Gitter bei verschiedenen 'Energien' betrachten zu können.  $V$  kann man sozusagen als spektralen *Shiftparameter* ansehen.

Bei der späteren Implementation wird uns nur die Anwendung der Matrix auf einen Vektor interessieren. Natürlich muss dafür nicht die komplette Matrix aufgestellt werden. Die Anwendung besteht für einen Knoten des Gitters aus der Berechnung eines Kreuzproduktes und eines Skalarproduktes für jeden Nachbarn des Knotens. In (5.5) ist dies für eine Komponente  $\vec{v}_i = (\vec{q}_i, w_i)^T$  dargestellt

$$\begin{pmatrix} \vec{q}'_i \\ w'_i \end{pmatrix} = \sum_{(i,j) \in E} \begin{pmatrix} \vec{d}_{ij} \times (\vec{q}_i - \vec{q}_j) - m\vec{d}_{ij}w_i \\ m\vec{d}_{ij} \cdot \vec{q}_i \end{pmatrix} + V \begin{pmatrix} \vec{q}_i \\ w_i \end{pmatrix}. \quad (5.5)$$

Betrachten wir das Transformationsverhalten der Kodierung genauer, um einen geeigneten Startvektor zu finden. Eine Permutation der Knoten wirkt sich durch  $\mathbf{P}_\sigma = \mathbf{P}'_\sigma \otimes \mathbf{I}_4$  auf die Konstruktion aus. Die euklidische Bewegung  $\vec{x}' = R\vec{x} + \vec{a}$  wirkt sich jetzt durch eine Matrix  $\mathbf{U}_\varepsilon$  verschieden von der Identität aus. Die Wirkung der euklidischen Bewegung ist für alle Komponenten des Knotenraums gleich und lässt sich somit als  $\mathbf{U}_\varepsilon = (\mathbf{I}_n \otimes \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}) = (\mathbf{I}_n \otimes R')$  schreiben. Somit ergibt sich für die gesamte Wirkung

von Permutation und Rotation

$$\mathbf{P}_\sigma \mathbf{U}_\varepsilon = (\mathbf{P}'_\sigma \otimes \mathbf{I}_4)(\mathbf{I}_n \otimes R') = \mathbf{P}'_\sigma \otimes R'. \quad (5.6)$$

Der Startvektor  $\mathbf{v}_0$  muss den Bedingungen (3.6) genügen und sollte zusätzlich identisch für alle Gitter (gleicher Knotenzahl) sein. In letzterer Forderung besteht auch ein Grund für die Wahl einer 4-dimensionalen Darstellung der Matrixkomponenten, weil sich so sehr einfach ein Vektor angeben lässt

$$(\vec{\mathbf{v}}_0)_i = (0, 0, 0, 1)^T. \quad (5.7)$$

Der Vektor erfüllt offenbar die Bedingungen (3.6), also

$$\mathbf{P}_\sigma \mathbf{U}_\varepsilon \vec{\mathbf{v}}_0 = \vec{\mathbf{v}}_0,$$

und ist zusätzlich für alle Knoten identisch. Würde man eine dreidimensionale Darstellung wählen und die Geometriematrizen würden bei Koordinatentransformationen wie ein Tensor transformieren, dann existiert außer der Null kein Vektor, der die Bedingungen eines Startvektors erfüllt und identisch für alle Gitter ist. Somit ist man fast gezwungen, eine vierdimensionale Darstellung zu wählen. Wir nennen die ersten drei Komponenten Ortskomponenten und die vierte bezeichnen wir als isotrope Komponente, welche durch die Masse  $m$  mit den drei Ortskomponenten koppelt. Nach der ersten Anwendung der Kodierung auf dem Startvektor erhält man (5.7)

$$(\vec{\mathbf{v}}_1)_i = \sum_{(i,j) \in E} \begin{pmatrix} -m \vec{d}_{ij} \\ 0 \end{pmatrix} + V_0 \begin{pmatrix} \vec{0} \\ 1 \end{pmatrix} = \begin{pmatrix} -m \sum_{(i,j) \in E} \vec{d}_{ij} \\ V_0 \end{pmatrix}.$$

In den ersten drei Komponenten steht die Summe über die Differenzvektoren des Knotens  $i$  zu seinen Nachbarn. Die vierte Komponente hängt nach der ersten Anwendung noch nicht von der Gestalt des Gitters ab. In Abbildung 5.1 ist der Merkmalbildungsprozess durch Mittelung graphisch veranschaulicht. In die Merkmale fließen nur die isotropen Komponenten des Vektors ein.

Neben der Merkmalsgewinnung durch Mittelung wurde auch die Merkmalsbildung durch Histogramme eingeführt. Eine Histogrammbildung über die einzelnen  $4n$  Komponenten der Vektoren  $\vec{\mathbf{v}}_k$  ist, wie in der Einführung schon erwähnt, nur permutationsinvariant. Benutzen wir jedoch unsere Definition (3.10) eines Merkmalshistogramms

$$I_{k\lambda}(m) := \vec{\mathbf{v}}_0^\dagger f(\vec{\mathbf{v}}_k - \lambda \vec{\mathbf{v}}_0),$$

wobei  $\vec{\mathbf{v}}_0$  wie in (5.7) gewählt wird, dann ist  $I_{k\lambda}(m)$  tatsächlich auch ein lageinvariantes Merkmal, da die Ortskomponenten bei der Skalarproduktbildung mit  $\vec{\mathbf{v}}_0$  heraus fallen.

Merkmalshistogramme, die durch eine Rechteckfunktion gewonnen worden sind, eignen sich nicht besonders gut zum direkten Vergleich, da sie sehr störepfindlich sind. Zum Vergleich sind weiche glockenförmige Urnenfunktionen viel besser geeignet, was ja gerade einer Tiefpassfilterung des störepfindlichen Rechteckhistogramms entspricht. In den folgenden Versuchen werden zum Vergleich von Objekten wieder nur die Mittelungsmerkmale betrachtet, jedoch werden die Histogramme zur Veranschaulichung des Merkmalbildungsprozesses dienen.

## 5.2 Implementation und Schnittstellen

Im Rahmen der Versuche mit 3D-Drahtgittermodellen wurde eine *GUI*-Anwendung entwickelt, die zur Analyse und Untersuchung verschiedener Methoden der invarianten Merkmalsgewinnung dienen soll. Folgende Analysewerkzeuge wurden implementiert:

1. **Objekttransformationen.** Wie verhalten sich die Objekte bei Rotation, Translation, Skalierung und kleinen Störungen?
2. **Ähnlichkeitsmatrizen.** Wie separieren die Merkmale die Objektmengen? Wie groß müssen Störungen sein, dass zwei verschiedene Objekte nicht mehr unterscheidbar sind?
3. **Objektsuche.** Welche Objekte sind einem gegebenen Objekt ähnlich?

Die Anwendung wurde in C++ mit Hilfe vom *QT-Windowing*-System entworfen. Sie wurde möglichst modular gestaltet, damit sich leicht neue Schnittstellen, Objekttypen und Merkmalsberechnungen einbinden lassen. Es wurden Schnittstellen zu drei verschiedenen 3D-Objekt-Formaten integriert:

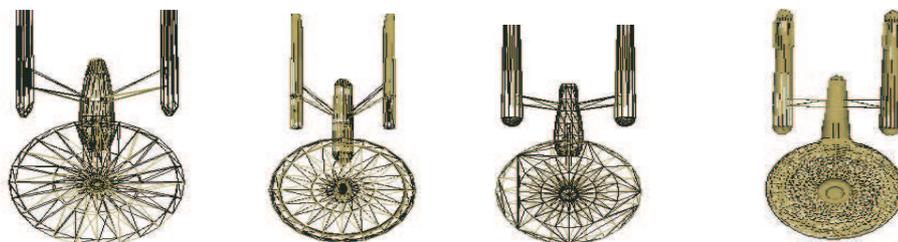
1. **(.poly) Ascii Text.** [17] Beliebige polygonale Strukturen. Daten bestehen aus einer Punktmenge  $X = \{x_i | i \in \{1, \dots, n\}\}$  und einer Polygonmenge  $P = \{p_i | i \in \{1, \dots, m\}\}$ , wobei ein Polygon  $p_i$  durch eine geordnete Punktmenge gegeben ist, die nicht notwendigerweise in einer Ebene liegen.
2. **(.3ds) Autodesk 3DStudio.** Beliebige triangulierte Strukturen. Daten bestehen aus einer Punktmenge  $X$  und einer Dreieckmenge  $P$ .
3. **(.pdb) Protein Database.** Molekulare Strukturen. Daten bestehen aus einer Atommenge  $X$ , wobei jedes Atom durch seinen Ort und seine Elementbezeichnung charakterisiert ist. Als Bindungsinformation wird jedem Atom zusätzlich die Menge seiner Nachbaratome (die Atome, die mit diesem in chemischer Bindung stehen) zugeordnet.

Bei Objekttypen wird zwischen polygonalen Oberflächenmodellen (.poly und .3ds) und molekularen Modellen (.pdb) unterschieden. Molekulare Modelle besitzen zusätzlich Knotenattribute (Elementbezeichnung) und Kantenattribute (z.B. Doppel- oder Einzelbindung) und unterscheiden sich in der internen Repräsentation von den Oberflächenmodellen. Im Anhang wird die entwickelte Anwendung noch etwas genauer beschrieben.

Das WWW stellt eine Vielzahl von Objekten in den obigen Formaten zum freien *Download* bereit (*3D Cafe*, *Protein Database*). Im Rahmen dieser Arbeit wurden solche Objekte gesammelt und archiviert.

### 5.3 Experimente

In den folgenden Experimenten soll nun die oben vorgestellte Kodierung auf ihre Tauglichkeit überprüft werden. Im vorigen Kapitel konnte schon gesehen werden, dass unsere Merkmale von der 'Bauweise' der Testobjekte abhängen, das heißt die Merkmale waren sensitiv gegenüber der Parametrisierung des Polygonzuges. Für den menschlichen Betrachter ähnliche Objekte, die sich aber in der Anzahl der Knoten unterscheiden, lieferten unterschiedliche Merkmale. Für 3-dimensionale Oberflächenmodelle kommen diese Unterschiede noch weit mehr zum tragen, da der 'Konstrukteur' eines Objektes eine viel größere Freiheit beim Design seines Objektes besitzt. Betrachtet man zum Beispiel die folgenden vier Raumschiffe,



dann sollten die Merkmale die Gemeinsamkeiten der Objekte (dass es sich um die *Enterprise* handelt) erkennen. Dies ist jedoch von unseren Merkmalen nicht zu erwarten. Deshalb sollten zum Test Objekte herangezogen werden, die auf 'einheitliche' Weise konstruiert worden sind. Das heißt die Oberflächen sollten auf identische Weise trianguliert worden sein oder besser gesagt, die dem Gitter zu Grunde liegenden Graphen sollten isomorph sein. Andererseits könnte gerade die Sensitivität der Merkmale gegenüber der Gitterstruktur ausgenutzt werden, um Objekte zu vergleichen, dessen zugrunde liegender Graph ausschlaggebend für dessen Identität ist. Hier bieten sich sehr gut Moleküle an. Die Atome stehen gerade für die Knoten des Gitters und die

atomaren Bindungen werden als Verbindungsstruktur des Gitters interpretiert. In den folgenden Untersuchungen werden also zwei Datensätze untersucht. Der eine wird aus bekannten Molekülen der Schulchemie, der zweite wird aus einer Sammlung von Fischen bestehen. Fische bieten sich gerade deshalb an, da deren Körper in der Regel topologisch äquivalent zu einer 2-Sphäre ist.

Zum Vergleich werden die in Abschnitt 2.1 vorgestellten *ShapeDistribution* verwendet. Diese wurden jedoch nur im Zusammenhang mit triangulierten Oberflächen eingeführt. Deshalb wird hier die Methode auch für allgemeine Gitter definiert und somit können auch Moleküle betrachtet werden. Die zufälligen Punkte werden nicht mehr auf Oberflächen verteilt, sondern auf den Verbindungsgeraden benachbarter Knoten. Diese Vorgehensweise nennen wir ab jetzt *GridDistribution*.

### 5.3.1 Moleküle

Beschäftigen wir uns also zuerst mit den Molekülen. Im Anhang sind alle benutzten 55 Moleküle in einer Tabelle dargestellt. In Abbildung 5.3 sind beispielhaft für einige Moleküle die durch die oben eingeführte Kodierung erzeugten Merkmale dargestellt. Die Merkmale werden sowohl durch Mittelung als auch mit Histogrammen ermittelt. Ebenso wenden wir die eben vorgestellte *GridDistribution* auf die Moleküle an. Die Mittelungsmerkmale werden wieder mit Iterationsvorschrift (3.14) gewonnen und die Winkelanteile  $I'_k$  als Merkmale benutzt. Unabhängig von der Anzahl der Atome eines Moleküls berechnen wir im folgenden immer eine feste Anzahl  $M = 100$  der Merkmale  $I'_k$ . Für die Histogrammbildung wird ebenfalls Iterationsvorschrift (3.14) benutzt und als Urnenfunktion eine Rechteckfunktion gewählt. Die *GridDistribution* wird mittels der *ShapeFunction* **D1** (siehe Abschnitt 2.1) gewonnen. Es wird also der euklidische Abstand zwischen zwei zufällig gewählten Punkten auf der Verbindungsstruktur des Moleküls gemessen.

Betrachten wir Abbildung 5.3. Auf den ersten Blick ähneln sich die Mittelungsmerkmale doch sehr, obwohl die Moleküle im Grunde sehr unterschiedlich sind. Die Histogramme scheinen da um einiges reichhaltiger zu sein, jedoch fällt, wie schon erwähnt, bei Histogrammen der Vergleich schwerer. Sie spiegeln in gewisser Weise die Symmetrieeigenschaften der Moleküle wieder. Betrachten wir als erstes 'methane'. Das Histogramm besteht aus 2 Zweigen, der eine ist mit dem zentralen *C*-Atom zu assoziieren, der zweite besteht eigentlich aus vier einzelnen, den *H*-Atomen zugeordneten Zweigen. Dass diese vier Zweige nicht auf verschiedene Niveaus aufspalten, liegt gerade an der Symmetrie des Atoms. Hierauf werden wir später am Beispiel von 'benzene' noch genauer eingehen.

Betrachtet man die *GridDistribution*, so ist deutlich zu erkennen, wie für Mo-

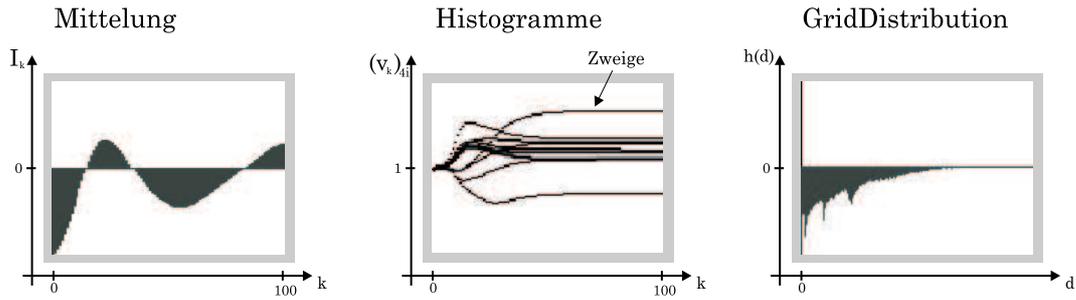


Abbildung 5.2: **Diagrammerklärung.** Die Histogramme sind als überlagerte Graphen der isotropen Komponentenfunktionen zu verstehen.

Name	Objekt	Mittelung	Histogramm	GridDistrib.
ethane ( $C_2H_6$ )		"b_m" calctime:1ms ethane.pdb Atoms:8 	"Histo" calctime:1ms ethane.pdb Atoms:8 	"Grid Distribution" calctime:137ms ethane.pdb Atoms:8 
benzene ( $C_6H_6$ )		"b_m" calctime:1ms Benzene.pdb Atoms:12 	"Histo" calctime:2ms Benzene.pdb Atoms:12 	"Grid Distribution" calctime:172ms Benzene.pdb Atoms:12 
toluene		"b_m" calctime:0ms Toluene.pdb Atoms:15 	"Histo" calctime:1ms Toluene.pdb Atoms:15 	"Grid Distribution" calctime:181ms Toluene.pdb Atoms:15 
methane ( $CH_4$ )		"b_m" calctime:0ms methane.pdb Atoms:5 	"Histo" calctime:1ms methane.pdb Atoms:5 	"Grid Distribution" calctime:111ms methane.pdb Atoms:5 
boranel		"b_m" calctime:2ms borane01.pdb Atoms:44 	"Histo" calctime:4ms borane01.pdb Atoms:44 	"Grid Distribution" calctime:291ms borane01.pdb Atoms:44 
menthol		"b_m" calctime:2ms menthol.pdb Atoms:31 	"Histo" calctime:2ms menthol.pdb Atoms:31 	"Grid Distribution" calctime:249ms menthol.pdb Atoms:31 

Abbildung 5.3: **Beispiele**

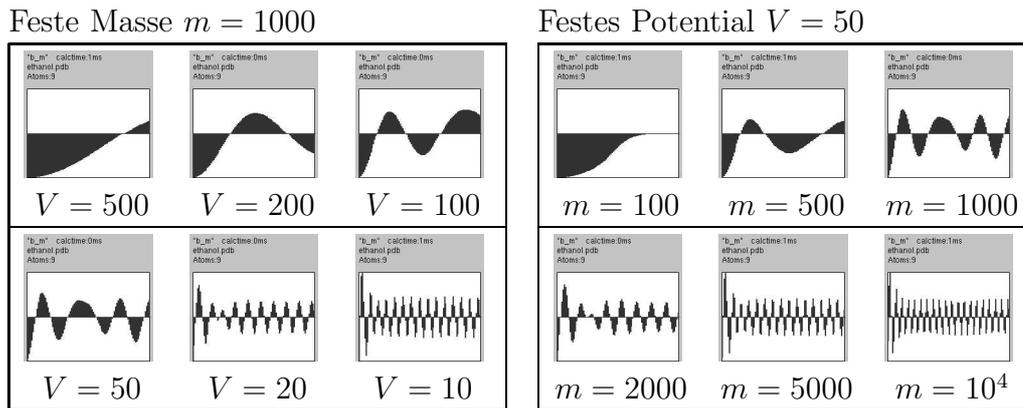


Abbildung 5.4: **Merkmale von Ethanol für verschiedene Parameter.**

leküle mit kleinem Durchmesser, wie 'methane', die Abstandshäufigkeiten nur für kleine Abstände von Null verschieden sind. Für die zwei größeren Moleküle kann beobachtet werden, dass für das eher längliche Molekül 'methane' die Häufigkeiten für größere Abstände schneller ausdünnen als für das voluminösere Molekül 'borane', was auch zu erwarten ist.

In Abbildung 5.4 wurden die Mittelungsmerkmale beispielhaft für verschiedene Parameter berechnet. Man sieht, wie mit steigendem Masseparameter die Frequenz der Merkmale zunimmt, im Gegensatz dazu fällt die Frequenz bei steigendem Potential. Die Parameter  $1/V$  und  $m$  spielen also eine ähnliche Rolle, jedoch hängen die Merkmale nicht nur vom Verhältnis  $m/V$  ab. Betrachtet man die Merkmale mit  $m = 1000, V = 500$  und  $m = 100, V = 50$ , so kann man erkennen, dass sich die Merkmale unterscheiden. Wählt man die Masse zu groß bzw. das Potential zu klein, wird die Frequenz im Vergleich zur 'Abtastrate' zu groß, so dass sich instabile Merkmale ergeben, wie später in den Störungsuntersuchungen zu sehen sein wird. Die Frequenz der Merkmale ist weitgehend unabhängig von dem zugrundeliegenden Objekt, wie schon in Abbildung 5.3 zu erkennen war. Für geringe Frequenzen ist eine hohe Korrelation zwischen benachbarter Merkmalskomponenten  $I_k$  und  $I_{k+1}$  zu erkennen. Für höhere Frequenzen sind Korrelationen zwischen den Komponenten  $I_k$  und  $I_{k+\Delta k}$  zu beobachten, wobei  $1/\Delta k$  etwa der Frequenz der Merkmale entspricht.

Um die Suchfähigkeiten zu testen, wurde ein Teil der Moleküle in drei Klassen unterteilt: Kohlenwasserstoffe, Aromate und Säuren. Den Kohlenwasserstoffen sind die typischen Moleküle wie Ethan, Propan, Butan usw. zuzurechnen. Aromate sind durch die typischen Benzolringe ('benzene') gekennzeichnet. In Abbildung 5.5 wurden Mittelungsmerkmale mit den Parametern  $M = 100, m = 1000, V = 50$  benutzt. Es sind drei verschiedene Suchanfragen nach 'benzene' (Aromat), 'rr-tacid' (Säure) und 'hexane' (Kohlenwasserstoff) dargestellt. Die Rangliste ist nach dem euklidischen Abstand

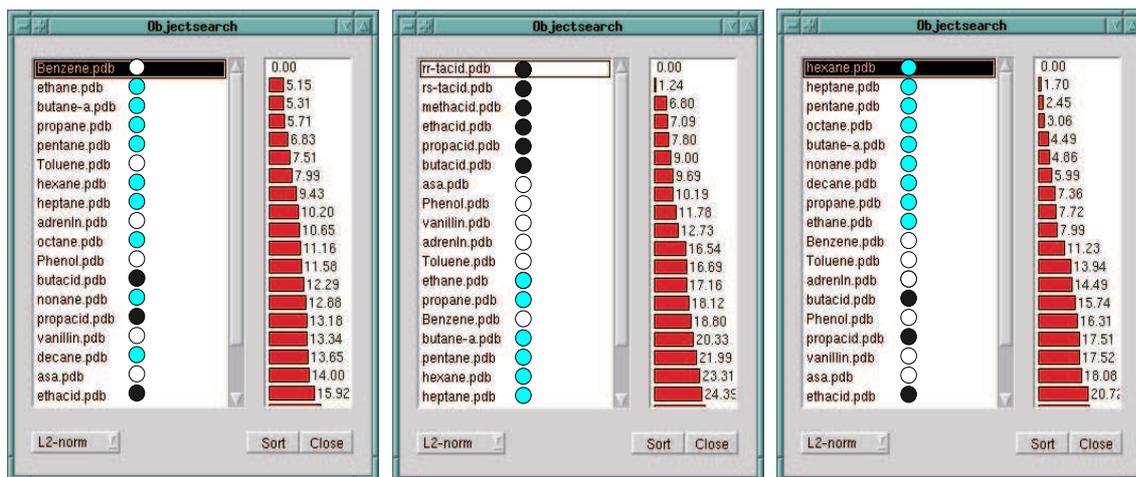


Abbildung 5.5: **Molekülsuche.** Trennende Merkmale für Kohlenwasserstoffe und Säuren.



Abbildung 5.6: **Ähnlichkeitsmatrizen für 3 Klassen.** Links: Invarianten durch Mittelung. Rechts: *GridDistribution*

zum gesuchten Objekt geordnet. Offenbar können die Merkmale zwischen Säuren und Kohlenwasserstoffen unterscheiden, auf die Anfrage 'benzene' jedoch liefert die Suche als erstes vier Kohlenwasserstoffe, erst dann kommt das erste Aromat. In Abbildung 5.6 ist links die zugehörige Ähnlichkeitsmatrix dargestellt, rechts daneben wurde die Ähnlichkeitsmatrix mit der *GridDistribution* gewonnen.

### Störungsuntersuchungen

Ähnlich wie bei den 2-dimensionalen Experimenten wurde die Separationsfähigkeit der Merkmale unter kleinen Störungen untersucht. Für jedes Molekül im Datensatz wurden wieder mehrere gestörte Versionen erzeugt und diese mit dem in Abschnitt 4.3.1 vorgestellten primitiven Klassifikator gegenübergestellt und der prozentuale Zuordnungsfehler  $F$  berechnet. In der folgenden Tabelle sind die Ergebnisse für Mittelungsmerkmale

$\epsilon$	$10^3/50$	$10^3/200$	$200/100$	$100/50$	Grid $10^5$	Grid $10^6$
0	0%	0%	0%	0%	2%	0%
0.01	5%	3%	3%	2%	8%	1%
0.04	20%	11%	9%	10%	10%	1%
0.08	40%	20%	10%	12%	12%	1%
0.12	86%	72%	70%	47%	25%	12%
Zeit	0.01s	0.01s	0.01s	0.01s	0.2s	2s

Tabelle 5.1: **Klassifikationsfehler.**

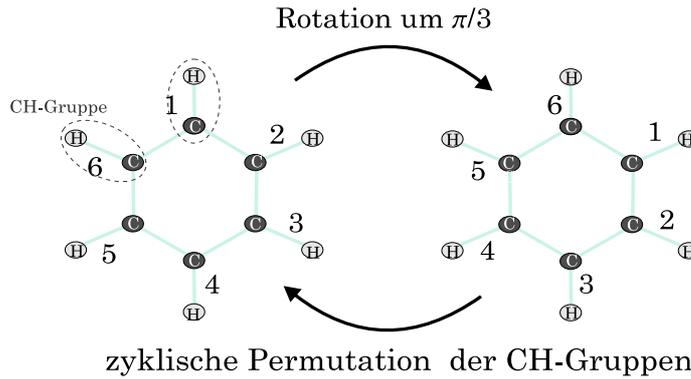
mit verschiedenen Parametern  $m/V$  angegeben. Die Anzahl der Merkmale  $M$  ist für alle Parameter fest mit  $M = 100$  gewählt. Zum Vergleich sind zwei Experimente mit *Grid-Distribution* durchgeführt worden. Hierbei wurde die Anzahl der Abstandsmessungen  $S$  variiert, wobei  $S = 10^5$  bzw.  $S = 10^6$  gewählt wurde. Die letztere Wahl entspricht der Empfehlung der Autoren von [1]. Zusätzlich sind noch die mittleren Berechnungszeiten pro Objekt in Sekunden ermittelt worden (auf einem AMD Duron 1Ghz mit 256MB RAM). Die Tabelle 5.1 fasst die Ergebnisse zusammen. Offenbar ist die *Grid-Distribution* im wesentlichen den Mittelungsmerkmalen vom Klassifikationsfehler her überlegen. Jedoch kann es bei der *GridDistribution* passieren, dass durch deren nicht-deterministischen Ansatz für  $\epsilon = 0$  Klassifikationsfehler ungleich Null auftreten. Da für identische Objekte nicht exakt identische Merkmale produziert werden, ist es möglich, dass sich die Merkmale insoweit unterscheiden, dass das Objekt falsch klassifiziert wird, was jedoch für eine hohe Anzahl von Messungen  $S$  höchst unwahrscheinlich ist.

Die Mittelungsmerkmale hingegen liefern, dank ihrer deterministischen Berechnungsweise, Fehler, die exakt Null sind. In Abbildung 5.4 war zu erkennen, wie kleine  $m/V$ -Verhältnisse zu Merkmalsvektoren mit geringer Frequenz führen. Für solche Verhältnisse nimmt laut obiger Tabelle der Klassifikationsfehler ab, das heißt Merkmalsvektoren mit geringer Beweglichkeit sind offenbar nicht so störanfällig wie solche mit großer Beweglichkeit. Jedoch nimmt durch die geringere Frequenz auch die Korrelation benachbarter Merkmalskomponenten zu. Es wurden nur Merkmale mit Verhältnissen größer 2 betrachtet, da für kleinere Verhältnisse die Merkmale nahezu konstant werden.

Der Aufwand der Mittelungsmerkmale ist im Vergleich zur *GridDistribution* verschwindend gering, was mit der geringen Anzahl der Atome bzw. Verbindungen der Atome zusammenhängt. Der Berechnungsaufwand der Mittelungsmerkmale ist bekanntlich linear in der Anzahl der Knoten, die der *GridDistribution* logarithmisch in der Anzahl der Verbindungen.

## Symmetrie und Symmetriebrechung

Betrachten wir 'benzene' ( $C_6H_6$ ) bezüglich seiner Symmetrien. Für den zugrundeliegenden Graphen gibt es Graphenisomorphismen  $\sigma$ , die den Graphen in sich selbst überführen. Diese sind gerade die zyklischen Permutationen der CH-Gruppen. Den gleichen Effekt, also die Umindizierung der CH-Gruppen, kann auch durch Rotation des Gitters erreicht werden. Wenn  $m$  das Gitter zu 'benzene' ist, dann gibt es also eine Rotation  $\varepsilon$  und eine Umindizierung  $\sigma$ , so dass  $m = \varepsilon(\sigma(m))$ , wobei die Gleichheit hier als echte Gleichheit zu verstehen ist (auch die Indizierungen sind gleich).



Dies kann sofort auf die zugehörige Kodierung übertragen werden. Sei  $R' = \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix}$  die Drehmatrix zu  $\varepsilon$ , und  $\mathbf{P}$  die Permutationsmatrix zu  $\sigma$ , dann muss gelten:

$$\mathbf{H} = (\mathbf{P} \otimes R')^\dagger \mathbf{H} (\mathbf{P} \otimes R').$$

Die Matrix  $\mathbf{P} \otimes R'$  wirkt sich nicht auf den Startvektor  $\mathbf{v}_0$  aus, wenn er wie in (5.7) gewählt wurde, da  $R'$  sich nicht auf die isotropen Komponenten auswirkt, und alle isotropen Komponenten des Vektors gleich sind, weshalb auch  $\mathbf{P}$  keine Wirkung hat. Also findet man für die Vektoren  $\mathbf{v}_k = (\mathbf{P} \otimes R')^\dagger \mathbf{H}^k \mathbf{v}_0$  oder

$$\mathbf{v}_k = (\mathbf{P} \otimes R') \mathbf{v}_k.$$

Bei den Histogrammerkmalen sind gerade die isotropen Komponenten dieser Vektoren  $\mathbf{v}_k$  mit den einzelnen Zweigen zu assoziieren. Sei der Vektor  $\mathbf{v}_k^{iso}$  der  $n$ -komponentige Vektor, der aus den isotropen Komponenten des  $4n$ -komponentigen Vektors  $\mathbf{v}_k$  besteht. Da  $R'$  bekanntlich nicht auf diese wirkt, gilt:

$$\mathbf{v}_k^{iso} = \mathbf{P} \mathbf{v}_k^{iso}. \quad (5.8)$$

Was heißt dies also für das Merkmalshistogramm eines Gitters? Falls eine euklidische Bewegung  $\varepsilon$  existiert, ungleich der Identität, die das Gitter auf sich selbst abbildet, also  $m \simeq \varepsilon(m)$ , dann sind die isotropen Komponenten genau der Knoten gleich, die bei

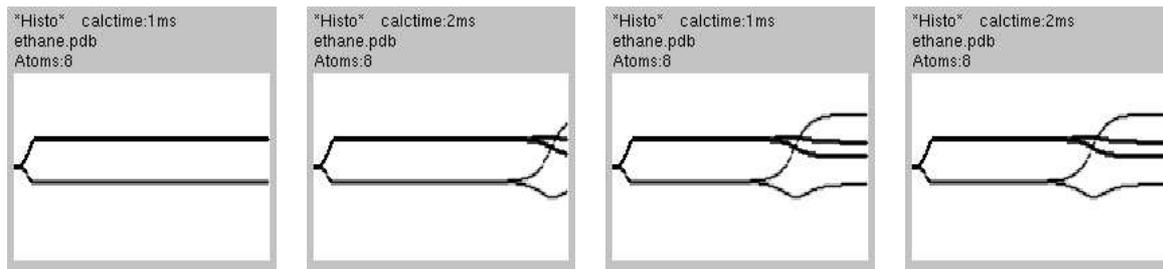
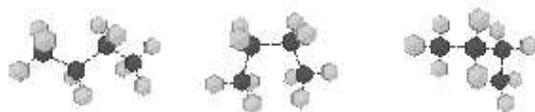


Abbildung 5.7: **Histogramme für 'sehr' kleine Störungen von Ethan.** 'Sehr' bedeutet hier, die Störungen sind lediglich durch numerische Ungenauigkeiten bei Rotationen entstanden. Die Störsensibilität kommt durch 'numerische Symmetriebrechung' zustande, die bei sehr symmetrischen Objekten auftritt.

der euklidischen Bewegung aufeinander geschoben werden. Für das oben betrachtete 'benzene' kann jedes  $H$ -Atom mit jedem anderen  $H$ -Atom durch eine Rotation um Vielfache von  $\pi/3$  in Deckung gebracht werden, somit müssen alle  $H$ -Atome dieselbe isotrope Komponente besitzen. Das Gleiche gilt für die  $C$ -Atome. Das Histogramm sollte also nur aus zwei Zweigen bestehen, der eine für die sechs  $H$ -Atome, der andere für die sechs  $C$ -Atome. Betrachtet man jedoch das Histogramm für 'benzene' in Abbildung 5.3, dann sind am Anfang noch zwei Zweige zu sehen, die sich dann aber plötzlich in vier Zweige aufspalten. Die Symmetriebedingung (5.8) wird also für große  $k$  zusehends durch numerische Ungenauigkeiten gebrochen. In Abbildung 5.7 sind Merkmals-histogramme von 'ethane' für 'sehr' kleine Störungen abgebildet. Theoretisch dürften hier genauso nur zwei Zweige auftreten, jedoch spalten diese wie beim 'benzene' wieder spontan auf. Der Aufspaltungszeitpunkt hängt dabei sehr sensitiv von kleinsten numerischen Ungenauigkeiten ab. Es hat sich aber gezeigt, dass sich solche numerische Aufspaltungen nur auf Histogramme und nicht auf die Mittelungsmerkmale auswirken. Die Summe über die isotropen Komponenten  $I_k = \sum_{i=1}^n (\mathbf{v}_k^{iso})_i$ , die bekanntlich als Mittelungsmerkmale eingeführt wurden, bleibt von solchen Aufspaltungen unbeeinflusst, das heißt es ändern sich lediglich die Summanden, die Summe an sich bleibt konstant.

### Drei Ausprägungen von Butan

Das Molekül Butan ( $C_4H_{10}$ ) tritt in der Natur in drei verschiedenen stabilen Formen auf. Die Formen unterscheiden sich jeweils in den Orientierungen der  $C$ -Atome zueinander, wie in der folgenden Abbildung zu erkennen ist.



Es hat sich herausgestellt, dass die vorgestellten Mittelungsmerkmale für alle drei Versionen des Moleküls identische Merkmale liefern. Man kann also behaupten, dass das Verfahren auf gewisse Weise die natürlichen physikalischen Gegebenheiten 'erkennt'.

### 5.3.2 Fische

Beschäftigen wir uns nun noch zum Schluss mit dem Fischdatensatz, der aus 60 Modellen besteht. Die Fischmodelle setzen sich in der Regel aus ca. 7000 Dreiecken zusammen, die sich aus unabhängigen Gittern für Körper und Flossen zusammensetzen. Die Mittelungsmerkmale  $I_k$  sind für zwei unzusammenhängende Gitter auf natürliche Weise durch die Summe der Merkmale der einzelnen Gitter gegeben. Solche Merkmale sind insbesondere invariant unter unabhängigen euklidischen Bewegungen der einzelnen Komponenten des Modells. Dies ist nicht unbedingt immer wünschenswert, da die relative Stellung der einzelnen Komponenten (Flossen und Körper) zueinander durchaus von Bedeutung ist. Nun gibt es natürlich die Möglichkeit, künstlich Nachbarschaftsverhältnisse zu schaffen, jedoch verlangt dies nach störungsrobusten (insbesondere lageinvarianten) effizienten Algorithmen, die zusätzlich Verbindungen in das Gitter einfügen. Für unseren speziellen Fischdatensatz wäre es möglich, Körper und Flossen getrennt zu vergleichen, was aber nur mit entsprechendem Vorwissen über die Ordnung der Flossen möglich ist. Im folgenden werden wir einfach über die Merkmale der einzelnen Komponenten der Fische summieren. Zum Vergleich wird nun die originale *ShapeDistribution* benutzt.

In Abbildung 5.8 sind Ergebnisse auf zwei Anfragen dargestellt. Im Vergleich sind die Ergebnisse jeweils mit den Mittelungsmerkmalen  $500/10^3$  und der *ShapeDistribution* gewonnen worden. Für die Anfrage nach 'SiameseTiger' liefert die *ShapeDistribution* drei von Gestalt und Ausdehnung ähnliche Fische, was ja zu erwarten war. Hingegen reagieren die Mittelungsmerkmale scheinbar sensibler auf die Form der Flossen der Fische, alle drei Anfrageergebnisse haben die gezackte Rückenflosse mit dem 'SiameseTiger' gemein und auch alle anderen Flossen sind in der Anzahl und Gestalt vorhanden. Man könnte also spekulieren, dass die Mittelungsmerkmale eher auf die Einzelheiten der Fische achten, während die *ShapeDistribution* eher auf den Körper und die Ausdehnung der Fische achtet.

In Tabelle 5.2 sind wieder, wie im vorigen Abschnitt, Klassifikationsfehler für verschiedene Störintensitäten festgehalten. Es ist wieder zu erkennen, dass die Mittelungsmerkmale für kleine  $m/V$ -Verhältnisse geringere Störanfälligkeit aufweisen. Jedoch konnten hier, im Gegensatz zu den Molekülen, auch kleinere Verhältnisse gewählt werden. Dies hängt mit der größeren Punktdichte bei den Fischen zusammen.

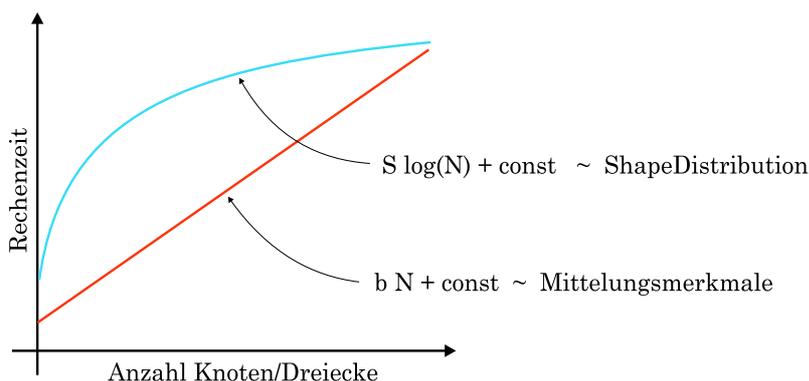
Die *ShapeDistribution* zeigt bei großen Messraten  $S = 10^6$  wieder gute Ergebnisse,

$\epsilon$	$10^3/1500$	$500/10^3$	$10^2/10^3$	Shape $10^4$	Shape $10^5$	Shape $10^6$
0	0%	0%	0%	30%	0%	0%
0.01	0%	0%	0%	40%	1%	0%
0.05	13%	4%	0%	40%	1%	1%
0.1	40%	30%	3%	40%	2%	1%
0.5	72%	61%	56%	70%	18%	12%
Zeit	0.2s	0.2s	0.2s	0.05s	0.5s	4s

Tabelle 5.2: **Klassifikationsfehler.**

jedoch sind die Ergebnisse bereits bei  $S = 10^5$ , im Gegensatz zu den Molekülexperimenten, akzeptabel. Dies ist durch die einfache Struktur der Fische im Vergleich zu den Molekülen zu erklären. Es genügen schon wenige Messungen, um die Struktur stabil zu repräsentieren.

Die Vorteile der Mittelungsmerkmale hinsichtlich des Aufwands kommen hier wegen der großen Anzahl der Knoten bzw. Dreiecken nicht mehr so sehr zum Tragen. Dies ist unmittelbar durch den logarithmischen Aufwand der *ShapeDistribution* im Vergleich zum linearen Aufwand bei den Mittelungsmerkmalen begründet:



Hierbei ist zu beachten, dass der Aufwand der Mittelungsmerkmale linear in der Anzahl der Knoten ist, der Aufwand der *ShapeDistribution* ist logarithmisch in der Anzahl der Dreiecke.

## 5.4 Kommentare

In den obigen Experimenten konnte die Anwendbarkeit der zu Beginn des Kapitels eingeführten Kodierung gezeigt werden. Im Rahmen dieser Experimente wurde eine *GUI*-Anwendung entwickelt, die Schnittstellen zu bekannten Formaten und Analysewerkzeugen zur Verfügung stellt. Neben den neu vorgestellten Merkmalen wurden auch die Methode der *ShapeDistribution* [1] in die Anwendung integriert und diese zum Vergleich benutzt.

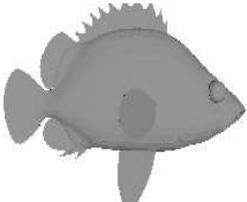
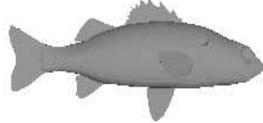
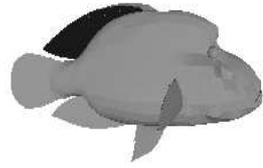
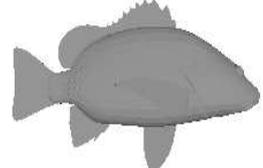
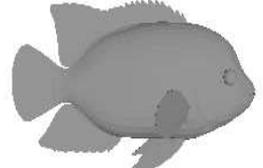
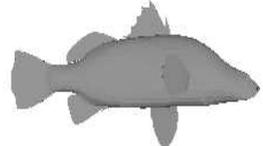
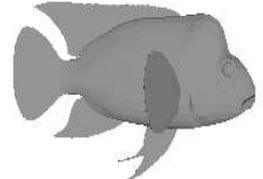
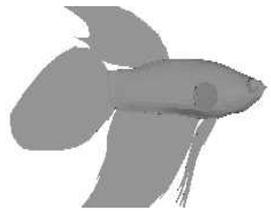
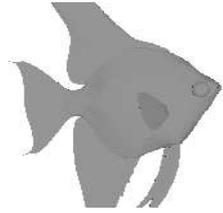
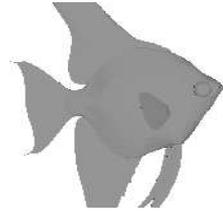
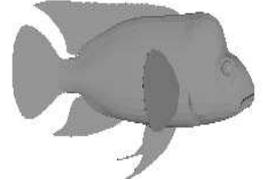
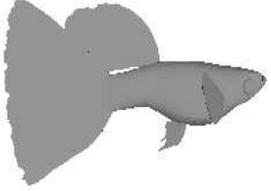
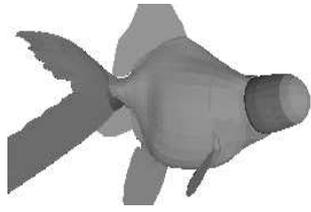
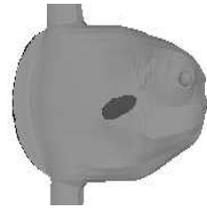
Anfrage	Mittelung	ShapeDistrib.
 SiameseTiger	 Suzuki	 Napoleon
	 MangroveJack	 Redhead
	 Barramundi	 Reddevil
 RedBetta	 Angel	 Angel
	 RedDevil	 BlueGrassGuppy
	 Demekin	 Manbou

Abbildung 5.8: **Anfrageergebnisse**. Es sind jeweils die drei ähnlichsten Fische zu den abgebildeten Anfragen angegeben. Die Ähnlichsten sind zu oberst.

Zu Beginn wurde an einem kleinem Moleküldatensatz das Verhalten der Mittelungsmerkmale demonstriert und eine leicht modifizierte Version der in [1] vorgestellten Methode, die *GridDistribution*, eingeführt. In den Störungsuntersuchungen wurden die beiden Methoden hinsichtlich ihrer Separationseigenschaften unter kleinen Störungen untersucht, wobei sich die *GridDistribution* als robuster erwiesen hat. Unter gleichen Störintensitäten zeigte sie deutlich geringere Zuordnungsfehler als die neu eingeführten Merkmale. Auf der anderen Seite konnten letztere hinsichtlich der Rechenzeit die *GridDistribution* klar dominieren, was unmittelbar mit der geringen Knotenzahl der verwendeten Moleküle zusammenhängt. Dieser Vorteil ging aber somit bei den Fischen, die mit weit mehr Knoten repräsentiert worden waren, wieder verloren. Bei diesen konnte zusätzlich beobachtet werden, dass für die *ShapeDistribution* weit weniger Messungen  $S$  ausreichen, um relativ stabile Merkmale zu produzieren, als für die bei den Molekülen verwendete *GridDistribution*. Zusammenfassend kann gesagt werden, dass die Anwendbarkeit der neu eingeführten Merkmale gezeigt werden konnte, diese aber von der Rauschempfindlichkeit der *ShapeDistribution* unterlegen sind. Durch den linearen Rechenaufwand hat sich jedoch die neue Methode für Objekte mit kleiner Knotenzahl als sehr gut anwendbar herausgestellt.

# Kapitel 6

## Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde eine neue Art der invarianten Merkmalsgewinnung für Drahtgittermodelle vorgestellt. Als erstes wurde in Kapitel 2 eine Übersicht über bestehende Ansätze geliefert, die sich nicht nur auf Drahtgittermodelle beschränken; es wurden auch Methoden für kontinuierliche Konturen in der Ebene, sphärische 3D-Modelle, Oberflächenmodelle und Volumenmodelle betrachtet. Durch physikalische Anschauungen motiviert, wurde im darauf folgenden Kapitel das Rahmenwerk zur Invariantengewinnung eingeführt. Mit einfachen Mitteln der Linearen Algebra wurde der Methodik eine theoretische Grundlage verschafft, die der konkreten Umsetzung in den Kapiteln 4 und 5 Richtlinien auferlegt hat. In Kapitel 4 wurde dann eine Anwendung der vorgestellten Methodik auf Polygonzüge in der Ebene vorgestellt. In diesem Kontext konnte der intuitiven Vorstellung von 'reichhaltigen' Merkmalen mittels der Konditionierungszahl der Merkmalsmatrix eine fundierte Grundlage gegeben werden. In den folgenden Experimenten wurde dem Leser ein Gefühl für das Verhalten der Merkmale vermittelt und die Merkmale unter lokalen Störungen untersucht. Es zeigte sich dabei jedoch, dass unter den gegebenen Bedingungen der Merkmalsbildungsprozess nicht invertierbar ist, was durch die schnelle Konvergenz der Merkmalsvektoren begründet werden konnte. In Kapitel 5 wurde dann der Ansatz auf allgemeine dreidimensionale Drahtgittermodelle angewendet. Es wurde kurz das in diesem Rahmen entwickelte Analysewerkzeug vorgestellt und in den darauf folgenden Experimenten benutzt. Als Referenzmethode wurden die schon zu Beginn vorgestellten *ShapeDistribution* verwendet. In den Störungsuntersuchungen wurde gezeigt, dass die vorgestellten Merkmale sich störsensibler als die Referenzmethode verhalten, jedoch sind durch den deterministischen Ansatz exaktere Vergleiche als mit der *ShapeDistribution* möglich. In anschließenden Symmetriebetrachtungen wurde die Anfälligkeit der Merkmale bei sehr symmetrischen Objekten gezeigt und anhand der Histogramme ein Gefühl für die Funktionsweise der Merkmale vermittelt. Für das Molekül Butan, welches in der Natur in drei verschiedenen Formen vorkommt, wurde festgestellt, dass die verwendete

Kodierung für alle drei identische Merkmale produziert.

Es existieren sicher noch eine Reihe interessanter Fragen, die es wert sind, untersucht zu werden. Die folgenden Punkte erscheinen dem Autor für weitere Untersuchungen am interessantesten:

- **Eindeutigkeit von (anti-)hermiteschen Kodierungen.** In den Vollständigkeitsbetrachtungen in Kapitel 4 konnte lediglich die Eindeutigkeit einer asymmetrischen Kodierung gezeigt werden. Der weit interessantere Fall ist aber die Eindeutigkeit einer (anti-)hermiteschen oder unitären Kodierung, da in diesem Fall nur linear in  $n$  viele Merkmale erzeugt werden müssen.
- **Vollständigkeitsbetrachtungen für 3D-Kodierung.** Es wurden nur die Polygonmerkmale auf Vollständigkeit untersucht. Inwieweit ist mit der Kodierung aus Kapitel 5 Eindeutigkeit zu erreichen? Welche Eigenschaften der Kodierung sind für die identischen Merkmale der drei 'butane'-Ausprägungen verantwortlich? Auf welche Gitter muss man sich einschränken, damit die Kodierung eindeutig wird?
- **3Distanzkodierung im 3D-Raum.** Polygonzüge in der Ebene konnten mit Distanzen zwischen den Nachbarn beschrieben werden. Gibt es auch die Möglichkeit Oberflächen im 3D-Raum ausschließlich mit Distanzinformation zu beschreiben? Wie müssten solche speziellen 'Oberflächengitter' und deren zugrundeliegenden Graphen aussehen?
- **Verhalten beim Einfügen von Knoten.** Bei Experimenten mit der unitären Kodierung in Kapitel 4 wurde mit dem Parameter  $\sigma_n$  (siehe (4.6)) versucht, Polygonzüge mit verschiedenen Knotenzahlen vergleichbar zu machen. Jedoch wurde  $\sigma_n$  rein empirisch gewonnen und es ist nicht klar, wie die Wahl von  $\sigma_n$  theoretisch zu rechtfertigen ist. Es ist nicht einmal gesichert, ob es in diesem Rahmen überhaupt 'sinnvolle' Ähnlichkeitsmaße für Polygonzüge mit verschiedener Anzahl von Knoten gibt.
- **Kodierungen, die reguläre Merkmalsmatrizen erzeugen.** Das eigentliche Manko der vorgestellten Merkmale ist, dass die Reichhaltigkeit der Merkmale stark von der Konvergenzgeschwindigkeit der Merkmalsvektoren  $\mathbf{v}_k$  abhängt. Es sollte also nach Kodierungen gesucht werden, die in diesem Sinne ein 'gutes', also langsames Konvergenzverhalten aufweisen. In der Zusammenfassung in Kapitel 4 wurde schon eine Möglichkeit skizziert, wie mit Hilfe von parametrischen Kodierungen den Merkmalen mehr Reichhaltigkeit gegeben werden kann. Genauso könnten solche Parameter benutzt werden, um das Konvergenzverhalten zu stören bzw. zu verlangsamen.

- **Anwendung der Merkmale in Verbindung mit Klassifikatoren.** In den Störungsuntersuchungen wurde ein primitiver Nächste-Nachbar-Klassifikator benutzt, um einen Klassifikationsfehler zu berechnen, der dazu diente, die Merkmale auf deren Störempfindlichkeit zu untersuchen. Es liegt nahe, solche Untersuchungen mit höher entwickelten Klassifikatoren, wie *Support-Vector* Maschinen oder Polynomklassifikatoren [8], zu unternehmen. Solche Klassifikatoren bieten zusätzlich die Möglichkeit, die Merkmale empirisch auf ihre Separationsfähigkeit zu untersuchen. Man könnte zum Beispiel versuchen festzustellen, wie sich einzelne Objekte im Merkmalsraum überlagern, welche Objekte sich besonders ähnlich sind usw..
- **Erweiterung auf attributierte Gitter.** Die vorgestellte Methodik könnte auf mehrere Art und Weisen auf attributierte Gitter erweitert werden. Nimmt man Bezug auf die Kodierung in Kapitel 5, so wäre es ein leichtes, die Elementbezeichnungen der Atome als Atommassen in den Parameter  $m$  der Kodierung einfließen zu lassen. Statt den Parameter  $m$  fest und für alle Knoten gleich zu wählen, nimmt er in Abhängigkeit des Elementes verschiedene Werte an. Eine andere Möglichkeit wäre die Erweiterung der Kodierung auf mehr als drei Dimensionen. Die zusätzlichen Dimensionen könnten dann als kontinuierliche Attribute der Knoten angesehen werden.
- **Anwendbarkeit der Merkmalshistogramme.** Die Merkmalshistogramme wurden bis jetzt nur zur Veranschaulichung benutzt. Inwieweit macht es Sinn, solche Histogramme zum Vergleich zu benutzen?
- **Anwendbarkeit auf Oberflächenmodelle.** Da die Merkmale stark von der zugrundeliegenden Triangulierung der Oberflächenmodelle abhängen, diese jedoch im allgemeinen nicht für die 'Bedeutung' der Oberfläche von belang sind, bietet es sich an, vorab eine Normalisierung des Gitters vorzunehmen. Für die Normalisierung von Oberflächenmodellen gibt es in [12] einen interessanten Ansatz. Eine andere Möglichkeit wäre, mittels Finite Elementen zu versuchen, die Merkmale unabhängiger von den zugrundeliegenden Knoten zu machen. Hier gibt es bereits Methoden [28], die auf ähnliche Weise versuchen, Korrespondenzen in Bildern zu finden.
- **Andere Ähnlichkeitsmaße.** In den Experimenten wurde nur die euklidische Distanz, die auf der 2-Norm basiert, zwischen den Merkmalsvektoren als Ähnlichkeitsmaß benutzt. Welche anderen Normen (1-Norm,  $\infty$ -Norm) machen noch Sinn, als Ähnlichkeitsmaß benutzt zu werden?

- **Verhalten unter globalen Störungen.** In den Störungsuntersuchungen wurde nur das Verhalten unter lokalen Störungen betrachtet, das heißt die Verschiebungen der einzelnen Knoten waren unkorreliert. Genauso könnten auch globale Störungen betrachtet werden, zum Beispiel kleine Parallelverschiebungen, Torsionen oder ähnliches.

Es gibt also noch eine Menge Fragen und Probleme, die gelöst werden müssen, bevor die vorgestellten Merkmale eine echte Anwendungen finden können. Es erscheint vor allem die Erweiterung auf attributierte Gitter '*straight forward*'. Für Vollständigkeit im 3D könnte die Einschränkung auf spezielle normierte Gitter erfolgsversprechend sein.



# Anhang A

## Symbole

Symbol	Erklärung
$\mathbb{N}$	Die natürlichen Zahlen.
$\mathbb{C}$	Der komplexe Zahlenkörper.
$\mathbb{R}$	Der reelle Zahlenkörper.
$\mathbb{K}$	Ein Körper.
$\mathbb{G}$	Die Menge aller Gitter.
$\mathbf{I}_n$	Die Identitätsmatrix. $\mathbf{I}_n \in \text{Mat}(n, \mathbb{C})$ .
$\mathbf{U}$	Eine unitäre Matrix $\mathbf{U} \in U(n)$ .
$\tilde{\mathbf{U}}$	Eine unitäre Matrix $\tilde{\mathbf{U}} \in U_{\mathbf{v}}(n)$ , die den Vektor $\mathbf{v}_0$ fest lässt.
$\mathbf{U}_\varepsilon$	Durch die euklidische Bewegung $\varepsilon$ induzierte unitäre Transformation (Seite 19).
$\mathbf{P}_\sigma$	Die Permutationsmatrix zu $\sigma$ (Seite 19).
$\{\mathbf{e}_k\}$	Die Menge der Basisvektoren $\mathbf{e}_k$ , mit $\mathbf{e}_k^\dagger \mathbf{e}_j = \delta_{kj}$ . Wenn nicht weiter spezifiziert, handelt es sich um die Standardbasis.
$\mathbf{v}$	Ein Knotenvektor, die Komponenten sind mit Gitterknoten zu assoziieren.
$\{v_i\}$	Ein Knotenvektor in Komponentenschreibweise. $\mathbf{v} = \{v_i\}$ .
$\{c_{ij}\}$	Eine Matrix in Komponentenschreibweise. $\mathbf{C} = \{c_{ij}\} = \{\mathbf{e}_i^\dagger \mathbf{C} \mathbf{e}_j\}$ .
$\delta_{ij}$	Das Kroneckerdelta. Falls $i = j$ ergibt sich 1, sonst 0.
$(\mathbf{v})_i$	Die $i$ -te Komponente des Vektors $\mathbf{v}$ in der Standarddarstellung. $(\mathbf{v})_i = \mathbf{e}_i^\dagger \mathbf{v}$ , wobei $\mathbf{e}_i$ der $i$ -te Basisvektor in Standarddarstellung.
$\vec{a}$	Ein Raumvektor $\vec{a} \in \mathbb{R}^d$ , die Komponenten sind mit Raumrichtungen zu assoziieren.
$\vec{v}$	Ein Vektor aus dem Produktraum von Knoten- und geometrischem Raum, $\vec{v} \in \mathbb{R}^n \otimes \mathbb{R}^d$ .
$m$	Ein Gitter $m \in \mathbb{G}$ (Seite 15).
$\mathbf{H}(m)$	Bezeichnet die Kodierung eines Gitters $m$ (Seite 18).
$I_k(m)$	Bezeichnet die Merkmale eines Gitters $m$ (Seite 21).
$I_{\lambda k}(m)$	Bezeichnet die Histogrammmerkmale eines Gitters $m$ (Seite 21).
$\mathbf{C}(m)$	Die Merkmalsmatrix Gitters $m$ (Seite 20).
$\mathbf{Diag}(\mathbf{z})$	Eine Diagonalmatrix mit den Komponenten des Vektors $\mathbf{z}$ auf der Diagonale. $\mathbf{e}_k^\dagger \mathbf{Diag}(\mathbf{z}) \mathbf{e}_j = \delta_{kj} z_k$ .
$\text{span}(\mathbf{v}_0, \dots, \mathbf{v}_k)$	Der durch die Vektoren $\mathbf{v}_k$ aufgespannte Raum.
$\mathbf{1}$	Der Einsvektor $\mathbf{e}_k^\dagger \mathbf{1} = 1$ .

Symbol	Erklärung
$\mathbf{i}$	Die imaginäre Einheit.
$\text{Re}\{\}$	Der Realteil einer komplexen Zahl.
$\text{Im}\{\}$	Der Imaginärteil einer komplexen Zahl.
$\otimes$	Das Kroneckerprodukt.
$\ \cdot\ $	die 2-Norm, $\ \mathbf{v}\  = \sqrt{\mathbf{v}^\dagger \mathbf{v}}$ .
$\simeq$	Gittergleichheit unter Umindizierung der Knoten (Seite 15).
$\sim$	Gittergleichheit unter Umindizierung der Knoten und euklidischer Bewegung (Seite 15).
$\Delta$	Der Laplace-Operator. $\Delta f = \frac{d^2 f}{dx_1^2} + \frac{d^2 f}{dx_2^2} + \frac{d^2 f}{dx_3^2}$ .
$r(\lambda)$	Die Rechteckfunktion (Seite 22).
$[\vec{a}]_\times$	Die Kreuzproduktmatrix mit $\vec{a} \times \vec{b} = [\vec{a}]_\times \vec{b}$ .
$d_{ij}$	Der euklidische Abstand $d_{ik} = \ \vec{x}_i - \vec{x}_j\ $ zweier Knoten.
$D(m, m')$	Die euklidische Distanz der Merkmalsvektoren zweier Gitter.
$F$	Der 'leave one out'-Klassifikationsfehler (Seite 44).
$\sigma_n$	Skalierungsfaktor für unitäre Kodierung (Seite 39).

# Anhang B

## Beschreibung der Implementation

### B.1 Benutzte Pakete und Umgebungen

Die in dieser Arbeit entwickelten Applikation wurde in C++ in einer Linux-Umgebung entwickelt. Im Rahmen der GUI-Entwicklung wurden folgende Bibliotheken benutzt:

- **Trolltech QT-Bibliotheken.** Grundlegendes Windowing-Framework. Weitere Informationen sind unter [www.trolltech.com](http://www.trolltech.com) zu finden.
- **OpenGL.** Grafische Darstellung der 3D Objekte. Die OpenGL Homepage ist unter [www.opengl.org](http://www.opengl.org) zu finden.
- **Lib3ds.** Importieren des 3DS Datei Format. Eine offene Bibliothek unter GNU-Lizenz, verfügbar unter [lib3ds.sourceforge.net](http://lib3ds.sourceforge.net). Für die Beschreibung des Formats siehe [www.the-labs.com/Blender/3DS-details.html](http://www.the-labs.com/Blender/3DS-details.html).
- **Pdb2Vrml.** Importieren des PDB Datei Format. Ein offenes Konvertierwerkzeug unter GNU-Lizenz, verfügbar unter [www.geocities.com/gnubioq/pdb2vrml](http://www.geocities.com/gnubioq/pdb2vrml). Für die Beschreibung des Formats siehe [www.rcsb.org/pdb](http://www.rcsb.org/pdb).

In Abbildung B.1 ist beispielhaft ein *Screenshot* der entwickelten Anwendung dargestellt. Das Klassendiagramm in Abbildung B.2 fasst die wichtigsten Klassen der Anwendung zusammen. Für eine einheitliche Behandlung wurden die zwei Klassen *Invariante* und *Item* eingeführt, die als Mutterklassen für die Invariantenberechnung und Objekt-handhabung dienen. Die Klasse *Invariante* wird zur Darstellung der Invarianten und zur Vermittlung der Berechnungsparameter zwischen Benutzer und der eigentlichen Invariantenberechnung benutzt. Die Klasse *Item* ist für die verschiedenen geometrischen Objektmanipulationen und für die Darstellung der beiden Objekttypen *Polymesh* und *Molecule* verantwortlich.

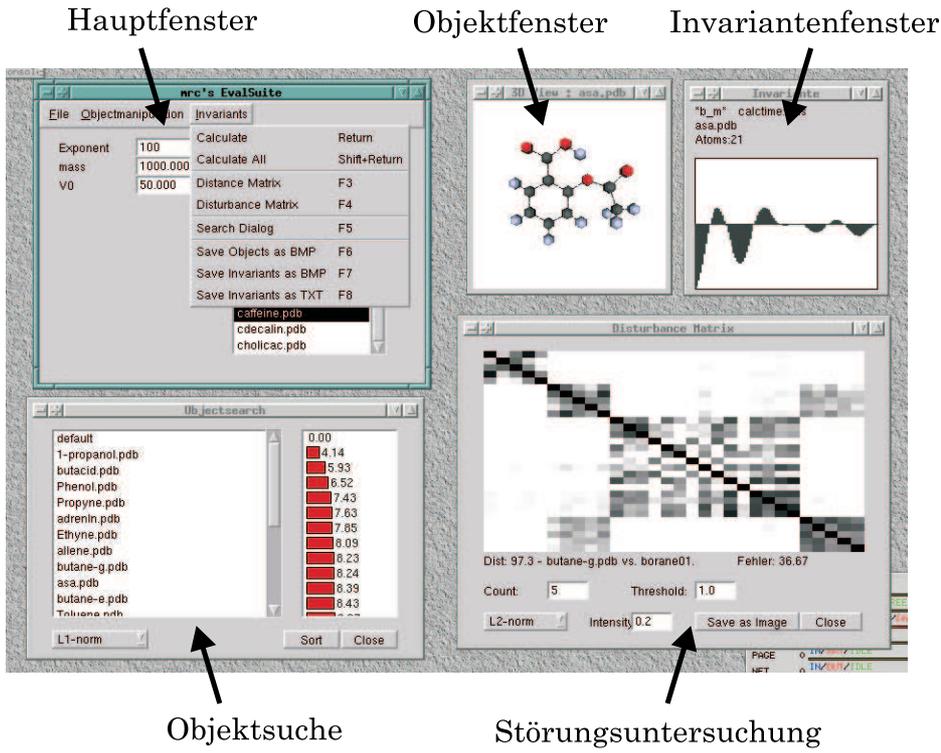


Abbildung B.1: Screenshot.

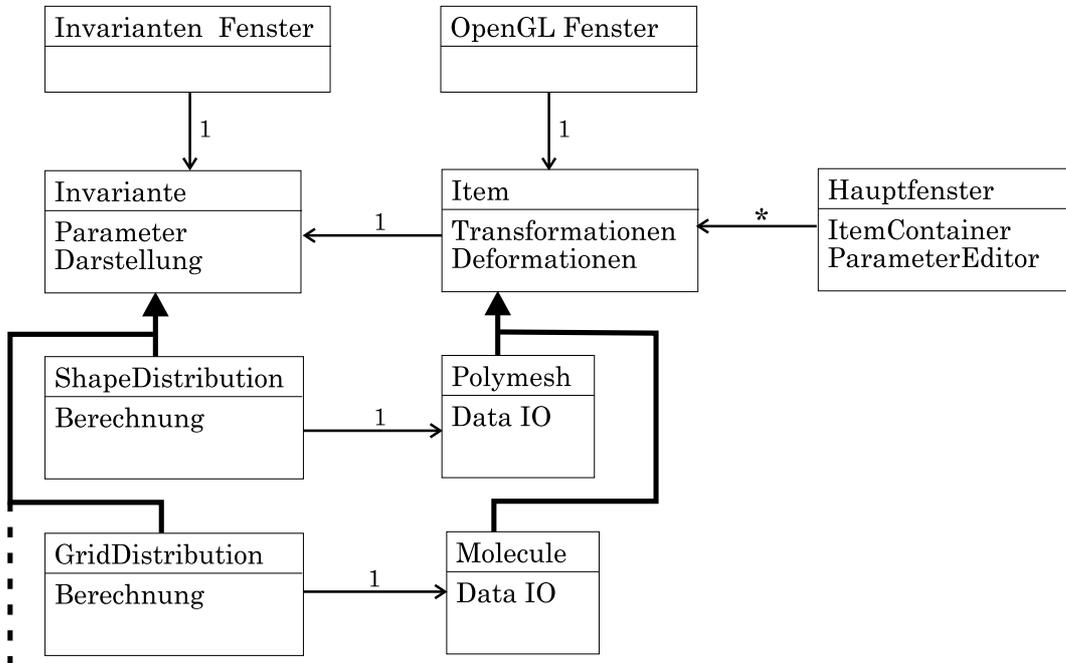


Abbildung B.2: Klassendiagramm.

# Anhang C

## Moleküle

2me-3ane 	b-10ane 	b-11ane 	borane 	butane-a 	butane-e 
butane-g 	cyclo-5ane 	cyclo-6ane 	d-7ane 	decane 	ethane 
heptane 	hexane 	methane 	nonane 	octane 	pentane 
propane 	adrenalin 	caffeine 	lsd 	nicotine 	valium 
water 	1-propanol 	Benzene 	Ethyene 	Phenol 	Propyne 

Abbildung C.1: Moleküle

toluene 	allene 	ammoniak 	asa 	butacid 	cdecalin 
cholicac 	ethactetate 	ethacid 	ethanol 	i-cyclo-5ane 	kekulene 
menthol 	methacid 	methanol 	piperine 	propacid 	q-cyclene 
quinine 	r2bu-ts 	rr-tacid 	rs-tacid 	s34ane 	srdimecp 
ssdimecp 					

Abbildung C.2: Moleküle.

# Literaturverzeichnis

- [1] Michael Kazhdan, Thomas Funkhouser und Szymon Rusinkiewicz: *Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors*, Department of Computer Science, Princeton University, Princeton NJ, 2003
- [2] Olaf Ronneberger, Hans Burkhardt, Eckart Schultz: *General-purpose Object Recognition in 3D Volume Data Sets using Gray-Scale Invariants - Classification of Airborne Pollen-Grains Recorded with a Confocal Laser Scanner Microscope*, Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, 2002
- [3] Titus Zaharia, Françoise Preteux: *Shape-based retrieval of 3D mesh models*, Institute National des Telecommunications, Unite de Projets ARTEMIS, 2002
- [4] Robert Osada, Thomas Funkhouser, Bernard Chazelle und David Dobkin: *Matching 3D Models with Shape Distribution*, Department of Computer Science, Princeton University, Princeton NJ, 2001
- [5] Remco Veltkamp, Hans Burkhardt, Hans-Peter Kriegel: *State-of-the-Art in Content-Based Image and Video-Retrieval*, Kluwer Academic Press Publishers, 2001
- [6] Remco C. Veltkamp: *Shape Matching: Similarity Measures and Algorithms*, Department of Computer Science, Utrecht University, Netherlands, 2001
- [7] Marcin Novotni and Reinhard Klein: *A Geometric Approach to 3D Object Comparison*, Computer Graphics Group, University of Bonn, 2001
- [8] R. Duda, P. Hart, D. Stork *Pattern Classification*, Wiley-Interscience, 2nd Edition, 2000
- [9] Brian C. Hall: *An Elementary Introduction to Groups and Representations*, University of Notre Dame, Departement of Mathematics, Notre Dame IN USA, 2000

- [10] Xiong Wang und Jason T.L. Wang: *Fast Similarity Search in Three-Dimensional Structure Databases* Department of Computer Science, New Jersey Institute of Technology, Newark NewJersey, 1999
- [11] S. Theodoridis, K. Koutroubas *Pattern Recognition*, Academic Press, 1999
- [12] I. Guskov, K. Vidimce, W. Sweldens, P. Schröder *Normal Meshes*, Siggraph 2000, Computer Graphics Proceedings, 2000
- [13] I. Bronstein, K. Semendjadew, G. Musiol, H. Mühlig *Taschenbuch der Mathematik*, 4.Auflage Verlag Harri Deutsch, Frankfurt a. Main, Thun, 1999
- [14] Sven Siggelkow, Hans Burkhardt: *Image Retrieval based on local invariant features* Proceedings of the IASTED International Conference Signal and Image Processing, Las Vegas, Nevada, USA, 1998
- [15] Nikolaos Canterakis: *Fast 3D Zernike Moments and Invariants*, Lehrstuhl für Mustererkennung, Albert-Ludwig-Universität Freiburg, Interner Bericht 5/97, 1997
- [16] R.B. Lehouen, D.C. Sorensen, C. Yang: *ARPACK Users' Guide: Solution of Large Scale EigenValue Problem with Implicitly Restarted Arnoldi Method*, 1997
- [17] Patrick J. Flynn, Richard J. Campbell: *A WWW-accessible Database for 3D Vision Research*, School of Electrical Engineering and Computer Science, Washington State University, 1997, <http://www.eecs.wsu.edu/flynn/3DDB>
- [18] F. Schwabl: *Quantenmechanik für Fortgeschrittene*, Springer Verlag, Berlin, Heidelberg, 1997
- [19] Bernd Jähne *Digitale Bildverarbeitung*, 4.Auflage, Springer Verlag, 1997
- [20] G. Fischer: *Lineare Algebra*, 11.Auflage, Vieweg Verlag, 1997
- [21] Lydia E. Kavradi: *Geometry and Discovery of New Ligands* Stanford University, Stanford, USA, 1996
- [22] Nikolaos Canterakis: *Complete Moment Invariants and Pose Determination for Orthogonal Transformations of 3D Objects*, Technische Informatik, TU Hamburg Harburg, 1996
- [23] Jürgen Schürmann: *Pattern Classification*, Daimler-Benz ResearchCenter Ulm, John Wiley and Sons Inc, 1996

- [24] Matrin Juvan, Bojan Mohar, Janez Zerovnik: *Distance-related invariants of polygraphs*, Department of Mathematics, University of Ljubljana, Slovenia, 1996
- [25] Bojan Mohar: *Some Applications of Laplace Eigenvalues of Graphs*, Department of Mathematics, University of Ljubljana, Slovenia, 1995
- [26] Hanns Schulz-Mirbach: *Anwendungen von Invarianzprinzipien zur Merkmalsgewinnung in der Mustererkennung*, Erschienen als Fortschrittsbericht (Reihe 10, Nr. 372) der VDI-Zeitschriften, VDI Verlag, Düsseldorf, 1995
- [27] David Parsons, John Canny: *Geometric Problems in Molecular Biology and Robotics*, University of California, Berkeley, 1994
- [28] Stan Scarloff und Alex Pentland: *Modal Matching for Correspondence and Recognition*, Perceptual Computing Section, The Media Laboratory, Massachusetts Institute of Technology, 1993
- [29] Chandrajit Bajaj, Fausto Bernadini, Kokichi Sugihara: *A Geometric Approach to Molecular Docking and Similiarity*, Department of Computer Science, Purdue University, West Lafayette, 1993
- [30] W. Nolting: *Grundkurs theoretische Physik, Quantenmechanik, Teil 1: Grundlagen*, Verlag Zimmermann-Neufang, 1992
- [31] Rafael C. Gonzalez, Richard E. Woods: *Digital Image Processing*, Addison-Wesley Pub. Company, 1992
- [32] H. R. Schwarz: *Numerische Mathematik*, Teubner Stuttgart, 1988
- [33] Felix R. Gantmacher: *Matrizentheorie*, Springer Verlag, 1986
- [34] R. Zurmühl, S.Falk: *Matrizen und ihre Anwendungen*, Springer Verlag, 1984
- [35] H. Burkhardt: *Transformationen zur lageinvarianten Merkmalsgewinnung*, Habilitationsschrift, Universität Karlsruhe, 1979, Erschienen als Fortschrittsbericht (Reihe 10, Nr. 7) der VDI-Zeitschriften, VDI Verlag