Institut für
Neuroinformatik

Ruhr-Universität
Bochum

Internal Report 96–09

Lernen lokaler Objektmerkmale mit Bananenwavelets

by

Gabriele Peters

Ruhr-Universität Bochum
Institut für Neuroinformatik
44780 Bochum

IR-INI 96–09
August 1996
ISSN 0943-2752

Fakultät für Mathematik
Ruhr-Universität Bochum

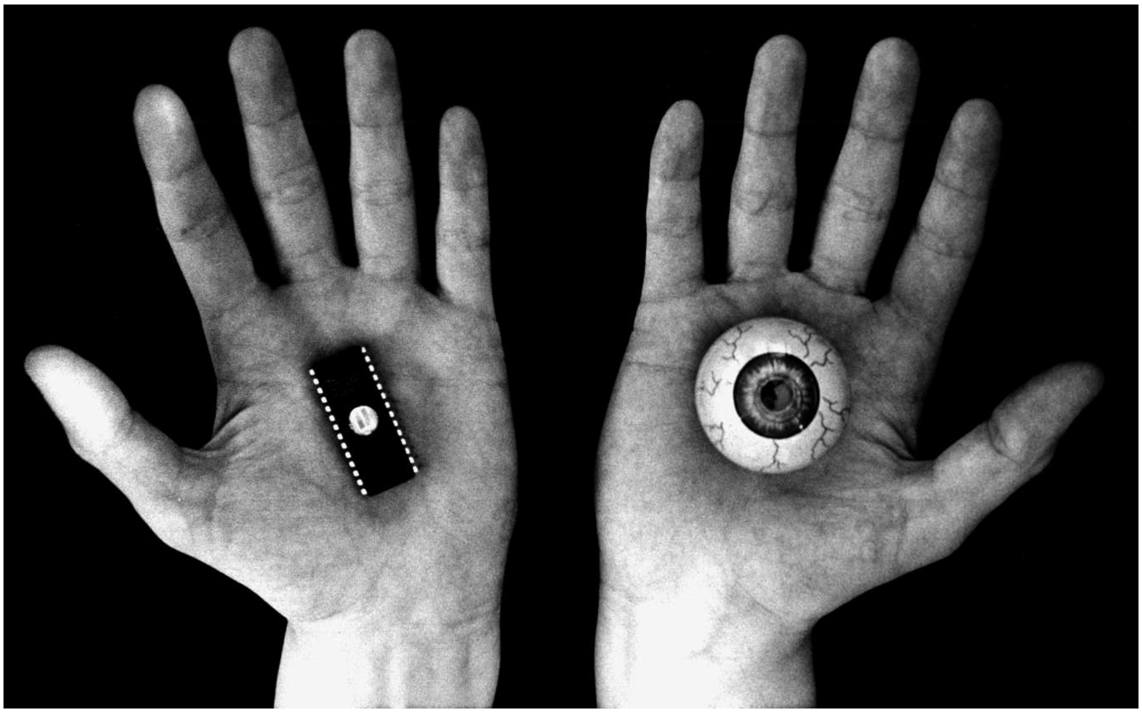
Lernen lokaler Objektmerkmale mit Bananenwavelets

Diplomarbeit

Gabriele Peters

August 1996

Diese Arbeit wurde durchgeführt an der Fakultät für Mathematik und am Institut für Neuroinformatik der Ruhr-Universität Bochum. Sie wurde betreut von Prof. Dr. Albeverio (Fakultät für Mathematik) und Prof. Dr. von der Malsburg (Institut für Neuroinformatik). Die Grundidee, für die Bildvorverarbeitung gekrümmte Wavelets zu verwenden, stammt von Norbert Krüger, der als Betreuer im Verlauf der Arbeit ständig für Diskussionen und Anregungen zur Verfügung stand. Eine große Hilfe waren mir ebenfalls die freundliche Atmosphäre und die Unterstützung durch die Gruppe am Institut für Neuroinformatik. Ihnen allen möchte ich an dieser Stelle herzlich danken.



Computervision

Inhaltsverzeichnis

1 Einleitung 1

2 Problemstellung 3
- 2.1 Objektklassenrepräsentation durch Bunchgraphen 3
- 2.2 Diskussion der Repräsentation durch Bunchgraphen 7

3 Lösungsvorschlag 10
- 3.1 Lernkonzept 10
- 3.2 Diskussion des Lernkonzepts 12
  - 3.2.1 Bildvorverarbeitung 12
  - 3.2.2 Erzeugung einer Objektklassenrepräsentation 13
- 3.3 Neurobiologische und psychologische Hintergründe 16
  - 3.3.1 Kortikale Bildvorverarbeitung 16
  - 3.3.2 Höhere kortikale Verarbeitungsstufen 19

4 Bananenwavelets 23
- 4.1 Bananenwavelets als Verallgemeinerung von Gaborwavelets 23
- 4.2 Stauden von Bananenwavelets 27
- 4.3 Staudenstamm 29
- 4.4 Implementierung von Bananenwavelets 29
  - 4.4.1 Wahl der charakteristischen Parameter einer Staude 29
  - 4.4.2 Unabhängigkeit von Variationen in der Bildhelligkeit 30
- 4.5 Bananentransformation 31
  - 4.5.1 Bildtransformation mit Bananenwavelets 31
  - 4.5.2 Antwortverhalten von Bananenwavelets bei der Anwendung auf natürliche Bilder 31
- 4.6 Implementierung der Bananentransformation 34

5 Lernen lokaler Merkmale 37
- 5.1 Überblick 37
- 5.2 Durchführung und Ergebnis der Bananentransformation 39
- 5.3 Extraktion der signifikanten Merkmale eines Trainingsbildes 44
  - 5.3.1 Signifikante Merkmale eines Trainingsbildes 44
  - 5.3.2 Algorithmische Realisierung 45
- 5.4 Lernen aus Beispielen 45
  - 5.4.1 Zusammenfassung der signifikanten Merkmale einzelner Trainingsbilder 46
  - 5.4.2 Clusteranalyse der signifikanten Merkmale 47

5.4.3 Gelernte Repräsentation einer Objektklasse 51

6 Experimentelle Ergebnisse 53
6.1 Getränkedosen 54
6.2 Gesichter 54
6.3 Weitere Simulationen 60

7 Schlußfolgerungen 62

8 Ausblick 63

Kapitel 1

Einleitung

Die Neuroinformatik macht sich Kenntnisse über die Funktionsweise von Gehirnen verschiedener Spezies zunutze, um mit Methoden von Disziplinen wie der Informatik, der Mathematik oder der Physik Modelle für Teilfunktionen des Gehirns zu erstellen und diese in technischen Systemen nachzuvollziehen.
Ein Teilproblem der Neuroinformatik, dem sehr großes Gewicht beigemessen wird, ist die Modellierung sensorischer Fähigkeiten, und hier spielt insbesondere das visuelle System eine wichtige Rolle. Damit ein Organismus angemessen auf seine Umwelt reagieren kann, ist es unerläßlich, daß in seinem Gehirn eine Gliederung der sensorischen Eingangsinformation vorgenommen wird, die dazu führt, daß er verschiedene Objekte voneinander unterscheiden kann. Aus Erkenntnissen insbesondere der Psychologie und der Biologie weiß man, wie gut z.B. das visuelle System des Menschen in der Lage ist, Objekte unter verschiedenen Bedingungen - seien es unterschiedliche Beleuchtungen, verschiedene Entfernungen, wechselnde Hintergründe oder auch teilweise Verdeckungen - wiederzuerkennen. Obwohl durch all diese unterschiedlichen Bedingungen ein völlig anderes Bild desselben Objektes auf die Netzhaut des Auges projiziert werden kann, gelingt es dem Gehirn, diese Unterschiede auszublenden und das Objekt zu erkennen.
In dieser Diplomarbeit wird ein Modell visueller Informationsverarbeitung vorgestellt, das die eben genannten Aspekte berücksichtigt. Dieses Modell beschreibt das Lernen lokaler Objektmerkmale anhand von dargebotenen Beispielbildern verschiedener Objekte derselben Klasse. Das Ziel des Lernens ist die Beschreibung einer Objektklassenansicht durch eine zweidimensionale Repräsentation der Klasse. Diese Repräsentation soll wichtige, lokale Merkmale von Objekten dieser Klasse in sich vereinigen. In dem Modell, das im folgenden vorgestellt wird, werden die gerade beschriebenen, variierenden Bedingungen, unter denen die Objekte aufgenommen werden können (insbesondere variierende Hintergründe oder Beleuchtungen, die verschiedene Reflexe oder Schatten auf den Objekten hervorrufen), nur noch einen geringen Einfluß auf die zu lernende Repräsentation der Objektklasse haben.

Das Lernen lokaler Objekteigenschaften im Rahmen dieser Arbeit läßt sich grob gliedern in die Bildvorverarbeitung mit sogenannten *Bananenwavelets* - einer hier erstmalig vorgestellten Verallgemeinerung von Gaborwavelets - und den Lernalgorithmus an sich.
In Kapitel 2 wird zunächst die Problemstellung präzisiert. Insbesondere werden dort das am Institut für Neuroinformatik verwendete Verfahren der Objekterkennung beschrieben (Abschnitt 2.1) und einige Probleme diskutiert (Abschnitt 2.2). In Kapitel 3 wird ein Lösungsvorschlag entwickelt, der die aufgezeigten Probleme beseitigen soll. In Abschnitt

3.1 wird zuerst das Konzept, das dem Lernalgorithmus zugrunde liegt, vorgestellt, um dann in Abschnitt 3.2 in konkreter Anwendung auf das hier vorliegende Problem ausführlich diskutiert zu werden. Eine Vertiefung der neurobiologischen und psychologischen Befunde, an die sich die Modellbildung dieser Arbeit anlehnt, bildet mit Abschnitt 3.3 den Abschluß von Kapitel 3. Kapitel 4 befaßt sich mit der Bildvorverarbeitung. Hier werden die Bananenwavelets definiert und analysiert (Abschnitte 4.1 bis 4.4) und die Bildtransformation mit Bananenwavelets erläutert (Abschnitte 4.5 und 4.6). In Kapitel 5 wird der Algorithmus beschrieben, mit dem die lokalen Objektmerkmale gelernt werden. Kapitel 4 und 5 bilden also die mathematische Formulierung des vorgestellten Modells. Das Modell wird anhand zweier Objektklassen in Computersimulationen ausgeführt. Deren Ergebnisse findet man in Kapitel 6. Nach den Schlußfolgerungen in Kapitel 7 werden schließlich im Ausblick (Kapitel 8) Anwendungs- und Verbesserungsmöglichkeiten angeregt.

Kapitel 2

Problemstellung

Um ein Objekt in einer Szene zu erkennen, muß ein Informationsverarbeitungssystem - sei es nun das Zentrale Nervensystem eines Lebewesens oder ein Computer - zunächst eine „Vorstellung“, ein „Konzept“, von diesem Objekt besitzen, d.h. es muß eine wie auch immer geartete Definition oder eine Beschreibung der Klasse, der das Objekt angehört, verinnerlicht haben. Diese Beschreibung kann als Repräsentation der Objektklasse angesehen werden. Zu dieser Repräsentation gelangt das Informationsverarbeitungssystem durch einen Lernprozeß (siehe auch die Ausführungen zu „Schemata“ im Abschnitt 3.3.2). Das Erkennen eines Objektes derselben Klasse erfolgt dann aufgrund eines Vergleichs mit dieser gelernten Repräsentation. Im folgenden Abschnitt wird ein am Institut für Neuroinformatik verwendetes Verfahren zur Objekterkennung beschrieben. Der darauffolgende Abschnitt diskutiert einige Probleme dieses Verfahrens.

2.1 Objektklassenrepräsentation durch Bunchgraphen

In [LVB^+92] und [WFK^+96] ist ein Repräsentationsbildungsprozeß beschrieben, dessen kurze Zusammenfassung nun erfolgt.
Die Repräsentation eines individuellen Objektes ist ein „etikettierter“ Graph G, ein *Bildgraph*. Er besteht aus N Knoten und E Kanten, die die Knoten verbinden. Knoten und Kanten sind mit Informationen versehen (etikettiert), die charakteristisch für das individuelle Objekt sind. Die Position der Knoten $\vec{x}_n \in \mathbb{R}^2$, $n = 1, \ldots, N$, ist ebenfalls objektspezifisch. Sie ergibt sich im einfachsten Fall aus einer manuellen Vorgabe, die die Positionen so bestimmt, daß die Knoten spezifische Landmarken des individuellen Objektes bezeichnen. Üblicherweise werden hier Objektkanten als Landmarken gewählt und andere Stellen, die das Objekt besonders gut von Objekten anderer Klassen zu diskriminieren vermögen. Bei Gesichtern sind diese Landmarken z.B. Augen, Nase oder Mund (siehe Abbildung 2.1).
Eine Kante e, $e = 1, \ldots, E$, eines Bildgraphen ist mit der vektoriellen Distanz

$$\Delta\vec{x}_e = \vec{x}_n - \vec{x}_{n'}$$

zwischen den der Kante zugehörigen Knoten n und n' etikettiert. Das Etikett eines Knotens ist eine Repräsentation seiner lokalen Bildumgebung, die auf einer *Gaborwavelettransformation* beruht. *Gaborwavelets* sind biologisch motivierte Filter [Dau88], [LVB^+92], die aus dem Produkt einer zweidimensionalen Gaußglocke mit einer ebenen Welle gebildet werden (siehe die Ausführungen zu „Kantensensitive Zellen in V1“ in Abschnitt 3.3.1 und Abbil-

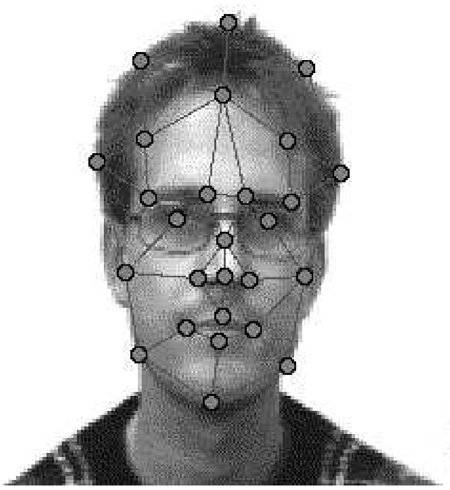


Abbildung 2.1: BILDGRAPH. Die Knoten eines Bildgraphen befinden sich an ausgezeichneten Landmarken des Objektes und sind mit Information etikettiert, die die Umgebung der jeweiligen Landmarke beschreiben.

dung 2.2). Die Bezeichnung von Gaborwavelets als „Filter“ rührt her aus der Tatsache,

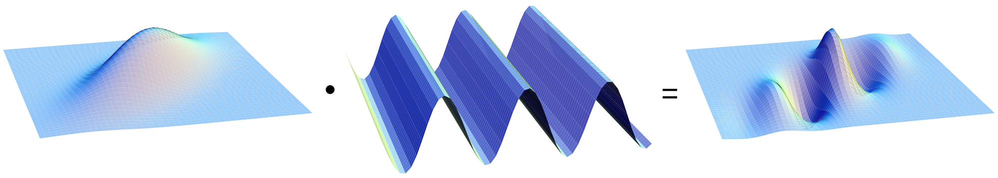


Abbildung 2.2: GABORWAVELET. Ein Gaborwavelet ist das Produkt einer Gaußglocke mit einer ebenen Welle. Die Darstellung zeigt den Realteil eines komplexen Gaborwavelets. Der Imaginärteil zeichnet sich durch eine Phasenverschiebung der Welle aus. Die Abbildung wurde aus [Pöt94] entnommen.

daß sie bei der Konvolution mit natürlichen Bildern als Detektoren von lokalen Strukturen bestimmter Frequenzen und Orientierungen verwendet werden. Eine *Familie von Gaborwavelets* enthält Filter, die sich hinsichtlich ihrer Frequenz und ihrer Orientierung unterscheiden. Zum Beispiel erhält man für die in [LVB^+92] getroffene Parameterwahl

$$\psi_j\left(\vec{x}\right) \;=\; \frac{k_j^2}{\sigma^2}\exp\left(-\frac{k_j^2x^2}{2\sigma^2}\right)\left[\exp\left(i\vec{k}_j\vec{x}\right)-\exp\left(-\frac{\sigma^2}{2}\right)\right]$$

mit den Parametern

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu \cos\varphi_\mu \\ k_\nu \sin\varphi_\mu \end{pmatrix}, \qquad (2.1)$$

$$k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \quad \varphi_\mu = \mu\frac{\pi}{8}$$

und den Indizes

$$\nu = 0,\ldots,4, \quad \mu = 0,\ldots,7 \quad \text{und} \quad j = \mu + 8\nu.$$

Hier liegen fünf verschiedene Frequenzen (Index ν) und acht verschiedene Orientierungen (Index μ) vor. Da $j = 0,\ldots,39$ gilt, besteht diese Familie aus 40 Gaborwavelets.

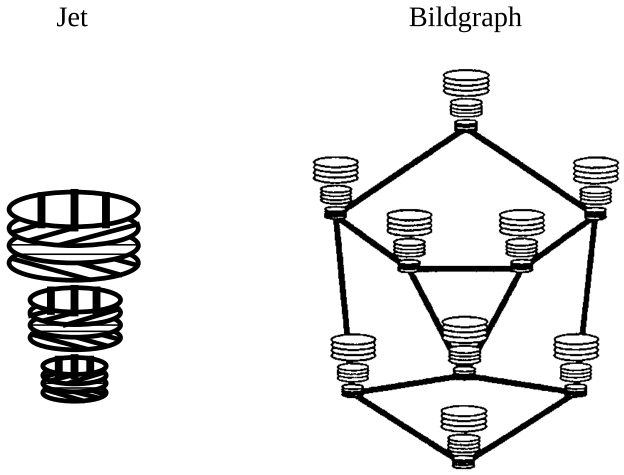


Abbildung 2.3: JET UND BILDGRAPH. Dieser Jet wird aus einer Familie von Gaborwavelets mit drei Frequenzen und vier Orientierungen gebildet. Jeder Knoten des Bildgraphen ist mit einem solchen Jet etikettiert.

In einem Beispielbild, realisiert durch eine meßbare Funktion $I : \mathbb{R}^2 \longrightarrow \mathbb{R}$ (siehe auch Definition 4.6), wird an der Position $\vec{x}_n = (x_n, y_n) \in \mathbb{R}^2$ des Graphenknoten n eine Konvolution mit einer Familie von Gaborwavelets durchgeführt:

$$\mathcal{J}_j\left(\vec{x}_n\right) := \int\limits_{\mathbb{R}^2} I(\vec{x})\, \psi_j\left(\vec{x}_n - \vec{x}\right) d^2\vec{x}. \tag{2.2}$$

Die Vereinigung der Faltungsergebnisse eines jeden Filters an der Position $\vec{x}_n$ bildet dann einen *Jet*

$$\mathcal{J}_n := \left\{ \mathcal{J}_j\left(\vec{x}_n\right) \;\middle|\; j = \mu + 8\nu \right\},$$

das Etikett des Knotens n. Ein Jet kann als eine lokale Beschreibung einer Landmarkenumgebung interpretiert werden (siehe Abbildung 2.3).

Auf diese Weise können *Modellgraphen* gewonnen werden. Das sind Bildgraphen, die einer Galerie zugeführt wurden. Um zu einer allgemeinen Repräsentation einer ganzen Objektklasse zu gelangen, wird nun eine repräsentative Auswahl von M individuellen Modellgraphen, die von M unterschiedlichen Trainingsbildern generiert wurden, getroffen und zu einem *Bunchgraph* B zusammengefaßt. Das ist ein Graph, der die gleiche Struktur wie jeder einzelne Modellgraph aufweist. Ein Knoten n eines Bunchgraphen ist jedoch im Gegensatz zu einem Modellgraphen nicht mit einem Jet etikettiert, sondern mit einem *Bunch* von Jets

$$\mathcal{J}_n^{\mathsf{B}} := \left\{ \mathcal{J}_n^m \;\middle|\; m = 1, \ldots, M \right\}.$$

Das ist die Menge aller Jets aus den jeweils korrespondierenden Knoten der Modellgraphen (siehe Abbildung 2.4). Die Kanten von B sind mit den Durchschnittsdistanzen

$$\Delta\vec{x}_e{}^{\mathsf{B}} \;=\; \frac{1}{M} \sum_{m=1}^{M} \Delta\vec{x}_e{}^m$$

aus den einzelnen Modellgraphen etikettiert, wobei e über die Kanten von B läuft und $\Delta\vec{x}_e{}^m$ für die Kante e des m-ten Modellgraphen definiert ist.

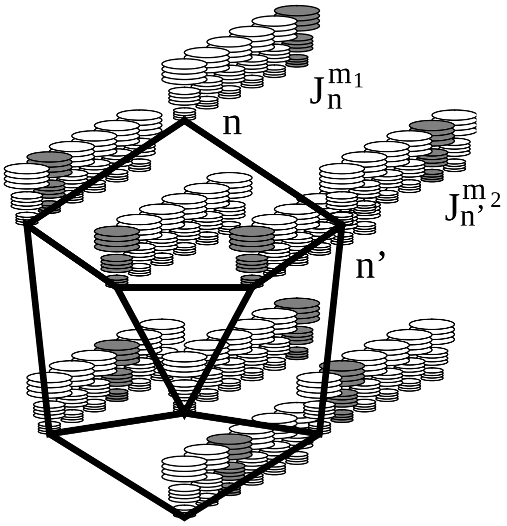


Abbildung 2.4: BUNCHGRAPH. Die Knoten sind mit Bunches von Jets etikettiert. Die Repräsentation einer Objektklasse durch einen Bunchgraph deckt allgemeinere Variationsmöglichkeiten ab als eine Repräsentation durch die bloße Sammlung aller Modellgraphen, denn der Bunchgraph beinhaltet alle Kombinationsmöglichkeiten der Jets aus verschiedenen Modellgraphen (hier durch die dunkel hervorgehobenen Jets gekennzeichnet, die als *lokale Experten* interpretiert werden können).

Das Auffinden eines (neuen) Objektes der entsprechenden Objektklasse in einer komplexen Szene wird mit dem Begriff „Anpassen“ bezeichnet. Dabei wird im Testbild zunächst die grobe Position ermittelt, die die größte Ähnlichkeit (s.u., insbesondere Gleichung 2.3) zu einem aus dem Bunchgraph errechneten Durchschnittsgraphen aufweist. Diese Position wird dann schrittweise verfeinert, wobei pro Knoten der „am besten passende“ Jet des entsprechenden Bunches aus dem Bunchgraph ermittelt wird (s.u.). Für das Anpassen wird also die gesamte Information, die im Bunchgraph enthalten ist, benötigt, um Ähnlichkeiten zu dem Testbild zu entdecken. Die Ähnlichkeiten werden durch Jetvergleiche festgestellt. In den folgenden Erläuterungen werden Jets unabhängig von der Knotenposition $\vec{x}_n$ behandelt, so daß statt $\mathcal{J}_j(\vec{x}_n)$ und $\mathcal{J}_n$ jetzt lediglich $\mathcal{J}_j$ bzw. $\mathcal{J}$ geschrieben wird. $\mathcal{J}_j$ wie in Gleichung 2.2 kann auch geschrieben werden als

$$\mathcal{J}_j \;=\; a_j \exp(i\phi_j)$$

mit Amplituden $a_j \in \mathbb{R}$ und Phasen $\phi_j \in \mathbb{R}$, wobei (a_j, ϕ_j) eine Ersatzrolle für $\vec{k}_j$ aus Gleichung 2.1 spielt. Eine Ähnlichkeitsfunktion für Jets, die in [LVB^+92] vorgeschlagen wurde, ist dann z.B.

$$S(\mathcal{J}, \mathcal{J}') \;=\; \frac{\sum_j a_j {a'}_j}{\sqrt{\sum_j a_j^2 \sum_j {a'}_j^2}}. \tag{2.3}$$

Es wird die Ähnlichkeit eines Jets $\mathcal{J}'$ eines Knotens des neuen Objektes zu jedem Jet $\mathcal{J}$ im Bunch des entsprechenden Knoten im Bunchgraph berechnet. Derjenige, der die größte Ähnlichkeit zu $\mathcal{J}'$ aufweist - diese berechnet sich nach $\max_{\mathcal{J}} S(\mathcal{J}, \mathcal{J}')$ - wird als *lokaler Experte* für den Vergleich zwischen der Bunchgraph-Repräsentation und dem neuen Objekt herangezogen. (In [WFK^+96] wird neben der Funktion 2.3 eine zusätzliche Ähnlichkeitsfunktion S gewählt, die im Gegensatz zu Gleichung 2.3 auch die Phaseninformation berücksichtigt. Für eine genauere Beschreibung des *Elastic Bunch Graph Matching* siehe [WFK^+96].)

2.2 Diskussion der Repräsentation durch Bunchgraphen

Bevor einige Probleme der Objektklassenrepräsentation durch Bunchgraphen diskutiert werden, soll ein Vorzug besonders hervorgehoben werden. Wie aus Abbildung 2.4 hervorgeht, kann mit Hilfe eines Bunchgraphen auch ein Objekt in einem Testbild erkannt werden, an dessem *einen* Knoten n die Beschreibung der Landmarkenumgebung $\mathcal{J}_n$ die größte Ähnlichkeit mit einem Jet $\mathcal{J}_n^{m_1}$ des Trainingsbildes m_1 aufweist, während an einem *anderen* Knoten n' im Testbild der Jet $\mathcal{J}_{n'}$ die größte Ähnlichkeit zu einem Jet $\mathcal{J}_{n'}^{m_2}$ des Trainingsbildes m_2 (mit $m_1 \neq m_2$) aufweist. Das bedeutet z.B. für die Gesichtserkennung, daß nicht für jede Kombination von Merkmalsausprägungen (z.B. „große Nase, großer Mund“, „große Nase, kleiner Mund“, usw.) ein eigenes Gesicht als Modellgraph vorhanden sein muß, sondern es genügt das Vorhandensein einer Merkmalsausprägung (z.B. „große Nase“) in *irgendeinem* Trainingsbild, um alle möglichen Kombinationen erkennen zu können. Das liegt an der unabhängigen Behandlung verschiedener, lokaler Landmarkenumgebungen (siehe auch Prinzip P1 des Lernkonzepts in Abschnitt 3.1).

Die Darstellung einer Objektklasse in Form eines Bunchgraphen birgt jedoch auch einige Nachteile in sich.

Ein Nachteil besteht in der großen Datenmenge, die ein Bunchgraph beinhaltet. Sie ist proportional zur Anzahl der verwendeten Trainingsbilder, obwohl durch die Hinzunahme weiterer Trainingsbilder die zusätzlich gewonnene Information über die Objektklasse nicht in gleichem Maße wächst. Durch die Ähnlichkeit zweier Objekte der gleichen Klasse enthält ein Bunchgraph häufig redundante Information über diese ähnlichen Merkmale.

Ein zweiter Nachteil folgt direkt aus dem ersten. Die große Datenmenge führt zu einem hohen zeitlichen Aufwand beim Anpassen, da Vergleiche mit jedem einzelnen Jet $\mathcal{J}_n^m$, $m = 1, \ldots, M$, eines jeden Bunches eines jeden Knotens n durchgeführt werden müssen.

Ein dritter Nachteil liegt im Bereich der Zuverlässigkeit von Erkennungsergebnissen, da ein Bunchgraph als Folge seiner Generierung auch solche Informationen enthalten kann, die gar nicht spezifisch für die betrachtete Objektklasse sind. Denn für die Objektklasse untypische Merkmale von Trainingsobjekten werden nicht als solche erkannt, sondern gleichberechtigt zu den typischen Merkmalen in den Bunchgraph aufgenommen. Um dieses Problem zu umgehen, muß eine Vorabauswahl der „günstigen“ Trainingsbeispiele von Hand stattfinden. Zudem beinhaltet ein Jet die Beschreibung eines Bildausschnitts *als Ganzes*, ohne die Möglichkeit, einzelne Merkmale aus dieser Beschreibung separieren zu können.

Diese beiden letztgenannten Aspekte scheinen die bedeutsamsten zu sein. Die Zuordnung eines Objektes zu einer Klasse durch Vergleichsoperationen mit Bunchgraphen ist das Vergleichen lokaler Bildausschnitte des fraglichen Objektes mit korrespondierenden lokalen Bildausschnitten aller zuvor „gesehenen“ Objekte derselben Klasse, *ohne daß zuvor ein Lernprozeß stattgefunden hat.* Diese Vorgehensweise entspricht nicht dem Vorgang in biologischen Systemen (siehe die Erläuterungen zur „Erkennung mäßig komplexer Merkmale“ im Abschnitt 3.3.2). Es wird keine intelligente Vorauswahl *relevanter* Objektmerkmale getroffen, sondern es befinden sich sämtliche, aus allen Trainingsbildern extrahierte Merkmale, d.h. sowohl relevante als auch für die Objektklassenrepräsentation unrelevante

Merkmale, im Bunchgraph.
Ein erster Schritt in Richtung auf einen Lernprozeß wird bereits in [KPM96] beschrieben. Dort wird vor der Aufnahme eines neuen Jets $\mathcal{J}_n$ für einen Knoten n in den Bunchgraph B mit einem einfachen Clusteralgorithmus geprüft, ob bereits ein ähnlicher Jet $\mathcal{J}_n{}'$ für diesen Knoten in B existiert. Falls das der Fall ist, wird $\mathcal{J}_n$ nicht in den Bunchgraph aufgenommen. Auf diese Weise wird Redundanz in B vermieden. Durch diese Lösung bleibt jedoch weiterhin unrelevante Information, die von Beispielen mit untypischen Merkmalen herrührt, im Bunchgraphen erhalten, da sich untypische Merkmale in der Regel *stark* von typischen Merkmalen unterscheiden und diese durch die Ähnlichkeitsfunktion also *nicht* eliminiert werden können. Außerdem beinhaltet ein Jet sowohl die Information des Objektes als auch die des Hintergrundes, sofern dieser in der Landmarkenumgebung vorhanden ist.
Das Ziel dieser Arbeit besteht nun darin, durch eine intelligente Auswahl relevanter Merkmale aus Trainingsbildern und deren statistische Analyse zu einer kompakteren Darstellung einer Objektklasse zu gelangen.

Ein Problem, das dabei an erster Stelle gelöst werden muß, ist die Frage nach der Qualität der Merkmale, auf die das Augenmerk gerichtet werden soll. Vielfältige Möglichkeiten sind hier denkbar, wie z.B. Farbe, Textur oder Kanten.
Ein weiteres Problem, das bei fast allen Arbeiten im Bereich der Objekterkennung zu bewältigen ist und bereits in der Einleitung angedeutet wurde, stellt sich durch variierende Hintergründe, unterschiedliche Beleuchtungssituationen, verschiedene Aufnahmewinkel der Objekte und vielfältige andere Variationsmöglichkeiten. In Kapitel 3 (und ausführlich in Kapitel 5) wird ein Algorithmus vorgestellt, der sich robust gegen variierende Hintergründe, gegen Reflexe oder Schatten durch unterschiedliche Beleuchtungen und in begrenztem Maße auch gegen unterschiedliche Betrachtungswinkel verhält.

Eine weitere Hürde für die Objekterkennung stellt das *Korrespondenzproblem* dar. Mit dem Begriff „Korrespondenzproblem“ wird die Schwierigkeit bezeichnet, in zwei verschiedenen Bildern desselben Objektes die jeweils sich entsprechenden Punkte des physikalischen Objektes einander zuzuordnen. Eine besondere Herausforderung ist die Tatsache, daß es zu manchen Punkten durch teilweise Verdeckung oder Rotation des Objektes gar keine korrespondierenden Punkte im anderen Bild gibt (siehe Abbildung 2.5).

Abbildung 2.5: Korrespondenzproblem. Im linken Bild befindet sich die Ohrspitze an Pixel $(48, 21)$, im rechten Bild an Pixel $(68, 11)$ (Kreis). Das Auge ist im linken Bild an Position $(22, 49)$ vorhanden (Quadrat), während es im rechten Bild keinen korrespondierenden Pixel gibt.

Kapitel 3

Lösungsvorschlag

Im ersten Abschnitt dieses Kapitels wird das Lernkonzept, das dem Lernalgorithmus dieser Arbeit zugrunde liegt, vorgestellt. In Abschnitt 3.2 wird dieses Konzept näher erläutert. Mit Einblicken in neurobiologische Befunde, die den Lernalgorithmus motiviert haben, bildet Abschnitt 3.3 den Abschluß dieses Kapitels.

3.1 Lernkonzept

Dem in den folgenden Kapiteln vorzustellenden Algorithmus zum Lernen einer zweidimensionalen Objektklassenrepräsentation liegt ein von N. Krüger entwickeltes Konzept zugrunde. Es basiert auf einem Ansatz, der bereits in [Krü95] auf das Problem des Lernens von Metriken und in [KPM96] auf das Problem des Gesichtsfindens und Posenschätzens angewendet wurde. Im Hinblick auf den Lernalgorithmus wird es in den Kapiteln 4 und 5 mathematisch formalisiert.

Das Konzept basiert auf der Annahme, daß bereits „inneres Wissen“ über den Gegenstand des Lernens vorhanden sein muß, um eine Lernaufgabe erfolgreich lösen zu können. Damit steht es in der Tradition Kants, der einerseits inneres Wissen voraussetzt, um aus Wahrnehmung Erkenntnis zu erlangen, andererseits aber auch die Möglichkeit des Lernens mit Hilfe dieses inneren Wissens zuläßt [Kan1781]. Dieses „innere Wissen“ soll hier durch *a priori-Voraussetzungen* repräsentiert werden, die über den Gegenstand des Lernens aufgestellt werden. Der Sinn der a priori-Annahmen ist die Reduzierung der Dimension des Lernproblems, so daß auch schwere (d.h. hochddimensionale) Lernprobleme in akzeptabler Rechenzeit algorithmisch gelöst werden können, denn die Mehrzahl der Probleme, die unter Nicht-Labor-Bedingungen auftreten, sind hochddimensional.

Die Schwierigkeit besteht in der Auswahl angemessener a priori-Voraussetzungen. Einerseits sollen sie *allgemein* sein, andererseits aber auch *effektiv*. Im Kontext dieser Arbeit (das Lernen einer Objektklassenrepräsentation anhand von Beispielen) bedeutet die Forderung nach Allgemeinheit, daß mit Hilfe der
a priori-Annahmen wesentliche Probleme der Objekterkennung gelöst werden können, ohne daß aufgabenspezifisch neue Annahmen getroffen werden bzw. alte entfallen müssen. Die Effektivität der a priori-Voraussetzungen drückt sich in der Fähigkeit aus, wesentliche Struktureigenschaften der visuellen Welt abzubilden, um damit die Dimension des Lernproblems zu reduzieren.

Bereits in [KPM96] wurden folgende a priori-Voraussetzungen, die beide Kriterien erfüllen, postuliert:

P1 (Lokalität) Merkmale, die verschiedene Landmarken bezeichnen, werden als unabhängig betrachtet.

P2 (Maximale Diskriminierungfähigkeit) Merkmale, die innerhalb einer Objektklasse kaum, jedoch zwischen verschiedenen Objektklassen stark variieren, werden bevorzugt gelernt.

P3 (Unabhängigkeit gelernter Information) Redundanz von gelernter Information soll reduziert werden.

Hier wird nun eine zusätzliche, weitgehende Annahme über die lokale Struktur der visuellen Welt getroffen:

P4 (Lokale Merkmalsauswahl) Wesentliche Merkmale der lokalen, visuellen Welt sind gekrümmte Kanten.

Diese letzte Annahme P4 wird durch die bereits in der Einleitung erwähnten Bananenwavelets formalisiert. Sie haben die fünf Eigenschaften Ort ($(x, y) \in \mathbb{R}^2$), Frequenz ($k \in \mathbb{R}$), Orientierung ($\alpha \in \mathbb{R}$), Krümmung ($c \in \mathbb{R}$) und Größe ($s \in \mathbb{R}$), so daß sie sich in einem sechsdimensionalen[1] Raum anordnen lassen. Dieser Raum wird in Abschnitt 4.4.1 als *Parameterraum der Bananentransformation* definiert (siehe Definition 5.1). Er ist auf eine natürliche Weise angeordnet. Durch eine *Metrik* auf dem Parameterraum wird diese Ordnung zusätzlich präzisiert.[2]

Das Lernproblem wird durch Prinzip P4 so reduziert, daß mit Hilfe von Kriterium P2 die wichtigen Merkmale einer Objektklasse durch *Auswahl* mittels statistischer Kriterien gelernt werden können. Im Unterschied zu einem Backpropagation-Algorithmus, bei dem zwar auch das Prinzip der Lokalität (P1) angewendet werden kann (siehe z.B. [SS96]), wird hier also der potentielle Lösungsraum *aller* Merkmale bewußt auf eine kleine Teilmenge reduziert, um an die Stelle des Lernens durch ein Gradientenverfahren ein Lernen durch Auswahl von Merkmalen treten zu lassen.

Die Formalisierung der a priori-Annahme P2 wird durch den Aspekt „Häufigkeit“ verwirklicht. Diejenigen Merkmale, die besonders häufig in unterschiedlichen Beispielen auftreten, sollen bevorzugt gelernt werden, denn das häufige Auftreten eines Merkmals bedeutet dessen Invarianz unter verschiedenen Aufnahmebedingungen. Da *exakt* dasselbe Merkmal jedoch *nicht* in vielen verschiedenen Trainingsbeispielen zu erwarten ist, wird hier die Metrik des Parameterraums benötigt, um über eine Ähnlichkeitsfunktion eine Aussage darüber zu erhalten, ob zwei Merkmale „annähernd gleich“ sind. Also wird maximale Diskriminierungfähigkeit denjenigen Merkmalen zugeschrieben, die in vielen Beispielen *ähnlich* sind und besonders *häufig* auftreten.

Um Prinzip P3 zu verwirklichen, soll das Ziel des Lernens die Beschreibung eines lokalen Bereichs einer Objektklasse durch möglichst *wenige* Merkmale sein. Diese wenigen Merkmale sollen aus einer großen Menge möglicher Merkmale ausgewählt werden. Das geschieht durch einen Clusteralgorithmus (siehe Abschnitt 5.4.2). Mit Hilfe der Metrik ist es möglich, Merkmale nach Maßgabe ihrer Ähnlichkeit in *Cluster* einzuordnen. Ein ausgewähltes Merkmal ist dann jeweils ein Vertreter eines Clusters vieler ähnlicher Merkmale.

---

[1] Der Ort ist zweidimensional.

[2] In dieser Arbeit werden die Bananenwavelets und der Lernalgorithmus formal für den sechsdimensionalen Parameterraum eingeführt. In den Simulationen werden jedoch lediglich die Qualitäten „Orientierung“ und „Krümmung“ verwendet. Auf die Erweiterung zu sechs Dimensionen wird im Abschnitt 6.3 und im Ausblick eingegangen (siehe auch [KP]).

Ein solches Merkmal soll lediglich die Eigenschaften „präsent“ bzw. „nicht präsent“ haben. Das bedeutet, eine bestimmte, gekrümmte Kante soll an einer bestimmten Stelle in einer Landmarkenumgebung entweder vorhanden oder nicht vorhanden sein. Die Merkmale sollen also *binär* sein. Dieses Kodierungsprinzip (die Beschreibung eines Objektes durch wenige, binäre Merkmale, die aus einem großen Merkmalsraum ausgewählt werden) wird mit dem Begriff „Sparse Coding“ bezeichnet. Die Vorteile des Sparse Coding gegenüber einem *Compact Coding* wurden bereits in [Pal80] und [Fie94] beschrieben. So ist z.B. für ein schnelles Anpassen eines Objektes in einer komplexen Szene von entscheidender Bedeutung, daß lediglich wenige Operationen benötigt werden, um einen Bildbereich auf das Vorhandensein eines Objektes hin zu überprüfen. Dieses wird durch ein Sparse Coding ermöglicht.

Für den in dieser Arbeit vorzustellenden Lernalgorithmus ist es wichtig, die verschiedenen, lokalen Landmarkenumgebungen unabhängig zu behandeln. Um das Prinzip der Lokalität (P1) zu verwirklichen, besteht die Notwendigkeit, das Korrespondenzproblem[3] bereits vor dem Lernen zu lösen. Wird dieses Problem hier noch durch manuelle Vorgabe korrespondierender Punkte in unterschiedlichen Bildern gelöst, so wird es bereits in [KP] durch visuomotorische Algorithmen gehandhabt.

3.2 Diskussion des Lernkonzepts

Die Unterschiede zwischen dem hier vorzustellenden Lösungsvorschlag und dem in Abschnitt 2.1 beschriebenen Verfahren (Repräsentation durch Bunchgraphen) liegen sowohl in der Art und Weise, in der die Bildvorverarbeitung durchgeführt wird, als auch in der Qualität der zu lernenden Objektklassenrepräsentation. Diese beiden Punkte werden nun unter Anwendung des gerade vorgestellten Lernkonzeptes näher erläutert.

3.2.1 Bildvorverarbeitung

In jedem Trainingsbild werden Landmarken vorgegeben. Das sind Punkte, deren Umgebungen hinsichtlich bestimmter Merkmale untersucht werden sollen. Das bedeutet, daß das Korrespondenzproblem für diese Arbeit als gelöst betrachtet werden soll. (Im Ausblick wird erläutert, wie diese Einschränkung umgangen werden kann.)

Die Landmarken werden in den Trainingsbildern so vergeben, daß sie jeweils korrespondierende Punkte des Objektes bezeichnen. Beispielsweise bezeichnet eine bestimmte Landmarke in allen Trainingsbildern immer die Mitte der linken Kante einer Getränkedose. Jede Umgebung einer Landmarke wird dann auf Lokalität und Ausprägung von bestimmten Merkmalen untersucht. Diese Merkmale sollen, wie bereits in der a priori-Voraussetzung P4 des Lernkonzeptes (Abschnitt 3.1) ausgedrückt, *gekrümmte Kanten* sein. Für die Umgebung einer Landmarke in einem Trainingsbild wird festgestellt, an welchen Stellen sich gekrümmte Kanten befinden und durch welche Parameter sich die Qualität der jeweiligen Krümmungen beschreiben läßt (siehe Abbildung 3.1). Dieser Vorgehensweise liegt der Gedanke zugrunde, daß die visuelle Umwelt sich hinsichtlich der gestellten Aufgabe, eine zweidimensionale Objektklassenrepräsentation zu lernen, einfach und doch effizient durch eine Verteilung von verschiedenartig gekrümmten Kanten beschreiben läßt (siehe auch „Krümmungssensitive Zellen in V1“ in Abschnitt 3.3.1). Besonders bei der Trennung von Objekten vom Hintergrund, dem *Figur-Grund-Problem*, liefern die Objektkanten wichtige

---

[3] Zum Korrespondenzproblem siehe das Ende von Abschnitt 2.2 und Kapitel 8 (Ausblick).

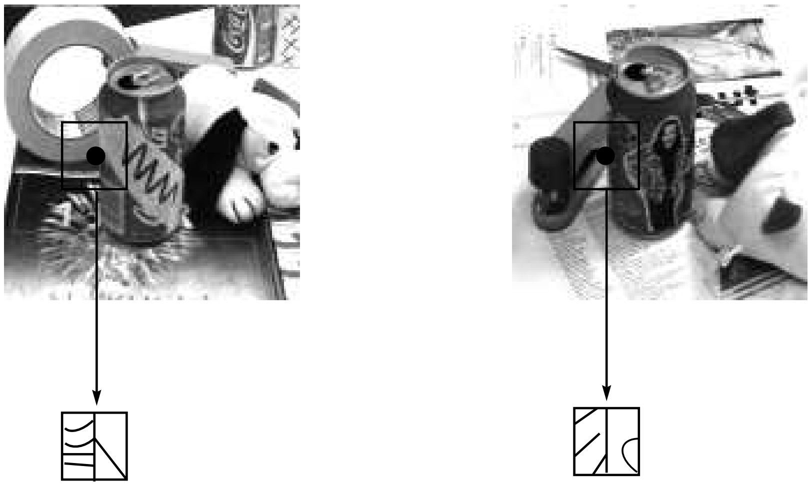


Abbildung 3.1: LANDMARKENUMGEBUNG. In der oberen Reihe sind zwei Beispiele der verwendeten Trainingsbilder abgebildet. Neben anderen Landmarken wird in allen Trainingsbildern auch die gekennzeichnete ausgewählt.
In der unteren Reihe ist wiedergegeben, wie sich die jeweiligen Umgebungen der Landmarke durch gekrümmte Kanten beschreiben lassen sollen. (Der Begriff „gekrümmte Kanten“ schließt die nicht-gekrümmten Kanten ein.) „Spezifisch“ für die zu lernende Repräsentation der Objektklasse „Getränkedose“ ist hier jeweils nur die gerade, senkrechte Linie in der Mitte. Die anderen Merkmale entstehen durch Variationen im Hintergrund bzw. durch das Dosendesign. Diese Darstellung ist eine idealisierte. In Abbildung 6.3 befindet sich die Darstellung, die tatsächlich vom Algorithmus geleistet wird.

Hinweise. Objektkanten lassen sich in einer zweidimensionalen Ansicht des Objektes fast immer als (oft jedoch lückenhafte) Aneinanderreihung verschieden gekrümmter Kanten interpretieren. Dabei können diese sich im Ausmaß ihrer Krümmung unterscheiden, in ihrer Länge und Orientierung.
Die Wahl, sich bei der Analyse einer Landmarkenumgebung auf die Suche nach gekrümmten Kanten zu beschränken, erscheint somit als sinnvolle a priori-Bedingung zur Reduzierung des Lernproblems.
Im Unterschied zu der in Abschnitt 2.1 beschriebenen Methode der Bildvorverarbeitung mit Gaborwavelets wird hier eine Transformation mit „gekrümmten Gaborwavelets“, den Bananenwavelets, eingeführt, die gerade bei der Konvolution mit gekrümmten Kanten ein spezifisches Antwortverhalten zeigen und somit den zuvor genannten Forderungen Rechnung tragen (siehe die Abbildungen in Abschnitt 4.5.2).

3.2.2 Erzeugung einer Objektklassenrepräsentation

Im Gegensatz zu einer Objektklassenrepräsentation durch Bunchgraphen, die auch unwichtige, aus den Trainingsbildern zur Verfügung gestellte Information enthält, wird hier eine Repräsentation durch gezielte *Auswahl* lediglich der als *wichtig* befundenen Information gebildet.
Bei der Suche nach gekrümmten Kanten werden sowohl solche entdeckt, die spezifisch für die Objektklasse sind, als auch solche, die z.B. im Hintergrund vorhanden sind (siehe wiederum Abbildung 3.1). Unter der Prämisse, daß die zu lernenden, für eine Objektklasse typischen Kanten sich in *sehr vielen* Trainingsbeispielen (genauer: in sehr vielen korrespondierenden Landmarkenumgebungen) *ähneln*, die störenden „Hintergrundkanten“

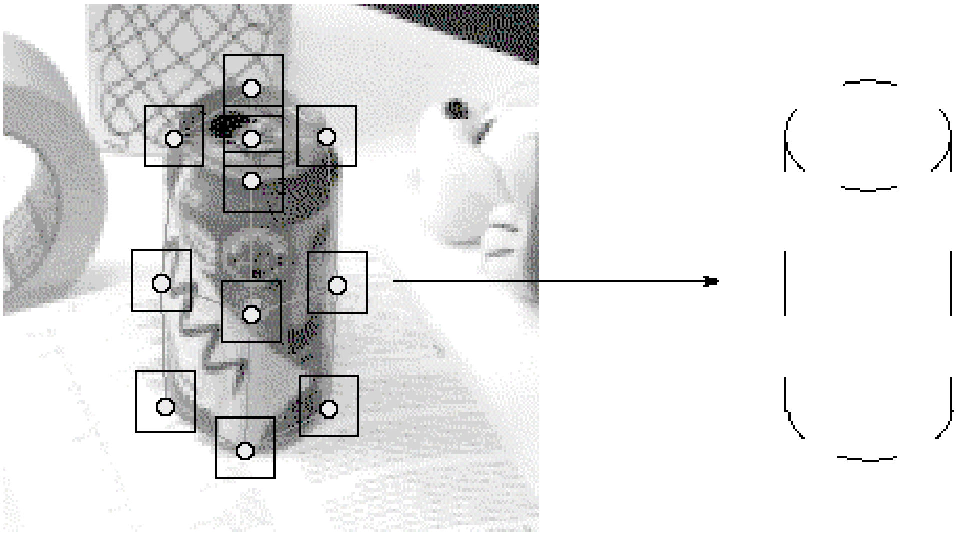


Abbildung 3.2: Erwartete Repräsentation. Werden in allen Trainingsbildern die Landmarken wie links gewählt (weiße Punkte), so wird durch das Aussortieren der wichtigen von den unwichtigen Merkmalen in den jeweiligen Landmarkenumgebungen (Quadrate, die die Landmarken umgeben) eine Repräsentation wie rechts erwartet. Diese Repräsentation kann als „Prototyp“ der Objektklasse „Getränkedose“ bezeichnet werden. (Auch hierbei handelt es sich um eine idealisierte Darstellung. Das tatsächliche Ergebnis ist in Abbildung 6.4 abgebildet.)

jedoch unregelmäßig auftreten, wird in diesem Lösungsansatz gezielt nach Gemeinsamkeiten in den verschiedenen Beispielen gesucht. Merkmale (gekrümmte Kanten), die *häufig* in unterschiedlichen Beispielen auftreten, werden als repräsentativ gewertet, andere, selten auftretende als unwichtig. In die Objektklassenrepräsentation werden dann nur diejenigen Merkmale aufgenommen, die häufig genug in unterschiedlichen Beispielbildern entdeckt wurden (siehe Abbildung 3.2). So wird aus zweidimensionalen Ansichten von Vertretern einer bestimmten Objektklasse eine zweidimensionale, ansichtsabhängige Repräsentation gelernt. Dabei dürfen die Ansichten der Beispielobjekte in einem gewissen Rahmen variieren[4].
Die Häufigkeits- und Ähnlichkeitsanalyse der in allen Landmarkenumgebungen gefundenen Merkmale geschieht durch einen Clusteralgorithmus, der in Abschnitt 5.4.2 detailliert erläutert wird.

Die Vorteile einer auf diese Art erzeugten Objektklassenrepräsentation im Vergleich zu einem Bunchgraphen liegen auf der Hand. An die Stelle einer bloßen Datensammlung tritt eine aktive Auswahl der wichtigen Information, d.h. ein *Lernprozeß*. Die damit verbundene Datenreduktion bringt eine für die Computersimulation vorteilhafte Einsparung von Speicherkapazitäten mit sich.
Der Verzicht auf unwichtige Information in der Repräsentation ermöglicht potentiell ein verläßlicheres Anpassen. Zusätzlich kann das Anpassen beschleunigt werden, da nun nicht mehr Vergleiche mit allen Jets durchgeführt werden müssen, sondern lediglich mit den relativ wenigen gelernten Merkmalen (siehe Kapitel 8).

---

[4] So sind z.B. für die in Kapitel 6 beschriebene Simulation Bilder von Getränkedosen aus drei leicht unterschiedlichen Aufsichtswinkeln aufgenommen worden (siehe Abbildung 6.2).

Außerdem kommt die Bildung einer Repräsentation durch *Verallgemeinerung aus einer Reihe von Beipielen* den Erkenntnissen der Psychologie näher (siehe Abschnitt 3.3.2). Insofern hat die hier vorgeschlagene Objektklassenrepräsentation auch in dieser Hinsicht eine andere Qualität als ein Bunchgraph.

Ein weiterer Vorteil besteht in der *Visualisierbarkeit* der gelernten Repräsentation. Werden die gelernten Merkmale (in bestimmter Weise gekrümmte Kanten an bestimmten Stellen einer Landmarkenumgebung) als gekrümmte Linien gezeichnet, so wird als Ergebnis ein zweidimensionaler *Prototyp* der Objektklasse wie in Abbildung 3.2 erwartet. Die Bezeichnung „Prototyp“ ist dem analogen, psychologischen Terminus entlehnt, der in Abschnitt 3.3.2 erläutert wird (siehe die Ausführungen zu „Schemata“).

Zusammenfassend läßt sich sagen, daß die bereits in [KPM96] geforderte, systematische Auswahl komplexerer Merkmale (hier: gekrümmter Kanten) aus einer größeren Menge zu einer Repräsentation führt, die die *Essenz* der Beispiele beinhaltet.

3.3 Neurobiologische und psychologische Hintergründe

Dieses Kapitel soll Einblick geben in einige Erkenntnisse über bestimmte Funktionsweisen menschlicher und tierischer Gehirne, die für diese Arbeit relevant sind. Diese Erkenntnisse können für die abstrakte Modellbildung und Simulation bestimmter Hirnfunktionen in technischen Systemen genutzt werden. Parallelen zwischen den biologischen und psychologischen Befunden einerseits und Aspekten des in dieser Arbeit vorgestellten Modells zur Bildung einer Objektklassenrepräsentation andererseits werden also bewußt gezogen. Es wird jedoch ausdrücklich darauf hingewiesen, daß aus der Funktionstüchtigkeit eines technischen Systems, z.B. bei der Objekterkennung, nicht auf eine analoge Funktionsweise biologischer Systeme geschlossen werden kann. Die Aspekte neurobiologischer Vorgänge dienen hier lediglich als Quelle der Inspiration bei der Modellierung von Hirnfunktionen in künstlichen Systemen[5].

Während der erste Abschnitt („Kortikale Bildvorverarbeitung“) neurobiologische Parallelen zur *Bildvorverarbeitung* in künstlichen Objekterkennungssystemen vorstellt, werden im zweiten Abschnitt („Höhere kortikale Verarbeitungsstufen“) biologische und psychologische Befunde diskutiert, die mit dem Lernkonzept und der Formulierung des Lernalgorithmus in Beziehung gesetzt werden können.

3.3.1 Kortikale Bildvorverarbeitung

Ein Bild, das auf die Retina eines Auges fällt, erzeugt dort in den Photorezeptoren visuelle Signale, die durch den Sehnerv über die Sehnervenkreuzung und die seitlichen Kniehöcker zum *primären visuellen Kortex (V1)* geleitet werden (siehe Abbildung 3.3).

Kantensensitive Zellen in V1

Nachdem bereits in der Retina eine gewisse Vorverarbeitung der Lichtsignale vorgenommen wurde (die man als Kontrastverstärkung bezeichnen kann), findet im primären visuellen Kortex die erste genauere Analyse statt. In V1 befinden sich Zellen mit unterschiedlichem Komplexitätsgrad. Die *einfachen Zellen* beziehen ihre Information von relativ kleinen, lokalen Bereichen der Retina, haben also kleine *rezeptive Felder* und nehmen eine Analyse kleiner, lokaler Bereiche im Gesichtsfeld vor (vergleiche mit der a priori-Annahme P1 in Abschnitt 3.1). Während in [HW62] noch davon ausgegangen wurde, daß der optimale Stimulus einer einfachen Zelle ein in bestimmter Richtung orientierter Balken in ihrem rezeptiven Feld sei, belegen neuere Arbeiten, daß sich die rezeptiven Felder einfacher Zellen angemessener durch zweidimensionale Gaborwavelets (siehe Definition 4.2) beschreiben lassen [JP87], [dVdV88]. Das begründet die Relevanz von Gaborwavelets für die Bildvorverarbeitung in künstlichen Objekterkennungssystemen (vergleiche mit Abschnitt 2.1).

Krümmungssensitive Zellen in V1

Im Gebiet V1 sind neben kantensensitiven Zellen auch solche Zellen lokalisiert worden, die spezifisch auf *Krümmungen* reagieren. [DZC87] haben durch Einzelzellableitungen im primären visuellen Kortex von Katzen herausgefunden, daß die sogenannten *endinhibierten Zellen*, denen man bisher lediglich das Erkennen von Balken oder Kanten einer *bestimmten*

---

[5] Umgekehrt können natürlich auch die Erkenntnisse der Neuroinformatik neue Fragen in Biologie und Psychologie aufwerfen und motivierend für neue Theorien wirken.

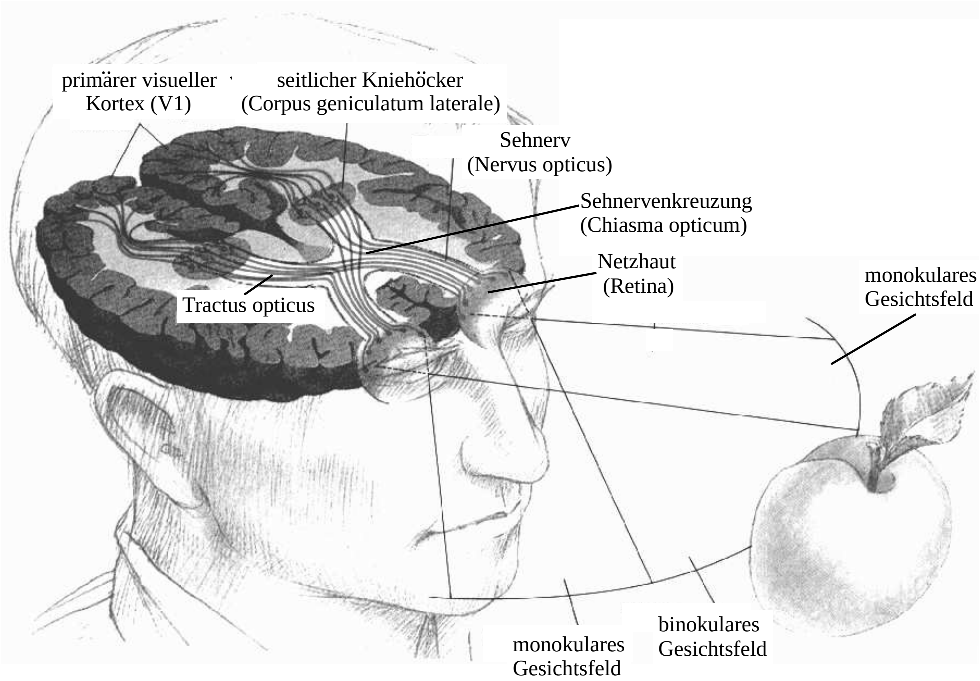


Abbildung 3.3: SEHBAHN. Die Abbildung zeigt die menschlichen Großhirnhemisphären im Horizontalschnitt. An der Sehnervenkreuzung wechselt jeweils etwa die Hälfte der Sehnervenfasern auf die gegenüberliegende Hemisphäre, so daß beide Hemisphären jeweils Input von beiden Augen erhalten, jedoch die linke Seite ausschließlich Information aus der rechten Hälfte der visuellen Umgebung und umgekehrt. Die Abbildung stammt in leichter Veränderung aus [Sha93].

Länge zuschrieb, mit einer besonders hohen Antwortrate reagieren, wenn sich in ihrem rezeptiven Feld eine Kante mit einer bestimmten Krümmung befindet. Nicht-endinhibierte Zellen weisen diese Eigenschaft nicht auf. Zur Erklärung der Krümmungssensitivität wird in [DZC87] ein Modell angegeben, das schon die Funktion der Endinhibition erklären konnte [Hub89]. Eine endinhibierte Zelle E erhält *exzitatorischen* Input von einer einfachen Zelle S mit *kleinem* rezeptiven Feld und *inhibitorischen* Input von einer einfachen Zelle L mit *großem* rezeptiven Feld, das im selben Punkt des Gesichtsfeldes zentriert ist. Wenn R_S die Antwort der kleinen einfachen Zelle S und R_L die Antwort der großen einfachen Zelle L auf ein bestimmtes Muster ist, läßt sich die Antwort R_E der endinhibierten Zelle E schreiben als

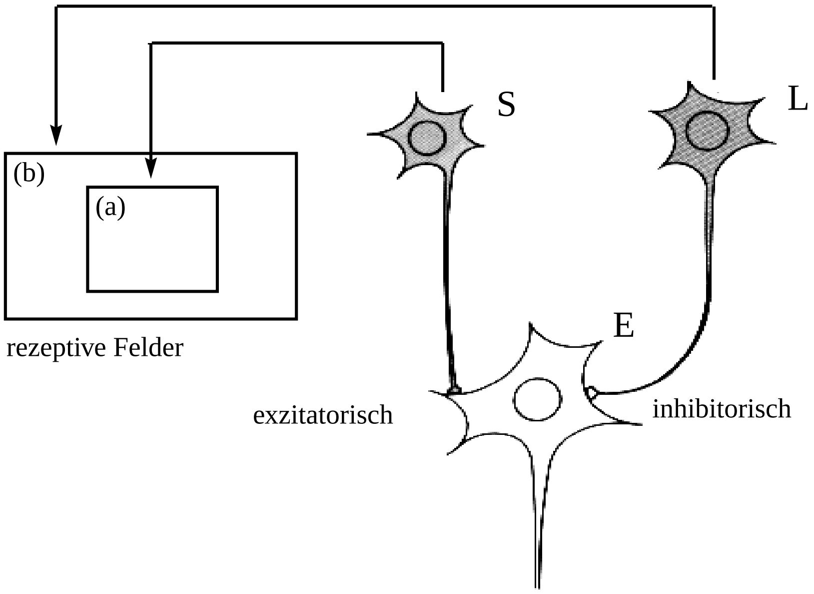


Abbildung 3.4: MODELL FÜR DIE FUNKTIONSWEISE EINER ENDINHIBIERTEN ZELLE E. (a) ist das rezeptive Feld der erregenden Zelle S, (b) das rezeptive Feld der hemmenden Zelle L. Damit diese Verschaltung funktioniert, muß man voraussetzen, daß die hemmende Zelle L auf einen kurzen Lichtbalken, der (a) stimuliert, nur schwach, auf einen langen dagegen stark reagiert. Die Abbildung stammt in leichter Veränderung aus [Hub89].

$$R_E = \phi\left(c_S \cdot \phi\left(R_S\right) - c_L \cdot \phi\left(R_L\right)\right)$$

mit

$$\phi(X) = \begin{cases} X & X > 0 \\ 0 & X \leq 0 \end{cases}$$

und c_S, c_L positive Konstanten, die die Größendifferenz der rezeptiven Felder normalisieren (siehe Abbildung 3.4). Die Form von ϕ spiegelt die Eigenschaft von Zellen des primären visuellen Kortex wider, keine negative Aktivität kodieren zu können.

Befindet sich nun ein Stimulus mit einem bestimmten Krümmungsradius in den rezeptiven Feldern der beiden einfachen Zellen, so fällt ein Großteil der Kurve in die exzitatorische Region des rezeptiven Feldes der kleinen einfachen Zelle. Es trifft jedoch lediglich ein zu kleiner Anteil des Stimulus auf den exzitatorischen Bereich des rezeptiven Feldes der großen einfachen Zelle, um eine effektive Hemmung auf die endinhibierte Zelle auszuüben. Also wird die endinhibierte Zelle erregt (siehe Abbildung 3.5).

So können verschiedene Kombinationen von kleinen erregenden und großen hemmenden einfachen Zellen Sensitivität für verschieden starke Krümmungen in endinhibierten Zellen bewirken. In [DZC87] wird bereits die Bedeutung von Krümmungssensitivität für die Repräsentationsbildung betont, da in natürlichen Bildern weit mehr Kurven als z.B. Endpunkte von Kanten oder Balken vorhanden seien (vergleiche mit der a priori-Voraussetzung P4 in Abschnitt 3.1).

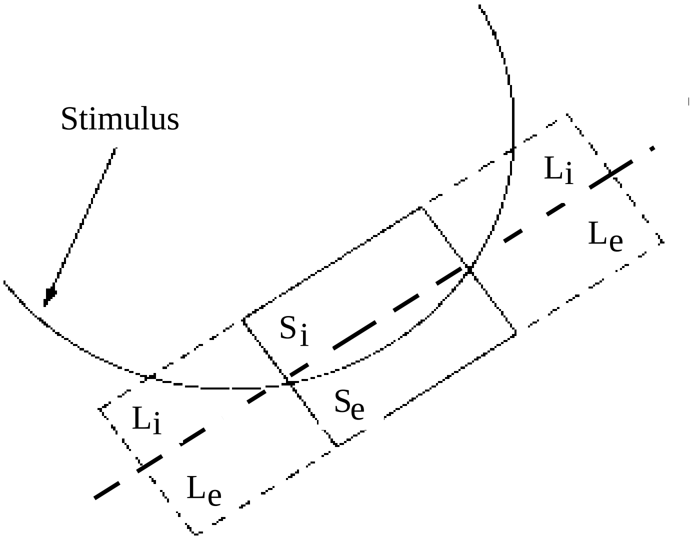


Abbildung 3.5: EIN WIRKSAMER STIMULUS FÜR EINE ENDINHIBIERTEN ZELLE. S_e bezeichnet die exzitatorische Region des rezeptiven Feldes der (exzitatorisch wirkenden) einfachen Zelle S, L_i bezeichnet die inhibitorische Region des rezeptiven Feldes der (inhibitorisch wirkenden) einfachen Zelle L. Die Abbildung wurde in leichter Veränderung aus [Hub89] entnommen.

3.3.2 Höhere kortikale Verarbeitungsstufen

Vom primären visuellen Kortex (V1) wird die Information in den *visuellen Assoziationskortex* geleitet [Car91]. Zu diesem gehören unter anderem die kortikalen Gebiete V2 bis V5, die den primären visuellen Kortex umgeben, und der *inferiore Temporalkortex* (siehe Abbildung 3.6). Vereinfacht dargestellt kommt V1 und V2 nach [Car91] die Funktion eines Postamtes zu, das eingehende Information vorverarbeitet und an die Gebiete V3, V4 bzw. V5 weiterleitet, je nachdem, ob Form, Farbe bzw. Bewegung zu analysieren ist. Der visuelle Assoziationskortex erfüllt, wie auch die Assoziationskortizes der anderen Sinnesmodalitäten, Funktionen für höhere, kognitive Fähigkeiten wie z.B. Wahrnehmung und Erinnerung. Die höchste Stufe der visuellen Analyse erfolgt im inferioren Temporalkortex, der seinen Input unter anderem von den Gebieten V1 bis V5 bezieht. Hier findet die Identifizierung von Objekten statt - also gerade die *Objekterkennung* [Car91].

Erkennung mäßig komplexer Merkmale

Experimente haben nach [Tan93] ergeben, daß in Kolumnen angeordnete Zellen im vorderen Teil des inferioren Temporalkortex „mäßig komplexe“ Stimuli benötigen, um ein maximales Antwortverhalten zu zeigen. Dabei bedeutet „mäßig komplex“, daß der Stimulus komplexer sein muß als eine bloße Kante oder ein Balken, jedoch nicht so komplex, daß eine einzelne Zelle ein ganzes Objekt kodieren könnte. In Abbildung 3.7 sind einige Beispiele solcher mäßig komplexen Merkmale abgebildet. Hier könnte man sich auch die Kombination einiger gekrümmter Linien als kritisches Merkmal denken (vergleiche mit a priori-Voraussetzung P4 in Abschnitt 3.1). Interessanterweise ändert sich die Reaktion dieser Zellen nur wenig, wenn das Bild eines Objektes sich mit anderen Beleuchtungen oder Blickwinkeln stark ändert. Außerdem reagieren Neurone im vorderen Teil des in-

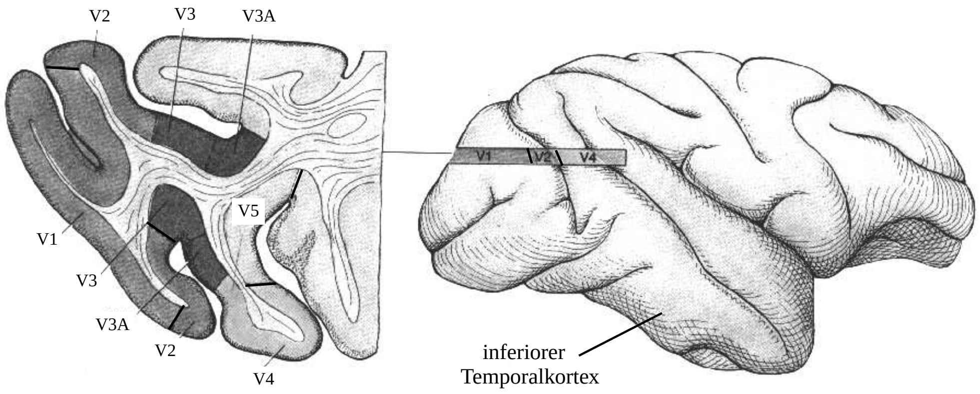


Abbildung 3.6: GEBIETE DES VISUELLEN ASSOZIATIONSKORTEX EINES MAKAKENGEHIRNS. Die Abbildung stammt in leichter Veränderung aus [Zek93].

ferioren Temporalkortex nicht nur auf Inputbilder, sondern sind auch bei der internen Repräsentation von Objekten aktiv. Das wurde nach [Tan93] anhand von Experimenten mit Makakenaffen belegt. Den Tieren wurde ein Stimulus präsentiert, der nach einer Verzögerung wiedererkannt werden sollte. Dabei waren einige der Zellen auch - und andere ausschließlich - während der Pause aktiv.

Im hinteren Teil des inferioren Temporalkortex befinden sich Zellen, die auf weniger komplexe Merkmale reagieren, wie z.B. einzelne, gekrümmte Linien. Diese Neurone sind jeweils auf drei bis fünf Zellen im vorderen Teil des inferioren Temporalkortex verschaltet. Daher kann angenommen werden, daß ein mäßig komplexes Merkmal eine Komposition mehrerer wenig komplexer Merkmale ist. So einer Komposition wenig komplexer Merkmale entspricht im Rahmen dieser Arbeit die Beschreibung einer lokalen Landmarkenumgebung durch wenige gekrümmte Linien (siehe die Ausführungen zur a priori-Annahme P3 in Abschnitt 3.1 und Abbildung 3.1).

In [Tan93] wird der Schluß gezogen, daß es sich bei der Objekterkennung in Primatengehirnen nicht um einen einfachen Vergleich eines Inputbildes mit bereits gespeicherten Bildern handelt (wie es in niederen Tieren der Fall sei), sondern um einen weitaus flexibleren Prozeß. Diese Tatsache spricht dafür, bei der Objekterkennung in künstlichen Systemen das Modell der Bunchgraphen, bei dem die Erkennung eher einem Vergleich

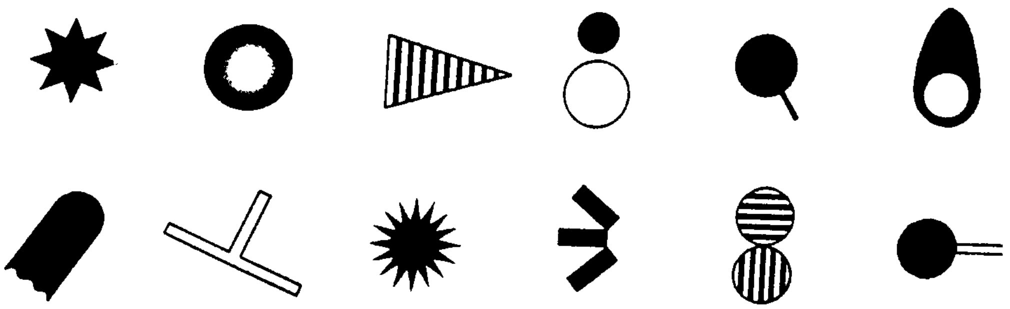


Abbildung 3.7: MÄSSIG KOMPLEXE MERKMALE. Diese Merkmale können eine Zelle im vorderen Teil des inferioren Temporalkortex stimulieren. Die Abbildung stammt aus [Tan93].

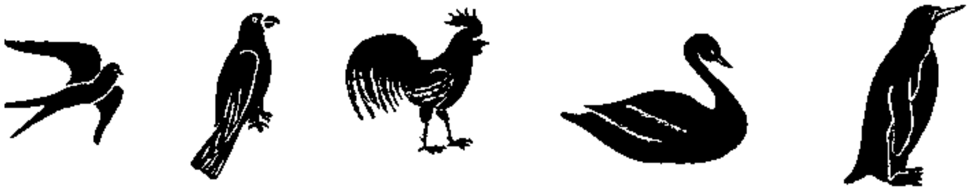


Abbildung 3.8: VERSCHIEDENE MITGLIEDER DES SCHEMAS „VOGEL“. Menschen nennen auf die Bitte, einen Vogel zu nennen, häufiger die Schwalbe als den Pinguin. In ihrem Denken ist die Schwalbe zum Prototyp des Schemas „Vogel“ geworden. Die Gemeinsamkeiten eines Pinguins mit dem Prototyp sind gering [Ros78]. Die Abbildung wurde aus [SS89] entnommen.

mit bereits gespeicherten Bildern entspricht, durch eine flexiblere Repräsentation - wie sie etwa in dieser Arbeit vorgestellt wird - zu ergänzen.

Schemata

Während die Zellen im hinteren Teil des inferioren Temporalkortex noch eine Selektivität für Stimuli *an bestimmten retinalen Positionen* aufweisen, stellt der vordere Teil des inferioren Temporalkortex die erste Stufe in der kortikalen Bildverarbeitung dar, bei der sich eine *Invarianz gegenüber der Stimulusposition* feststellen läßt [Tan93]. Mit höheren Verarbeitungsstufen entwickelt sich die Repräsentation eines gesehenen Objektes also weg von wahrnehmungsmäßigen Details hin zu einer abstrakteren, bedeutungsmäßigen Kodierung. Diese bedeutungsmäßige Repräsentation wird in der psychologischen Literatur mit dem Begriff „Schema“ bezeichnet. Schemata sind Wissenseinheiten, die typische Eigenschaften von Mitgliedern allgemeiner Kategorien durch Kombination ihrer Merkmale kodieren [And88], [SS89]. Für jedes Schema existiert ein beispielhaftes Exemplar, ein *Prototyp.* Die Klassenzugehörigkeit eines Objektes wird nach seiner Ähnlichkeit mit dem Prototyp bemessen (siehe Abbildung 3.8).

Empirische Befunde sprechen dafür, daß dabei zunächst Komponenten eines Objektes einzeln identifiziert und anschließend zu einem größeren Muster kombiniert werden. Die Bedeutungen, die durch Schemata kodiert werden, entspringen den Beziehungen zwischen den Einzelkomponenten [And88].

Schemata werden erworben, d.h. *gelernt* [SS89]. Jüngere Kinder beziehen sich in ihrer Vorstellung stärker auf wahrnehmungsmäßige, detaillierte Vorstellungsbilder, während Erwachsene eher von bedeutungsmäßigen Repräsentation ausgehen. Das wurde in [Kos80] durch Experimente belegt. Danach findet ein *Lernprozeß* von konkreten zu abstrakten Repräsentationen statt.

Begriffe und Begriffshierarchien

In der Psychologie werden nach [SS89] *Begriffe* als kognitive Einheiten definiert, in denen Gemeinsamkeiten von Personen und Gegenständen in Erscheinung treten. Nach der *Merkmalstheorie* [Kli78] entstehen Begriffe bzw. Klassen von Gegenständen durch eine Kombination von Merkmalen. Mit wachsendem Abstraktionsgrad fällt die Anzahl der zur Definition der Begriffe benötigten Merkmale. So können Begriffe innerhalb gleicher Klassen durch *Begriffshierarchien* nach ihrem Abstraktionsgrad geordnet werden. Nach [RMG76] ist die Zahl der Über- und Unterordnungen nicht beliebig. Vielmehr kommt es bei der natürlichen Begriffsbildung lediglich zum Aufbau von drei Abstraktionsebenen. Dabei wird die mittlere Ebene von einem „Grundbegriff“ besetzt (z.B. „Tisch“), der verallgemeinert („Möbel“) oder spezialisiert („Eßtisch“) werden kann. Auf die Bildung von Begriffshierarchien wird im Ausblick näher eingegangen.

Kapitel 4

Bananenwavelets

In diesem Kapitel werden erstmals die Begriffe „Bananenwavelet“, „Staude von Bananenwavelets“ und „Staudenstamm“ eingeführt und in die allgemeine Gaborwavelettheorie eingebettet. Dabei werden sich Gaborwavelets als Spezialfall der Bananenwavelets erweisen. In Abschnitt 4.1 erfolgt zunächst eine Definition der Bananenwavelets. Dann werden in Abschnitt 4.2 in Anlehnung an einen Satz von Gaborwavelets Stauden von Bananenwavelets eingeführt. In Abschnitt 4.3 wird die Existenz eines Mutterwavelets (der Staudenstamm), aus dem sich eine Staude berechnen läßt, bewiesen. Nachdem in Abschnitt 4.4 eine Erläuterung der implementatorischen Einzelheiten von Bananenwavelets erfolgte, wie z.B. die spezielle Wahl der charakteristischen Parameter einer Staude und eine Anpassung der Bananenformel, um bei der Transformation von der Bildhelligkeit unabhängige Ergebnisse zu erhalten, wird in Abschnitt 4.5 die Bananentransformation formal und anhand einiger Beispiele vorgestellt. Mit Ausführungen zur Implementierung der Bananentransformation in Abschnitt 4.6, insbesondere mit der Anwendung der Fouriertransformation auf Bilder und Wavelets, wird dieses Kapitel abgeschlossen.

4.1 Bananenwavelets als Verallgemeinerung von Gaborwavelets

Ein zweidimensionales Bananenwavelet $B_{(k,\alpha,c,s)}$ wird konstruiert als Produkt aus einer gekrümmten, in bestimmter Richtung orientierten, zweidimensionalen Gaußglocke und einer in gleicher Weise gekrümmten und orientierten ebenen Welle, wie es in Abbildung 4.1 dargestellt ist. Ein Bananenwavelet hat vier freie Parameter, deren Bedeutung nach der folgenden Definition erläutert wird.

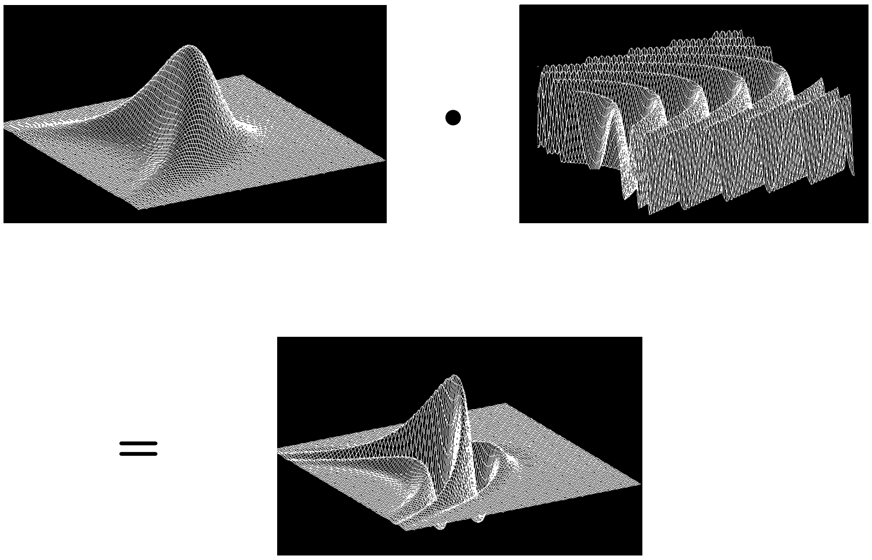


Abbildung 4.1: BANANENWAVELET. Ein Bananenwavelet ist das Produkt einer gekrümmten Gaußglocke mit einer gekrümmten, ebenen Welle. Da Bananenwavelets komplexe Funktionen sind, ist hier nur der Realteil dargestellt. Der Imaginärteil zeichnet sich durch eine Phasenverschiebung der Welle aus.

Definition 4.1 Ein *Bananenwavelet* $B_{(k,\alpha,c,s)}$ ist eine Abbildung

$$B_{(k,\alpha,c,s)} : \mathbb{R}^2 \rightarrow \mathbb{C}$$

$$B_{(k,\alpha,c,s)}(x,y) := \gamma \cdot \exp\left(-\frac{k^2}{2}\left(\left(\frac{x_s}{\sigma_x}\right)^2 + \left(\frac{y_s}{\sigma_y}\right)^2\right)\right) \cdot \exp\left(i \cdot k \cdot x_s\right) \tag{4.1}$$

mit

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} := \mathcal{S}_s\left(\mathcal{C}_c\left(\mathcal{M}_\alpha\begin{pmatrix} x \\ y \end{pmatrix}\right)\right) \tag{4.2}$$

und $k, \alpha, c, \gamma \in \mathbb{R}$ und $s \in \mathbb{R}\setminus\{0\}$, wobei gilt

$$\mathcal{M}_\alpha := \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix},$$

$$\mathcal{C}_c\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} x - c \cdot y^2 \\ y \end{pmatrix}$$

und

$$\mathcal{S}_s := \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{s} \end{pmatrix}.$$

Da in dieser Arbeit Bananenwavelets als lokale Filter verwendet werden (vergleiche mit den Ausführungen zu Gaborwavelets als lokale Filter in Abschnitt 2.1), wird im folgenden der Begriff „Bananenfilter“ als Synonym für „Bananenwavelet“ gebraucht.

k, α, c und s sind die charakteristischen Parameter eines Bananenfilters. k bestimmt die Frequenz der Welle. Die Orientierung der Welle und der gekrümmten Gaußglocke wird durch den Winkel α festgelegt, so daß Welle und Glocke die gleiche Orientierung aufweisen. c (*curvature*) gibt den Krümmungsgrad des Bananenwavelets an. Welle und Gaußglocke weisen ebenfalls stets die gleiche Krümmung auf. Schließlich wird durch s (*size*) die Größe des Wavelets bestimmt.
Stellt man Formel 4.1 in einer anderen Schreibweise dar, wird die Rolle des Parameters s verständlich. Formel 4.1 ist gleichbedeutend mit

$$B_{(k,\alpha,c,s)}(x,y) := \gamma \cdot \exp\left(-\frac{k^2}{2}\left(\left(\frac{x_c}{\sigma_x}\right)^2 + \left(\frac{y_c}{s \cdot \sigma_y}\right)^2\right)\right) \cdot \exp\left(i \cdot k \cdot x_c\right) \qquad (4.3)$$

mit

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} := \mathcal{C}_c\left(\mathcal{M}_\alpha \begin{pmatrix} x \\ y \end{pmatrix}\right), \qquad (4.4)$$

$\mathcal{C}_c$ und $\mathcal{M}_\alpha$ wie in Definition 4.1. Die Filtergröße wird also durch die Ausdehnung der Gaußglocke in y_c-Richtung bestimmt (vergleiche in Abbildung 4.3 z.B. die Realteile der beiden Filter $B_{(4\pi,0,1,1)}$ und $B_{(4\pi,0,1,\frac{3}{2})}$).
Aus Definition 4.1 wird ersichtlich, daß zunächst die die x-y-Ebene definierenden Koordinaten (x,y) um den Winkel α gedreht werden. Dadurch erhält man die gedrehten Koordinaten

$$\begin{pmatrix} x_\alpha \\ y_\alpha \end{pmatrix} := \mathcal{M}_\alpha \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos\alpha + y\sin\alpha \\ y\cos\alpha - x\sin\alpha \end{pmatrix}. \qquad (4.5)$$

Dann wird die x-Komponente des so erhaltenen Koordinatensystems mit dem Maß c gekrümmt, und man erhält die Koordinaten (x_c, y_c) (Gleichung 4.4). Danach erfolgt durch $\mathcal{S}_s$ eine Veränderung der Größe des Wavelets, woraus die Koordinaten (x_s, y_s) resultieren (Gleichung 4.2). Auf diese wird dann eine Gaborformel anwendet. Gaborwavelets sind wie folgt definiert (siehe auch Abbildung 2.2):

Definition 4.2 Die *Menge aller Gaborwavelets* läßt sich ausdrücken als

$$\mathcal{G} := \left\{G_{(k,\alpha)} : \mathbb{R}^2 \to \mathbb{C} \;\middle|\; k,\alpha \in \mathbb{R}\right\}$$

mit

$$G_{(k,\alpha)}(x,y) \;=\; \gamma \cdot \exp\left(-\frac{k^2}{2\sigma^2}\left(x_\alpha^2 + y_\alpha^2\right)\right) \cdot \exp\left(i \cdot \vec{k} \cdot \vec{r}\right), \qquad (4.6)$$

wobei $\begin{pmatrix} x_\alpha \\ y_\alpha \end{pmatrix}$ durch (4.5) definiert ist und

$$\vec{k} = k \cdot \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}, \qquad \vec{r} = \begin{pmatrix} x \\ y \end{pmatrix} \qquad \text{und} \qquad \gamma \in \mathbb{R}$$

gilt.

Der Hauptunterschied zwischen Gaborwavelet und Bananenwavelet besteht also in der zusätzlichen Einführung des Krümmungsgrades und der Größe. Ein weiterer Unterschied zum häufig verwendeten Gaborfilter, für den die Standardabweichungen der Gaußglocke in x- und y-Richtung gleich sind, besteht darin, daß beim Bananenfilter in der Regel $\sigma_x < \sigma_y$ gewählt wird (siehe auch Abschnitt 4.4.1).

Definition 4.3 Die *Menge aller Bananenwavelets* wird mit $\tilde{\mathcal{B}}$ bezeichnet:

$$\tilde{\mathcal{B}} := \left\{ B_{(k,\alpha,c,s)} : \mathbb{R}^2 \to \mathbb{C} \;\middle|\; k, \alpha, c \in \mathbb{R},\ \ s \in \mathbb{R}\setminus\{0\} \right\}$$

mit $B_{(k,\alpha,c,s)}$ gegeben durch Definition 4.1.

(In Abschnitt 4.2 wird eine Teilmenge $\mathcal{B}$ dieser Menge definiert.)

Bemerkung 4.1 Es gilt

$$\mathcal{G} \subset \tilde{\mathcal{B}}, \tag{4.7}$$

das heißt, die Menge der Bananenwavelets bildet eine Obermenge zur Menge der Gaborwavelets.

Beweis: Sei $G \in \mathcal{G}$. Setze $\sigma_x := \sigma, \sigma_y := \sigma, s := 1, c := 0$. Dann gilt

$$\begin{aligned} G(x,y) &= \gamma \cdot \exp\left(-\tfrac{k^2}{2\sigma^2}\left(x_\alpha^2 + y_\alpha^2\right)\right) \cdot \exp\left(i \cdot (k \cdot x \cdot \cos\alpha + k \cdot y \cdot \sin\alpha)\right) \\ &= \gamma \cdot \exp\left(-\tfrac{k^2}{2}\left(\left(\tfrac{x_\alpha}{\sigma}\right)^2 + \left(\tfrac{y_\alpha}{\sigma}\right)^2\right)\right) \cdot \exp\left(i \cdot k \cdot x_\alpha\right) \quad \in \quad \tilde{\mathcal{B}} \end{aligned}$$

$\square$

Bemerkung 4.2 Außerdem gilt offensichtlich $B_{(k,\alpha,0,1)} \in \mathcal{G}$, falls $\sigma_x = \sigma_y$.

(Siehe in Abbildung 4.3 jeweils die Filter mit $c = 0$.)

4.2 Stauden von Bananenwavelets

Ebenso wie man bei der Anwendung von Gaborwavelets in der Regel eine ganze Reihe verwandter Wavelets benutzt, kann man auch hier eine Menge von Bananenwavelets definieren:

Definition 4.4 Eine diskrete Menge von Bananenwavelets

$$\mathcal{B} := \left\{ B_{(k,\alpha,c,s)} : \mathbb{R}^2 \to \mathbb{C} \;\middle|\; \begin{array}{l} k = k_1 \,, \ldots, k_{\mathtt{L}} \\ \alpha = \alpha_1 \,, \ldots, \alpha_{\mathtt{D}} \\ c = c_1 \,, \ldots, c_{\mathtt{B}} \\ s = s_1 \,, \ldots, s_{\mathtt{M}} \end{array} \right\} \tag{4.8}$$

mit $\mathtt{L}, \mathtt{D}, \mathtt{B}, \mathtt{M} \in \mathbb{N}\backslash\{0\}$, mit vorgegebenen $k_1, \ldots, k_{\mathtt{L}}$, $\alpha_1, \ldots, \alpha_{\mathtt{D}}$, $c_1, \ldots, c_{\mathtt{B}}$, $s_1, \ldots, s_{\mathtt{M}}$, und $B_{(k,\alpha,c,s)}$ wie in Definition 4.1 heißt eine *Staude von Bananenwavelets.*

Die Werte `L`, `D`, `B` und `M` sind dann die Anzahlen der verschiedenen Frequenzen, Orientierungen, Krümmungen bzw. Größen der Wavelets einer Staude. Für die Implementierung einer Staude wurde der Definitionsbereich der Bananenwavelets in der Weise eingeschränkt, daß gilt

$$\mathcal{B} = \left\{ B_{(k,\alpha,c,s)} : [-a,a]^2 \to \mathbb{C} \;\middle|\; \begin{array}{l} k = k_1 \,, \ldots, k_{\mathtt{L}} \\ \alpha = \alpha_1 \,, \ldots, \alpha_{\mathtt{D}} \\ c = c_1 \,, \ldots, c_{\mathtt{B}} \\ s = s_1 \,, \ldots, s_{\mathtt{M}} \\ a \in \mathbb{R}^+ \end{array} \right\} \tag{4.9}$$

mit $B_{(k,\alpha,c,s)}$ wie in Definition 4.1. Abbildung 4.2 zeigt den Real- und Imaginärteil eines speziellen Bananenwavelets einer Staude.

Abbildung 4.2: Real- und Imaginärteil eines Bananenwavelets. Links ist der Real-, rechts der Imaginärteil des Bananenwavelets $B_{\left(\frac{4\pi}{\sqrt{2}},0,-\frac{1}{2},1\right)}$ einer speziellen Staude abgebildet (vergleiche mit Abbildung 4.3). Bei der konkreten Implementierung wurde $[-63,64]^2$ als Definitionsbereich des Bananenwavelets gewählt. Zur Wahl des Parametertupels (k,α,c,s) siehe Abschnitt 4.4.1.

Abbildung 4.3 zeigt eine Staude von Bananenwavelets in Ausschnitten um die jeweiligen Mittelpunkte $(x_0, y_0) = (0,0)$ der einzelnen Wavelets.

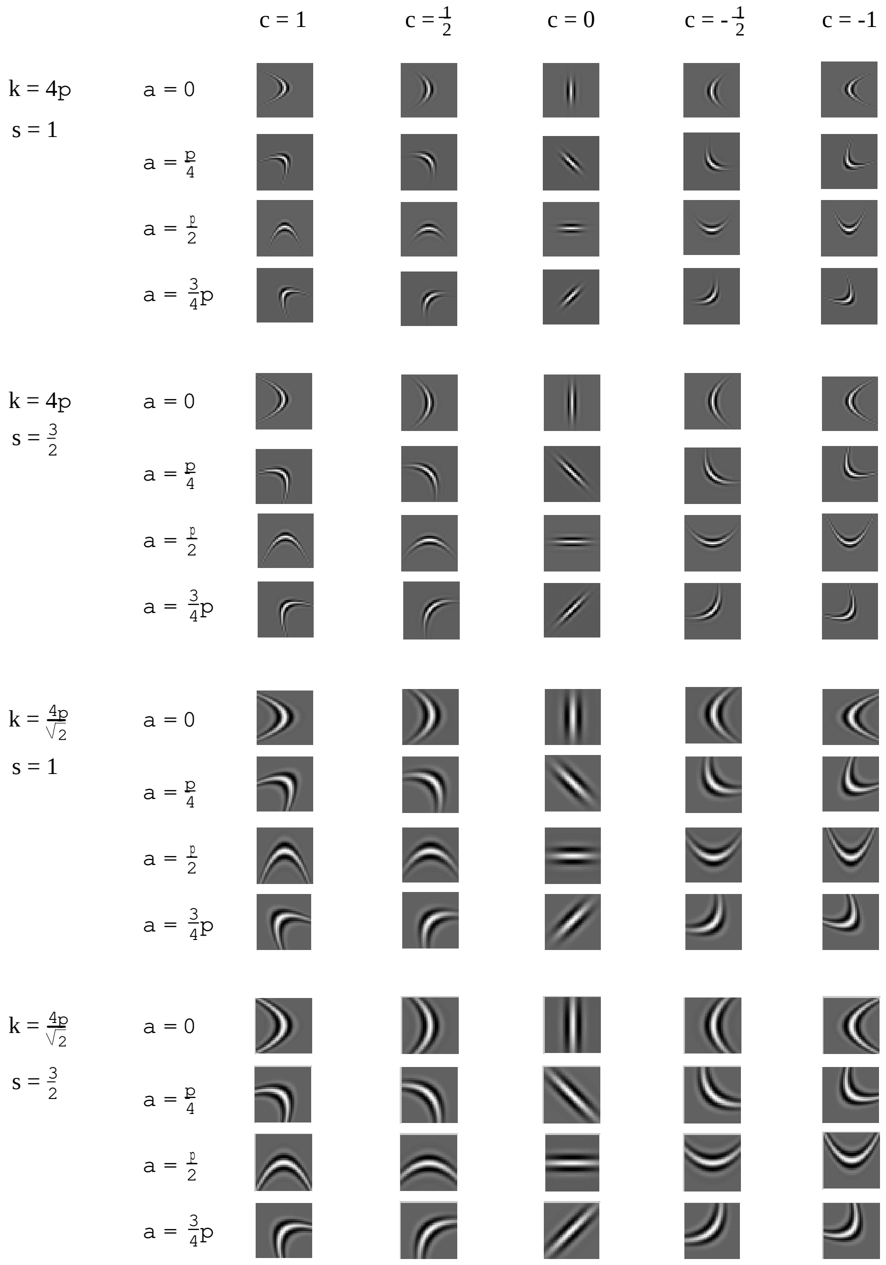


Abbildung 4.3: Staude von Bananenwavelets. Diese Staude besteht aus Bananenwavelets, die zwei verschiedene Frequenzen (`L` = 2), vier verschiedene Orientierungen (`D` = 4), fünf verschiedene Krümmungen (`B` = 5) und zwei verschiedene Größen aufweisen (`M` = 2). Die einzelnen Filter geben die Realteile der Funktionen $B_{(k,\alpha,c,s)}$ für die jeweils angegebenen Parameter k, α, c und s wieder, wobei weiße Pixel hohe Funktionswerte und schwarze Pixel niedrige Funktionswerte bedeuten. Die Abbildungen geben aus Gründen der Übersichtlichkeit lediglich Ausschnitte um den Mittelpunkt der Bananenwavelets wieder. Ein vollständiges Bananenwavelet ist in Abbildung 4.2 dargestellt. Es entspricht hier dem Wavelet in der 4. Spalte und 9. Zeile. Die Werte der Parameter entstehen aus der Wahl der Laufindizes im Algorithmus zur Erzeugung der Filter (siehe Definition 4.5).

4.3 Staudenstamm

In der Gabortheorie wird ein Satz von Wavelets durch ein sogenanntes *Mutterwavelet* erzeugt. Dieses Mutterwavelet ist für einen Gaborfiltersatz der Filter $G_{(0,1)}$. Es wird nun gezeigt, daß auch eine Staude von Bananenwavelets ein erzeugendes Wavelet hat: den *Staudenstamm.* Wählt man den Bananenfilter mit den Parametern $k = 1$, $\alpha = 0$, $c = 0$ und $s = 1$ als Staudenstamm, so läßt sich ein Bananenwavelet mit beliebigem Parametersatz (k, α, c, s) aus $B_{(1,0,0,1)}$ durch Rotation, Krümmung und Stauchung bzw. Streckung erzeugen.

Definition und Satz 4.1 Das Bananenwavelet $B_{(1,0,0,1)} \in \mathcal{B}$ heißt *Staudenstamm der Staude* $\mathcal{B}$. Ein beliebiges Bananenwavelet $B_{(k,\alpha,c,s)} \in \mathcal{B}$ läßt sich aus $B_{(1,0,0,1)}$ erzeugen nach

$$B_{(k,\alpha,c,s)}(x,y) \quad = \quad B_{(1,0,0,1)}\left(k \cdot \mathcal{S}_s\left(\mathcal{C}_c\left(\mathcal{M}_\alpha\begin{pmatrix} x \\ y \end{pmatrix}\right)\right)\right)$$

mit $\mathcal{S}_s$, $\mathcal{C}_c$ und $\mathcal{M}_\alpha$ wie in Definition 4.1.

Beweis:

$$\begin{aligned} & B_{(1,0,0,1)}\left(k \cdot \mathcal{S}_s \cdot \mathcal{C}_c\left(\mathcal{M}_\alpha\begin{pmatrix} x \\ y \end{pmatrix}\right)\right) \\ = \quad & \gamma \cdot \exp\left(-\frac{1}{2}\left(\left(\frac{kx_s}{\sigma_x}\right)^2 + \left(\frac{ky_s}{\sigma_y}\right)^2\right)\right) \cdot \exp\left(i \cdot k \cdot x_s\right) \\ = \quad & B_{(k,\alpha,c,s)}(x,y) \end{aligned}$$

□

4.4 Implementierung von Bananenwavelets

Die Parameter der Bananenwavelets müssen im Hinblick auf ihre Anwendung als Detektoren von gekrümmten Kanten angemessen gewählt werden. Die Wahl der Parameter wird im ersten Teil dieses Abschnitts vorgestellt. Desweiteren muß bei der Implementierung beachtet werden, die Bananenwavelets *DC-frei* zu wählen, um Robustheit gegen variierende Bildhelligkeiten zu gewährleisten. Dieses wird im zweiten Teil dieses Abschnitts näher beschrieben.

4.4.1 Wahl der charakteristischen Parameter einer Staude

Definition 4.5 Die Wahl des Parametertupels (k, α, c, s) eines Bananenwavelets $B_{(k,\alpha,c,s)}$ erfolgt nach foldenden Vorschriften:

$$\begin{array}{rcll} k &=& k_{max} \cdot k_{fak}^{l} & mit \quad l = 0,\ldots,\mathtt{L}-1, \\ \alpha &=& \Delta\alpha \cdot d & mit \quad d = 0,\ldots,\mathtt{D}-1, \\ c &=& \begin{cases} 0 & \text{für } \mathtt{B} = 1 \\ 1 - \frac{2b}{\mathtt{B}-1} & \text{sonst} \end{cases} & mit \quad b = 0,\ldots,\mathtt{B}-1, \\ s &=& 1 + \frac{m}{2} & mit \quad m = 0,\ldots,\mathtt{M}-1 \end{array}$$

mit $k_{max} = 4\pi$, $k_{fak} = \frac{1}{\sqrt{2}}$, $\Delta\alpha = \frac{\pi}{\mathtt{D}}$ und L, D, B und M wie in Definition 4.4. Das Tupel (l, d, b, m) heißt *Erzeugendentupel* des Bananenwavelets $B_{(k,\alpha,c,s)}$.

Die Laufindizes l, d, b und m laufen also bei Null beginnend über die Anzahl der Frequenzen, Richtungen, Krümmungen bzw. Größen. Werden für die Laufindizes z.B. folgende Werte gewählt

$$l = 0, \ldots, 1\ , \quad d = 0, \ldots, 7\ , \quad b = 0, \ldots, 10 \quad \text{und} \quad m = 0, \ldots, 2,$$

so besteht eine Staude aus $2 \cdot 8 \cdot 11 \cdot 3 = 528$ Bananenwavelets, die zwei verschiedene Frequenzen, acht verschiedene Richtungen, elf verschiedene Krümmungen und drei verschiedene Größen aufweisen (siehe auch Abbildung 4.3).
Außerdem wurden folgende Werte festgelegt: $\sigma_x = \pi$ und $\sigma_y = 2\pi$ (siehe Definition 4.1) und $a = 64$ (siehe Gleichung 4.9).
Der Vorfaktor γ in Definition 4.1 sollte so gewählt werden, daß Unterschiede im Antwortverhalten von Filtern verschiedener Größe und Frequenz kompensiert werden. Hier wurde

$$\gamma = \frac{k^2}{s^2 \cdot \left(\frac{\sigma_x + \sigma_y}{2}\right)^2} \tag{4.10}$$

gewählt.

4.4.2 Unabhängigkeit von Variationen in der Bildhelligkeit

Die Rolle von Variationen in der Helligkeit verschiedener Bilder kann bereits auf der Ebene der Bildvorverarbeitung minimiert werden. Zu diesem Zweck wird anstelle der Formel 4.1 folgende Formel verwendet:

$$B_{(k,\alpha,c,s)}(x,y) = \gamma \cdot \exp\left(-\frac{k^2}{2}\left(\left(\frac{x_s}{\sigma_x}\right)^2 + \left(\frac{y_s}{\sigma_y}\right)^2\right)\right) \cdot \left[\exp\left(ikx_s\right) - \mathrm{D}_{(k,\alpha,c,s)}\right] \tag{4.11}$$

mit γ siehe Gleichung 4.10, wobei $\mathrm{D}_{(k,\alpha,c,s)}$, der *DC-Part*, so gewählt wird, daß

$$\int\limits_{\mathbb{R}^2} B_{(k,\alpha,c,s)}\left(x,y\right)\ dxdy = 0 \tag{4.12}$$

gilt, d.h. der Bananenfilter *mittelwertfrei* oder *DC-frei* ist. Gleichung 4.12 ist gleichbedeutend damit, daß die nullte Fourierkomponente[1] verschwindet:

$$\int\limits_{\mathbb{R}^2} B_{(k,\alpha,c,s)}\left(x,y\right)\ dxdy = 0 \qquad \Longleftrightarrow \qquad \hat{B}_{(k,\alpha,c,s)}\left(\vec{0}\right) = 0.$$

Aus den Gleichungen 4.11 und 4.12 folgt

$$\mathrm{D}_{(k,\alpha,c,s)} \quad = \quad \frac{\int_{\mathbb{R}^2} \exp\left(f\left(x_s, y_s\right)\right) \cdot \exp\left(i \cdot k \cdot x_s\right)\ dxdy}{\int_{\mathbb{R}^2} \exp\left(f\left(x_s, y_s\right)\right)\ dxdy}$$

mit

$$f\left(x_s, y_s\right) \quad = \quad -\frac{k^2}{2}\left(\left(\frac{x_s}{\sigma_x}\right)^2 + \left(\frac{y_s}{\sigma_y}\right)^2\right).$$

---

[1] Zur Fouriertransformation siehe Abschnitt 4.6.

4.5 Bananentransformation

In diesem Abschnitt wird die Anwendung der Bananenwavelets als lokale Filter für Grauwertbilder beschrieben.

4.5.1 Bildtransformation mit Bananenwavelets

Zunächst soll ein Grauwertbild definiert werden.

Definition 4.6 Ein *Grauwertbild* ist eine meßbare Abbildung

$$I : \mathcal{D} \longrightarrow \mathbb{R}$$

mit $\mathcal{D} \subseteq \mathbb{R}^2$.

Da alle im Rahmen dieser Arbeit verwendeten Bilder die gleiche, quadratische Größe aufweisen, wird der Definitionsbereich aller verwendeten Bilder im folgenden mit $\mathcal{D}$ bezeichnet.
$\mathcal{D}$ ist dabei von der Form $\{0, \ldots, \mathtt{N}-1\} \times \{0, \ldots, \mathtt{N}-1\}$, $\mathcal{D} \neq \emptyset$. Für die konkrete Implementierung wurde $\mathtt{N} = 128$ gewählt, und der Wertebereich wurde eingeschränkt auf $\{0, \ldots, 255\}$.

Die Bildtransformation mit Bananenfiltern wird im folgenden als *Bananentransformation* bezeichnet. Dabei wird eine Staude von Bananenwavelets auf ein Grauwertbild angewendet, indem an jedem Bildpunkt die Antwort eines jeden Bananenwavelets aus der Staude berechnet wird. Die Antwort berechnet sich als Faltungsprodukt aus Filter und Bild. Das Faltungsprodukt zweier Funktionen f und g berechnet sich nach

$$(f * g)(x_0, y_0) = \int\limits_{\mathbb{R}^2} f(x, y)\, g(x_0 - x, y_0 - y)\; dxdy.$$

Nun kann die Antwort eines Bananenfilters definiert werden:

Definition 4.7 Die *Antwort* $\mathcal{J}_{(k,\alpha,c,s)}$ *eines Bananenfilters* $B_{(k,\alpha,c,s)}$ an einem Aufpunkt (x_0, y_0) eines Grauwertbildes $I(x, y)$ ist die Konvolution des Bildes mit dem Filter:

$$\begin{aligned} \mathcal{J} \quad &: \quad I \longrightarrow \mathbb{C} \\ \mathcal{J}_{(k,\alpha,c,s)}(x_0, y_0) \quad &:= \quad \left(I * B_{(k,\alpha,c,s)}\right)(x_0, y_0) && (4.13) \\ &= \quad \int\limits_{\mathbb{R}^2} I(x, y)\, B_{(k,\alpha,c,s)}(x_0 - x, y_0 - y)\; dxdy. && (4.14) \end{aligned}$$

Diese Art der Definition einer Filterantwort ist analog zur Bildverarbeitung mit Gaborfiltern, die z.B. in [LVB^+92] ausführlich beschrieben ist. Die Faltung in Definition 4.7 läßt sich als Korrelation zwischen Filter und Bild interpretieren. Zur Veranschaulichung sei auf die Abbildungen in Abschnitt 4.5.2, insbesondere Abbildung 4.4, verwiesen. (Zur Behandlung der Bild- und Filterränder siehe Abschnitt 4.6.)

4.5.2 Antwortverhalten von Bananenwavelets bei der Anwendung auf natürliche Bilder

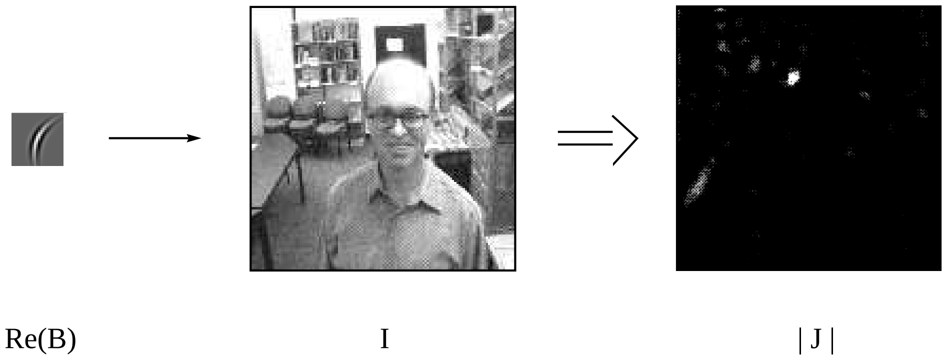


Abbildung 4.4: Beispiel für das Antwortverhalten eines Bananenwavelets. Diese Abbildung wird im Text erläutert.

Bei der Transformation eines Grauwertbildes I mit einem Bananenwavelet erhält man eine komplexe Funktion - die Funktion $\mathcal{J}$ aus Definition 4.7 - die genau in denjenigen Punkten (x, y) des Bildes I lokale Maxima annimmt, in denen eine Struktur vorhanden ist, die der Frequenz, der Orientierung, der Krümmung und der Größe des Wavelets besonders nahe kommt. Da in dieser Arbeit für Bananenwavelets die Standardabweichung σ_x der Gaußglocke in x-Richtung im Vergleich zur Standardabweichung σ_y in y-Richtung klein gewählt wird (siehe Abschnitt 4.4.1), und in einem Wavelet daher nur sehr wenige Wellenzüge vorhanden sind (siehe Abbildung 4.3), erscheint es gerechtfertigt, diese Strukturen, die von Bananenwavelets erkannt werden, als *Kanten* zu bezeichnen. Bananenfilter erweisen sich also bei der Anwendung auf natürliche Bilder als *Detektoren gekrümmter Kanten.*

In Abbildung 4.4 ist das Antwortverhalten eines Bananenwavelets anhand eines Beispielbildes dargestellt. Wird die in Definition 4.7 erklärte Konvolution für das Bananenwavelet B, dessen Realteil links dargestellt ist, und das Bild I durchgeführt, so erhält man als Betrag der Funktion $\mathcal{J}$ die im rechten Bild gezeigte Darstellung. Weiße Stellen im Bild bedeuten hohe Absolutwerte der Funktion $\mathcal{J}$. Bei einem Vergleich des Faltungsergebnisses mit dem ursprünglichen Bild I erkennt man an vielen Stellen lokale Maxima der Betragsfunktion $|\mathcal{J}|$. Die höchsten befinden sich jedoch dort, wo Strukturen vorhanden sind, die der Frequenz, Richtung, Krümmung und Größe des verwendeten Filters B sehr ähnlich sind wie z.B. links am Kopf der Person.

Eine systematischere Darstellung des Antwortverhaltens befindet sich in Abbildung 4.5. An der im Beispielbild gekennzeichneten Landmarke wurde eine Konvolution mit allen Filtern einer Staude durchgeführt. In den Diagrammen sind die Beträge der Funktion $\mathcal{J}$ für diese Landmarke dargestellt. Jeder Punkt in den Diagrammen steht für die Antwort eines bestimmten Filters der Staude. Die verwendete Staude besteht hier aus einer Frequenz ($\mathtt{L} = 1$), acht Orientierungen ($\mathtt{D} = 8$), fünfzehn Krümmungen ($\mathtt{B} = 15$) und drei Größen ($\mathtt{M} = 3$). Für jede Richtung wurde jeweils ein Diagramm erstellt. Auf den Abszissen sind jeweils alle Krümmungen *und* alle Größen aufgetragen. Daher geben jeweils drei benachbarte Punkte die Antworten für *eine* Krümmung und *drei* verschiedene Größen wieder. Zum Beispiel stehen die ersten drei Punkte im ersten Diagramm für die Absolutantworten der Filter mit der Orientierung $d = 0$, der Krümmung $b = 0$ und den Größen $m = 0$, $m = 1$ und $m = 2$. Die nächsten drei Punkte kodieren die Antworten für die Krümmung $b = 1$ und die Größen $m = 0$, $m = 1$ und $m = 2$ usw.

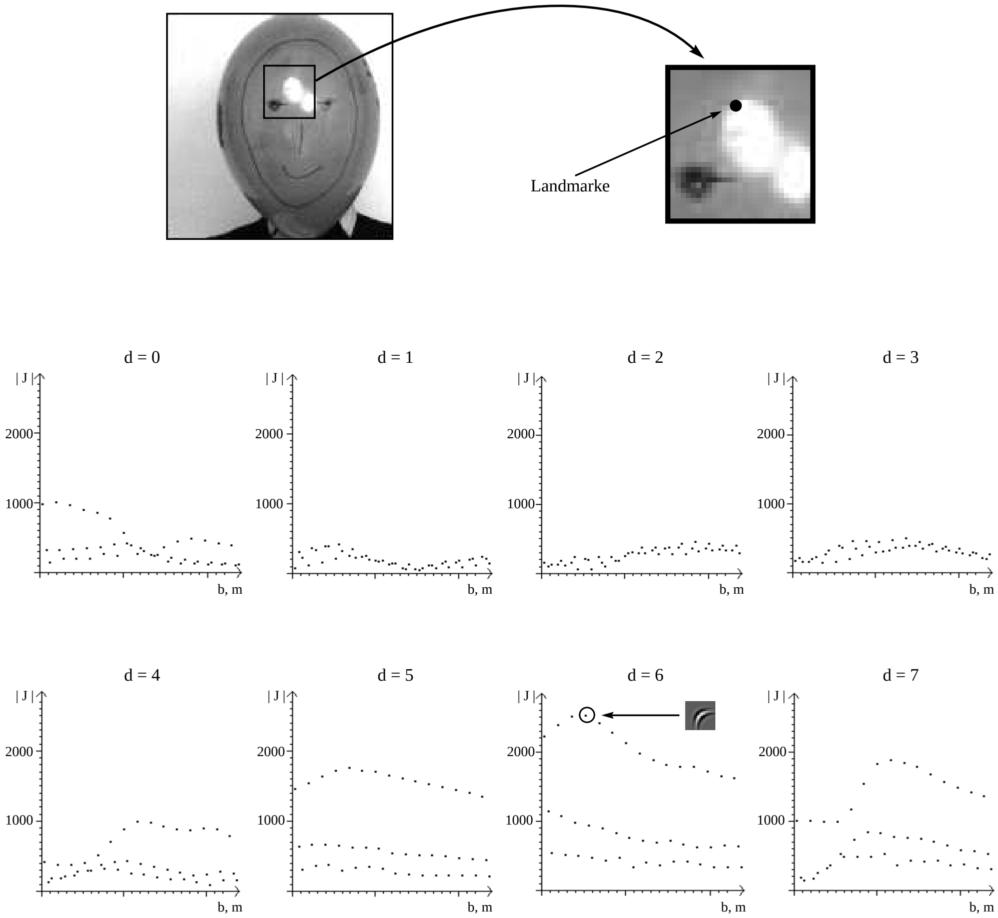


Abbildung 4.5: Antworten einer Staude an einer Landmarke. Um die Präzision der Bananenwavelets als Detektoren gekrümmter Kanten darzustellen, wurde dieses einfache Bild als Beispiel gewählt. Die Diagramme werden im Text erläutert.

Zu erkennen ist hier das stetige Antwortverhalten der Filter einer Staude bzgl. aller Parameter. Das absolute Maximum von $|\mathcal{J}|$ liegt für das Bananenwavelet mit der Richtung $d = 6$, der Krümmung $b = 3$ und der Größe $m = 0$ vor. Es ist dasjenige Wavelet der Staude, das die Struktur im betrachteten Bildpunkt am besten beschreibt. Im entsprechenden Diagramm ist es gekennzeichnet.

Hier bleibt noch anzumerken, daß Bananenwavelets, die durch Formel 4.11 beschrieben werden, gegen Helligkeitsvariationen robuste Filter sind. Das bedeutet, daß Beleuchtungsunterschiede, die das Bild als Ganzes in seiner Helligkeit verändern, bereits durch die Bildvorverarbeitung ausgeglichen werden. Beleuchtungsvariationen, die verschiedene Schatten oder Reflexe auf den aufgenommenen Objekten verursachen, werden erst später durch den Lernalgorithmus kompensiert.

4.6 Implementierung der Bananentransformation

Bild und Filter werden vor der Konvolution einer *Fouriertransformation* unterzogen. In dieser Arbeit wird die Fouriertransformation

$$\hat{f}\left(\vec{\omega}\right) = \frac{1}{2\pi} \int\limits_{\mathbb{R}^2} f\left(\vec{x}\right) e^{i\vec{\omega}\vec{x}}\, d^2\vec{x}$$

verwendet. $\hat{f}$ ist die *Fouriertransformierte* von f. Die *inverse Transformation* wird beschrieben durch

$$f\left(\vec{x}\right) = \frac{1}{2\pi} \int\limits_{\mathbb{R}^2} \hat{f}\left(\vec{\omega}\right) e^{-i\vec{x}\vec{\omega}}\, d^2\vec{\omega}.$$

Ebenfalls benötigt wird der

Faltungssatz Die Fouriertransformierte des Faltungsprodukts zweier Funktionen ist bis auf den Faktor 2π gleich dem Produkt der Fouriertransformierten der einzelnen Faktoren:

$$\frac{1}{2\pi} \int\limits_{\mathbb{R}^2} \left(f * g\right)\left(\vec{x}\right) e^{i\vec{\omega}\vec{x}}\, d^2\vec{x} = 2\pi \cdot \hat{f}\left(\vec{\omega}\right) \cdot \hat{g}\left(\vec{\omega}\right).$$

Da es sich bei Gleichung 4.13 um eine Faltung handelt, läßt sich $\mathcal{J}_{(k,\alpha,c,s)}$ jetzt einfacher berechnen, indem man die Fouriertransformierten von I und $B_{(k,\alpha,c,s)}$ errechnet (siehe Abbildung 4.6). Dann löst sich das Integral in Gleichung 4.14 nach dem Faltungssatz in einer einfachen Multiplikation auf:

$$\hat{\mathcal{J}}_{(k,\alpha,c,s)}\left(\omega_1,\omega_2\right) = 2\pi \cdot \hat{I}\left(\omega_1,\omega_2\right) \cdot \hat{B}_{(k,\alpha,c,s)}\left(\omega_1,\omega_2\right). \tag{4.15}$$

Anschließend genügt es, für $\hat{\mathcal{J}}_{(k,\alpha,c,s)}$ die inverse Transformation vorzunehmen. Diese Überlegungen wurden bei der Implementierung der Bananentransformation berücksichtigt, da sich hierdurch eine enorme Zeitersparnis erzielen läßt. Die Berechnung des Integrals 4.14 erfordert nämlich einen Aufwand in der Größenordnung von $O\left(n^2\right)$, wenn n die Größe des Filters ist, während der Weg über Formel 4.15 nur $O\left(n \cdot \log n\right)$ Operationen benötigt, wenn zur Berechnung der Fouriertransformierten die *FFT* (Fast Fourier Transform) angewendet wird [PTV^+94]. Eine Darstellung der beiden Methoden befindet sich in Abbildung 4.7.

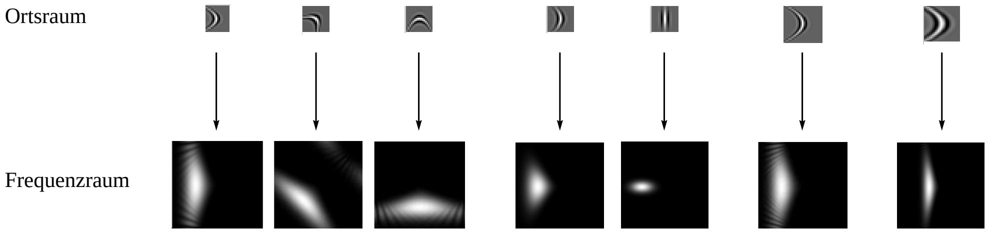


Abbildung 4.6: EINIGE BANANENWAVELETS NACH DER FOURIERTRANSFORMATION. Im Ortsraum ist jeweils der Realteil der Filter in einem Ausschnitt um ihren Mittelpunkt dargestellt (vergleiche auch mit Abbildung 4.3). Im Frequenzraum ist jeweils der Betrag der transformierten Wavelets abgebildet.

Bei der Berechnung der Fouriertransformierten $\hat{I}$ von I und $\hat{B}$ von B wurden periodische Fortsetzungen von I und B benutzt. Dabei grenzt jeweils der untere an den oberen, der linke an den rechten Bildrand (Torus). Die Multiplikation der Faltung wird dann lediglich für die Größe der Bilder durchgeführt. Randeffekte spielen für den im folgenden beschriebenen Algorithmus jedoch keine Rolle, da die betrachteten Bildausschnitte nicht davon berührt werden.

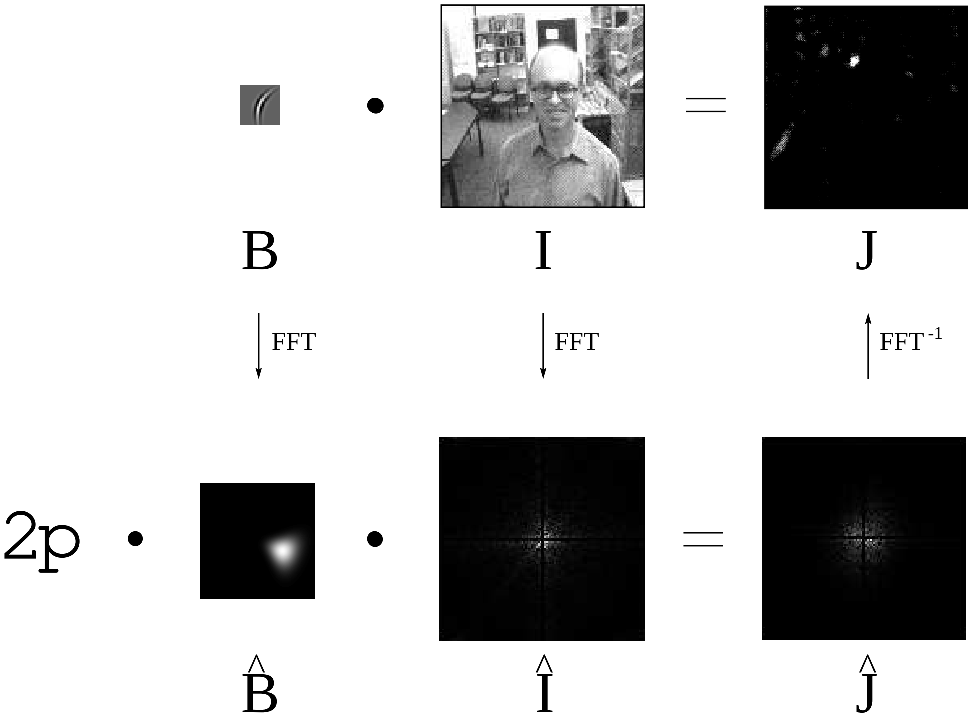


Abbildung 4.7: Bananentransformation im Fourierraum.
Obere Bilderzeile: Die Bananentransformation im Ortsraum, wie sie durch Definition 4.7 beschrieben wird. (Dargestellt ist nur der Realteil des Filters und der Betrag des Ergebnisses.)
Untere Bilderzeile: Die Bananentransformation im Frequenzraum, wie sie durch Formel 4.15 beschrieben wird. Zuerst werden die Fouriertransformierten $\hat{B}$ und $\hat{I}$ vom Filter B und vom Bild I berechnet. Als Ergebnis der Multiplikation in Formel 4.15 erhält man $\hat{\mathcal{J}}$. (Hier ist nur der Realteil von $\hat{B}$, der Betrag von $\hat{I}$ und der Betrag von $\hat{\mathcal{J}}$ dargestellt.) Anschließend wird $\hat{\mathcal{J}}$ der inversen Transformation unterzogen. Man erhält das gleiche Ergebnis wie bei der aufwendigeren Berechnung des Integrals im Ortsraum.

Kapitel 5

Lernen lokaler Merkmale

In diesem Kapitel wird ein Algorithmus vorgestellt, der anhand einiger Bilder von Objekten einer bestimmten Objektklasse typische, ansichtsabhängige Eigenschaften dieser Objektklasse lernt. Eine bearbeitete Objektklasse ist z.B. die Klasse der Getränkedosen (siehe Kapitel 6). Die dargebotenen Vertreter einer Objektklasse dienen dabei als Beispiele der Klasse, die auf Gemeinsamkeiten hin untersucht werden. Die so gewonnenen Merkmale können dann als Repräsentanten der entsprechenden Objektklasse betrachtet werden. Es ist *nicht* nötig, daß *alle* präsentierten Beispielobjekte charakteristisch für ihre Klasse sind, da der Algorithmus durch ein statistisches Verfahren, genauer: durch eine Clusteranalyse (siehe Abschnitt 5.4.2), nur diejenigen Merkmale extrahiert, die einem *Großteil* der Vertreter gemeinsam sind. Insbesondere werden durch die statistische Analyse Invarianzen ausgeblendet, die z.B. durch verschiedenartige Beleuchtungen der Objekte (Reflexe oder Schattenbildung), unterschiedliche Texturen und Hintergründe, teilweise Verdeckungen und leicht verschiedene Aufnahmewinkel und -entfernungen entstehen, sofern sie nicht regelmäßig auftreten.
Da pro Klassenvertreter jeweils nur einige lokale Stellen betrachtet werden, ist im folgenden vom „Lernen lokaler Merkmale“ die Rede.

5.1 Überblick

Auf jedem der als Beispiele präsentierten Bilder befindet sich ein relevantes Objekt aus der Klasse, deren typische Eigenschaften gelernt werden sollen. Da jeweils nur lokale Stellen der Objekte betrachtet werden sollen, werden pro Trainingsbild die Landmarken, auf deren Umgebung[1] sich die weitere Untersuchung konzentrieren soll, vorgegeben (siehe Abbildung 3.2). Das Lernen lokaler Merkmale geschieht dann in mehreren Schritten. Die folgende Algorithmusbeschreibung ist sehr allgemein formuliert und wird später konkretisiert werden (jeweils durch Rahmungen gekennzeichnet). Außerdem befindet sich in Abbildung 5.1 eine schematische Darstellung des Algorithmus, auf die in diesem Kapitel wiederholt hingewiesen werden wird.

---

[1] Im folgenden wird der Begriff „Bildausschnitt“ synonym zu „Landmarkenumgebung“ verwendet.

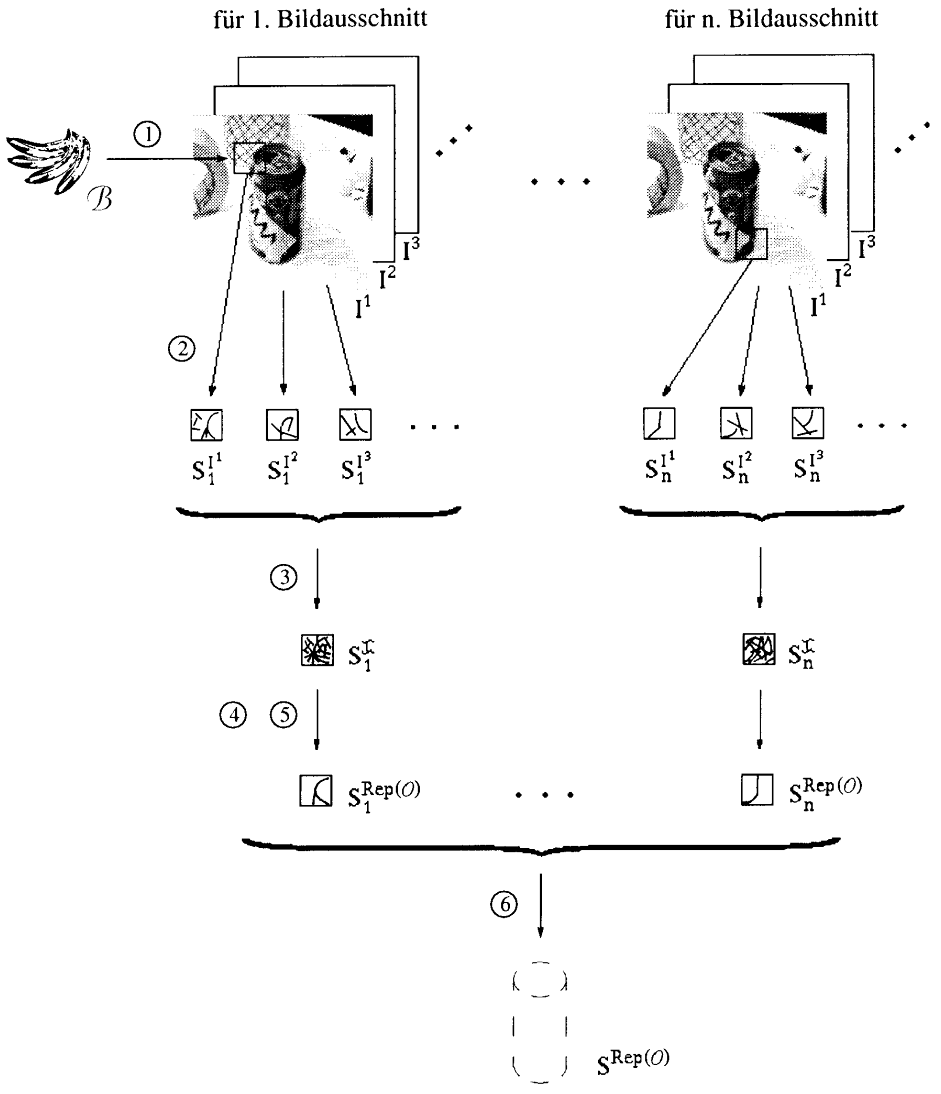


Abbildung 5.1: LERNALGORITHMUS. Die eingekreisten Nummern bezeichnen die Schritte des Algorithmus, wie er in diesem Kapitel beschrieben wird.

Lernalgorithmus:

1. Zuerst wird ein Bildausschnitt mit einer Staude von Bananenwavelets transformiert, d.h. jeder Filter der Staude wird mit dem Bildausschnitt korreliert, und zwar mit jeweils jedem Pixel des Bildausschnitts als Aufpunkt (wird in Abschnitt 5.2 konkretisiert).
2. Aus den so erhaltenen Filterantworten werden die „starken“ ausgewählt. Das sind diejenigen, deren Ursprung ein signifikantes Merkmal[2] im Bildausschnitt war (wird in Abschnitt 5.3 konkretisiert).
3. Nachdem Punkt 1 und 2 für alle Trainingsbilder mit dem entsprechenden Bildausschnitt durchgeführt wurden, werden die jeweils ausgewählten Filterantworten zu einer Menge zusammengefaßt (wird in Abschnitt 5.4.1 konkretisiert).
4. Diese Menge wird daraufhin einer Clusteranalyse unterzogen, bei der sich wenige, große Cluster und sehr viele, kleine Cluster ergeben werden (wird in Abschnitt 5.4.2 konkretisiert).
5. Die zu lernende Repräsentation für den betrachteten Ausschnitt ergibt sich schließlich aus den Centroiden der großen Cluster (wird in Abschnitt 5.4.3 konkretisiert).
6. Die Punkte 1 bis 5 werden für alle Bildausschnitte wiederholt. Die Repräsentation einer Objektklasse wird dann aus den Ausschnittrepräsentationen gebildet (wird ebenfalls in Abschnitt 5.4.3 konkretisiert).

5.2 Durchführung und Ergebnis der Bananentransformation

Wie aus dem ersten Punkt des Lernalgorithmus hervorgeht, wird zunächst ein Bild I mit einer Staude $\mathcal{B}$ von Bananenwavelets transformiert. Wenn das geschehen ist, liegen pro Bildpunkt (x, y) so viele Filterantworten vor, wie Bananenfilter in der Staude vorhanden sind. Da jeder Pixel des Bildes als Aufpunkt verwendet wird, erhält man als Resultat einer Bananentransformation die komplexen Zahlen $\mathcal{J}_{(k,\alpha,c,s)}\,(x, y)$ für alle k, α, c, s aus der Staude und für alle $(x, y) \in \mathcal{D}$ (siehe Definition 4.7).
Wenn (l, d, b, m) das Erzeugendentupel (siehe Definition 4.5) des Bananenwavelets $B_{(k,\alpha,c,s)}$ ist, kann ein 6-Tupel (x, y, l, d, b, m) als Punkt in einem sechsdimensionalen, metrischen Raum interpretiert werden, der jetzt definiert werden soll.

---

[2] Siehe Definition 5.6 eines signifikanten Merkmals.

Definition 5.1 Sei $\mathcal{B}$ eine Staude von Bananenwavelets und $\mathtt{L},\ \mathtt{D},\ \mathtt{B},\ \mathtt{M} \in \mathbb{N} \setminus \{0\}$ wie in Definition 4.4. Sei ferner $\mathcal{D} = \{0, \ldots, \mathtt{N}-1\} \times \{0, \ldots, \mathtt{N}-1\}$ der Definitionsbereich aller Bilder. Es wird folgende Menge definiert:

$$\mathcal{P} := \left\{ \quad (x,y,l,d,b,m) \quad \middle| \quad \begin{array}{ll} x,y \in & \{0,\ldots,\mathtt{N}-1\} \\ l \quad \in & \{0,\ldots,\mathtt{L}-1\} \\ d \quad \in & \{0,\ldots,\mathtt{D}-1\} \\ b \quad \in & \{0,\ldots,\mathtt{B}-1\} \\ m \quad \in & \{0,\ldots,\mathtt{M}-1\} \end{array} \right\}.$$

Durch die Abbildung

$$\begin{array}{rcl} & d : \mathcal{P} \times \mathcal{P} \to \mathbb{R} & \\ d\,(p_i, p_j) & := & \min\left(\tilde{d}\,(p_i, p_j)\,,\, \tilde{d}\,(p_i, \hat{p}_j)\right) \end{array}$$

wird $\mathcal{P}$ zu einem metrischen Raum. $\mathcal{P}$ heißt *Parameterraum der Bananentransformation.* Dabei gilt

$$\begin{array}{rcl} & \tilde{d} : \hat{\mathcal{P}} \times \hat{\mathcal{P}} \to \mathbb{R} & \\ \tilde{d}\,(\hat{p}_i, \hat{p}_j) & := & \Bigg[\quad \left(\frac{x_i - x_j}{\mathtt{N}}\right)^2 + \left(\frac{y_i - y_j}{\mathtt{N}}\right)^2 + \left(\frac{l_i - l_j}{\mathtt{L}}\right)^2 + \\ & & \left(\frac{\hat{d}_i - \hat{d}_j}{\mathtt{D}}\right)^2 + \left(\frac{\hat{b}_i - \hat{b}_j}{\mathtt{B}}\right)^2 + \left(\frac{m_i - m_j}{\mathtt{M}}\right)^2 \quad \Bigg]^{\frac{1}{2}} \end{array}$$

und

$$\hat{p} \;:=\; \left(x, y, l, \hat{d}, \hat{b}, m\right) \quad \forall \;\; p = (x,y,l,d,b,m) \in \mathcal{P}$$

mit

$$\hat{d} = \begin{cases} d + \mathtt{D} & \text{für } d < \frac{\mathtt{D}}{2} \\ d - \mathtt{D} & \text{sonst} \end{cases} \qquad \textit{und} \qquad \hat{b} \;:=\; \mathtt{B} - b - 1.$$

$\hat{\mathcal{P}}$ ist eine Erweiterung von $\mathcal{P}$, die durch die Definition von $\hat{p}$ erklärt wird[3]. $\hat{\mathcal{P}}$ wird durch die Abbildung $\tilde{d}$ zu einem metrischen Raum.

Die Metrik $\tilde{d}$ des Raumes $\hat{\mathcal{P}}$ spielt für die Definition der Metrik d des Raumes $\mathcal{P}$ eines Hilfsrolle. Bei der Distanzberechnung zweier Punkte $p_i, p_j \in \mathcal{P}$ muß die besondere Struktur des Raumes $\mathcal{P}$ berücksichtigt werden. Diese Struktur läßt sich bezüglich der Koordinaten d und b als *Möbiusband* beschreiben (siehe Abbildung 5.2). Würde für die Abstandsberechnung lediglich die „direkte" Metrik $\tilde{d}$ benutzt, würden sich inkorrekte Distanzen ergeben. Abbildung 5.3 erklärt dieses Phänomen an einem Beispiel. Die Hilfsmetrik $\tilde{d}$ ist eine der euklidischen Metrik verwandte Metrik, bei der für jede einzelne Dimension der Abstand zweier Werte auf die Ausdehnung der entsprechenden Dimension normiert wird (jeweils die Nenner in den quadrierten Summanden). Da die Intervallgrenzen der einzelnen Dimensionen von $\mathcal{P}$ in verschiedenen Größenordnungen liegen können, wird diese Normierung vorgenommen, um unterschiedliche Dimensionsausdehnungen bei der Abstandsberechnung auszugleichen. Dadurch erhalten alle Dimensionen die gleiche Gewichtung.

---

[3] Genauer:

$$\hat{\mathcal{P}} := \left\{ \quad \left(x,y,l,\hat{d},\hat{b},m\right) \quad \middle| \quad \begin{array}{ll} x,y \in & \{\;\; 0,\ldots,\mathtt{N}-1 \;\;\} \\ l \quad \in & \{\;\; 0,\ldots,\mathtt{L}-1 \;\;\} \\ \hat{d} \quad \in & \{-\mathtt{D},\ldots,2\mathtt{D}-1\} \\ \hat{b} \quad \in & \{\;\; 0,\ldots,\mathtt{B}-1 \;\;\} \\ m \quad \in & \{\;\; 0,\ldots,\mathtt{M}-1 \;\;\} \end{array} \right\}.$$

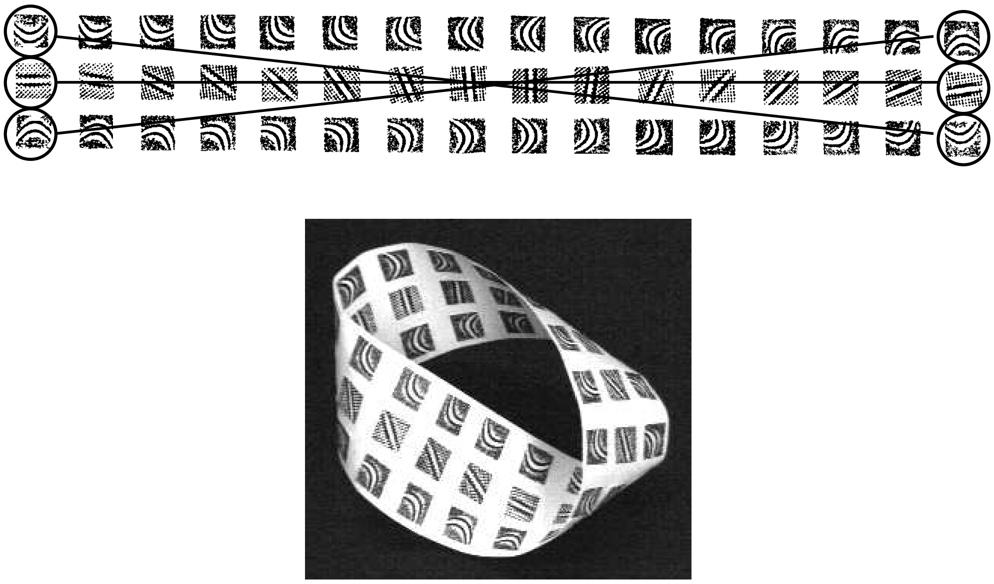


Abbildung 5.2: MÖBIUSBAND DES PARAMETERRAUMS. Der Unterraum des Parameterraums, der durch die Koordinaten d (Orientierung) und b (Krümmung) gegeben ist, besitzt die Struktur eines Möbiusbandes. Zur Veranschaulichung wurde hier eine Staude mit einer Frequenz (`L` $= 1$), 16 Orientierungen (`D` $= 16$), drei Krümmungen (`B` $= 3$) und einer Größe gewählt (`M` $= 1$). In der oberen Abbildung sind die Bananenwavelets auf der linken Seite durch Linien mit ihren jeweils benachbarten Wavelets auf der rechten Seite verbunden. Verbindet man diese Nachbarn miteinander, so erhält man das Möbiusband in der unteren Abbildung. (Siehe auch Abbildung 5.3.)

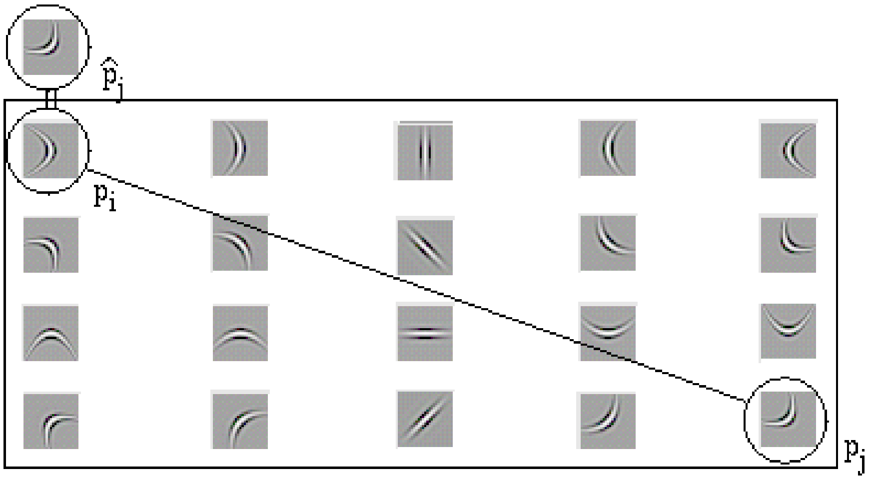


Abbildung 5.3: METRIK IM PARAMETERRAUM. Die Bananenwavelets in der Box sind ein Ausschnitt aus Abbildung 4.3. Sei $p_j = (x_j, y_j, l_j, d_j, b_j, m_j) \in \mathcal{P}$ dargestellt durch das Bananenwavelet mit dem Erzeugendentupel (l_j, d_j, b_j, m_j) bei festem (x_j, y_j) (unten rechts eingekreist), $p_i \in \mathcal{P}$ entsprechend. Die Hilfsmetrik $\tilde{d}$ würde zwischen p_i und p_j den Abstand berechnen, der durch die Diagonale gekennzeichnet ist. Die wahre Distanz beträgt jedoch lediglich $d\,(p_i, p_j)$ und ist durch die Doppellinie (oben links) gekennzeichnet. p_j ist zuvor in die neuen Koordinaten $\hat{p}_j \in \hat{\mathcal{P}}$ transformiert worden. (Siehe auch Abbildung 5.2.)

Da $\mathcal{P}$ ein metrischer Raum ist, läßt sich eine (später benötigte) *offene Kugel* definieren:

Definition 5.2 Sei $\tilde{p} \in \mathcal{P}$ ein Punkt und $r > 0$. Dann heißt

$$K(\tilde{p}, r) \quad := \quad \{\, p \in \mathcal{P} \mid d(\tilde{p}, p) \; < \; r \,\}$$

die *offene Kugel* mit Mittelpunkt $\tilde{p}$ und Radius r bzgl. der Metrik d.

Da für das weitere Vorgehen lediglich der Absolutbetrag von $\mathcal{J}_{(k,\alpha,c,s)}(x,y)$ (siehe Definition 4.7) benötigt wird und anhand *einer ganzen Reihe* von Trainingsbildern eine Repräsentation einer Objektklasse gelernt werden soll, werden die nächsten beiden Definitionen benötigt:

Definition 5.3 Sei (l, d, b, m) das Erzeugendentupel des Bananenwavelets $B_{(k,\alpha,c,s)}$. Die *Absolutantwort eines Bananenwavelets* $B_{(k,\alpha,c,s)}$ *an einem Bildpunkt* $(x,y) \in \mathcal{D}$ berechnet sich als Funktionswert der Abbildung

$$\mathcal{A} : \mathcal{P} \longrightarrow \mathbb{R}^+$$

$$\mathcal{A}(x, y, l, d, b, m) \; := \; \left| \, \mathcal{J}_{(k,\alpha,c,s)}(x,y) \, \right|$$

mit $\mathcal{J}_{(k,\alpha,c,s)}$ wie in Definition 4.7.

Definition 5.4 Die *Menge aller Trainingsbilder* $\mathcal{I}$ wird definiert als

$$\mathcal{I} := \bigcup_{k=1}^{K} I^k,$$

wobei K die Anzahl aller Trainingsbilder I^k ist.

Für jedes Trainingsbild $I^k \in \mathcal{I}$ werden Bildausschnitte vorgegeben, auf die sich der Lernalgorithmus bei seiner Suche nach Merkmalen, die für die untersuchte Objektklasse relevant sind, konzentrieren soll. Die Konstellation dieser Ausschnitte ist demnach objektklassenspezifisch. Für die Objektklasse „Getränkedose“ ist die Anordnung der Bildausschnitte für die gewählte Ansicht z.B. so wie in Abbildung 3.2 aus Abschnitt 3.2.2 dargestellt. Der j-te Bildausschnitt wird definiert über seinen Mittelpunkt (x_j, y_j) und seine halbe Seitenlänge r_j:

Definition 5.5 Sei I^k ein Grauwertbild aus der Menge aller Trainingsbilder $\mathcal{I}$, $\mathcal{D}$ der Definitionsbereich[4] von I^k, $(x_j, y_j) \in \mathcal{D}$ und $r_j \in \mathbb{N}^+$. Sei ferner

$$\mathcal{D}_j \;:=\; \{\, x_j - r_j \,, \ldots, \, x_j + r_j \,\} \;\times\; \{\, y_j - r_j \,, \ldots, \, y_j + r_j \,\}$$

so, daß $\mathcal{D}_j \subseteq \mathcal{D}$. Der *j-te Ausschnitt des Bildes* I^k ist dann die Einschränkung von I^k auf $\mathcal{D}_j$:

$$I_j^k \;:=\; I^k\Big|_{\mathcal{D}_j}.$$

Dabei ist $k = 1, \ldots, K$ und $j = 1, \ldots, n$ mit $K = |\mathcal{I}|$ und $n =$ Anzahl der Ausschnitte pro Trainingsbild.

Ein Punkt (x_j, y_j) bezeichnet in jedem Bild die jeweils korrespondierende Landmarke. In jedem Trainingsbild kann (x_j, y_j) jedoch unterschiedliche Werte annehmen (zum Korrespondenzproblem siehe Abschnitt 2.2, insbesondere Abbildung 2.5).
Mit dieser Definition kann nun die erste Konkretisierung des Lernalgorithmus erfolgen (siehe auch Abbildung 5.1):

1. Wende eine Staude $\mathcal{B}$ von Bananenwavelets auf den betrachteten Bildausschnitt I_j^k an und berechne die Funktionswerte der Funktion $\mathcal{A}$.

---

[4] Siehe die Bemerkungen nach Definition 4.6.

5.3 Extraktion der signifikanten Merkmale eines Trainingsbildes

Pro betrachtetem Bildausschnitt I_j^k werden für alle Trainingsbilder I^k deren signifikanten Merkmale ermittelt. Das können sowohl Kennzeichen sein, die für die betrachtete Objektklasse Allgemeingültigkeit besitzen und deshalb zu den tatsächlich zu lernenden Merkmalen gehören, wie z.B. typische Objektkanten, als auch solche Charakteristika, die nur spezifisch für das aktuelle Trainingsbild sind und später durch den Lernprozeß außer acht gelassen werden, wie z.B. Auffälligkeiten im Hintergrund (siehe Abbildung 3.1).

5.3.1 Signifikante Merkmale eines Trainingsbildes

Ein *Merkmal* eines Bildausschnitts I_j^k soll ein Punkt $p = (x, y, l, d, b, m)$ aus dem Parameterraum $\mathcal{P}_j$ sein. $\mathcal{P}_j$ ist der Parameterraum gemäß Definition 5.1, bei dem der Definitionsbereich $\mathcal{D}$ nun der des Bildausschnitts I_j^k ist. $\mathcal{A}(p)$ kann dann als Gewichtung dieses Merkmals interpretiert werden. Eine mögliche Interpretation kann z.B. lauten: „Je größer $\mathcal{A}(p)$ ist, desto mehr spricht dafür, daß an der Stelle (x, y) im Bildausschnitt eine Kante mit der Schärfe k, der Orientierung α, der Krümmung c und der Größe s vorliegt.“ (zur Interpretation von Bananenwavelets als Kantendetektoren siehe Abschnitt 4.5.2).

Ein solches Merkmal p soll *signifikant* heißen, wenn erstens $\mathcal{A}(p)$ eine bestimmte Schwelle überschreitet (dann wird die Gewichtung als ausreichend betrachtet) und wenn zweitens die Funktion $\mathcal{A}$ in p ein Maximum in einer *Umgebung einer bestimmten Größe* annimmt. Diese Umgebung ist durch eine offene Kugel um p gegeben (siehe Definition 5.2). Das bloße Vorhandensein eines *lokalen* Maximums in p genügt nicht. Damit soll verhindert werden, daß zwei signifikante Merkmale einen beliebig kleinen Abstand zueinander haben können. Ein signifikantes Merkmal soll sich ausreichend von den anderen unterscheiden. Da für jeden Bildausschnitt I_j^k mehrere signifikante Merkmale auftreten können, werden sie numeriert.

Die soeben geschilderten Sachverhalte finden in der folgenden Definition ihren Ausdruck.

Definition 5.6 Ein Punkt $s_{ijk} \in \mathcal{P}_j$ heißt *i-tes signifikantes Merkmal des j-ten Bildausschnitts I_j^k des k-ten Bildes $I^k \in \mathcal{I}$* mit $i = 1, \dots m(I_j^k)$, falls gilt

i) $\mathcal{A}(s_{ijk}) > s_1$ für eine feste Schwelle $s_1 \in \mathbb{R}^+$ und

ii) $\mathcal{A}(s_{ijk}) \geq \mathcal{A}(p) \quad \forall \ p \in K(s_{ijk}, s)$ für einen festen Radius $s \in \mathbb{R}^{+}$[5].

Nun werden alle signifikanten Merkmale eines Bildausschnitts I_j^k zu einer Menge zusammengefaßt:

Definition 5.7 Die *Menge aller signifikanten Merkmale eines Bildausschnitts I_j^k eines Trainingsbildes I^k* ist die Menge

$$\mathcal{S}_j^{I^k} := \left\{ \; s_{ijk} \in \mathcal{P}_j \; \middle| \; \begin{array}{l} s_{ijk} \text{ ist signifikantes Merkmal des Bildausschnitts } I_j^k \\ \text{des Trainingsbildes } I^k, \ i = 1, \dots m(I_j^k) \end{array} \right\}.$$

---

[5] In dem unwahrscheinlichen Fall, daß in einer offenen Kugel zwei Punkte p_1 und p_2 liegen, die eine *genau gleiche* Absolutantwort liefern ($\mathcal{A}(p_1) = \mathcal{A}(p_2)$), und für die außerdem Eigenschaft i) gilt, gelten nach ii) natürlich beide als signifikantes Merkmal. Um einen ausreichend großen Abstand zweier signifikanter Merkmale auch in diesem Fall zu gewährleisten, könnte man sich als Verbesserung vorstellen, hier nur einen der Punkte p_1 und p_2 auszuwählen oder den Durchschnitt zwischen p_1 und p_2 zu wählen.

Eine Veranschaulichung dieser Menge findet man in Abbildung 5.1, in Experimenten erhaltene Ergebnisse signifikanter Merkmale sind in Kapitel 6, Abbildungen 6.3 und 6.7 dargestellt.
An dieser Stelle kann der zweite Schritt des Lernalgorithmus konkretisiert werden:

2. Berechne die Menge $\mathcal{S}_j^{I^k}$ für den gewählten Bildausschnitt I_j^k des Trainingsbildes I^k. Das wiederhole für jedes Trainingsbild aus $\mathcal{I}$.

5.3.2 Algorithmische Realisierung

Um die Menge $\mathcal{S}_j^{I^k}$ zu berechnen, werden bei der Implementierung die beiden Eigenschaften aus Definition 5.6 algorithmisch umgesetzt.
Für die Berechnung der Schwelle s_1 aus Eigenschaft i) wird zunächst ein Wert w berechnet:

$$w \;=\; \text{round}\,\Big(q_1 \;\cdot\; \mathtt{N}^2 \cdot \mathtt{L} \cdot \mathtt{D} \cdot \mathtt{B} \cdot \mathtt{M}\Big) \tag{5.1}$$

mit $q_1 \in (0,1)$, $\mathtt{N}$ wie in Definition 5.1 und den anderen Größen wie in Definition 4.4. q_1 bezeichnet dabei den Prozentanteil der Werte, die über der Schwelle liegen sollen, bezogen auf das Volumen von $\mathcal{P}_j$[6]. Mit „round“ wird die kaufmännische Rundungsfunktion bezeichnet. Seien nun die Absolutantworten $\mathcal{A}\,(.)$ in absteigender Reihenfolge angeordnet:

$$\mathcal{A}\,(p_1) \geq \mathcal{A}\,(p_2) \geq \ldots \geq \mathcal{A}\,(p_{\mathtt{N}^2 \cdot \mathtt{L} \cdot \mathtt{D} \cdot \mathtt{B} \cdot \mathtt{M}})\,.$$

Dann berechnet sich s_1 aus Eigenschaft i) zu

$$s_1 \quad = \quad \mathcal{A}\,(p_w)\,.$$

Für die Bedingung ii) wurde $s := 1$ gesetzt. Die Kugel K wurde bei der Implementierung durch einen sechsdimensionalen Quader mit jeweils einem Pixel als Kantenlänge angenähert[7].
Die Umgebung um ein potentielles[8], signifikantes Merkmal s_{ijk}, die durch die offene Kugel $K\,(s_{ijk}, s)$ beschrieben wird, wird auf mögliche weitere signifikante Merkmale $\tilde{s}_{ijk}$ hin untersucht. Falls es in $K\,(s_{ijk}, s)$ kein $\tilde{s}_{ijk}$ mit $\mathcal{A}\,(\tilde{s}_{ijk}) \geq \mathcal{A}\,(s_{ijk})$ gibt, wird s_{ijk} als signifikantes Merkmal anerkannt.

5.4 Lernen aus Beispielen

Das Grundprinzip des Lernens anhand von verschiedenen, dargebotenen Beispielen ist das Trennen der „wichtigen“, in vielen Beispielen wiederkehrenden Information, die gelernt werden soll, von der „unwichtigen“ Information, die für die Objektklasse nicht relevant ist.
Der „Informationspool“ besteht an dieser Stelle des Algorithmus sowohl aus den *wichtigen*, signifikanten Merkmalen eines Bildausschnitts als auch aus den für die gestellte Aufgabe

---

[6] q_1 muß mindestens so groß gewählt werden, daß nicht $w = 0$ eintritt. Zur konkreten Wahl von q_1 siehe Gleichung 6.2 in Kapitel 6.

[7] Es ist auch denkbar, s ebenso wie s_1 in Abhängigkeit von $\mathcal{P}_j$ zu bestimmen.

[8] Potentiell: aufgrund von Eigenschaft i)

unwichtigen, signifikanten Merkmalen eines Ausschnitts, die alle aus der gesamten Galerie der Trainingsbilder extrahiert wurden. Dieser Informationspool wird in Abschnitt 5.4.1 definiert.

Unter der Prämisse, daß objektklassentypische Merkmale sich in den meisten Trainingsbildern ähneln und somit häufig auftreten und auffällige Merkmale, die nicht objektklassenspezifisch sind (z.B. Auffälligkeiten im Hintergrund), von Beispiel zu Beispiel stärker variieren, sich somit nicht stark ähneln und seltener auftreten, werden die bis hierher erhaltenen signifikanten Merkmale aller Trainingsbilder einer Ähnlichkeits- und Häufigkeitsanalyse unterzogen. Diese wird in Abschnitt 5.4.2 erläutert.

5.4.1 Zusammenfassung der signifikanten Merkmale einzelner Trainingsbilder

An dieser Stelle kann folgende wichtige Vereinigung erfolgen:

Definition 5.8 Die *Vereinigung aller signifikanten Merkmale* $\mathcal{S}_j^{I^k}$ *des j-ten Bildausschnitts aller Trainigsbilder* $\mathcal{I}$ ist die Menge

$$\mathcal{S}_j^{\mathcal{I}} := \bigcup_{k=1}^{K} \mathcal{S}_j^{I^k}.$$

für $j = 1, \ldots, n$.

Die Menge $\mathcal{S}_j^{\mathcal{I}}$ entsteht also dadurch, daß in jedem Trainingsbild der jeweils korrespondierende Bildausschnitt betrachtet wird und alle dort gefundenen signifikanten Merkmale eingesammelt werden. Zur Veranschaulichung siehe Abbildung 5.1.

Nun kann der dritte Schritt des Lernalgorithmus eine Konkretisierung erfahren:

3. Bilde die Vereinigungsmenge $\mathcal{S}_j^{\mathcal{I}}$ für den j-ten Bildausschnitt aller Trainingsbilder $\mathcal{I}$.

5.4.2 Clusteranalyse der signifikanten Merkmale

Bevor auf die Einzelheiten der hier durchgeführten, speziellen Clusteranalyse eingegangen wird, werden zunächst clusteranalytische Verfahren im allgemeinen beschrieben. Hinter dem Begriff „Clusteranalyse“ verbirgt sich eine Vielzahl verschiedener, heuristischer Verfahren, die (meist große) Datenmengen auf wenige, überschaubare Interpretationseinheiten reduzieren. Die Elemente der Datenmenge[9] werden dabei nach bestimmten Prinzipien möglichst optimal in Klassen oder Gruppen, sogenannte *Cluster*, eingeordnet. Das soll so geschehen, daß Elemente innerhalb eines Clusters eine große Ähnlichkeit aufweisen, sich jedoch von denen anderer Cluster gut separieren lassen. Außerdem soll jedes Element genau einem Cluster angehören.

Eine völlig optimale Zuordnung ist bei einer realistischen Datenmenge jedoch nicht möglich. Dieses würde erfordern, alle möglichen Aufteilungen (*Partitionierungen*, siehe Definition 5.9) der Datenmenge zu berechnen. Die Anzahl der möglichen Partitionierungen wächst jedoch exponentiell mit der Zahl der aufzuteilenden Elemente. Bei n Elementen, die auf k Cluster aufzuteilen sind, gibt es

$$S\,(n,k)=\frac{1}{k!}\cdot\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n$$

Möglichkeiten. Die $S\,(n,k)$ heißen *Stirling'sche Zahlen zweiter Art*. Da die Anzahl der Cluster ebenfalls variieren kann, geben die *Bell'schen Zahlen*

$$B_n=\sum_{k=1}^{n}S\,(n,k)$$

die Anzahl aller möglichen Partitionierungen an [SL77]. Schon für 20 Elemente ergeben sich so über 51 Billionen mögliche Aufteilungen. Aus diesem Grund verwendet man Methoden, die zu Partitionierungen führen, die meist nur lokale Maxima des Optimierungskriteriums bilden.

Die Einordnung der Elemente in bestimmte Cluster erfolgt nach Maßgabe ihrer Ähnlichkeit, die aufgrund bestimmter Kriterien bestimmt wird[10]. Neben der Auswahl der Zuordnungskriterien (s.u.) kommt also dem Ähnlichkeitsmaß eine entscheidende Rolle für das Ergebnis einer Clusteranalyse zu. Für die Wahl des Ähnlichkeitsmaßes sollen inhaltliche Überlegungen ausschlaggebend sein, d.h. die inhaltlich wichtig erscheinenden Ähnlichkeitsaspekte sollten durch das verwendete Maß formal abgebildet werden.

Ein weiterer wichtiger Punkt ist die Vorschrift, nach der eine Zuordnung eines Elementes zu einem Cluster erfolgt. Man kann dabei zwischen hierarchischen und nicht-hierarchischen Verfahren unterscheiden. Im Unterschied zu den nicht-hierarchischen Verfahren bleibt bei den hierarchischen Verfahren eine einmal getroffene Zuordnung eines Elementes zu einem Cluster bis ans Ende der Analyse erhalten. Die hierarchischen Verfahren lassen sich noch einmal unterteilen in Verfahren, bei denen zu Beginn alle Elemente ein einziges, großes Cluster bilden, das dann sukzessive in kleinere aufgespaltet wird, und solchen, bei denen mit der feinsten Partitionierung[11] begonnen wird und nach und nach Elemente und kleinere Cluster zu größeren Clustern zusammengefaßt werden. Dabei reduziert sich mit jedem

---

[9] Im Falle dieser Arbeit werden dies die Elemente der Menge $\mathcal{S}_j^{\mathcal{I}}$ sein, also die signifikanten Merkmale.

[10] In dem hier zur Diskussion stehenden Fall wird die Ähnlichkeit zweier Punkte aus $\mathcal{P}_j$ mit Hilfe der Metrik d bestimmt (siehe Definition 5.1).

[11] Feinste Partitionierung: Jedes Element bildet ein eigenes Cluster.

Schritt die Anzahl der Cluster um eins [Bor85].

Diese zuletzt beschriebene Methode nennt man auch „agglomerative“ Clusteranalyse, und eine solche wurde für den hier zu beschreibenden Algorithmus verwendet. Wie bereits oben angedeutet, ist hier die zu analysierende Datenmenge die Menge $\mathcal{S}_j^{\mathcal{I}}$, die als Punktwolke in $\mathcal{P}_j$ gedeutet werden kann. Diese Punktwolke soll nun sinnvoll in mehrere, kleinere Punktwolken gegliedert werden. Anders ausgedrückt: die signifikanten Merkmale sollen in eine Menge von *wichtigen* (zu lernenden) und in eine Menge von *unwichtigen* Merkmalen, die nicht zu lernenden Auffälligkeiten in einzelnen Trainingsbildern entsprechen, eingeteilt werden.
Die Elemente der Menge $\mathcal{S}_j^{\mathcal{I}}$ werden auf ihre gegenseitige Ähnlichkeit hin untersucht. Elemente, die einander ähneln, werden zu Teilmengen von $\mathcal{S}_j^{\mathcal{I}}$, eben den Clustern, zusammengefaßt. Dabei wird die Ähnlichkeit zwischen zwei signifikanten Merkmalen über die Metrik d (siehe Definition 5.1) definiert. Jedes Element wird auf diese Weise genau einem Cluster zugewiesen, so daß man als Ergebnis eine Partitionierung der Menge $\mathcal{S}_j^{\mathcal{I}}$ erhält:

Definition 5.9 Eine *Partitionierung* der Menge $\mathcal{S}_j^{\mathcal{I}}$ ist ein ein System $\mathcal{C}_j$ von Teilmengen von $\mathcal{S}_j^{\mathcal{I}}$:

$$C_{1j}, \ldots, C_{tj} \in \mathcal{C}_j, \quad t \in \mathbb{N} \text{ Anzahl der Cluster,}$$

mit

1. $C_{1j}, \ldots, C_{tj} \subset \mathcal{S}_j^{\mathcal{I}}$,
2. $C_{rj} \cap C_{r'j} = \phi \quad \forall\, r \neq r', \quad r, r' \in \{1, \ldots, t\}$ (paarw. Disjunktheit) und
3. $\bigcup_{r=1}^{t} C_{rj} = \mathcal{S}_j^{\mathcal{I}}$.

Als Schreibweise wird festgelegt

$$\bigcup \mathcal{C}_j := \bigcup_{r=1}^{t} C_{rj}.$$

Für diese Arbeit wurde nun kein allgemeingültiges, aus der einschlägigen Literatur bekanntes Verfahren verwendet, sondern ein auf das spezielle, vorliegende Problem zugeschnittener Clusterbildungsalgorithmus entwickelt. Das ist zulässig aufgrund der Beschaffenheit der zu clusternden Daten. Aus dem Algorithmus zur Generierung der signifikanten Merkmale folgt die Kenntnis der prinzipiellen Qualität der Daten, nämlich das Vorhandensein weniger, großer, dichter Cluster und vieler, kleiner, wenig dichter Cluster. Aus Gründen eines geringeren Rechenaufwandes gegenüber herkömmlichen Verfahren erfolgt die Zuordnung der Elemente von $\mathcal{S}_j^{\mathcal{I}}$ zu einem Cluster nun nach folgendem Algorithmus:

Clusteralgorithmus:

1. Initialisiere eine Zählervariable ($l := 1$).
2. Wähle ein beliebiges Element $s \in \mathcal{S}_j^{\mathcal{I}}$ und bilde daraus ein neues (einelementiges) Cluster C_{lj}.
3. Füge sukzessive alle übriggebliebenen Elemente $\tilde{s} \in \mathcal{S}_j^{\mathcal{I}}$ zu C_{lj} hinzu, für die gilt:
   $$\exists \;\; s \in C_{lj} \;\; : \quad d\,(\tilde{s}, s) \; < \; s_2$$
   mit einer festen Schwelle $s_2 \in \mathbb{R}^+$.
4. Inkrementiere die Zählervariable ($l := l + 1$) und wiederhole die Schritte 2 und 3 mit den jetzt noch übriggebliebenen Elementen aus $\mathcal{S}_j^{\mathcal{I}}$.
5. Führe die Schritte 2 bis 4 solange aus, bis alle Elemente aus $\mathcal{S}_j^{\mathcal{I}}$ einem Cluster zugeordnet wurden.

Bei der algorithmischen Realisierung wurde für Schritt 3

$$s_2 \quad = \quad q_2 \, \cdot \, \mathtt{N}^2 \cdot \mathtt{L} \cdot \mathtt{D} \cdot \mathtt{B} \cdot \mathtt{M} \tag{5.2}$$

gewählt (vergleiche mit Gleichung 5.1) mit $q_2 \in (0, 1)$, `N` wie in Definition 5.1 und den anderen Größen wie in Definition 4.4. $q_2 \in (0, 1)$ ist der Prozentanteil, um den ein Punkt aus $\mathcal{S}_j^{\mathcal{I}}$ durchschnittlich pro Dimension von einem anderen Punkt aus $\mathcal{S}_j^{\mathcal{I}}$ abweichen darf, um noch in das gleiche Cluster geordnet zu werden[12]. s_2 ist die Minimaldistanz zwischen zwei Clustern. Es soll nun gezeigt werden, daß der Clusteralgorithmus eine *eindeutige* Partitionierung findet. Dazu wird folgende Definition benötigt:

Definition 5.10 Ein *Weg* W zwischen zwei Punkten $z_0, z_n \in \mathcal{S}_j^{\mathcal{I}}$, $z_0 \neq z_n$, $n \geq 1$, ist ein Tupel

$$W \;:=\; (z_0, z_1, \ldots, z_n) \quad \text{mit} \quad z_0, \ldots, z_n \in \mathcal{S}_j^{\mathcal{I}}$$

und

$$d\,(z_i, z_{i+1}) < s_2 \quad \forall \; i \in \{0, \ldots, n{-}1\}\,.$$

Der Clusteralgorithmus findet eine Partitionierung, für die gilt, daß zwischen jeweils zwei Elementen *innerhalb* eines Clusters ein Weg existiert, während dies für zwei Elemente aus verschiedenen Clustern nicht gilt. Es wird folgender Satz bewiesen:

---

[12] Zur konkreten Wahl von q_2 siehe Gleichung 6.2 in Kapitel 6.

Satz 5.1 Die Menge $\mathcal{S}_j^{\mathcal{I}}$ läßt sich *eindeutig* in ein System $C_{1j}, \ldots, C_{tj} \subset \mathcal{C}_j$ von Clustern partitionieren, so daß $\forall\ s, \tilde{s} \in \mathcal{S}_j^{\mathcal{I}},\ s \neq \tilde{s}$, gilt

i) $s, \tilde{s} \in C_{kj}$ mit $k \in \{1, \ldots, t\} \quad \Longrightarrow$
es gibt einen Weg W zwischen s und $\tilde{s}$

ii) $s \in C_{kj},\ \tilde{s} \in C_{\tilde{k}j}$ mit $k, \tilde{k} \in \{1, \ldots, t\},\ k \neq \tilde{k} \quad \Longrightarrow$
es gibt keinen Weg W zwischen s und $\tilde{s}$.

Beweis: $C_{1j}, \ldots, C_{tj} \in \mathcal{C}_j$ sei eine Partitionierung der Menge $\mathcal{S}_j^{\mathcal{I}}$, die die Eigenschaften i) und ii) erfüllt. Sei $C'_{1j}, \ldots, C'_{t'j} \in \mathcal{C}'_j$ eine *andere* Partitionierung der Menge $\mathcal{S}_j^{\mathcal{I}}$, die die Eigenschaften i) und ii) ebenfalls erfüllt, also $\mathcal{C}_j \neq \mathcal{C}'_j$. Aus der Tatsache, daß $\mathcal{C}_j$ und $\mathcal{C}'_j$ Partitionierungen der Menge $\mathcal{S}_j^{\mathcal{I}}$ sind, folgt nach Definition 5.9

$$\bigcup \mathcal{C}_j = \mathcal{S}_j^{\mathcal{I}} \quad \text{und} \quad \bigcup \mathcal{C}'_j = \mathcal{S}_j^{\mathcal{I}}.$$

Da aber $\mathcal{C}_j \neq \mathcal{C}'_j$ gilt, gibt es ein $s \in \mathcal{S}_j^{\mathcal{I}}$, ein $k \in \{1, \ldots, t\}$ und ein $k' \in \{1, \ldots, t'\}$, so daß $s \in C_{kj} \in \mathcal{C}_j$ und $s \in C'_{k'j} \in \mathcal{C}'_j$, aber $C_{kj} \neq C'_{k'j}$.
$C_{kj} \neq C'_{k'j} \quad \Longrightarrow \quad$ (o.B.d.A.) $\exists\ c \in \mathcal{S}_j^{\mathcal{I}},\ c \neq s,$ mit $c \in C_{kj}$, aber $c \notin C'_{k'j}$.
Einerseits folgt nun aus $c \in C_{kj}$ und $s \in C_{kj}$ nach Eigenschaft i) der Partitionierung $\mathcal{C}_j$:

Es gibt einen Weg W zwischen c und s.

Im Widerspruch dazu folgt andererseits aus $c \notin C'_{k'j}$ und $s \in C'_{k'j}$ nach Eigenschaft ii) der Partitionierung $\mathcal{C}'_j$:

Es gibt keinen Weg W zwischen c und s.

□

Da es also nur eine Möglichkeit gibt, die Menge $\mathcal{S}_j^{\mathcal{I}}$ auf die oben beschriebene Weise zu partitionieren, ist die durch den Clusteralgorithmus gefundene Aufteilung bis auf die Numerierung der Cluster ebenfalls eindeutig. Insbesondere hängt das Ergebnis nicht von der Reihenfolge der Auswahl der Elemente in Schritt 2 und Schritt 3 ab.
Die Anzahl der entstehenden Cluster ist nicht vorbestimmt, sondern hängt lediglich von der Beschaffenheit der Daten ab. Es ist a priori *nicht* festgelegt, wieviele Cluster entstehen sollen, d.h. wieviele Merkmale als repräsentativ für die betrachtete Objektklasse gelernt werden sollen. Statt dessen funktioniert der Lernalgorithmus - mit relativ wenig einschränkenden a priori-Kriterien (siehe Abschnitt 3.1) - datengetrieben.
Als vermeindlicher Nachteil der Zuordnungsvorschrift könnte der sogenannte „Chaining-Effekt“ angeführt werden. Das bedeutet, daß die Einordnung zweier nahe beieinander liegender Einzelelemente in das gleiche Cluster zur Bildung von kettenförmigen Clustern führen könnte [Bor85]. Diese Gefahr besteht jedoch nur auf den ersten Blick. Das Chaining-Problem kann nicht auftreten, da das Antwortverhalten der Bananenfilter präzise genug ist, um tatsächlich deutlich ausgebildete, lokale Maxima zu erzeugen (siehe Abschnitt 4.5.2, insbesondere Abbildung 4.5), die sich in $\mathcal{P}_j$ zu nahezu „kugelförmigen“, dichten Punktwolken anordnen.
Aus dem gleichen Grund erschien eine Befolgung der üblichen Empfehlung, das Ergebnis einer hierarchischen Clusteranalyse durch ein nachfolgendes, nicht-hierarchisches Verfahren zu verbessern, nicht notwendig, da die Güte der Ergebnisse als hinreichend betrachtet

wird (siehe Kapitel 6, insbesondere die Abbildungen 6.4 und 6.8). Punkt 4 des Lernalgorithmus lautet daher:

4. Unterziehe die Menge $\mathcal{S}_j^{\mathcal{I}}$ aller signifikanten Merkmale des j-ten Bildausschnitts aller Trainingsbilder $\mathcal{I}$ dem soeben beschriebenen Clusteralgorithmus.

5.4.3 Gelernte Repräsentation einer Objektklasse

Da die Ausprägung eines wichtigen (zu lernenden) Merkmals einer Objektklasse in vielen Trainingsbeispielen sehr ähnlich ist, wird als ein Ergebnis der oben beschriebenen Clusteranalyse erwartet, daß all diejenigen signifikanten Merkmale aus $\mathcal{S}_j^{\mathcal{I}}$, die eine konkrete Ausprägung einer abstrakten, zu lernenden Eigenschaft der Objektklasse kodieren, in das gleiche Cluster fallen. Außerdem sollte dieses signifikante Merkmal in relativ vielen Trainingsbildern vorhanden sein, so daß das zugehörige Cluster relativ viele Elemente enthalten wird ("Lernen von Häufigkeiten", siehe Abschnitt 3.1).
Anders ist es mit signifikanten Merkmalen aus $\mathcal{S}_j^{\mathcal{I}}$, die von zufälligen Variationen in den Trainingsbildern herrühren. Sie sollten eine eher große Distanz zu allen anderen Elementen aus $\mathcal{S}_j^{\mathcal{I}}$ haben und in Cluster fallen, die eher wenige Elemente enthalten.
Eingedenk dieser Überlegungen ist nun die Frage nach der Interpretation der Ergebnisse der Clusteranalyse leicht zu beantworten: „eher große und dichte" (*wichtige*) Cluster repräsentieren typische Objekteigenschaften, „eher kleine und wenig dichte" (*unwichtige*) Cluster repräsentieren zufällige, ausgeprägte Merkmale auf *einzelnen* Trainingsbildern, die z.B. durch Variationen im Hintergrund, in der Beleuchtung usw. entstanden sind und keine spezifischen Eigenschaften der Objektklasse darstellen.
Also gilt es im vorletzten Schritt des Algorithmus, die wichtigen von den unwichtigen Clustern zu trennen.

Definition 5.11 Ein Cluster $C_{rj} \in \mathcal{C}_j$ mit $r \in \{1, \ldots, t\}$ ist ein *wichtiges Cluster*, wenn gilt

$$| \, C_{rj} \, | \quad \geq \quad q_3 \; \cdot \; \left| \, \mathcal{S}_j^{\mathcal{I}} \, \right|$$

mit $q_3 \in (0,1)$.

Es werden also lediglich diejenigen Cluster als wichtig extrahiert, deren Mächtigkeit eine bestimmte Schwelle überschreitet. Diese Schwelle berechnet sich als ein gewisser Prozentsatz q_3[13] der Gesamtmenge $\mathcal{S}_j^{\mathcal{I}}$.
Als Repräsentant eines wichtigen Clusters wird dessen Centroid gewählt. Bei seiner Berechnung wird jeder Punkt des Clusters mit seiner jeweiligen Absolutantwort gewichtet.

Definition 5.12 Der *gewichtete Centroid* c eines Clusters
$C_{rj} = \{ \, p_i \mid i = 1, \ldots, n\}$ wird definiert als

$$c \; := \; round \left(\frac{\sum_{i=1}^{n} \mathcal{A}\,(p_i) \cdot p_i}{\sum_{i=1}^{n} \mathcal{A}\,(p_i)} \right)$$

mit $\mathcal{A}\,(.)$ wie in Definition 5.3. *round* ist die kaufmännische Rundungsfunktion.

---

[13] Zur konkreten Wahl von q_3 siehe Gleichung 6.2 in Kapitel 6.

Nun kann die gelernte Repräsentation des betrachteten Ausschnitts definiert werden. Sie soll die Menge der Centroide der wichtigen Cluster sein, also:

Definition 5.13 Für die Objektklasse $\mathcal{O}$ wird die *gelernte Repräsentation des j-ten Bildausschnitts* definiert als

$$\mathcal{S}_j^{Rep(\mathcal{O})} := \left\{ (x,y,l,d,b,m) \in \mathcal{P}_j \;\middle|\; \begin{array}{l} \exists \text{ ein wichtiges Cluster } C_{rj} \in \mathcal{C}_j \text{ mit} \\ (x,y,l,d,b,m) \text{ ist Centroid von } C_{rj} \\ \text{und } r \in \{1,\ldots,t\} \end{array} \right\}$$

mit $\mathcal{P}_j$ wie in Abschnitt 5.3.1 und $\mathcal{C}_j$ wie in Definition 5.9.

Diese Menge ist ebenfalls in Abbildung 5.1 dargestellt. Damit kann der nächste Schritt des Lernalgorithmus folgendermaßen formuliert werden:

5. Berechne für den j-ten Bildausschnitt die Menge $\mathcal{S}_j^{Rep(\mathcal{O})}$.

Mit Definition 5.13 erhält man die Repräsentation einer Objektklasse:

Definition 5.14 Die *gelernte Repräsentation einer Objektklasse* $\mathcal{O}$ ist die Menge

$$\mathcal{S}^{Rep(\mathcal{O})} := \bigcup_j \mathcal{S}_j^{Rep(\mathcal{O})}.$$

(Siehe wiederum Abbildung 5.1.) So kann also der letzte Schritt des Lernalgorithmus konkretisiert werden:

6. Berechne die Menge $\mathcal{S}^{Rep(\mathcal{O})}$.

Kapitel 6

Experimentelle Ergebnisse

Der im letzten Kapitel beschriebene Algorithmus zum Lernen lokaler Objektmerkmale wurde in FLAVOR (Flexible Library for Active Vision and Object Recognition) implementiert. Dabei handelt es sich um ein am Institut für Neuroinformatik entwickeltes, objektorientiertes C++-Klassensystem zur Objekterkennung.
Die Visualisierung der gelernten Merkmale erfolgt, indem für jedes gefundene Element $p_0 = (x_0, y_0, l_0, d_0, b_0, m_0) \in \mathcal{P}$ eine schwarze Linie mit der Richtung α_0, der Krümmung c_0 und der Größe s_0 an die Stelle (x_0, y_0) eines ansonsten weißen Bildes gezeichnet wird, wenn (l_0, d_0, b_0, m_0) das Erzeugendentupel des Bananenwavelets $B_{(k_0,\alpha_0,c_0,s_0)}$ ist. (Für die Visualisierung der signifikanten Merkmale des j-ten Bildausschnitts des k-ten Bildes ist $p_0 \in \mathcal{S}_j^{I^k}$, für die Repräsentation der Objektklasse $\mathcal{O}$ ist $p_0 \in \mathcal{S}^{Rep(\mathcal{O})}$.) Die gezeichnete Linie entspricht den Koordinaten $(0, y_{s_0})$ des Wavelets $B_{(k_0,\alpha_0,c_0,s_0)}(x,y)$ für $y \in [-a, a]$ mit $a = 16 \cdot s_0 \cdot \max\left(\frac{\sigma_x}{k}, \frac{\sigma_y}{k}\right)$. Dabei entspricht der Punkt $(0,0)$ des Bananenwavelets dem Punkt (x_0, y_0) des Visualisierungsbildes. Die Wahl von a bedeutet, daß die Krümmung eines Wavelets lediglich für einen Ausschnitt um den Waveletmittelpunkt gezeichnet wird, und nicht über die gesamte Auflösung von 128 × 128 Pixeln (siehe Abbildung 6.1).
Der Algorithmus wurde für zwei Objektklassen - Getränkedose und Frontalgesicht (im folgenden kurz „Gesicht“ genannt)- auf einer Sparc-20-Workstation ausgeführt. Die Ergbnisse werden in den ersten beiden Abschnitten dieses Kapitels vorgestellt. Für diese Simulationen wurde eine Staude von 56 Bananenwavelets mit den Werten

$$\mathtt{L} = 1\,, \quad \mathtt{D} = 8\,, \quad \mathtt{B} = 7 \quad \text{und} \quad \mathtt{M} = 1 \tag{6.1}$$

gewählt (vergleiche mit Definition 4.4).
Nach der Festlegung der Parameter der benutzten Staude bleiben noch drei freie Parameter, die das Ergebnis des Algorithmus beeinflussen. Diese sind die Größen q_1 für die Bestimmung der signifikanten Merkmale eines Bildausschnitts (siehe Gleichung 5.1 in Abschnitt 5.3.2), q_2 für die Clusterbildung (siehe Gleichung 5.2 in Abschnitt 5.4.2) und q_3 für die Bestimmung der wichtigen Cluster (siehe Definition 5.11). Für die beiden ersten Simulationen wurde das (heuristisch ermittelte) Parametertupel

$$(\, q_1,\ q_2,\ q_3\,) \;=\; (\, 0.005,\ 0.00003,\ 0.2\,) \tag{6.2}$$

gewählt. Im dritten Abschnitt dieses Kapitels werden weitere Simulationen beschrieben, für die sowohl eine andere Staude als auch ein anderes Parametertupel verwendet wurden. Alle benutzten Bilder haben eine Auflösung von 128 × 128 Pixeln. Jeder Pixel kann eine von 256 Graustufen annehmen. Die Bildausschnitte I_j^k (siehe Definition 5.5) haben für beide Simulationen eine Größe von 17 × 17 Pixeln mit den Landmarken als Mittelpunkt.

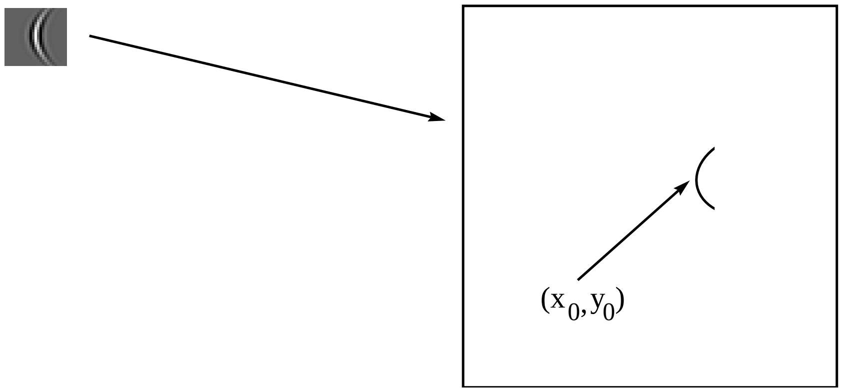


Abbildung 6.1: Visualisierung der gelernten Merkmale. Um $\mathcal{S}^{Rep(\mathcal{O})}$ zu visualisieren, wird für alle $p_0 = (x_0, y_0, l_0, d_0, b_0, m_0) \in \mathcal{S}^{Rep(\mathcal{O})}$ die „Mittellinie“ des Bananenwavelets $B_{(k_0,\alpha_0,c_0,s_0)}$ an die Stelle (x_0, y_0) gezeichnet. Links ist ein Ausschnitt des Realteils des Bananenwavelets $B_{(k_0,\alpha_0,c_0,s_0)}$ dargestellt (vergleiche mit Abbildung 4.3), rechts die Visualisierung des Merkmals $p_0 = (x_0, y_0, l_0, d_0, b_0, m_0) \in \mathcal{S}^{Rep(\mathcal{O})}$.

6.1 Getränkedosen

Um eine Repräsentation für die Objektklasse „Getränkedose“ zu lernen, wurden für die Menge aller Trainingsbilder $\mathcal{I}$ (siehe Definition 5.4) $K = 60$ Bilder von Dosen aufgenommen, die sich hinsichtlich der Beleuchtung und des Hintergrundes stark voneinander unterscheiden. Leichte Variationen liegen bezüglich des Aufsichtswinkels und der Größe der Dosen vor. Abbildung 6.2 zeigt eine Auswahl dieser Aufnahmen. In jedem Trainingsbild wurden die Landmarken so vorgegeben, wie es in Abbildung 3.2 gezeigt ist, es wurden also 11 Landmarken pro Dose gewählt.

Ein Beispiel für eine vom Algorithmus erzeugte Darstellung der signifikanten Merkmale (siehe Definition 5.6) aller Bildausschnitte des k-ten Bildes I^k wird in Abbildung 6.3 gezeigt. Das Endergebnis, d.h. die von Algorithmus gelernte Repräsentation der Objektklasse „Getränkedose“ $\mathcal{S}^{Rep(Getränkedose)}$, ist in Abbildung 6.4 dargestellt.

Für den Algorithmus mit der gewählten Staude von Bananenwavelets und das gewählte Parametertupel (siehe die Gleichungen 6.1 und 6.2) wurden für das Beispiel der Objektklasse „Getränkedose“ folgende CPU-Laufzeiten auf einer Sparc-20-Maschine gemessen:

Transformation eines Bildes	*ca.* 8.8 sec
Berechnung der Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ eines Bildes I^k	*ca.* 57 sec
Lernen von $\mathcal{S}^{Rep(Getränkedose)}$ bei gegebenen Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$	*ca.* 9.8 sec

6.2 Gesichter

Für das Lernen einer Repräsentation der Objektklasse „Gesicht“ wurden $K = 30$ Trainingsbilder verwendet, die in Abbildung 6.5 gezeigt sind. Die Landmarken wurden in jedem Trainingsbild so vergeben, wie es in Abbildung 6.6 beispielhaft dargestellt ist, d.h. pro Gesicht wurden 31 Landmarken vorgegeben. Auch für die Objektklasse „Gesicht“ sind Darstellungen der signifikanten Merkmale aller Bildausschnitte für jedes Trainingsbild $I^k \in \mathcal{I}$ erzeugt worden. Ein Beispiel für eine solche Darstellung befindet sich in Abbildung 6.7. Die vom Lernalgorithmus erzeugte Objektklassenrepräsentation für frontale Gesichter

Abbildung 6.2: TRAININGSBILDER FÜR DIE OBJEKTKLASSE „GETRÄNKEDOSE“. Hier sind 30 der 60 verwendeten Trainingsbilder dargestellt.

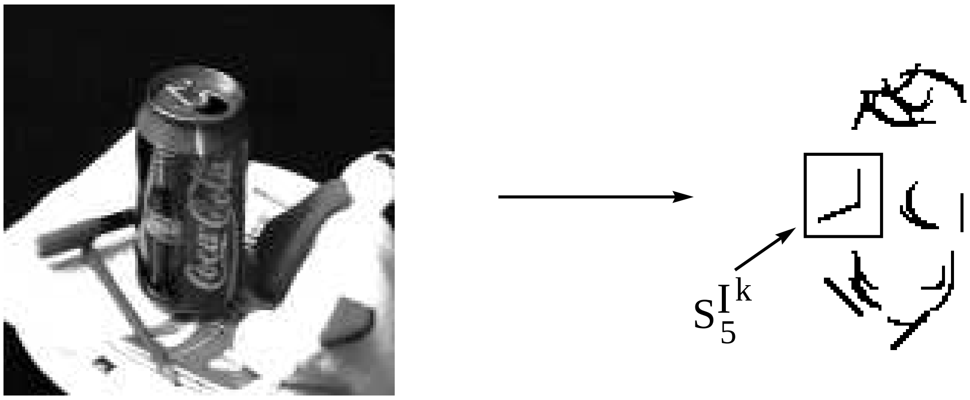


Abbildung 6.3: Signifikante Merkmale eines Trainingsbildes für die Objektklasse „Getränkedose“. Links ist das Bild $I^k \in \mathcal{I}$ abgebildet, rechts die vom Algorithmus erzeugte Darstellung der Menge $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ aller signifikanten Merkmale des Bildes I^k. Hervorgehoben ist die Menge $\mathcal{S}_5^{I^k}$ des 5. Bildausschnitts. Dort ist sowohl die senkrechte Begrenzungslinie der Dose, als auch die Stiftkante im Hintergrund als signifikant erkannt worden. Die Stiftkante ist in der Menge $\mathcal{S}_5^{I^k}$ noch vorhanden. Durch den Lernalgorithmus wird sie aber als unwichtig aussortiert werden, so daß sie in der Menge $\mathcal{S}^{Rep(Getränkedose)}$ nicht mehr präsent sein wird (siehe Abbildung 6.4). Vergleiche auch mit dem erwarteten Ergebnis in Abbildung 3.1.

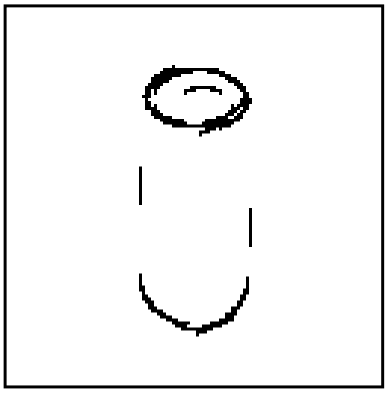


Abbildung 6.4: Gelernte Objektklassenrepräsentation für Getränkedosen. Dargestellt ist die Menge $\mathcal{S}^{Rep(Getränkedose)}$. Vergleiche mit der erwarteten Repräsentation, die in Abbildung 3.2 abgebildet ist.

Abbildung 6.5: Trainingsbilder für die Objektklasse „Gesicht“. Dargestellt ist die Menge $\mathcal{I}$ mit $K = 30$ (siehe Definition 5.4).

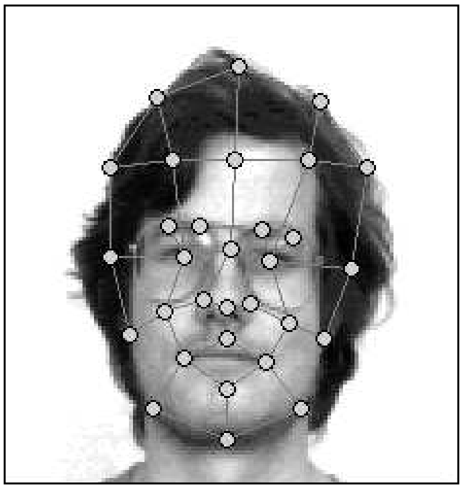


Abbildung 6.6: GESICHTERLANDMARKEN. In jedem der 30 verwendeten Trainingsbilder für die Objektklasse „Gesicht“ wurden die Landmarken so wie in diesem Beispiel gewählt.

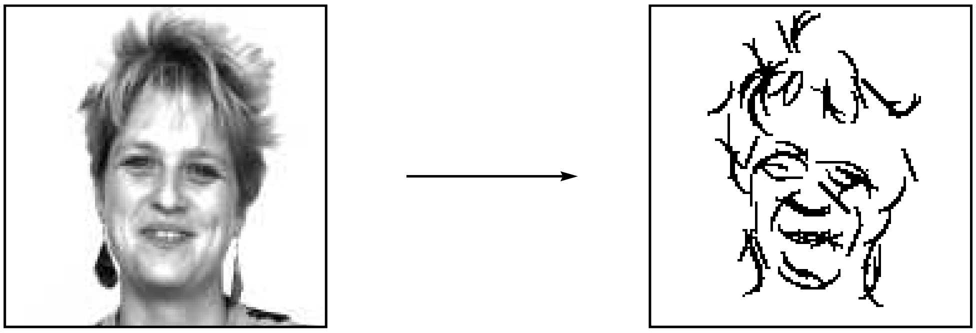


Abbildung 6.7: SIGNIFIKANTE MERKMALE EINES TRAININGSBILDES FÜR DIE OBJEKTKLASSE „GESICHT“. Links ist das Bild $I^{14} \in \mathcal{I}$ abgebildet, rechts die vom Algorithmus erzeugte Darstellung der Menge $\bigcup_{j=1}^{31} \mathcal{S}_j^{I^{14}}$ aller signifikanten Merkmale des Bildes I^{14} (siehe auch Abbildung 6.3).

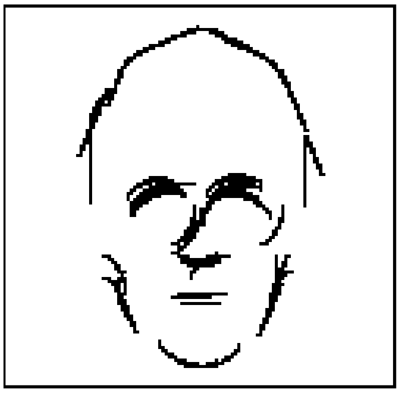


Abbildung 6.8: GELERNTE OBJEKTKLASSENREPRÄSENTATION FÜR GESICHTER. Dargestellt ist die Menge $\mathcal{S}^{Rep(Gesicht)}$.

$\mathcal{S}^{Rep(Gesicht)}$ ist in Abbildung 6.8 dargestellt.
Für die Generierung der Objektklassenrepräsentation „Gesicht“ wurden auf einer Sparc-20-Maschine mit den gewählten Parametern (siehe die Gleichungen 6.1 und 6.2) folgende Laufzeiten gemessen:

Transformation eines Bildes	*ca.* 8.8 sec
Berechnung der Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ eines Bildes I^k	*ca.* 160 sec
Lernen von $\mathcal{S}^{Rep(Gesicht)}$ bei gegebenen Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$	*ca.* 7.1 sec

Die längeren Laufzeiten für die Berechnung der Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ eines Bildes erklären sich aus der größeren Anzahl von Landmarken pro Gesicht (31) im Vergleich zu den Getränkedosen (11).

6.3 Weitere Simulationen

Zusätzlich wurden Simulationen mit veränderten Stauden durchgeführt.
Eine Staude mit

$$\mathtt{L} = 1\,,\quad \mathtt{D} = 8\,,\quad \mathtt{B} = 1 \quad \text{und} \quad \mathtt{M} = 1,$$

d.h. eine Staude, die lediglich Gaborwavelets enthält ($\mathtt{B} = 1$), wurde mit dem gleichen Parametertupel des Algorithmus (6.2) auf die Trainingsbilder für die Objektklasse „Gesicht“ angewendet. Das Ergebnis ist in Abbildung 6.9 abgebildet. Im Vergleich mit der zuvor gelernten Repräsentation (Abbildung 6.8) wird mit dieser Staude ein qualitativ schlechteres Ergebnis erzielt, was das Konzept des Suchens nach *gekrümmten* Kanten bestätigt (siehe Abschnitt 3.1). Für diese Simulation wurden folgende Zeiten gemessen:

Transformation eines Bildes	*ca.* 1.2 sec
Berechnung der Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ eines Bildes I^k	*ca.* 3.3 sec
Lernen von $\mathcal{S}^{Rep(Gesicht)}$ bei gegebenen Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$	*ca.* 1.7 sec

Eine weitere Simulation wurde sowohl mit einer anderen Staude, als auch mit einem um einen Parameter veränderten Parametertupel für die Objektklasse „Getränkedose“ durchgeführt. Als Staudenwerte wurden gewählt:

$$\mathtt{L} = 2\,,\quad \mathtt{D} = 8\,,\quad \mathtt{B} = 7 \quad \text{und} \quad \mathtt{M} = 2,$$

Das Parametertupel war für diese Simulation

$$(\, q_1,\ q_2,\ q_3 \,) \;=\; (\, 0.005,\ 0.000006,\ 0.2 \,)\,,$$

d.h. q_2 wurde verändert. Die Hauptsimulationen, die in den Abschnitten 6.1 und 6.2 beschrieben wurden, sind lediglich in vier Dimensionen des Parameterraums der Bananentransformation $\mathcal{P}$ (siehe Definition 5.1) durchgeführt wurden, da $\mathtt{L} = 1$ und $\mathtt{M} = 1$ gewählt worden sind. Hier sind nun alle sechs Dimensionen des Parameterraums ausgenutzt worden. Das Ergebnis ist in Abbildung 6.10 wiedergegeben. Für diese Simulation wurden folgende Zeiten gemessen:

Transformation eines Bildes	*ca.* 32 sec
Berechnung der Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ eines Bildes I^k	*ca.* 750 sec
Lernen von $\mathcal{S}^{Rep(Getränkedose)}$ bei gegebenen Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$	*ca.* 8.6 sec

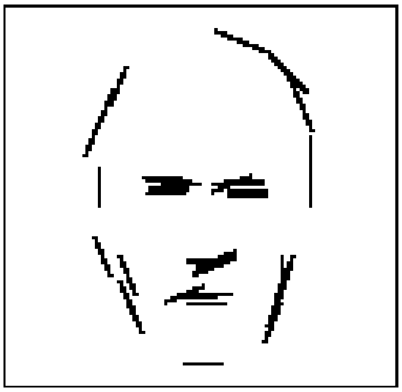


Abbildung 6.9: MIT GABORWAVELETS GELERNTE OBJEKTKLASSENREPRÄSENTATION FÜR GESICHTER. Dargestellt ist die Menge $\mathcal{S}^{Rep(Gesicht)}$ für eine Staude mit $\mathtt{L} = 1, \mathtt{D} = 8, \mathtt{B} = 1, \mathtt{M} = 1$. Vergleiche mit der gelernten Repräsentation in Abbildung 6.8.

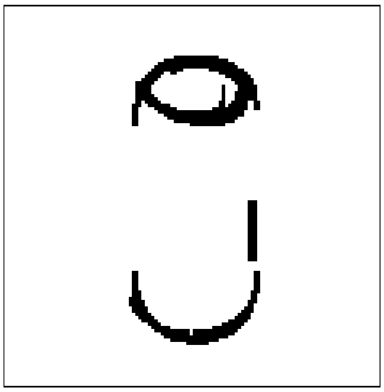


Abbildung 6.10: MIT VOLLEM PARAMETERRAUM GELERNTE OBJEKTKLASSENREPRÄSENTATION FÜR GETRÄNKEDOSEN. Dargestellt ist die Menge $\mathcal{S}^{Rep(Getränkedose)}$ für eine Staude mit $\mathtt{L} = 2, \mathtt{D} = 8, \mathtt{B} = 7, \mathtt{M} = 2$. Obwohl der Lernalgorithmus mit einer Staude ausgeführt wurde, die zwei verschiedene Frequenzen und zwei verschiede Größen aufweist, sind unter den gelernten signifikanten Merkmalen $s \in \mathcal{S}^{Rep(Getränkedose)}$ lediglich solche der niedrigeren Frequenz und der kleineren Größe vorhanden. Durch eine andere Definition des Vorfaktors γ der Bananenformel könnte dieser Nachteil behoben werden. Vergleiche mit der gelernten Repräsentation in Abbildung 6.4.

Kapitel 7

Schlußfolgerungen

In dieser Arbeit wurde ein Algorithmus vorgestellt, der anhand von Beispielbildern verschiedener Objekte derselben Klasse eine zweidimensionale, ansichtsabhängige Repräsentation der Objektklasse lernt. Für diese Repräsentation wurden in den ersten Kapiteln einige Forderungen aufgestellt, die als erfüllt betrachtet werden können:

1. Die Repräsentation beinhaltet nahezu ausschließlich *repräsentative* lokale Merkmale der Objektklasse. Hintergrund- und Beleuchtungsinformation ist im Gegensatz zur Repräsentation durch Bunchgraphen *nicht* in der Repräsentation enthalten.
2. Die Repräsentation ist weitgehend redundanzfrei.
3. Die Repräsentation wird im Vergleich zu Bunchgraphen durch eine wesentlich kleinere Datenmenge kodiert.
4. Die Repräsentation wird im Gegensatz zu Bunchgraphen durch einen *Lernprozeß* generiert, der wichtige von unwichtigen lokalen Objektmerkmalen trennen kann.

Das in Abschnitt 3.1 vorgestellte Lernkonzept hat sich im Hinblick auf die in den Computersimulationen erzielten Ergebnisse ebenfalls als sinnvoll erwiesen. Insbesondere die Analyse von natürlichen Bildern hinsichtlich gekrümmter Kanten kann als nützlich für die Erzeugung einer Objektklassenrepräsentation gewertet werden.

Wenn die Modellierung noch wenig verstandener, höherer Hirnfunktionen wie Wahrnehmung oder zielgerichtetes Handeln als ein fernes Ziel der Neuroinformatik angenommen wird, so ist es unerläßlich, zuvor das Lernen von Begriffen zu simulieren. Zielgerichtetes Handeln ist kaum denkbar ohne die Annahme einer gedanklichen Vorwegnahme von Handlungszielen. Der in dieser Arbeit vorgestellte Lernalgorithmus kann als ein - wenn auch sehr kleiner - Schritt in die Richtung einer Simulation von Begriffsbildung betrachtet werden.

Kapitel 8

Ausblick

Im Anschluß an diese Arbeit können einige Verbesserungen des vorgestellten Algorithmus erfolgen. Zum Beispiel könnte durch eine andere Definition des Vorfaktors γ der Bananenwavelets (siehe Definition 4.1) eine Verbesserung der Ergebnisse beim Lernen mit vollem (sechsdimensionalem) Parameterraum erzielt werden (siehe Abbildung 6.10). Auch eine Beschleunigung der Berechnung der Mengen $\bigcup_{j=1}^{n} \mathcal{S}_j^{I^k}$ eines Bildes I^k ist denkbar.
Im folgenden sollen noch einige weitreichendere Aspekte diskutiert werden.

Anpassen mit Bananenwavelets
Ein Hauptzweck des Lernens einer zweidimensionalen Objektklassenrepräsentation ist das Anpassen eines Objektes der entsprechenden Klasse. In Abschnitt 3.2.2 wurde die Hoffnung ausgesprochen, daß sich das Anpassen aufgrund der geringeren Datenmenge der gelernten Repräsentation im Vergleich zu Bunchgraphen beschleunigen wird. Die Ergebnisse des Anpassens mit Bananenwavelets werden in [KP] beschrieben.

Approximation der Bananenwavelets durch Gaborwavelets
Um das Lernen einer Repräsentation und das Anpassen von Objekten zu beschleunigen, kann die relativ große Filteranzahl, die durch eine Staude gegeben ist, verringert werden, indem die verschiedenen Bananenwavelets durch relativ wenige Gaborwavelets approximiert werden. Diese Vorgehensweisen wird ebenfalls in [KP] erläutert.

Korrespondenzproblem
Für diese Arbeit wurde das Korrespondenzproblem als gelöst betrachtet, indem für jedes Trainingsbild die korrespondierenden Landmarken von Hand vorgegeben wurden. Eine Möglichkeit, diese Einschränkung zu umgehen, besteht in der Anwendung visuomotorischer Algorithmen, wie sie in [KP] beschrieben werden.
Das Verfolgen (*Tracking*) eines Objektes in einer Bewegungssequenz ist eine andere Möglichkeit, das Korrespondenzproblem zu umgehen. Dabei muß jedoch noch die Anfangsposition des Objektes manuell bestimmt werden. Eine Beschreibung dieses Verfahrens ist in [Mau96] nachzulesen.
Eine weitere Möglichkeit, das Korrespondenzproblem für Objekte, die aus derselben Ansicht aufgenommen wurden, die sich jedoch an unterschiedlichen Positionen in den Trainingsbilder befinden, zu lösen, kann darin bestehen, einen ähnlichen Algorithmus, wie den in dieser Arbeit vorgestellten, nicht auf das Lernen *lokaler Merkmale*, sondern auf das Lernen von *metrischen Relationen zwischen lokalen Merkmalen* anzuwenden. Dabei kann sich z.B. ergeben, daß sich in vielen Trainingsbildern bestimmte lokale Merkmale besonders

häufig in bestimmten Konstellationen zueinander befinden. Diese häufigen Anordnungen von lokalen Merkmalen würden als Objektrepräsentation gelernt werden. Ein solcher Algorithmus hätte gegenüber einem Tracking-Algorithmus den Vorteil, daß auf die manuelle Vorgabe der Startposition des sich bewegenden Objektes verzichtet werden kann. Außerdem wäre es möglich, auch aus Bildern, die in keinem zeitlichen Zusammenhang stehen, Gemeinsamkeiten zu lernen. Ein Problem stellt allerdings die große Datenmenge dar, da zwischen allen gefundenen signifikanten Merkmalen eines Trainingsbildes Relationen gespeichert werden müßten.

Bildung von Begriffshierarchien

Die Bedeutung der Bildung von Begriffshierarchien für die menschliche Kognition wurde in Abschnitt 3.3.2 erläutert. Mit Hilfe des vorgestellten Algorithmus könnte es gelingen, die Bildung von Begriffshierarchien in Computersimulationen nachzuvollziehen. Dazu könnte man Teilmengenbeziehungen zwischen gelernten Merkmalen von Objekten ausnutzen. Beispielsweise könnte sich dabei ergeben, daß die Repräsentation der Objektklasse „Gesicht" $\mathcal{S}^{Rep(Gesicht)}$ eine Teilmenge der Repräsentation der Objektklasse „Gesicht mit zusätzlichem Merkmal Brille" $\mathcal{S}^{Rep(Gesicht_{Brille})}$ ist, kurz

$$\mathcal{S}^{Rep(Gesicht)} \subset \mathcal{S}^{Rep(Gesicht_{Brille})}.$$

Dabei könnten beide Repräsentationen, sowohl $\mathcal{S}^{Rep(Gesicht)}$ als auch $\mathcal{S}^{Rep(Gesicht_{Brille})}$, aus dem gleichen Satz von Trainingsbilder gelernt werden, wenn sich darin ein gewisser Prozentsatz von Gesichtern mit Brillen befindet. Wenn $\mathcal{S}^{Rep(Gesicht)}$ z.B. durch den Lernalgorithmus mit dem Parametertupel (q_1^1, q_2^1, q_3^1) (siehe Kapitel 6) erzeugt wird, dann kann $\mathcal{S}^{Rep(Gesicht_{Brille})}$ erzeugt werden mit einem Parametertupel (q_1^2, q_2^2, q_3^2), wobei $q_3^2 < q_3^1$ zu wählen ist. (q_3 bestimmt die Größe eines wichtigen Clusters, siehe Definition 5.11.)
Aus einer Beziehung $\mathcal{S}^{Rep(\mathcal{O}_1)} \subset \mathcal{S}^{Rep(\mathcal{O}_2)}$ kann man also folgern, daß die Objektklasse $\mathcal{O}_2$ ein Spezialfall der Objektklasse $\mathcal{O}_1$ ist, da mit sinkender Merkmalsanzahl der Abstraktionsgrad steigt (siehe die Ausführungen zu Begriffen und Begriffshierarchien in Abschnitt 3.3.2).
Aus $\mathcal{S}^{Rep(\mathcal{O}_1)} \cap \mathcal{S}^{Rep(\mathcal{O}_2)} \neq \emptyset$ kann eine verwandtschaftliche Beziehung zwischen den Objektklassen $\mathcal{O}_1$ und $\mathcal{O}_2$ gefolgert werden. Es wird z.B. gelten

$$\mathcal{S}^{Rep(Gesicht_{Brille})} \cap \mathcal{S}^{Rep(Gesicht_{Bart})} = \mathcal{S}^{Rep(Gesicht)},$$

d.h. die Objektklasse „Gesicht" ist ein Oberbegriff für die Klassen „Gesicht mit zusätzlichem Merkmal Brille" und „Gesicht mit zusätzlichem Merkmal Bart". Die Behandlung von Begriffsverwandtschaften in der menschlichen Informationsverarbeitung wurde z.B. in [Kli78] untersucht.

Dreidimensionale Repräsentation von Objektklassen

In dieser Arbeit wurde ein Algorithmus zum Lernen einer lediglich zweidimensionalen Objektklassenrepräsentation vorgestellt, die zudem ansichtsabhängig ist. Es wäre ein großer Schritt für die künstliche Objekterkennung, gäbe es einen Algorithmus, der aus Beispielbildern von einfachen, geometrischen Objekten eine Objektklassenrepräsentation lernen könnte, die dreidimensional und nicht mehr ansichtsabhängig ist. Bei der Entwicklung eines solchen Algorithmus könnten die Eigenschaften der Bananenwavelets und ihres Parameterraumes genutzt werden. In Abbildung 8.1 sind als Beispiel für einfache, geometrische

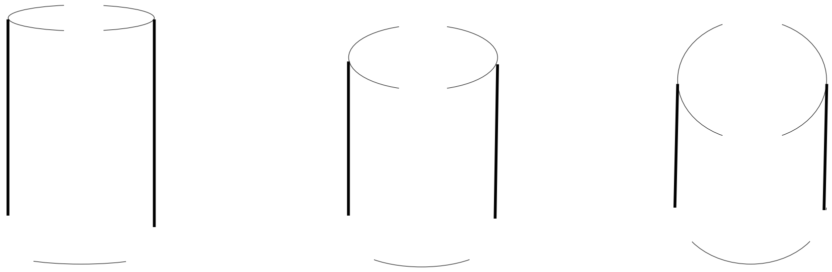


Abbildung 8.1: EINFACHES GEOMETRISCHES OBJEKT IN DREI VERSCHIEDENEN ANSICHTEN. Die Abbildung wird im Text erläutert.

Objekte gelernte Repräsentationen für drei verschiedene Ansichten von Getränkedosen in idealisierter Weise dargestellt: links eine eher seitliche Ansicht, rechts eher eine Ansicht von oben. Der in dieser Arbeit vorgestellte Lernalgorithmus würde für die drei verschiedenen Ansichten Repräsentationen lernen, deren Merkmale sich jeweils nur in einer der sechs Dimension des Parameterraumes der Bananentransformation $\mathcal{P}$ unterscheiden. Beispielsweise würden sich die Merkmale, die den oberen Dosenrand beschreiben, lediglich in der Dimension der Krümmung unterscheiden (links: starke Krümmung, rechts: schwache Krümmung), während die Parameter, die die Ausprägungen auf den anderen Dimensionen beschreiben (Ort, Frequenz, Orientierung und Größe), annähernd gleich wären. Die senkrechten Begrenzungslinien der drei Repräsentationen würden sich hauptsächlich in ihren Größen unterscheiden, während sich die anderen Parameter ähneln würden. Für den unteren Dosenrand läge der Hauptunterschied wieder in der Krümmungseigenschaft. Selbst bei „komplizierteren“ Objekten, wie z.B. bei Gesichtern, werden sich bei *leichten*, räumlichen Rotationen die signifikanten Merkmale der Objekte nur in wenigen Parametern des Raumes $\mathcal{P}$ verändern.

Diese Ähnlichkeit von Repräsentationen gleicher Objekte in verschiedenen Ansichten könnte für die Generierung einer dreidimensionalen und weitgehend ansichtsunabhängigen Objektklassenrepräsentation genutzt werden.

Abbildungsverzeichnis

2.1 Bildgraph 4
2.2 Gaborwavelet 4
2.3 Jet und Bildgraph 5
2.4 Bunchgraph 6
2.5 Korrespondenzproblem 9

3.1 Landmarkenumgebung 13
3.2 Erwartete Repräsentation 14
3.3 Sehbahn 17
3.4 Funktionsweise einer endinhibierten Zelle 18
3.5 Wirksamer Stimulus einer endinhibierten Zelle 19
3.6 Visueller Assoziationskortex 20
3.7 Mäßig komplexe Merkmale 20
3.8 Beispiel eines Schemas 21

4.1 Bananenwavelet 24
4.2 Real- und Imaginärteil eines Bananenwavelets 27
4.3 Staude von Bananenwavelets 28
4.4 Antwortverhalten eines Bananenwavelets 32
4.5 Antworten einer Staude 33
4.6 Bananenwavelets im Frequenzraum 35
4.7 Bananentransformation im Fourierraum 36

5.1 Lernalgorithmus 38
5.2 Möbiusband des Parameterraums 41
5.3 Metrik im Parameterraum 41

6.1 Visualisierung der gelernten Merkmale 54
6.2 Trainingsbilder für die Objektklasse „Getränkedose“ 55
6.3 Signifikante Merkmale eines Trainingsbildes für Getränkedosen 56
6.4 Gelernte Objektklassenrepräsentation für Getränkedosen 56
6.5 Trainingsbilder für die Objektklasse „Gesichter“ 57
6.6 Gesichterlandmarken 58
6.7 Signifikante Merkmale eines Trainingsbildes für Gesichter 58
6.8 Gelernte Objektklassenrepräsentation für Gesichter 58
6.9 Mit Gaborwavelets gelernte Repräsentation 61
6.10 Mit vollem Parameterraum gelernte Repräsentation 61

8.1 Einfaches geometrisches Objekt 65

Literaturverzeichnis

[And88] J. R. Anderson. *Kognitive Psychologie, Eine Einführung.* Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, 1988.

[Bor85] J. Bortz. *Statistik für Sozialwissenschaftler.* Springer, Berlin, Heidelberg, New York, Tokyo, 1985.

[Car91] N. R. Carlson. *Physiology of Behavior.* Allyn and Bacon, Needham Heights, MA, 1991.

[Dau88] J. D. Daugman. *Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression.* IEEE Trans. Acoustics, Speech, and Signal Processing, 36:1169-1179, 1988.

[dVdV88] R. L. d. Valois, K. K. d. Valois. *Spatial Vision.* Oxford Press, 1988.

[DZC87] A. Dobbins, S. Zucker, M. S. Cynader. *Endstopped neurons in the visual cortex as a substrate for calculating curvature.* Nature, 329:438-441, 1987.

[Fie94] D. Field. *What is the Goel of Sensory Coding?* Neural Computation, 6(4):561-601, 1994.

[Hub89] D. H. Hubel. *Auge und Gehirn, Neurobiologie des Sehens.* Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, 1989.

[HW62] D. H. Hubel, T. N. Wiesel. *Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex.* Journal of Physiology, London, 160:106-154, 1962.

[JP87] J. P. Jones, L. A. Palmer. *An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex.* Journal of Neurophysiology, 58(6):1233-1258, 1987.

[Kan1781] I. Kant. *Kritik der reinen Vernunft.* Atlas-Verlag, Köln, 1781.

[Kli78] F. Klix. *On the representation of semantic information in human long-term memory.* Zeitschrift für Psychologie, 186:26-38, 1978.

[Kos80] S. M. Kosslyn. *Image and Mind.* Harvard University Press, Cambridge, MA, 1980.

[KP] N. Krüger, G. Peters. *Object Recognition with a Sparse and Autonomously Learned Representation Based on Banana Wavelets.* (in Arbeit)

[KPM96] N. Krüger, M. Pötzsch, C. v. d. Malsburg. *Determination of Face Position and Pose with a Learned Representation based on Labeled Graphs.* IR-INI 96-03. Institut für Neuroinformatik, Ruhr-Universität Bochum, 1996.

[Krü95] N. Krüger. *Learning Weights in Discrimination Functions using a priori Constraints.* In: *Mustererkennung 1995, Verstehen akustischer und visueller Informationen, 17. DAGM-Symposium*, 110-117. Springer-Verlag, Berlin, Heidelberg, 1995.

[LVB^+92] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. v. d. Malsburg, R. P. Würtz, W. Konen. *Distortion Invariant Object Recognition in the Dynamik Link Architecture.* IEEE Transactions on Computers, 42(3):300-311, 1992.

[Mau96] T. Maurer, C. v. d. Malsburg. *Tracking and Learning Graphs on Image Sequences of Faces.* In: *Proceedings of the ICANN 1996*, 323-328. Springer Verlag, Berlin, Heidelberg, New York, 1996.

[Pal80] G. Palm. *On associative memory.* Biological Cybernetics, 36:19-31, 1980.

[Pöt94] M. Pötzsch. *Die Behandlung der Wavelet-Transformation von Bildern in der Nähe von Objektkanten.* IR-INI 94-04. Institut für Neuroinformatik, Ruhr-Universität Bochum, 1994.

[PTV^+94] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C. The Art of Scientific Computing.* Cambridge University Press, Cambridge, New York, Melbourne, second edition, 1994.

[RMG76] E. Rosch, C. D. Mervis, W. D. Gray. *Basic objects in natural categories.* Cognitive Psychology, 8:382-439, 1976.

[Ros78] E. Rosch. *Cognition and categorization.* Erlbaum, Hillsdale, 1978.

[Sha93] C. J. Shatz. *The Developing Brain.* In: *Mind and brain: readings from Scientific American magazine*, 15-26. W.H.Freeman and Company, New York, 1993.

[SL77] D. Steinhausen, K. Langer. *Clusteranalyse.* Walter de Gruyter & Co., Berlin, New York, 1977.

[SS89] W. Schönpflug, U. Schönpflug. *Psychologie, Allgemeine Psychologie und ihre Verzweigungen in die Entwicklungs-, Persönlichkeits- und Sozialpsychologie.* Psychologie-Verlags-Union, München, 1989.

[SS96] L. Shams and J. Spoelstra. *Learning Gabor-based Features for Face Detection.* Eingereicht zu: World Congress on Neural Networks (WCNN), San Diego, 1996.

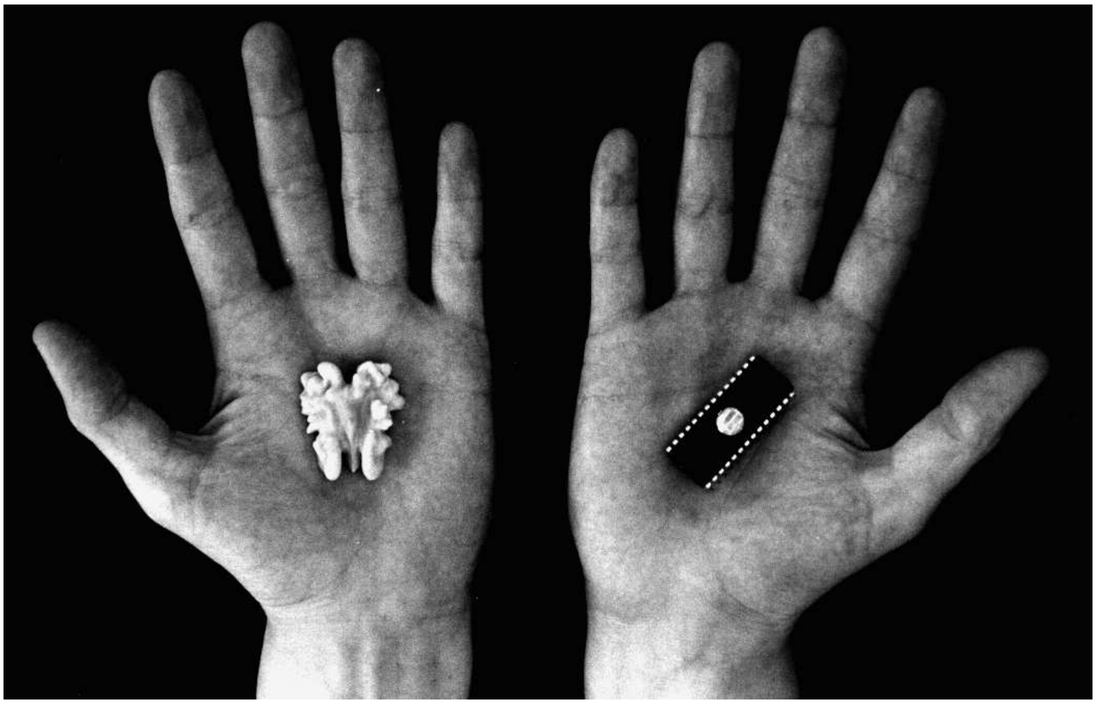
[Tan93] K. Tanaka. *Neuronal Mechanisms of Objekt Recognition.* Science, 262:685-688, 1993.

[WFK^+96] L. Wiskott, J.-M. Fellous, N. Krüger, C. v. d. Malsburg. *Face Recognition by Elastic Buch Graph Matching.* IR-INI 96-08. Institut für Neuroinformatik, Ruhr-Universität Bochum, 1996.

[Zek93] S. Zeki. *The Visual Image in Mind and Brain.* In: *Mind and brain: readings from Scientific American magazine*, 27-39. W.H.Freeman and Company, New York, 1993.

Die vorliegende Arbeit habe ich selbständig verfaßt. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Gabriele Peters



Neuroinformatik