

Studienrichtung Vermessungswesen  
Technische Universität Wien

G E O W I S S E N S C H A F T L I C H E  
M I T T E I L U N G E N

Heft 40

**Methodische Erweiterung der Landnutzungsklassifikation und  
Implementierung auf einem Transputernetzwerk**

von  
Klaus Steinnocher

Veröffentlichung des Institutes für Photogrammetrie und Fernerkundung

Geowiss. Mitt.  
40, 1994

Wien, im Juli 1994

Studienrichtung Vermessungswesen  
Technische Universität Wien

G E O W I S S E N S C H A F T L I C H E  
M I T T E I L U N G E N

---

Heft 40

**Methodische Erweiterung der Landnutzungsklassifikation und  
Implementierung auf einem Transputernetzwerk**

von  
Klaus Steinnocher

Veröffentlichung des Institutes für Photogrammetrie und Fernerkundung

Geowiss. Mitt.  
40, 1994

Wien, im Juli 1994

Herausgeber und Verleger: O. Prof. Dr.-Ing. Karl Kraus  
Vorstand des Institutes für Photogrammetrie und Fernerkundung  
der Technischen Universität Wien  
A-1040 Wien, Gußhausstr. 27-29

Diese Arbeit wurde an der Technisch-naturwissenschaftlichen Fakultät der Technischen Universität Wien zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften eingereicht.

Die Kosten für den Druck wurden aus der ordentlichen Dotation und aus eigenen Einnahmen des Institutes für Photogrammetrie und Fernerkundung der Technischen Universität Wien getragen.

Druck: Eurographik Ges.m.b.H.

1234 Wien

Auflage: 400 Stück

## **KURZFASSUNG**

Die vorliegende Arbeit bemüht sich um eine umfassende Betrachtung der Erfassung von Landnutzung aus Fernerkundungsdaten. Neben den klassischen Verfahren der multispektralen Klassifikation wird im Detail auf die Texturanalyse eingegangen und deren Methodik anhand von praktischen Beispielen demonstriert. Verfahren der Postklassifikation, die Oberflächenbedeckungsarten zu Landnutzungsklassen aggregieren bilden einen weiteren Schwerpunkt. In diesem Zusammenhang wird ein Algorithmus vorgestellt, der eine Postklassifikation über ein hierarchisches Regelsystem durchführt.

Der zweite Teil beschäftigt sich mit der Implementierung dieser Algorithmen auf einem Transputernetzwerk. Im Mittelpunkt steht die Entwicklung von Programmschalen, die eine problemlose Implementierung von lokalen Bildverarbeitungsalgorithmen ermöglichen. Darüberhinaus wird auf Aspekte der Visualisierung und der Datenkompression eingegangen.

Im dritten Teil wird die Anwendung der entwickelten Methoden anhand einer Landnutzungs-klassifikation für Funkausbreitungsmodelle beschrieben. Ein integrativer Klassifikations-ansatz, der multispektrale, texturale und Postklassifikationsmethoden beinhaltet, wird auf einen hybriden Datensatz angewendet. Als Alternativverfahren erfolgt eine visuelle Interpretation von Orthophotos. Vergleiche der beiden Methoden und deren Beurteilung mittels Referenzdaten bilden den Abschluß der Arbeit.

## **ABSTRACT**

The aim of the presented work is a comprehensive view of land use classification methods applying remote sensing data. Besides the traditional approach of multispectral classification the theory of texture analysis is described and demonstrated by means of concrete examples. Next a variety of postclassification algorithms, that aggregate land cover types to land use classes, is discussed. In this context a postclassification method is introduced, that is based on a hierarchical rule system.

The second part demonstrates the implementation of these algorithms on a Transputer network. It focusses on the design of programm shells, that allow a simple implementation of local image processing algorithms. Besides aspects of visualisation and data compression are discussed.

The application of these methods by means of a land use classification for the design of mobile radio systems makes up the third part. An integrative classification approach including multi-spectral, textural and postclassification methods, is applied to a hybrid data set. Furthermore visual interpretations of orthophotos are performed as an alternative method. Comparison of both methods and evaluation based on reference data conclude the study.

## VORWORT

Die vorliegende Arbeit entstand während meiner Tätigkeit als Assistent am Institut für Photogrammetrie und Fernerkundung der Technischen Universität Wien. Die Kosten meiner Anstellung als Vertragsassistent gegen Refundierung wurden zum Teil über den Fond der Nationalbank, und über eine halbe Karenzstelle finanziert, der Großteil der Aufwendungen wurde jedoch vom Institut selbst getragen. Trotzdem war es mir möglich, den überwiegenden Teil meiner Zeit für Forschung und Entwicklung zu verwenden, da ich weder durch Administration oder Lehrtätigkeit noch durch Projektarbeiten sonderlich belastet wurde.

Mein besonderer Dank gilt daher meinem Institutsvorstand und Dissertationsvater, Prof. Kraus, der mir nicht nur die Möglichkeit zur unabhängigen Forschung gegeben hat, sondern auch immer offen war für einen Gedankenaustausch und neue Ideen. In diesem Zusammenhang möchte ich auch Prof. Purgathofer für die Übernahme des Koreferates danken. Bei den Mitarbeitern vom Institut bedanke ich mich für die angenehme Atmosphäre und die Unterstützung, die sie mir in Form ihres Fachwissens und ihrer kollegialen Verbundenheit zukommen ließen. Ein spezielles Dankeschön richtet sich dabei an meine Kollegin Jana Babnik, mit der ich gemeinsam die Höhen und Tiefen der parallelen Programmierung durchwandert habe. Die Zusammenarbeit mit ihr war eine nicht immer leichte Reise durch interdisziplinäre Welten, von der ich jedoch enorm profitiert habe.

Den entscheidenden Anstoß für den praktischen Einsatz meiner Arbeit verdanke ich Hermann Bühler, Assistent am Institut für Nachrichten- und Hochfrequenztechnik der Technischen Universität Wien, der mir die Zusammenhänge zwischen Landnutzung und Funkwellenausbreitung verdeutlicht hat. Ein weiterer Dank geht an die Projektgruppe des Hochschullehrgangs Geoinformationswesen, namentlich Petra Staufer, Michael Franzen und Gerhard Muggenhuber, ohne deren großartigen Einsatz im Februar 1993 die Durchführung der Fallstudie wohl kaum möglich gewesen wäre.

Ein letztes, deshalb aber nicht minderes Dankeschön gilt meinen Eltern, die mir die geistigen, ethischen und nicht zu unterschätzen, die finanziellen Grundlagen vermittelt haben, und meiner Freundin, die mich von der fachlichen bis zur emotionalen Seite unermüdlich unterstützt hat.

# Inhaltsverzeichnis

<b>0. Einleitung</b>	1
<b>1. Mustererkennung in der Fernerkundung</b>	3
1.1 Einleitung	3
1.2 Mustererkennung	3
1.3 Spektrale Klassifikation	5
1.3.1 Digitale Bilder	5
1.3.2 Multispektrale Klassifikation	5
1.3.3 Multisensorale und multitemporale Klassifikation	7
1.4 Textur	9
1.4.1 Beschreibung von Textur im digitalen Bild	9
1.4.2 Grauwert-Abhängigkeits-Matrizen	10
1.4.3 Texturmerkmale	12
1.4.4 Skalierung von Texturmerkmalsbildern	15
1.5 Texturale Klassifikation	17
1.5.1 Segmentierung von Texturmerkmalsbildern	17
1.5.2 Texturmerkmalsbilder als künstliche Kanäle	20
1.6 Postklassifikation	21
1.6.1 <i>Low level</i> Postklassifikationsalgorithmen	22
1.6.2 Raumbezogene kontexturale Operatoren	23
1.6.3 Externe Daten	29
1.6.4 Wissensbasierte Expertensysteme	29
1.7 Integration von Fernerkundung und GIS	33
<b>2. Bildverarbeitung mit Transputern</b>	34
2.1 Einleitung	34
2.2 Hardware	34
2.2.1 Transputer	34
2.2.2 Impuls System 2400	36
2.3 Software	40
2.3.1 Voraussetzungen	40
2.3.2 Software Design	47
2.3.3 <i>Screen-to-Screen</i> Verarbeitung	49
2.3.4 <i>Disk-to-Disk</i> Verarbeitung	60
2.3.5 Einsatz in der Bildverarbeitung	65
<b>3. Fallstudie Agglomerationsraum Wien</b>	66
3.1 Einleitung	66
3.2 Aufgabenstellung	66
3.2.1 Zellularer Mobilfunk	66
3.2.2 Landnutzungsklassen	67

3.2.3 Untersuchungsgebiet und Datenquellen	68
3.3 Methodik	70
3.3.1 Multispektrale Klassifikation	71
3.3.2 Texturanalyse	74
3.3.3 Rektifizierung	74
3.3.4 Verschneidung der Klassifikationsergebnisse	76
3.3.5 Postklassifikation	78
3.3.6 Visuelle Klassifikation	80
3.4 Analyse der Ergebnisse	80
3.4.1 Vergleich der Methoden	83
3.4.2 Referenzdaten	92
3.4.3 Diskussion	96
3.5 Perspektiven	97
<b>4. Conclusio</b>	<b>100</b>
<b>Literaturverzeichnis</b>	<b>102</b>
<b>Lebenslauf</b>	<b>107</b>

## 0. EINLEITUNG

Das Institut für Photogrammetrie und Fernerkundung (I.P.F.) der TU-Wien hat eine lange Tradition in der Verarbeitung von Fernerkundungsdaten. Im Mittelpunkt des Interesses stehen dabei Entwicklungen sowohl im geometrischen als auch im interpretatorischen Bereich. Seit dem Zeitpunkt, als zum ersten Mal Aufnahmen aus dem Weltraum die Erdoberfläche aus einer neuen Perspektive zeigten, hat die Fernerkundung einen enormen Entwicklungsprozeß durchlaufen. Visuelle Interpretationen analoger Aufnahmen aus den Anfängen der Fernerkundung wurden von automatischen Klassifikationen digitaler Bilder abgelöst. In diesem Zwischenstadium endete die Bearbeitung der Fernerkundungsdaten meist in Form kartographischer Produkte, die von verschiedenen Disziplinen als Informationsquelle genützt wurden. Durch den Einsatz von Sensoren mit höherer Auflösung und dank der rasanten Entwicklung im Bereich der Datenverarbeitung im allgemeinen und der geographischen Informationssysteme (GIS) im speziellen, wurden in der Fernerkundung während der letzten Jahre neue Impulse gesetzt. Sie ist nicht mehr nur Datenlieferant für kartographische Produkte verschiedener Art, sondern hat sich ihren Platz im Bereich der geographischen Informationsverarbeitung geschaffen. Die neue Situation macht es erforderlich, erweiterte Methoden zur Informationsgewinnung aus Satellitenbildern zu entwickeln und geeignete Technologien bei deren Verarbeitung einzusetzen.

Der erste Teil der vorliegenden Arbeit beschäftigt sich mit Verfahren der Klassifikation von Fernerkundungsdaten zur Detektion von Landnutzungsformen. Neben der traditionellen Methode der multispektralen Klassifikation, die die spektralen Charakteristika eines Bildes untersucht, wird die Analyse von Textur in digitalen Bildern und deren Einbindung in den Klassifikationsprozeß besprochen. Durch die hohe räumliche Auflösung der heutigen Fernerkundungssensoren beinhalten die Ergebnisse der Klassifikationen nicht die eigentliche Landnutzung, sondern Oberflächenbedeckungskategorien, aus denen sich die Landnutzung zusammensetzt. Es ist daher notwendig, Postklassifikationsalgorithmen einzusetzen, die durch Analyse der Bedeckungskategorien die gesuchte Landnutzung ermitteln. In diesem Zusammenhang wird ein Verfahren vorgestellt, das im Zuge dieser Arbeit entwickelt wurde. Dieses Verfahren arbeitet nach einem hierarchisch strukturierten Regelsystem, wobei die Entscheidungen auf der Verteilung der Oberflächenbedeckungskategorien in einer lokalen Umgebung basieren. Den Abschluß des ersten Teiles bilden Überblicke über den Einsatz von wissensbasierten Expertensystemen im Klassifikationsprozeß und über den status quo der Integration von Fernerkundung und GIS.

Um die enormen Datenmengen, die bei der Analyse von Fernerkundungsdaten anfallen, in akzeptablen Zeiträumen verarbeiten und adäquat visualisieren zu können, sind geeignete

Technologien erforderlich. Daher wurde am I.P.F. ein Parallelrechner erworben, der mit einem hochauflösenden Graphikschirm ausgestattet ist und über einen PC als Host gesteuert wird. Dieses System 2400 der Wiener Firma IMPULS basiert auf Transputermodulen der britischen Firma INMOS. Bei Lieferung stand keine Software zur Verfügung, daher mußte die Programmentwicklung bei Null begonnen werden. Als Programmiersprache bot sich OCCAM an, da diese Sprache nach der gleichen Philosophie wie der Transputer konzipiert ist. Kapitel 2 beschreibt kurz die Hardwarekomponenten des Systems, gibt eine Einführung in das Konzept kommunizierender, sequentieller Prozesse und geht dann im Detail auf die Entwicklung von Bildverarbeitungsprogrammen ein. Eine genaue Beschreibung der Hardwareaspekte, der entwickelten Benutzeroberfläche und der *low-level* Routinen, auf denen die Anwendungsprogramme aufsetzen, findet sich in der parallel entstehenden Dissertation von Babnik [1994]. In der vorliegenden Arbeit wird ein besonderes Augenmerk auf das Design paralleler Programmschalen gelegt, die eine problemlose Implementierung lokaler Bildverarbeitungsprozesse ermöglichen. In diesem Zusammenhang werden zwei Konzepte vorgestellt, die die Verarbeitung von Bilddaten *screen-to-screen* und *disk-to-disk* ermöglichen. Dabei kommen Aspekte der Kommunikation und Visualisierung im System 2400, der Datenkompression und der Effizienz paralleler Implementierungen zur Sprache.

Den Abschluß der Arbeit bildet die in Kapitel 3 beschriebene Fallstudie. Sie stellt die Umsetzung der in Kapitel 1 vorgestellten Methoden unter Einsatz der in Kapitel 2 entwickelten Software in die Praxis dar. Zur effizienten Planung und Modellierung von zellularen Mobilfunknetzen benötigen die Nachrichtentechniker - neben einem digitalen Höhenmodell - aktuelle, flächendeckende Landnutzungsdaten in digitaler Form. Satellitenbilder bieten sich als Quelle der gesuchten Information an. Ein hybrider Datensatz (Landsat TM und SPOT panchromatisch) vom Raum Wien bildet die Grundlage für die Erprobung einer integrativen Klassifikationsmethode. Eine multispektrale Klassifikation des TM-Bildes ermittelt die Oberflächenbedeckungen des Untersuchungsraumes; eine Texturanalyse der SPOT-Szene ermöglicht eine scharfe Abgrenzung des verbauten Gebietes; eine Verschneidung dieser Ergebnisse liefert die Ausgangsdaten für eine Postklassifikation, die die gesuchten Landnutzungsklassen ermittelt. Als Alternative zum automatischen Ansatz führen zwei Bearbeiter unabhängig voneinander visuelle Interpretationen analoger Orthophotos durch. Um die Plausibilität der eingesetzten Methoden zu prüfen, werden die Ergebnisse mit punktuell im Gelände erhobenen Referenzdaten verglichen und beurteilt. Eine Kalkulation der Kosten sowohl des automatischen als auch des visuellen Verfahrens bildet den Abschluß der Studie.

Kapitel 4 enthält eine Zusammenfassung der vorliegenden Arbeit und zeigt mögliche Perspektiven hinsichtlich der Integration von Fernerkundung und geographischer Informationsverarbeitung auf.

# 1. MUSTERERKENNUNG IN DER FERNERKUNDUNG

## 1.1 EINLEITUNG

Der Begriff *Remote Sensing* (Fernerkundung) stammt aus den 60er Jahren dieses Jahrhunderts und beschreibt die "Gewinnung von Information über Art und Eigenschaft entfernter Objekte" [Kraus et al., 1988, S. 7]. Eine engere Definition gibt Curran [1985, S. 1]: "Verwendung elektromagnetischer Strahlungssensoren zur Aufnahme von Abbildern unserer Umwelt, deren Interpretation nützliche Information liefert". Die Anwendungen lassen sich allgemein mit Erfassung und Überwachung von Zuständen und Veränderungen der Erdoberfläche beschreiben, wobei sowohl natürliche als auch von Menschen geformte Objekte betroffen sind, sofern sie durch die physikalischen Gegebenheiten von den Aufnahmesystemen erfaßt werden können.

Die rasanten Veränderungen und die hohe Belastungen, denen unsere Umwelt ausgesetzt ist, machen eine laufende Erfassung und Kontrolle der Erdoberfläche notwendig. Terrestrische Methoden oder Interpretationen von Luftbildern liefern zwar genaue Informationen, können aber in ihrer Fülle nicht bewältigt werden. Die Fernerkundung liefert aktuelle, flächendeckende Information in Form digitaler oder digitalisierbarer Bilder, die einer automatischen oder zumindest semi-automatischen Verarbeitung unterworfen werden können. Die Interpretation dieser Bilddaten ist ein interdisziplinärer Prozeß, der einerseits auf dem Wissen der Experten in den verschiedenen Anwendungsbereichen, andererseits auf dem sinnvollen Einsatz bildverarbeitender Prozesse in Computern zur Unterstützung der Anwender basiert. Der zweite Aspekt führt in das umfassende Gebiet der Mustererkennung (*pattern recognition*).

## 1.2 MUSTERERKENNUNG

Unter Mustererkennung versteht man "die Einteilung von Bilddaten in identifizierbare Klassen durch die Extraktion signifikanter Merkmale und Attribute dieser Daten" [vgl. Argialas und Harlow, 1990]. Ziel der Mustererkennung ist "die Transformation von Bildinhalten in einen semantisch festgelegten Bedeutungsinhalt" [vgl. Kraus 1990, S. 564]. Methodisch umgesetzt bedeutet dies, das visuelle Wahrnehmungssystem des Menschen im Computer zu simulieren, also dem Computer "sehen" und "verstehen des Gesehenen" beizubringen. Viele Prozesse der Mustererkennung resultieren daher aus den Erkenntnissen der Wahrnehmungsforschung. Einen guten Einblick in die Grundlagen der visuellen Wahrnehmung findet man in *Wahrnehmung und visuelles System* [1986].

Bildinterpretationsmethoden werden zumeist in drei Niveaus eingeteilt, wobei die Grenzen dieser Einteilung als fließend zu betrachten sind:

- Zum *low level* (niedriges Niveau) zählt man **Nachbarschaftsoperationen**, die die Umgebung eines Bildelements heranziehen und daraus einen neuen Wert für das Bildelement berechnen. Diese Prozesse werden für die Bildverbesserung und Bildaufbereitung eingesetzt, dienen aber auch zur einfachen Segmentierung des Bildinhaltes. Die Datenmenge wird durch diese Verfahren in der Regel nicht vermindert, es erfolgt lediglich eine numerische Transformation der Grauwerte.
- Der *medium level* (mittleres Niveau) beinhaltet vor allem **Segmentierungsverfahren**, die komplexere Merkmale zur Verarbeitung heranziehen. Dazu gehören *region growing*, Erfassung von Objektgrenzen, die multispektrale Klassifikation und Analyse von Textur und Objektform. Dabei wird die Datenmenge zugunsten einer höheren Informationsebene des Bildinhaltes reduziert. Das Ergebnis dieser Verfahren sind Bildsegmente oder Objekte, inklusive deren aus den Bildern abgeleiteten Attribute.
- Der *high level* (hohes Niveau) beschäftigt sich mit der Umsetzung der Objekte in **semantische Beschreibungen**. Diese Verfahren bedienen sich der Erkenntnisse aus dem Bereich der *artificial intelligence*. In wissensbasierten Expertensystemen werden die Resultate der niedrigeren Niveaus zusammen mit externer Information und menschlicher Erfahrung in Form von Regelsystemen in komplexen bildverstehenden Prozessen vereinigt. Als Ergebnis werden Bildbeschreibungen gefordert, in der Art, wie sie ein erfahrener Luftbildinterpret liefern würde.

Einen ausgezeichneten Überblick über den aktuellen Stand heute eingesetzter Bildinterpretationsmodelle, der auch als primäre Quelle für diese Zusammenfassung verwendet wurde, findet man in *Argialas und Harlow* [1990].

Der praktische Einsatz der Mustererkennung in der Fernerkundung reicht heute bis zum mittleren Niveau. Allerdings kann dieser Bereich bei weitem nicht als im wissenschaftlichen Sinne abgeschlossen betrachtet werden, wie auch die Vielzahl der aktuellen Publikationen auf diesem Gebiet zeigt (siehe Kapitel 1.4 und 1.6). Auch die vorliegende Arbeit versucht einen Beitrag zu diesem Bereich der Mustererkennung zu liefern.

## 1.3 SPEKTRALE KLASSIFIKATION

### 1.3.1 DIGITALE BILDER

Digitale Bilder sind diskrete, zweidimensionale Abbildungen von dreidimensionalen Objekten in Form einer Matrix, deren Elemente Pixel (Kurzform für *picture elements* = Bildelemente) genannt werden. Jedem Pixel sind ein Punkt (Fläche) im Objektraum und ein Wert (Grauwert) zugeordnet, der je nach Aufnahmeverfahren die Intensität der Reflexion und/oder Emission elektromagnetischer Wellen des Objektpunktes repräsentiert. Multispektrale Bilder setzen sich aus mehreren digitalen Bildern zusammen, die denselben Objektraum darstellen (gleiche Aufnahmegeometrie), aber unterschiedliche Spektralbereiche repräsentieren. Ein multispektrales Bild ist somit eine dreidimensionale Matrix, auch Bildwürfel genannt, deren dritte Dimension der Anzahl der erfaßten Spektralbereiche - auch Kanäle oder Bänder genannt - entspricht. Ein Objektpunkt wird dementsprechend durch einen Pixelvektor dargestellt. In der Folge bezeichnet der Ausdruck Pixel sowohl ein Einzelpixel als auch einen Pixelvektor.

In der Fernerkundung ist der abgebildete Objektraum die Oberfläche der Erde. Ein Pixel beinhaltet dementsprechend die Emission/Reflexion eines Teiles der Erdoberfläche. Die Spektralbereiche und die Größe des Flächenteils hängen von der spektralen und von der räumlichen Auflösung des Sensors ab.

**Multispektrale Fernerkundungsbilder** sind von einem Sensor aufgenommene, mehrkanalige Bilder. Unter **multisensoralen Fernerkundungsbildern** versteht man multispektrale Bilder, deren Kanäle aus Aufnahmen verschiedener Sensoren resultieren. **Multitemporale Fernerkundungsbilder** setzen sich aus Aufnahmen eines Sensors zusammen, die zu unterschiedlichen Zeitpunkten durchgeführt wurden.

### 1.3.2 MULTISPEKTRALE KLASSIFIKATION

Unter Klassifikation versteht man die "systematische Einteilung von Dingen oder Begriffen, die durch gemeinsame Merkmale miteinander verbunden sind, in Klassen und Teilklassen" [vgl. Brockhaus, 1984]. Angewandt auf digitale Bilder bedeutet das die Segmentierung der Bilddaten und die Zuordnung der Segmente zu vorgegebenen Klassen.

Multispektrale Bilder spannen einen mehrdimensionalen, spektralen Merkmalsraum auf. Die Dimension dieses Merkmalsraumes entspricht der Anzahl der Kanäle im multispektralen Bild. Pixel mit einer ähnlichen spektralen Charakteristik bilden Punktwolken (*cluster*) im

Merkmalsraum. Diese *cluster* werden durch statistische Größen beschrieben und zur Segmentierung des Bildes herangezogen. Die Bildung der *cluster* kann auf zweierlei Arten erfolgen: überwacht und unüberwacht.

Bei der **überwachten Klassifikation** werden zuerst sogenannte Trainingsgebiete definiert. Das sind Bereiche im Bild, die jeweils ausschließlich Elemente einer Klasse enthalten. Die spektralen Signaturen - statistische Beschreibungen - dieser Trainingsgebiete definieren die *cluster* im Merkmalsraum. Anschließend wird jedes Pixel über eine Distanzfunktion einem *cluster* und damit einer Klasse zugeordnet.

Die **unüberwachte Klassifikation** bedient sich des sogenannten *cluster*-Analyse [vgl. z.B. Hair et al., 1987], um Punktwolken im Merkmalsraum zu finden. Das sind statistische Verfahren, die ohne interaktive, menschliche Eingriffe den Merkmalsraum in einzelne *cluster* zerlegen. Die Segmentierung erfolgt in zwei Schritten. Zuerst werden die *cluster* definiert, dann erfolgt die Zuordnung der einzelnen Pixel zu den *clustern* [vgl. z.B. ERDAS, 1991]. Den Abschluß der Klassifikation bildet die interaktive Zuweisung der Objektklassen zu den Bildsegmenten.

Es seien noch zwei Charakteristika dieser Klassifikationsmethoden hervorgehoben, die sich aus den letzten Absätzen ableiten lassen:

- Beide Verfahren führen "nur" die Segmentierung automatisiert durch, die thematische Interpretation der Objekte erfolgt interaktiv über die Zuordnung der Objektklassen zu den *clustern*, im ersten Fall vor, im zweiten nach der Segmentierung.
- Beide Verfahren arbeiten pixelorientiert, d.h. die Segmentierung erfolgt ohne Rücksicht auf Nachbarschaftsbeziehungen im Bild. Auf diese Problematik wird in den folgenden Kapiteln noch näher eingegangen.

Ein weiteres Merkmal der multispektralen Klassifikation ist die Wahl der geeigneten Distanzfunktion, d.h. die Frage, nach welchem Kriterium ein Pixel dem einen oder anderen *cluster* im n-dimensionalen Merkmalsraum zugeordnet werden soll. Die am meisten verbreiteten Funktionen seien kurz charakterisiert:

- Die Klassifizierung nach **kürzester Entfernung** (*minimum distance*) berechnet die räumliche Distanz des zu klassifizierenden Pixels zu den Schwerpunkten aller *cluster*. Zugeordnet wird jenem *cluster*, dessen Schwerpunkt die minimale Distanz zum Pixel aufweist. Geometrisch gesehen wird ein *cluster* durch eine n-dimensionale Kugel repräsentiert.

- Die Klassifizierung nach **Quadern** (*parallel epiped*) verwendet zusätzlich zum Mittelwert auch die Standardabweichung in den  $n$  Achsenrichtungen des Merkmalsraumes. Um die Schwerpunkte der *cluster* werden  $n$ -dimensionale Quader gelegt, deren Ausdehnung einem Vielfachen der jeweiligen Standardabweichung entspricht. Liegt ein Pixel innerhalb eines Quaders, wird es dem *cluster* zugeordnet. Kommt ein Pixel in keinem Quader zu liegen, gilt es als unklassifizierbar und fällt in die sogenannte Rückweisungsklasse. Die Klassifizierung nach Quadern ist qualifizierter, da die verschiedenen starke Streuung der *cluster* in den Achsenrichtungen in der Dimensionierung der Quader berücksichtigt wird.
- Noch einen Schritt weiter geht die Methode der **maximalen Mutmaßlichkeit** (*maximum likelihood*). Dabei wird für jeden *cluster* eine  $n$ -dimensionale Normalverteilung berechnet. Das Distanzkriterium, über das die Zuweisung der Pixel erfolgt, ist in diesem Fall die Wahrscheinlichkeitsdichte. Geometrisch betrachtet handelt es sich um  $n$ -dimensionale Ellipsoide, die sich natürlich noch besser als die bisher beschriebenen Geometrien an die Punktwolken anpassen. Man sollte dabei aber nicht vergessen, daß diese Methode von der Annahme ausgeht, daß es sich bei den gesuchten Clustern um normalverteilte Grundgesamtheiten handelt.

Detailliertere Beschreibungen der multispektralen Klassifikation finden sich in Fachbüchern über Fernerkundung und Digitale Bildverarbeitung. Als Beispiele seien Gierloff-Emden [1989], Gonzales und Wintz [1987] und Kraus [1990] genannt.

Die Entscheidung, welcher der angebotenen Verfahrenswege eingeschlagen wird - ob überwacht oder unüberwacht, *parallel epiped* oder *maximum likelihood* - hängt sowohl vom verfügbaren Bildmaterial, als auch von der Zielsetzung des konkreten Anwendungsfalles, und nicht zuletzt von den subjektiven Präferenzen des Bearbeiters ab. Trotzdem läßt sich der Trend in Richtung überwachte, multispektrale *maximum likelihood* Klassifikation nicht übersehen. Sie ist das am häufigsten angewendete automatische Verfahren zur Detektion der Oberflächenbedeckung aus Fernerkundungsbildern [Bolstad und Lillesand, 1992]. Durch ihre fundierte theoretische Basis und ihr transparentes, gut nachvollziehbares Ablaufschema bildet sie einen geeigneten Ausgangspunkt für weitere Verarbeitungsschritte und Analysen, wie sie in den folgenden Kapiteln näher erläutert werden.

### 1.3.3 MULTISENSORALE UND MULTITEMPORALE KLASSIFIKATION

Eine Erweiterung der multispektralen Klassifikation ist die Einbeziehung von Bilddaten verschiedener Sensoren in einen gemeinsamen Klassifikationsprozeß. Der Vorteil dieser

Methode liegt im vergrößerten Informationsspektrum hybrider Datensätze. Heutige Fernerkundungssensoren liefern eine Vielzahl verschiedener Datensätze, die in ihren spektralen und räumlichen Ausprägungen variieren. Die Charakteristika eines Fernerkundungsbildes ergeben sich aus der spektralen Auflösung und Bandbreite der Bildkanäle sowie der räumlichen Auflösung eines Pixels. In der Praxis stehen spektral hochauflösende Sensoren mit schwacher räumlicher Auflösung räumlich hochauflösenden Sensoren mit schwacher spektraler Auflösung gegenüber. Eine Kombination solcher Datensätze mit entgegengesetzten Charakteristika nützt die Vorteile beider Seiten und verbessert das Ergebnis im Vergleich zur rein multispektralen Klassifikation.

Diese Methode wurde im Zuge einer Diplomarbeit am I.P.F. für eine Landnutzungs-klassifikation des Nationalparks Neusiedlersee - Fertő Tó angewandt [vgl. Sindhuber, 1993, Csaplovics et al., 1993]. Ziel der Arbeit war eine Unterscheidung extensiv und intensiv genutzter Naturräume im Seewinkel. Das verwendete Datenmaterial umfaßte multispektrale Landsat TM-Daten und gescannte KFA-1000-Aufnahmen russischer Satelliten der Kosmos Generation. Durch verschiedene Aufnahmezeitpunkte - Juli und Oktober 1991 - enthält die Kombination auch noch multitemporale Charakteristika. Die TM-Szene verfügt über sechs spektrale Kanäle - von Blau bis zum mittleren Infrarot - und eine Pixelauflösung von  $30 \times 30 \text{m}^2$  am Boden. Die KFA-Aufnahme wurde mit  $28 \mu\text{m}$  gescannt - das entspricht  $7.5 \times 7.5 \text{m}^2$  auf der Erdoberfläche - und bietet zwei spektrale Bänder - sichtbares Rot und nahes Infrarot. Die beiden Datensätze entsprechen also den vorhin gestellten Forderungen; auch die Ergebnisse erfüllten die Erwartungen. Konnte mit TM-Daten alleine nur ein ungenaues Resultat erzielt werden, so ermöglichte die multisensorale Klassifikation eine deutliche Unterscheidung der gesuchten Flächen.

Ein wichtiger Aspekt bei der multisensoralen Klassifikation ist die Rektifizierung. Da verschiedene Fernerkundungssensoren verschiedene Aufnahmegeometrien aufweisen, müssen die einzelnen Datensätze vor der Klassifikation geometrisch korrigiert werden. Normalerweise wird man beide Datensätze auf ein übergeordnetes Koordinatensystem geokodieren (z.B. Gauß-Krüger). Dabei werden alle Bilddaten einem *resampling* unterworfen und auf eine gemeinsame Pixelgröße transformiert - d.h., die Klassifizierung wird nicht mehr mit den Originaldaten, sondern mit interpolierten Daten durchgeführt.

Es besteht aber auch die Möglichkeit, eine Aufnahme auf die andere zu rektifizieren. Das hat den Vorteil, daß ein Datensatz abgesehen von der Veränderung der Pixelgröße, unverändert bleibt. Rektifiziert man die räumlich hoch aufgelösten Daten auf die spektral hochaufgelösten und wählt man als gemeinsame Pixelgröße einen ganzzahligen Teiler der groben Auflösung (z.B.  $10 \times 10 \text{m}^2$  bei TM), so bleiben die spektral signifikanten Bilddaten völlig unverändert. Die Änderung, die das *resampling* bei den räumlich fein aufgelösten Daten herbei-

führt, ist nicht so gravierend, da diese Daten in erster Linie strukturelle und nicht spektrale Information liefern. Im Anschluß an die gemeinsame Klassifikation muß natürlich das Ergebnis zur weiteren raumbezogenen Verarbeitung auf ein übergeordnetes Koordinatensystem geokodiert werden.

Eine Alternative zur oben erwähnten Problematik bieten Methoden der Postklassifikation. Dabei werden multisensorelle Daten einzeln klassifiziert, die Ergebnisse rektifiziert und erst dann miteinander verschnitten. Dieser Vorgangsweise wird in der in Kapitel 3 beschriebenen Fallstudie angewendet.

## 1.4 TEXTUR

### 1.4.1 BESCHREIBUNG VON TEXTUR IM DIGITALEN BILD

Die Fragestellung bezieht sich auf eine mathematisch beschreibbare Quantifizierung des Ausdruckes Textur. Begriffe wie 'fein, grob, glatt, linienhaft, körnig usw.' sind im sprachlichen Kontext verständlich, entziehen sich aber einer eindeutigen mathematischen Definition. Weiters haben sie eine komplexe Struktur, d.h. sie besitzen mehrere Freiheitsgrade und sind dementsprechend nicht eindimensional beschreibbar, im Gegensatz zum Parameter Ton, der über seine Intensität eindeutig definiert werden kann.

**Textur und Ton** stehen in unmittelbarem Zusammenhang. Weist ein Bildausschnitt wenig Variation in den Grauwerten auf, so ist der Ton die dominante Eigenschaft. Bei starken Grauwertschwankungen hingegen ist die Textur das vorherrschende Merkmal. Entscheidend wirkt sich auch die Größe des Gebietes im Kontext mit der geometrischen Auflösung des Bildes aus. Mit zunehmender Anzahl von Bildelementen erhöht sich die texturale Ausprägung des Gebietes, im krassen Gegensatz dazu liefert ein einzelnes Bildelement keine texturale Information [vgl. Irons und Petersen, 1981].

Im Sinne eines komplexeren Klassifikationsansatzes muß die gesuchte Beschreibung der texturalen Eigenschaften eines Bildes in den Klassifikationsprozeß einfließen. Gefordert wird dementsprechend ein digitales Bild, dessen Grauwerte die Textur des Ausgangsbildes repräsentieren. Da solche Texturbilder nicht direkt erfaßt werden können, müssen sie vom Ausgangsbild abgeleitet werden.

In der Literatur finden sich eine Vielzahl von Ansätzen, die sich mit **Quantifizierung von Textur** in digitalen Bildern auseinandersetzen. Einen aktuellen Überblick geben Sali und Wolfson [1992], dessen Quintessenz hier wiedergegeben wird:

- a) *Statistische Parameter erster Ordnung*, wie z.B. Varianz oder Mittelwert, sind einfach zu berechnen und zu interpretieren, treffen aber keine Aussage über die räumliche Verteilung benachbarter Bildelemente.
- b) *Statistische Parameter zweiter Ordnung* unter Verwendung von Grauwertabhängigkeits-Matrizen berücksichtigen sowohl die spektrale als auch die räumliche Verteilung von Grauwerten.
- c) *Fouriertransformation* digitaler Bilder und Analyse des Amplitudenspektrums ermöglicht das Erkennen regelmäßiger Muster im Bild.
- d) Weitere Methoden beschreiben Textur mit Hilfe von *Faltungsoperatoren*, *Histogrammanalysen* oder *fraktalen* Eigenschaften des Bildes.

Anwendungen der o. a. Ansätze finden sich unter anderem bei *Irons und Petersen* [1981], *Franklin und Peddle* [1990], *Webster und Bracken* [1992].

Im folgenden wird näher auf die in Punkt b) angeführten statistischen Parameter zweiter Ordnung eingegangen.

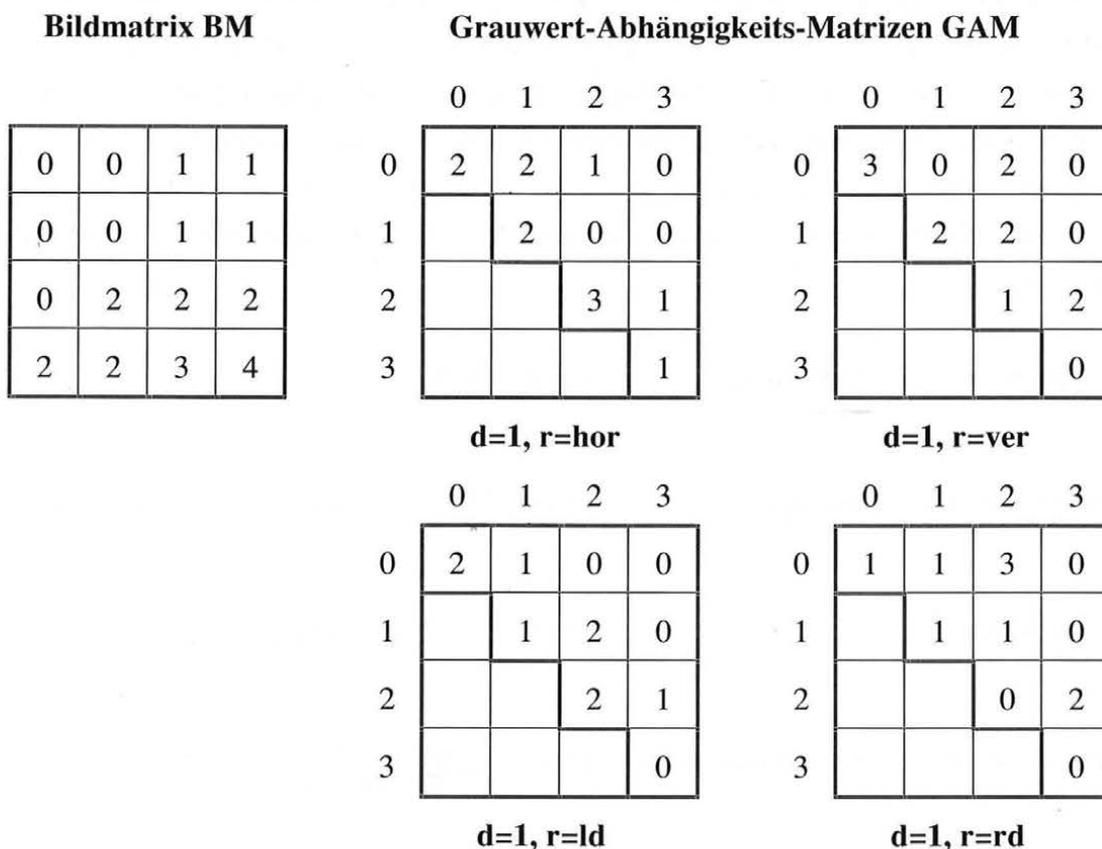
#### 1.4.2 GRAUWERT-ABHÄNGIGKEITS-MATRIZEN

Das menschliche Wahrnehmungssystem ist in der Lage, Texturen zu unterscheiden, deren Statistiken erster und/oder zweiter Ordnung differieren. Texturpaare, die verschiedene Statistiken dritter Ordnung, aber idente Statistiken zweiter und erster Ordnung aufweisen, können nicht mehr differenziert werden. Weiters gilt, daß zwei Texturen, die in ihrer Statistik n-ter Ordnung übereinstimmen, in all ihren Statistiken niederer Ordnung, also erster, zweiter bis (n-1)-ter Ordnung ident sind [vgl. Julesz, 1986]. Es liegt daher nahe, zur Beschreibung texturaler Eigenschaften digitaler Bilder einen Ansatz zu wählen, der Statistiken zweiter Ordnung berücksichtigt, da ein solcher dem menschlichen Wahrnehmungsvermögen am nächsten kommt.

*Haralick et al.* [1973] verwenden Grauwert-Abhängigkeits-Matrizen (GAM) zur Bestimmung texturaler Parameter in einem digitalen Bild. Eine GAM beschreibt die räumliche Beziehung von Grauwerten innerhalb einer Bildmatrix (BM). Sie beinhaltet die normierten Häufigkeiten  $p_{i,j}$ , mit der zwei benachbarte Pixel in einem bestimmten räumlichen Abstand (d) in einer bestimmten Richtung (r) im Bild auftreten, das eine mit dem Grauwert i, das andere mit dem Grauwert j. Die GAM ist somit eine Funktion über das Bild mit den Parametern d und r:  $GAM = f(d,r,BM)$ . Die Reihenfolge der Grauwerte in der BM spielt

keine Rolle, d.h.  $p_{i,j} = p_{j,i}$ , die GAM ist symmetrisch und wird infolge dessen als Dreiecksmatrix dargestellt. Ihre Ausdehnung entspricht der Anzahl der Grauwertstufen in der ursprünglichen Bildmatrix BM.

Zum besseren Verständnis des eben Beschriebenen möge ein Beispiel dienen: Ausgehend von einer Bildmatrix BM mit nur vier Grauwertstufen [0,3] werden die GAMs in den vier Hauptrichtungen berechnet - horizontal (hor), vertikal (ver), links (ld) und rechts diagonal (rd). Die räumliche Distanz (d) beträgt Eins (siehe Abb. 1.1). Zur Erstellung der GAM mit den Parametern Richtung gleich horizontal und Distanz gleich Eins wird in die Matrixelemente  $P_{i,j}$  die absolute Häufigkeit der horizontal im Abstand Eins im Bild BM auftretenden Grauwertpaare  $i$  und  $j$  eingetragen. Das horizontale Grauwertepaar 0,1 zum Beispiel kommt im Bild BM zwei Mal vor.



**Abb. 1.1: Bildmatrix mit Grauwert-Abhängigkeits-Matrizen**  
(entnommen aus *Haralick et al.* [1973])

Die Darstellung einer GAM läßt bereits eine Interpretation des zu analysierenden Bildes zu. Je mehr sich die besetzten Elemente ( $>0$ ) um die Hauptdiagonale scharen, desto homogener ist das Bild, weil benachbarte Grauwerte nicht stark differieren ( $0 \leq (i - j) \ll 256$ ). Von der Hauptdiagonale entfernte Elemente bedeuten starke Grauwertsprünge ( $0 \ll (i - j) < 256$ ) im Bild. Neben der numerischen Darstellung einer GAM (siehe Abb. 1.1) läßt sie sich auch in Form einer Bildmatrix visualisieren. Abb. 1.2 zeigt zwei Beispiele, die die GAMs von Ausschnitten aus Satellitenbilddaten zeigen. Die vertikale Strukturierung der Felder (Abb. 1.2a) drückt sich deutlich in der vertikalen GAM aus (Abb. 1.2b). Die richtungsunabhängig hohe Texturierung der verbauten Struktur (Abb. 1.2c) bewirkt stark streuende GAMs in allen Richtungen (Abb. 1.2d).

### 1.4.3 TEXTURMERKMALE

Wie im letzten Kapitel gezeigt, führt eine visuelle Interpretation der GAMs zu Aussagen über die texturalen Eigenschaften einer vorgegebenen Bildmatrix. Zur weiteren rechnerischen Verarbeitung ist jedoch eine Quantifizierung der Textur in Form eines repräsentativen Wertes gesucht. Über statistische Ansätze läßt sich die in den GAMs enthaltene Information in sogenannten Texturparametern quantifizieren. *Haralick et al.* [1973] geben 14 solcher Texturparameter an, die sich aus einer GAM ableiten lassen. Vier dieser Texturmerkmale<sup>1</sup> sind im folgenden angegeben:

$$\textit{angular second moment (Homogenität)} \quad \text{ASM} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j}^2 \quad (1.1)$$

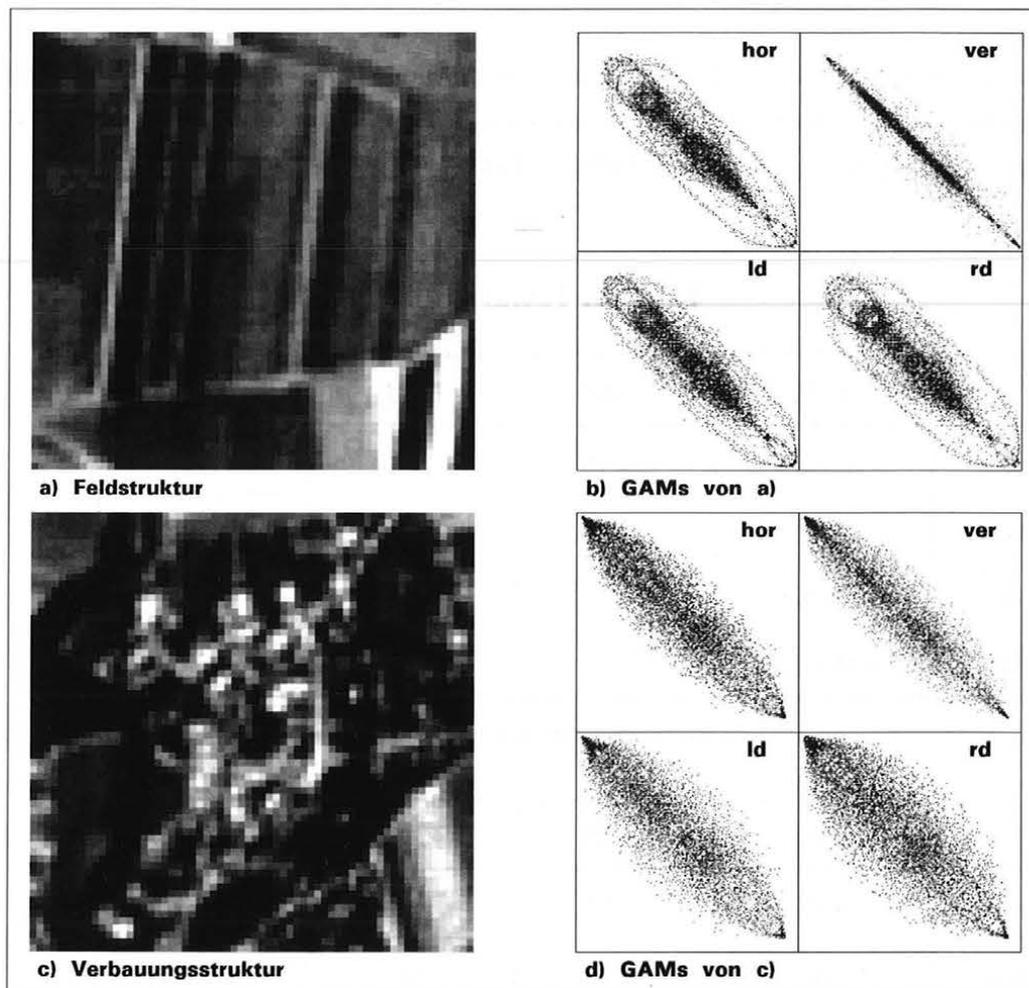
$$\textit{contrast (Kontrast zweiter Ordnung)} \quad \text{CON} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - j)^2 * p_{i,j} \quad (1.2)$$

$$\textit{entropy (Heterogenität)} \quad \text{ENT} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{i,j} * \log(p_{i,j}) \quad (1.3)$$

$$\textit{inverse difference moment (Inverser Kontrast)} \quad \text{IDM} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1 + (i - j)^2} * p_{i,j} \quad (1.4)$$

Die normierten Häufigkeiten  $p_{i,j}$  sind die absoluten Häufigkeiten  $P_{i,j}$  dividiert durch die Summe aller Einträge. Die  $p_{i,j}$  sind dementsprechend sehr kleine Zahlen, die maximal den Wert Eins annehmen können. Da das ASM die Elemente quadratisch aufsummiert, hängt sein

<sup>1</sup> Die deutschen Bezeichnungen sind *Henker* [1989] entnommen.



**Abb. 1.2: Grauwert-Abhängigkeits-Matrizen verschiedener Bildstrukturen**

Wert von der Anzahl der auftretenden Grauwertübergänge ab. Je mehr Grauwertübergänge auftreten, desto größer ist das ASM und desto inhomogener das Bild. Die Grauwertdifferenz in den Übergängen wird nicht berücksichtigt. Die Entropie verhält sich ähnlich, nur daß durch den Logarithmus die Zuordnung invertiert wird - je inhomogener das Bild, umso kleiner die Entropie.

Im Gegensatz dazu berücksichtigen IDM und CON die Größe der Grauwertdifferenzen, ausgedrückt durch den Term  $(i - j)$ . Die Gewichtung erfolgt beim CON direkt - je inhomogener das Bild, umso höher der Wert -, beim IDM reziprok - je inhomogener das Bild, umso niedriger der Wert. Durch die starke Gewichtung hoher Grauwertdifferenzen erkennt der CON plötzliche starke Grauwertsprünge sehr gut. Das IDM reagiert sensibler auf schwache Grauwertschwankungen. Tab. 1.1 zeigt die Werte der beschriebenen Texturparameter, abgeleitet aus den Beispielen der Abb. 1.2.

Es sei noch auf einige Literaturstellen hingewiesen, die die erwähnten Texturmerkmale und ihre Charakteristika sowohl theoretisch als auch in der Anwendung beschreiben: *Franklin und Peddle* [1990], *Sali und Wolfson* [1992], *Webster und Bracken* [1992].

**Tab. 1.1: Texturparameter**

	ASM	CON	ENT	IDM
<b>hor</b>	0.00028	380	8.4	0.092
<b>ver</b>	0.00053	80	7.9	0.246
<b>ld</b>	0.00026	400	8.5	0.081
<b>rd</b>	0.00025	510	8.5	0.081

**Texturparameter berechnet aus Abb. 1.2 a)**

	ASM	CON	ENT	IDM
<b>hor</b>	0.00021	550	8.6	0.077
<b>ver</b>	0.00023	430	8.6	0.090
<b>ld</b>	0.00020	780	8.7	0.061
<b>rd</b>	0.00018	1070	8.7	0.058

**Texturparameter berechnet aus Abb. 1.2 c)**

Um Texturinformation in den Prozeß einer Klassifikation einfließen zu lassen, benötigt man jedoch nicht einzelne, das ganze Bild beschreibende Texturmerkmale, sondern ein digitales Bild, das die räumliche Verteilung der Textur des Ausgangsbildes darstellt. Um jedem Pixel des ursprünglichen Bildes einen Texturmerkmalswert zuordnen zu können, wird zuerst für die Umgebung des Pixels eine GAM berechnet, aus der dann ein Texturmerkmal abgeleitet und als Grauwert in das Texturbild geschrieben wird. Das Resultat einer solchen lokalen Texturanalyse ist ein digitales Bild (Texturmerkmalsbild), dessen Grauwerte die texturale Charakteristik über ihre Intensität repräsentieren (z.B. hell - homogen, dunkel - inhomogen).

Wählt man den zuvor beschriebenen Ansatz der Texturanalyse über die GAM und deren Texturmerkmale, so ist leicht zu erkennen, daß sich aus einem vorgegebenen digitalen Bild eine Vielzahl von Texturmerkmalsbildern ableiten läßt. Abgesehen von der Wahl eines geeigneten

Texturmerkmale bestehen Variationsmöglichkeiten in den Parametern Richtung und Distanz sowie in der Größe der zu untersuchenden räumlichen Nachbarschaft.

Die zu analysierenden Pixelpaare werden durch die Parameter Richtung und Distanz definiert. Durch die Matrixform digitaler Bilder ergibt sich eine Abhängigkeit der Parameter Richtung und Distanz. Definiert man die Distanz Eins als nächsten Nachbarn, so hat dieser zwar in den Richtungen horizontal und vertikal den räumlichen Abstand Eins, in diagonaler Richtung jedoch den Abstand  $\sqrt{2}$ , d.h. es können horizontale und vertikale Texturmerkmalsbilder nicht mit diagonalen Texturmerkmalsbildern quantitativ verglichen werden.

#### 1.4.4 SKALIERUNG VON TEXTURMERKMALSBILDERN

Der Wertebereich der Texturmerkmale umfaßt theoretisch die reellen positiven Zahlen. Zur Darstellung eines Texturmerkmalsbildes muß dieser Wertebereich auf die zur Verfügung stehenden Grauwerte skaliert werden. Der de facto notwendige Wertebereich variiert natürlich in Abhängigkeit des jeweiligen Bildes und der gewählten Parameter.

Wird das Texturmerkmalsbild als *real*-Wert-Bild bei der Berechnung gespeichert, so kann während der Verarbeitung eine Bildstatistik erstellt werden - minimaler und maximaler Grauwert, Mittelwert und Standardabweichung -, mittels derer anschließend die Texturdaten auf den Grauwertebereich - z.B. 0 bis 255 - skaliert werden können. Diese Methode ist üblich, hat aber den Nachteil, daß sie sehr speicherplatzintensiv ist. Um ein Bild in *real*-Werten zu speichern, ist zumindest der vierfache Speicherplatz des Ausgangsbildes notwendig, sofern man von einer Byte- Repräsentation (256 Grauwerte) desselben ausgeht. Es stellt sich daher die Frage, ob es nicht möglich sei, a priori Aussagen über die zu erwartenden Texturmerkmalswerte zu treffen.

Ein repräsentativer Wert, von dem man ausgehen könnte, ist der eines Texturmerkmals des gesamten Bildes (vgl. Tab. 1.1). Er ist schnell zu berechnen und stellt eine Art mittleren Texturwert für das gesamte Bild dar. Inwieweit er tatsächlich das arithmetische Mittel des Texturmerkmalsbildes repräsentiert, gilt es festzustellen.

Berechnet man die globale GAM (**GAM**) eines gesamten Bildes, so werden die Häufigkeiten der Grauwertpaare in einer Distanz  $d$  und einer Richtung  $r$  erfaßt. Läßt man ein Fenster pixelweise über das Bild wandern, innerhalb dessen jeweils eine lokale GAM (**gam**) erstellt wird, so werden mit Sicherheit dieselben Grauwertpaare erfaßt, die meisten allerdings mehrfach. Zur Illustration des Gesagten ein Beispiel: Beträgt die Fenstergröße  $3 \times 3$ , die Distanz Eins und ist die Richtung horizontal oder vertikal, so wird jedes Grauwertpaar sechsmal erfaßt - an

den sechs möglichen Positionen innerhalb der 3x3 Umgebung. Eine Ausnahme stellen Randpixel dar, die jedoch vernachlässigt werden können. Summiert man alle **gams** und normiert sie, so ergibt sich die **GAM**. Mit anderen Worten, die normierten globalen Häufigkeiten  $P_{i,j}$  sind der Mittelwert der normierten lokalen Häufigkeiten  $p_{i,j}$ ,  $P_{i,j} = \Sigma p_{i,j} / R$ , mit  $R = \text{Anzahl der gams im Bild}$ . Die aus den GAMs abgeleiteten Texturmerkmale sind Summen einer Funktion der normierten Häufigkeiten, allgemein  $\Sigma \Sigma f(P_{i,j})$ . Es gilt also:

$$\begin{aligned} TM &= \Sigma \Sigma f(P_{i,j}) && \text{globales Texturmerkmal} \\ tm &= \Sigma \Sigma f(p_{i,j}) && \text{lokales Texturmerkmal} \end{aligned} \quad (1.5)$$

$$P_{i,j} = \Sigma p_{i,j} / R$$

Die Frage ist, ob sich daraus ableiten läßt, daß das globale Texturmerkmal der Mittelwert der lokalen Texturmerkmale ist, also ob

$$TM = \Sigma (tm / R) \quad (1.6)$$

gültig ist. Durch Einsetzen der Terme aus Formel (1.5) in Formel (1.6) ergibt sich

$$\Sigma \Sigma f(P_{i,j}) = \Sigma (\Sigma \Sigma f(p_{i,j}) / R) \quad (1.7)$$

und daraus abgeleitet

$$\Sigma \Sigma f(\Sigma p_{i,j} / R) = \Sigma (\Sigma \Sigma f(p_{i,j}) / R) \quad (1.8)$$

Formel (1.8) hat nur dann Gültigkeit, wenn es sich um eine lineare Funktion  $f$  handelt, denn nur dann gilt, daß der Mittelwert der Funktionswerte der Variablen dem Funktionwert des Mittelwertes der Variablen entspricht. Diese Forderung wird von den Texturmerkmalen CON und IDM erfüllt.

Die Berechnung eines CON- oder IDM-Bildes erfolgt demgemäß in zwei Schritten. Zuerst wird der mittlere Texturmerkmalswert des Ausgangsbildes berechnet. Danach dient dieser Wert bei der Erstellung des Texturmerkmalsbildes als Skalierungsfaktor:

$$\text{Multiplikationskonstante} = \text{mittlerer Grauwert} / \text{mittlerer Texturwert.}$$

## 1.5 TEXTURALE KLASSIFIKATION

Texturmerkmalsbilder, wie sie in Kapitel 1.4.3 beschrieben wurden, sind digitale Bilder, deren Grauwerte texturale Merkmale eines Bildes repräsentieren. Sie können auf zwei Arten einer Klassifikation unterworfen werden:

- über eine Segmentierung einer Gruppe von Texturmerkmalsbildern, oder
- als künstliche Kanäle im Zuge einer erweiterten multispektralen Klassifikation.

### 1.5.1 SEGMENTIERUNG VON TEXTURMERKMALSBIldERN

Im ersten Ansatz bilden  $n$  Texturmerkmalsbilder einen  $n$ -dimensionalen Merkmalsraum. Die Bilder entstehen durch Variation der Parameter der Texturanalyse - verschiedene Richtungen, Distanzen und Umgebungsgrößen. Über statistische Verfahren wird der Merkmalsraum in einzelne *cluster* geteilt. Das Ergebnis sind segmentierte Bilder, deren Klassen Gebiete statistisch ähnlicher Texturen repräsentieren. Solche Texturklassifikationen zeigen eine flächige Ausprägung, da sich durch die Einbeziehung der räumlichen Umgebung bei der Berechnung die Texturmerkmalswerte nur fließend ändern. Dieses Verfahren ist vor allem für einkanalige Bilddaten geeignet - z.B. SPOT panchromatisch, Radarbilder, gescannte Luftaufnahmen -, da sich diese nicht über den multispektralen Ansatz klassifizieren lassen [Barber und LeDrew, 1991, Connors et al., 1984, Peddle und Franklin, 1991, Sali und Wolfson, 1992].

Als Beispiel für eine Segmentierung von Texturmerkmalsbildern wird ein Projekt präsentiert, das als Teil einer Diplomarbeit am I.P.F. durchgeführt wurde [vgl. Aimet, 1992]. Ziel war die Erkennung verbauter Gebiete in einem panchromatischen SPOT Bild (Abb. 1.3a). Die Aufnahme zeigt ein ländliches Gebiet südlich von Tulln. Für den menschlichen Betrachter sind drei Landnutzungsformen deutlich erkennbar: landwirtschaftlich genutztes Gebiet, verbautes Gebiet und Wald. Die Unterscheidung der genannten Bereiche erfolgt primär über texturale Charakteristika und erst sekundär über den Ton. Wald ist homogen, Felder weisen eine linienhafte Struktur in zwei ungefähr rechtwinklig aufeinander stehenden Richtungen auf und verbaute Gebiete zeichnen sich durch eine richtungsunabhängige hohe Texturierung aus.

Der gewählte Ansatz für die Texturanalyse folgt der Aussage des letzten Satzes: es werden diejenigen Bildbereiche gesucht, die in den vier Richtungen - horizontal, vertikal, links und rechts diagonal hohe Texturwerte aufweisen. Zuerst mußte jedoch das geeignete Texturmerkmal gewählt werden. ASM und ENT haben eine zu schwache Ausprägung, um eine deutliche Unterscheidung zu ermöglichen. CON liefert sehr deutliche Feldgrenzen und einzelne Punkte

im verbauten Gebiet - starkes Ansprechen auf linienhafte und punktförmige Grauwertänderungen. Es ergeben sich jedoch keine flächigen "Flecken" in den verbauten Gebieten. Da das IDM auf schwache Grauwertschwankungen stärker anspricht und flächigere Ergebnisse liefert, wurde es als geeignetes Texturmerkmal ausgewählt (vgl. Kap. 1.4.3).

Als räumliche Distanz wird Zwei gewählt, da bei größeren Distanzen Gebiete, die visuell als homogen erkannt werden, in zunehmendem Maße als hochtexturiert ausgewiesen werden. Innerhalb der verbauten Gebiete scheint diese Distanz (20m in der Natur) adäquat zur Erkennung dörflicher Strukturen. Die Berechnung erfolgte in einer 3x3 Umgebung (Abb. 1.3b).

Als Richtung wurden alle vier verfügbaren Orientierungen verwendet - horizontal, vertikal, links und rechts diagonal -, weil dadurch die Richtungsunabhängigkeit am verlässlichsten erkannt werden kann. Wie bereits in Kap. 1.4.3 erwähnt, lassen sich die Texturmerkmalsbilder mit Ausrichtung horizontal oder vertikal mit diagonalen Richtungsbildern nicht absolut vergleichen. Daher wurde eine Vorgehensweise gewählt, die die Verknüpfung der einzelnen Texturmerkmalsbilder erst nach einer Verarbeitungsphase, die die Werte relativiert, erlaubt.

Im ersten Schritt wurden die Bilder mit einem 9x9 Mittelwertfilter geglättet, was einer Texturmerkmalsberechnung mit einer 9x9 Matrix entspricht (vgl. Kap. 1.4.4, Abb. 1.3c). Danach erfolgte eine Segmentierung der einzelnen Bilder in stark contra schwach texturierte Gebiete mittels eines Schwellwertes. Die Wahl der Schwellwerte erfolgte über kleine Testgebiete in verbauten Bereichen. Das Ergebnis sind vier binäre Bilder, deren hoch respektive niedrig texturierte Bereiche den Wert 0 respektive 1 erhalten (Abb. 1.3d). Die einzelnen Bilder weisen natürlich mehr als nur die verbauten Gebiete aus, da jeweils nur die Textur einer Richtung erfaßt wurde. In ihrer Gesamtheit jedoch beinhalten sie die gesuchte Information. Durch eine Verschneidung der vier binären Bilder über eine logische UND-Verknüpfung ergibt sich ein Layer, der nur die in allen vier Eingangsbildern als hoch texturiert ausgewiesenen Gebiete enthält. Dieser Layer wird über einen 5 x 5 Medianfilter geglättet, um einzelne Ausreißer zu eliminieren. Das Ergebnis zeigt deutlich die verbauten Gebiete, weist allerdings einzelne kleine Flächen in landwirtschaftlich genutzten Bereichen auf (Abb. 1.3e). Diese sind auf lokale Phänomene zurückzuführen - z.B. Feldecken, an denen mehrere Linienstrukturen aufeinandertreffen und die daher in allen Richtungen eine hohe Textur aufweisen. Die Größe dieser Fehlklassifizierungen beträgt jedoch nur einige Pixel. Da verbaute Gebiete per definitionem eine bestimmte Größe nicht unterschreiten, können diese Effekte über eine einfache Regel beseitigt werden: Alle zusammenhängenden Flächen kleiner als 300 Pixel (=30000 m<sup>2</sup>) werden eliminiert (der Gegenklasse zugeordnet). Das Ergebnis ist als Overlay über das Original in Abb. 1.3f dargestellt.

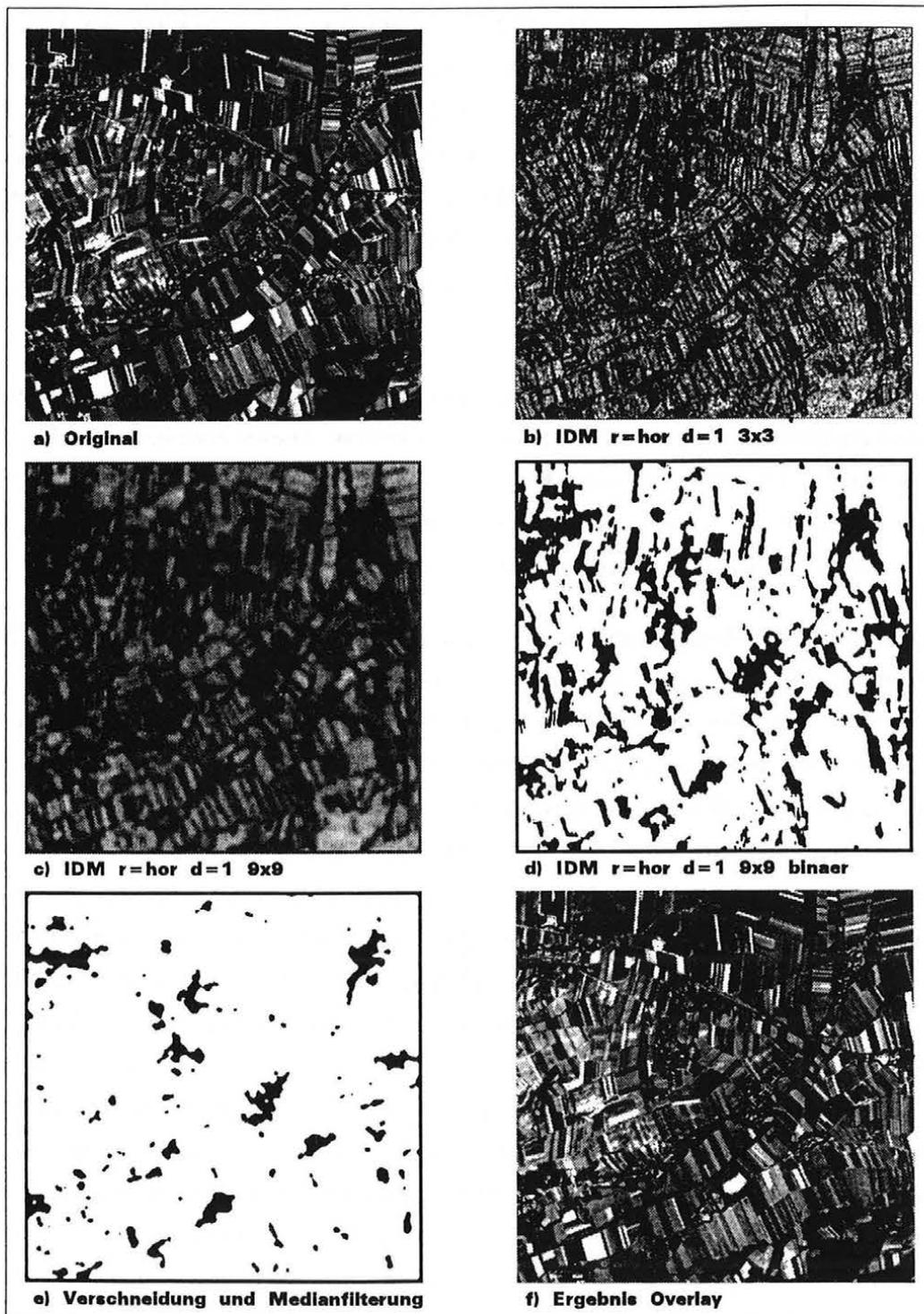


Abb. 1.3 Texturanalyse zur Bestimmung verbauter Gebiete

### 1.5.2 TEXTURMERKMALSBLDER ALS KÜNSTLICHE KANÄLE

In der multispektralen Klassifikation wird ein mehrdimensionaler spektraler Merkmalsraum über statistische Verfahren in einzelne Cluster geteilt, die die verschiedenen Klassen repräsentieren (vgl. Kap. 1.3.2). Als Ausgangsbild dienen mehrere spektrale Bänder eines digitalen Bildes. Die Methode basiert auf der pixelweisen Klassifizierung des multispektralen Bildes, d.h. es werden die Pixelvektoren unabhängig voneinander verarbeitet; Nachbarschaftsbeziehungen werden nicht berücksichtigt. Da aber das visuelle Wahrnehmungsvermögen des Menschen sowohl spektrale Information als auch deren räumliche Verteilung interpretiert, erscheint es naheliegend, in den Prozeß der Klassifikation auch Grauwertverteilungsgrößen einfließen zu lassen.

Texturmerkmalsbilder, die aus den Kanälen eines multispektralen Bildes abgeleitet wurden, können als künstliche Kanäle den Merkmalsraum erweitern. Dieser erweiterte Merkmalsraum wird einer multispektralen(-texturalen) Klassifikation unterworfen. Ein Vergleich spektraler, texturaler und kombinierter Klassifikationen zeigt eine deutliche Verbesserung der Ergebnisse durch Einbindung von Texturinformation [vgl. Franklin und Peddle, 1990].

Ein Problem dieser Methode stellt das Anwachsen der Datenmenge durch die Einbeziehung der Texturmerkmalsbilder dar. Geht man von einer 6-kanaligen Landsat TM-Szene aus, in der aus zwei oder drei Kanälen Texturmerkmalsbilder in vier Richtungen, mit den Distanzen 1, 2 und 3 und verschiedenen Umgebungsgrößen abgeleitet werden, so erhält man Kanalkombinationen, deren Ausmaß beträchtlich ist. Will man darüber hinaus noch mit multisensoralen Datensätzen arbeiten, wird selbst das beste Bildverarbeitungssystem überfordert sein. Um aber trotzdem nicht auf die texturale Information verzichten zu müssen, bieten sich statistische Methoden an, die eine optimale Auswahl der Kanäle für eine gewählte Applikation ermöglichen. Ein solcher Ansatz, der anhand eines multivariaten Kriteriums die Qualität einer multispektralen Landnutzungsklassifizierung unter Einbeziehung verschiedener Texturkanäle schätzt, findet sich in *Pyka und Steinnocher* [1993]. Die Analyse der Datensätze auf Grund eines Spurkriteriums erlaubt eine a priori Auswahl der optimalen Datensätze ohne großen Datenverwaltungsaufwand, da nur die Daten innerhalb der ausgewählten Trainingsgebiete verwendet werden. Darüber hinaus ermöglicht diese Methode eine Datenreduktion in der Art einer Hauptkomponententransformation, deren Parameter nur aus den Trainingsdaten abgeleitet sind. Die Ergebnisse dieser Arbeit zeigen, daß die Klassifikation ausgewählter Datensätze inklusive Texturmerkmalsbildern selbst nach einer Datenreduktion deutlich aussagekräftiger ist als eine Klassifikation rein multispektraler Daten.

Zum Abschluß dieses Kapitels sei noch auf ein Problem hingewiesen, das fast zwingend bei der Verarbeitung von Texturmerkmalsbildern auftritt. Grenzen in einem Bild zwei homo-

gene, aber spektral differierende Flächen aneinander - z.B. verschieden bebaute Felder -, so werden diese im Texturbild zwar als homogen ausgewiesen, weisen jedoch entlang ihrer Kante eine hohe Textur auf. Dieser Randstreifen hat je nach gewählter Umgebungsgröße eine Ausdehnung von einigen Pixeln. Klassifiziert man ein soches Gebiet rein multispektral, entsteht kein Fehler. Inkludiert man den Texturkanal, entstehen jedoch statt zwei vier Klassen - Feld eins homogen, Feld eins inhomogen, Feld zwei inhomogen, Feld zwei homogen. Die erste und die vierte Klasse sind die gesuchten, die mittleren beiden sind Fehlklassifikationen, die jedoch leicht korrigiert werden können: Handelt es sich um eine überwachte Klassifikation, so sind die Randbereiche normalerweise nicht trainiert, sondern werden mit hoher Wahrscheinlichkeit der Rückweisungsklasse zugeordnet. Handelt es sich um eine unüberwachte Klassifikation, entstehen eigene "Randklassen", die bei der Interpretation als Rückweisungsklassen ausgewiesen werden. Die entstehende Lücke zwischen den Feldern kann über einen *region growing*-Algorithmus wieder geschlossen werden.

## 1.6 POSTKLASSIFIKATION

Unter Postklassifikation versteht man Prozesse, die auf das Ergebnis einer pixelweisen Klassifikation angesetzt werden, um sie zu überarbeiten. Im ersten Kapitel werden simple Algorithmen beschrieben, die zufällige Fehler, wie sie durch Rauschen im Originalbild entstehen, korrigieren. Sie sind einfach einzusetzen, haben kurze Laufzeiten und bewirken in erster Linie eine optische Verbesserung eines Klassifikationsergebnisses. Daher sind sie vor allem dann zu verwenden, wenn das multispektral klassifizierte Bild in Form einer thematischen Karte präsentiert werden soll. Dient das Klassifikationsergebnis jedoch nicht als Endprodukt, sondern wird es einer weiteren Verarbeitung unterworfen, empfiehlt es sich, diesen Schritt der Postklassifikation zu unterlassen, da im Zuge der Fehlerkorrektur eine Generalisierung erfolgt, die möglicherweise für Folgeprozesse relevante Information zerstört.

Landnutzung setzt sich aus einer Kombination verschiedener Oberflächenbedeckungskategorien zusammen. Durch die immer höher werdende räumliche Auflösung von Fernerkundungsdaten erfaßt ein einzelnes Pixel vor allem die Oberflächenbedeckungen der einzelnen Kategorien. Der zweite Teil dieses Abschnittes beschäftigt sich mit dem Problem, wie aus diesen Oberflächenkategorien über deren räumlichen Kontext übergeordnete Landnutzungsarten abgeleitet werden können.

Der letzte Teil stellt einen Vorstoß in den Bereich der Künstlichen Intelligenz dar. Der Wunsch automatisierte Klassifikationsprozesse auf das Niveau eines erfahrenen menschlichen Interpretors zu bringen, führt zum Einsatz wissensbasierter Methoden und Modelle. Dazu

kommt, daß immer mehr Datenquellen zur Verfügung stehen - verschiedenste Fernerkundungssensoren mit unterschiedlichen Charakteristika, Geographische Informationssysteme mit einer Vielzahl digitaler raumbezogener Daten -, die in den Klassifikationsprozeß einbezogen werden sollten. Es ist zu bezweifeln, ob klassische Methoden diese Datenkomplexe sinnvoll bewältigen können. Expertensysteme sind flexibel genug, Informationen verschiedenster Natur zu verknüpfen und über Regelsysteme zu klassifizieren. Allerdings ist der Aufwand sowohl im Aufbau der Wissensbasis als auch in der rechnerischen Umsetzung enorm. Dennoch zeichnet sich ein Trend in diese Richtung ab, wenngleich sich dieser noch auf wissenschaftliche Anwendungsbereiche beschränkt.

### **1.6.1 LOW LEVEL POSTKLASSIFIKATIONSALGORITHMEN**

Die einfachste Form der Überarbeitung klassifizierter Bilder geht davon aus, daß Klassifikationsergebnisse meist eine unruhige Ausprägung und viele isolierte Einzelpixel aufweisen, hervorgerufen durch Rauschen im Originalbild und Fehlklassifikationen. Wie bereits erwähnt, werden Nachbarschaftsbeziehungen bei einer pixelweisen Klassifikation nicht berücksichtigt. Man kann jedoch davon ausgehen, daß ein thematischer Zusammenhang zwischen benachbarten Pixeln besteht.

Um das klassifizierte Bild zu glätten und Einzelpixel zu eliminieren, kann ein *majorityfilter* verwendet werden. Dabei wird ein, meist quadratisches Fenster in Pixelschritten über das Bild bewegt. Dem mittleren Pixel dieses Fensters wird diejenige Klasse zugeordnet, die die Mehrheit innerhalb des Fensters aufweist. Über einen Schwellwert kann eine minimale Mehrheit definiert werden. Überschreitet keine der einzelnen Häufigkeiten diesen Schwellwert, bleibt das mittlere Pixel unverändert.

Die Wirkung des *majorityfilters* kann über die Fenstergröße und den Schwellwert variiert werden. Je größer das Fenster und je kleiner der Schwellwert, umso stärker wird das Bild geglättet. Das kann jedoch zur Elimination kleiner Strukturen führen, die erhalten werden sollen. Daher empfiehlt es sich, mit einer kleinen Fenstergröße (3x3 oder 5x5), aber dafür iterativ zu arbeiten. Der Schwellwert sollte größer als die halbe Anzahl der im Fenster auftretenden Pixel sein (z.B.  $\geq 5$  bei 3x3), d.h. es muß zumindest eine absolute Mehrheit erreicht werden, um eine Veränderung zu bewirken.

In einigen Softwarepaketen wird unter dem Bereich Postklassifikation auch ein Medianfilter angeboten. Ein solcher ist bei der Bearbeitung von Klassifikationsergebnissen mit Vorsicht einzusetzen, da er von nach ihrem Wert geordneten Zahlen ausgeht. Bei klassifizierten Bildern handelt es sich jedoch um über Zahlen kodierte thematische Klassen, die keiner

Reihung unterliegen, sondern gleichwertig nebeneinander stehen (z.B. ist eine Klasse mit der Kodierung 5 nicht höherwertig als eine mit der Kodierung 3). Zur Verdeutlichung sei ein Beispiel angeführt:

6	5	2
6	4	2
6	6	2

Die Reihung der Klassen ist (2, 2, 2, 4, 5, 6, 6, 6, 6), der Median beträgt 5 und wird dem mittleren Pixel zugeordnet, anstelle der Klasse 4. Daß diese Zuordnung im thematischen Kontext eine willkürliche ist, scheint offensichtlich.

Ein Algorithmus, der bereits an der Grenze zu kontextverarbeitenden Methoden steht, ist das *Small Class Merging* [Guo und Moore, 1991]. Dabei werden "kleine" Klassen spektral ähnlichen "größeren" Klassen zugeordnet. Unter "kleinen" Klassen versteht man in diesem Zusammenhang solche, die sowohl thematisch unbedeutend, als auch in ihrer räumlichen Ausprägung gering sind - kein flächenhaftes Auftreten, geringe Gesamthäufigkeit. Die Definition dieser "kleinen" Klassen geschieht interaktiv über Schwellwerte. Die Verarbeitung erfolgt wieder in einem Fenster, wobei dem mittleren Pixel, sofern es einer "kleinen" Klasse angehört, die spektral ähnlichste, "große" Klasse innerhalb des Fensters zugeordnet wird. Gibt es keine "große" Klasse im Fenster, wird die spektral ähnlichste aller "großen" Klassen zugeordnet. Dieser Algorithmus eignet sich sehr gut im Anschluß an einen *majorityfilter*.

### 1.6.2 RAUMBEZOGENE KONTEXTURALE OPERATOREN

Die nächste Stufe der Postklassifikation sind Methoden, die den Ausdruck Postklassifikation nicht nur im Sinne "nach der Klassifikation" verstehen, sondern selbst Klassifikationsverfahren darstellen, die auf dem Ergebnis einer pixelweisen Klassifikation aufsetzen. Die behandelte Problematik entsteht aus dem bereits am Beginn dieses Kapitels kurz angerissenen Widerspruch von Landbedeckung und Landnutzung (*landcover versus landuse*). Konnte man bei Satellitendaten, deren räumliche Auflösung im Bereich des Landsat MSS (80x80m<sup>2</sup>) liegen, noch davon ausgehen, daß ein Pixel eine bestimmte grobe Landnutzung repräsentiert - z.B. Wald, Stadt oder landwirtschaftliche Nutzfläche -, so wird die direkte Landnutzungs-zuordnung mit zunehmender räumlicher Auflösung immer schwieriger. Die in ihrer Bedeutung (Thematik) homogenen Landnutzungsklassen (z.B. Park, dicht/mittel/locker verbaute

Gebiete, Industrieflächen) werden durch die höhere Auflösung der heutigen Sensoren in ihre spektral verschiedenen Komponenten zerlegt. Daher ist schwer es bis unmöglich, direkt aus der am Pixel ansetzenden multispektralen Klassifikation Landnutzungsklassen zu identifizieren (vgl. Zhang et al., 1988). Will man zum Beispiel verschiedene Verbauungsdichten in urbanen Räumen differenzieren, wird die multispektrale Klassifikation nur Oberflächenkomponenten wie Ziegeldächer, Asphalt- oder Betonflächen, Grünland etc. erkennen. Diese Komponenten kommen jedoch in allen Verbauungsklassen vor, wobei sie sich nur durch ihre prozentuellen Anteile unterscheiden. Daher ist es notwendig, neben der spektralen Information auch deren räumliche Verteilung zu analysieren, und daraus die gesuchten Landnutzungsklassen zu ermitteln. In der Folge werden die spektral ermittelten Oberflächenbedeckungen als primäre Klassen, die daraus gewonnenen Landnutzungsklassen als sekundäre Klassen, das aus einer pixelweisen Klassifikation hervorgegangene Bild als Ausgangsbild und das dem Postklassifikationsalgorithmus unterworfenene Bild als Ergebnisbild bezeichnet.

## Räumliche Frequenz

*Template Histogram Matching Classification* (THMC) ist eine einfache, überwachte Klassifikationsmethode, die bestimmte Muster von primären Klassen identifiziert und sie sekundären Klassen zuordnet [Guo und Moore, 1991]. Die relevanten Muster im Ausgangsbild werden über Testgebiete definiert. Das Ergebnis des Trainingsprozesses sind normierte Histogramme ( $H_k$ ), die die Häufigkeit der in den  $k$  Testgebieten auftretenden  $n$  primären Klassen repräsentieren. Die Klassifikation erfolgt über ein Fenster, das in Pixelschritten über das Ausgangsbild bewegt wird. In jeder Position des Fensters ( $i,j$ ) wird wiederum das normierte Histogramm ( $h_{i,j}$ ) bestimmt und über eine Distanzfunktion mit den trainierten Histogrammen ( $H_k$ ) verglichen. Die Distanz ( $D_{i,j}$ ) errechnet sich dabei nach folgender Formel:

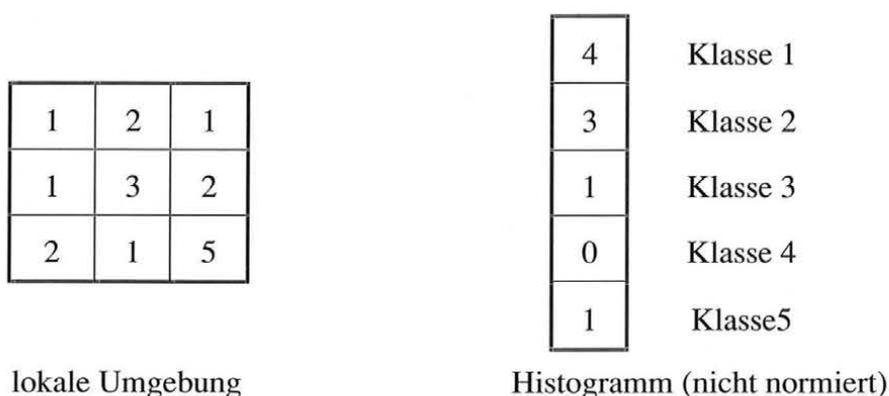
$$D_{i,j} = \sum_{l=1}^n |H_k(l) - h_{i,j}(l)| \quad (1.9)$$

Das mittlere Pixel wird jener sekundären Klasse zugeordnet, deren Distanz minimal ist. Um untrainierte Muster einer Fehlklassifikation zu entziehen, wird zusätzlich ein Schwellwert eingeführt, der ein Distanzkriterium darstellt. Sind alle der berechneten Distanzen größer als der Schwellwert, bleibt das Mittelpixel unverändert.

Die Größe des bewegten Fensters ist ein entscheidender Parameter, da bei zunehmender Ausdehnung desselben immer langwelligere Muster erkannt werden können. Zur Unterscheidung verschieden frequenter, aber ähnlicher Muster, kann dieses Verfahren auch mehrfach mit verschiedenen Fenstergrößen eingesetzt werden. Die einzelnen Ergebnisbilder müssen dann

allerdings interaktiv verschnitten werden, um das gesuchte Endresultat zu erhalten. Ein interessantes Fallbeispiel zu diesem Thema unter Verwendung multispektraler SPOT-Daten findet sich in [vgl. Gong und Howarth, 1992].

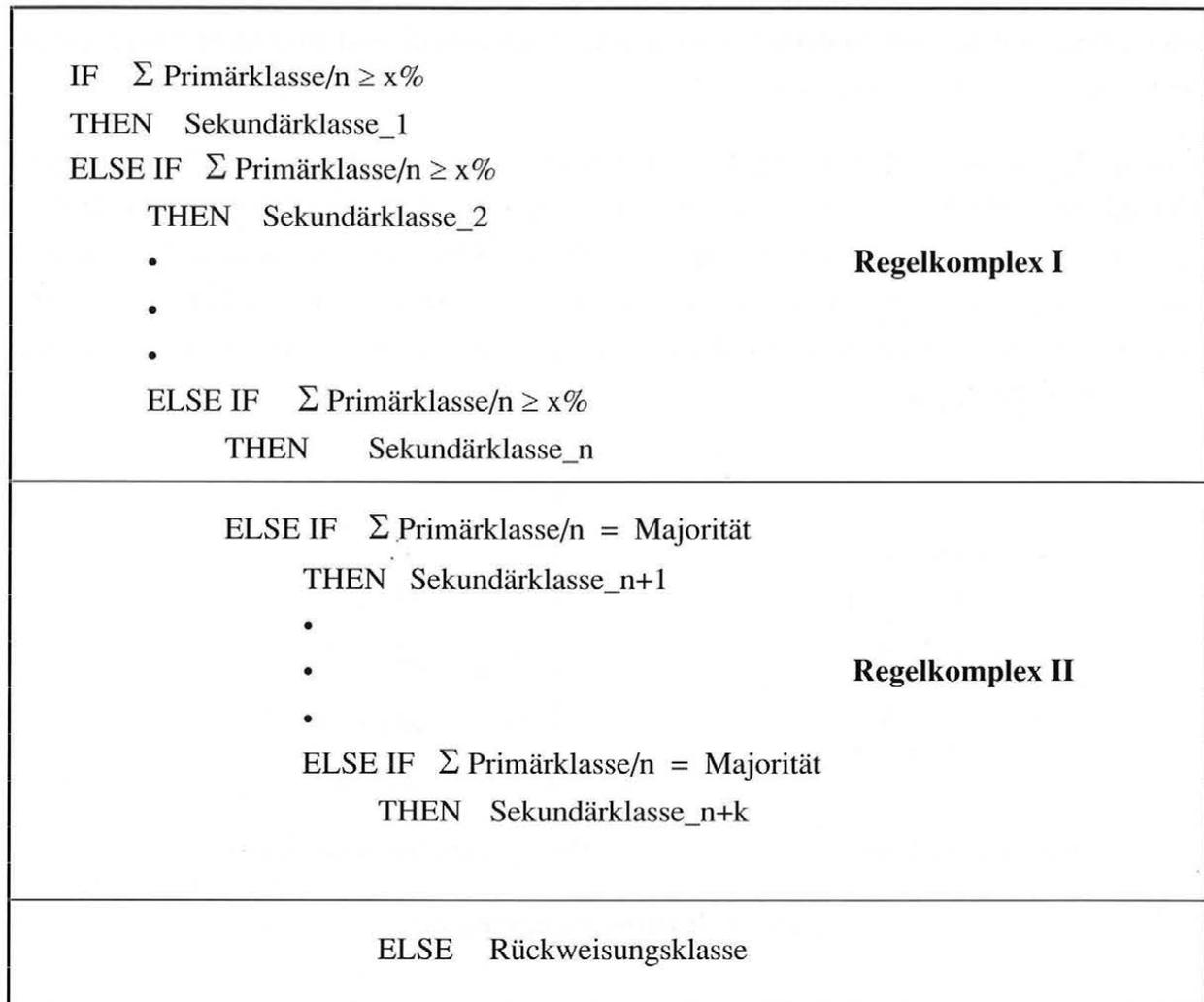
Eine in den Ansätzen ähnliche Methode der Postklassifikation, die *Spatial Context Rules Classification* (SCRC), wurde am I.P.F. entwickelt [vgl. Bühler und Steinnocher, 1991]. Auch hier wird die räumliche Verteilung der primären Klassen durch ein normiertes Histogramm beschrieben (siehe Abb. 1.4). Im Gegensatz zur THMC erfolgt die Klassifikation jedoch nicht über Vergleiche mit trainierten Testgebieten, sondern über ein hierarchisches zweigeteiltes Regelsystem.



**Abb. 1.4: Histogrammbildung**

Die Regeln beziehen sich auf die prozentuellen Anteile der primären Klassen innerhalb des Fensters. Ist die Häufigkeit einer oder mehrerer Klassen im Histogramm größer oder gleich dem vom Benutzer definierten Wert der Regel, so wird diese Regel "gefeuert" und die dazugehörige sekundäre Klasse zugewiesen. Ist dies nicht der Fall, so wird die nächste Regel angewandt (siehe Abb. 1.5). Wird keine der Regeln als "wahr" erkannt, kommt der zweite Regelkomplex zum Einsatz. Dieser untersucht, ob gewisse primäre Klassenkombinationen eine Majorität im Histogramm aufweisen und weisen dementsprechend sekundäre Klassen zu. Kommt auch hier keine der Regeln zum Tragen, weist das System eine Rückweisungsklasse zu, die im Anschluß an die automatische Klassifikation interaktiv überarbeitet werden kann.

Die Regeln sind heuristisch, d.h. sie repräsentieren das Wissen und die Erfahrung des Bearbeiters und können daher verschiedenen Bilddaten und Aufgabenstellungen angepaßt werden. Die Erstellung erfolgt interaktiv über die Eingabe der klassenrelevanten Grenzwerte. Die

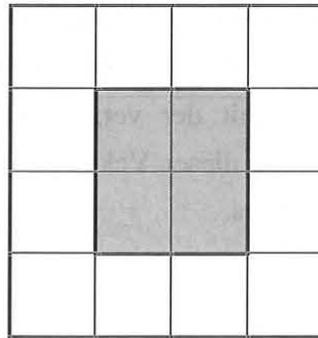


**Abb. 1.5: Regelsystem**

Größe der untersuchten Umgebung ist ein entscheidender Parameter der Klassifikation. Wird das Fenster zu klein gewählt, werden nur kleinste Strukturen erkannt und das Bild behält seinen unruhigen Charakter. Größere Fenster ermöglichen das Erkennen niedrigfrequenter Strukturen und bewirken eine stärkere Generalisierung. Das Maß der Generalisierung kann auch über die Pixelgröße des postklassifizierten Bildes gesteuert werden. So kann z.B. eine 4x4 Umgebung auf die mittleren 2x2 Pixel abgebildet werden (siehe Abb. 1.6).

Die Anwendung der SCRC erwies sich am Beginn als iterativer Lernprozeß, da erst ein Gefühl für den Effekt der Regeln entwickelt werden mußte. Dank der eingesetzten parallelen Hardware und der Tatsache, daß die Abarbeitung des Regelkomplexes auf arithmetischen Operationen, konkret auf Matrizenmultiplikationen, basiert, war der zeitliche Aufwand gering. Nach einigen Testläufen mit verschiedenen Datensätzen wurde der Algorithmus auf ein

konkretes Projekt angewandt [Steinnocher et al., 1993]. Der dritte Abschnitt der vorliegenden Arbeit (Kap. 3) beschreibt diese Fallstudie am Beispiel des Agglomerationsraumes Wien.



**Abb. 1.6: 4x4 Pixel Umgebung abgebildet auf 2x2 Pixel**

### Räumliche Anordnung

Die beiden zuvor beschriebenen Postklassifikationsmethoden analysieren zwar die Häufigkeit der auftretenden primären Klassen innerhalb einer räumlichen Umgebung, berücksichtigen allerdings nicht deren räumliche Anordnung. Zur Unterscheidung gewisser sekundärer Klassen - z.B. verschiedene Bebauungsstrukturen - ist diese Information jedoch von entscheidender Bedeutung. Anhand eines Beispiels wird die Problematik verdeutlicht.

Bild A		
W	G	G
B	G	G
W	B	W

Bild B		
G	W	W
B	G	B
G	W	G

Legende: W - Wiese, B - Bäume, G - Gebäude

**Abb. 1.7: Bildausschnitte mit verschiedener Bebauung**  
(entnommen aus *Barnsley und Barr* [1992])

Die beiden Ausschnitte eines klassifizierten Bildes (Abb. 1.7) weisen die gleiche Häufigkeit von primären Klassen auf (G:W:B = 4:3:2), stellen jedoch für den menschlichen Betrachter eindeutig verschiedene Bebauungsstrukturen dar. Das Bild A könnte eine Blockbauweise repräsentieren, wie sie in mittleren Stadtgebieten oder Industriezonen vorkommt; das Bild B stellt eine lockere Verbauung dar, z.B. ein Villenviertel oder ländlichen Siedlungsraum. Mit den zuvor beschriebenen Algorithmen können die beiden Beispiele nicht unterschieden werden. *Barnsley und Barr* [1992] haben daher einen **Adjazenz-Vektor** (*adjacency vector*) eingeführt, der die Häufigkeiten enthält, mit der verschiedene Klassen innerhalb eines Fensters aneinandergrenzen. Die Berechnung dieser Vektoren für das Beispiel aus Abb. 1.7 führt zu dem in Tab. 1.2 dargestellten Ergebnis.

**Tab. 1.2: Adjazenz-Vektoren der Bilder aus Abb. 1.7**

Nachbarschaftstyp	Bild A	Bild B
Gebäude - Gebäude	6	3
Gebäude - Wiese	5	6
Gebäude - Bäume	4	5
Wiese - Wiese	0	1
Wiese - Bäume	4	5
Bäume - Bäume	1	0

Die unterschiedliche Anordnung der primären Klassen drückt sich deutlich in den beiden Vektoren aus. Ist bei Bild A die Nachbarschaft Gebäude - Gebäude vorherrschend, so dominieren bei Bild B die Nachbarschaften Gebäude - Wiese, Gebäude - Bäume und Wiese - Bäume.

Die Klassifikation erfolgt wie bei der THMC. Der Adjazenz-Vektor des Fensters wird über eine Distanzfunktion mit den Vektoren der Trainingsgebiete verglichen. Das Mittelpixel wird der sekundären Klasse zugeordnet, die die minimale Distanz aufweist, sofern diese kleiner als ein definierter Schwellwert ist (vgl. Kapitel 1.6.2.1, *Template Histogram Matching Classification*).

### 1.6.3 EXTERNE DATEN

In vielen Bereichen der Erde, besonders in den Ländern der ersten Welt, existiert bereits eine Vielzahl raumbezogener Information. Der Einsatz der Fernerkundung in diesen Gebieten zielt vor allem auf eine laufende Erfassung der Veränderung der Bedeckung und Nutzung der Erdoberfläche (*change detection*). Dabei kann die bestehende Information wertvolle Dienste leisten. Konkret bedeutet das, raumbezogene Daten, wie sie in Geographischen Informationssystemen (GIS) zur Verfügung stehen, in den Klassifikationsprozeß einfließen zu lassen.

Die bisher beschriebenen Methoden versuchen die sekundären Klassen auf Grund der Verteilung der primären Klassen in einer räumlichen Umgebung zu identifizieren. Der konträre Ansatz geht vom Objekt aus. Ein Objekt ist eine Fläche, innerhalb der nur eine Klasse auftritt. Die Geometrie der Objekte wird entweder aus dem Ausgangsbild abgeleitet [Johnsson und Kanonier, 1991] oder aus einem GIS gewonnen [Janssen et al., 1990, Janssen und van Amsterdam, 1991]. Voraussetzung für die zweite Methode ist, daß die Mehrheit der Pixel innerhalb eines Objektes richtig klassifiziert ist. Die Klassifikation selbst ist simpel: Die Häufigkeiten der primären Pixel innerhalb eines Objektes werden erfaßt; ist eine Klasse vorherrschend, wird sie dem Objekt zugeordnet; sind mehrere Klassen dominant, braucht man entweder eine feinere Objektaufteilung der primären Klassen [vgl. Janssen et al., 1990], oder verwendet Kontextregeln zur Zuweisung sekundärer Klassen [vgl. Kap. 1.6.2.1].

Die eingebundenen geographischen Daten beschränken sich natürlich nicht auf die Geometrie, sondern können auch topographischer (Geländehöhe, -neigung und -ausrichtung) oder thematischer Natur (Bodentypen, vorherrschende Vegetationsarten etc.) sein [vgl. Kontoes et al., 1991]. Die Verarbeitung dieser Information zusammen mit den aus den Fernerkundungsbildern abgeleitenden Daten erfolgt über regelbasierte Systeme, wie sie im folgenden Kapitel vorgestellt werden.

### 1.6.4 WISSENSBASIERTE EXPERTENSYSTEME

Der Bereich wissensbasierte Expertensysteme (*knowledge-based expert systems*) geht weit über den Rahmen der vorliegenden Arbeit hinaus. Da er aber eine logische Konsequenz der hier vorgestellten und angewandten Methoden darstellt, wird er im folgenden an Hand aktueller Beispiele aus der Literatur grob umrissen.

Die letzten Kapitel haben deutlich gezeigt, daß es sinnvoll ist, spektrale, texturale und kontexturale Information aus Bildern zu extrahieren und in den Klassifikationsprozeß einfließen zu lassen. Auch die Einbindung geographischer Daten aus einem GIS erhöht die Zuverlässig-

keit der Ergebnisse. An dieser Stelle seien noch die verschiedenen Arten von Kontext angesprochen, die in der Literatur nicht ganz einheitlich verwendet werden. Unter räumlichem Kontext (*spatial context*) versteht man die Berücksichtigung lokaler Nachbarschaften, wie sie in Kap. 1.6.2 besprochen wurden. Dabei wird nur vom klassifizierten Bild ausgegangen. Geographischer Kontext (*geographic context*) bezieht sich dagegen auf Information, die aus externen Daten gewonnen wird. Dazu zählen sowohl statische Parameter, wie Bodentypen oder Geländehöhe, -neigung und -ausrichtung, als auch dynamische Parameter, wie Erreichbarkeit über das Straßennetz oder Distanz zum Stadtzentrum.

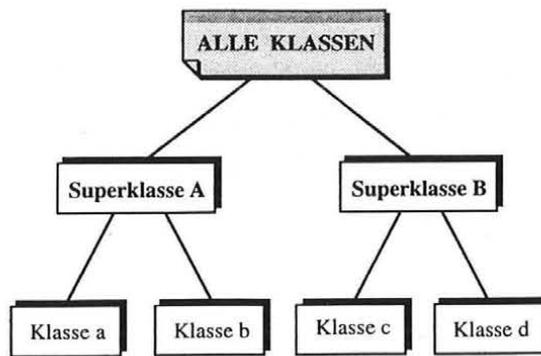
Diese Methoden sind jedoch bis jetzt nur als alleinstehende Ansätze angewandt worden. Eine Verknüpfung aller verfügbaren Informationsebenen stellt ein komplexes Problem dar, das nicht mehr mit mathematischen oder statistischen Methoden gelöst werden kann. Es ist vielmehr notwendig, heuristische Modelle zum Einsatz zu bringen, wie sie in wissensbasierten Expertensystemen verwendet werden [Argialas und Harlow, 1990].

Expertensysteme verwenden Wissen, das auf menschlicher Erfahrung und nicht auf statistischen Modellen beruht, zur Lösung von Problemen. Dabei wird das Wissen in eine Wissensbasis und einen Kontrollmechanismus zerlegt. Die Wissensbasis wird in Form von unabhängigen Fakten und Regeln definiert, der Kontrollmechanismus gibt an, wie die Fakten und Regeln eingesetzt werden [Srinivasan und Richards, 1990].

Regelbasierte Systeme können auf Gewißheit - *evidential approach* - oder Hierarchien - *hierarchical approach* - beruhen. Der *evidential approach* stützt sich auf ein relatives Maß der Evidenz von Hypothesen, also einer Art "quantifiziertem Wissen", wobei nach Abarbeitung der Regeln diejenige Hypothese gewählt wird, die das höchste Maß der Gewißheit akkumuliert hat. Beim *hierarchical approach* wird ein Regelbaum durchlaufen, wobei sukzessive die Regeln abgearbeitet und Hypothesen eliminiert werden, bis nur mehr eine Hypothese übrig bleibt [Bolstad und Lillesand, 1992].

Der Unterschied zwischen der hierarchischen und der *evidential* Klassifikationsmethode soll anhand eines Beispiels demonstriert werden. Die gestellten Hypothesen werden durch die Klassen repräsentiert, die mögliche Ergebnisse der Klassifikation darstellen (Abb. 1.8).

In einem hierarchischen Modell durchläuft das zu klassifizierende Objekt - z.B. ein Pixel - die Baumstruktur. An jedem Knoten wird über Regeln entschieden, welcher Weg zu wählen ist. Wird an einem Knoten eine Fehlentscheidung getroffen, so ist diese nicht mehr rückgängig zu machen. Erreicht der Durchlauf ein Blatt des Baumes, wird die dort auftretende Klasse als Ergebnis zugewiesen. Es ist klar ersichtlich, daß nur die Klassen a, b, c, d als Ergebnis in Frage kommen.



**Abb. 1.8: Klassifikationsmodell**

Evidenzansätze verwenden ihre Regeln außerhalb der Baumstruktur. Die Regeln entscheiden nicht über Verzweigungen, sondern vergeben Evidenzwerte an die möglichen Klassen. Das zu klassifizierende Objekt durchläuft das gesamte Regelsystem. Die Regeln beurteilen die Merkmale des Objektes in Bezug auf die einzelnen Klassen. Trifft eine Regel zu, wird der bestehende Evidenzwert einer Klasse mit dem der Regel verknüpft und es entsteht ein neuer Evidenzwert. Sind alle Regeln abgearbeitet, weist jede Klasse (oder Superklasse) einen akkumulierten Evidenzwert auf. Das Pixel wird daraufhin jener Klasse zugeordnet, die den höchsten Evidenzwert hat. Es ist durchaus möglich, daß im Falle einer Unsicherheit zwischen zwei Unterklassen die übergeordnete Klasse einen höheren Wert aufweist als die beiden Subklassen. Das deutet darauf hin, daß nicht genügend Information für eine feinere Klassifikation vorhanden ist und dementsprechend sollte das Objekt der Überklasse zugeordnet werden. Das bedeutet, daß alle Klassen (a, b, c, d und A, B) als Ergebnis in Frage kommen.

Einen numerischen Ansatz für den Umgang mit Evidenzen bietet die *Dempster-Shafer theory of evidence*. Dabei werden den Hypothesen sogenannte *basic probability assignments* (b.p.a.) zugewiesen, die über eigene Verknüpfungsregeln kombiniert werden können. Eine nähere Beschreibung dieser Theorie und einen Vergleich mit einem nicht-numerischen, qualitativen Ansatz geben *Srinivasan und Richards* [1990].

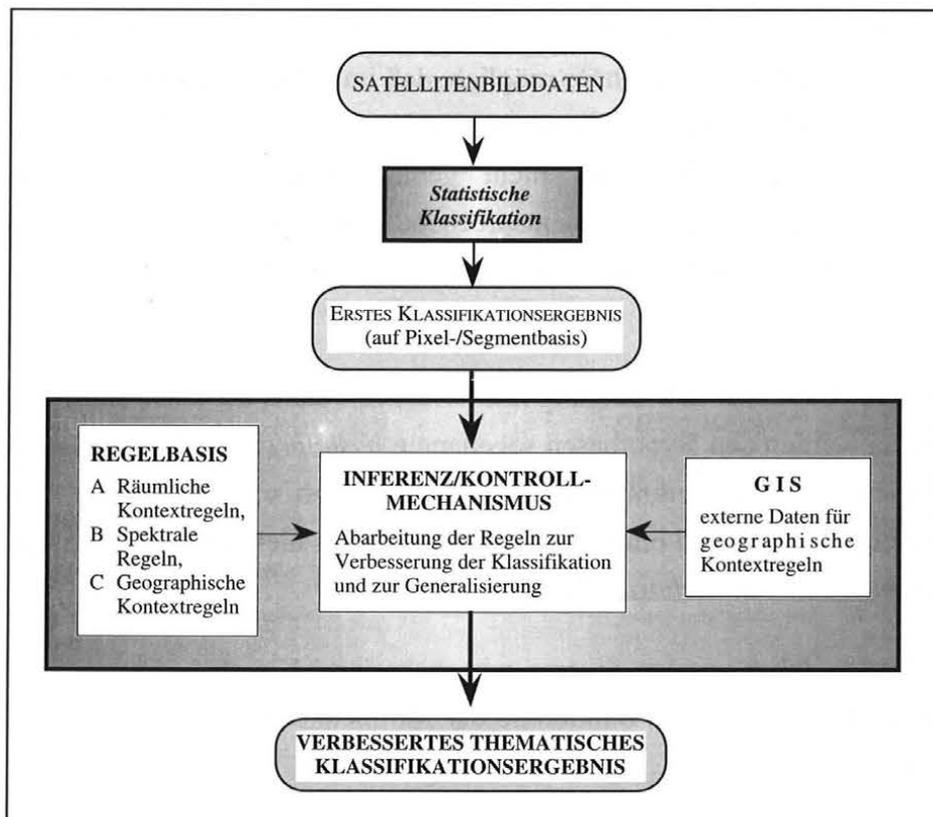
Im Laufe der letzten Jahre wurden diverse wissensbasierte Klassifikationsansätze getestet. Eine Auswahl daraus, die auch die Bandbreite der Möglichkeiten demonstrieren soll, sei im folgenden gegeben.

Ein Beispiel für den hierarchischen Ansatz findet man in *Johnsson und Kanonier* [1991]. Ausgehend vom Ergebnis einer multispektralen Klassifikation werden einzelne Objekte gebildet, abgeleitet aus den Segmenten des Ausgangsbildes (vgl. Kap. 1.6.3). Diese Objekte

werden beschrieben durch eine fortlaufende Nummer, einen thematischen Wert (entspricht der primären Klasse), durch ihre Größe, ihren Umfang und ihre Form. Zusätzlich wird der räumliche Kontext der Objekte erfaßt. Diese Informationen werden in einem Expertensystem über hierarchische Regeln klassifiziert.

*Mehldau und Schowengerdt* [1990] gehen von segmentierten Objekten in einem digitalisierten Luftbild aus. Diese Objekte werden in drei Phasen postklassifiziert - über spektrale, geometrische und räumliche Kontext-Regeln. In jeder der Phasen vergeben die Regeln Werte an mögliche Klassen. Die Klasse mit dem höchsten akkumulierten Wert wird dem Objekt zugewiesen.

*Møller-Jensen* [1990] beginnt mit einer pixelweisen Klassifikation einer TM-Szene von Bangkok. Dazu werden aus dem Bild linienhafte Elemente extrahiert (Straßen, Flußläufe etc.), die einzelne Bildsegmente begrenzen. Für jedes Segment werden spektrale, texturale und räumlich kontexturale Merkmale berechnet, über die mit heuristischen Regeln die Postklassifikation durchgeführt wird.



**Abb. 1.9: Schema eines Expertensystems**  
(nach *Wilkinson und Burrill* [1991])

Eine Verbesserung der Genauigkeit einer multispektralen Klassifikation durch Einbeziehung externer Daten aus einem GIS findet man in *Wang und Civco* [1991]. Die Postklassifikation beschränkt sich auf möglicherweise fehlklassifizierte Pixel (geringe Wahrscheinlichkeit bei der Maximum Likelihood Klassifikation). Diese Pixel werden über Informationen aus Geländehöhe, Neigung und Ausrichtung, einem Verkehrs- und einem Gewässernetz korrigiert.

Einen umfassenden Ansatz versuchen *Wilkinson und Burrill* [1991]. Ausgehend von einer Pixelklassifikation werden über räumliche Kontext-Regeln, spektrale Regeln und geographische Kontext-Regeln aus dem Bild abgeleitete und externe Informationen verarbeitet [vgl. auch *Kontoes et al.*, 1991]. Als externe Daten werden Geländehöhe, Bodentyp und Zugänglichkeit zu Verkehrswegen eingeführt. Die verwendete Methode basiert auf der Dempster Shafer Theorie und arbeitet auf einer Pixelbasis (vgl. Abb. 1.9).

## **1.7 INTEGRATION VON FERNERKUNDUNG UND GIS**

War die Fernerkundung lange Zeit ein Instrument zur Erstellung von Satellitenbildkarten oder themakartographischen Produkten, so ist heute der Trend zur Integration mit GIS unübersehbar. Die Grenze zwischen Bildverarbeitung, als Instrument der Fernerkundung, und geographischer Informationsverarbeitung ist dabei nicht eindeutig definiert. Gehören Klassifikation und Texturanalyse noch zum Bereich der Bildverarbeitung, so stellt sich bei den in diesem Abschnitt beschriebenen Algorithmen der Postklassifikation bereits die Frage, ob sie nicht der geographischen Informationsverarbeitung zuzuordnen sind. Die Einbeziehung externer Daten aus einem GIS und der Einsatz von Expertensystemen zeigen deutlich die Notwendigkeit und die Vorteile einer Verbindung der beiden Fachbereiche. Der status quo der Integration befindet sich heute auf der Ebene lose gekoppelter Systeme. Es wird die Aufgabe der nächsten Jahre sein, geeignete Datenmodelle und Verarbeitungsmethoden zu entwickeln, um die noch bestehende Kluft zwischen Fernerkundung und GIS zu überbrücken und ganzheitliche Lösungen anzubieten [vgl. z.B. *Ehlers et al.*, 1991, *Faust et al.*, 1991, *Ryerson*, 1989].

## 2. BILDVERARBEITUNG MIT TRANSPUTERN

### 2.1 EINLEITUNG

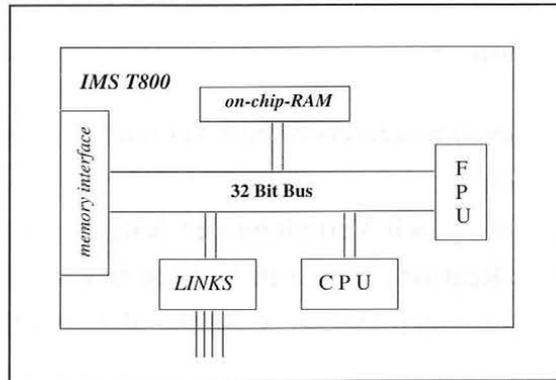
Die Verarbeitung von Satellitenbildern erfordert hohe Rechnerleistungen und geeignete Visualisierungsmöglichkeiten. Die erste Forderung ergibt sich aus den überdurchschnittlich hohen Datenmengen und den zum Teil rechenintensiven Algorithmen. Der Wunsch nach qualitativ hoher Visualisierung folgt der Notwendigkeit der Kontrolle und Bewertung der eingesetzten Algorithmen und der klaren Darstellung der erzeugten Resultate. Um diesen Anforderungen Rechnung zu tragen, wurde am I.P.F. das System 2400 der Wiener Firma IMPULS erworben.

Bei diesem System handelt es sich um einen Parallelrechner, der auf Transputermodulen der britischen Firma INMOS basiert. Der Rechner wird über einen PC als Host gesteuert und ist mit einem hochauflösenden Echtfarbenschirm ausgestattet. Die oben genannten Forderungen nach hoher Rechnerleistung und guter Visualisierungsmöglichkeit werden über die Parallelität der Prozessoren und die hochauflösende Grafik gewährleistet. Eine detaillierte Beschreibung der Hardwarekomponenten findet sich in Kap. 2.2. Da für die spezielle Konfiguration keine Software zur Verfügung stand, bestand die eigentliche Arbeit in der Entwicklung und Implementierung allgemeiner Bildverarbeitungsprogramme und spezieller Algorithmen, wie sie im letzten Abschnitt beschrieben wurden. Kap. 2.3 beschäftigt sich eingehend mit dieser Problematik und mit der Frage, warum Parallelität gerade in der Bildverarbeitung effizient eingesetzt werden kann.

### 2.2 HARDWARE

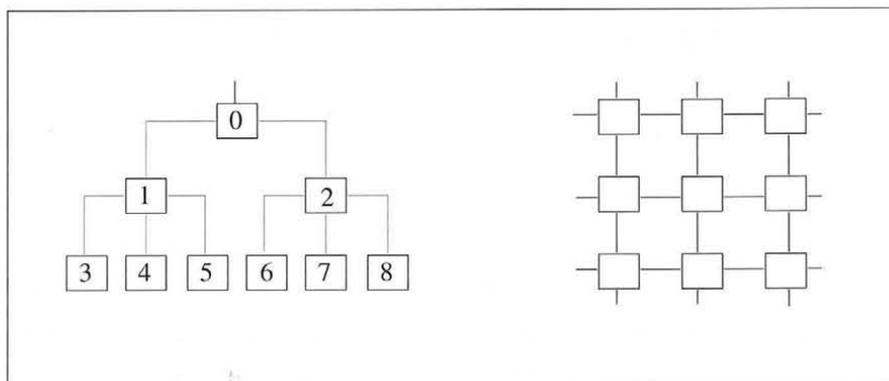
#### 2.2.1 TRANSPUTER

Die INMOS-Transputer bilden eine Familie von 16- und 32-Bit Mikroprozessoren, die Verarbeitungs-, Speicher- und Kommunikationseinheiten auf einem einzelnen *VLSI-chip* (*very large scale integration-chip*) vereinen. Sie bestehen aus einer *central processing unit* (CPU), vier *links* (Kommunikationsschnittstellen), einem *memory interface* und *on-chip-RAM*, die über einen internen Bus miteinander verbunden sind. Die letzte Entwicklung, der IMS T800 (Abb. 2.1), beinhaltet darüber hinaus eine 64-Bit *floating point unit* (FPU) und ein erweitertes *memory interface*, das die Adressierung von bis zu 4 GByte *memory* ermöglicht [INMOS, 1987].



**Abb. 2.1: Architektur des IMS T800**

Transputer können als Einzelprozessoren verwendet werden, ihre besondere Stärke liegt jedoch in der Vernetzung mit anderen Transputern. Zur diesem Zweck besitzt jeder Transputer 4 serielle Punkt-zu-Punkt-Verbindungen, sogenannte *links*, über die er mit anderen Transputern kommunizieren kann. Dank dieser *links* können Transputer zu Netzwerken konfiguriert werden, ohne zusätzliche Elemente zu benötigen (Abb. 2.2).



**Abb. 2.2: Netzwerktopologien**

Da Transputer als Multitasking-Maschinen ausgeführt sind, können parallele Prozesse auch auf einem Transputer verarbeitet werden. Dieses Konzept ermöglicht auch die verlustfreie Kommunikation über die *links*, die parallel zur Tätigkeit der CPU stattfindet. Das bedeutet, daß ein Transputer gleichzeitig über alle *links* Daten übertragen kann, während die CPU ungestört weiterrechnet.

## 2.2.2 IMPULS SYSTEM 2400

Bevor das System 2400 im Detail beschrieben wird, sei noch kurz auf Parallelrechner im allgemeinen eingegangen. Gemeint sind damit Computer, die mehrere parallel arbeitende CPUs oder Rechenwerke besitzen und durch Verteilung der Aufgaben höhere Leistungen erzielen als sequentielle Rechner. Die Realisierung der Parallelität in Form von Rechnerarchitekturen erfolgt nach verschiedenen Konzepten [Meuer & Strohmaier, 1992]. Die Klassifikation nach Flynn unterscheidet dabei nach parallel ausgeführten Instruktionen und parallel verarbeiteten Daten (vgl. Tab. 2.1).

**Tab. 2.1: Klassifikation von Parallelrechnern nach Flynn**

	<i>DATA</i>	
<i>INSTRUCTION</i>	<i>Single</i>	<i>Multiple</i>
<i>Single</i>	SISD von Neumann	SIMD Vektor-, Arrayprozessoren
<i>Multiple</i>	MISD -----	MIMD eigentliche Parallelrechner

Diese Klassifikation besteht zwar in ihrer Einfachheit, weist jedoch gravierende Nachteile auf, da die Aufteilung in die einzelnen Klassen sehr ungleich gewichtet ist. Die Klasse SISD (*single instruction / single data*) enthält nur die klassischen von Neumann Rechner, der Klasse MISD (*multiple instruction / single data*) kann überhaupt keine Architektur zugeordnet werden. Die Gruppe der SIMD Rechner (*single instruction / multiple data*) beinhaltet Rechnertypen, deren Rechenwerke synchron dieselbe Instruktion auf parallelen Datenströmen ausführen. Die meisten Parallelrechner fallen jedoch in die Gruppe der MIMD Systeme (*multiple data / multiple instruction*), die verschiedene Instruktionen auf verschiedene Daten anwenden können. Diese Architektur erlaubt es, die Parallelität von Prozessen und deren Wechselwirkung zu realisieren. Da die Prozesse (*tasks*) asynchron arbeiten, muß zur Realisierung der Wechselwirkungen an einzelnen Zeitpunkten eine Synchronisation erfolgen. Diese Synchronisation wird entweder über einen gemeinsamen Speicher oder über direkten Datenaustausch zwischen den *tasks* ermöglicht. Damit ergibt sich eine weitere Unterteilung der MIMD Rechner in *shared memory* oder *distributed memory* Modelle. Bei *shared memory*

Architekturen erfolgt der Datenaustausch zwischen den *tasks* über einen gemeinsamen Speicher, wobei von allen *tasks* auf alle Daten zugegriffen werden kann. Allerdings muß dieser Speicherzugriff synchronisiert werden und stellt einen Flaschenhals dar. *Distributed memory* Architekturen dagegen verwenden nur lokale Speicher; die Kommunikation zwischen den *tasks* wird über ein Verbindungsnetzwerk ermöglicht.

Parallelrechner, die auf Transputernetzwerken basieren, so auch das System 2400, gehören zur Gruppe der MIMD Rechner mit verteiltem Speicher. Der Datenaustausch und die damit verbundene Synchronisation der *tasks* erfolgt über die Transputerlinks. Abb. 2.3 gibt einen Überblick über die Hardwarekonfiguration. Die Charakteristika der einzelnen Komponenten werden in den folgenden Kapiteln beschrieben.

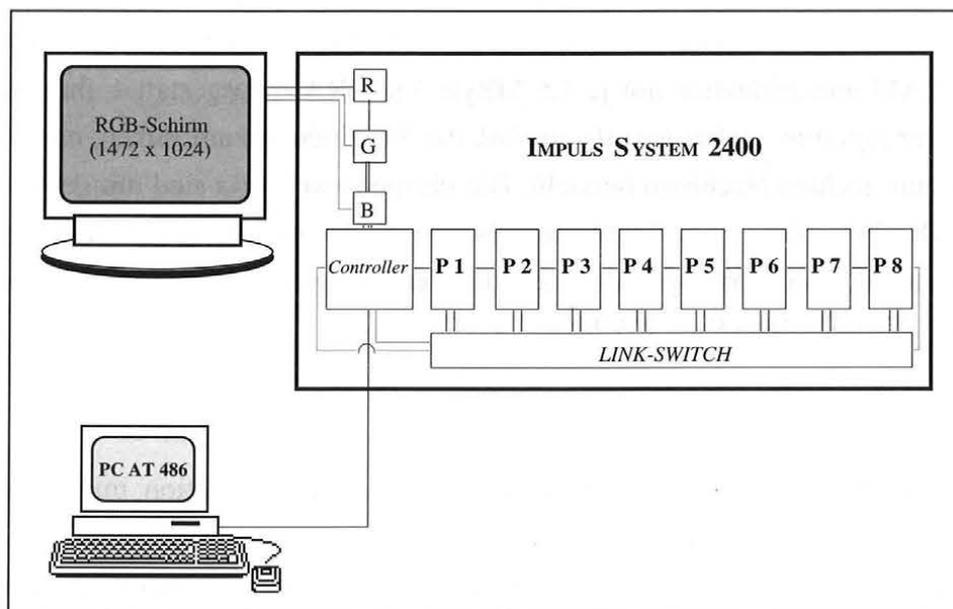


Abb. 2.3: Hardwarekonfiguration IMPULS System 2400

### Host/Root

Transputer Netzwerke benötigen zur Kommunikation mit der Außenwelt einen Hostrechner. Das System 2400 verwendet dafür einen PC, im konkreten Fall handelt es sich um einen AT mit einem Intel 80486 Prozessor, einer 120 MByte Harddisk und einer 640 MByte Harddisk, die über eine SCSI Schnittstelle angesprochen wird. Die Verbindung zum Netzwerk bildet eine Einschubkarte, die einen T800 mit 2 MByte RAM beinhaltet (IMS B008 *evaluation*

*board*) [INMOS, 1988a]. Dieser Transputer wird in der Folge als Root bezeichnet. Ein *link* des Root ist über einen *linkadapter* mit dem PC internen Bussystem verbunden. Ein weiterer *link* ist mit dem ersten Transputer im System verbunden, dem sogenannten Controller.

### **Controller**

Eine spezielle Funktion innerhalb des Systems hat der Controller. Auch hier handelt es sich um einen T800 mit 2 MByte RAM. Seine Aufgaben bestehen in der Kontrolle des *displays* inklusive der *look-up-tables*, in der Konfiguration des Linkswitch (siehe unten) und in der Verwaltung des Datenaustausches mit dem Root bzw. Hostrechner. Zwei *links* dienen als Verbindung zum Root bzw. zur Working Unit, die übrigen zwei sind mit dem Linkswitch verbunden. Davon dient jedoch nur einer der Datenübertragung, der andere steuert die Konfiguration des Linkswitch.

### **Working Unit**

Als Working Unit (WU) werden die restlichen acht T800 bezeichnet. Sie sind wiederum mit je 2 MByte RAM und zusätzlich mit je 1.5 MByte VideoRAM ausgestattet. Ihre Anordnung entspricht einer *pipeline*, wobei jeweils ein *link* die Verbindung zum linken, ein zweiter die Verbindung zum rechten Nachbarn herstellt. Die übrigen zwei *links* sind mit dem *Linkswitch* verbunden. Die WU dient, wie ihr Name schon sagt, der eigentlichen Datenmanipulation, wobei die Steuerung der Arbeitsprozesse und die Verteilung der Daten vom *Controller* überwacht werden (Details siehe Kap. 2.3.2).

### **Linkswitch**

Der programmierbare Linkswitch IMS C004 kann in seiner Funktion mit einem Kreuzschienenverteiler verglichen werden. Er hat 32 *link*-Ein/Ausgänge, die beliebig miteinander verbunden werden können, unter der Bedingung, daß die Verbindungen eindeutig sind [Hill, 1987]. Die Steuerung erfolgt über einen Konfigurations*link*, der beim System 2400 vom Controller angesprochen wird. Da jeder Transputer der WU über zwei *links* mit dem Linkswitch gekoppelt ist, kann eine Vielzahl von Netztopologien kreiert werden (Abb. 2.4). Die Konfiguration des Linkswitch erfolgt softwaremäßig und kann während der *run time* geändert werden. Diese Eigenschaft erlaubt auch denjenigen Transputerpaaren die direkte Kommunikation, die keine direkte *link*-Verbindung miteinander haben (z.B. P1 und P4 der WU, die ohne Linkswitch nur über P2 und P3 kommunizieren könnten).

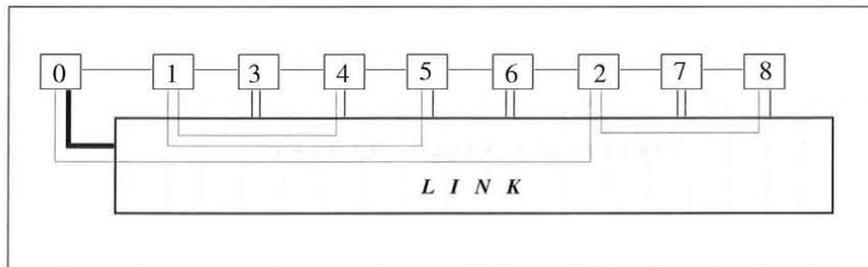


Abb. 2.4: Linkswitchkonfiguration für Baumstruktur (vgl. Abb. 2.2, links)

## Display

Jeder der acht Transputer der WU verfügt über 1.5 MByte VideoRAM. Daraus ergibt sich ein *video memory* (VM) von 12 MByte und eine virtuelle Auflösung von 2048 x 2048 Bildpunkten mit 24 Bit Bildtiefe, d.h. drei Farbenen (RGB) mit je 256 Graustufen. Die Auflösung des RGB-Bildschirmes beträgt 1472 x 1024 Bildpunkte. *Zoom*- und *pan*-Funktionen und die *look-up-tables* sind hardware-implementiert und werden über *memory*-Adressen im Controller gesteuert.

Da es sich um ein System mit verteiltem Speicher handelt, kann jeder Transputer nur auf seinen Teil des VM zugreifen. Die Zuordnung des virtuellen VM auf die Transputer erfolgt spaltenweise. Stellt man sich die lokalen VM der einzelnen Transputer als Bildmatrizen mit einer Ausdehnung von jeweils 256 x 2048 Bildpunkten vor, und das virtuelle VM als Bildmatrix mit 2048 x 2048 Bildpunkten, dann repräsentieren die ersten acht Spalten des virtuellen VM die acht ersten Spalten der einzelnen lokalen VM. Die neunte Spalte des virtuellen VM repräsentiert dann die zweite Spalte des lokalen VM des ersten Transputers usw. (Abb. 2.5). Definiert man  $vVM_i$  als  $i$ -te Spalte des virtuellen VMs und  $VM_{j,k}$  als  $k$ -te Spalte des Transputers  $j$  (mit  $0 \leq i < 2048$ ,  $0 \leq k < 256$ ,  $0 \leq j < 8$ ), läßt sich die Zuordnung mit der folgenden Formel beschreiben:

$$vVM_i = VM_{j,k} \text{ wenn } i = (k * 8) + j \text{ bzw. } k = (i - j) / 8 \quad (2.1)$$

Der Vorteil dieser Aufteilung liegt in der gleichmäßigen Auslastung der gesamten WU, wenn ein am virtuellen VM gespeichertes Bild verarbeitet wird. Auf der anderen Seite wird die Implementierung bestimmter Algorithmen durch die spaltenweise Zuordnung komplizierter.

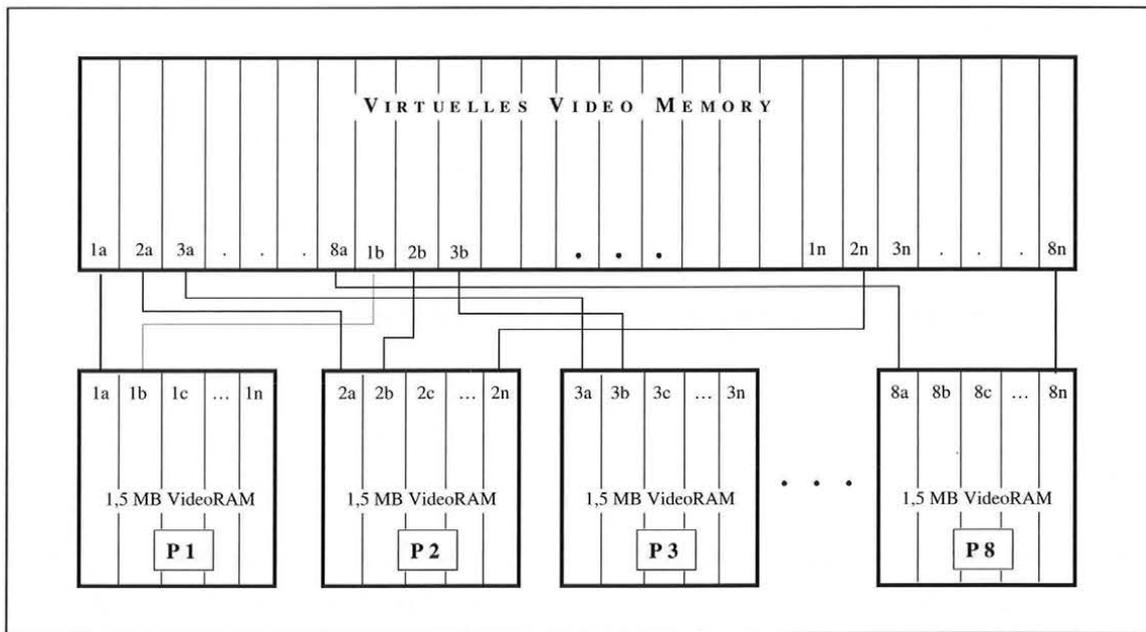


Abb. 2.5: Zuordnung der lokalen *video memories* zum virtuellen *video memory*

## 2.3 SOFTWARE

### 2.3.1 VORAUSSETZUNGEN

Angewandte Bildverarbeitung in der Fernerkundung ist in hohem Maße mit dem Problem konfrontiert, daß die Verarbeitung der sehr großen Datenmengen in akzeptalen Zeiträumen die Leistungsgrenzen der Systeme erreicht oder überschreitet [Faust et al. 1991]. Einzelne Algorithmen, meist dem Bereich der *low-level* Bildverarbeitung zugehörig (vgl. Kap. 1.2), werden als Hardware-Realisierungen angeboten. Sie ermöglichen eine effiziente Vorverarbeitung von Bilddaten in Echtzeit. Die eigentliche Bildanalyse bleibt jedoch den Software-Lösungen vorbehalten [Mihatsch und Kille 1990]. Der Einsatz paralleler Rechner bietet hier eine Alternative, die nicht durch die physikalische Leistungsgrenze von Einzelprozessoren limitiert ist.

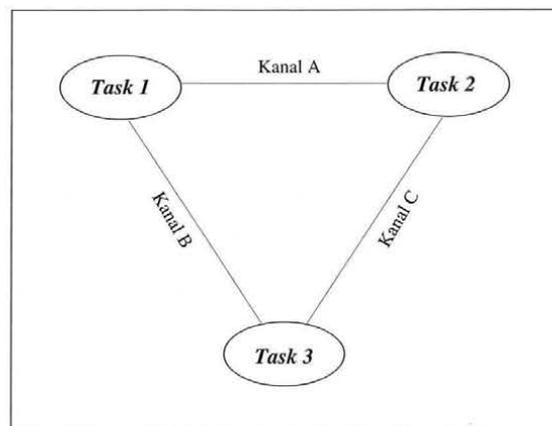
Die Charakteristika der Bildverarbeitung zeigen eine deutliche Eignung zur Parallelisierung [Webb 1992]. Einerseits beinhaltet die Struktur von Bilddaten eine natürliche Parallelität, da Bilder zur Verarbeitung meist in logisch zusammenhängende Teile zerlegt und diese anschließend unabhängig voneinander verarbeitet werden können. Darüber hinaus werden gleiche Operationen oft auf den gesamten Bildinhalt angewendet. Diese Eigenschaft erlaubt sogar eine regelmäßige Unterteilung der Bilder und erleichtert die optimale Auslastung der

einzelnen Prozessoren (*load balancing*). Die äußere Schale der Programme ist bei vielen Bildverarbeitungsalgorithmen ident, wodurch parallele Rahmenprogramme für die Algorithmen nur einmal entwickelt werden müssen.

## OCCAM

Die Programmiersprache OCCAM basiert auf dem Konzept kommunizierender sequentieller Prozesse (CSP - *communicating sequential processes*). Die Beschreibung einer Applikation erfolgt in Form unabhängiger Prozesse (*tasks*), die gleichzeitig ablaufen und über Kanäle miteinander kommunizieren können (Abb. 2.6). Jeder *task* verfügt über lokale Variablen, globale Variablen für mehrere *tasks* sind nicht erlaubt. Der Austausch von Daten ist nur über Kanäle möglich, die jeweils zwei *tasks* verbinden. Die Kommunikation ist synchronisiert, d.h. es kann nur kommuniziert werden, wenn sowohl der sendende als auch der empfangende Prozeß dazu bereit sind [May 1987].

OCCAM hat eine spezielle Beziehung zu INMOS Transputern, die gewissermaßen die Hardware-Umsetzung der OCCAM Philosophie darstellen. OCCAM Programme werden mehr oder weniger direkt von Transputern ausgeführt. Die Implementierung von parallelen *tasks* kann sowohl auf einem Transputer als auch auf einem Netzwerk von Transputern realisiert werden.



**Abb.2.6: Kommunizierende Tasks**

OCCAM Programme [INMOS, 1988b] bestehen aus drei primitiven Prozessen:

$v := e$     *assignment*  
 $c ! e$     *output*  
 $c ? v$     *input*

Das *assignment* weist der Variablen  $v$  den Ausdruck  $e$  zu. Der *output* -Prozeß ermöglicht die Zuweisung des Ausdruckes  $e$  zum Kanal  $c$  und der *input* -Prozeß liest einen Wert vom Kanal  $c$  und weist ihn der Variablen  $v$  zu.

Diese primitiven Prozesse werden in folgenden Konstrukten zusammengefaßt:

*SEQ*    sequentiell  
*IF*    konditionell

*PAR*    parallel  
*ALT*    alternativ

Ein Konstrukt stellt wiederum einen Prozeß dar, der als Teil eines anderen Konstruktes verwendet werden kann.

Die innerhalb eines PAR-Konstruktes befindlichen Prozesse werden parallel abgearbeitet, dürfen nicht auf gemeinsame Variablen zugreifen und kommunizieren nur über Kanäle.

*PAR*<sup>2</sup>  
... *Prozeß eins*  
... *Prozeß zwei*  
... *Prozeß drei*

ALT-Konstrukte ermöglichen die Kommunikation eines Prozesses mit einem von mehreren anderen parallel laufenden Prozessen. Die einfachste Form eines ALT-Konstruktes besteht aus *input*-Prozessen, denen ihrerseits Prozesse nachfolgen.

---

<sup>2</sup> Drei Punkte (...) vor einem Text deuten an, daß es sich um einen nicht näher beschriebenen Prozeß handelt, der durch den Text kurz beschrieben wird. Zwei Striche (--) vor einem Text definieren einen Kommentar im Programm.

*ALT*

*chan1 ? v*

*... Prozeß eins*

*chan2 ? v*

*... Prozeß zwei*

*chan3 ? v*

*... Prozeß drei*

Je nach dem über welchen Kanal - chan1, chan2 oder chan3 - eine Nachricht eintrifft, wird der dazugehörige Prozeß - und nur dieser - ausgeführt.

OCCAM verfügt über zwei weitere Prozesse, SKIP und STOP, die spezielle Charakteristika aufweisen. SKIP ist ein Prozeß, der startet, nichts tut und terminiert. Er wird eingesetzt, wenn die Syntax einen Prozeß verlangt, aber nichts getan werden soll. STOP ist ein Prozeß, der startet, nichts tut, aber auch nicht terminiert. Er wird bei der Programmentwicklung zur Fehlersuche eingesetzt (siehe Kap. 2.3.1.2). In diesem Zusammenhang sei auf die Bedeutung des Begriffes **terminieren** hingewiesen. Ein Prozeß, der alle seine Aufgaben erledigt hat, terminiert. Wird ein Prozeß gestoppt, so arbeitet er zwar nicht mehr, er terminiert aber auch nicht. Das passiert, wenn ein Prozeß auf ein Ereignis wartet, das nicht eintritt, z.B. auf Grund eines Programmfehlers. Man spricht dann von einem **deadlock**, der meist das gesamte Netzwerk lahmlegt. *Deadlocks* treten z.B. dann auf, wenn der Kommunikationsablauf fehlerhaft ist:

*PAR*

*SEQ*                    -- Prozeß A

*A\_to\_B ! x*

*B\_to\_A ? y*

*SEQ*                    -- Prozeß B

*B\_to\_A ! y*

*A\_to\_B ? x*

Prozeß A versucht über den Kanal A\_to\_B eine Nachricht an Prozeß B zu schicken, der seinerseits versucht über den Kanal B\_to\_A eine Nachricht an den Prozeß A abzusetzen. Syntaktisch ist das Programm korrekt, die Exekution führt jedoch zu einem *deadlock*. In diesem Fall ist der Fehler leicht zu beheben. Es genügt, im Prozeß A oder im Prozeß B die beiden Programmzeilen zu vertauschen. Bei komplexeren Kommunikationen zwischen mehreren Prozessen kann die Fehlersuche jedoch aufwendig werden.

Mit Hilfe der beschriebenen Prozesse läßt sich die Gleichzeitigkeit der realen Welt in natürlicher Weise darstellen. Zum besseren Verständnis soll ein Beispiel dienen, das dem alltäglichen Leben entnommen ist, auch wenn es in dieser Art wohl kaum als Programm umgesetzt werden wird. Man stelle sich eine Sekräterin vor, die eine Anzahl von Briefen zu schreiben und außerdem das Telefon und den Türöffner des Büros zu bedienen hat. Ihre Hauptaufgabe besteht also aus Schreiben, wobei sie parallel dazu auf das Läuten der Türklingel oder des Telefons zu achten hat. Das alles allerdings nur bis zum Ende ihrer Arbeitszeit um 17 Uhr. Eine absolut problemlose Aufgabe für einen Menschen. Versucht man diese Tätigkeiten in einem sequentiellen Programm zu realisieren, wird man auf beträchtliche Schwierigkeiten stoßen. In OCCAM läßt sich das Problem elegant lösen, wie das nachfolgende Beispielprogramm zeigt.

*PRI PAR*

```

-- Hör-Prozeß -----
SEQ
  ende = FALSE
  WHILE NOT ende
    ALT
      tür ? ton                -- Türklingel läutet
        ...Türöffner betätigen
      tel ? ton                -- Telefon läutet
        SEQ
          ...Telefon abnehmen
          ...Telefongespräch führen
          ...Telefon auflegen
      stop ? ende              -- Arbeitszeit zu Ende
    SKIP

-- Schreib-Prozeß -----
SEQ
  zeit := 9
  WHILE zeit < 17
    SEQ
      ...Brief schreiben
      uhr ? zeit
  stop ! TRUE

```

Dazu noch eine kurze Erklärung der im Beispiel verwendeten Ausdrücke, die noch nicht beschrieben wurden: WHILE definiert eine Schleife, die solange läuft, als die dazugehörige Bedingung wahr ist; TRUE und FALSE sind boolesche Konstanten im Sinne ihrer Namen.

Solange über die Kanäle **tür** und **tel** keine Nachrichten kommen, wird der Schreib-Prozeß am Ende des Beispiels ausgeführt. Sobald einer der beiden *input*-Prozesse aktiv wird, wird der Schreib-Prozeß unterbrochen und der jeweilige Folgeprozeß ausgeführt. Ist dieser beendet, kehrt das Programm zum "Briefe schreiben" zurück. Nach jedem geschriebenen Brief wird kontrolliert, ob die Arbeitszeit nicht schon vorbei ist. Tritt dieser Fall ein, schickt der Schreib-Prozeß über den Kanal Arbeit eine Nachricht zum Hörprozeß, der daraufhin seine Tätigkeit beendet.

Diese Situation ist ein Beispiel für Parallelität auf einem Prozessor, wobei die eigentliche Parallelität im gleichzeitigen Schreiben und Hören besteht. Dem PAR ist in diesem Falle noch ein PRI (für Priorität) vorangestellt, das bewirkt, daß der Hör-Prozeß dem Schreib-Prozeß vorgezogen wird, sobald ein *input*-Prozeß aktiv wird. Die Prozesse, die über die Kanäle **tür**, **tel** und **uhr** mit dem System verbunden sind, laufen parallel zum beschriebenen System und könnten ihrerseits wiederum mit weiteren externen Prozessen in Verbindung stehen, wie das ja auch in der realen Welt der Fall ist.

## Entwicklungsumgebung

Für die Softwareentwicklung auf Transputern bietet INMOS eine Entwicklungsumgebung an, die sogenannten ITOOLS, die aus folgenden Komponenten bestehen [INMOS, 1989]:

OCCAM COMPILER

LINKER

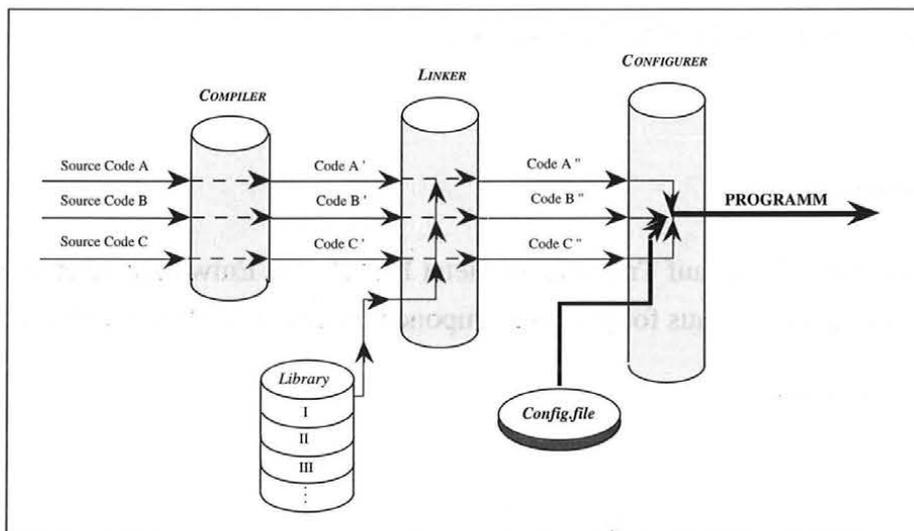
CONFIGURER

Bevor compilierte und gelinkte OCCAM *tasks* auf einem Transputernetzwerk exekutiert werden können, muß die Verteilung der einzelnen *tasks* auf dem Netzwerk definiert werden. Das geschieht mittels eines Konfigurationsfiles, in dem die Zuordnung der *tasks* auf die Prozessoren und der Kanäle auf die Transputer *links* beschrieben wird. Für die Erstellung dieses Files ist der Softwareentwickler verantwortlich. Der CONFIGURER interpretiert den Inhalt des Files, hängt diese Information an die einzelnen *tasks* und verbindet alle *tasks* zu einem OCCAM Programm. Erst ein solches Programm ist auf

einem Transputernetzwerk lauffähig. Abb. 2.7 zeigt schematisch, wie ein OCCAM Programm entsteht.

#### SERVER

Um ein OCCAM Programm zu exekutieren, muß der Code auf das Netzwerk geladen werden. Diese Aufgabe übernimmt der SERVER, ein Programm das auf dem Host läuft. Er liest das OCCAM Programm von der Harddisk und schickt es weiter an den ersten Transputer im Netzwerk (Root). Dort wird der für diesen Transputer relevante Code auf das lokale *memory* gelesen und gestartet. Der restliche Code wird an den nächsten Transputer weitergeleitet, der dann seinerseits die für ihn bestimmten *tasks* exekutiert usf. Während der Laufzeit des OCCAM Programmes sorgt der SERVER für die Kommunikation zwischen Host und Netzwerk. Im Fall des System 2400 übernimmt der SERVER alle I/O Funktionen (*keyboard input, screen output, Harddisk I/O*) abgesehen vom RGB-Schirm, der systemintern verwaltet wird (vgl. Abb. 2.3).



**Abb. 2.7: Entstehung eines OCCAM Programmes**

#### DEBUGGER

Um die Fehlersuche bei der Programmentwicklung zu erleichtern, bietet INMOS auch einen *post-mortem-DEBUGGER* an. Terminiert ein Programm nicht, sondern stoppt, kann der Code und der Inhalt des *memory* des gesamten Netzwerkes auf ein File herausgeschrieben werden. Im Anschluß daran kann das fehlerhafte Programm mit Hilfe

des DEBUGGERS auf Source-Code- oder Assembler-Code-Ebene analysiert werden (welcher *task* hat gestoppt, wo hat er gestoppt, haben andere *tasks* gestoppt, wurde über *links* kommuniziert, welchen Wert haben Variablen usf.). Dabei ist der OCCAM Prozeß STOP sehr nützlich, da das Programm gezielt angehalten und untersucht werden kann. Allerdings beschränkt sich die Analyse auf den Zeitpunkt des Programmstops. Ereignisse, die vorher passiert sind, oder Variablen, die außerhalb des aktuellen Bereichs liegen, können nicht analysiert werden.

#### LIBRARIEN

Der Vollständigkeit halber sei noch auf die Möglichkeit verwiesen, eigene Bibliotheken anzulegen, auf die beim Linken von Programmen zugegriffen werden kann.

### 2.3.2 SOFTWARE DESIGN

Da es sich beim System 2400 um ein reines Hardware Produkt handelte, ohne einer wie immer gearteten Software, mußte die Programmentwicklung an der Basis begonnen werden [vgl. Babnik und Steinnocher,1991]. Die Rahmenbedingungen für die Softwaregestaltung waren nur durch die Architektur des Rechners gegeben, Beschränkungen von Seiten der Software, wie z.B. vorgegebene Kommunikationsroutinen, lagen nicht vor. Die sich daraus ergebende Freiheit in der Programmgestaltung beinhaltet jedoch im Vergleich zur herkömmlichen Programmierung eine enorme Steigerung des Entwicklungsaufwands. Ein zusätzlicher Aspekt ist der Wechsel vom Konzept der sequentiellen zu dem der parallelen Programmierung. Dazu bedarf es des Verständnisses von Parallelität in Problemstellungen, Umsetzung dieser Parallelität auf kommunizierende Prozesse und Abbildung dieser Prozesse auf die Hardware. Die Hauptschwierigkeit lag am Beginn der Arbeit vor allem im Erkennen möglicher Parallelität. Interessanterweise schlug das Pendel nach der Einarbeitungsphase zeitweise zu stark in Richtung Parallelität aus, d.h., es wurde versucht, Abläufe zu parallelisieren, die effizienter in Form sequentieller Prozesse laufen, z.B. auf Grund eines zu hohen Kommunikationsaufwands bei paralleler Implementierung. Das endgültige Software Design, das aus den Erfahrungen der ersten Monate entstanden ist, basiert dementsprechend auf einfachen Ablaufschemata, die die Parallelität der Problemstellungen effizient ausnützen. Weiters wurde darauf geachtet, allgemeine Lösungen zu kreieren, die die Erweiterung der Programme um ähnlich geartete Algorithmen ohne zusätzlichen Aufwand ermöglichen.

## **Kernel und Oberfläche**

Das folgende Kapitel beschäftigt sich mit zwei Themen, die im Detail in der parallel entstehenden Dissertation [vgl. Babnik, 1994] beschrieben werden. Um die Konsistenz und Vollständigkeit der vorliegenden Arbeit zu wahren, wird die Problematik kurz umrissen.

Transputer brauchen kein Betriebssystem im herkömmlichen Sinn. Es genügt, das OCCAM Programm über einen *link* an einen Transputer zu schicken. Sobald dieser den Code empfängt, führt er ein *reset* durch, lädt den Code ins *memory*, und startet die Verarbeitung desselben. Dadurch können verschiedene Programme hintereinander gestartet und abgearbeitet werden. Im Fall des System 2400 stellt sich jedoch ein Problem: Angenommen man liest Bilddaten mit einem Programm auf das VideoRAM der WU; mit dem folgenden Programm sollen diese Daten verarbeitet werden; leider führt der Transputer beim *booten* des zweiten Programmes ein *reset* durch, das sich zeitweise auf das VideoRAM auswirkt; die Folge sind Veränderungen der Bilddaten am VideoRAM. Um dieses Problem zu umgehen, wurde ein *kernel* entwickelt, der wie ein kleines Betriebssystem agiert. Er läuft während der gesamten Verarbeitungszeit und lädt die einzelnen Programme dynamisch. Dadurch wird das *reset* beim Programmstart vermieden, die Fehler am VideoRAM treten nicht auf.

Da die I/O Library der ITOOLS nur einfache, alphanumerische Ein- und Ausgaben unter DOS zuläßt, wurde eine mausgestützte Benutzeroberfläche unter WINDOWS entwickelt. Zu diesem Zweck mußte der ITOOLS SERVER durch eine WINDOWS-Applikation ersetzt werden. Das eingesetzte Entwicklungswerkzeug ist COMMON VIEW. Als problematisch erwies sich dabei die Kommunikation zwischen der Oberfläche und dem *Root*.

In der endgültigen Version wird die Oberfläche zusammen mit dem *kernel* gestartet. Die Auswahl der Module erfolgt über *pull-down-menus* in der Benutzeroberfläche. Nach Wahl eines Moduls wird der betreffende Code dynamisch auf das Netzwerk geladen und exekutiert. Die Kommunikation zwischen Benutzer und dem gewählten Modul erfolgt über die betreffende Prozedur der Oberfläche. Zum Zeitpunkt der Verfassung der vorliegenden Arbeit standen der *kernel* voll und die Oberfläche als Testversion für einige Module zur Verfügung.

## **Bildverarbeitungsmodule**

Die anwendungsorientierte Verarbeitung von Bilddaten erfolgt oft in zwei Schritten. Zuerst werden Algorithmen auf ihre Eignung bei einer bestimmten Problemstellung überprüft und die geeigneten Werte der Parameter bestimmt. Das geschieht normalerweise anhand eines repräsentativen Bildausschnittes und nicht am gesamten Bild, um die Laufzeiten während der

Testphase zu minimieren. Sind die geeigneten Algorithmen gefunden, werden sie auf die zu verarbeitenden Bilder angewandt. Der erste Schritt erfordert geeignete Visualisierungsmöglichkeiten, da die Beurteilung der eingesetzten Algorithmen und ihrer Parameter meist visuell erfolgt. Die Verarbeitung sollte auf dem VideoRAM (*screen-to-screen*) erfolgen, ohne Zugriffe auf die externe Harddisk. Der zweite Schritt benötigt ein Datenverwaltungsprogramm, das in der Lage ist, beliebig große Bilder der Verarbeitung zuzuführen. Die Visualisierung ist dabei nicht relevant. Die folgenden beiden Kapitel beschreiben zwei Rahmenprogramme, die die eben gestellten Aufgaben für Algorithmen erfüllen, die auf der Analyse der lokalen Umgebung eines Pixels aufsetzen.

### **2.3.3 SCREEN-TO-SCREEN VERARBEITUNG**

Wie bereits beschrieben, erfolgt die Aufteilung des virtuellen, am RGB-Schirm sichtbaren VideoRAMs auf die Transputer der WU nach Spalten mit einer Breite von einem Pixel. Daher erfordert jede *screen-to-screen* Operation, die mehr als ein Pixel bzw. eine Pixelspalte in ihre Rechnung einbezieht (also Nachbarschaftsoperation), Kommunikation zwischen den Prozessoren. Ein typisches Beispiel dafür ist die Faltung. Während ihrer schrittweisen Bewegung über das Bild überlagert die Faltungsmatrix in jeder Position mehrere Pixelspalten. Die Multiplikation der Elemente der Faltungsmatrix mit den Bildelementen kann zwar lokal durchgeführt werden, die Summenbildung und Zuweisung zum Mittelelement erfordert jedoch Datenaustausch zwischen den Prozessoren.

#### **Parallele Faltung**

Ausgehend von einer 3x3 Faltungsmatrix soll eine Faltung auf dem verteilten VideoRAM durchgeführt werden. Die Faltungsmatrix sei auf allen Prozessoren vollständig verfügbar. In der Anfangsposition liegt die Faltungsmatrix über den ersten drei Spalten des Bildes, die auf drei Prozessoren verteilt sind. Die Multiplikation der Matrixelemente mit den Bildelementen und die Summenbildung innerhalb der drei Spalten wird auf allen drei Prozessoren parallel durchgeführt. Anschließend wird die Produktsumme des linken und des rechten Prozessors an den mittleren Prozessor geschickt, der die empfangenen Summen zu seiner lokalen Produktsumme addiert und dem Mittelpixel zuweist. Danach wird der Ablauf, um eine Position nach rechts verschoben, wiederholt.

Es ist offensichtlich, daß diese Art der Parallelisierung nicht sehr sinnvoll ist, da immer nur drei Prozessoren parallel arbeiten. Natürlich könnte man die drei folgenden Prozessoren zur

gleichen Zeit mit derselben Aufgabe beschäftigen, dann blieben bei insgesamt acht Prozessoren immer noch zwei übrig, die untätig warten müßten. Um die Parallelität des Systems voll auszunützen, ist es notwendig, den Algorithmus so zu zerlegen, daß alle Prozessoren annähernd denselben Arbeitsaufwand haben (*load balancing*). Im Fall der Faltung läßt sich das relativ einfach realisieren.

Die mathematische Formulierung der Faltung ist eine Produktsumme, die folgendermaßen dargestellt werden kann (nach *Kraus*, [1990]):

$$G_{i,j} = \sum_{k=-n}^n \sum_{l=-n}^n g_{i-k,j-l} * w_{n+1+k,n+1+l} \quad (2.2)$$

G repräsentiert die Bildelemente des Ergebnisbildes, g die des Ausgangsbildes und w die Faltungsmatrix mit der Größe  $2n+1$ . Die Indizes i und j beschreiben die Position in der Matrix des Ergebnisbildes. Da die Faltung für die Elemente der ersten n und der letzten n Zeilen und Spalten des Ausgangsbildes nicht durchgeführt werden kann, ist das Ergebnisbild um 2n Spalten und 2n Zeilen kleiner als das Ausgangsbild. Bei der Berechnung eines Ergebnisbildes der Ausdehnung  $I \times J$  mit einer Faltungsmatrix  $(2n+1) \times (2n+1)$ , durchlaufen i und j die Wertebereiche  $1 \leq i \leq I$  und  $1 \leq j \leq J$ . Bei sequentieller Programmierung ergeben sich daraus vier Schleifen, die ineinander verschachtelt sind:

```

for 1 ≤ i ≤ I      -- Schleife über die Bildmatrix
  for 1 ≤ j ≤ J
    for -n ≤ k ≤ n  -- Schleife über die Faltungsmatrix
      for -n ≤ l ≤ n
        ... Produktsumme
      ... Zuweisung zum Mittelpixel

```

Störend für die Parallelisierung im Sinne des verteilten VideoRAM ist die Tatsache, daß die Spaltenindizes l der Faltungsmatrix, die Kommunikation erfordern, innen liegen. Der Schleifenkomplex läßt sich mit Einführung von Zwischensummen auch folgendermaßen beschreiben:

```

for -n ≤ l ≤ n
  for -n ≤ k ≤ n
    for 1 ≤ i ≤ I
      for 1 ≤ j ≤ J
        ... Produktsumme      -- I*J Produktsummen
      ... Summe
    ... Zuweisung zum Mittelpixel

```

Die Reihenfolge ist nur leicht verändert. Zuerst wird das Ausgangsbild Spalte für Spalte mit der ersten Spalte der Faltungsmatrix multipliziert, aufsummiert und am lokalen RAM gespeichert. Danach wird das Bild Spalte für Spalte mit der zweiten Spalte der Faltungsmatrix multipliziert, aufsummiert und lokal gespeichert usw. Sind alle  $2n+1$  Spalten der Faltungsmatrix auf diese Weise abgearbeitet, bestehen  $I*J*(2n+1)$  Produktsummen, die zu  $I*J$  Werten aufsummiert und den  $I*J$  Pixeln des Ergebnisbildes zugewiesen werden. Bei einer parallelen Verarbeitung erfordert nur die letzte Summenbildung eine Kommunikation zwischen den Prozessoren, alle vorangegangenen Produktsummen können lokal gebildet werden.

Besteht das Netzwerk aus  $N$  Transputern beträgt der Arbeitsaufwand eines Transputers dementsprechend:

$I*J*(2n+1) / N$  Produktsummenberechnungen plus

$I*J*(2n) / N$  Summationen.

Der Kommunikationsaufwand pro Transputer beläuft sich auf:

$I*J*(2n) / N$  parallel empfangene und gesendete Werte.

Zum Datenaustausch zwischen den Prozessoren kommt es jeweils nach der Produktsummenbildung einer Spalte der Faltungsmatrix mit dem Ausgangsbild. Die Kommunikation erfolgt über den Linkswitch, der jeweils nach Verarbeitung einer Spalte der Faltungsmatrix neu konfiguriert wird. Der Abstand zwischen den Prozessoren, die verbunden werden müssen, entspricht dem Abstand der aktuellen Spalte von der Mittelspalte der Faltungsmatrix. Zum Beispiel wird nach Verarbeitung der ersten Spalte einer  $5 \times 5$  Faltungsmatrix die errechnete Produktsumme zwei Spalten nach rechts, also zwei Prozessoren nach rechts geschickt (Abb. 2.8). Im nächsten Schritt wird das Verarbeitungsergebnis der zweiten Spalte der Faltungsmatrix um einen Prozessor nach rechts geschickt usw. Da die Erstellung der Produktsummen für alle Prozessoren nahezu gleich lang dauert, kann der Datenaustausch gleichzeitig erfolgen.

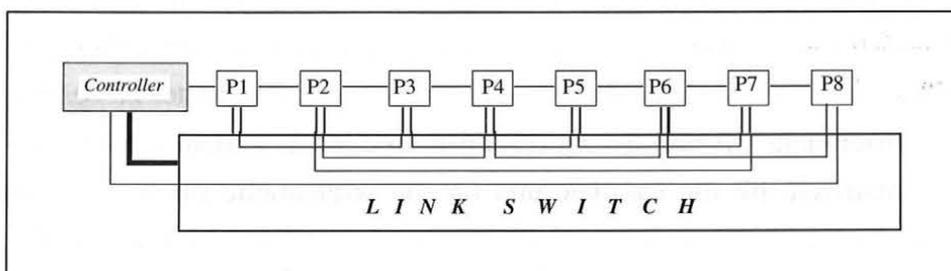


Abb. 2.8: Linkswitch-Konfiguration (Abstand zwei)

Abb. 2.9 zeigt die Implementierung des Faltungsalgorithmus auf dem System 2400. Zuerst wird die Faltungsmatrix über die *pipeline* an alle Prozessoren geschickt (*param*). Während die WU eine Spalte der Faltungsmatrix verarbeitet (*convolve*), setzt der Controller den Linkswitch für die nächste Datenübertragung (*set.ls*). Dann erfolgt eine Synchronisation über die *pipeline* (*sync*) um sicherzustellen, daß alle Prozesse und der Linkswitch bereit sind, zu kommunizieren. Danach erfolgt der Datenaustausch, bei dem parallel gesendet und empfangen wird (*send, receive*). Nach einer weiteren Synchronisation beginnt der Ablauf mit der nächsten Spalte der Faltungsmatrix von vorne.

root	controller	working unit 1	working unit 2	...
inp.param				
send.param ->	rec.send.param ->	rec.send.param ->	rec.send.param ->	
	LOOP	LOOP	LOOP	
	set.ls	convolve	convolve	
	sync ->	sync ->	sync ->	
		PAR	PAR	
		send	send	
		receive	receive	
	sync <-	sync <-	sync <-	
	loop end ?	LOOP END	LOOP END	
end <-	end <-	end <-	end <-	

**Abb. 2.9: Ablaufschema parallele Faltung**

### Effizienz der parallelen Implementierung

Der vorliegende Algorithmus wurde herangezogen, um die Effizienz der Parallelität auch quantitativ zu testen. Dank der in OCCAM verfügbaren *timer* konnte die Laufzeit der einzelnen Operationen gemessen werden. Zusätzlich wurde ein sequentieller Faltungsalgorithmus für den Root entwickelt, um den *speed-up* von einem auf acht Transputer zu bestimmen. Tab. 2.2 zeigt den prozentuellen Anteil der Operationen an der Gesamtlaufzeit bei drei verschiedenen Faltungsmatrizen, für die parallele und für die sequentielle Implementierung. In der Spalte Berechnung sind die Anteile der Faltungsschleife mit Produktsummenbildung, in der Spalte Kommunikation die Anteile für Datenaustausch und Synchronisation angegeben. Alle anderen Operationen fallen in die Spalte Verwaltung. Bei der sequentiellen Verarbeitung bleibt die Spalte Kommunikation natürlich leer, da nur ein Prozessor verwendet wird.

**Tab. 2.2: Prozentuelle Anteile der Operationen an der Gesamtlaufzeit**

	Faltungsmatrix	Verwaltung	Berechnung	Kommunikation
parallel	9x9	6.5	92.5	1.0
	5x5	14.4	84.2	1.4
	3x3	30.0	68.6	1.8
sequentiell	9x9	2.7	97.3	---
	5x5	8.1	91.9	---
	3x3	19.7	80.3	---

Es ist deutlich zu sehen, daß der Kommunikationsaufwand im Vergleich zur Berechnungszeit gering ist. Der Anteil für Kommunikation nimmt auch mit zunehmender Größe der Faltungsmatrix ab, da der Berechnungsaufwand quadratisch und der Kommunikationsaufwand linear in Abhängigkeit der Größe der Faltungsmatrix steigt<sup>3</sup>. Der höhere Verwaltungsaufwand bei der parallelen Verarbeitung ist einerseits auf zusätzliche Aufgaben zurückzuführen, die durch die parallele Verarbeitung entstehen, andererseits auch auf die Tatsache, daß pro Prozessor nur ein Achtel des Bildes verarbeitet wird, ein Teil des *Verwaltungsoverheads* jedoch konstant bleibt.

Der Vergleich der absoluten Laufzeiten zwischen paralleler und sequentieller Verarbeitung ist in Tabelle 2.3 dargestellt. Der angegebene *speed-up* errechnet sich aus dem Verhältnis der sequentiellen zur parallelen Laufzeit, d.h. er kann bei acht Prozessoren maximal den Faktor acht erreichen.

Die Effizienz<sup>4</sup> der parallelen Implementierung liegt zwischen 82.5 und 95%, der mittlere *overhead*<sup>5</sup> beträgt 11.4%. Diese Ergebnisse zeigen, daß die parallele Implementierung von Bildverarbeitungsoperationen schon bei kleinen Bildern und kleiner Faltungsmatrix interessant ist.

<sup>3</sup> Bei einer Größe der Faltungsmatrix von  $N \times N$  und einer Größe des Ergebnisbildes von  $I \times J$  werden  $2N^2 \cdot I \cdot J$  Rechenoperationen durchgeführt und  $(N-1) \cdot I \cdot J$  Daten übertragen.

<sup>4</sup> Effizienz =  $100 \cdot t_s / (N \cdot t_p)$

mit  $t_s$  gleich der Laufzeit der sequentiellen und  $t_p$  gleich der Laufzeit der parallelen Implementierung auf  $N$  Prozessoren.

<sup>5</sup> *overhead* =  $100 - \text{Effizienz}$

**Tab. 2.3: *Speed-up* = Laufzeit sequentiell / Laufzeit parallel**

Größe der Faltungsmatrix	Größe des Bildes		
	512x512	256x256	128x128
3x3	6.8	6.7	6.6
5x5	7.1	7.0	6.7
9x9	7.3	7.2	6.9
13x13	7.4	7.3	7.1
17x17	7.6	7.4	7.2

### Lokale Operatoren

Der im letzten Kapitel beschriebene Algorithmus stellt eine Methode der parallelen Implementierung einer Faltung dar. Andere lokale Operationen könnten zwar nach diesem Konzept entwickelt werden, nicht jedoch ohne einen gewissen Aufwand, da die Bildverarbeitungsoperationen durch die Kommunikation zerteilt werden. Interessanter ist dagegen ein Ansatz, der die Durchführung der lokalen Operationen unabhängig von der Kommunikation erlaubt. Dazu wird das Ausgangsbild in annähernd gleich große Teile zerlegt und auf die acht Prozessoren verteilt. Dadurch kann jeder Prozeß seinen Bildteil unabhängig von den übrigen Prozessen verarbeiten. Nach Beendigung der Operation werden die verarbeiteten Bildteile wieder so auf die Prozesse rückverteilt, daß sich das Ergebnisbild am RGB-Schirm darstellen läßt.

Bei der Aufteilung des Ausgangsbildes müssen die Überlappungsbereiche berücksichtigt werden, die an der Grenze zwischen zwei Bildteilen auftreten. Das sind Bereiche, die in beiden Bildteilen verfügbar sein müssen, da lokale Bildverarbeitungsoperationen Nachbarpixel in die Verarbeitung miteinbeziehen. Die Größe der Überlappungsbereiche hängt von der Ausdehnung des gewählten Fensters ab<sup>6</sup>.

---

<sup>6</sup> Nimmt man eine ungerade Fenstergröße  $N$  an, so ergibt sich der notwendige Überlappungsbereich als  $(N-1)/2$ .

Das gesuchte Programm arbeitet also in drei Schritten: Umverteilung der Daten, Verarbeitung und Rückverteilung der Daten. Die Problematik der Umverteilung läßt sich am besten anhand einer Graphik aufzeigen (vgl. Abb. 2.10).

1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
1	2	3	4	5	6	7	8	1	2	3	...
·	·	·									·
·	·	·									·
·	·	·									·

1	1	1	1	1	1	1	1	1	1	1	...
1	1	1	1	1	1	1	1	1	1	1	...
2	2	2	2	2	2	2	2	2	2	2	...
2	2	2	2	2	2	2	2	2	2	2	...
3	3	3	3	3	3	3	3	3	3	3	...
3	3	3	3	3	3	3	3	3	3	3	...
4	4	4	4	4	4	4	4	4	4	4	...
4	4	4	4	4	4	4	4	4	4	4	...
·	·	·									·
·	·	·									·
·	·	·									·

**Abb. 2.10: Bildmatrix vor und nach der Umverteilung**

Die linke Teil Abb. 2.10 zeigt eine Bildmatrix, die so auf die Prozessoren verteilt ist, daß sie am RGB-Schirm als Bild darstellbar ist. Die Zahlen in den Elementen der Bildmatrix entsprechen jeweils dem Prozessor, dessen lokalen VideoRAM das jeweilige Bildelement zugeordnet ist. Im rechten Teil der Abbildung ist die Bildmatrix in einzelne zusammenhängende Teile zerlegt, die den Prozessoren zugeordnet sind und unabhängig voneinander verarbeitet werden können.

Betrachtet man die Bildteile, die den Prozessoren nach der Umverteilung zugeordnet sind, so sieht man, daß sich jene aus Daten aller Prozessoren zusammensetzen. Um den Kommunikationsaufwand zu minimieren, erfolgt der Datenaustausch ähnlich strukturiert wie beim Faltungsalgorithmus. Zuerst werden die Daten am lokalen VideoRAM nach Zielprozessoren aufgeteilt (vgl. Abb. 2.11).

Prozessor 1										Prozessor 2										...									
1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	...	...									
1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	...	...									
2	2	2	2	2	2	2	2	2	...	2	2	2	2	2	2	2	2	2	...	...									
2	2	2	2	2	2	2	2	2	...	2	2	2	2	2	2	2	2	2	...	...									
3	3	3	3	3	3	3	3	3	...	3	3	3	3	3	3	3	3	3	...	...									
3	3	3	3	3	3	3	3	3	...	3	3	3	3	3	3	3	3	3	...	...									
4	4	4	4	4	4	4	4	4	...	4	4	4	4	4	4	4	4	4	...	...									
.									.	.									.	.									
.									.	.									.	.									
.									.	.									.	.									

**Abb. 2.11: Aufteilung nach Zielprozessoren am lokalen VideoRAM**

Die Zahlen in den Elementen des VideoRAM repräsentieren denjenigen Prozessor, auf dem die Daten verarbeitet werden sollen. Die *kursiv* gesetzten Bereiche werden nicht verschickt, sondern bleiben am jeweiligen Prozessor. Der Datenaustausch erfolgt über den Linkswitch, wobei schrittweise vorgegangen wird. Zuerst werden die Datengruppen verschickt, die um einen Prozessor "weiterwandern" (Zielprozessor = lokaler Prozessor + 1). Im nächsten Schritt wird zwei Prozessoren weiter geschickt (Zielprozessor = lokaler Prozessor + 2) usw. Die Kommunikation erfolgt parallel auf allen Prozessoren; zwischen den Kommunikationsschritten wird der Linkswitch vom Controller neu konfiguriert (siehe Abb. 2.12).

controller		working unit 1		working unit 2		working unit 3	...
LOOP		LOOP		LOOP		LOOP	
set.ls							
sync	->	sync ->		sync ->		sync ->	
		PAR		PAR		PAR	
		send		send		send	
		receive		receive		receive	
sync	<-	sync <-		sync <-		sync <-	
LOOP END		LOOP END		LOOP END		LOOP END	

**Abb. 2.12: Ablaufschema der Kommunikation bei der Umverteilung**

Nach Abschluß des Datenaustausches liegt auf jedem Prozessor ein zusammenhängender Bildausschnitt vor, der sequentiell verarbeitet wird. Dazu wird jeweils ein der Größe des Bearbeitungsfensters entsprechender Ausschnitt einer Bearbeitungsprozedur übergeben, von dieser verarbeitet, und das Resultat an den Hauptprozeß retourniert. Schematisch dargestellt sieht der Ablauf folgendermaßen aus:

```

for i          -- Bildspalten
  for j        -- Bildzeilen
    procedure(window[i,j,n,m],result[i,j]) -- Bearbeitungsprozedur

```

Die Parameter i,j repräsentieren die Pixel des Bildausschnittes, die Parameter n,m die Größe des Umgebungsfensters. Die Prozedur übernimmt also einen Bildausschnitt an der Stelle i,j mit der Größe nxm, verarbeitet diesen und gibt das Resultat für das Pixel i,j zurück. Was innerhalb der Prozedur passiert, ist für den Ablauf der Datenverwaltung nicht relevant. Dementsprechend können beliebige lokale Bildverarbeitungsprozesse an dieser Stelle implementiert werden.

Nach Verarbeitung der Bildausschnitte auf den acht Prozessoren, werden die Daten wieder so rückverteilt, daß sie der Aufteilung des VideoRAM entsprechen. Dieser Vorgang ist eine Umkehr der eingangs beschriebenen Umverteilung und wird daher nicht näher beschrieben.

Der gesamte Ablauf des Programmes ist schematisch in Abb. 2.13 dargestellt.

<b>controller</b>	<b>working unit 1</b>	<b>working unit 2</b>	<b>...</b>
Linkswitch Kontrolle	Datenumverteilung <->	Datenumverteilung <->	
---	lokale Verarbeitung	lokale Verarbeitung	
Linkswitch Kontrolle	Datenrückverteilung <->	Datenrückverteilung <->	

**Abb. 2.13: Ablaufschema des *screen-to-screen* Programmes**

Der Verwaltungsaufwand des *screen-to-screen* Programmes ist natürlich höher als der des zuvor beschriebenen parallelen Faltungsalgorithmus. Um abzuschätzen, wie sehr der Vorteil einer allgemeinen Lösung das Laufzeitverhalten beeinträchtigt, wurde die Laufzeit des reinen Faltungsalgorithmus mit der Laufzeit der im *screen-to-screen* Programm implementierten Faltung verglichen (Tab. 2.4).

**Tab. 2.4: Laufzeitverhältnis der Faltungsprogramme**

Größe der Faltungsmatrix	Größe des Bildes		
	512x512	256x256	128x128
3x3	68	67	65
5x5	83	83	81
9x9	84	84	83
13x13	92	92	92
17x17	92	91	88

Angaben in %

Tab. 2.4 zeigt, daß ab einer Größe der Faltungsmatrix von 5 x 5 die *screen-to-screen* Version zwischen 10% und 20% langsamer ist als das spezielle Faltungsprogramm. Die Differenz in der Laufzeit hält sich damit in einem akzeptalen Rahmen, zumal sie mit zunehmendem Rechenaufwand kleiner wird. Die Vereinfachung der Programmentwicklung, die hier auf Kosten der zeitlichen Effizienz gewonnen wird, rechtfertigt die Einbußen in jedem Fall.

### Konzept der Quadranten

Die Bilddaten, die mittels des beschriebenen Programmes verarbeitet werden, befinden sich vor und nach ihrer Bearbeitung auf dem VideoRAM. Wie bereits erwähnt, beträgt die Ausdehnung desselben 2048 x 2048 Pixel, von denen 1472 x 1024 auf dem RGB-Schirm dargestellt werden können. Um die Wirkungsweise von Operationen zu testen, wird meistens ein überschaubarer Bildausschnitt herangezogen. Dieser Bildausschnitt erreicht nur selten die volle darstellbare Größe von 2048 x 2048 Pixel, in der Regel verwendet man Ausschnitte von 512 x 512 Pixel oder kleiner. Um einen Algorithmus anhand eines solchen Bildausschnittes zu testen, wird zunächst der Ausschnitt von der Harddisk auf das *video memory* (VM) gelesen. Danach wird der zu bearbeitende Bereich des VM definiert, die Verarbeitung durchgeführt und das Ergebnisbild auf einen anderen definierten Bereich des VM geschrieben. Dieser Bereich kann durchaus mit dem Ausgangsbereich ident sein, das Ergebnisbild überschreibt in diesem Fall das Originalbild.

Die Wahl dieser Ausgangs- und Ergebnisbereiche am VM erfolgt über die Angabe von lokalen Koordinaten, entweder über die Tastatur oder mittels eines Cursors am RGB-Schirm. Beide Möglichkeiten wurden im Rahmen des vorliegenden Programmes realisiert. Zusätzlich wurde eine vereinfachte Bereichsdefinition verwirklicht, die auf einer regelmäßigen Unterteilung des VM basiert. Die 2048 x 2048 Pixel große Matrix läßt sich in 16 Quadrate mit einer Ausdehnung von je 512 x 512 Pixel aufteilen, die als Quadranten bezeichnet und über einen Wert zwischen 0 und 15 angesprochen werden (siehe Abb. 2.14).

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

**Abb. 2.14: Quadrantenschema des VideoRAM**

Die Koordinaten des linken oberen Eckpunktes der Quadranten lassen sich einfach aus der numerischen Bezeichnung der Quadranten ableiten:

$$\begin{aligned}
 x &= (q \text{ MOD } 4) * 512 && (\text{MOD} = \text{Modulo}) \\
 y &= (q \text{ DIV } 4) * 512 && (\text{DIV} = \text{Integer Division})
 \end{aligned}
 \tag{2.3}$$

ausgehend von einem Koordinatensystem, das seinen Ursprung im linken, oberen Eckpunkt des VM hat, dessen x-Achse zum rechten, oberen und dessen y-Achse zum linken, unteren Eckpunkt zeigt.

Das Konzept der Quadranten zieht sich durch die gesamte entwickelte Software. Bei allen Operationen, die Manipulationen am VideoRM durchführen, kann der Zugriff über Quadranten gesteuert werden. Besonders interessant wird dieses Konzept in Zusammenhang mit einem Programm zur Visualisierung mehrerer Bilder am RGB-Schirm. Berechnet man durch Variation der Parameter eines Algorithmus aus einem Bild mit 512 x 512 Pixel 15 Folgebilder, die auf den Quadranten 1 bis 15 abgelegt werden, so können die Ergebnisse nicht gemeinsam am Schirm dargestellt werden. Außerdem ist es nahezu unmöglich, Bilder, die mehr als einen Quadranten voneinander entfernt sind, visuell zu vergleichen. Um dieses

Problem zu umgehen, bietet das Programm die Möglichkeit, wahlweise einen Quadranten am Bildschirm darzustellen und zwischen den Quadranten zu "springen" (*toggeln*). Der Wechsel zwischen den Quadranten erfolgt durch Eingabe einer Taste am Keyboard. Weiters kann ein vergrößerter Ausschnitt eines Quadranten dargestellt werden, wobei beim Wechsel die Position des Ausschnittes innerhalb des Quadranten gleich bleibt. Zusätzlich wird zu jedem Quadranten auch die dazugehörige *look-up-table* abgelegt und bei Wahl des Quadranten aktiviert. Dank der Hardware-Implementierung von *zoom*, *pan* und der *look-up-tables* erfolgt das *toggeln* zwischen den Quadranten in *real-time*<sup>7</sup>.

### 2.3.4 DISK-TO-DISK VERARBEITUNG

#### **Master-Slave-Modell**

Neben der Möglichkeit Bilddaten sozusagen *on-screen* zu manipulieren, besteht auch die Notwendigkeit, Bilder größer als 2048 x 2048 Pixel *disk-to-disk* zu verarbeiten. Der Ablauf einer solchen Verarbeitung ist ein iterativer Vorgang. Ein Teil des Bildes wird von der Harddisk ins *memory* gelesen, dort verarbeitet und zurück auf die Harddisk geschrieben. Danach folgt der nächste Teil des Bildes usw., bis das ganze Bild verarbeitet ist. Die Größe des bei einem Schritt zu bearbeitenden Bildteiles hängt vom verfügbaren Platz im *memory* ab. Der Einsatz eines Multiprozessorrechners erhöht den Datendurchsatz solcher Operationen, indem mehrere Bildteile parallel verarbeitet werden. Ein Konzept, das diese Aufgabenstellung realisiert, ist das *master-slave*-Modell. Dabei übernimmt ein Prozessor - *master* - die Verwaltung und beschickt die anderen Prozessoren - *slaves* - mit Aufgaben. Hat ein *slaves* seine Arbeit erfüllt, signalisiert er dieses dem *master*, der ihm darauf die Ergebnisse abnimmt und eine neue Aufgabe zuteilt. Die zugeteilten Aufgaben können sowohl *tasks* mit den dazugehörigen Daten als auch nur Daten sein.

Das System 2400 erfüllt in seiner Hardwarekonzeption die Anforderungen an ein *master-slave*-Modell. Als *master* bietet sich der Controller an, als *slaves* die acht Transputer der WU. Da es vorteilhaft ist, wenn der Master zu jedem *slave* eine direkte Datenverbindung besitzt, der Controller aber nur mit dem ersten Transputer der WU über einen *link* verbunden ist, wird wiederum der Linkswitch zur Datenübertragung eingesetzt [Hill 1987]. Für die Bearbeitung von Bilddaten sieht das Konzept dann folgendermaßen aus: Der Root liest die Bildteile

---

<sup>7</sup> Das Konzept der Quadranten und des *real-time Toggelns* mag nicht die neueste Erkenntnis bei der Visualisierung von Bilddaten sein, allerdings hat sich die Methode beim praktischen Einsatz sehr bewährt. Darüber hinaus ist dem Autor kein System bekannt, das diese *Features* in ähnlicher Art und Weise anbietet.

von der Harddisk und schickt sie an den Controller, der sie an die Transputer der WU verteilt. Sobald ein *slave* mit seinem Teil fertig ist, schickt er die Ergebnisse an den Controller, der diese an den Root weiterleitet. Der *slave* erhält einen neuen Bildteil zur Bearbeitung.

Entscheidend bei der Erstellung eines solchen Modelles ist eine effiziente und zuverlässige Organisation der Kommunikation, da es sonst sehr leicht zu unnötigen Verzögerungen oder *deadlocks* kommen kann. In diesem Zusammenhang stehen beim System 2400 zwei Fragestellungen im Vordergrund:

- Der Zugriff auf die Harddisk ist im Vergleich zur Datenübertragung über die *links* langsam und stellt somit einen Flaschenhals dar, und
- die Verteilung der Datenpakete auf die *slaves* soll nach dem Prinzip *first come, first serve* erfolgen; für zeitlich naheliegende Anfragen muß daher eine Warteschlange implementiert werden.

Dem Problem des langsamen Zugriffes auf die Harddisk kann in zweierlei Hinsicht Rechnung getragen werden. Zum einen wird der Root aller Verwaltungsaufgaben enthoben und kann seine Tätigkeit auf die I/O Operationen konzentrieren. Die Verwaltung der Kommunikation obliegt allein dem Controller. Um die Menge der zu lesenden und schreibenden Daten zu reduzieren und damit den Durchsatz zu erhöhen, bieten sich Datenkompressionsmethoden an. Implementiert man einen solchen Algorithmus am Controller, so kann dieser einen Datensatz dekomprimieren, während der Root bereits den nächsten Datensatz liest und vice versa. Dieser Ansatz ist sinnvoll, solange der Komprimierungs/Dekomprimierungsprozeß nicht mehr Zeit in Anspruch nimmt, als der Lese/Schreibprozeß des Root. Es ist dementsprechend ein simpler Kompressionsalgorithmus gefragt, dessen Stärke nicht in der optimalen Datenreduktion sondern in minimaler Laufzeit liegen soll. Das folgende Kapitel beschreibt Methoden und Einsatz solcher Kompressionsalgorithmen.

## **Datenkompression**

Ganz allgemein bilden Kompressionsalgorithmen Folgen von Eingabezeichen auf kürzere Folgen von Ausgabezeichen ab, wobei die Eingabefolge aus der Ausgabefolge wieder rekonstruierbar sein muß. Häufig zur Anwendung kommende Verfahren sind die Huffman Kodierung und die Lempel-Ziv-Welch Methode. Erstere basiert auf der Häufigkeit der vorkommenden Zeichen - häufig auftretende Zeichen werden mit kürzeren Bitfolgen kodiert als selten auftretende Zeichen. Die zweite Methode kodiert Zeichenfolgen, die bei wiederholtem Auftreten durch einen Rückverweis repräsentiert und dementsprechend nicht noch einmal kodiert werden. Auch die Kombination beider Verfahren wird eingesetzt und führt zu sehr

guten Kompressionsraten (z.B. LHARC oder PKZIP) [vgl. Bauernöppel, 1991]. Eine ausführliche Beschreibung der genannten Methoden inklusive Angaben zur Effizienz und Stabilität der Verfahren findet sich in Lelewer und Hirschberg [1987]. Die Implementierungen dieser Verfahren haben einige Jahre Entwicklungszeit hinter sich, sind zumeist in Assemblercode in Kombination mit einer Hochsprache programmiert und zielen auf optimale Komprimierung beliebiger Datensätze, d.h. sie sind nicht zeitoptimiert [vgl. zur Nedden 1992].

Im Gegensatz zu den oben genannten Verfahren kann bei der Komprimierung von Bilddaten von einer Systematik der Dateninhalte ausgegangen werden. In Anwendungsbereichen, bei denen der Schwerpunkt auf der Visualisierung von Bilddaten liegt, kann eine Reduktion der Datenmenge durch Elimination irrelevanter Information erreicht werden. Als Beispiel sei die Diskrete Kosinus Transformation genannt, bei der die statistisch voneinander abhängigen Bildpunkte in voneinander möglichst unabhängige Koeffizienten transformiert werden. Die Datenreduktion wird durch das Weglassen von Koeffizienten erreicht, die keine relevante Information beinhalten. Natürlich kann bei einer Rücktransformation nicht mehr der originale Bildinhalt rekonstruiert werden, was jedoch für Visualisierungszwecke nicht relevant ist. Der Informationsverlust durch die Kompression ist für den Betrachter kaum bis gar nicht erkennbar [vgl. Hoffmann 1991].

Die numerische Verarbeitung von Bilddaten, wie sie in der quantitativen Bildverarbeitung auftritt, erfordert jedoch eine vollständige Rekonstruktion des Originals. Bei Fernerkundungsbildern kann man davon ausgehen, daß räumlich benachbarte Grauwerte meist nicht stark variieren. Diese Eigenschaft läßt sich effizient umsetzen, indem nur mehr die Differenzen zwischen den Grauwerten kodiert werden. Da diese Differenzen innerhalb eines geringeren Wertebereichs als die Grauwerte selber liegen, benötigt man weniger Platz für ihre Kodierung [Gonzales und Wintz, 1987]. Geht man von einem Bild der Größe  $I \times J$  aus, dessen Pixel durch je ein Byte (= 256 Grauwerte) repräsentiert sind, so beträgt der benötigte Speicherplatz  $I \times J$  Byte (bei Vernachlässigung eines *headers*). Beträgt die durchschnittliche Differenz zwischen benachbarten Grauwerten weniger als 64, so lassen sich die Differenzen mit durchschnittlich 6 Bit darstellen, was eine Reduktion des Datenvolumens von cirka 25% zur Folge hat.

Kodiert man ein Bild mittels dieses Differenzenverfahrens zeilenweise, so erhält man beim Übergang vom Ende einer Zeile zum Anfang der nächsten eine Sprungstelle, da die beiden Pixel nicht räumlich benachbart sind. Außerdem wird die räumliche Nachbarschaft innerhalb der Spalten nicht ausgenützt. Dieser Nachteil kann umgangen werden, indem man das Bild in kleine Quadrate zerlegt. Innerhalb der Quadrate besteht ein hoher räumlicher Bezug zwischen den Pixeln. Die Kodierung erfolgt schrittweise für jedes Quadrat. Zuerst werden der minimale und der maximale Grauwert innerhalb des Quadrats bestimmt. Die Differenz zwischen

den Extremen bestimmt die Anzahl der Bits, die für die Kodierung notwendig sind. Der minimale Grauwert und die Kodierungstiefe ( $k$ ) werden als *header* gespeichert. Sodann wird die Differenz der einzelnen Grauwerte zum Minimum berechnet und kodiert. Geht man von einem Quadrat der Größe  $N \times N$  aus, innerhalb dessen eine maximale Grauwertdifferenz von 15 auftritt, so genügen 4 Bit zur Kodierung der einzelnen Differenzen. Der benötigte Speicherplatz beträgt dementsprechend zwei Byte für den *header* plus  $(N \times N) / 2$  Bytes für die kodierten Grauwertdifferenzen. Die Effizienz ( $e$ ) dieses Komprimierungsverfahrens läßt sich allgemein für ein Quadrat folgendermaßen darstellen:

$$e = \frac{k}{8} + \frac{2}{N \cdot N} \quad (2.4)$$

Es ist klar erkennbar, daß bei voller Kodierung ( $k = 8$ ) das Datenvolumen die Originalgröße übersteigt. Dieser Fall tritt erfahrungsgemäß bei Fernerkundungsdaten kaum auf. Bei einer Größe des Quadrates von  $8 \times 8$  Pixel genügt jedoch schon eine Kodierungstiefe von 7, um eine, wenn auch geringe Kompressionsrate (ca. 10%) zu erreichen. Empirische Tests mit Satellitenbildern der Systeme Landsat TM und Spot ergaben eine Kompressionsrate zwischen 20 und 30%. Ein Vergleich mit dem Kompressionsprogramm LHarc auf einem Intel 80486 ergab etwas schlechtere Komprimierungsraten für das hier vorgestellte Differenzenverfahren (ca. -10%), dafür aber eine deutliche Verringerung der Laufzeit.

Um die Effizienz des Differenzenverfahrens zu testen, wurde der Algorithmus in ein bestehendes Programm implementiert. Dieses Programm (*read*) hat die simple Aufgabe, Bilddaten von der Harddisk zu lesen und auf dem VM darzustellen. Wie bereits erwähnt, ist der Zugriff auf die Harddisk bedeutend langsamer als die Übertragung der Daten über die Transputerlinks. Die Laufzeit des Leseprogramms hängt also in erster Linie von der gelesenen Datenmenge ab. Im Falle der komprimierten Daten reduziert sich der Leseaufwand um den Faktor der Kompressionsrate. Die Dekomprimierung, die im Controller erfolgt, parallel zur Lese-prozedur des Root, ist schnell genug, um den Gesamtprozeß nicht zu bremsen. Der erhoffte Effekt der Beschleunigung des I/O wurde damit in vollem Ausmaß erreicht.

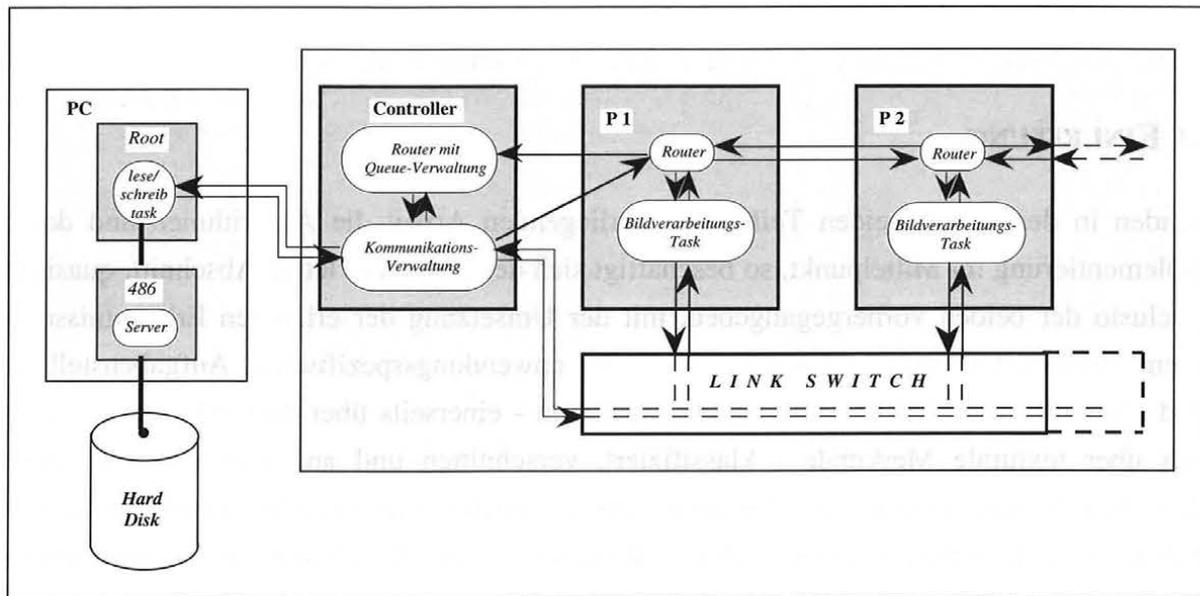
### **Implementierung des *Master-Slave-Modells***

Nach diesem Exkurs über Datenkompression zurück zur ursprünglichen Problemstellung, der Implementierung des *master-slave-Modells* auf dem System 2400. Die Aufgaben des Root und der WU sind aus oben genannten Gründen klar definiert. Der Root übernimmt den Zugriff auf die Harddisk und schickt die Daten zum bzw. empfängt die Daten vom Controller. Die Transputer der WU empfangen Datenpakete vom Controller, führen die Bildverarbei-

tungsoperationen aus und senden die verarbeiteten Daten zurück zum Controller. Die gesamte Verwaltung des Datenaustausches obliegt dem Controller. Zu diesem Zweck sind mehrere parallele Prozesse nötig.

Für die Kommunikation zwischen Controller und Root existieren nur zwei Fälle. Entweder soll ein neues Datenpaket empfangen oder ein verarbeitetes gesendet werden. In beiden Fällen sendet der Controller einen *request* an den Root, dem die gewünschte Aktion folgt.

Die Kommunikation mit den Prozessoren der WU ist komplexer. Als Ausgangsbasis für die folgenden Erläuterungen stelle man sich einen Zeitpunkt während der Verarbeitung vor - alle *slaves* sind mit der Bearbeitung ihrer Daten beschäftigt, der Controller ist in einem Wartezustand. Nun beendet einer der *slaves* seine Arbeit und will dem Controller dieses mitteilen. Über den Linkswitch ist das nicht möglich, da der Controller immer nur mit einem der *slaves* in Verbindung stehen kann. Er weiß allerdings nicht, welcher *slave* als nächster fertig ist. Infolgedessen muß der *slave* die Nachricht über die *pipeline* senden, also durch alle Transputer, die zwischen ihm und dem Controller liegen. Dieses Weitersenden von Nachrichten ermöglicht ein *router*, ein *task* der parallel zum Bildverarbeitungstask auf jedem *slave* läuft [vgl. Mattos 1987]. Dieser *task* macht nichts anderes als darauf zu warten, ob über einen *link* eine Nachricht eintrifft, die er sofort an den nächsten Prozessor weitergibt. Da diese Nachrichten sehr kurz sind, erfordert das Empfangen und Senden einen sehr geringen Aufwand, sodaß die eigentliche Verarbeitung davon nur vernachlässigbar beeinträchtigt wird. Die Nachricht (*request*) erreicht also über die *pipeline* den Controller, der daraufhin den Linkswitch so konfiguriert, daß eine Verbindung zum Absender entsteht. Um dem Controller mitzuteilen, welcher *slave* der Absender ist, besteht der *request* aus der Prozessornummer des Absenders. Sobald die Verbindung besteht, sendet der Controller wiederum über die *pipeline* den *request* zurück an den Absender, der daraufhin mit der Übertragung der verarbeiteten Daten beginnt. Während dieser Datentransfer läuft, könnte ein zweiter *slave* seine Aufgabe beendet haben und seinerseits einen *request* an den Controller senden. Da der Controller jedoch mit dem ersten *slave* beschäftigt ist, kann er den neuen *request* nicht sofort erfüllen. Allerdings muß dieser erhalten bleiben, damit der Controller nach Beendigung der Kommunikation mit dem ersten *slave* den nächsten *slave* versorgen kann. Dafür sorgt ein *routing-task*, der parallel zum Haupttask auf dem Controller läuft. Dieser *router* verwaltet eine Warteschlange (*queue*), die die eingehenden *requests* der Reihe nach speichert und nach Bedarf dem Haupttask weitergibt. Ist der Haupttask mit der Datenübertragung zu einem *slave* fertig, holt er sich den nächsten *request* aus der *queue* und beginnt mit dem Aufbau der neuen Kommunikation. Abb. 2.15 zeigt eine schematische Darstellung der im letzten Absatz beschriebenen Prozesse.



**Abb. 2.15: master-slave Implementierung**

Natürlich stellt die Implementierung noch weitere Anforderungen an dieses Schema, wie z.B. *start-up* Prozeduren oder die Verteilung der Überlappungsbereiche, deren Beschreibung aber den Rahmen der vorliegenden Arbeit sprengen würden.

An die Stelle "*Bildverarbeitungstask*" in Abb. 2.15 können wie beim *screen-to-screen* Programm wiederum beliebige Nachbarschaftsoperationen implementiert werden. Da der Schleifenaufbau des bewegten Fensters bei beiden Programmen ident ist (vgl. Kap. 2.3.2.3), genügt es, die Bildverarbeitungsprozeduren einmal zu entwickeln; sie können in identer Form in beide Programme implementiert werden.

### 2.3.5 EINSATZ IN DER BILDVERARBEITUNG

Neben Standardprozeduren wie Faltung oder Medianfilter wurden die in Abschnitt 1 diskutierten Verfahren zur Texturanalyse und Postklassifikation in die oben beschriebenen Rahmenprogramme implementiert. Nach Austesten der Datenverwaltung und Kommunikation verlief die Implementierung problemlos. Die Programme sind stabil und wurden im Zuge von Lehrveranstaltungen und bei mehreren Applikationen verwendet [Aimet, 1992, Kraus et al. 1992, Steinnocher et al., 1993]. Auch bei der im folgenden Abschnitt beschriebenen Fallstudie kamen die Programme zum Einsatz.

## **3. FALLSTUDIE AGGLOMERATIONSRAUM WIEN**

### **3.1 EINLEITUNG**

Standen in den ersten beiden Teilen der vorliegenden Arbeit die Algorithmen und deren Implementierung im Mittelpunkt, so beschäftigt sich der dritte und letzte Abschnitt, quasi als conclusio der beiden vorhergegangenen, mit der Umsetzung der erlangten Erkenntnisse in einem konkreten Projekt. Ausgehend von einer anwendungsspezifischen Aufgabenstellung wird ein multisensoraler Fernerkundungs-Datensatz - einerseits über spektrale und andererseits über texturale Merkmale - klassifiziert, verschnitten und anschließend einer Postklassifikation unterworfen. Als Alternativmethode erfolgt eine visuelle Interpretation von Orthophotos desselben Gebietes, deren Ergebnisse den Resultaten der Satellitenbildklassifikation gegenübergestellt werden. Zusätzlich ermöglicht ein Netz von detailliert untersuchten Referenzpunkten eine Beurteilung der eingesetzten Methoden.

### **3.2 AUFGABENSTELLUNG**

Seit Mitte 1991 besteht eine Kooperation zwischen dem I.P.F. und dem Institut für Nachrichtentechnik und Hochfrequenztechnik der Technischen Universität Wien mit dem Ziel, Landnutzungsklassifikationen städtischer Gebiete und Analysen der Verbauungsdichte auf der Basis von Fernerkundungsdaten durchzuführen. Das Institut für Nachrichtentechnik und Hochfrequenztechnik benötigt zur Planung und Modellierung zellulärer Mobilfunknetze - zusätzlich zu einem digitalen Höhenmodell - flächendeckende, aktuelle digitale Daten über die Art der Landnutzung bzw. Oberflächenbedeckung, da diese beiden Komponenten die Ausbreitung von Funkwellen wesentlich beeinflussen [Bühler, 1990].

#### **3.2.1 ZELLULARER MOBILFUNK**

Die ständig steigende Nachfrage der Konsumenten nach mobiler Telekommunikation zwingt die Netzbetreiber, die aufgrund technischer Parameter limitierte Anzahl der verfügbaren Funkkanäle effizient zu nutzen und die Basisstationen der einzelnen Versorgungsbereiche optimal im Raum zu plazieren. Diese Anzahl ist ausschlaggebend für die Gesamtkapazität des Systems, denn durch sie kann in einem definierten Bereich eine größere Anzahl von Teilnehmern funktechnisch versorgt werden. Deshalb wird das zu versorgende Gebiet - funktech-

nisch gesehen - in kleinere abgegrenzte Gebiete, die Funkzellen, aufgeteilt und durch Frequenzwiederbenutzung die Kapazität gesteigert. Dies bedingt relativ niedere Antennenhöhen an der Basisstation und bewirkt, daß sich die Funkwellen nahe der Erdoberfläche ausbreiten und wegen ihrer geringen Wellenlänge von der Flächennutzung maßgeblich beeinflusst werden.

Zur effizienten Planung von Mobilfunknetzen ist es daher notwendig, die Funkausbreitung in Computermodellen zu simulieren. Dazu benötigt man neben einem digitalen Höhenmodell (DHM) auch digitale Daten über die Oberflächenbedeckung/Landnutzung, die einen möglichst aktuellen Stand ausweisen und von ihrer Auflösung her hinreichend genaue Informationen liefern sollten. Nachrichtentechniker und Funknetzplaner sind daher mit dem Problem konfrontiert, mögliche Datenquellen zu nutzen, die relativ kostengünstig flächendeckend die geforderten digitalen Primärinformationen (Inputdaten) für diese Funkausbreitungsmodelle zur Verfügung zu stellen imstande sind.

Um die benötigten Informationen über die Oberflächenbedeckung zu erhalten, bieten sich Fernerkundungsdaten an. Die Datenakquisition und erste Bearbeitungsschritte (*preprocessing*) werden unabhängig von menschlichen Einflüssen durchgeführt und sind daher nicht - wie kartographische Produkte bzw. Daten-/Informationsquellen - subjektiv geprägt. Sie stellen vielmehr quasi-Primärdaten dar, da die Pixel-Werte eine integrierte Reflexion eines korrespondierenden Ausschnittes der Erdoberfläche widerspiegeln. Es ist wichtig anzumerken, daß diese auflösungsbedingte Integration über die Pixel-Fläche der einzige Transformations- bzw. Aggregationsschritt während der Phase des Datensammelns ist, und somit eine Art "Objektivität" des Ausgangsdatenmaterials vorausgesetzt werden kann.

### **3.2.2 LANDNUTZUNGSKLASSEN**

Die für die Funkwellenausbreitungsmodellierung benötigten Landnutzungsklassen wurden von Seiten der Nachrichtentechniker explizit vorgegeben [vgl. Bühler und Steinnocher 1991]. Sie gehen von keinem spezifischen Modell aus, sondern sind als Grundlage für beliebige Ausbreitungsuntersuchungen konzipiert (siehe Tab. 3.1). Besondere Aufmerksamkeit wurde den Klassen im urbanen Raum gewidmet, da verbaute Gebiete naheliegenderweise einen starken Einfluß auf die Funkwellenausbreitung aufweisen. Dabei werden zwei Faktoren zur Beschreibung des Verbauungsgrades herangezogen. Zum einen ist das die Verbauungsdichte  $D$ , die den prozentuellen Anteil der verbauten Fläche repräsentiert, zum anderen der Verbauungskoeffizient  $H/B$ , der das Verhältnis zwischen charakteristischer Gebäudehöhe und Gebäudeabstand beschreibt. Daraus folgen die in Tab. 3.1 definierten Stadtklassen, die in der

Folge als "Stadt dicht" (Klasse 2), "Stadt mittel" (Klasse 3) und "Stadt locker" (Klasse 4) bezeichnet werden. Die Klassen 11, 12 und 13 treten innerhalb des Untersuchungsgebietes nicht auf.

**Tab. 3.1: Landnutzungsklassen**

Klasse	Beschreibung
1	Wald
2	Stadt dicht verbaut <span style="float: right;">D&gt;85% H/B&gt;1</span>
3	Stadt, Kleinstadt, Vorstadt <span style="float: right;">D&gt;60% H/B≅1</span>
4	Vorstadt, Stadtrandsiedl., Dorf, Villenviertel <span style="float: right;">D&gt;40% H/B&lt;1</span>
5	quasi offen, Park (Wiese mit wenigen Baumgruppen)
6	offen (Wiese, Felder, Weingärten)
7	Gewässer
10	versiegelt, Schotterflächen
11	Fels
12	Alpine Matten
13	Eis, Firn, Gletscher
15	undefiniert

### 3.2.3 UNTERSUCHUNGSGEBIET UND DATENQUELLEN

Als Untersuchungsgebiet wurde der Agglomerationsraum Wien gewählt. Zum einen, da er aufgrund der unterschiedlichen Verbauungsformen hohe Anforderungen an den Klassifikationsalgorithmus stellt, zum anderen aus Gründen der Datenverfügbarkeit. Die Ortskenntnis, die einen wichtigen Aspekt bei der Verifizierung der eingesetzten Methode darstellt, sprach ebenfalls für Wien.

**Tab. 3.2: Datenquellen**

Datenquelle	Spektralbereich	Umfang	Auflösung
Landsat TM	multispektral	1024 x 1024 pixel	30 x 30 m <sup>2</sup>
SPOT	panchromatisch	800 x 1000 pixel	10x 10 m <sup>2</sup>
Orthophotos	panchromatisch	25 x 50 km <sup>2</sup>	1 : 10000

Für die automatische Klassifizierung standen zwei Fernerkundungsdatensätze zur Verfügung: Ein Ausschnitt aus einer Landsat TM Szene, die den gesamten Großraum Wien abdeckt (ca. 900 km<sup>2</sup>), und eine panchromatische SPOT Aufnahme von ca. 80 km<sup>2</sup> des Untersuchungs-



**SPOT panchromatisch Oktober 1986**

**Untersuchungsgebiet**

**Abb. 3.1: panchromatische SPOT-Aufnahme des Testgebietes**

raumes. Als alternativer Datensatz für die visuelle Interpretation kamen zwei analoge Orthophotos (ÖLK 7635-100 und 7635-102) im Maßstab 1:10000 zum Einsatz. Sie decken ein Testgebiet von ca. 50 km<sup>2</sup> innerhalb der SPOT Aufnahme ab (siehe Tab. 3.2, Abb. 3.1).

### 3.3 METHODIK

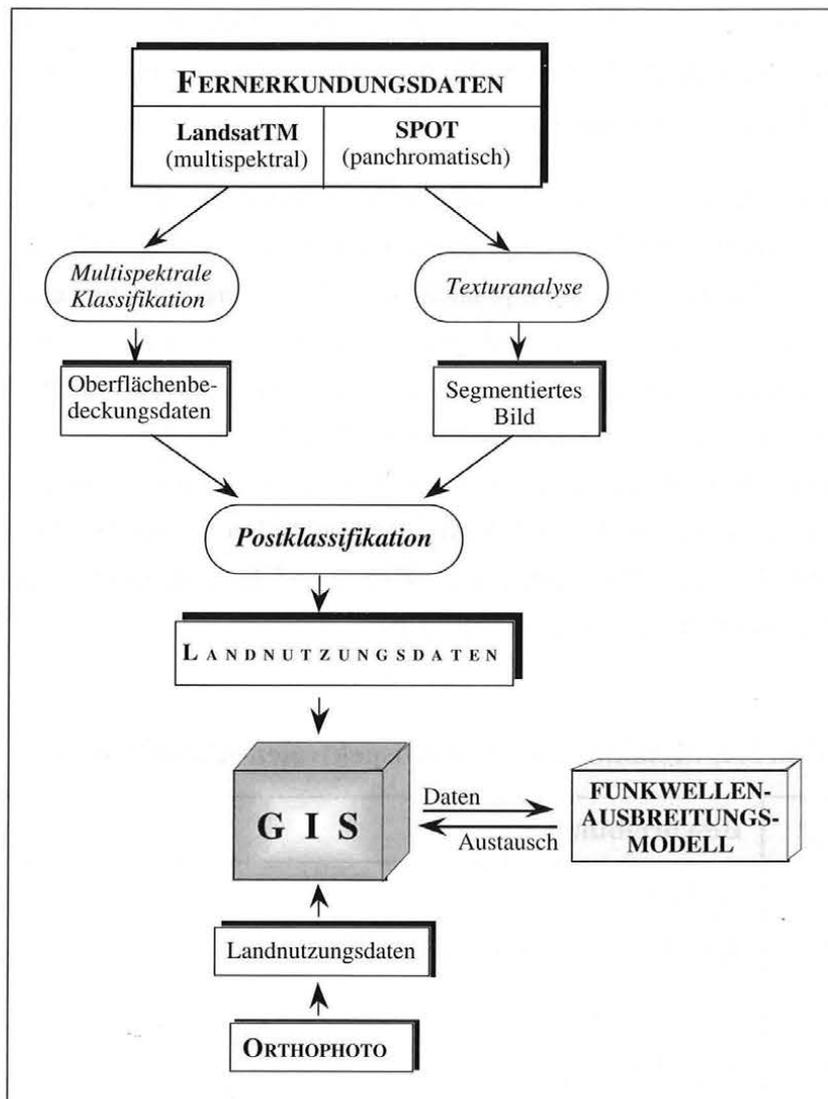
Die in Tabelle 3.1 definierten Landnutzungsklassen lassen sich in zwei Gruppen einteilen. Die erste Gruppe umfaßt Oberflächenbedeckungsarten, die sich direkt aus einer multispektralen Klassifizierung der TM-Daten ableiten lassen, wie z.B. Wald oder offenes Gelände. Die zweite Gruppe enthält Landnutzungen, die sich aus verschiedenen Oberflächenbedeckungen zusammensetzen. Dazu zählen vor allem die urbanen Klassen, aber auch die als quasi offen definierte Klasse. Zur Bestimmung dieser Gruppe ist es notwendig, eine Postklassifikation im Anschluß an die multispektrale Klassifikation durchzuführen (vgl. Kap. 1.6).

Zusätzliche Information über urbane Gebiete liefert die panchromatische SPOT Szene. Die höhere geometrische Auflösung ermöglicht über eine Texturanalyse eine scharfe Abgrenzung von verbautem und unverbautem Gebiet (vgl. Kap. 1.5.1).

Die Verknüpfung des Ergebnisses der multispektralen Klassifikation mit dem der Texturanalyse liefert die Ausgangsdaten für den Postklassifikationsalgorithmus. Die endgültigen Resultate werden zur weiteren Modellierung in ein Geographisches Informationssystem (GIS) übernommen. Abb. 3.2 zeigt ein graphisches Schema der eben beschriebenen Vorgangsweise. Die eingesetzten Hard- und Softwarekomponenten sind in Tab. 3.3 angegeben.

**Tab. 3.3: Hard- und Softwarekomponenten**

<b>Hardware</b>	<b>Software</b>	<b>Anwendung</b>
IMPULS SYSTEM 2400	TIPS	Texturanalyse Postklassifikation
HP 9000/730	ERDAS	multispektrale Klassifikation Rektifizierung
HP 9000/720	ARC/INFO	Analyse der Ergebnisse



**Abb. 3.2: Ablaufschema der Klassifikation**

### 3.3.1 MULTISPEKTRALE KLASSIFIKATION

Die Kanäle 1-5 und 7 der Landsat TM-Szene wurden einer überwachten multispektralen Klassifikation unterworfen. Die Auswahl der Trainingsgebiete erfolgte einerseits unter Einsatz von Ortskenntnis, andererseits, vor allem in agrarisch genutzten Gebieten, über die multi-spektrale Charakteristik der Kanalkombination 3, 4 und 7. Trainiert wurden 36 spektrale Klassen, die im Anschluß an die Klassifikation zu den in Tab. 3.4 angeführten Metaklassen aggregiert wurden. Die Zusammenfassung von Klassen, die sich sowohl spektral

als auch thematisch unterscheiden, erfolgt in Abhängigkeit der aktuellen Applikation. Als Beispiel seien unterschiedliche landwirtschaftliche Nutzungen angeführt, deren Differenzierung für die Aufgabenstellung "Mobildfunk" nicht relevant ist und die daher zur Metaklasse "Feld/Wiese" vereint wurden.

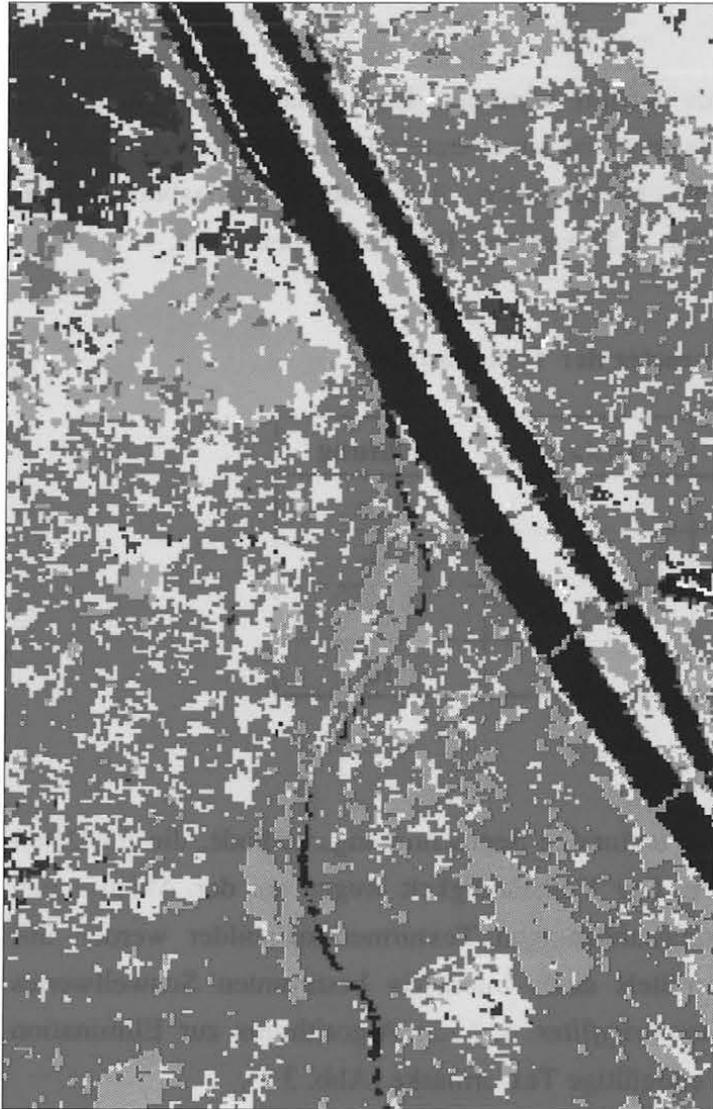
Besonderes Augenmerk sei noch auf die Klasse V - versiegelte Flächen - gerichtet. Die spektrale Charakteristik dieser Klasse zeigt eine hohe Reflexion in allen Kanälen, außer dem Vegetations-Kanal 4. Das bedeutet, sie repräsentiert insbesondere Schotter- und Betonflächen ohne Vegetation, nicht jedoch brachliegende Böden. Als Beispiele für diese Klasse innerhalb des Untersuchungsraumes seien die Baustelle des Kraftwerkes Greiffenstein, Teile der Donauinsel, aber auch alle großen Gleisanlagen und größere Flachdächer, wie sie bei Industriebetrieben oder beim AKH auftreten, genannt. Die Klasse repräsentiert also sowohl verbautes als auch unverbautes Gebiet, wobei auf Basis der spektralen Charakteristik keine weitere Differenzierung möglich ist. Die von dieser Klasse erfaßten Bereiche müssen dementsprechend einer Nachbearbeitung unterworfen werden.

**Tab. 3.4: Metaklassen der multispektralen Klassifikation**

<b>Klasse</b>	<b>Beschreibung</b>
I	Feld / Wiese
II	brachliegende Böden (incl. Weingärten)
III	City-Gebiete
IV	Vorstadt,
V	versiegelte Flächen
VI	Wald
VII	Park
VIII	Wasser
IX	undefiniert

Die Klasse IX - undefiniert - beschreibt Gebiete, die nicht im Sinne der übrigen Klassen interpretiert werden können. Im konkreten Anwendungsfall sind das vor allem Wolken und Wol-kenschatten, die kleine Gebiete im Südwesten des Untersuchungsraumes bedecken.

Das Resultat der multispektralen Klassifikation ist in Abb. 3.3 dargestellt. Eine grobe, visuelle Kontrolle anhand der Orthophotos zeigt ein zufriedenstellendes Ergebnis.



-  **Klasse I: Feld**
-  **Klasse II: Boden**
-  **Klasse III: City Gebiete**
-  **Klasse IV: Vorstadt**
-  **Klasse V: versiegelt**
-  **Klasse VI: Mischwald**
-  **Klasse VII: Laubwald**
-  **Klasse VIII: Wasser**

**Landsat TM Juni 1985  
Multispektrale Klassifikation**

**Abb. 3.3: Ergebnis der multispektralen Klassifikation**

### 3.3.2 TEXTURANALYSE

Die panchromatische SPOT Szene wurde auf ihre texturalen Eigenschaften untersucht. Ziel der Analyse war die Trennung von verbautem und unverbautem Gebiet. Ersteres wird durch richtungsunabhängige hohe Texturierung charakterisiert. Aus dem Ausgangsbild wurden daher Texturmerkmalsbilder mit den in Tab. 3.5 gezeigten Parametern berechnet.

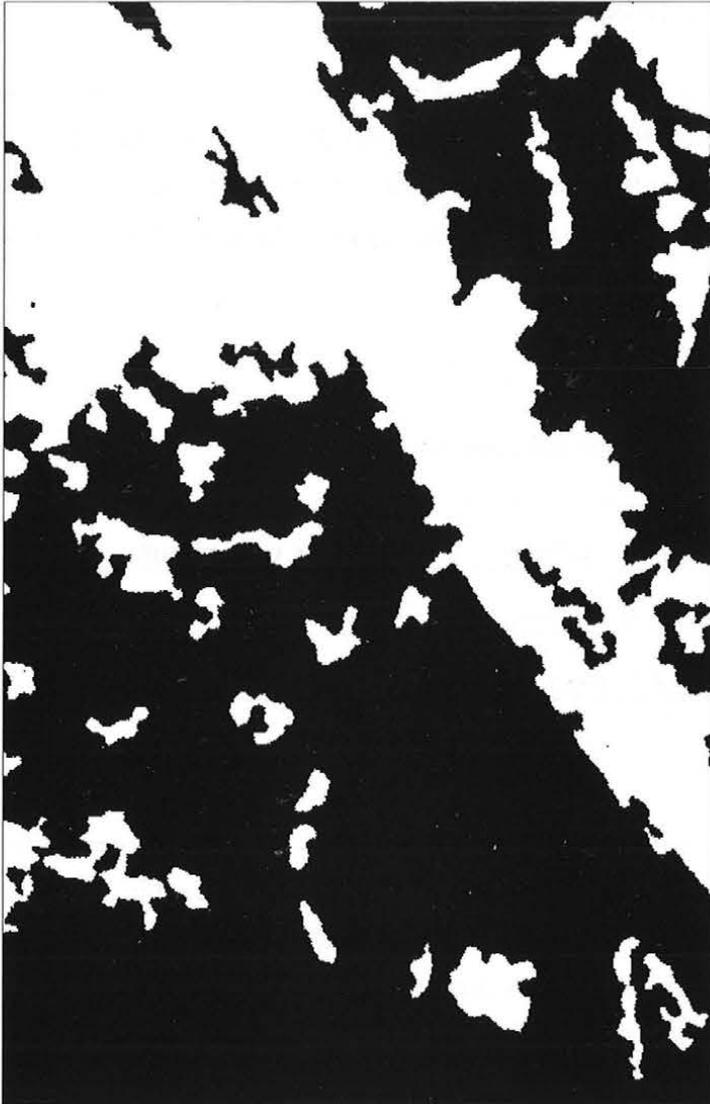
**Tab. 3.5: Parameter der Texturanalyse**

Texturmerkmal	Distanz	Richtung
IDM	2	hor
IDM	2	ver
IDM	2	ld
IDM	2	rd

Die weitere Vorgangsweise entspricht der texturalen Segmentierungsmethode, die im Detail bereits in Kap. 1.5.1 besprochen wurde. Der Vollständigkeit wegen sei der Ablauf noch einmal kurz umrissen. Die vier richtungsabhängigen Texturmerkmalsbilder werden mit einem 9x9 Mittelwertfilter geglättet, mittels eines interaktiv bestimmten Schwellwertes binarisiert und verschnitten. Ein 5x5 *majorityfilter* und ein Algorithmus zur Elimination kleiner Insel-flächen im Bild liefern die endgültige Texturmaske (Abb. 3.4).

### 3.3.3 REKTIFIZIERUNG

Um den räumlichen Bezug der Fernerkundungsdaten herzustellen, muß eine Rektifizierung der Bilddaten durchgeführt werden. Da das Verhältnis der Höhenunterschiede innerhalb des Untersuchungsgebietes zur Flughöhe sehr klein ist ( $< 0.5\%$ ), kann der Einfluß der Topographie vernachlässigt und eine Affintransformation zwischen Bild- und Landeskoordinatensystem angesetzt werden. Die Bestimmung der Landeskoordinaten der Paßpunkte erfolgte über die ÖK 50 (TM) und über die zur Verfügung stehenden Orthophotos (SPOT). Die korrespondierenden Bildkoordinaten wurden jeweils in den Originalbilddaten identifiziert, die Transformation jedoch erst auf die bereits klassifizierten Ergebnisse angewandt. Als *resampling*-Methode kam daher "*Nearest Neighbor*" zur Anwendung. Der mittlere Fehler der Affintransformation lag bei beiden Datensätzen im Subpixelbereich.



**SPOT panchromatisch Oktober 1986**

**Texturmaske**

**Abb. 3.3: Texturmaske**

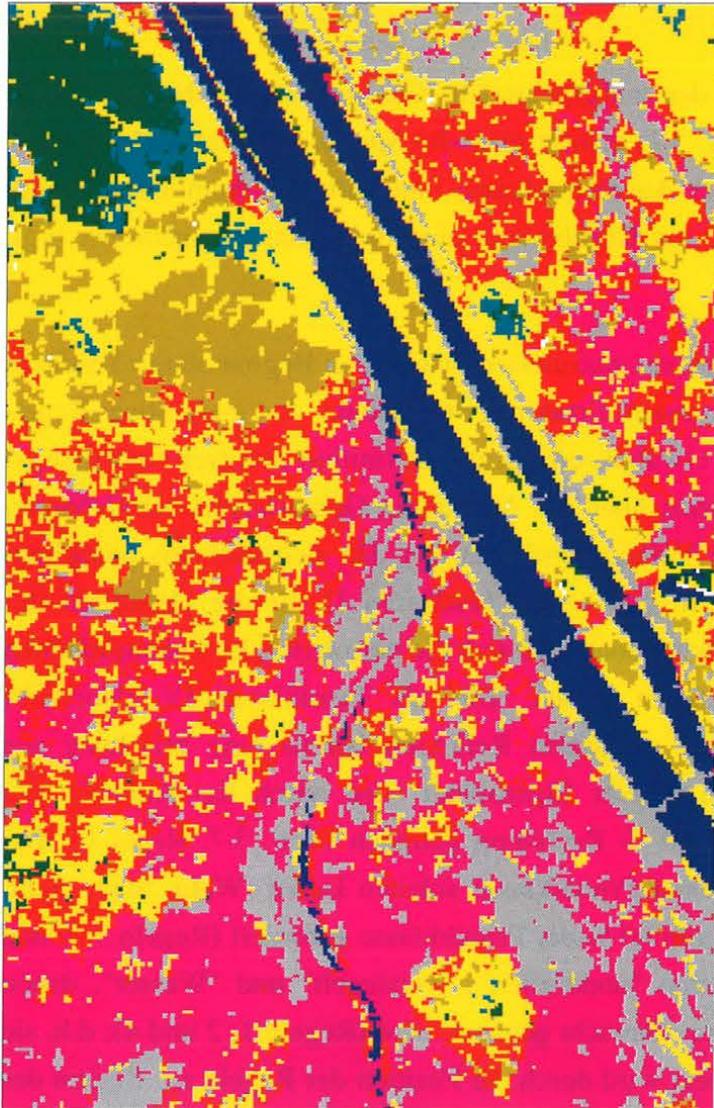
### 3.3.4 VERSCHNEIDUNG DER KLASSIFIKATIONSERGEBNISSE

Die Verschneidung der Texturmaske mit dem Ergebnis der multispektralen Klassifikation erfolgt nach den in Tab. 3.6 dargestellten Regeln. Als Ausgangsgrößen werden die Klassen III (City-Gebiete) und IV (Vorstadt) der multispektralen Klassifikation und die Klassen "verbaut" und "unverbaut" der Texturanalyse herangezogen. Alle übrigen Klassen werden durch die Verschneidung nicht verändert.

**Tab. 3.6: Verschneidungsregeln der Texturmaske**

	<b>Textur verbaut</b>	<b>Textur unverbaut</b>
<b>III City-Gebiete</b>	III City-Gebiete	V versiegelte Flächen
<b>IV Vorstadt</b>	IV Vorstadt	I Feld/Wiese

Wie erwartet bleibt der Großteil der multispektralen Klassifikation unverändert, nur in einzelnen Gebieten bewirkt die Verschneidung Veränderungen (vgl. Abb. 3.5 contra Abb. 3.3). Ein Vergleich der korrigierten Gebiete mit dem Orthophoto zeigt die Plausibilität der Vorgangsweise. Einige agrarisch genutzte Flächen südlich des Kahlenberges, auf Grund ihrer spektralen Ähnlichkeit als Vorstadt klassifiziert, werden über ihre texturale Charakteristik erkannt und ihrer eigentlichen Klasse zugewiesen. Innerhalb der City-Gebiete können die markanten Gleisanlagen des Frachtenbahnhofs Praterstern im 20. Bezirk ausgewiesen werden. Korrigiert wird auch der als Vorstadt klassifizierte Streifen entlang dem südwestlichen Donauufer, der die Autostraße Richtung Klosterneuburg repräsentiert. Die Anzahl der korrigierten Pixel beträgt circa 5 % des Testgebietes. Da diese Gebiete kontrolliert und die Korrekturen als richtig erkannt wurden, bedeutet die Einbeziehung der Texturinformation eine Verbesserung der Klassifikation um etwa 5 %.



-  **Klasse I: Feld**
-  **Klasse II: Boden**
-  **Klasse III: City Gebiete**
-  **Klasse IV: Vorstadt**
-  **Klasse V: versiegelt**
-  **Klasse VI: Mischwald**
-  **Klasse VII: Laubwald**
-  **Klasse VIII: Wasser**

**Landsat TM Juni 1985  
Multispektrale Klassifikation**

**SPOT panchromatisch Oktober 1986  
Texturanalyse**

**Verschneidung**

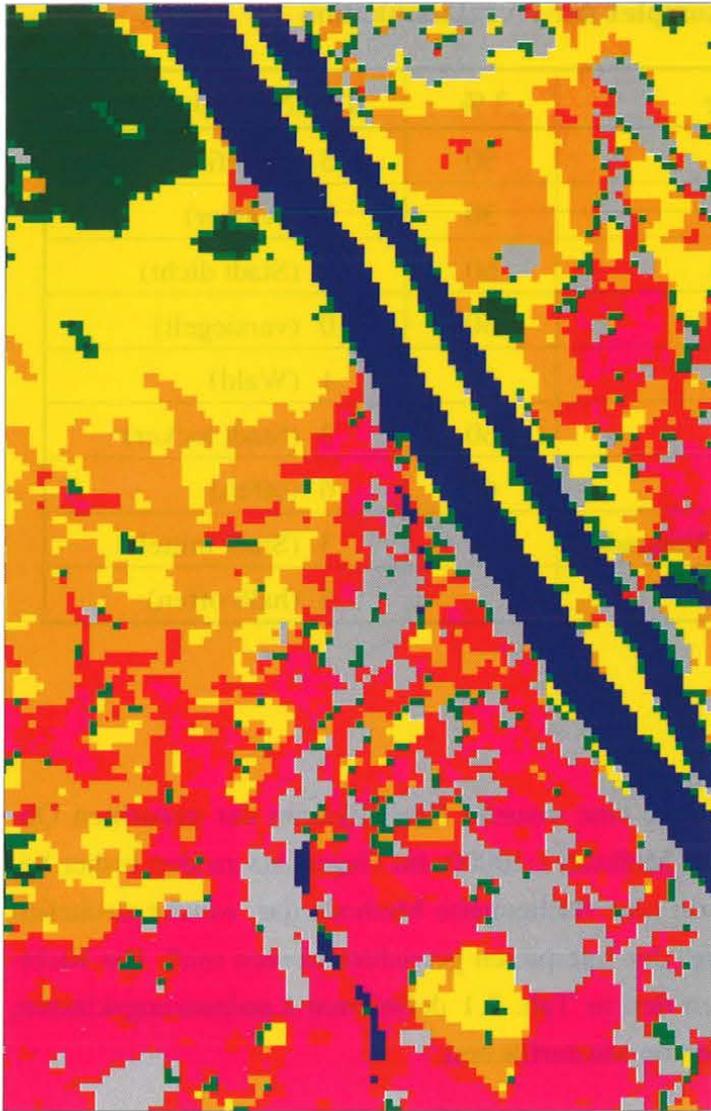
**Abb. 3.5: Ergebnis der Verschneidung von multispektraler Klassifikation und  
Texturmaske**

### 3.3.5 POSTKLASSIFIKATION

Die Transformation der Metaklassen der multispektralen Klassifikation (Tab. 3.4) in die geforderten Landnutzungsklassen (Tab. 3.1) erfolgt über die *Spatial Context Rules Classification*. Dabei wird die Verteilung der Ausgangsklassen innerhalb der räumlichen Umgebung eines Pixels untersucht und über ein Regelsystem eine neue Klasse zugeordnet (vgl. Kap. 1.6.2.1).

Um die Wirkungsweise des Regelkomplexes auszutesten, wurden zu Beginn einige Testläufe durchgeführt. Aus den daraus gewonnenen Erfahrungen entstand der endgültige Regelkomplex (vgl. Tab. 3.7). Wie bereits erwähnt, erfolgt die Definition der drei Verbauungsklassen über zwei Parameter; zum einen über das Verhältnis Gebäudehöhe zu Gebäudeabstand (H/B), zum anderen über die Verbauungsdichte (D). Da sich H/B aus Satellitenbildern nicht bestimmen läßt, wurde in den ersten Testläufen D für die Definition der Stadtklassen herangezogen. Es zeigte sich jedoch, daß sich die definierten Verbauungsdichten nicht direkt in die Regeln übernehmen lassen. Die als Inputklasse IV (Vorstadt) klassifizierten Pixel beinhalten in ihrer spektralen Reflexion bereits eine Mischung aus verbauten und un-verbauten Flächen. Dieser Eigenschaft wird durch das Anheben der geforderten Häufigkeit von 40 % auf 60 % Rechnung getragen (Tab. 3.7, Regel 6). Bei denjenigen Klassen, die sich nur aus einer Primärklasse ableiten lassen (Wald, Stadt dicht, offen), wird eine minimale Häufigkeit von 60 % der Primärklasse gefordert (Regeln 3, 5 und 7). Ausnahmen bilden dabei die Klassen "undefiniert", "versiegelt" und "Wasser", denen durch geringere Prozentwerte ein höheres Gewicht gegeben wird (Regeln 1, 2 und 4); d.h. sie bleiben bevorzugt bestehen. Dieser Effekt wird durch die Position der Regeln am Beginn des Regelkomplexes verstärkt. Die Regel 8 definiert die Klasse "Stadt mittel" als Mischklasse zwischen dichter und lockerer Verbauung. Da es sich um ein hierarchisches Regelsystem handelt, kann diese Regel erst zum Tragen kommen, wenn die beiden anderen "Stadtregeln" (Regel 3 und 6) bereits verworfen wurden. Die Klasse "halb offen" wird all den Bereichen zugeordnet, die keiner anderen Regel genügen (Regel 9).

Die untersuchte Umgebung wurde mit 4x4 Pixeln festgelegt, die auf 2x2 Pixel abgebildet wurden - das entspricht einem Untersuchungsraum von 100x100 m<sup>2</sup> und einem Resampling von 25x25 m<sup>2</sup> auf 50x50 m<sup>2</sup> Pixelgröße in der Natur (Abb. 3.6).



- |   |                               |
|---|-------------------------------|
|    | <b>Klasse 1: Wald</b>         |
|    | <b>Klasse 2: Stadt dicht</b>  |
|    | <b>Klasse 3: Stadt mittel</b> |
|  | <b>Klasse 4: Stadt locker</b> |
|  | <b>Klasse 5: halb offen</b>   |
|  | <b>Klasse 6: offen</b>        |
|  | <b>Klasse 7: Wasser</b>       |
|  | <b>Klasse 10: versiegelt</b>  |

**Landsat TM Juni 1985  
Multispektrale Klassifikation**

**SPOT panchromatisch Oktober 1986  
Texturanalyse**

**Verschneidung**

**Postklassifikation 100x100m**

**Abb. 3.6: Ergebnis der Postklassifikation**

**Tab. 3.7: Regelkomplex der Postklassifikation**

<b>Regel</b>	<b>IF Input-Klasse</b>	<b>3 %</b>	<b>THEN Output-Klasse</b>
1	IX (undefiniert)	50	15 (undefiniert)
2	VIII (Wasser)	30	7 (Wasser)
3	III (City-Gebiete)	60	2 (Stadt dicht)
4	V (versiegelt)	50	10 (versiegelt)
5	VI (Wald), VII (Park)	60	1 (Wald)
6	IV (Vorstadt)	60	3 (Stadt locker)
7	I (Feld), II (Boden)	60	6 (offen)
8	III (City-Gebiete), IV (Vorstadt)	50	3 (Stadt mittel)
9	ELSE		5 (halb offen)

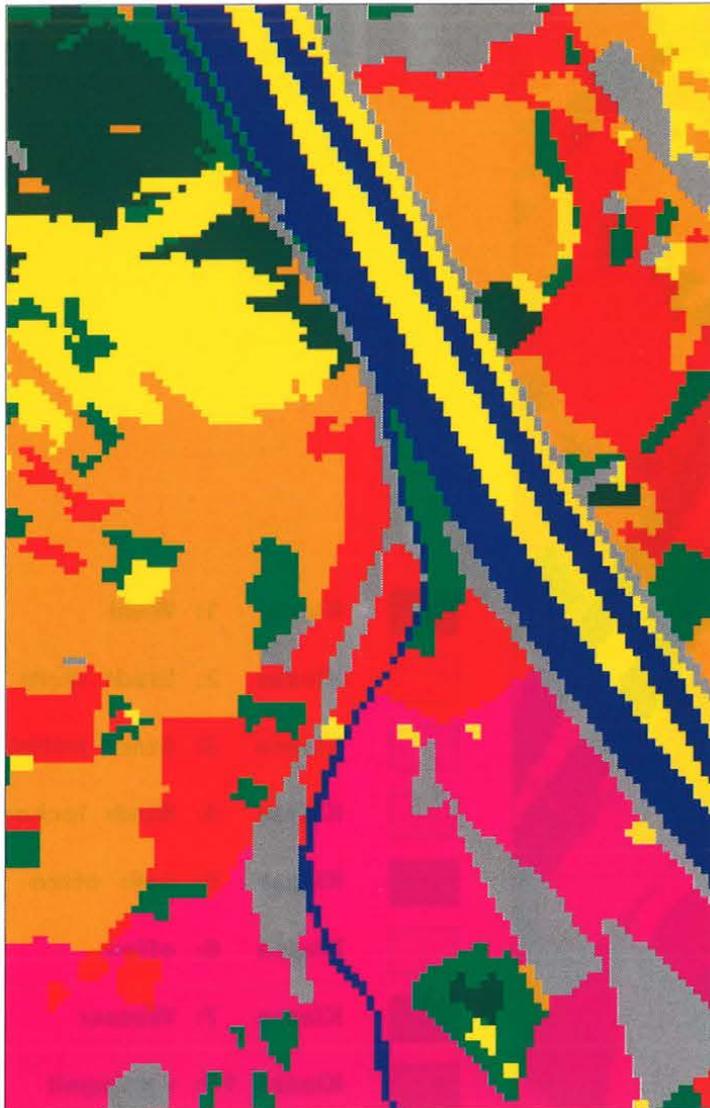
### 3.3.6 VISUELLE KLASSIFIKATION

Parallel zum automatischen Ansatz erfolgte eine visuelle Klassifikation der gesuchten Objektklassen mit Hilfe der Orthophotos im Maßstab 1:10000. Im Gegensatz zu den bisher beschriebenen Verfahren stellt dieser Schritt eine flächenhafte Methode dar, welche zusätzlich unter dem Gesichtspunkt der "Erfahrung" der Interpreten gewichtet werden muß. Die Richtlinien für die Klassifikation entsprachen den in Tab. 3.1 definierten Landnutzungsklassen, ohne weitere Abstimmung oder Vorgabe von Musterflächen.

Die beiden Orthophotos, welche alle gesuchten Objektklassen enthalten, wurden von 5 Testpersonen unabhängig klassifiziert. Die Begrenzungspolygone wurden digitalisiert und in das GIS ARC/INFO übernommen, bereinigt und topologisch aufbereitet. Anschließend erfolgte die Zuweisung der Objektklassen zu den einzelnen Polygonen. Um einen Vergleich mit der automatischen Klassifizierung zu ermöglichen, wurden die einzelnen Klassifikationsergebnisse einer Vektor-Raster Konvertierung unterworfen (Abb. 3.7 und 3.8).

### 3.4 ANALYSE DER ERGEBNISSE

Als zu bewertende Ergebnisse werden die Postklassifikation und zwei visuelle Klassifikationen verschiedener Interpreten herangezogen. Aus Gründen der Verfügbarkeit von Analysewerkzeugen wurden diese *Layer* in das Raster-GIS GRID von ARC/INFO übernommen.



-  Klasse 1: Wald
-  Klasse 2: Stadt dicht
-  Klasse 3: Stadt mittel
-  Klasse 4: Stadt locker
-  Klasse 5: halb offen
-  Klasse 6: offen
-  Klasse 7: Wasser
-  Klasse 10: versiegelt

**Orthophoto 1 : 10 000**

**Visuelle Klassifikation 1**

**Abb. 3.7: Visuelle Interpretation**



-  Klasse 1: Wald
-  Klasse 2: Stadt dicht
-  Klasse 3: Stadt mittel
-  Klasse 4: Stadt locker
-  Klasse 5: halb offen
-  Klasse 6: offen
-  Klasse 7: Wasser
-  Klasse 10: versiegelt

**Orthophoto 1 : 10 000**

**Visuelle Klassifikation 2**

**Abb. 3.8: Visuelle Interpretation**

### 3.4.1 VERGLEICH DER METHODEN

Ein visueller Vergleich der drei Ergebnislayer zeigt bereits deutlich die unterschiedliche Methodik der Klassifikationen. Die visuelle Interpretation tendiert zu scheinbar scharf definierten, großflächigen Gebieten einer Objektklasse, innerhalb derer jedoch stark generalisiert wird (vgl. Abb. 3.7 und 3.8). Die automatische Klassifikation geht auf Grund ihrer Pixelorientierung auf lokale Klassenunterschiede ein, wodurch das Ergebnisbild einen unruhigen Charakter bekommt (vgl. Abb. 3.6).

Die Unterschiede zwischen den Klassifikationsergebnissen liegen aber nicht nur in der räumlichen Verteilung der zugewiesenen Klassen sondern auch in der Häufigkeit des Auftretens der einzelnen Klassen. Tab. 3.8 zeigt diese Häufigkeit in Prozent des untersuchten Testgebietes.

Lediglich die Klassen 1 (Wald) und 7 (Wasser) zeigen nahezu idente Häufigkeiten in allen drei Klassifikationen. In den übrigen Klassen weicht jeweils eine Klassifikation deutlich von den anderen beiden ab, wobei die Ausreißer sowohl im Ergebnis der visuellen als auch der automatischen Methode zu finden sind.

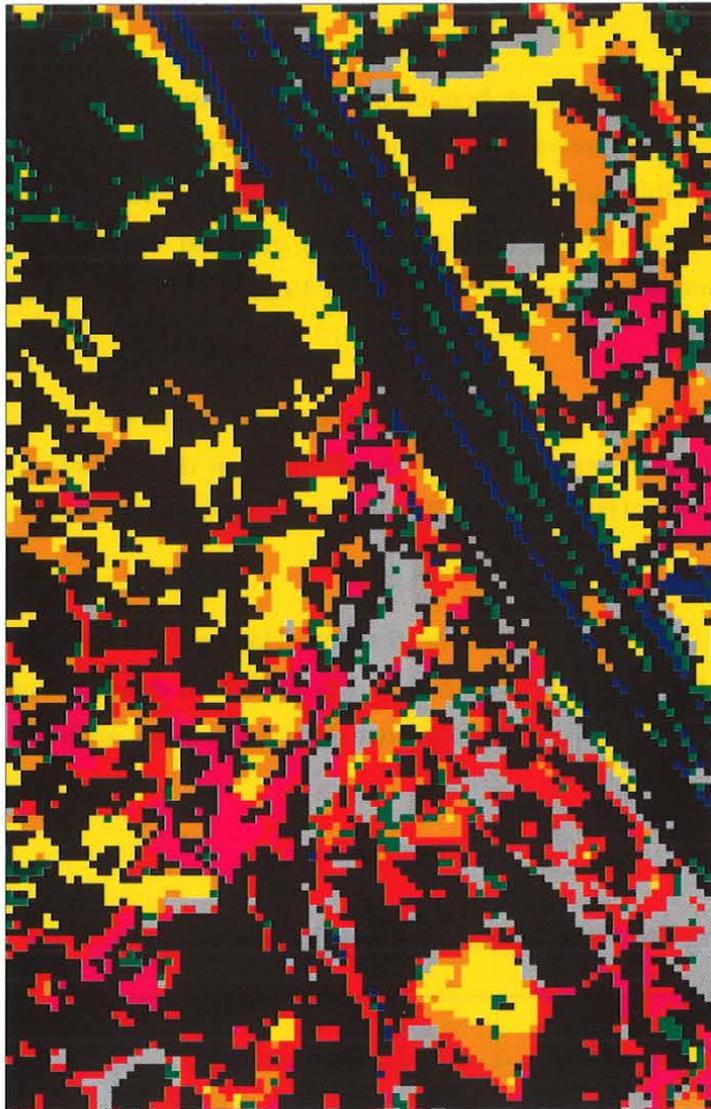
**Tab. 3.8: Häufigkeitsverteilung der Objektklassen in den Ergebnislayern**

Objektklasse	1	2	3	4	5	6	7	10
visuell I	4.8	33.8	12.1	15.4	7.7	11.7	7.4	7.2
visuell II	5.2	33.4	<b>20.7</b>	<b>7.3</b>	7.4	10.7	7.1	8.1
automatisch	4.0	<b>23.8</b>	13.8	13.6	<b>4.2</b>	<b>18.8</b>	7.6	<b>14.4</b>

Häufigkeiten in %  
signifikante Abweichungen **fett**

Als nächster Schritt wurde die räumliche Verteilung der Unterschiede untersucht. Dieser Vergleich erfolgte durch Verschneidung der einzelnen Layer und Erstellung von Differenzbildern, wobei im Falle von Abweichungen jeweils das Ergebnis der ersten Klassifikation dargestellt ist (vgl. Abb. 3.9, 3.10 und 3.11).

Da diese Darstellung zwar die Unterscheidung von ident und nicht ident klassifizierten Bereichen ermöglicht, jedoch keine qualitativen Aussagen über die Unterschiede erlaubt, wurde



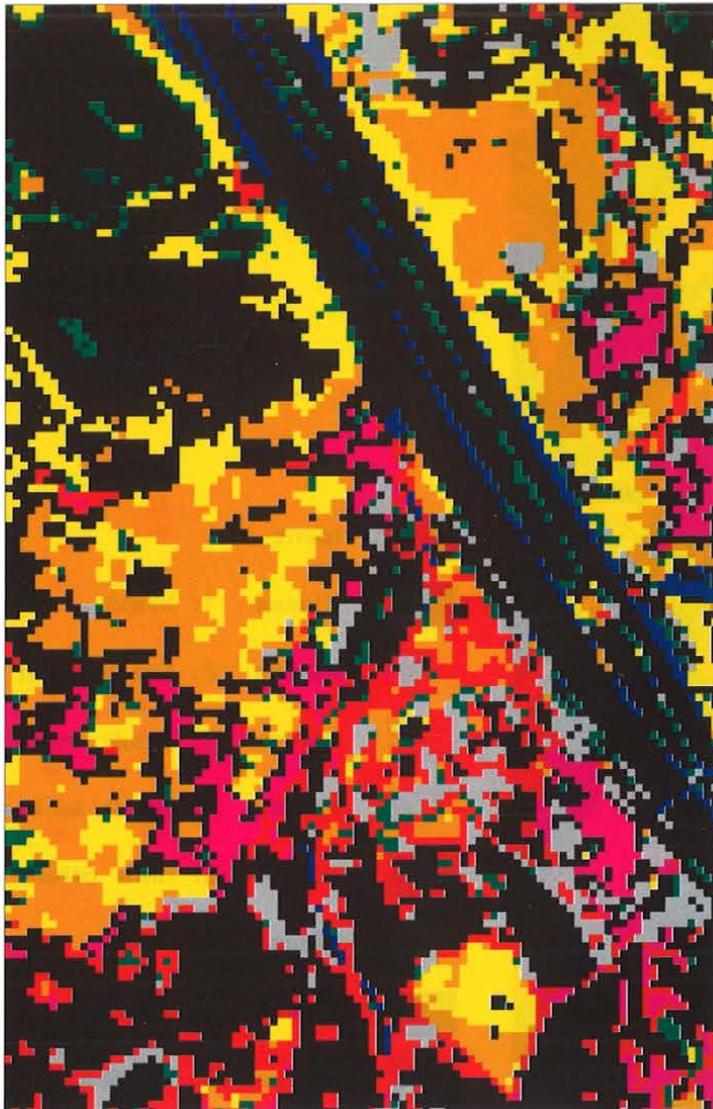
	<b>ident</b>
	<b>Klasse 1: Wald</b>
	<b>Klasse 2: Stadt dicht</b>
	<b>Klasse 3: Stadt mittel</b>
	<b>Klasse 4: Stadt locker</b>
	<b>Klasse 5: halb offen</b>
	<b>Klasse 6: offen</b>
	<b>Klasse 7: Wasser</b>
	<b>Klasse 10: versiegelt</b>

**Postklassifikation 100x100m**

**Orthophoto 1 : 10 000**  
**Visuelle Klassifikation I**

**Differenzbild**

**Abb. 3.9: Differenzbild**



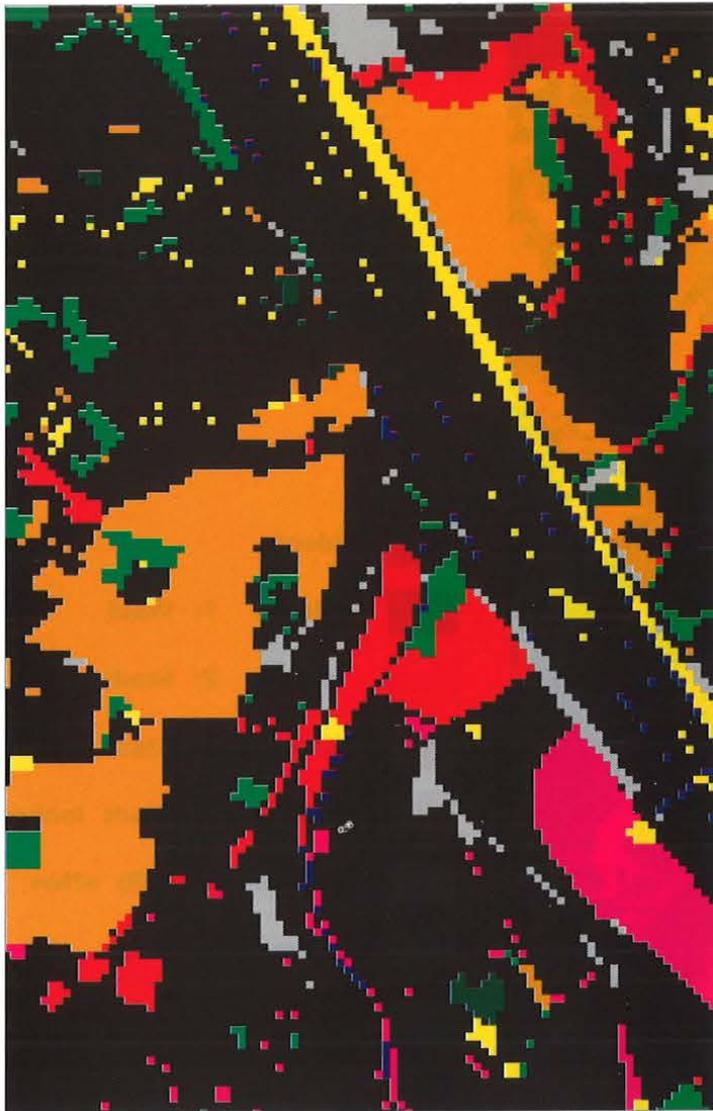
- ident
- Klasse 1: Wald
- Klasse 2: Stadt dicht
- Klasse 3: Stadt mittel
- Klasse 4: Stadt locker
- Klasse 5: halb offen
- Klasse 6: offen
- Klasse 7: Wasser
- Klasse 10: versiegelt

**Postklassifikation 100x100m**

**Orthophoto 1 : 10 000  
Visuelle Klassifikation II**

**Differenzbild**

**Abb. 3.10: Differenzbild**



- ident**
- Klasse 1: Wald**
- Klasse 2: Stadt dicht**
- Klasse 3: Stadt mittel**
- Klasse 4: Stadt locker**
- Klasse 5: halb offen**
- Klasse 6: offen**
- Klasse 7: Wasser**
- Klasse 10: versiegelt**

**Orthophoto 1 : 10 000  
Visuelle Klassifikation I**

**Orthophoto 1 : 10 000  
Visuelle Klassifikation II**

**Differenzbild**

**Abb. 3.11: Differenzbild**

zusätzlich eine Methode zur Bewertung der Differenzen entwickelt. Diese Bewertung geht von Definitionsunsicherheiten der einzelnen Klassen aus und bewertet dementsprechend Differenzen zwischen thematisch ähnlichen Klassen geringer als solche zwischen thematisch fremden Klassen (vgl. Tab. 3.9).

**Tab. 3.9: Bewertung der Differenzen**

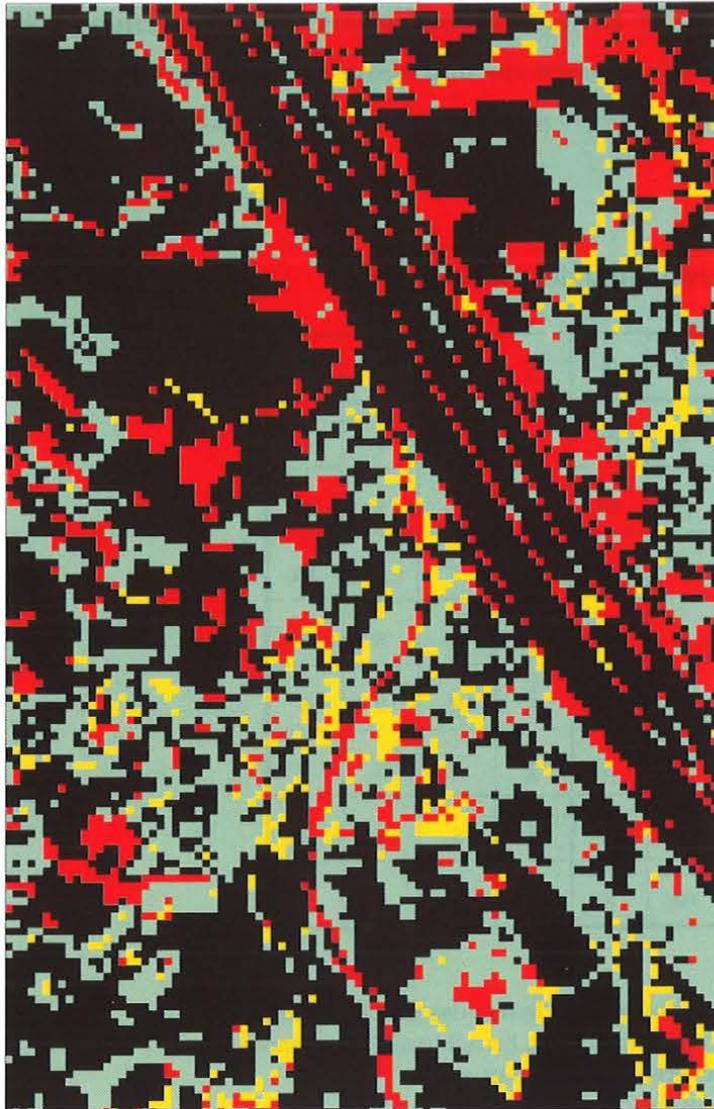
	1	2	3	4	5	6	7	10
1	i	s	s	m	g	s	s	s
2	s	i	g	m	s	s	s	m
3	s	g	i	g	m	s	s	m
4	m	m	g	i	g	s	s	s
5	g	s	s	g	i	g	s	s
6	s	s	s	s	g	i	s	s
7	s	s	s	s	s	s	i	s
10	s	m	m	s	s	s	s	i

Bewertung:

- i ... ident (beide Klassifizierungen liefern das gleiche Ergebnis)
- g ... gut (die Klassen sind thematisch benachbart, daher können Definitionsunsicherheiten leicht auftreten)
- m ... mittel (Definitionsunsicherheiten sind möglich)
- s ... schlecht (Definitionsunsicherheiten sind nicht möglich)

Die Bewertungen der Differenzbilder mittels der oben angeführten Matrix sind in den Abb. 3.9, 3.10 und 3.11 dargestellt. Es ist deutlich zu sehen, daß sich ein Großteil der Differenzen zwischen thematisch ähnlichen Klassen ergibt (Bewertung: gut). Eine numerische Darstellung der bewerteten Differenzen gibt Tab. 3.10.

Ohne auf die "Richtigkeit" der Ergebnisse einzugehen, soll an dieser Stelle ein vorläufiger Vergleich zwischen den vorhandenen Ergebnissen gezogen werden. Es ist deutlich zu sehen, daß die beiden visuellen Interpretationen ein höheres Maß an Übereinstimmung aufweisen, als die automatische Klassifikation mit jeweils einer von ihnen aufweist. Diese Ähnlichkeit, die sich aus dem Einsatz derselben Methode ergibt, ist jedoch in weit geringerem Maße vorhanden, als anzunehmen war. Die auftretenden Differenzen zwischen den visuellen Interpre-

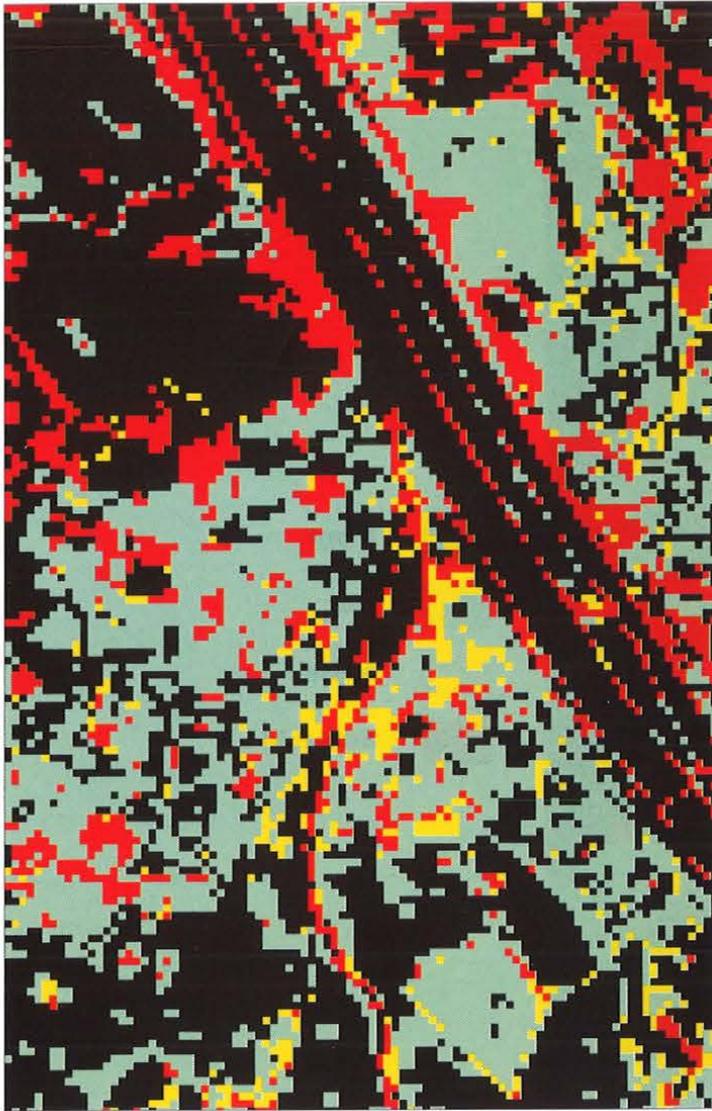


**Postklassifikation 100x100m**

**Orthophoto 1 : 10 000  
Visuelle Klassifikation I**

**Bewertete Differenzen**

**Abb. 3.12: Differenzbild bewertet**

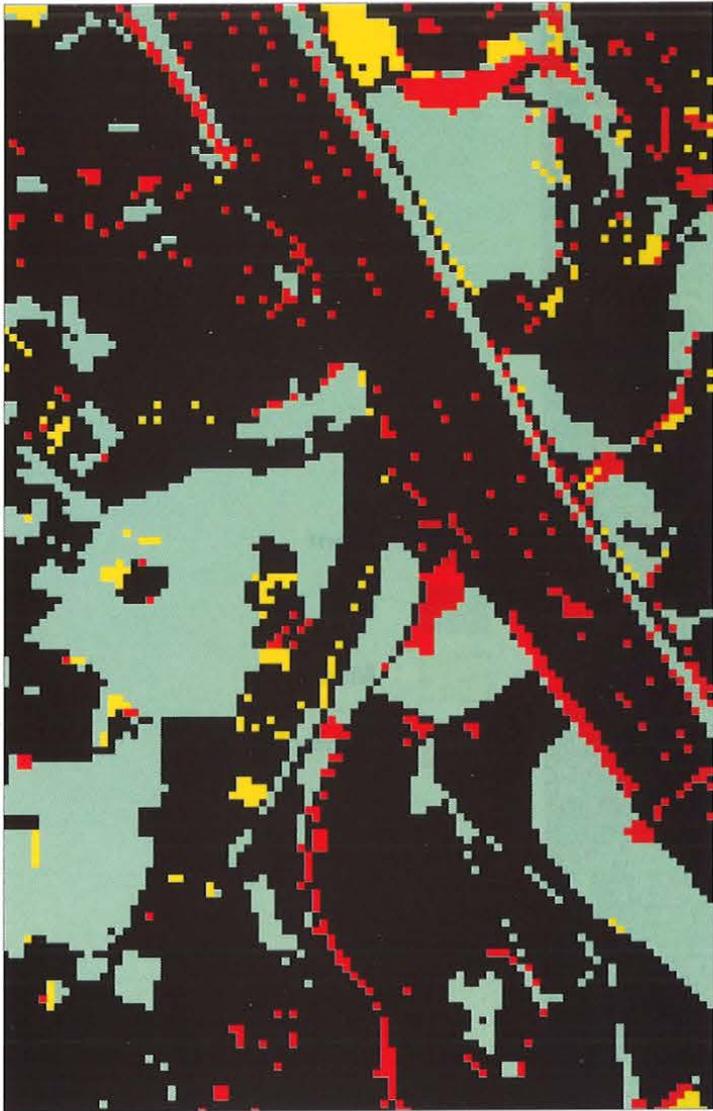


**Postklassifikation 100x100m**

**Orthophoto 1 : 10 000**  
**Visuelle Klassifikation II**

**Bewertete Differenzen**

**Abb. 3.13: Differenzbild bewertet**



**Orthophoto 1 : 10 000**  
**Visuelle Klassifikation I**

**Orthophoto 1 : 10 000**  
**Visuelle Klassifikation II**

**Bewertete Differenzen**

**Abb. 3.14: Differenzbild bewertet**

tationen sind auf Definitionsunsicherheiten in den einzelnen Klassen zurückzuführen. Dabei lassen sich zwei Problemstellungen unterscheiden:

**Tab. 3.10: Summenhäufigkeitsverteilung der Bewerteten Differenzen**

	ident	+ gut	+ mittel	+ schlecht
auto : vis I	58.8	83.2	86.5	100.0
auto : vis II	54.0	81.8	85.7	100.0
vis I : vis II	72.6	93.3	95.5	100.0

Häufigkeit in %

Mischgebiete, die in ihrer thematischen Charakteristik Merkmale zweier Klassen vereinen, werden nach subjektivem Empfinden der einen oder anderen Klasse zugeordnet - z.B.: Ist das untersuchte Gebiet mittel oder locker verbaut ?

Übergänge zwischen räumlich benachbarten Objektklassen sind nur unsicher zu definieren - z.B.: Wo liegt geometrisch die Grenze zwischen mittel und locker verbauter Stadt ?

Beide Fragestellungen beschreiben das Problem der diskreten Darstellung unscharfer Objekte. Natürlich tritt dieses Problem auch im automatischen Ansatz auf, wobei jedoch durch die Pixelorientierung der Effekt nur kleinräumig zum Tragen kommt. Außerdem bewirkt die hierarchische Regelstruktur, daß in Übergangsbereichen zwingend Mischklassen zugeordnet werden und es dementsprechend zu keinen sprunghaften Übergängen kommen kann.

Eine konsequente Lösung dieses Dilemmas könnte mit Modellen der Fuzzy-Logik gefunden werden [vgl. z.B. Bezdek und Pal, 1992, Zimmermann, 1993]. Dabei wird ein Objekt nicht eindeutig einer Klassen zugeordnet, sondern das Maß seiner Zugehörigkeit zu allen Klassen bestimmt. Die oben angesprochenen Arten der Unsicherheit entsprechen dabei den beiden in der Fachliteratur als *ambiguity* und *vagueness* bezeichneten Kategorien von Unschärfe [vgl. Klir, 1987].

### 3.4.2 REFERENZDATEN

Um die automatisch und visuell klassifizierten Resultate einer Plausibilitätskontrolle unterziehen zu können, wurden an knapp 200 Punkten (ca. 1 % der klassifizierten Pixel) Referenzinterpretationen im Orthophoto vorgenommen. Konnte anhand des Orthophotos und der Ortskenntnis keine eindeutige Entscheidung getroffen werden, wurde am fraglichen Referenzpunkt eine Begehung durchgeführt. Die Verteilung der Punkte entspricht den Knotenpunkten eines regelmäßigen Netzes mit 500 m Maschenweite über dem gesamten Testgebiet. Im Gegensatz zur visuellen Klassifizierung handelt es sich dabei nicht um eine flächenhaft integrierende Methode, sondern um Untersuchungen räumlich klar definierter Punkte und ihrer unmittelbaren Umgebung. Die Auswahl der Punkte kann als näherungsweise zufällig interpretiert werden, da keine Systematik zwischen der Geometrie des Netzes und der räumlichen Verteilung der untersuchten Objekte besteht. Unterstützt wird diese Behauptung durch die Häufigkeitsverteilung der Objektklassen in den Referenzpunkten (Tab. 3.11) im Vergleich zur Häufigkeitsverteilung der Objektklassen in den Ergebnislayern (Tab. 3.8).

**Tab. 3.11: Häufigkeitsverteilung der Objektklassen in den Referenzpunkten**

Objektklasse	1	2	3	4	5	6	7	10
Referenzdaten	5.3	17.1	10.2	17.6	11.8	15.0	9.1	13.9
visuell I	5.3	18.2	13.2	21.3	8.9	14.8	8.0	9.5
visuell II	6.9	20.2	26.7	8.4	7.4	12.3	9.0	8.4
automatisch	5.3	16.6	9.0	16.6	4.8	27.8	9.1	10.7

Der Vergleich der Klassifikationen mit den Referenzdaten wird mittels Konfusionsmatrizen anschaulich gemacht (vgl. Abb. 3.12, 3.13 und 3.14). Die Spalten repräsentieren die Referenzdaten, die Zeilen die Ergebnisse der einzelnen Klassifikationen in Prozent der Gesamtanzahl der referenzierten Punkte. Die letzte Spalte beschreibt die innerhalb einer Klasse korrekt klassifizierte Pixel in Prozent der Häufigkeit der Klasse. Die letzte Zeile beschreibt die innerhalb einer Klasse richtig erkannten Referenzdaten in Prozent der Häufigkeit der Klasse. Das letzte Element der Matrix - letzte Zeile, letzte Spalte - beinhaltet die insgesamt korrekt klassifizierte Pixel in Prozent der Gesamtanzahl der referenzierten Punkte.

Die Gesamtzahl der korrekt klassifizierten Punkte zeigt bei der automatischen Klassifikation mit 77 % den höchsten Wert. Die visuelle Klassifikation I liegt mit 70 % knapp dahinter, die zweite visuelle Klassifikation fällt mit 60 % stark ab. Eine detaillierte Betrachtung der einzelnen Klassen läßt Rückschlüsse auf die Fehlerquellen zu. Bei der automatischen Klassifikation kommt ein Großteil der Fehlklassifikationen in thematisch benachbarte Klassen zu liegen. Setzt man das oben angeführte Bewertungsschema (Tab. 3.9) ein, so fallen 12.8 % in thematisch ähnliche Klassen, d.h. knapp 90 % der Referenzpunkte werden als "gut" bewertet.

Auffallend fehlerbehaftet zeigt sich die Klasse 6 (offen), die nur zu 50 % richtig erkannt wurde. Die Fehlklassifikationen fallen vor allem in die Klassen 4 (Stadt locker) und 5 (halb offen). Die Analyse betroffener Gebiete zeigt, daß diese Fehler auf die multispektrale Klassifikation zurückzuführen sind. Die Klasse VII (Laubwald), die unter anderem auch die innerstädtischen Baumbestände vor allem in Parks erfassen soll, ist offensichtlich unterrepräsentiert. Besonders augenscheinlich ist das im 2. Bezirk im Augarten zu sehen, der fast zur Gänze als Wiese klassifiziert wurde (vgl. Abb. 3.3, als Feld/Wiese klassifiziertes Gebiet im Süden des Testgebietes). In der Postklassifikation bewirkt das Fehlen der "Waldpixel" eine Verschiebung derjenigen Mischklassen, die potentiell Laubwald enthalten könnten, in Richtung der Klasse "offen" - und das sind die Klassen 4 und 5. Die Klasse 1 (Wald) ist von

**Tab. 3.12: Konfusionsmatrix der Postklassifikation**

Postkl.	Referenzdaten								% ident
	1	2	3	4	5	6	7	10	
1	<b>5.3</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100
2	0.0	<b>15.5</b>	0.0	0.0	0.0	0.0	0.0	1.1	93
3	0.0	1.1	<b>6.9</b>	0.5	0.0	0.0	0.0	0.5	77
4	0.0	0.0	2.1	<b>12.3</b>	1.1	0.0	0.0	1.1	74
5	0.0	0.0	0.0	0.0	<b>3.7</b>	1.1	0.0	0.0	77
6	0.0	0.0	1.1	4.8	6.9	<b>13.9</b>	0.0	1.1	50
7	0.0	0.0	0.0	0.0	0.0	0.0	<b>9.1</b>	0.0	100
10	0.0	0.5	0.0	0.0	0.0	0.0	0.0	<b>10.2</b>	95
<b>% ident</b>	100	91	68	70	32	93	100	73	77

Tab. 3.13: Konfusionsmatrix der visuellen Klassifikation I

Vis. I	Referenzdaten								% ident
	1	2	3	4	5	6	7	10	
1	<b>4.8</b>	0.0	0.0	0.0	0.5	0.0	0.0	0.0	90
2	0.0	<b>15.0</b>	3.2	0.0	0.0	0.0	0.0	0.0	72
3	0.0	2.1	<b>4.8</b>	2.1	1.0	0.5	0.0	2.7	39
4	0.5	0.0	1.6	<b>13.4</b>	3.2	1.0	0.0	1.6	64
5	0.0	0.0	0.0	1.0	<b>5.3</b>	1.6	0.0	1.0	67
6	0.0	0.0	0.0	1.0	1.0	<b>11.2</b>	1.6	0.0	72
7	0.0	0.0	0.5	0.0	0.0	0.0	<b>7.0</b>	0.5	93
10	0.0	0.0	0.0	0.0	0.5	0.5	0.5	<b>8.0</b>	83
<i>% ident</i>	90	88	47	76	45	75	76	58	<b>70</b>

Tab. 3.14: Konfusionsmatrix der visuellen Klassifikation II

Vis. II	Referenzdaten								% ident
	1	2	3	4	5	6	7	10	
1	<b>4.8</b>	0.0	0.0	0.5	1.6	0.0	0.0	0.0	69
2	0.0	<b>13.9</b>	2.1	0.5	0.0	0.5	0.0	3.2	68
3	0.0	2.1	<b>6.4</b>	10.7	2.7	1.1	0.0	3.7	24
4	0.5	0.0	0.0	<b>5.3</b>	2.1	0.5	0.0	0.0	63
5	0.0	0.5	0.5	0.0	<b>4.3</b>	1.6	0.5	0.0	57
6	0.0	0.0	0.0	0.0	1.1	<b>10.7</b>	0.5	0.0	87
7	0.0	0.0	0.5	0.0	0.0	0.0	<b>8.0</b>	0.5	88
10	0.0	0.5	0.5	0.5	0.0	0.5	0.0	<b>6.4</b>	75
<i>% ident</i>	90	81	63	30	36	71	88	46	<b>60</b>

diesem Fehler nicht betroffen, da geschlossene Waldgebiete in der multispektralen Klassifikation richtig erkannt werden. Um solche Fehler zu vermeiden, empfiehlt es sich,

bereits die multispektrale Klassifikation einer Plausibilitätskontrolle mit Referenzdaten zu unterziehen. Immerhin betrifft der dadurch entstandene Fehler mit 13 % mehr als die Hälfte aller Fehl-klassifikationen. Eine Elimination dieses Fehlers durch Korrektur der multispektralen Klassifikation würde das Gesamtergebnis auf knapp 90 % verbessern.

Weitere Unsicherheiten, wenn auch in weit kleinerem Ausmaß als die eben beschriebenen, sind bei den Klassen 3 (Stadt mittel) und 4 (Stadt locker) und bei der Klasse 10 (versiegelt) zu bemerken. Die Grenzen zwischen mittel und locker verbauter Stadt sind unscharf und daher schwer zu definieren. Die Klasse 10 steht in gewisser Weise außerhalb des Klassifizierungsschlüssels, da sie Gebiete beschreibt, die eigentlich einer der übrigen Klassen zugeordnet werden sollten, wobei dies aufgrund ihrer spektralen Charakteristika nicht möglich ist. Die Fehlklassifikationen dieser Klasse liegen daher durchwegs in potentiell korrekten Klassen - versiegelte Flächen sind plausibel im gesamten Stadtgebiet (Klassen 3, 4, 5) und in offenen Gebieten (Klasse 6).

Die visuelle Klassifikation I zeigt eine ähnliche Verteilung wie die automatische Klassifikation, wobei die Abweichungen von den Referenzdaten höher ausfallen. Die Anzahl der ident klassifizierten Punkte beträgt 70 %, unter Einbeziehung der als gut bewerteten Punkte erreicht sie 86 %. Auffallend große Differenzen zeigen sich zwischen den städtischen Klassen 2, 3, 4 und der Klasse 5 (halb offen). Diese Fehlklassifikationen sind auf den hohen Generalisierungsgrad dieser Methode zurückzuführen - kleinräumige Änderungen der Nutzungsart werden nicht ausgewiesen.

Die visuelle Klassifikation II fällt im Vergleich mit den anderen Ergebnissen deutlich ab. Ident klassifiziert sind 60 % der Punkte, unter Einbeziehung der als gut bewerteten Punkte erhöht sich der Wert um 21.3 % auf knapp über 83 %. Auch hier liegt wieder der Grund im hohen Grad der Generalisierung, zusätzlich treten aber Auffassungsunterschiede in der Interpretation der Klassen auf. Offensichtlich kommt es hier zu einer Verschiebung der Klassengrenzen im Interpretationsraum. Besonders deutlich ist dieser Effekt zwischen den Klassen 3 und 4 zu sehen, wo die Konfusionsmatrix eine Fehlinterpretation von 10.7 % in Richtung 3 nach 4, aber 0 % in Richtung 4 nach 3 ausweist.

Welche interpretatorischen Unsicherheiten außerdem noch auftreten können, zeigt ein Detail am östlichen Rand des Untersuchungsgebietes. In beiden visuellen Klassifikationen wird ein kleiner Teil der Alten Donau, der gerade noch ins Untersuchungsgebiet fällt, nicht als Wasserfläche erkannt, sondern wie der umliegende Park als halb offen klassifiziert. Der Grund dafür liegt in der Tatsache, daß in einem panchromatischen Bild Objekte eher durch Textur und räumlichen Kontext als durch spektrale Charakteristik erkannt werden. Im Falle der Alten Donau sind spektrale Unterschiede zwischen Wasser und umliegendem Park kaum

vorhanden, auch die Textur weist nur geringe Differenzen auf. Zur deutlichen Trennung der Nutzungsarten fehlt der räumliche Kontext, weil die Wasserfläche zu klein ist. In der automatischen Klassifikation wird die Wasserfläche über ihre spektrale Charakteristik eindeutig erkannt.

Zusätzlich zu den genannten Fehlerquellen dürften durch die mehrfache Transformation der visuell interpretierten Ergebnisse - Hochzeichnen der Interpretation, digitalisieren, Transformation ins Landessystem und Vektor-Raster Konversion - geometrische Unsicherheiten entstanden sein, wodurch einzelne Pixel in Übergangsbereichen der räumlichen Nachbarnutzung zugeordnet wurden. Dieser Effekt zeigt sich in den Differenzbildern (Abb. 3.9 und 3.10) im Randbereich zwischen Donau und Ufer.

### 3.4.3 DISKUSSION

Der in den letzten Kapiteln durchgeführte Vergleich verdeutlicht die unterschiedlichen Charakteristika der Klassifikationsansätze. Die visuelle Methode wirkt großflächig integrierend, wodurch ein optisch (kartographisch) ansprechender Eindruck entsteht. Die automatische Klassifikation dagegen geht auf lokale Klassenunterschiede ein und liefert somit die für die Modellierung der Funkwellenausbreitung relevantere Information. Unterstützt wird diese Aussage durch die höhere Übereinstimmung der automatisch ermittelten Flächennutzungen mit den Referenzdaten. Darüber hinaus zeigen die visuellen Klassifikationen untereinander ein unerwartet hohes Maß an Variation. Diese Differenzen sind vor allem auf die subjektiven Interpretationen der Auswerter zurückzuführen. Zusätzlich stellt sich das Problem einer geeigneten Definition der einzelnen Klassen, die als objektive Basis vom Interpretieren herangezogen werden kann. Im Gegensatz dazu bietet der automatische Ansatz über den Regelkomplex eine klar definierte, nachvollziehbare Methodik. Vor allem im Sinne einer Wiederholbarkeit der Klassifikation zu einem späteren Zeitpunkt zur Erkennung von Veränderungen (*change detection*) ist die automatische Klassifikation der visuellen vorzuziehen.

Abschließend sei noch kurz auf die Wirtschaftlichkeit der beiden Verfahren eingegangen. Als Rahmen für die Kalkulationen (Tab. 3.15 und 3.16) wird der Großraum Wien inklusive dem Wiener Becken herangezogen, der als stark besiedeltes Gebiet für den Mobilfunk von Interesse ist. Die räumliche Ausdehnung reicht in Nord-Süd-Richtung von Korneuburg bis Wiener Neustadt, in West-Ost-Richtung vom Wienerwald bis Schwechat, und beträgt 60 x 40 km<sup>2</sup>. Das Gebiet deckt ca. 30 % einer TM Viertelszene bzw. 70 % einer SPOT Szene ab. Für eine

visuelle Klassifikation werden 96 Orthophotos benötigt. Die Kalkulation versteht sich als Abschätzung, alle Kosten sind in öS angegeben und auf  $\pm 50.--$  gerundet.

Um die unterschiedlichen Anforderungen zu berücksichtigen, wird für die visuelle Interpretation ein Stundensatz von 500.-- öS, für die automatische Klassifikation ein Stundensatz von 1000.-- öS gerechnet. In den Posten Projektplanung und -management fallen vorbereitende und organisatorische Arbeiten, wie die Auswahl der notwendigen Daten oder die Bestimmung geeigneter Trainings- und Testgebiete. Da diese Arbeit bei beiden Verfahren anfällt, werden die Kosten äquivalent angesetzt. Die Kalkulation der anfallenden EDV-Kosten erfolgt über prozentuelle Anteile der Arbeitsstunden. Der höhere Prozentsatz gilt für den Einsatz von Bildverarbeitungssystemen und GIS, der niedrigere für die Verwendung des CAD Programmes zum Digitalisieren.

Bei der visuellen Klassifikation wachsen die Kosten linear mit der Anzahl der benötigten Orthophotos. Die Kosten der automatischen Klassifikation dagegen ändern sich nicht in Abhängigkeit der klassifizierten Fläche, solange das Untersuchungsgebiet innerhalb einer Satellitenszene liegt. Die Verringerung des Arbeitsaufwandes bei einem kleineren Untersuchungsgebiet - leichtere Handhabung der Daten, kürzere Rechenzeiten etc. - werden hier nicht berücksichtigt. Ohne Overheadkosten einzukalkulieren, kostet ein  $\text{km}^2$  visuelle Interpretation ca. öS 107.--. Ab einer Untersuchungsfläche von mehr als  $1600 \text{ km}^2$  ist daher die automatische Klassifikation trotz der hohen Grundkosten die wirtschaftlichere Methode. Bei einer flächendeckenden Untersuchung eines Gebietes von  $180 \times 180 \text{ km}^2$  (= 1 TM Szene, 9 SPOT Szenen) erreicht die Kostenrelation ihr Maximum (automatische Klassifikation ca. 0.8 Mio., visuelle Klassifikation ca. 3.5 Mio.).

### **3.5 PERSPEKTIVEN**

Aufgabe der vorgestellten Studie war es, die Eignung von Fernerkundungsdaten zur Detektion von Landnutzung für die Planung von Mobilfunknetzen zu testen, zu bewerten und mit einem alternativen Verfahren, der visuellen Interpretation von Orthophotos, zu vergleichen.

Der Vorteil der automatischen Klassifikation liegt im methodisch klar definierten Ablauf. Unsicherheiten können in der multispektralen Klassifikation auftreten, die als der sensibelste Teil des Gesamtprozesses anzusehen ist. Eine geeignete Wahl der Trainingsgebiete und Kontrollen mittels Referenzdaten sollten die auftretenden Fehler jedoch minimieren. Die Texturanalyse erweist sich als wertvolles Korrekturinstrument für die Trennung verbauter

**Tab. 3.15: Aufwandsabschätzung visuelle Klassifikation**

Projektplanung - Projektmanagement	24 h	à 1000.--		<b>24 000.--</b>
+ 20 % Mwst.				<b>4 800.--</b>
96 Orthophotos		à 850.--		<b>81 600.--</b>
Arbeitsaufwand	96 Orthophotos	à 1 500.--		<b>144 000.--</b>
<hr/>				
Interpretation und Digitalisierung	1.5 h	à 500.--	750.--	
Datenaufbereitung und Kodierung	1.5 h	à 500.--	750.--	
<hr/>				
+ 20 % Mwst.				<b>28 800.--</b>
EDV-Kosten (AutoCad)	+10 %			<b>7 200.--</b>
EDV-Kosten (ArcInfo)	+20 %			<b>14 400.--</b>
<hr/>				
Gesamtkosten				<b>304 800.--</b>

**Tab 3.16: Aufwandsabschätzung automatische Klassifikation**

Projektplanung - Projektmanagement	24 h	à 1000.--		<b>24 000.--</b>
+ 20 % Mwst.				<b>4 800.--</b>
Landsat TM Viertelszene				<b>26 000.--</b>
SPOT P Szene				<b>37 500.--</b>
Arbeitsaufwand	80.0 h	à 1000.--		<b>80 000.--</b>
<hr/>				
Rektifizierung ohne DHM	16.0 h			
multispektrale Klassifikation	24.0 h			
Texturanalyse	16.0 h			
Postklassifikation	24.0 h			
<hr/>				
+ 20 % Mwst.				<b>16 000.--</b>
EDV-Kosten (Erdas, Impuls, ArcInfo) + 20 %				<b>16 000.--</b>
<hr/>				
Gesamtkosten				<b>204 300.--</b>

und unverbauter Gebiete. Die regelbasierte Postklassifikation stellt ein geeignetes Instrument zur Erkennung der gesuchten Landnutzungsklassen dar. Die Zuverlässigkeit der erlangten Ergebnisse liegt höher als im alternativen Verfahren.

Die visuelle Interpretation zeigt starke subjektive Einflüsse der Interpreten, die durch einen gut dokumentierten Klassifikationsschlüssel mit für die einzelnen Klassen markanten Trainingsgebieten zwar verringert, aber nicht eliminiert werden könnten. Als zweiter Nachteil erweist sich der Generalisierungseffekt, durch den die kleinräumige Verteilung der Landnutzungsklassen in stark strukturierten Gebieten übergangen wird. Es stellt sich die Frage, ob diese Generalisierung ohne signifikant vergrößerten Arbeitsaufwand reduziert werden kann.

Letztlich zeigt die Abschätzung der Wirtschaftlichkeit der beiden Verfahren, daß ab einer regionalen Größe des Untersuchungsgebietes die automatische Verarbeitung kostengünstiger als die visuelle Interpretation ist.

Wie bereits mehrmals erwähnt, soll die vorliegende Klassifikation zusammen mit einem digitalen Höhenmodell als Basis für Funkausbreitungsmodelle dienen. Um Aussagen über die Eignung der Landnutzungsinformation für diesen Zweck zu treffen, müßte das Ausbreitungsmodell auf die erstellten Datenlayer aufgesetzt werden. Es wäre naheliegend, das Modell innerhalb des Raster-GIS zu implementieren. Die Problematik der räumlichen Ausbreitung von Funkwellen unter Berücksichtigung räumlicher Charakteristika fällt eindeutig in den Bereich geographischer Informationssysteme. Zusätzlich zur Modellierung im GIS wären funktechnische Referenzmessungen in Teilen des Untersuchungsgebietes notwendig, um die modellierten Größen in der Natur zu verifizieren. Erst damit wäre letztlich die Eignung der in diesem Abschnitt beschriebenen Methodik im gesamten Kontext bewertbar. Leider sind die eben beschriebenen Vorgangsweisen nur ein Ausblick auf zukünftige Tätigkeiten, da sie in hohem Maße von Aktivitäten außerhalb des Institutes für Photogrammetrie und Fernerkundung abhängen. Es steht zu hoffen, daß die Publikation der vorliegenden Arbeit einen gewissen Anstoß für weiterführende Entwicklungen auf diesem Gebiet gibt.

## 4. CONCLUSIO

Das Ziel der vorliegenden Arbeit ist eine umfassende Behandlung der Thematik "Erfassung von Landnutzung mit Methoden der Fernerkundung". Die Spannweite der behandelten Gebiete reicht von der Theorie der Mustererkennung über Aspekte der Implementierung bis hin zum praktischen Einsatz.

Am Beginn steht eine theoretische Betrachtung traditioneller Klassifikationsverfahren, die um die Methodik der Texturanalyse erweitert werden. Als Weiterführung wird auf Verfahren der Postklassifikation eingegangen, die primäre Klassifikationsergebnisse in einem thematischen Kontext überarbeiten und die Verbindung zur geographischen Informationsverarbeitung herstellen.

Der zweite Teil befaßt sich mit der Implementierung der behandelten Algorithmen auf einem Parallelrechner. Es handelt sich dabei um ein Transputernetzwerk, das über einen PC angesprochen wird und mit einem hochauflösenden Graphikschirm ausgestattet ist. Die Ausführungen konzentrieren sich auf die entwickelten Programmschalen, die eine effiziente Datenverwaltung und Kommunikation innerhalb des Systems ermöglichen. Das Design dieser Programme ist so ausgelegt, daß beliebige, lokale Bildverarbeitungsalgorithmen problemlos implementiert werden können. Vergleiche zwischen sequentieller und paralleler Implementierung der Algorithmen zeigen die hohe Effizienz, die durch den Einsatz des Transputernetzwerkes erreicht wird. Um die eingesetzten Verfahren der Bildverarbeitung sinnvoll austesten zu können, wird auch der Entwicklung geeigneter Visualisierungskonzepte hohe Aufmerksamkeit geschenkt. Das Programmsystem wurde im Zuge von Lehrveranstaltungen und bei mehreren Applikationen verwendet und hat sich im praktischen Einsatz bewährt.

Im dritten Abschnitt erfolgt, als Conclusio der beiden vorhergegangenen Teile, der Einsatz der beschriebenen und implementierten Methoden in einer aktuellen Applikation. Die Problemstellung kommt aus dem Fachgebiet der Nachrichtentechniker, die zur Modellierung zellulärer Mobilfunknetze flächendeckende Landnutzungsdaten benötigen. Ein integrativer Klassifikationsansatz, der sich aus den Verfahren der multispektralen Klassifikation, der Texturanalyse und der Postklassifikation zusammensetzt, wird auf einen hybriden Datensatz, bestehend aus einem Landsat-TM- und einem panchromatischen SPOT-Bild, angewandt. Als Alternativverfahren erfolgt eine visuelle Interpretation analoger Orthophotos. Ein Vergleich der Ergebnisse und die Verifizierung über punktuell erhobene Referenzdaten sprechen für den Einsatz der automatischen Methode. Darüberhinaus liegen die Kosten dieses Verfahrens ab einer Gebietsgröße von etwa 1600 km<sup>2</sup> unter denen der visuellen Interpretation.

Die hier vorgestellten Verfahren bilden einen geschlossenen Ansatz, der in seiner Gesamtheit dem Bereich der Fernerkundung zugeordnet werden kann. Die Ergebnisse liegen in Form eines raumbezogenen Datensatzes der Landnutzung vor. Die Anwendungsmodelle (z.B. Funkausbreitungsmodelle), die diesen Datensatz benötigen, sind im allgemeinen in Systemen implementiert, die dem Bereich der geographischen Informationsverarbeitung angehören (GIS). Die Landnutzungsdaten repräsentieren somit sowohl eine Schnittstelle, als auch eine scharfe Abgrenzung zwischen Fernerkundung und GIS. Allerdings stellt sich die Frage, inwieweit die beschriebenen Verfahren, insbesondere die der Postklassifikation, nicht schon der geographischen Informationsverarbeitung zuzurechnen sind. Daher ist eher davon auszugehen, daß die Bereiche Fernerkundung und GIS einander überlappen und eine scharfe Abgrenzung weder möglich noch sinnvoll erscheint.

Die logische Konsequenz obiger Ausführungen ist die verstärkte Integration von Fernerkundung und GIS, sowohl auf methodischer als auch auf systembezogener Ebene. Die Vorteile für beide Seiten liegen klar auf der Hand; einerseits könnte die Einbeziehung geographischer Information in den Prozeß der Klassifikation deren Ergebnisse verfeinern und die Zuverlässigkeit erhöhen, andererseits würde die Verfügbarkeit der gesamten im Klassifikationsprozeß anfallenden Information die Qualität der Modellierung verbessern. Die lose Koppelung der Systeme, wie sie heute verfügbar ist, erlaubt zwar eine gewisse Integration, allerdings nur unter Einsatz von Spezialkenntnissen und viel Zeit. Notwendig wären dementsprechend integrative Systeme mit geeigneten Datenstrukturen, die sowohl Bild- als geographische Information verarbeiten können und mit benutzerfreundlichen Oberflächen ausgestattet sind. Die Einbindung von Expertensystemen könnte in Folge dazu beitragen, Lösungen für bestimmte Applikationsbereiche zu entwickeln, in denen der Anwender nur mehr seine fachspezifische Aufgabenstellung zu meistern hat.

## Literatur

- AIMET U. (1992): Die Kombination von Fernerkundung und Geographischen Informationssystemen am Beispiel einer Kulturlandschaftstypenklassifikation. *Diplomarbeit, Grund- und Integrativwissenschaftlichen Fakultät, Universität Wien.*
- ARGIALAS D.P. und HARLOW C.A. (1990): Computational image interpretation models: an overview and a perspective. *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 6, S. 871-886.
- BABNIK J. (1994): Bildverarbeitung auf Transputern - Entwicklung systemunterstützender Software. *Dissertation, Technisch-naturwissenschaftliche Fakultät, Technische Universität, Wien*, in Arbeit.
- BABNIK J. und STEINNOCHER K. (1991): Digital Image Processing on a Transputer System. *Schriftenreihe der österreichischen Computergesellschaft*, Band 58, S. 1-9.
- BARBER D.G. und LEDREW E.F. (1991): SAR sea ice discrimination using texture statistics: a multivariate approach. *Photogrammetric Engineering and Remote Sensing*, Vol. 57, No. 4, S. 385-395.
- BARNESLEY M.J. und BARR S.L. (1992): Developing spatial re-classification techniques for improved land-use monitoring using high spatial resolution images. *International Archives of Photogrammetry and Remote Sensing*, Vol. 29, B 7, Commission VII. S. 646-654.
- BAUERNÖPPEL F. (1991): Imploding ... freezing ... done. Verfahren und Techniken zur Datenkompression. *c't*, 10/91, S. 278-286.
- BEZDEK J.C. und PAL S.K. (Hrsg.): *Fuzzy Models for Pattern Recognition*. IEEE Press, Piscataway, NJ., 1992, S. 6-27.
- BOLSTAD P.V. und LILLESAND T.M. (1992): Rule-based classification models: flexible integration of satellite imagery and thematic spatial data. *Photogrammetric Engineering and Remote Sensing*, Vol. 58, No. 7, S. 965-971.
- BÜHLER H. (1990): Feldstärkevorhersage für Mobilfunknetze. *Diplomarbeit, Fakultät für Elek-trotechnik, Technische Universität Wien.*
- BÜHLER H. und STEINNOCHER K. (1991): Deriving land cover data for mobile radio from satellite data. Paper presented at the *European Cooperation in the field of Scientific and Technical Research*, Lund. COST 231 TD (91)/032.
- CONNERS R.W., TRIVEDI M.M. und HARLOW C.A. (1984): Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, 25, 273-310.

CSAPLOVICS E., KANONIER J. UND SINDHUBER A. (1993): Zur Integration hochauflösender kosmischer Photoaufnahmen mit der Kamera KFA-1000 in regionale Geo-Informationssysteme - Fallstudie Nationalpark Neusiedler See - Fertő Tó. *Proceedings AGIT V*, 7.-9. Juli 1993, Salzburg, (=Salzburger Geographische Materialien, Heft 20), S. 283-289.

CURRAN P.J. (1985): *Principles of Remote Sensing*. Longman, London.

ERDAS (1991): *Field Guide*. ERDAS, Atlanta.

EHLERS M., GREENLEE D., SMITH T. UND STAR J. (1991): Integration of Remote Sensing and GIS: Data and Data Access. *Photogrammetric Engineering and Remote Sensing*, Vol. 57, No. 6, S. 669-675.

FAUST N.L., ANDERSON W.H. und STAR J.L. (1991): Geographic Information Systems and Remote Sensing Future Computing Environment. *Photogrammetric Engineering and Remote Sensing*, Vol. 57, No. 6, S. 655-668.

FRANKLIN S.E. und PEDDLE D.R. (1990): Classification of SPOT HRV imagery and texture features. *International Journal of Remote Sensing*, Vol. 11, No. 3, S. 551-556.

GIERLOFF-EMDEN H.G. (1989): Fernerkundungskartographie mit Satellitenaufnahmen. *Enzyklopädie der Kartographie*, Band IV/1, Franz Deuticke, Wien.

GONZALES R.C. und WINTZ P. (1987): *Digital Image Processing* (second edition). Addison-Wesley, New York.

GONG P. und HOWARTH P.J. (1992): Land-use classification of SPOT HRV data using a cover-frequency method. *International Journal of Remote Sensing*, Vol. 13, No. 8, S. 1459-1471.

GUO L.J. und MOORE J.M. (1991): Post-classification processing for thematic mapping based on remotely sensed image data. *Proceedings International Conference IEEE Geoscience and Remote Sensing Society*, 3.-7. Juni 1991, Espoo, Finnland, S. 2203-2206.

HAIR J.F., ANDERSON R.E. und TATHAM R.L. (1987): *Multivariate Data Analysis*, Macmillan, New York.

HARALICK R.M., SHANMUGAM K. und DINSTEN I. (1973): Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, No. 6, S. 610-621.

HENKER S. (1989): Untersuchungen zur Einbeziehung von Texturmerkmalen in Verarbeitungskonzepte zur rechnergestützten Flächennutzungsdechiffrierung aus kosmischen Daten der Fernerkundung. *Vermessungstechnik (Berlin Ost)*, Vol. 37, No. 1, S. 22-25.

HILL G. (1987): Designs and Applications for the IMS C004. *Technical note 19*, INMOS, Bristol.

HOFFMANN J. (1991): Redundanz raus. Bildkompression mit DCT und anderen Transformationen. *c't*, 6/91, S. 126-128.

INMOS (1987): IMS T800 Architecture. *Technical note 6*, INMOS, Bristol.

INMOS (1988a): IMS B008 User Guide and Reference Manual. INMOS, Bristol.

INMOS (1988b): *Occam 2 Reference Manual*. Prentice Hall.

INMOS (1989): Occam 2 toolset user manual. INMOS, Bristol.

IRONS R.I. und PETERSEN G.W. (1981): Texture transforms of remote sensing data. *Remote Sensing of Environment*, No. 8, S. 359-370.

JANSSEN L.L.F., JAARSMA M.N. und VAN DER LINDEN E.T.M. (1990): Integrating Topographic Data with Remote Sensing for Land-Cover Classification. *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 11, S. 1503-1506.

JANSSEN L.L.F. und VAN AMSTERDAM J.D. (1991): An object based approach to the classification of remotely sensed images. *Proceedings International Conference IEEE Geoscience and Remote Sensing Society*, 3.-7. Juni 1991, Espoo, Finnland, S. 2191-2195.

JOHANSSON K. und KANONIER J. (1991): Knowledge based land-use classification. *Proceedings International Conference IEEE Geoscience and Remote Sensing Society*, 3.-7. Juni, 1991, Espoo, Finnland, S. 1847-1850.

KLIR G.J. (1987): Where do we stand on measures of uncertainty, ambiguity, fuzziness and the like? *Fuzzy sets and systems*, Vol. 24, S. 141-160.

KRAUS K. und SCHNEIDER W. (1988): *Fernerkundung Band 1: Physikalische Grundlagen und Aufnahmetechniken*. Dümmler Verlag, Bonn.

KRAUS K. mit Beiträgen von JANSJA J. und SCHNEIDER W. (1990): *Fernerkundung Band 2: Auswertung photographischer und digitaler Bilder*. Dümmler Verlag, Bonn.

KAGER H., KRAUS K. und STEINNOCHER K. (1992): Photogrammetrie und digitale Bildverarbeitung angewandt auf den Behaim-Globus. *Zeitschrift für Photogrammetrie und Fernerkundung*, Vol. 60, No. 5, S. 142-148.

KONTOES C.C., ROKOS D., WILKINSON G.G. und MÉGIER J. (1991): The use of expert system and supervised relaxation techniques to improve SPOT image classification using spatial context. *Proceedings International Conference IEEE Geoscience and Remote Sensing Society*, 3.-7. Juni 1991. Espoo, Finnland, S. 1855-1858.

LELEWER D.A. und HIRSCHBERG D.S. (1987): Data Compression. *ACM Computing Surveys*, Vol. 19, No. 3, S. 261-296.

MATTOS P. (1987): Program design for concurrent systems. *Technical note 5*, INMOS, Bristol.

- MAY D. (1987): Communicating processes and occam. *Technical note 20*, INMOS, Bristol.
- MEHLDAU G. und SCHOWENGERDT R.A. (1990): A C-extension for rule-based image classification systems. *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 6, S. 887-892.
- MEUER H. und STROHMAIER E. (1990): Aktuelle Parallelrechner-Konzepte und Architekturen. *Informationstechnik*, Vol. 34, No. 1, S. 17-27.
- MIHATSCH A. und KILLE K. (1990): Transputer in der Bildverarbeitung. *Vision and Voice Magazine*, Vol. 4, No. 3, S. 225-227.
- MØLLER-JENSEN L. (1990): Knowledge-based classification of an urban area using texture and context information in Landsat-TM imagery. *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No. 6, S. 899-904.
- ZUR NEDDEN H. (1992): Squeeze, LZH & Co. Kompressionsverfahren implementiert. *c't*, 7/92, S. 231-238.
- PYKA K. und STEINNOCHER K. (1994): Auswahl eines optimalen Datensatzes für die multispektrale Klassifizierung unter Einbeziehung von Texturmerkmalsbildern. *Zeitschrift für Photogrammetrie und Fernerkundung*, in Druck.
- RYERSON R. (1989): Image Interpretation Concerns for the 1990s and Lessons from the Past. *Photogrammetric Engineering and Remote Sensing*, Vol. 55, No. 10, S. 1427-1430.
- SABINS F.S. (1987): *Remote Sensing - principles and interpretation*. Freeman, San Francisco.
- SALI E. und WOLFSON H. (1992): Texture classification in aerial photographs and satellite data. *International Journal of Remote Sensing*, Vol. 13, No. 18, S. 3395-3408.
- SINDHUBER A. (1993): Der Beitrag hochauflösender kosmischer Photoaufnahmen mit der Kamera KFA-1000 zur Landnutzungsanalyse in sensiblen Naturräumen - Fallbeispiel Nationalpark Neusiedlersee. *Diplomarbeit, Technisch-naturwissenschaftliche Fakultät, Technische Universität Wien*.
- SRINIVASAN A. und RICHARDS J.A. (1990): Knowledge-based methods for multi-source classification. *International Journal of Remote Sensing*, Vol. 11, No. 11, S. 505-525.
- STEINNOCHER K., STAUFER P. UND FRANZEN M. (1993): Landnutzungsdaten zur Modellierung zellulärer Mobilfunknetze: Der integrative Einsatz digitaler Bildverarbeitungstechniken und geographischer Informationssysteme zur Erfassung urbaner Strukturen. *Proceedings AGIT V*, 7.-9. Juli 1993, Salzburg, (=Salzburger Geographische Materialien, Heft 20), S. 307-318.
- WANG Y. und CIVCO D.L. (1992): Post-classification of misclassified pixels by evidential reasoning: a GIS approach for improving classification accuracy of remote sensing data. *International Archives of Photogrammetry and Remote Sensing*, Vol. 29, B 7, Commission VII. S. 80-86.

WEBB J.A. (1992): Steps Toward Architecture-Independent Image Processing. *Computer*, Vol. 25, No. 2, S. 21-31.

WEBSTER C.J. und BRACKEN I.J. (1992): Exploring the discrimination power of texture in urban image analysis. *International Archives of Photogrammetry and Remote Sensing*, Vol. 29, B 7, Commission VII. S. 942-948.

WILKINSON G.G. und BURRILL A. (1991): Integration of GIS-derived spatial data with satellite imagery for improved land cover mapping using an expert system. *Proceedings of the second European Conference on Geographical Information Systems EGIS*, 2.-5. April 1991, Brüssel, S. 1241-1249.

ZHANG Z., SHIMODA H., FUKUE K., MATSUMAE Y. UND SAKATA T. (1988): New classification algorithms using spatial information for high resolution image data. *International Archives of Photogrammetry and Remote Sensing*, Vol. 27, B 7, Commission VII, S. 778-785.

ZIMMERMANN H.J. (1993): *Fuzzy Technologien - Prinzipien, Werkzeuge, Potentiale*. VDI-Verlag, Düsseldorf.

dtv-Brockhaus-Lexikon, Band 9, S.327. Deutscher Taschenbuch Verlag, München, 1984.

Wahrnehmung und visuelles System. Spektrum der Wissenschaft: Verständliche Forschung. Heidelberg, 1986.

WEBB J.A. (1992): Steps Toward Architecture-Independent Image Processing. *Computer*, Vol. 25, No. 2, S. 21-31.

WEBSTER C.J. und BRACKEN I.J. (1992): Exploring the discrimination power of texture in urban image analysis. *International Archives of Photogrammetry and Remote Sensing*, Vol. 29, B 7, Commission VII. S. 942-948.

WILKINSON G.G. und BURRILL A. (1991): Integration of GIS-derived spatial data with satellite imagery for improved land cover mapping using an expert system. *Proceedings of the second European Conference on Geographical Information Systems EGIS*, 2.-5. April 1991, Brüssel, S. 1241-1249.

ZHANG Z., SHIMODA H., FUKUE K., MATSUMAE Y. UND SAKATA T. (1988): New classification algorithms using spatial information for high resolution image data. *International Archives of Photogrammetry and Remote Sensing*, Vol. 27, B 7, Commission VII, S. 778-785.

ZIMMERMANN H.J. (1993): *Fuzzy Technologien - Prinzipien, Werkzeuge, Potentiale*. VDI-Verlag, Düsseldorf.

dtv-Brockhaus-Lexikon, Band 9, S.327. Deutscher Taschenbuch Verlag, München, 1984.

Wahrnehmung und visuelles System. Spektrum der Wissenschaft: Verständliche Forschung. Heidelberg, 1986.

## Lebenslauf

Geboren am 26.6.1962 in Wien

Österreichischer Staatsbürger

1968 - 1972	4 Klassen Volksschule
1972 - 1980	8 Klassen Gymnasium (neusprachlich)
Juni 1980	Matura
Okt. 1980 - Mai 1981	Ableistung des Präsenzdienstes
1981 - 1988	Studium des Vermessungswesen an der TU-Wien
Jän. 1985	Erste Diplomprüfung
Nov. 1988	Zweite Diplomprüfung Sponson zum Diplomingenieur für Vermessungswesen
Jän. - Sept. 1989	Studium und Forschungstätigkeit an der Ohio State University, Columbus, Ohio
Nov. 1989 - Okt. 1993	Vertragsassistent am Institut für Photogrammetrie und Fernerkundung der TU-Wien
Seit April 1993	Wissenschaftlicher Angestellter im Österreichischen Forschungszentrum Seibersdorf, Hauptabteilung Umweltplanung

Bisher erschienen:

- Heft 1 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1970 - 1973, Dezember 1973.
- Heft 2 EGGER-PERDICH-PLACH-WAGENSOMMERER, Taschenrechner HP 45 und HP 65, Programme und Anwendungen im Vermessungswesen, 1. Auflage, März 1974, Special Edition in English, Juli 1974, 2. verbesserte Auflage, November 1974.
- Heft 3 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1973 - 1974, September 1974.
- Heft 4 EGGER-PALFINGER-PERDICH-PLACH-WAGENSOMMERER, Tektronix-Tischrechner TEK 31, Programmbibliothek für den Einsatz im Vermessungswesen, November 1974.
- Heft 5 K. LEDERSTEGE, Die horizontale Isostasie und das isostatische Geoid, Februar 1975.
- Heft 6 F. REINHART, Katalog von FK4 Horrebow-Paaren für Breiten von +30 bis +60, Oktober 1975.
- Heft 7 Arbeiten aus dem Institut für Höhere Geodäsie, Wien, Dezember 1975.
- Heft 8 Veröffentlichungen des Instituts für Photogrammetrie zum XIII. Internationalen Kongreß für Photogrammetrie in Helsinki 1976, Wien, Juli 1976.
- Heft 9 W. PILLEWIZER, Felsdarstellung aus Orthophotos, Wien, Juni 1976.
- Heft 10 PERDICH-PLACH-WAGENSOMMERER, Der Einsatz des programmierbaren Taschenrechners Texas Instruments SR-52 mit Drucker PC100 in der ingenieurgeodätischen Rechentechnik, Wien, Mai 1976.
- Heft 11 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1974 - 1976, November 1976.
- Heft 12 Kartographische Vorträge der Geodätischen Informationstage 1976, Wien, Mai 1977.
- Heft 13 Veröffentlichung des Instituts für Photogrammetrie anlässlich des 80. Geburtstages von Prof. Dr.h.c. K. Neumaier, Wien, Januar 1978.
- Heft 14 L. MOLNAR, Self Checking Analytical Relative Orientation and Strip Formation, Wien, Dezember 1978.
- Heft 15 Veröffentlichung des Instituts für Landesvermessung anlässlich des 80. Geburtstages von Prof. Dr. Alois Bavir, Wien, Januar 1979.
- Heft 16 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1976 - 1979, Wien, November 1979.
- Heft 17 E. VOZIKIS, Die photographische Differentialumbildung gekrümmter Flächen mit Beispielen aus der Architekturbildmessung, Wien, Dezember 1979.
- Heft 18 Veröffentlichung des Instituts für Allgemeine Geodäsie anlässlich des 75. Geburtstages von Prof. Dipl. Ing. Dr. F. Hauer, Die Höhe des Großglockners, Wien, 1981.
- Heft 19 H. KAGER, Bündeltriangulation mit indirekt beobachteten Kreiszentren, Wien, April 1981.
- Heft 20 Kartographische Vorträge der Geodätischen Informationstage 1980, Wien, Mai 1982.
- Heft 21 Veröffentlichung des Instituts für Kartographie anlässlich des 70. Geburtstages von Prof. Dr. Wolfgang Pillewizer: Glaziologie und Kartographie, Wien, Dezember 1982.

- Heft 22 K. TEMPFLI, Genauigkeitsschätzung digitaler Höhenmodelle mittels Spektralanalyse, Wien, Mai 1982.
- Heft 23 E. CSAPLOVICS, Interpretation von Farbinfrarotbildern, Wien, November 1982.
- Heft 24 J. JANSKA, Rektifizierung von Multispektral-Scanneraufnahmen - Entwicklung und Erprobung eines EDV-Programms, Wien, Mai 1983.
- Heft 25 Zusammenfassungen der Diplomarbeiten, Dissertationen und Habilitationen an den geodätischen Instituten der TU Wien, Wien, November 1984.
- Heft 26 T. WUNDERLICH, Die voraussetzungsfree Bestimmung von Refraktionswinkeln, Wien, August 1985.
- Heft 27 G. GERSTBACH (Hrsg.), Geowissenschaftliche/geotechnische Daten in Landinformations-systemen - Bedarf und Möglichkeiten in Österreich, Wien, Juni 1986.
- Heft 28 K. NOVAK, Orientierung von Amateuraufnahmen ohne Paßpunkte, Wien, August 1986.
- Heft 29 Veröffentlichung des Instituts für Landesvermessung und Ingenieurgeodäsie, Abteilung Ingenieurgeodäsie anlässlich des 80. Geburtstagstages von Prof. Dipl. Ing. Dr. F. Hauer, Wien, Oktober 1986.
- Heft 30 K.-H. ROCH, Über die Bedeutung dynamisch ermittelter Parameter für die Bestimmung von Gesteins- und Gebirgseigenschaften, Wien, Februar 1987.
- Heft 31 G. HE, Bildverbesserung mittels digitaler Filterung, Wien, April 1989.
- Heft 32 F. SCHLÖGELHOFER, Qualitäts- und Wirtschaftlichkeitsmodelle für die Ingenieurphotogrammetrie, Wien, April 1989.
- Heft 33 G. GERSTBACH (Hrsg.), Geowissenschaftliche/geotechnische Daten in Landinformations-systemen - Datenbestände und Datenaustausch in Österreich, Wien, Juni 1989.
- Heft 34 F. HOCHSTÖGER, Ein Beitrag zur Anwendung und Visualisierung digitaler Geländemodelle, Wien, Dezember 1989.
- Heft 35 R. WEBER, Lokale Schwerefeldmodellierung unter Berücksichtigung spektraler Methoden zur Geländereduktion, Wien, April 1990.
- Heft 36 o.Prof. Dr. Hans Schmid zum 70. Geburtstag. Veröffentlichung der Abteilung für Landesvermessung, Wien, Oktober 1990.
- Heft 37 G. GERSTBACH, H.P. HÖLLRIEGL und R. WEBER, Geowissenschaftliche Informationsbörse - Eine Nachlese zu GeoLIS II, Wien, Oktober 1990.
- Heft 38 R. ECKER, Rastergraphische Visualisierungen mittels digitaler Geländemodelle, Wien, August 1991.
- Heft 39 Kartographische Forschungen und Anwendungsorientierte Entwicklungen, herausgegeben von W. Stams und F. Kelnhofer zum 80. Geburtstag von Prof. Dr. W. Pillewizer, Wien, Juli 1991.
- Heft39a W. RIEGER, Hydrologische Anwendungen des digitalen Geländemodells, Wien, Juli 1992.
- Heft 40 K. STEINNOCHER, Methodische Erweiterungen der Landnutzungsklassifikation und Implementierung auf einem Transputernetzwerk, Wien, Juli 1994.
- Heft 41 G. FORKERT, die Lösung photogrammetrischer Orientierungs- und Rekonstruktionsaufgaben mittels allgemeiner kurvenförmiger Elemente, Wien, Juli 1994.