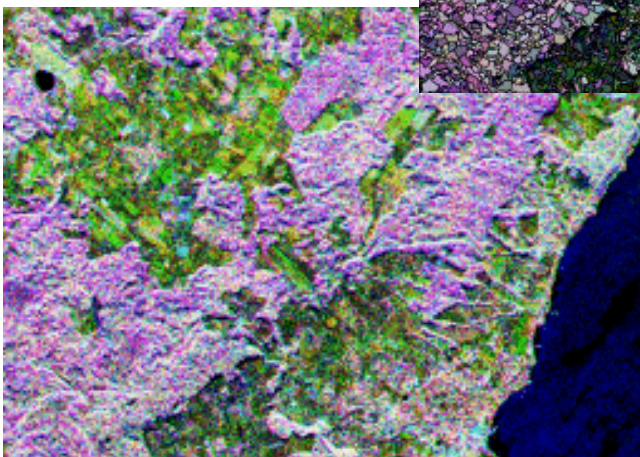
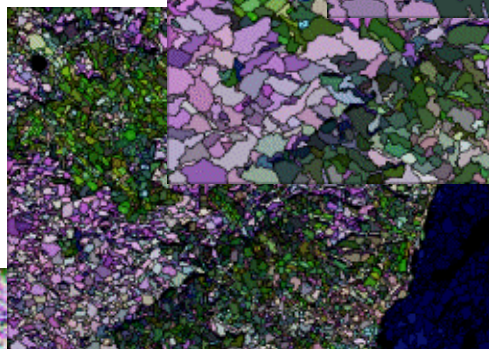
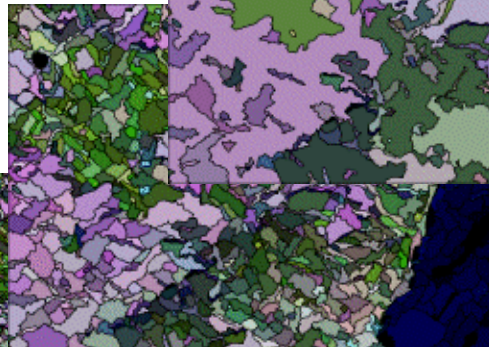
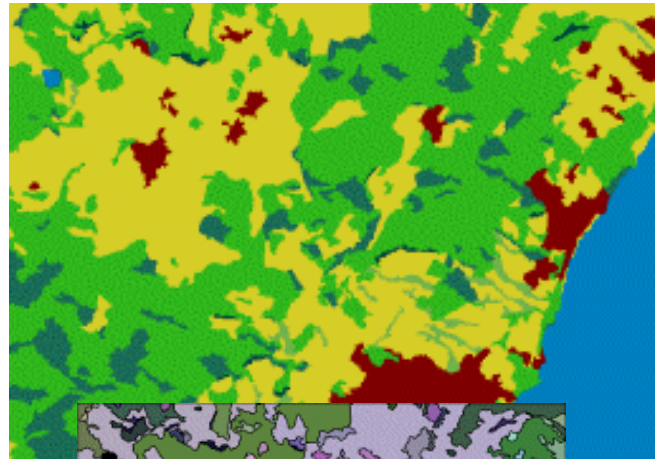


# UNIVERSITÄT HANNOVER

Institut für Photogrammetrie und Ingenieurvermessung

DIPLOMARBEIT

„Untersuchung des  
Bildanalyse-Systems  
*eCognition*“



Vorgelegt von  
Ralph Schmidt  
Januar 2000

# Inhalt

---

1	Einleitung .....	1
2	Grundlagen der Klassifikation .....	3
2.1	Einleitung.....	3
2.2	Multispektrale Klassifikation und Merkmalsräume .....	4
2.3	Klassifikationsverfahren.....	5
2.3.1	Überwachte und unüberwachte Klassifikation.....	5
2.3.2	Nearest Neighbour .....	9
2.3.3	Box-Klassifikator .....	10
2.3.4	Maximum Likelihood .....	11
2.3.5	Klassifikation mit neuronalen Netzwerken.....	12
2.4	Zusätzliche und modifizierte Kanäle.....	15
2.5	Bildsegmentierung.....	17
3	Das Bildanalyzesystem <i>eCognition</i> .....	20
3.1	Grundkonzept.....	20
3.2	Aufbau des Programms.....	22
3.3	Multiresolution Segmentation .....	25
3.4	Klassifikation .....	28
3.5	Klassifikatoren .....	34
3.5.1	Zugehörigkeitsfunktionen .....	34
3.5.2	Nearest Neighbour .....	36
3.6	Wissensbasierte Segmentierung.....	40
3.6.1	Knowledge based Fusion.....	41
3.6.2	Knowledge based Border Optimization.....	42
4	Anwendung des Bildanalyzesystems <i>eCognition</i> .....	44
4.1	Zielsetzung, Datenmaterial und Rechnerausstattung.....	44
4.2	Anwendung der Segmentierung .....	46

4.2.1	Allgemeine Vorgehensweise .....	46
4.2.2	Einstellen der Parameter.....	47
4.2.3	Segmentierung von panchromatischen Bildern.....	54
4.2.4	Verwendung zusätzlicher Kanäle .....	56
4.2.5	Segmentierung von Bildern anderer Auflösungsstufen .....	57
4.3	Anwendung der Klassifikation .....	60
4.3.1	Allgemeine Vorgehensweise.....	60
4.3.2	Klassifikation des Testgebietes.....	61
4.3.3	Klassifikation mit panchromatischen Bildern .....	69
4.3.4	Klassifikation mit Texturkanal.....	71
4.3.5	Texturanalyse mit <i>eCognition</i> .....	72
4.3.6	Klassifikation von Bildern anderer Auflösungsstufen.....	75
4.4	Übertragung der Klassenbeschreibungen auf andere Bilder .....	77
4.5	Vergleich mit anderen Klassifikationsverfahren.....	83
5	Zusammenfassung.....	86
6	Literaturverzeichnis .....	90
7	Anhang.....	93
7.1	Anbausorten .....	93
7.2	Segmentierung .....	94
7.3	Klassifikation .....	102
7.3.1	Selektive Klassifikation.....	102
7.3.2	Klassifikation mit einer einzigen Klassenhierarchie.....	110
7.3.3	Andere Auflösungen.....	114

# 1 Einleitung

---

Eine thematische Karte dient der Darstellung von abstrakten, raumbezogenen Sachverhalten und nicht der des konkreten Erscheinungsbildes der Landschaft. Mit der Bildanalyse von Fernerkundungsdaten durch Klassifikation ist es möglich, effizient große Datenmengen durch Automatisierung zu verarbeiten und in thematischen Karten darzustellen. Das Ergebnis einer Klassifikation kann außer einer physikalischen auch eine digitale Karte zur Integration in ein geographisches Informationssystem (GIS) sein. Anwendungen für thematische Karten finden sich in Bereichen wie Forst- und Landwirtschaft, Versorgungsunternehmen, Transport, Freizeit, Militär, Umwelt, Forschung und Unterricht. Die Themen reichen z.B. von einer Unterscheidung zwischen bebautem und unbebautem Gebiet für eine generelle Beschreibung einer Landschaft bis zur detaillierten Bestimmung von einzelnen Pflanzen zur Schadens- oder Ertragsanalyse.

Bei der Auswahl der Kategorien in der Karte muß gewährleistet sein, daß diese auch im vorhandenen Bildmaterial zu unterscheiden sind. Konventionelle Softwarelösungen basieren auf der multispektralen Klassifikation, d.h. die verschiedenen Objektklassen mit unterschiedlichen Reflexionseigenschaften werden an Hand von spektralen Zugehörigkeiten gebildet, da in den meisten Fällen die Fernerkundungsdaten in mehreren Kanälen mit einer Auflösung von z.B. jeweils 8 Bit vorliegen. Diese Art der Klassifikation eignet sich besonders für nicht sehr hoch auflösende Satellitensysteme (Landsat MSS) bzw. für kleine Maßstäbe, da die entscheidenden Informationen hier die Spektralwerte der einzelnen Pixel liefern, da nicht sehr viele Details im Bild erkennbar sind. Die örtlichen nachbarschaftlichen Beziehungen der einzelnen Bildelemente und die Formen von Objekten werden dabei allerdings vernachlässigt. In großmaßstäbigen Fernerkundungsdaten wie z.B. hochauflösenden Satellitenbildern oder Luftbildern sind die hier sichtbar werdenden charakteristischen geometrischen Formen bedeutende Informationsträger. Hierbei ist zu beachten, daß es für Menschen geradezu trivial ist, Gegenstände wie z.B. ein Haus von einer Straße an Hand von Formen, topologischen Zusammenhängen und Erfahrungen im Bild zu unterscheiden. Die Farbinformation liefert hier nur einen kleinen Teil an

Informationen. Ein Mensch kann auch ein Schwarzweißbild gegenüber einem Farbbild ohne große Einschränkungen interpretieren. Für bisherige Systeme ist es allerdings zunächst unmöglich, solche Kontextbezüge zu erfassen, da sie nur die spektrale Information des einzelnen Pixels auswerten. Deshalb wird versucht, in den Systemen Algorithmen zu implementieren, die topologische Gesetzmäßigkeiten und geometrische Formen berücksichtigen, um die multispektrale Klassifikation zu unterstützen.

Die Firma *Delphi2 Creative Technologies* hat auf der Basis von Erkenntnissen über die menschliche visuelle Wahrnehmung ein Programm entwickelt, das sich von reinen pixelbasierten Methoden löst und objektorientiert klassifiziert. Das Konzept der Software *eCognition* basiert auf der Fractal Net Evolution-Technik, die von Gerd Binnig bei *Delphi2* entwickelt wurde. Hierbei werden mit Hilfe von semantischen Netzen komplexe Sachverhalte dargestellt. In der vorliegenden Diplomarbeit soll nun die Leistungsfähigkeit der Software untersucht und ein Vergleich zu klassischen pixelbasierten Methoden durchgeführt werden.

## 2 Grundlagen der Klassifikation

---

### 2.1 Einleitung

Bei einer Klassifikation sollen die im vorliegenden Bild abgebildeten Objektflächen erkannt und Objekttypen zugeordnet werden, d.h. die Flächen der zu bildenden Objektklassen sollen gegeneinander abgegrenzt werden. Eine Klassifikation umfaßt herkömmlich mehrere Schritte:

- *Vorverarbeitung*: Zur Vorverarbeitung gehören zunächst die zahlreichen Bildverarbeitungsprozesse. Zuerst werden z.B. Bildstörungen je nach Art mit Median- oder Sigmafiltern beseitigt. Andere Filter z.B. zur Kantenextraktion, Grauwertanpassungen, radiometrische Korrekturen und Kontrastverbesserungen folgen. Gegebenenfalls ist eine Mosaikbildung notwendig, um das gewünschte Gebiet in einem zusammenhängenden Bild zu erhalten. Außerdem geht einer Klassifikation in der Regel eine geometrische Rektifizierung voraus, damit das Ergebnis als Karte Verwendung finden kann. Sämtliche Schritte sind allerdings optional, d.h. es kann auch im Originalbild klassifiziert werden.
- *Training*: Bei diesem Vorgang werden Pixel aus dem Bild extrahiert, die zur Anlernung (Training) des Klassifikationsverfahrens nötig bzw. sinnvoll sind, um Funktionen zur Unterscheidung (Diskriminator) der einzelnen Objektklassen zu bestimmen. Dieser Prozeß kann entweder vom Auswerter überwacht werden oder unüberwacht erfolgen (2.3.1).
- *Zuweisung zu Klassen*: Jetzt wird für jedes Bildelement der Diskriminator angewendet, und es wird der entsprechenden Objektklasse zugeordnet. Bei der überwachten Klassifikation ergeben sich die Klassen direkt aus den Diskriminatoren. Erfolgte die Klassifikation allerdings unüberwacht, so muß der Auswerter das Ergebnis noch nachbearbeiten.

Mit dem Endergebnis lässt sich nun eine thematische Karte erstellen, indem man zusätzlich Legenden und Beschriftungen einfügt, oder die Daten werden in ein GIS importiert. Eine Klassifikation dient also einer sinnvollen Datenreduktion, um dem Betrachter die Analyse des Bildes zu erleichtern und raumbezogene Informationen anschaulich darzustellen.

## 2.2 Multispektrale Klassifikation und Merkmalsräume

Liegen die Fernerkundungsdaten in mehreren Kanälen vor, so spricht man von einer multispektralen Klassifikation. Auch in einkanaligen Bildern (panchromatische Bilder) kann eine Klassifikation vorgenommen werden, indem man die Grauwerte nach dem Äquidensitenverfahren (Abb. 1) durch Schwellwerte voneinander trennt (Kraus 1990). Aufgrund der starken Streuung innerhalb einer Objektklasse ist die Klassifikation in einem eindimensionalen Merkmalsraum allerdings sehr fehlerbehaftet (Abb. 2).

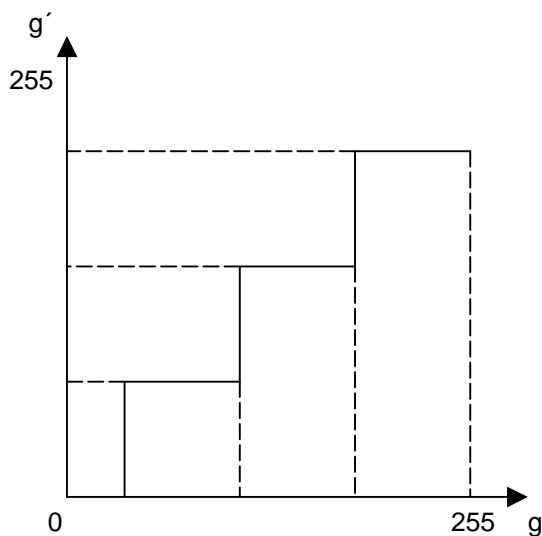


Abb. 1: Übertragungsfunktion für die Äquidensiten.

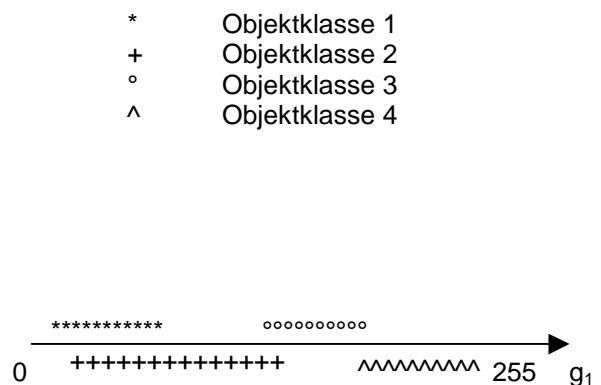


Abb. 2: Grauwert-Verteilung der Bildelemente von vier Objektklassen.

Die Voraussetzungen für eine Klassifikation sind in einem zweidimensionalen Merkmalsraum wesentlich besser, da nun im Idealfall Punktwolken (Cluster) entstehen, die einzelne Objektklassen repräsentieren (Abb. 3). Allerdings können sich auch hier die Punktwolken überschneiden, wenn die Streuung in den einzelnen

Klassen sehr hoch ist oder die Mittelpunkte der Cluster nicht weit genug voneinander entfernt liegen. Fügt man nun zusätzliche Kanäle hinzu, so erhält man z.B. abstrakte 3-, 4- oder 5-dimensionale Merkmalsräume, in denen die verschiedenen Objektklassen eventuell komplett voneinander abgegrenzt sind, und so eine eindeutige Klassifikation möglich wird. Die sieben Kanäle von einem Landsat TM Bild spannen z.B. einen 7-dimensionalen Merkmalsraum auf.

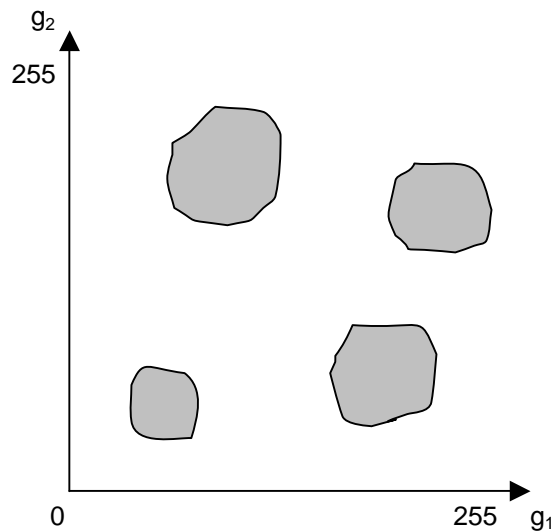


Abb. 3: Grauwertverteilung der Bildelemente von vier Objektklassen in zwei Kanälen.

## 2.3 Klassifikationsverfahren

### 2.3.1 Überwachte und unüberwachte Klassifikation

Bei der Klassifikation gibt es zwei grundsätzliche Vorgehensweisen. Bei der überwachten Klassifikation definiert man für jede vorhandene Objektklasse ein oder mehrere Testgebiete (Trainingsgebiete). Aufgrund dieser bekannten Stichproben wird dann der gesamte Bildinhalt bestimmt. Der Vorteil der überwachten Klassifikation liegt darin, daß man die für den jeweiligen Sachverhalt erforderlichen Objektklassen selbst bestimmen kann. Bei der Auswahl der Trainingsgebiete muß darauf geachtet werden, daß die Auswahl einerseits homogen und ein repräsentatives Beispiel für die jeweilige Objektklasse ist, und daß andererseits auch die



Streuung innerhalb der Klasse berücksichtigt wird (Abb. 4). Eine weitere Schwierigkeit besteht darin, daß Kenntnis über den Inhalt des Gebietes (Ground Truth) vorhanden sein muß. Diese Information kann man z.B. aus dem Bild selbst, aus Karten oder an Hand einer Gebietsbegehung gewinnen. Diese Aufgabe kann sehr arbeitsintensiv und teuer oder auch unmöglich sein.

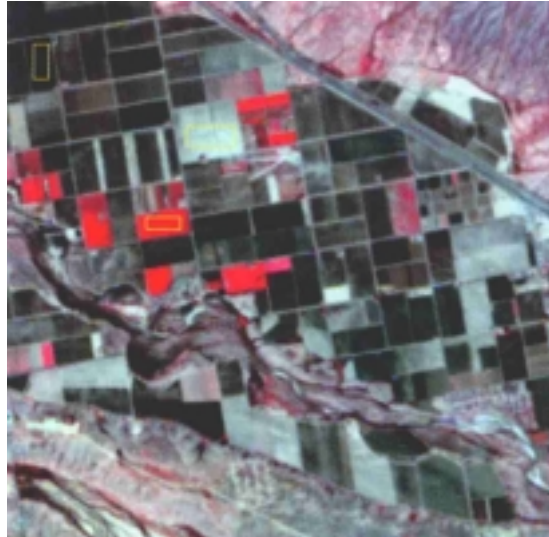


Abb. 4: Trainingsgebiete für die drei Klassen dunkler Boden, heller Boden und Getreide in einem Landsat TM Bild (Schowengerdt 1997).

Liegen keine Referenzdaten vor, so muß die Klassifikation ohne Bodenbeobachtung erfolgen, und die Einteilung der Klassen wird an Hand der Grauwerte durch Algorithmen vorgenommen. Der Algorithmus untersucht dabei das ganze oder Teile des Bildes und lokalisiert Ansammlungen von Pixeln im Merkmalsraum. Diese Cluster bilden dann die einzelnen Objektgruppen, die allerdings noch unidentifiziert sind. Die Benennung der Klassen geschieht hinterher. Der Nachteil dieser Methode ist, daß hierbei Klassen gebildet werden, die nicht vom Auswerter gewünscht sein können.

Ein Beispiel für eine Implementierung ist der K Means Algorithmus, wobei das K hier nicht für die Anzahl der Kanäle, sondern für die Anzahl der Cluster steht. In der Ausgangssituation wird für jeden vorhandenen Cluster ein beliebiger Initialisierungsvektor gebildet, aus denen dann die mittleren Vektoren<sup>1</sup> der Objektklassen (Schwer-

---

<sup>1</sup> Die Berechnung eines mittleren Vektors bzw. eines Schwerpunktes wird in (2.3.2) beschrieben.

punkte) entstehen sollen (Abb. 5) (Schowengerdt 1997). Jedes Pixel wird nun der Klasse zugeordnet, zu deren mittleren Vektor es am nächsten liegt.

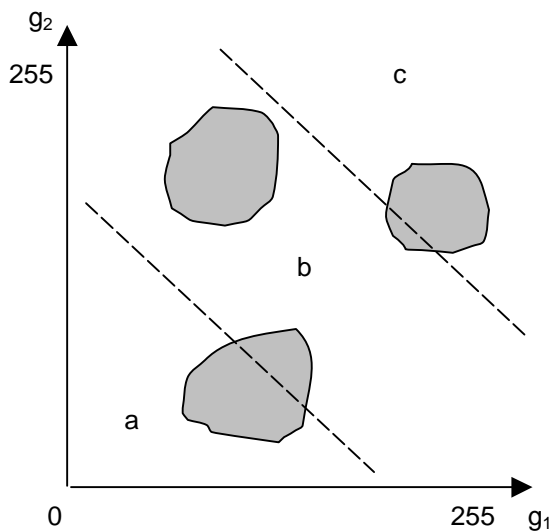


Abb. 5: Ausgangssituation für Datenmaterial mit drei Clustern. Die Initialisierungsvektoren verteilen sich äquidistant diagonal über den Merkmalsraum.

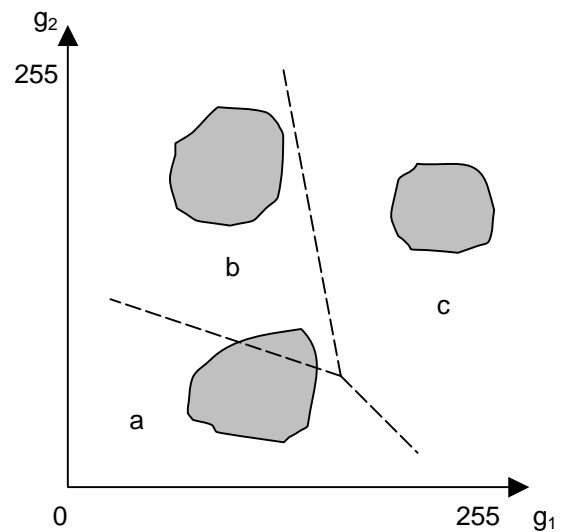


Abb. 6: Festlegung der Diskriminatoren durch den K Means Algorithmus nach der ersten Iteration.

Danach werden die Schwerpunkte neu berechnet, und somit liegen die ersten Diskriminatoren fest; die nächste Iteration kann beginnen (Abb. 6). Der K Means Algorithmus wird sich dabei den im Merkmalsraum vorhandenen Konzentrationen immer mehr annähern (Abb. 7 u. Abb. 8).

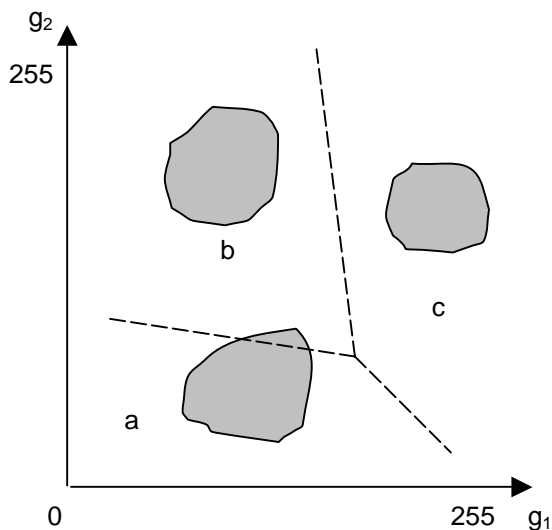


Abb. 7: Verteilung der Diskriminatoren nach der zweiten Iteration.

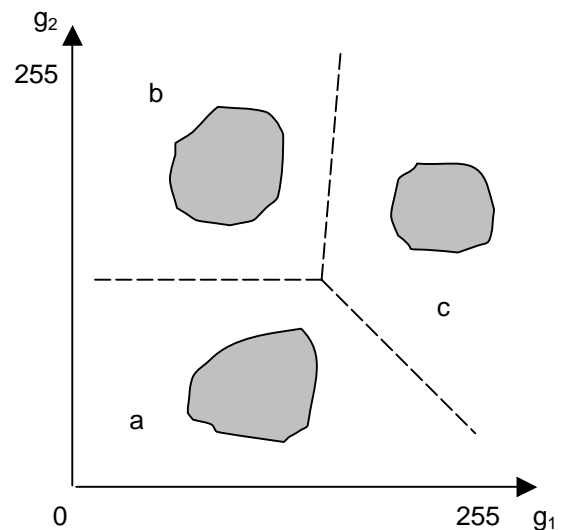


Abb. 8: Endgültige Verteilung der Diskriminatoren.

Die Iteration sollte so lange vorangetrieben werden, bis keine signifikante Änderung in der Pixelzuweisung mehr stattfindet. Ein Abbruchkriterium für die Iteration kann mit folgender Formel definiert werden (*Schowengerdt 1997*).

$$\Delta\mu(t) = \sum_{k=1}^K |\mu_k(t) - \mu_k(t-1)| \quad (1)$$

$\Delta\mu$  : Größenänderung des mittleren Vektors  
 $t$  : Iterationsschritt  
 $K$  : Anzahl der Cluster  
 $k$  : Zählindex für den jeweiligen Kanal

Hierbei handelt es sich um die Größenänderung des mittleren Vektors von einer Iteration  $t$  zur nächsten, die über alle  $K$  Cluster aufsummiert wird. Unterschreitet die Größenänderung  $\Delta\mu(t)$  einen bestimmten Wert, so wird die Iteration abgebrochen. Die erhaltenen endgültigen Schwerpunkte können sogleich für eine Klassifikation nach Nearest Neighbour (2.3.2) benutzt werden.

Das Problem bei dieser Vorgehensweise ist, daß die Pixelverteilung im Merkmalsraum in der Realität nicht derartige abgegrenzte Cluster ausbilden, wie es in Abb. 5 bis Abb. 7 dargestellt wird. Weiterhin muß die Festlegung der Anzahl der Cluster eventuell iterativ nach Beurteilung durch den Auswerter erfolgen. Ist die Anzahl zu gering, so bilden sich für manche Objektklassen keine eigenen Cluster heraus. Die Zahl der Cluster muß so lange erhöht werden, bis sich die gewünschten Objektklassen in einem eigenen Cluster abzeichnen.

Da die unüberwachte Klassifikation für den Auswerter nicht unbedingt sinnvolle Klassen bildet, und die überwachte Klassifikation nicht in jedem Fall Klassensignaturen ausbildet, die numerisch im Merkmalsraum voneinander zu trennen sind, kann man zu einer kombinierten Lösung übergehen. Dabei wird zunächst mit einer unüberwachten Klassifikation mit einer ausreichend hohen Anzahl von z.B. 50 Klassen begonnen. Der Auswerter muß nun das Ergebnis mit den Referenzdaten vergleichen, um den Klassen eine Bedeutung zuzuweisen. Hierbei müssen Cluster möglicherweise miteinander verbunden oder einzelne weiter unterteilt werden. Anschließend kann für die endgültige Karte eine überwachte Klassifikation durchgeführt werden.

### 2.3.2 Nearest Neighbour

Dieser Klassifikator wird auch Minimalentfernung oder in der englischsprachigen Literatur ebenso Minimum Distance genannt. Hierbei werden die unbekanntes Bildelemente den nächstliegenden Trainingspixeln zugeordnet (*Kraus 1990*).

$$\bar{g}_k^i = \frac{1}{N} \sum_{n=1}^N g_{k,n}^i \quad (2)$$

$k$  : Index für den jeweiligen Kanal, der von 1 bis  $K$  läuft

$N$  : Anzahl der Bildelemente im Trainingsgebiet mit Zählindex  $n$

$i$  : Index für die jeweilige Objektklasse

Dazu mittelt man zunächst für jedes Trainingsgebiet der jeweiligen Objektklasse die Grauwerte der einzelnen Kanäle (2) (*Schowengerdt 1997*). Man erhält damit für jede Objektklasse einen Schwerpunktsvektor im Merkmalsraum, der sich aus den Mittelwerten des Trainingsgebietes der insgesamt  $K$  Kanäle zusammensetzt ( $\bar{\underline{g}}^i = [g_1^i \ g_2^i \ \dots \ g_k^i \ \dots \ g_K^i]$ ). Für das zu klassifizierende Bildelement wird nun die Entfernung zu jedem Schwerpunkt berechnet. In einem Bild mit drei Kanälen entspricht dies einer räumlichen Strecke. Das Pixel wird jetzt der Objektklasse zugeteilt, zu deren Mittelpunkt die Entfernung am geringsten ist. Die Diskriminatoren sind im zweidimensionalen Fall Geraden, im dreidimensionalen Fall Ebenen und in höherdimensionalen Merkmalsräumen Hyperebenen. Diese Berechnung wird für jedes unbekanntes Pixel ausgeführt, und die Algorithmen arbeiten relativ schnell. Dagegen werden die statistischen Eigenschaften der Punktwolken, d.h. deren Kovarianzmatrix, nicht berücksichtigt, obwohl sie unterschiedliche Streuungen aufweisen können. Ferner ist zu erwähnen, daß bereits ein Trainingspixel ausreicht, um diese Methode anzuwenden.

### 2.3.3 Box-Klassifikator

Wie bei dem Nearest Neighbour Verfahren (2.3.2) werden auch hier die Schwerpunkte der einzelnen Objektklassen berechnet. Um diesen Mittelpunkt legt man dann ein  $K$ -dimensionales Rechteck, dessen Kantenlängen einem Vielfachen  $m$  der Standardabweichungen der Kanalmittelwerte (3) entsprechen (Kraus 1990).

$$\sigma_k^i = \sqrt{\frac{\sum_{n=1}^N (g_{k,n}^i - \bar{g}_{k,n}^i)^2}{N-1}} \quad (3)$$

Bezeichnungen entsprechen Formel (2)

Ein zu klassifizierendes Bildelement wird derjenigen Objektklasse zugewiesen, innerhalb deren Rechteck es liegt (Abb. 9). Fällt das Pixel in kein Rechteck, so bleibt es unklassifiziert und wird einer Zurückweisungsklasse zugeschrieben. Je größer man den Faktor  $m$  wählt, desto weniger Pixel fallen in die Zurückweisungsklasse. Nun kann es allerdings passieren, daß sich die Rechtecke von benachbarten Objektklassen überschneiden. Bei solchen Konflikten muß die Entscheidung z.B. mit Hilfe des Nearest Neighbour Klassifikators getroffen werden.

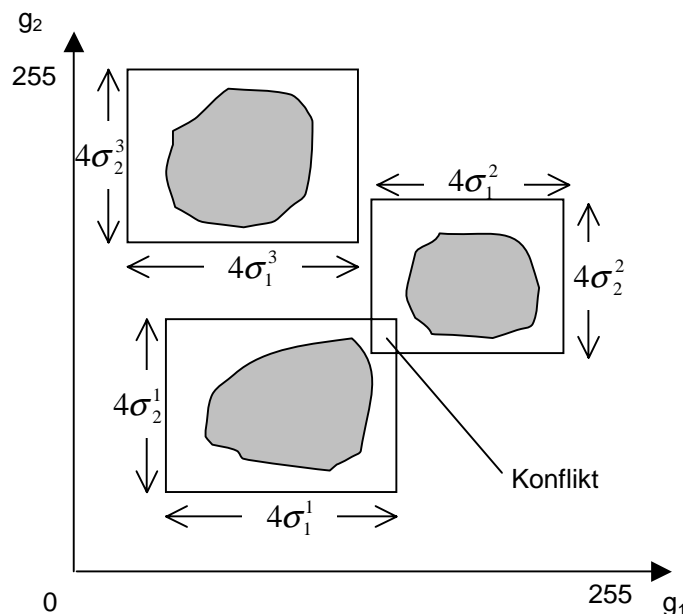


Abb. 9: Der Box-Klassifikator für die Trainingsgebiete von drei Objektklassen mit der vierfachen Standardabweichung ( $m = 4$ ) und einem Konflikt zwischen den Objektklassen 1 und 2.

Der Vorteil dieser Methode liegt in der sehr einfachen Berechnung. Aber auch hier können die Korrelationen nicht berücksichtigt werden, so daß sie vorher durch eine Hauptkomponententransformation (2.4) beseitigt werden könnten.

### 2.3.4 Maximum Likelihood

Bei der Klassifikation mittels Maximum Likelihood (maximaler Wahrscheinlichkeit) wird für jedes Trainingsgebiet und das zu klassifizierende Bildelement die Wahrscheinlichkeitsdichte berechnet. Das Pixel wird dann der Objektklasse mit der größten Wahrscheinlichkeitsdichte zugeordnet. Dazu muß zunächst die Kovarianzmatrix  $\underline{C}$  (4) für jedes Trainingsgebiet berechnet werden, wobei auf den Index  $i$  für die verschiedenen Objektklassen künftig verzichtet wird (*Kraus 1990*):

$$\underline{C} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1K} \\ \sigma_{21} & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & \\ \sigma_{K1} & \cdots & & \sigma_K^2 \end{bmatrix} \quad (4)$$

$\sigma_k^2$ : Standardabweichungen der Kanalmittelwerte

Auf der Hauptdiagonalen befinden sich die Quadrate der Standardabweichungen (Formel (3)). Außerhalb befinden sich die Kovarianzen (5), die sich beispielhaft für die Kanäle 1 und 2 folgendermaßen berechnen lassen:

$$\sigma_{12} = \frac{1}{N-1} \sum_{n=1}^N (g_{1n} - \bar{g}_1)(g_{2n} - \bar{g}_2) \quad (5)$$

Die Wahrscheinlichkeitsdichte läßt sich nun mit dem Vektor der Schwerpunkte  $\underline{\bar{g}}$  und der Kovarianzmatrix  $\underline{C}$  in Form einer  $K$ -dimensionalen Normalverteilung berechnen. Nun werden die Testgebiete nicht mehr durch einzelne Cluster, sondern durch die durch Formel (6) gegebene analytische Funktion beschrieben.

$$f(\underline{g}) = \left( \frac{1}{(2\pi)^{\frac{K}{2}} \sqrt{|\underline{C}|}} \right) \exp\left( -\frac{1}{2} (\underline{g} - \underline{\bar{g}})^T \underline{C}^{-1} (\underline{g} - \underline{\bar{g}}) \right) \quad (6)$$

$\underline{g}$  :  $K$ -dimensionaler Grauwertvektor eines beliebigen Bildelementes

$|\underline{C}|$  : Determinante der Kovarianzmatrix

Im Fall von zwei Kanälen ergeben sich Ellipsen mit dem Schwerpunkt  $(\bar{g}_1, \bar{g}_2)$  als Mittelpunktskoordinaten. Die Ausdehnung der Ellipsen läßt sich wieder mit dem Faktor  $m$  beeinflussen (Abb. 10). Wie man in Abb.10 sieht, passen sich die Ellipsen viel besser als der Box-Klassifikator an die Cluster an.

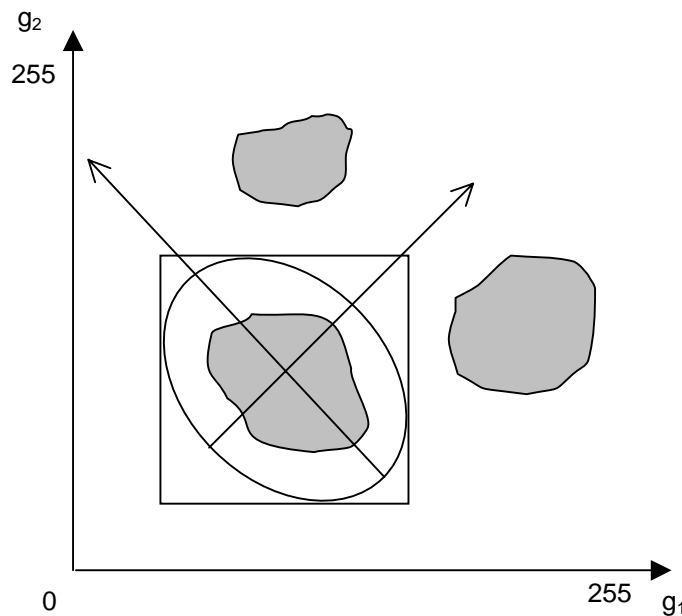


Abb. 10: Ellipse gleicher Wahrscheinlichkeitsdichte.

### 2.3.5 Klassifikation mit neuronalen Netzwerken

Dieser Klassifikator unterscheidet sich von den anderen dadurch, daß die Diskriminatoren nicht durch die Trainingsgebiete definiert werden, sondern iterativ durch Minimierung eines bestimmten Fehlerkriteriums bestimmt werden. In der englischen Literatur wird dieser Klassifikator Artificial Neural Network (ANN) Klassifizier genannt.

Ein neuronales Netz besteht aus einer Ansammlung von Primitivprozessoren (Knoten), die mit anderen Knoten des Netzes Informationen austauschen. Ein einfaches Netzwerk besteht aus drei Arten von Schichten: Das Input Layer beinhaltet in den einzelnen Knoten, die auch Units oder Nodes genannt werden, z.B. die Vektoren der Trainingspixel. Jeder Input-Knoten ist mit jedem Knoten der mittleren Verarbeitungsschicht, die auch Middle oder Hidden Layer genannt wird, mit einer bestimmten Gewichtung verknüpft. Die Units der mittleren Schicht erhalten die Werte des Input Layers und verarbeiten diese mit einem bestimmten Algorithmus und einer Aktivierungsfunktion. Die Knoten der Ausgabeschicht (Output Layer), die z.B. die gewünschten Klassen enthalten, sind wiederum mit den Knoten der mittleren Schicht untereinander verknüpft. Dieser Output Layer enthält das Ergebnis des Interpretationsprozesses (Abb. 11) (Schowengerdt 1997).

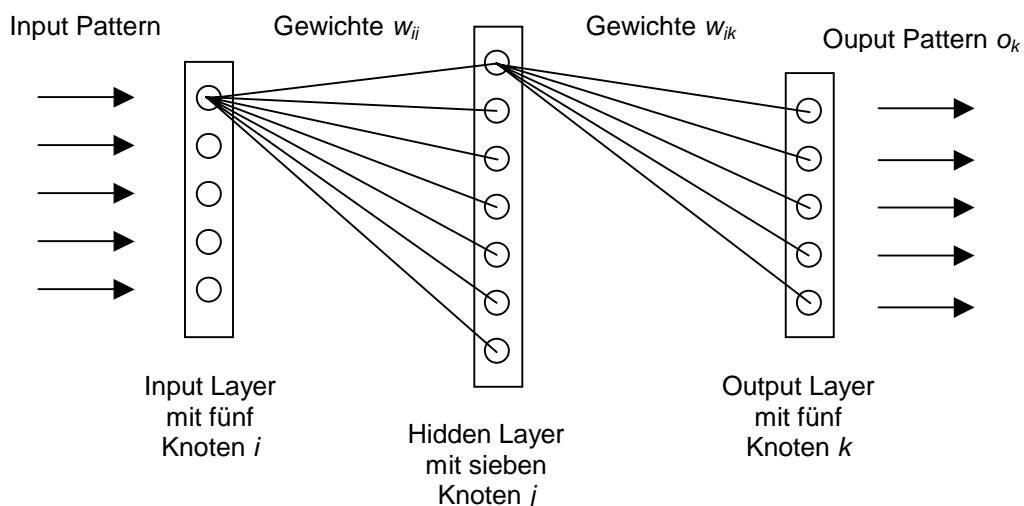


Abb. 11: Struktur eines neuronalen Netzwerkes mit drei Schichten. Zur Übersichtlichkeit sind nur die Verbindungen der ersten Knoten dargestellt. Tatsächlich sind die Knoten benachbarter Schichten alle untereinander verknüpft.

Die einzelnen Knoten können mehrmals während des Prozesses angesprochen werden, wobei sich die Ein- und Ausgabewerte ständig ändern, so daß ein Lernen stattfindet. Innerhalb eines Knotens der mittleren Schicht werden die Eingabewerte summiert und das Ergebnis transformiert:

$$S_j = \sum_i w_{ij} p_i$$

$$h_j = f(S_j) \quad (7)$$

Bezeichnungen entsprechen (Abb. 11)



Das Ergebnis  $h_j$  wird an jeden Knoten  $k$  der Ausgabeschicht weitergegeben und ausgewertet:

$$\begin{aligned} S_k &= \sum_j w_{jk} h_j \\ o_k &= f(S_k) \end{aligned} \quad (8)$$

Bezeichnungen entsprechen (Abb. 11)

Als Aktivierungsfunktion wird häufig die Sigmoid-Funktion (Abb. 12) benutzt:

$$f(S) = \frac{1}{1 + e^{-S}} \quad (9)$$

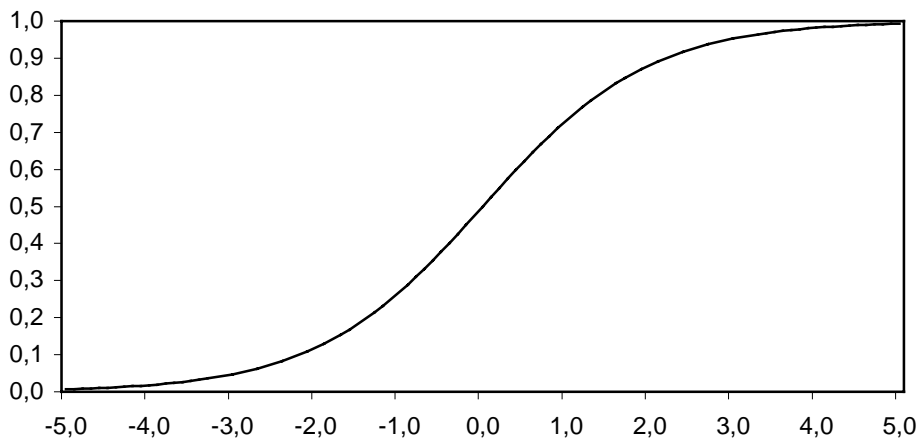


Abb. 12: Die Sigmoid-Funktion als Aktivierungsfunktion.

Der Back Propagation Algorithmus war der erste erfolgreiche Versuch eines ANN Klassifikators. Hierbei werden zuerst die Trainingspixel selektiert, und die Ergebnisvektoren  $d_k$  der jeweiligen Klasse  $k$  im output layer werden spezifiziert. Die Zielwerte, die der Algorithmus produzieren soll, sind z.B.  $d_m = 0,9$  für  $m = k$  und  $d_m = 0,1$  für  $m \neq k$ . Die Gewichte zwischen den einzelnen Knoten werden mit einer Zufallszahl zwischen 0 und 1 initialisiert, wobei typischerweise Zahlen nahe bei 0 gewählt werden. Für jedes Pixel des Trainingsgebietes wird nun ein Ausgabewert  $o$  ausgegeben, und der Fehler zum gewünschten Ergebnis  $d$  wird berechnet:

$$\frac{\|\mathcal{E}\|^2}{2} = \frac{1}{2} \sum_{p=1}^P \sum_k (d_k - o_k)^2 \quad (10)$$

$\|\mathcal{E}\|$  = Norm des Fehlers (der Faktor  $\frac{1}{2}$  dient nur der Vereinfachung)

P = Anzahl der Trainingspixel

Nachdem alle Pixel des Trainingsgebietes das Netz durchlaufen haben, werden die Gewichte neu berechnet (11). Das Netz wird nun so oft durchlaufen, bis der Fehler  $\epsilon$  einen bestimmten Schwellwert unterschreitet. Das System konvergiert so zu einer stabilen Lösung, wobei die Gewichte die Diskriminatoren für die Klassifikation festlegen.

$$\begin{aligned} \Delta w_{jk} &= LR \frac{\partial \mathcal{E}}{\partial w_{jk}} = LR \sum_{p=1}^P (d_k - o_k) \frac{d}{dS} f(S) \Big|_{S_k} h_j \\ \Delta w_{ij} &= LR \sum_{p=1}^P \left\{ \frac{d}{dS} f(S) \Big|_{S_j} \left[ (d_k - o_k) \frac{d}{dS} f(S) \Big|_{S_k} w_{jk} \right] p_i \right\} \end{aligned} \quad (11)$$

LR = Lernrate, um die Geschwindigkeit der Konvergenz zu kontrollieren

Der Algorithmus arbeitet allerdings sehr langsam, da viele Verknüpfungen vorliegen und sehr viele Iterationen durchlaufen werden müssen. Es kann auch passieren, daß keine globale Lösung mit minimalem Fehler gefunden wird, sondern der Algorithmus verfängt sich in einem lokalen Minimum. Außerdem erhält man bei zwei Durchläufen nicht das gleiche Ergebnis, da die Anfangswerte der Gewichte zufällig gewählt werden. Es gibt andere Algorithmen und Verfeinerungen, die hier aber nicht weiter beschrieben werden.

## 2.4 Zusätzliche und modifizierte Kanäle

Außer den Grauwerten eines Bildes kann man auch andere abgeleitete oder zusätzliche Informationen zur Klassifikation nutzen:

- *Modifizierte Grauwerte*: Die Grauwerte in den einzelnen Kanälen werden z.B. durch atmosphärische Einflüsse, Blickrichtung oder Geländeform proportional um Faktoren wie z.B. den Richtungskosinus abgeändert, die dann mathematisch korrigiert werden können. Die Standardabweichung in den einzelnen Klassen lassen sich dadurch oft um einen beachtlichen Betrag senken, so daß die Klassifikation stabiler und der Anteil der Fehlklassifikationen geringer wird (*Kraus 1988*).
- *Hauptkomponententransformation*: Die Verwendung vieler spektraler Kanäle ist nur dann sinnvoll, wenn sich die einzelnen Kanäle signifikant voneinander unterscheiden. Redundante Informationen erhöhen nur den Rechenaufwand. Mathematisch betrachtet treten bei identischen bzw. total abhängigen Kanälen singuläre, und bei sehr ähnlichen Kanälen schlecht konditionierte Kovarianzmatrizen auf. Bei einer Hauptkomponententransformation werden die Grauwerte in neue transformiert, so daß die Kovarianzen verschwinden (*Kraus 1990*):

$$\hat{C} = R^T C R \quad (12)$$

Die Drehmatrix  $R$  umfaßt die normierten Eigenvektoren der Matrix  $C$ . Die Kanäle mit den größten Hauptkomponenten enthalten die wesentlichen Informationen des Bildes, während die höheren Komponenten lediglich das Rauschen in den Kanälen wiedergeben.

- *Vegetationsindex*: Die Differenz aus der nahen infraroten (nIR) und der roten (R) Objektstrahlung ist bei der Klassifikation von Vegetationsarten sehr zweckmäßig. Diese Differenz wird Vegetationsindex genannt (*Kraus 1990*):

$$VI = \frac{nIR - R}{nIR + R} \quad (13)$$

Auf diese Weise läßt sich Datenmenge sinnvoll reduzieren, ohne daß relevante Informationen dabei verloren gehen.

- *Multitemporale Bilder*: Zu verschiedenen Zeitpunkten aufgenommene Bilder von Objekten liefern als zusätzliche Kanäle Informationen über Veränderungen in der Landschaft z.B. zur Überwachung von Vegetationsschäden.
- *Höhenmodell*: Stellt man die Höhen aus einem Oberflächenmodell als Grauwerte dar, so kann man diese Informationen in einem zusätzlichen Kanal nutzen. Dadurch ist es z.B. möglich, Klassen voneinander zu unterscheiden, die ähnliche Grauwerte aber verschiedene Höhen aufweisen, wie z.B. Straßen und Häuser mit grauen Dächern oder Bäume und Wiesen.
- *Textur*: In der Umgebung des zu klassifizierenden Bildelementes kann die Textur zur Unterstützung der Klassifikation genutzt werden. Die Texturinformationen können z.B. aus den lokalen statistischen Variationen der Helligkeit gewonnen werden. Der Range Filter liefert als Ergebnis die Differenz zwischen dem minimalen und dem maximalen Grauwert im Fenster des Filters. Dieser Wert kann noch mit einem Verstärkungsfaktor multipliziert werden, damit das Ergebnisbild für den Betrachter einen ansprechenden Kontrast und Helligkeit aufweist (*TNTmips 1998*).

### 2.5 Bildsegmentierung

Die bisherigen Methoden zur Klassifikation vernachlässigen die Tatsache, daß zwischen benachbarten Pixeln eine signifikante Korrelation besteht, da sie nur die spektralen Eigenschaften der einzelnen Pixel benutzen. Besonders in großmaßstäbigen Bildern sind jedoch nicht die Grauwerte der Bildelemente die ausschlaggebenden Informationsträger, sondern die charakteristischen Formen und Regelmäßigkeiten sowie die Bezüge von Objekten untereinander liefern die Entscheidung für eine Klassenzugehörigkeit. Die Bildsegmentierung<sup>2</sup> ist ein Verfahren, bei dem das zu klassifizierende Bild in zusammengehörende und homogene Bereiche unterteilt

---

<sup>2</sup> Die vorhergehend beschriebenen Methoden der multispektralen Klassifikation (2.3) können auch als eine Segmentierung betrachtet werden, jedoch versteht man unter einer Segmentierung im eigentlichen Sinn die in diesem Abschnitt beschriebenen Techniken.

wird, um einerseits die Korrelationen von Pixeln in der örtlichen Nachbarschaft zu berücksichtigen und um andererseits eine Datenreduktion durchzuführen, ohne daß wichtige Informationen verloren gehen.

Die Segmentierung kann auf zwei Arten erfolgen. Entweder werden durch Kantenextraktion markante Linien im Bild bestimmt, die anschließend in Grenzlinien von homogenen Flächen umorganisiert werden (edge-based), oder benachbarte Pixel werden durch Ähnlichkeitskriterien zu größeren Flächen zusammengeschlossen (region-based) (*Schowengerdt 1997*). Die Kantenextraktion kann im einfachsten Fall z.B. durch eine Filterung mit dem Laplaceoperator durchgeführt werden, wobei die zweite Ableitung eines Bildes vom Original subtrahiert wird. Das Ergebnis einer Kantenextraktion ist allerdings eine Ansammlung von Linienstücken, die im allgemeinen nicht eine sinnvolle Begrenzung eines Segmentes darstellen. Deshalb ist im Anschluß ein interaktiver Prozeß nötig, bei dem die erhaltenen Kanten in Vektoren umgewandelt werden und danach zu einem flächenbegrenzenden Liniennetz verknüpft werden.

Beim Gebietswachstum werden ähnliche Pixel zu einem Segment gruppiert, von denen angenommen wird, daß sie eine physikalische Fläche auf der Erdoberfläche repräsentieren. Als ein Beispiel wird folgend der Region Growing Algorithmus beschrieben. Dabei wird das Bild Pixel für Pixel von rechts nach links und von oben nach unten prozessiert, wobei die Nachbarschaft der Pixel untersucht wird. Ein Pixel kann bei einer 3x3 Nachbarschaft vier oder acht direkte Nachbarn besitzen. Je nach betrachteter Nachbarschaft können so unterschiedliche Flächen entstehen (Abb. 13) (*Schowengerdt 1997*). An jedem Pixel werden die Differenzen zu den jeweiligen Nachbarn gebildet und an Hand eines vorher festgelegten Schwellwertes wird entschieden, ob die Bildelemente verbunden werden oder nicht. Wird der Schwellwert unterschritten, so sind sich die Pixel ähnlich und werden zusammengeschlossen. Bei Überschreitung des Grenzwertes kann man davon ausgehen, daß die Pixel sich nicht sehr ähnlich sind und deshalb auch nicht zur selben physikalischen Fläche gehören und werden daher auch nicht miteinander verbunden. Der Algorithmus erfordert dabei eine Liste aller Pixel, die laufend aktualisiert wird, da sich die Zuordnung eines Pixels während eines Durchlaufes auch ändern kann. Bei entsprechend großen Bildern ist hier viel Arbeitsspeicher im

Computer nötig, um das ganze Bild zu prozessieren. Die erhaltenen Segmente bekommen als Farbe den Durchschnitt der in diesem Segment vorhandenen Pixel zugewiesen. Die eigentliche Klassifikation erfolgt danach an Hand dieser Flächen und nicht mehr pixelweise.

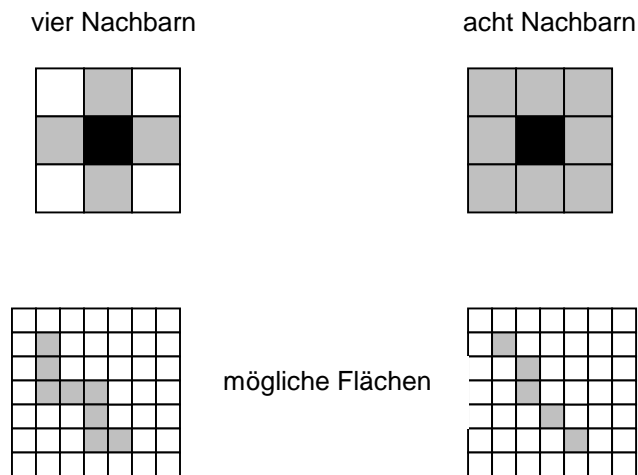


Abb. 13: Die zwei Möglichkeiten der Nachbarschaft und die daraus resultierenden Flächen.

## 3 Das Bildanalyzesystem eCognition

---

### 3.1 Grundkonzept

Klassische Satellitenbilder mit Auflösungen von z.B. 30 Metern (Landsat TM) eignen sich zur Klassifikation mit gewohnten Methoden (2.3). Die neuen Satellitensysteme erreichen allerdings höhere Auflösungen von bis zu 1 Meter (Ikonos), und auch flugzeuggetragene Scanner (DPA) werden immer häufiger eingesetzt. Diese Systeme haben mit ihrer hohen Auflösung das Potential, in vielen Bereichen Anwendung zu finden, welche bisher aufgrund zu hoher Ungenauigkeit nicht mit Fernerkundungsdaten arbeiten, wie z.B. Versicherungen, Touristikunternehmen, Grundstücksmakler, Polizei oder Medien. Je größer der Maßstab des auszuwertenden Bildes allerdings wird, desto mehr Bedeutung verlieren die spektralen Werte der einzelnen Pixel. Die in einem solchen Bild enthaltenen Informationen ergeben sich größtenteils nicht aus einzelnen Pixeln sondern aus Objekten und ihren gegenseitigen Beziehungen, die von den bisherigen Algorithmen nicht berücksichtigt werden.

Die Firma *Delphi2* hat mit dem Programm *eCognition* eine Software entwickelt, die mit Hilfe von Erkenntnissen über die menschliche Wahrnehmung einen neuen Weg der Bildanalyse beschreitet. Das Bildverständnis des Menschen geht über das Sehen weit hinaus, denn es ist ein hochaktiver kognitiver Prozeß mit zahlreichen Schritten zwischen den visuellen Eingangsdaten und der menschlichen Mustererkennung und den Erinnerungen. Das Gehirn geht dabei objektorientiert vor, d.h. bei der Betrachtung eines Bildes konzentriert es sich immer auf bestimmte Details (Objektprimitive), um sich danach ein Gesamtbild zu konstruieren. Bei der Wahrnehmung von Objekten vergleicht das Gehirn die aktuelle Ansicht aus dem vorliegenden Bild mit Erfahrungswerten, die von Erlerntem stammen können oder durch im Laufe der Evolution erworbene Anpassungen genetisch weitervererbt werden. Wenn im Bild ein Auto von der Vorderseite zu sehen ist, dann stellt sich der Mensch aufgrund seiner Erfahrung auch die Rückseite bzw. das ganze Objekt vor. Diese Vorstellung wird mit allen im Gehirn gespeicherten Eigenschaften des Objektes verknüpft, z.B. das Auto

ist aus Metall, und man kann sich damit fortbewegen. Als nächstes werden die umliegenden Objekte ebenso erkannt und untereinander in Verbindung gebracht, wodurch ein Gesamtbild entsteht. Stehen beispielsweise viele Autos ohne Fahrer nebeneinander, so schließen wir daraus, daß es sich um einen Parkplatz handeln muß. Dieser Vorgang erscheint trivial und läuft innerhalb von Sekundenbruchteilen ab. Es wird allerdings deutlich, daß die menschliche Wahrnehmung objektorientiert und nicht pixelorientiert abläuft, da die spektralen Werte allein nicht genügend Informationen zur Identifikation liefern. Ein farbiger Punkt läßt noch nicht auf das dazugehörige Objekt schließen.

Die Software *eCognition* ist natürlich bei weitem nicht so komplex wie die menschliche Wahrnehmung, jedoch wird auch hier objektorientiert vorgegangen. Es wird ein semantisches Netz aus einzelnen Objekten gebildet, die alle konkrete Attribute besitzen und mit ihren Nachbarn verknüpft sind. Verschiedene Auflösungen können wie bei der menschlichen Wahrnehmung gebildet werden, so daß ein hierarchisches Netzwerk entsteht, in dem Unterobjekte Teile von größeren Strukturen sind (Abb. 14). Diese Netzwerke sind weitestgehend selbstkonstruierend und dynamisch. Sie basieren auf dem Fractal Net Evolution-Konzept, welches vom wissenschaftlichen Leiter Prof. Dr. Gerd Binnig bei *Delphi2* entwickelt wurde (Baatz/Schäpe 1999).

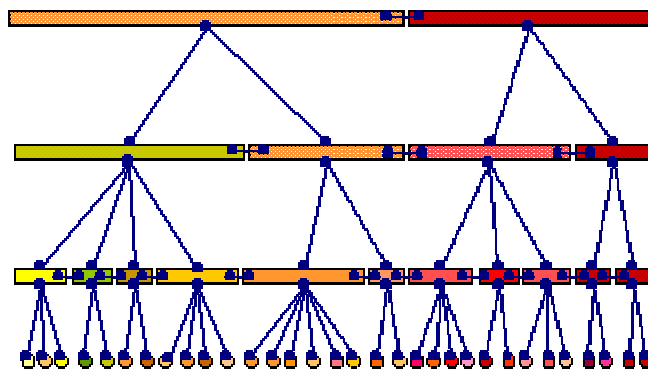


Abb. 14: Hierarchisches Netzwerk von Bildobjekten, die alle ihren Kontext, Nachbarschaft und Unterobjekte kennen. Die unterste Ebene wird von den einzelnen Pixeln des Bildes gebildet, und die oberste Ebene repräsentiert das ganze Bild als solches. Dazwischen befinden sich Ebenen mit Objekten, von oben nach unten betrachtet, höherer Auflösung.

Indem sinnvolle Beziehungen zwischen den vernetzten Objekten definiert werden, können zur Klassifikation neben spektralen, Form- und Texturinformationen auch Kontextinformationen verwendet werden. Das Programm arbeitet dabei in zwei



Phasen. Als erstes müssen Objektprimitive aus dem Bild extrahiert werden, und ein hierarchisches Netzwerk wird aufgebaut. Die bei *eCognition* Multiresolution Segmentation genannte Technik zur Bildsegmentierung (2.5) extrahiert Objekte aus Bildern beliebiger Auflösung. Sie eignet sich allerdings nur für Gebiete mit homogener Struktur wie sie z.B. in ländlichen Gegenden vorkommt. Städtische Gebiete weisen vielmehr eine heterogene Struktur auf, welche nicht unter der Annahme der Homogenität extrahiert werden können. *eCognition* stellt für solche Fälle eine wissensbasierte Segmentierung zur Verfügung. Dabei können auf der Basis von klassifizierten Objektprimitiven umfassendere Objekte gebildet werden. In der zweiten Phase werden die aus der Segmentierung erhaltenen Bildobjekte an Hand deren Eigenschaften und gegenseitigen Beziehungen klassifiziert. Das Ergebnis kann dann als ein Bitmapbild exportiert werden.

## 3.2 Aufbau des Programms

Das Programm benötigt zum Betrieb Windows NT 4.0 und orientiert sich bei der Bedienung an Windows Standards. Das Hauptfenster besteht aus einer Menüleiste, einer Symbolleiste, einer Statuszeile und dem Projektfenster (Abb. 15). Um das Aussehen des Programms seinen Bedürfnissen anzupassen, lassen sich die Statuszeile und Teile der Symbolleiste auch ausblenden, wobei die Einstellungen allerdings nicht gespeichert werden und beim nächsten Programmstart nicht mehr zur Verfügung stehen. Dies ist allerdings kein Problem, da dies die einzigen einstellbaren Optionen sind. Wenn mit der Maus auf einen Knopf in der Symbolleiste gezeigt wird, werden etwas zeitverzögert Kurzhinweise zur Bedeutung der Knöpfe, sogenannte Quickinfos, eingeblendet, die auch in einigen Dialogboxen zur Verfügung stehen.

Zu Beginn einer Klassifikation werden die zu bearbeitenden Bilder über PROJECT/Import Bitmaps... oder über den entsprechenden Schaltknopf der Symbolleiste geladen. Es können in der vorliegenden Beta-Version bis zu 32 Bilder bzw. Kanäle geladen werden. Die Bilder müssen allerdings im 8 Bit Bitmap-Format vorliegen, wobei ein Bild einem Kanal entspricht. Bei einem kompletten Landsat Bild arbeitet man also mit sieben Bitmaps. In der Release-Version sollen dann beliebig

viele Kanäle in jeder Bit-Auflösung verwendet werden können, und der Import üblicher Bildformate wird möglich sein.

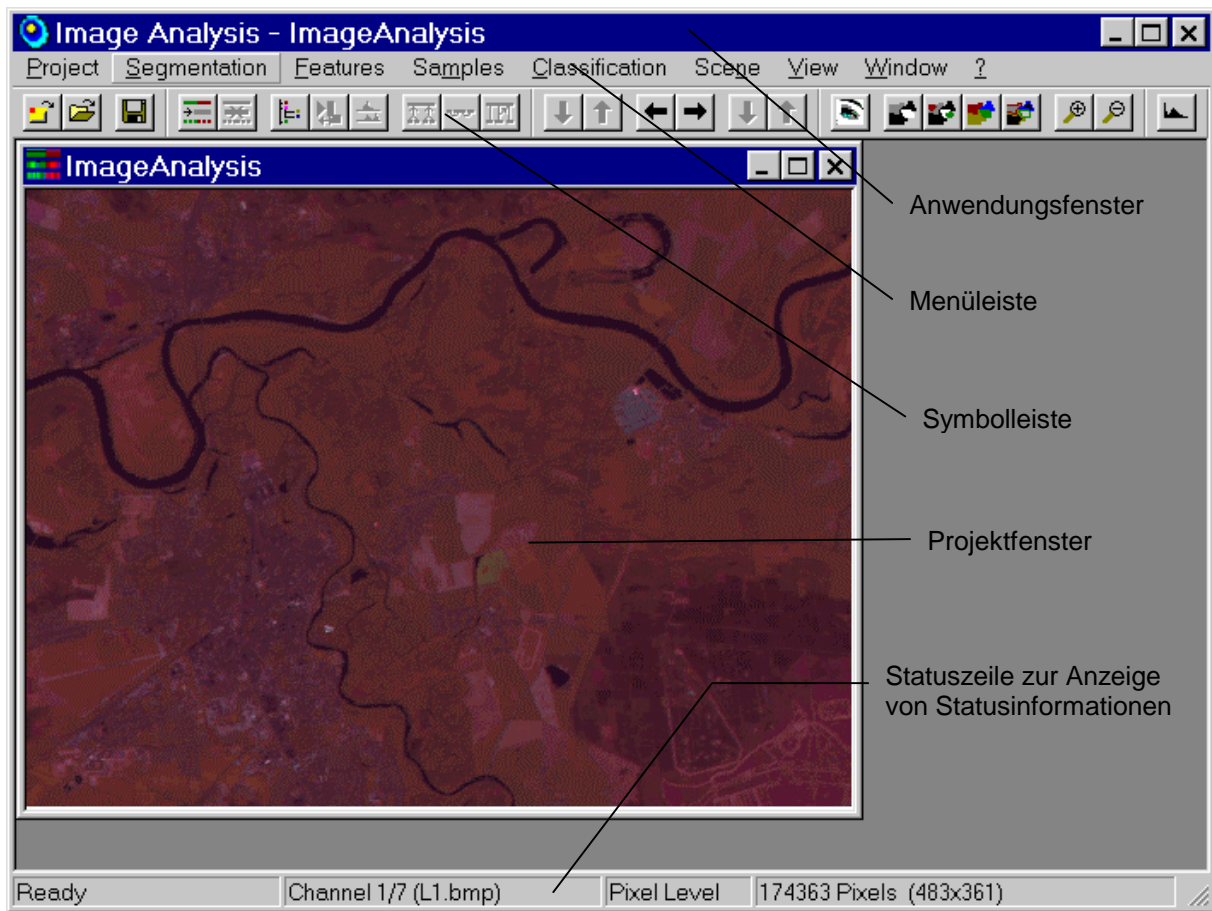


Abb. 15: Die Oberfläche von eCognition.

In der Statuszeile wird die Reihenfolge der Kanäle bzw. der gerade zu sehende Kanal angezeigt, und wie viele Kanäle insgesamt vorhanden sind. Beim Laden der Bilder wird zugleich auch die unterste Ebene des hierarchischen Netzes angelegt, welche Pixel Level genannt wird. Auch über die Bildgröße und die daraus resultierende Anzahl der Pixel gibt die Statuszeile Auskunft, wobei anzumerken ist, daß die Bilder eines Projektes exakt die gleiche Größe besitzen müssen. Als nächstes sollten die Einstellungen der Ansicht unter VIEW/View Settings... eingestellt werden (Abb. 16). Mit dem Listenfeld MODE wählt man als Ansicht unter dem Originalbild (Channel Mean), der Klassifikation (Classification) und den Beispielsegmenten (Samples) für eine Nearest Neighbour Klassifikation (2.3.2). Zur Kontrastverstärkung steht ein Gauß Filter zur Verfügung, der die Segmentierung und die Klassifikation allerdings nicht beeinflusst.

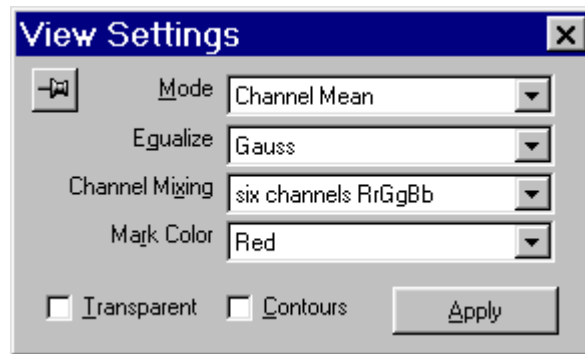


Abb. 16: Einstellungen der Ansicht.

Unter CHANNEL MIXING wählt man die Mischung der Kanäle aus. Es kann ein einzelner Kanal mit seinen Grauwerten (one channel grey), eine Mischung aus drei Bildern (three channels RGB) oder eine Mischung aus sechs Bildern (six channels RrGgBb) angezeigt werden. Es ist auch möglich, sich eine eigene Auswahl (user defined) zusammenzustellen (Abb. 17).

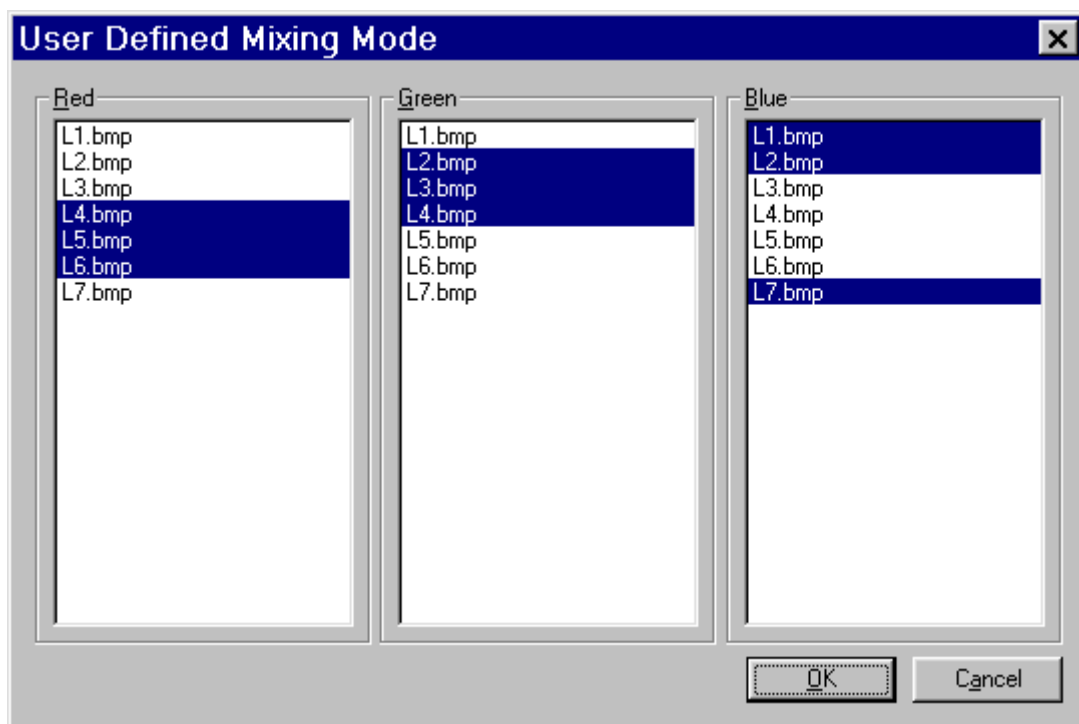



Abb. 17: Benutzerdefinierte Kanalmischung.

Als Farbe für ein markiertes Segment (MARK COLOUR) kann Rot, Weiß oder Schwarz gewählt werden. Mit dem Kontrollkästchen TRANSPARENT können die unmarkierten Segmente transparent dargestellt, also ausgeblendet werden, und mit CONTOURS wird nur der Umriß der Segmente gezeigt. Mit  kann das Dialogfeld fixiert werden,

so daß es nicht geschlossen wird, wenn auf [Apply] geklickt wird. Unter VIEW/Zoom In bzw. unter VIEW/Zoom Out können die Bilder beliebig vergrößert oder verkleinert werden, wobei die Größe der Bilder jedesmal verdoppelt bzw. halbiert wird. Auch diese Funktion ist in der Symbolleiste zu finden.

### 3.3 Multiresolution Segmentation

Nach dem Laden der Bilder und dem Einstellen der Ansicht kann mit der Multiresolution Segmentation begonnen werden, die unter SEGMENTATION/Multiresolution Segmentation... zu finden ist (Abb. 18).

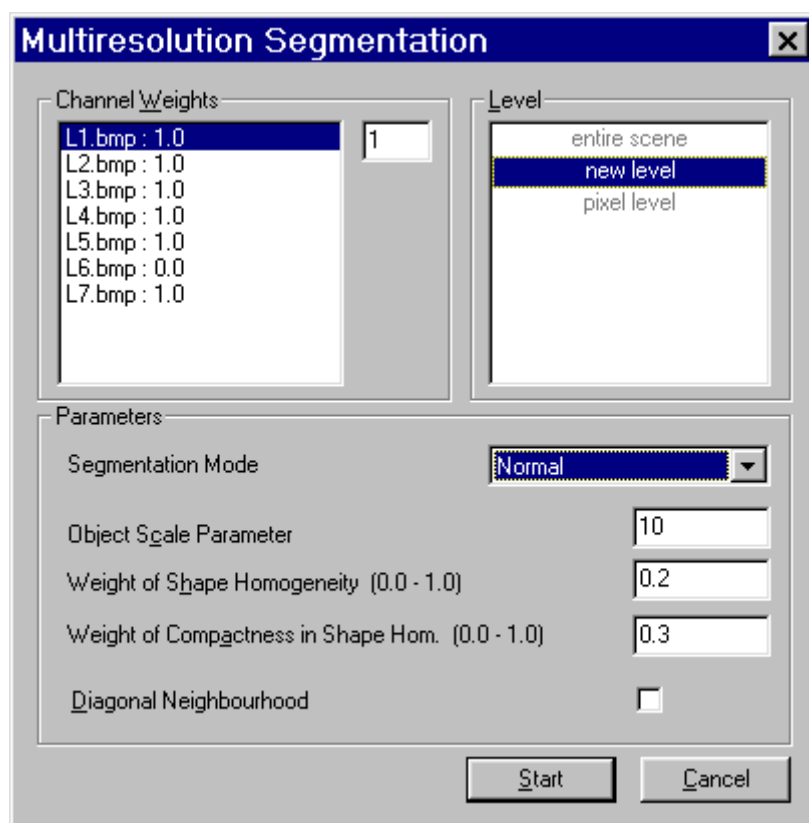


Abb. 18: Dialogfeld zur Multiresolution Segmentation.

Unter CHANNEL WEIGHTS sind alle verfügbaren Kanäle mit ihrer Gewichtung zu sehen. Der Standardwert von 1.0 kann verändert werden, um wie in diesem Beispiel den Infrarotkanal von Landsat TM, der nur eine Auflösung von ca. 120 Metern

aufweist, aus der Segmentierung auszuschließen, da nur die hochauflösenden Kanäle zur Segmentierung empfehlenswert sind. Dessen Informationen stehen danach trotzdem für die Klassifikation zur Verfügung. In dem Bereich LEVEL wird festgelegt, an welcher Stelle im hierarchischen Netzwerk die neue Ebene eingefügt wird. Dies ist zu Beginn trivial, da noch keine Objekthierarchie vorliegt, so daß sie zwischen dem Pixel Level und der gesamten Szene eingefügt wird. Es kann später allerdings auch eine bereits existierende Ebene ausgewählt werden, die nachfolgend dann durch die neue Segmentierung ersetzt wird. Der wichtigste einzustellende Parameter ist OBJECT SCALE PARAMETER, mit dem die mittlere Größe der resultierenden Objekte festgelegt wird. Er definiert eine obere Schranke, wieviel maximale Heterogenität pro Objekt zugelassen wird.<sup>3</sup> Hierbei ist zu beachten, daß die mittlere Größe über alle Kanäle aufsummiert wird, d.h. ein OBJECT SCALE PARAMETER von 50 bei der Verwendung von fünf Kanälen entspricht in etwa 10 bei der Verwendung nur eines Kanals. Liegen schon mehrere Ebenen vor, so muß darauf geachtet werden, an welcher Stelle die neue eingefügt wird. Legt man eine neue Ebene über einer anderen an, so muß die Objektgröße erhöht werden. Wenn die Objektgröße kleiner oder genauso groß wie bei der existierenden Ebene ist, so erhält man Objekte gleicher Größe wie zuvor. Ebenso muß bei der Erzeugung einer Unterebene der Object Scale Parameter geringer gewählt werden. Dies liegt darin begründet, daß die Objektgrenzen immer den Grenzen der nächsten Unterobjekte folgen müssen, und die Objekte einer Segmentierung durch die Objektgrenzen der nächsthöheren Ebene beschränkt sind. Dies kann bei vielen Ebenen leicht unübersichtlich werden und zu unerwünschten Ergebnissen führen, da die vormals verwendeten Größen nicht angezeigt werden (Abb. 19). Die Firma *Delphi2* beabsichtigt hier allerdings, bis zur Release-Version nachzubessern und dieses Feld etwas übersichtlicher zu gestalten.

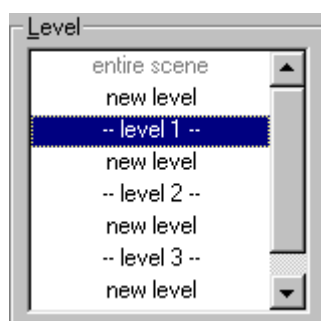



Abb. 19: Ebenenhierarchie mit drei vorhandenen Segmentierungsebenen.



<sup>3</sup> Eine genauere Definition der Heterogenität findet sich im Abschnitt (4.2.1).

Mit  kann eine nicht gewünschte Ebene aus der Hierarchie wieder entfernt werden. Der SEGMENTATION MODE sollte im allgemeinen auf NORMAL eingestellt sein. Die andere zur Verfügung stehende Option ist FORM SUBOBJECTS, welche der Erzeugung von Unterobjekten (Subobjekten) zur Form- und Texturanalyse dient (4.3.5). Der Wertebereich des OBJECT SCALE PARAMETERS liegt hierbei zwischen 0 und 1. Dies definiert die maximale relative Grenzlänge zu Nachbarobjekten, die Teil eines höher liegenden Objektes (Superobjekt) sind (*Baatz/Schäpe 1999*). Die nächsten beiden Parameter dienen zur Feineinstellung der Segmentierung und beeinflussen die Gestalt der resultierenden Objekte. Mit WEIGHT OF SHAPE HOMOGENEITY kann der Segmentierungsalgorithmus gesteuert werden, ob bei der Erzeugung der Objekte mehr Farbe oder mehr Form berücksichtigt werden soll. Die Formel hierzu lautet (*Baatz/Schäpe 1999*):


$$(1-\text{weight}) * \text{Color Homogeneity} + \text{weight} * \text{Shape Homogeneity} \quad (14)$$

Die Formhomogenität kann aus zwei Kriterien zusammengesetzt werden. Eines optimiert eine möglichst kompakte Form (Area Criterion), wobei die kompakteste Fläche einem Kreis entspricht. Das andere (Border Criterion) läßt durchaus längliche Objekte zu, aber optimiert eine möglichst geringe Randrauhigkeit. Mit WEIGHT OF COMPACTNESS IN SHAPE HOM. kann die Mischung aus diesen beiden Kriterien verändert werden (*Baatz/Schäpe 1999*).

$$(1-\text{weight}) * \text{Border Criterion} + \text{weight} * \text{Area Criterion} \quad (15)$$

Mit dem Kontrollkästchen DIAGONAL NEIGHBOURHOOD wählt man schließlich eine vierfache und eine diagonale achtfache Nachbarschaft (2.5) aus. Sind alle Parameter eingestellt, so kann durch Klicken auf [Start] die Segmentierung beginnen. Mit   kann man zwischen den einzelnen Ebenen umschalten, wobei die gerade aktive Ebene in der Statuszeile angezeigt wird.

### 3.4 Klassifikation

Hat man eine oder mehrere Segmentierungsebenen angelegt, so kann man jetzt mit der Klassifikation fortfahren, die grundsätzlich überwacht (2.3.1) erfolgt. Dazu öffnet man unter CLASSIFICATION/Edit Class Hierarchy... oder mit  die entsprechende Dialogbox, die zu Beginn noch keine Einträge enthält. Klickt man mit der rechten Maustaste in das große weiße Feld, so erscheint eine Schaltfläche, um eine neue Klasse einzufügen (Abb. 20).

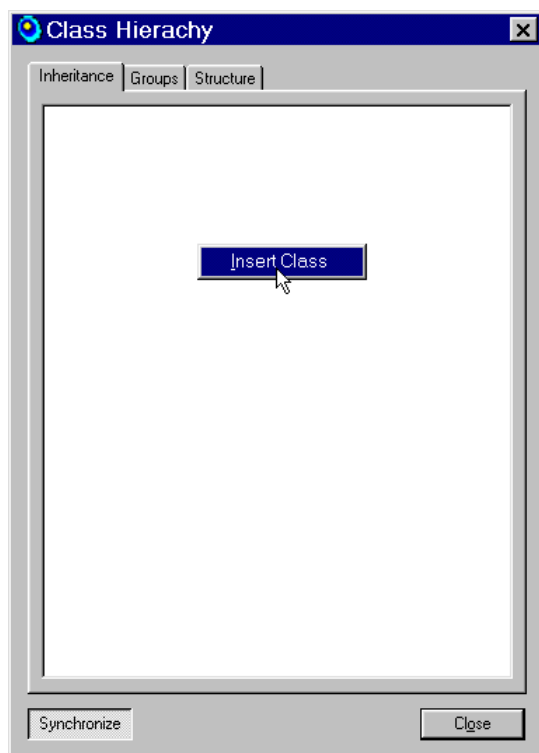


Abb. 20: Einfügen einer neuen Klasse.

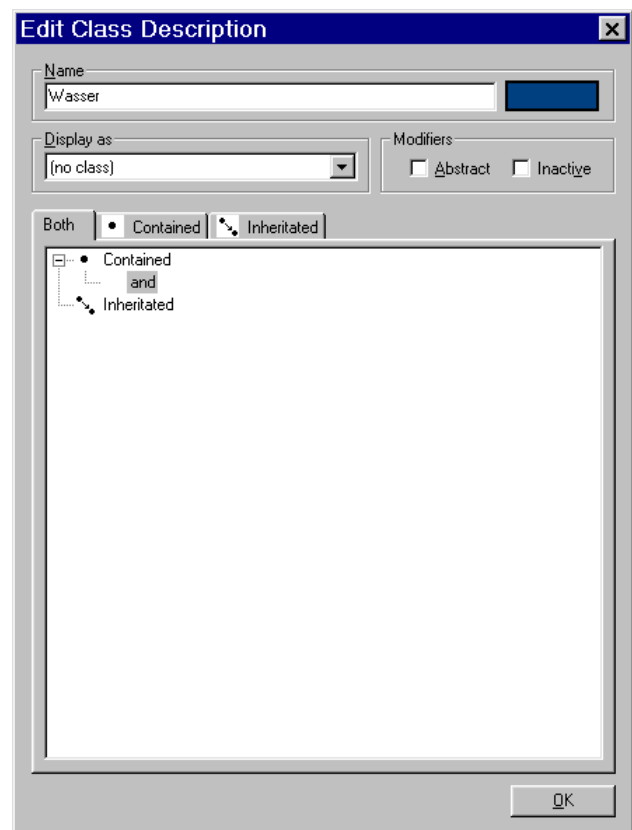


Abb. 21: Eigenschaften einer Klasse.

Daraufhin öffnet sich das Dialogfenster EDIT CLASS DESCRIPTION (Abb. 21), in dem die Eigenschaften der jeweiligen Klasse angezeigt werden. Unter NAME vergibt man eine Bezeichnung, und mit dem farbigen Kästchen daneben wählt man eine Farbe für die Klasse aus. Wie man weiterhin sieht, besteht die Klassenhierarchie aus drei verschiedenen Kategorien: INHERITANCE, GROUPS und STRUCTURE. Die ersten beiden werden zur Klassifikation genutzt, während die STRUCTURE Abteilung Strukturgruppen von Klassen zur wissensbasierten Klassifikation definiert. Ist die

[Synchronize] Schaltfläche aktiviert, so werden Veränderungen in einer der ersten beiden Klassen in die jeweilige andere übernommen.

- INHERITANCE: Beschreibungen von allgemeineren Klassen können auf speziellere Unterklassen vererbt werden. Eine Klasse kann auch die Eigenschaften von mehreren Oberklassen erben. Die Oberklasse Frucht kann z.B. aus den Unterklassen Äpfel, Bananen und Kirschen bestehen. Änderungen in der Klasse Frucht gehen auf die Unterklassen über. Verboten sind dagegen Kreisvererbungen, d.h. Klasse A vererbt auf Klasse B, B auf C und C auf A (*Baatz/Schäpe 1999*).
- GROUPS: Die Gruppen bestehen aus einer Ansammlung von Klassen, die logisch zusammengehören. Die Klassen Nadelwald und Laubwald können hier z.B. zu einer Oberklasse Wald zusammengefaßt werden. Formuliert man die Bedingung, daß Städte nicht in Wäldern zu finden sind, so muß die relative Kantenlänge von einem Stadtobjekt zu Waldobjekten weniger als 100% betragen. Ist ein Objekt zu 50% von Nadelwald und zu 50% von Laubwald umgeben, so könnte das Objekt fälschlicherweise als Stadt klassifiziert werden. Auch hier sind Kreisbeziehungen wie Klasse A gehört zu B, B zu C und C zu A nicht zugelassen (*Baatz/Schäpe 1999*).

Die Definition von Untergruppen geschieht dabei dadurch, daß die betreffende Gruppe einfach mit der linken Maustaste auf die Oberklasse gezogen wird. Hält man die rechte Maustaste gedrückt, so erscheint die Klasse zweimal und kann so Bestandteil mehrerer Oberklassen werden. Jetzt wird abgefragt, ob die Oberklasse in eine abstrakte Klasse umgewandelt werden soll, d.h., sie wird nicht zur Klassifikation von Objekten herangezogen. Dies macht Sinn, denn die Unterklassen differenzieren die von der Oberklasse geerbten Merkmalsbeschreibungen mit eigenen Merkmalen aus. Diese Option ist auch in dem Dialogfeld EDIT CLASS DESCRIPTION unter MODIFIERS vorhanden. Um den Vorgang rückgängig zu machen, braucht die Klasse nur aus ihrer Oberklasse gezogen werden. Auf diese Weise lassen sich komplexe semantische Beziehungen aufstellen, wobei gleichzeitig die Übersichtlichkeit gesteigert wird (Abb. 22).



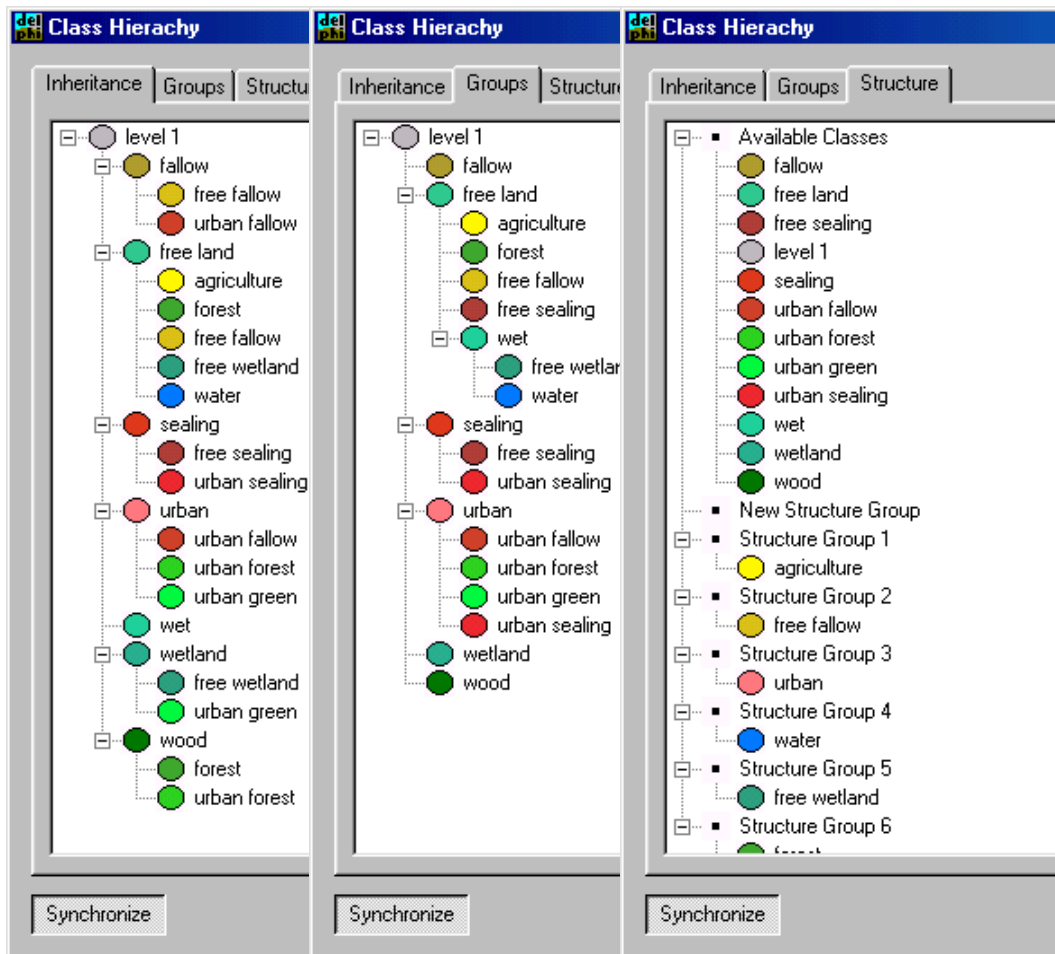


Abb. 22: Vererbung und Gruppierungen von Klassen (Batz/Schäpe 1999).

Um einer Klasse eine Beschreibung hinzuzufügen, klickt man mit der rechten Maustaste auf „and“ in der EDIT CLASS DESCRIPTION Dialogbox und anschließend auf INSERT NEW EXPRESSION, worauf sich die INSERT EXPRESSION Dialogbox öffnet (Abb. 23). Hier wählt man unter vielen verschiedenen Merkmalen ein geeignetes aus, wobei zu Beginn die Auswahl auf die Object Features beschränkt ist, da die Knowledge based Features schon klassifizierte Objekte voraussetzen. Durch einen Doppelklick auf das gewünschte Merkmal öffnet sich das Dialogfeld EDIT MEMBERSHIP FUNCTION. Welche Merkmale in INSERT EXPRESSION zur Verfügung stehen, kann unter FEATURES/Select active Features... eingestellt werden. Nachdem man die Zugehörigkeitsfunktion (3.5.1) für dieses Merkmal eingestellt hat, kann eine erste Klassifikation erfolgen. Durch Klicken auf CLASSIFIATION/Classify !... öffnet sich der Dialog zur Klassifikation (Abb. 24). Unter LEVEL IN THE OBJECT HIERARCHY wählt man eine oder alle Segmentierungsebenen aus und beginnt die Klassifikation mit [Start]. Die Optionen zur Kontextklassifikation brauchen jetzt noch nicht aktiviert zu werden, weil noch keine wissensbasierten Merkmale verfügbar sind.

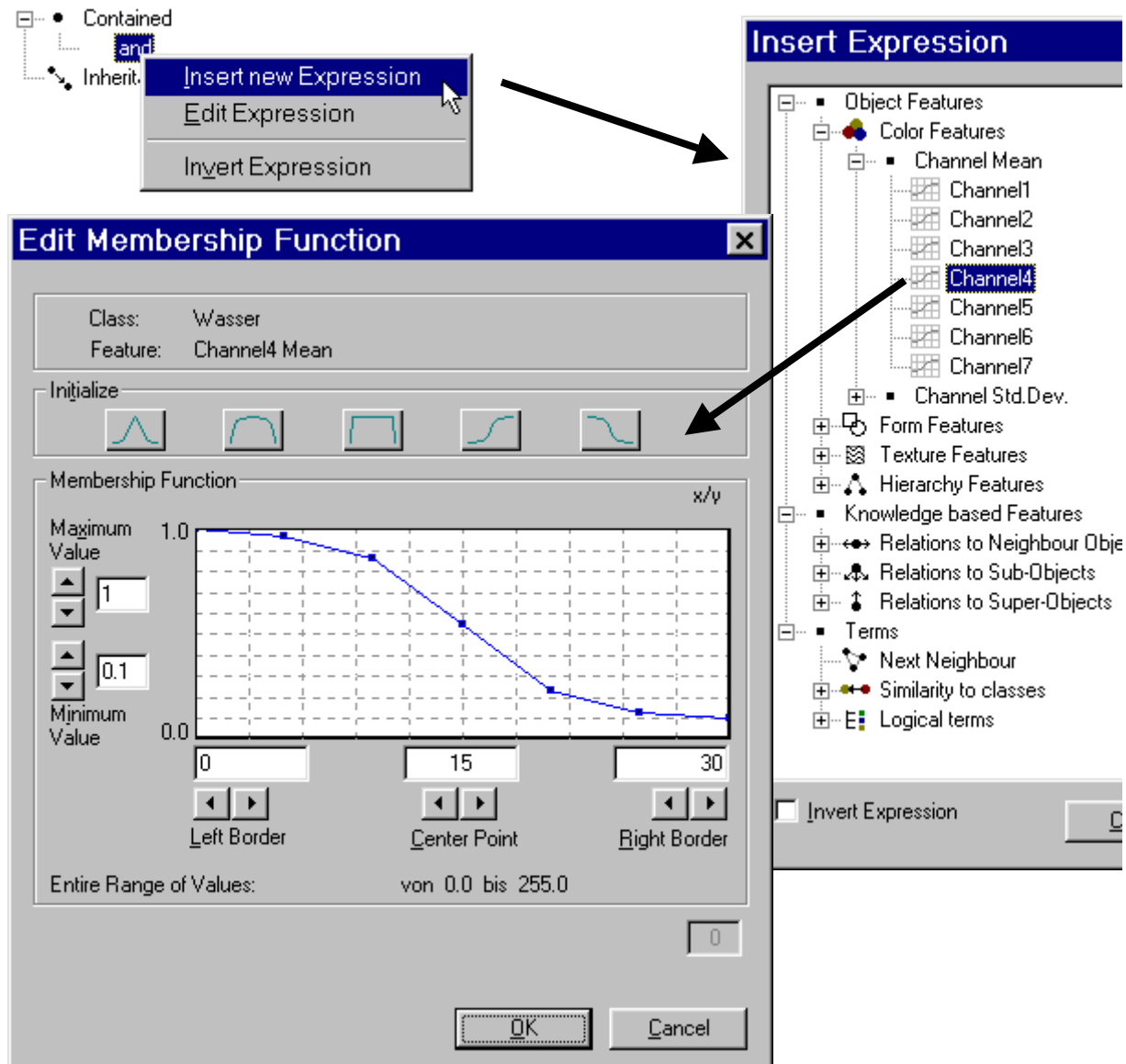


Abb. 23: Einfügen eines Merkmals für eine Klasse mit zugehöriger Membership Function.

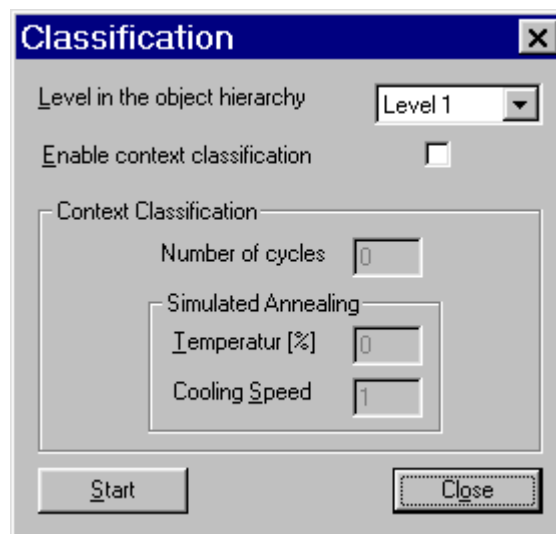


Abb. 24: Starten der Klassifikation

Ist man mit dem Ergebnis nicht zufrieden, so kann die Zugehörigkeitsfunktion angepaßt werden, oder es werden neue Merkmale definiert. Schließlich werden neue Klassen definiert, so daß man sich seinem Ziel immer weiter annähert. Für eine Klassifikation nach Objektmerkmalen (Mittelwerte, Standardabweichungen, Flächen, Grauwertdifferenzen zur Umgebung...) bietet sich außerdem die Methode Nearest Neighbour an (3.5.2 und 2.3.2).

Nach einer ersten Klassifikation können nun auch die Knowledge based Features genutzt werden, die den Kontext eines Objektes, d.h. seine Beziehungen zu Nachbarn, Super- und Subobjekten berücksichtigen. Der Bereich CONTEXT CLASSIFICATION wird durch Aktivierung des ENABLE CONTEXT CLASSIFICATION Kontrollkästchens zugänglich. Die NUMBER OF CYCLES gibt die Anzahl der Klassifikationsdurchläufe an. Ein mehrfaches Klassifizieren von allen Objekten in Zyklen ist erforderlich, wenn ein Objekt seine Klassenzugehörigkeit aufgrund der Klassifikation vernetzter Objekte ändert, denn es ist dann selber möglicherweise wiederum Kontext für andere Objekte. Hierbei können Instabilitäten entstehen, die durch einen „simulierten Auskühlungsprozeß“ (Simulated Annealing) aufgefangen werden können. Die berechnete Klassenzugehörigkeit eines Objektes wird hierbei zufällig verändert. Die Temperatur gibt dabei die Größe der möglichen zufälligen Änderungen der Klassenzugehörigkeit in Prozent an. Ein Zugehörigkeitswert von 0.5 (= 50%) kann bei einer Temperatur von 75% in einen Bereich von [0.125–0.875] transformiert werden. Nach jedem Zyklus wird die Temperatur mit COOLING SPEED neu berechnet, so daß nach dem Abkühlungsprozeß die Klassifikation einen stabilen Zustand erreicht (Baatz/Schäpe 1999).

$$T * \exp(-cycle/speed) \quad (16)$$

Die Bezeichnungen entsprechen der Classification Dialogbox (Abb. 24).

Unter VIEW/View Feature... öffnet sich eine Dialogbox, in der ein bestimmtes Merkmal zur Ansicht ausgewählt werden kann. Die Werte dieses Merkmals in den jeweiligen Objekten werden als Grauwerte dargestellt. Haben alle Objekte ungefähr den gleichen Wert, so kann man dies hier schnell feststellen. Dieses Merkmal ist dann nicht sehr hilfreich zur Unterscheidung der Klassen, und man kann nach

anderen Merkmalen zur Klassifikation suchen. Geeignete Objektmerkmale lassen sich auch mit dem IMAGE OBJECT INFORMATION Fenster auswählen, welches erscheint, wenn man ein Objekt mit der rechten Maustaste anklickt (Abb. 25). Hier werden im linken Bereich alle aktiven Merkmale und ihre Werte des Objektes angezeigt. Hat man das Bild schon klassifiziert, so wird für das selektierte Objekt auf der rechten Seite oben die aktuelle Klassenzugehörigkeit mit der entsprechenden Wahrscheinlichkeit angezeigt. In dem Fenster darunter sind alle möglichen Klassenzuordnungen mit ihren Wahrscheinlichkeiten aufgelistet. Unter SAMPLE-ASSIGNMENT wird die Klasse angezeigt, zu der das Objekt ein Beispiel für die Nearest Neighbour Klassifikation ist.

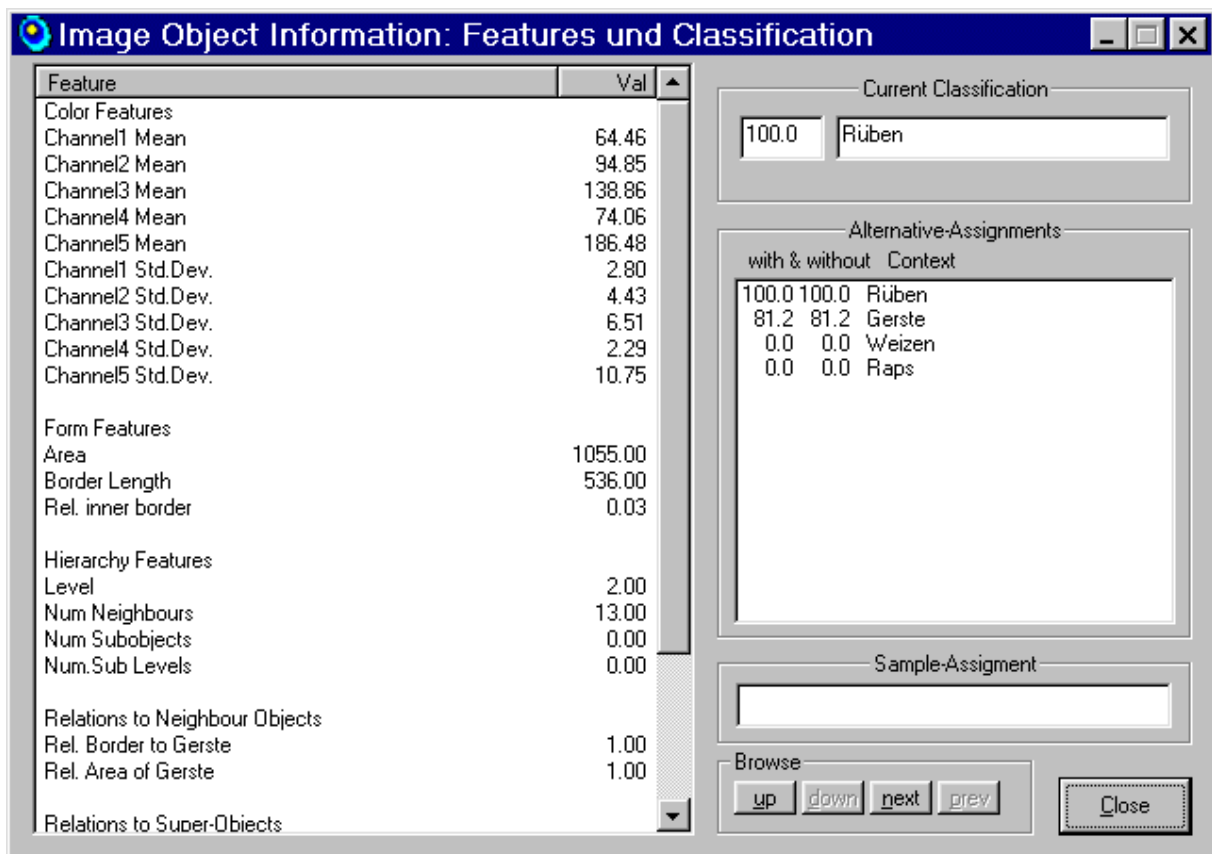


Abb. 25: Anzeige von Informationen über ein Objekt.

Im BROWSE Bereich navigiert man durch die verschiedenen Objekthierarchien von Ober- zu Unterobjekt oder von Nachbar zu Nachbar. Klickt man auf eine Klasse im Bereich ALTERNATIVE-ASSIGNMENTS, so schaltet das Fenster vom GENERAL ATTRIBUTES in den EVALUATION OF SINGLE CLASSES Anzeigemodus um (Abb. 26). Jetzt wird im linken Bereich eine Liste aller Merkmale für diese Klasse dargestellt. Klickt man in den linken Bereich, so gelangt man wieder in den alten Anzeigemodus.

Feature	Sngl	Cntx
Evaluation of Class Gerste	81.2	81.2
-----		
and	81.2	81.2
Channel5 Mean : 186.48	81.2	81.2
Channel3 Mean : 138.86	100.0	100.0
Channel2 Mean : 94.85	100.0	100.0

Abb. 26: EVALUATION OF SINGLE CLASSES Anzeigemodus.

## 3.5 Klassifikatoren

### 3.5.1 Zugehörigkeitsfunktionen

Die Zugehörigkeitsfunktionen lassen Formulierungen zu wie z.B. „Gewässer ist alles, was im Kanal 4 einen Grauwert von weniger als 30 aufweist“ und sind einer der beiden in *eCognition* zur Verfügung stehenden Klassifikatoren. Sie ähneln in ihrer Arbeitsweise dem Box-Klassifikator (2.3.3), sie sind allerdings um einige Funktionen erweitert worden. Zunächst basiert die Klassenzuteilung auf Fuzzy Logic, d.h. die strikte Zuweisung von Ja oder Nein zu einer Klasse wird durch eine kontinuierliche Funktion mit dem Wertebereich [0...1] ersetzt, wobei 0 ein exaktes Nein und 1 ein exaktes Ja bedeuten. Jeder andere Wert drückt eine Wahrscheinlichkeit der Klassenzugehörigkeit aus.

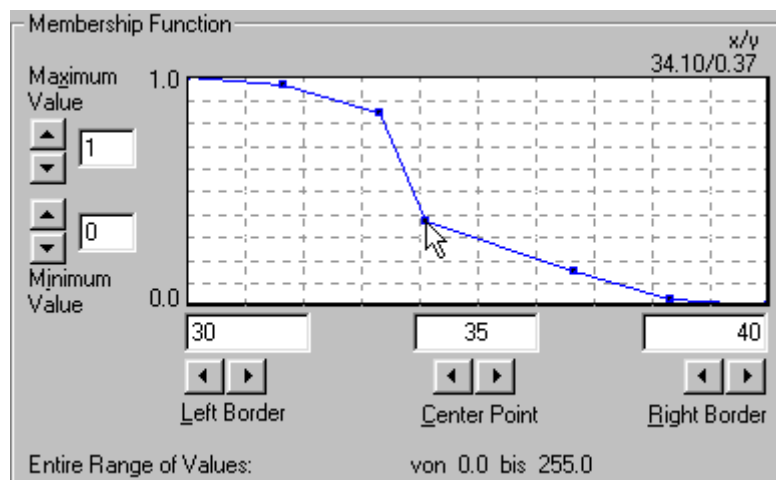


Abb. 27: Editieren der Zugehörigkeitsfunktion.

Alle verfügbaren Merkmale können mit Hilfe der Fuzzy Logic in logische Ausdrücke mit dem Wertebereich von [0...1] in einer Zugehörigkeitsfunktion dargestellt werden. Im INITIALIZE Feld der EDIT MEMBERSHIP FUNCTION Dialogbox kann eine Grundform für die Zugehörigkeitsfunktion ausgewählt werden. Mit der Maus kann diese Grundform allerdings noch den eigenen Bedürfnissen angepaßt werden, indem die blauen Punkte auf der Linie verschoben werden (Abb. 27).

In diesem Fall wird den Grauwerten kleiner als 30 der Wert 1 und den Grauwerten größer als 40 der Wert 0, also keine Klassenzugehörigkeit zugeordnet. Dazwischen folgt die Zugehörigkeit der Funktion. An der Y-Achse läßt sich der Wertebereich auch beschränken, um nur eine 90%ige Zugehörigkeit oder eine Mindestzugehörigkeit zu ermöglichen. Im kleinen Feld rechts unten kann die Reichweite in Pixeln eines Extended Neighbourhood Merkmals angegeben werden. Die so definierten Ausdrücke bzw. Merkmale können untereinander mit logischen Operatoren verknüpft werden, so daß auf diese Weise eine hierarchische, strukturierte Klassenbeschreibung entsteht. In dem Beispiel (Abb. 28) ergibt sich folgende Beschreibung für die Klasse Apfel: „Ein Apfel ist rund und süß und grün oder rot und nicht gelb.“

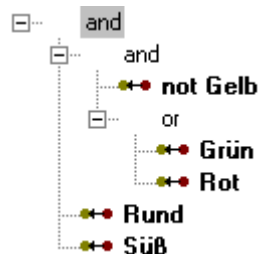


Abb. 28: Beschreibung der Klasse Apfel.

In *eCognition* stehen die logischen Operatoren and, or und arithmetic mean zur Verfügung, um Verknüpfungen zu bilden. Sie werden ebenfalls über die Dialogbox INSERT EXPRESSION eingefügt und können über das mit der rechten Maustaste erreichbare Dialogfeld editiert und auch invertiert (not and, not or und not arithmetic mean) werden. Wenn z.B. von vier Zugehörigkeitsfunktionen ein Ausdruck den Wert 0% und drei den Wert 100% aufweisen, dann liefert der and-Operator den Wert 0%, der or-Operator den Wert 100%, und arithmetic mean liefert den Wert 75% zurück. Ein weiterer Unterschied zum klassischen Box-Klassifikator ist die Möglichkeit der

Darstellung von Kontexteigenschaften der Objekte mit Hilfe von Zugehörigkeitsfunktionen, denn diese sind nicht nur auf Objektmerkmale beschränkt.

### 3.5.2 Nearest Neighbour

Neben den Zugehörigkeitsfunktionen steht der Nearest Neighbour Klassifikator zur Verfügung. Zu Beginn müssen hierzu Beispielobjekte für die einzelnen Klassen definiert werden, die den Trainingsgebieten bei einer überwachten Klassifikation (2.3.1) entsprechen. Wenn der Sample Editor (Abb. 29) unter SAMPLES/Edit Samples... geöffnet wird, befindet sich das Programm im Sample Input Modus.

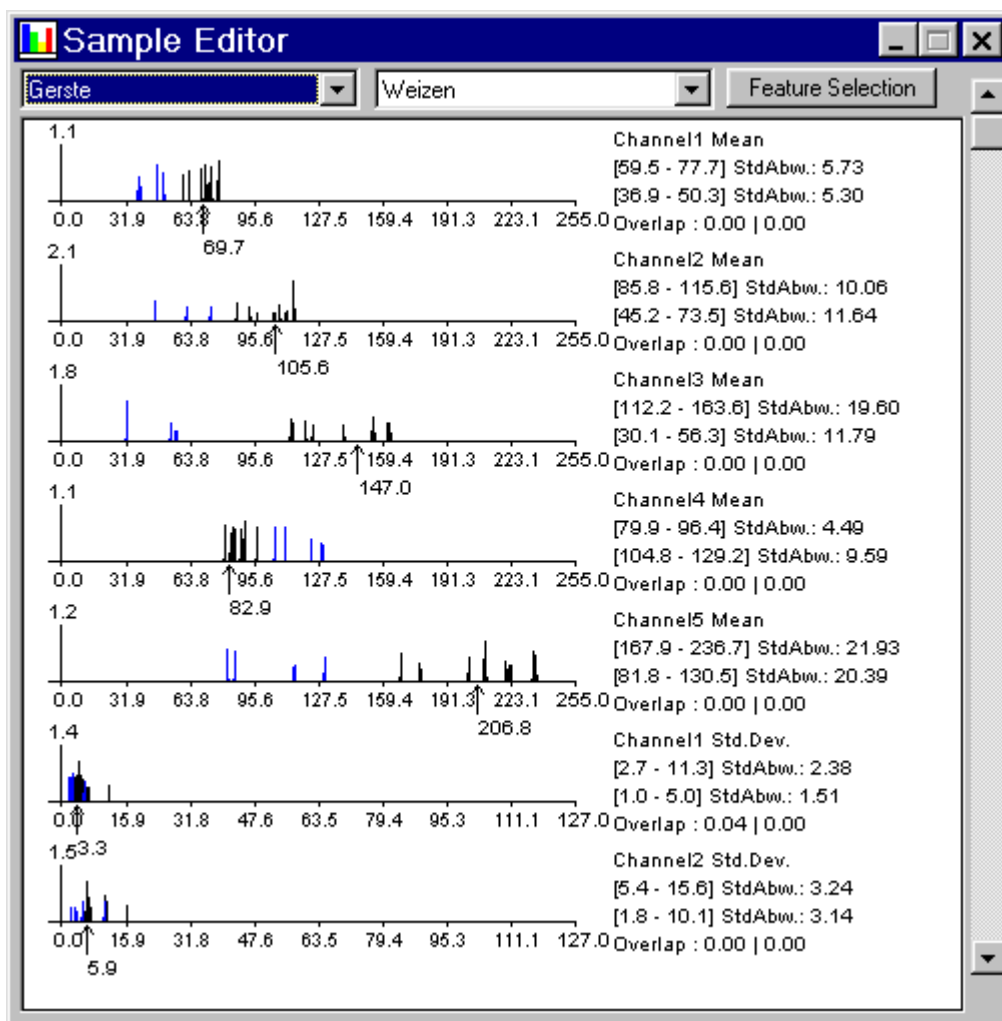


Abb. 29: Im Sample Editor werden die Werte der Beispiele angezeigt und gegenübergestellt.

Durch Doppelklicken von Objekten können diese als Beispiele der im linken Dropdown-Menü angezeigten Klasse definiert werden. Die Eigenschaften dieser Beispiele werden als kleine Histogramme angezeigt, und ein selektiertes Objekt wird durch einen Pfeil dargestellt. Die Eigenschaften der Beispiele einer anderen Klasse können mit dem zweiten Dropdown-Menü der ersten Klasse gegenübergestellt werden. Sie werden blau angezeigt, und man kann so anschaulich die Trennbarkeit zweier Klassen in bestimmten Merkmalen beurteilen.

Welche Merkmale hier für jede Klasse individuell angezeigt werden, wird über den [Feature Selection] Knopf festgelegt. Die hier zur Auswahl stehenden Merkmale werden über SAMPLES/Edit Sample Feature Space... festgelegt. Die hier angezeigten Merkmale sind wiederum von den Einstellungen unter FEATURES/Select active Features... abhängig. In der unter CLASSIFICATION/Edit Next Neighbour Classification... zu erreichenden Dialogbox SELECT FEATURE FOR EACH CLASS können nun simultan für alle Klassen ausgewählte Merkmale, die bei der Nearest Neighbour Klassifikation eingesetzt werden sollen, festgelegt werden. Die so definierten Merkmale werden in der EDIT CLASS DESCRIPTION Dialogbox mit **(generated)** gekennzeichnet (Abb. 30).

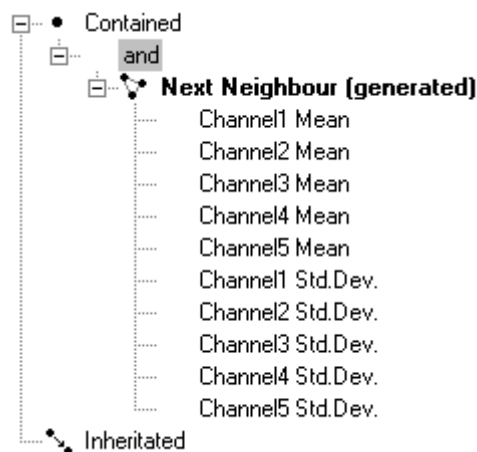


Abb. 30: Generierte Nearest Neighbour Merkmale.

Diese Dialogbox hat dieselbe Bezeichnung und sieht genauso aus wie die im Sample Editor unter [Feature Selection] erreichbare Dialogbox, die allerdings nur die Anzeige im Sample Editor steuert. Diese vielen verschiedenen Möglichkeiten der Auswahl von Merkmalen, die verwendet oder nicht verwendet werden, die angezeigt und nicht angezeigt werden und von den Einstellungen anderer Dialogboxen abhängen, sind



für den Einsteiger zunächst undurchschaubar und verkomplizieren die Bedienung unnötig. *Delphi2* arbeitet allerdings schon an einem verbesserten Dialog, aus dem die hierarchische Staffelung der Auswahlmöglichkeiten klarer hervorgeht. Man kann den Nearest Neighbour Klassifikator allerdings auch wie gewohnt unter INSERT EXPRESSION (Abb. 23) einfügen. Durch einen Doppelklick auf „Next Neighbour“<sup>4</sup> öffnet sich die NEXT NEIGHBOUR FEATURE SPACE Dialogbox, in der die zu verwendenden Merkmale für jede Klasse ausgewählt werden können. Da dieser Klassifikator eine Kontrakklassifikation durchführt, müssen wenigstens zwei Klassen definiert sein. Zusätzliche Klassen können nach Belieben später hinzugefügt werden, indem man einfach für eine neue Klasse Beispiele definiert, den Klassifikator wie gewohnt einfügt und die Klassifikation erneut startet.

Der Sample Editor kann auch zur automatischen Erzeugung von Zugehörigkeitsfunktionen genutzt werden. Dazu klickt man das gewünschte Merkmal mit der rechten Maustaste an, und ein Kontextmenü erscheint (Abb. 31).

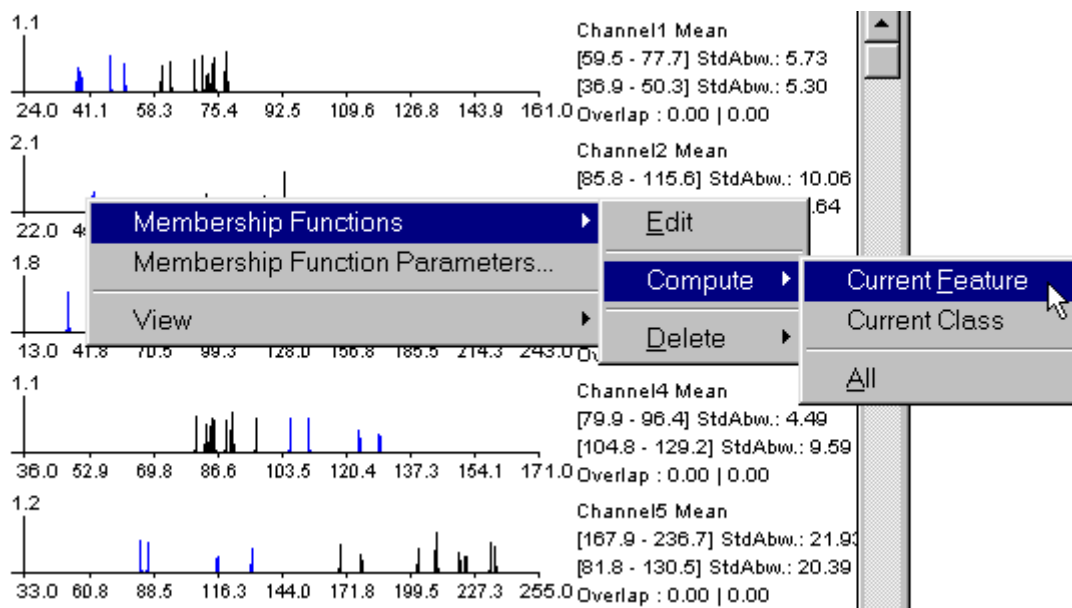


Abb. 31: Automatische Generierung von Zugehörigkeitsfunktionen.

Unter MEMBERSHIP FUNCTIONS/Compute/... wird für das bestimmte Merkmal, für die jeweilige Klasse oder für alle Klassen eine Zugehörigkeitsfunktion berechnet. Ebenso können sie unter MEMBERSHIP FUNCTIONS/Delete/... wieder gelöscht

<sup>4</sup> Der Nearest Neighbour Klassifikator wird hier fälschlicherweise Next Neighbour genannt. Der Fehler ist *Delphi2* allerdings bekannt und wird behoben.

werden. Eine bereits bestehende Funktion kann mit MEMBERSHIP FUNCTIONS/Edit bearbeitet werden. Ist noch keine vorhanden, so wird sie automatisch angelegt.

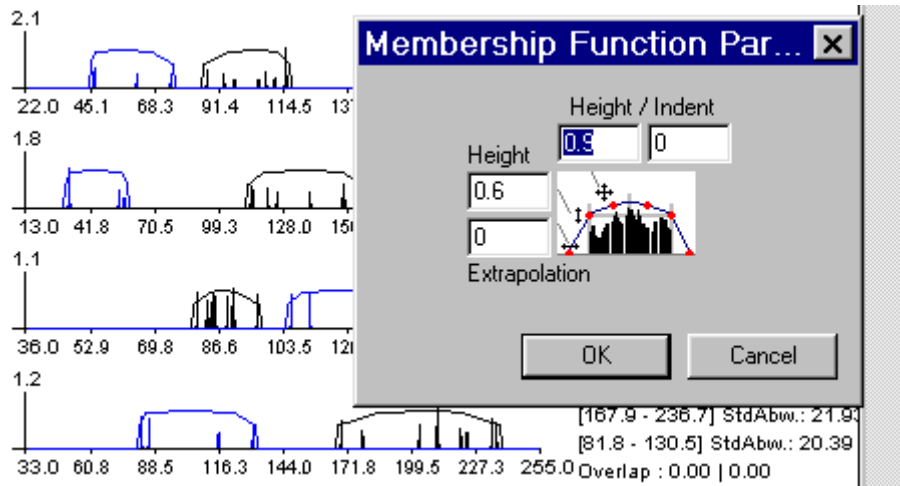


Abb. 32: Anpassung des Kurvenverlaufs der Zugehörigkeitsfunktionen.

Den Verlauf aller Funktionen kann man außerdem mit MEMBERSHIP FUNCTION PARAMETERS/... (Abb. 32) beeinflussen. Eine so erzeugte Zugehörigkeitsfunktion wird im EDIT CLASS DESCRIPTION Fenster mit **(generated)** gekennzeichnet. Unter VIEW/... öffnet sich ein Aufklappmenü, in dem man die Zugehörigkeitsfunktionen und die Achsenbezeichnungen ausblenden kann. Auch die Skalierung der Achsen kann auf den gesamten Wertebereich von z.B. [0...255] oder [0...127] ausgedehnt werden.

Ein weiteres Hilfsmittel zur Analyse bietet der FEATURE SPACE PLOT dar, der unter FEATURES/2D-Feature Plot... zu aktivieren ist (Abb. 33). Hier können zwei beliebige Merkmale in einem zweidimensionalen Merkmalsraum (2.2) gegenübergestellt werden. Durch kleine Kreuze werden die jeweiligen Objekte in ihrer Klassenfarbe angezeigt. Wird ein Objekt selektiert, so erscheint an seiner Stelle im Merkmalsraum ein kleines schwarzes Kreuz. Sind Nearest Neighbour Klassifikatoren vorhanden, so verbinden graue Linien das Bildobjekt mit dem besten Beispiel jeder Klasse. Auf diese Weise lassen sich überflüssige Merkmale und Kanäle finden und können dann zur Komplexitätsreduktion ausgeschlossen werden.

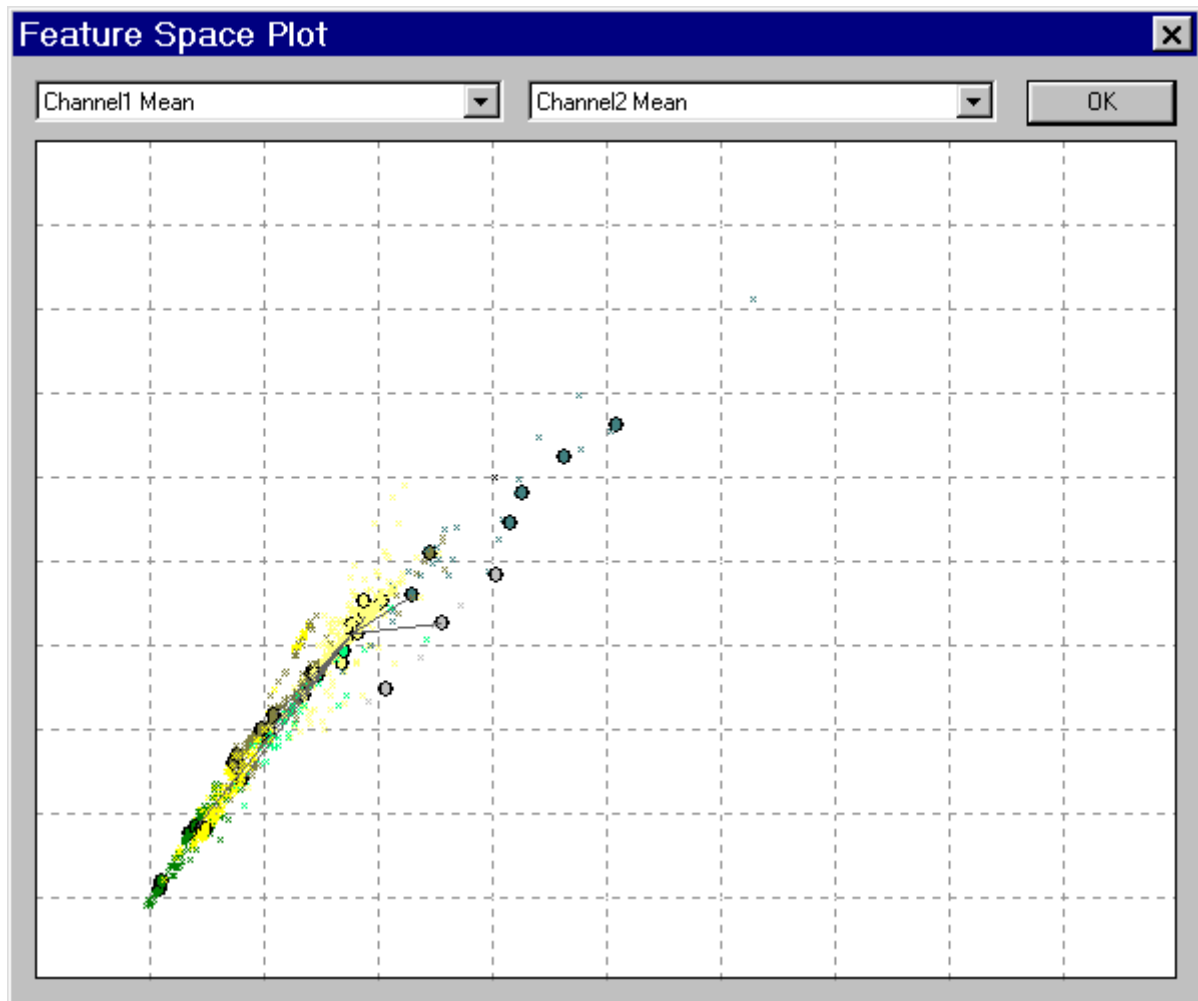


Abb. 33: Feature Space Plot.

### 3.6 Wissensbasierte Segmentierung

In sehr heterogenen Gebieten, wie z.B. in städtischen Bereichen, können die Bildobjekte nicht unter der Annahme der Homogenität extrahiert werden. Auf der Basis von bereits klassifizierten Objektprimitiven können in *eCognition* komplexe Gebiete zu größeren Objekten zusammengefaßt werden. Zwei Methoden stehen hierbei zur Verfügung: Knowledge based Fusion und Knowledge based Border Optimization.

### 3.6.1 Knowledge based Fusion

Zuerst werden mit der Multiresolution Segmentation die Originaldaten segmentiert:

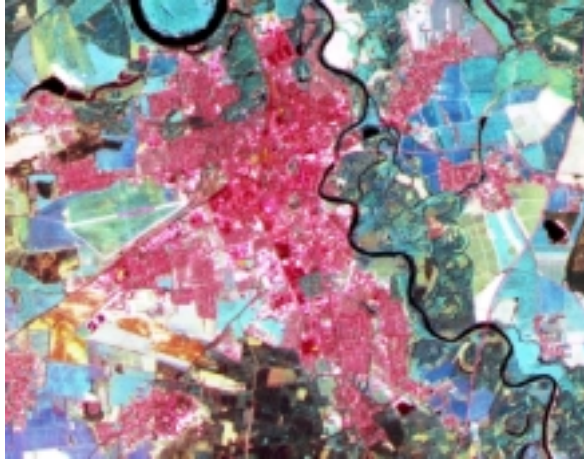


Abb. 35: Originaldaten.



Abb. 34: Segmentierte Daten.

Mit Hilfe von verschiedenen Objekt- und Kontextmerkmalen werden die erhaltenen Objekte klassifiziert (Abb. 36):

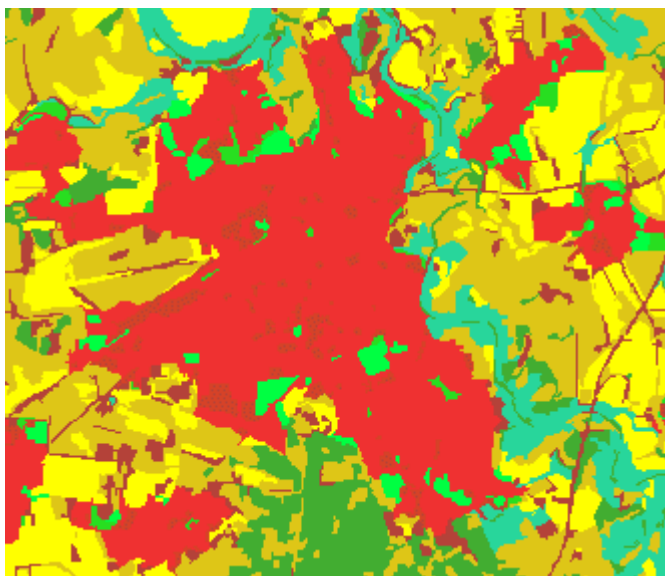


Abb. 36: Klassifizierte Daten.

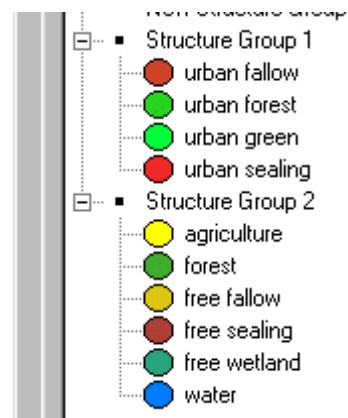


Abb. 37: Klassenhierarchie.

Nun werden bei der wissensbasierten Segmentierung in der Klassenhierarchie unter STRUCTURE zuerst logische Gruppierungen zu Strukturgruppen gebildet (Abb. 37). Danach wird unter SEGMENTATION/Knowledge Based Segmentation/... eine

erneute Segmentierung auf einer existierenden oder auf einer neuen Ebene durchgeführt, wobei die Strukturgruppen einfach zusammengefaßt werden (Abb. 38). Anschließend wird noch einmal klassifiziert (Abb. 39):



Abb. 38: Wissensbasierte Segmentierung.



Abb. 39: Klassifiziertes Endergebnis.

### 3.6.2 Knowledge based Border Optimization

Neben der bloßen Zusammenfassung von Bildobjekten bietet *eCognition* auch die Möglichkeit der wissensbasierten Korrektur durch Optimierung von Grenzen auf der Basis von klassifizierten Unterobjekten.

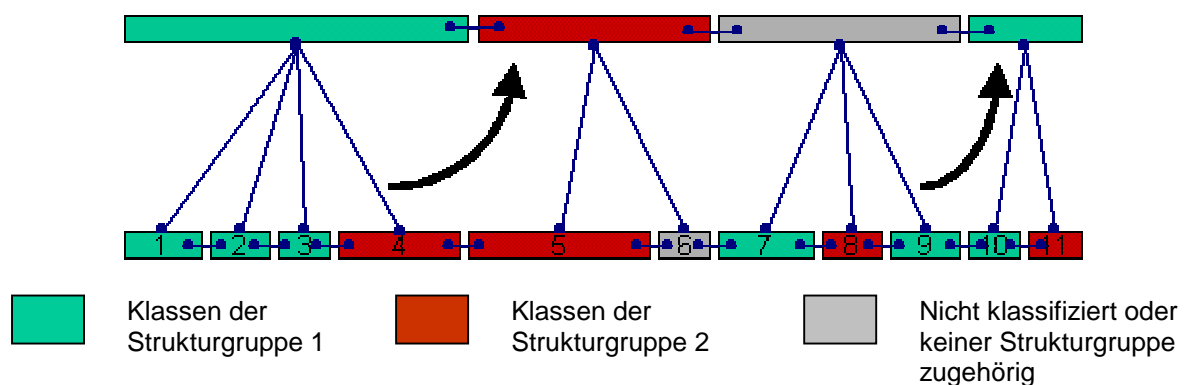


Abb. 40: Objektstruktur vor der Grenzoptimierung.

In diesem Beispiel sind sowohl die Objekte der oberen Ebenen als auch die Unterobjekte schon klassifiziert (Abb. 40). Unter SEGMENTATION/Knowledge

Based Segmentation/Border Optimization wird der Algorithmus gestartet. Dieser testet, ob jedes Unterobjekt an einer Grenze von einem Superobjekt (1, 4, 5, 6, 7, 9, 10 und 11) einer Strukturgruppe (1, 4, 5, 7, 9, 10 und 11) angehört. Ist dies der Fall (1, 5, 10), dann sorgt der Algorithmus dafür, daß die Strukturgruppe mit der des Superobjektes korrespondiert. Stimmen die Strukturgruppen nicht überein (4, 7, 9, 11), sucht der Algorithmus in der direkten Nachbarschaft nach einem Superobjekt derselben Strukturgruppe. Wird ein solches Objekt gefunden (4 und 9), wird das Unterobjekt von seinem ehemaligen Superobjekt gelöst und wird Teil von dem soeben gefundenen Superobjekt. Wird kein besser passendes Superobjekt gefunden (7 und 11), verbleibt das Unterobjekt bei seinem ursprünglichen Superobjekt. Die so erhaltenen Objekte sind in ihrer Form nun korrigiert (Abb. 41).

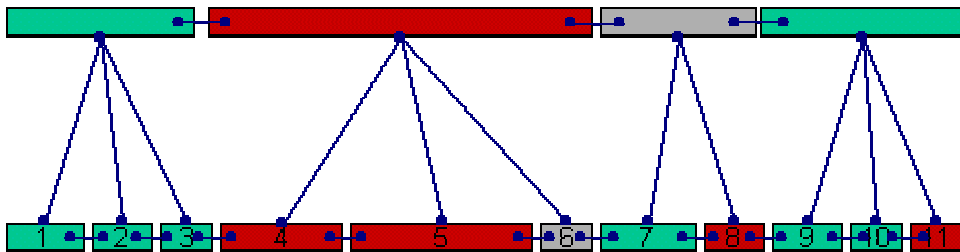


Abb. 41: Objektstruktur nach der Grenzoptimierung.

## 4 Anwendung des Bildanalysestystems eCognition

### 4.1 Zielsetzung, Datenmaterial und Rechnerausstattung

In diesem Kapitel sollen nun die Anwendungsmöglichkeiten des Bildanalysestystems *eCognition* vorgestellt und bewertet werden. Weiterhin wird ein Vergleich mit herkömmlichen Methoden der Klassifikation durchgeführt. An Hand eines Testbildes wird ein möglicher Auswertegang beschrieben, der indes aufgrund der Flexibilität des Programms auch anders gelöst werden könnte. Allerdings sind die hier gezeigten Arbeitsschritte typische Vorgehensweisen, die auch mit anderem Bildmaterial angewendet werden können. Bei diesem Bild handelt es sich um einen Ausschnitt aus einem Streifen, der mit dem DPA-Scanner aufgenommen wurde. Der DPA-Scanner ist die flugzeuggetragene Version der MOMS-Kamera, also ein digitaler Dreizeilenscanner (*Schlüter 1999*). Die Aufnahme entstand am 11.07.1997 um 11:50 Uhr (MEZ) aus einer Flughöhe von ca. 3300 m. Zur Auswertung liegen die multispektralen Kanäle (Rot, Grün, Blau und NIR) mit einer Bodenauflösung von ca. 0,8 m vor. Der panchromatische Kanal mit einer Auflösung von ca. 0,4 m steht nicht zur Verfügung.

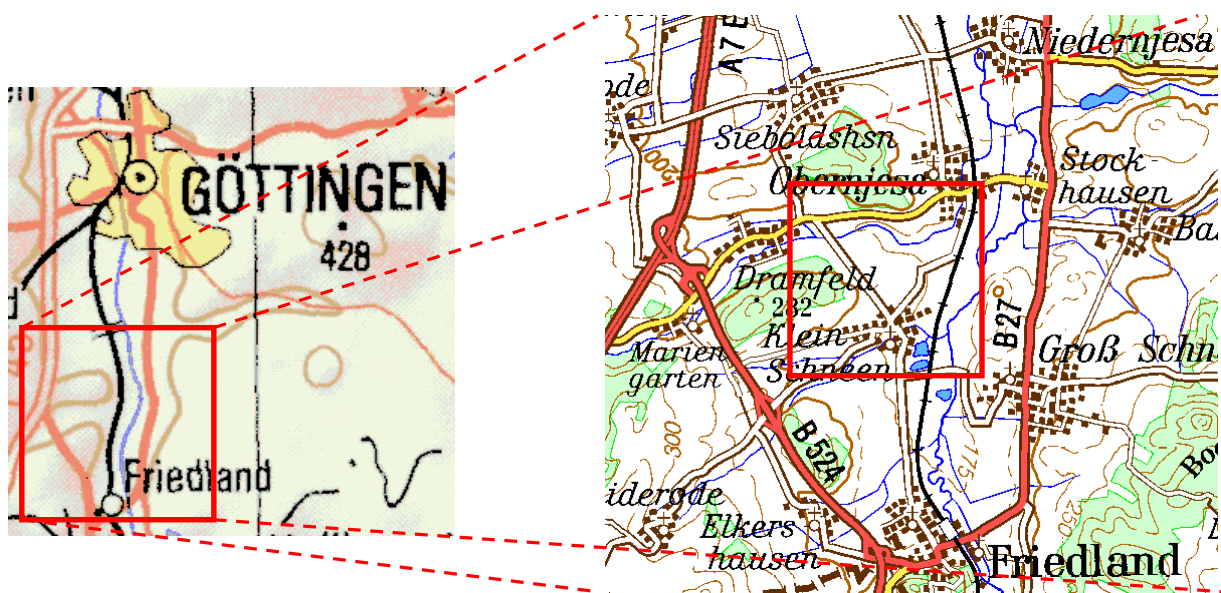


Abb. 42: Gewähltes Testgebiet der DPA-Aufnahme.

Der gewählte Ausschnitt zeigt ein Gebiet in der Nähe von Friedland bei Göttingen (Abb. 42), welcher sowohl Felder als auch Ortslagen enthält. Weiterhin sind Gewässer und eine Bahnlinie zu sehen, so daß eine facettenreiche Auswertung vorliegt. Um die Möglichkeiten des Programms *eCognition* zu untersuchen, wird der gesamte Ausschnitt mit allen sichtbaren Dingen klassifiziert, obwohl man sich in der Praxis manchmal auf ausgesuchte Objekte wie z.B. Felder konzentrieren würde. Die Daten über die Anbausorten in diesem Gebiet wurden in Zusammenarbeit mit dem Katasteramt Göttingen und dem Amt für Agrarstruktur Göttingen ermittelt.

Das Bild umfaßt pro Kanal ca. 10 MByte an Daten, so daß also 40 MByte an Bildmaterial zur Auswertung vorliegen. Die Berechnungen erfolgten auf einem Rechner mit Pentium II, 350 MHz Prozessor und 128 MByte Arbeitsspeicher. Es sei schon hier festgestellt, daß die Arbeitsspeicherausstattung des Rechners so groß wie möglich sein sollte, da das Programm bei der Segmentierung eine umfangreiche Matrix für die neue Segmentierungsebene im Arbeitsspeicher halten muß. Die 128 MByte sind bei diesem 40 MByte großen Datensatz schon zu wenig, denn bereits beim Einlesen der vier Kanäle werden die 128 MByte komplett ausgeschöpft, da während des Importierens der Bitmaps die unterste Ebene des hierarchischen Netzes angelegt wird. Bei den weiteren Berechnungen muß das Programm auf den langsamen virtuellen Arbeitsspeicher von Windows (Auslagerungsdatei) zurückgreifen. Obwohl die Segmentierung mit allen vier Kanälen noch möglich war, kann man bei dieser Datenmenge mit dem Testrechner nicht ohne Komplikationen weiterarbeiten. Möchte man die fertige Klassifikation z.B. exportieren, d.h. in die Zwischenablage kopieren, stürzt das Programm meistens ab, so daß das Bild in vier gleichgroße Teile zerlegt werden mußte.



## 4.2 Anwendung der Segmentierung

### 4.2.1 Allgemeine Vorgehensweise

Nachdem man die gewünschten Bilder in *eCognition* eingelesen hat, beginnt man als ersten Schritt der Auswertung mit der Segmentierung. Diese bildet die Grundlage für die anschließende Klassifikation und sollte entsprechend sorgfältig durchgeführt werden. Da die Segmentierung sehr rechenintensiv ist, sollte man die Segmentierungsparameter an Hand eines kleinen Ausschnittes der auszuwertenden Bilder einstellen. Um optimale Ergebnisse bei der Klassifikation zu erzielen, ist es zweckmäßig, ausgiebig mit den einzelnen Parametern zu experimentieren, bis man eine geeignete Einstellung erreicht hat. Danach kann die Segmentierung des gesamten Bildes erfolgen. Je brauchbarer die erhaltenen Objekte sind, desto genauer wird die anschließende Klassifikation, da diese sich immer auf die Objekte der Segmentierung stützt.

Der Segmentierungsalgorithmus basiert auf der Annahme, daß die Bildobjekte durch Kontinuität in Farbe und Form gekennzeichnet sind. Dieser Algorithmus ist ein heuristischer Optimierungsprozeß, bei dem die mittlere Heterogenität eines Objektes mit dessen Größe gewichtet wird, und die Summe dieser gewichteten Heterogenitäten über alle Objekte minimiert wird. Die Heterogenität eines Objektes wird in *eCognition* folgendermaßen definiert (*Baatz/Schäpe 1999*):

- *Farbheterogenität*: Summe über alle Kanäle von den Standardabweichungen der spektralen Werte der enthaltenen Pixel.
- *Formheterogenität*: Relative Abweichung der Kantenlänge von der kompaktesten Form (Kreis), die auf zwei Arten definiert werden kann:
  - *Flächenkriterium*: Kantenlänge eines Kreises mit demselben Flächeninhalt wie das Objekt. Diese Definition verhindert eine zu große Abweichung von der kompaktesten Form.

- *Kantenkriterium*: Kantenlänge des Rechtecks, welches das Objekt entlang des Rasters umschließt. Diese Definition verhindert eine zu große Abweichung von der kürzesten Kantenlänge.

Diese drei Kriterien können in gemischter Form bei der Segmentierung berücksichtigt werden. Sie entsprechen den Parametern WEIGHT OF SHAPE HOMOGENEITY und WEIGHT OF COMPACTNESS IN SHAPE HOM. in der Segmentierungsdialogbox (Abb. 18). Das Farbkriterium erweist sich als das wichtigste, um sinnvolle Objekte zu erhalten. Die Qualität der Objektextraktion kann allerdings durch Berücksichtigung des Formparameters optimiert werden, da die Kompaktheit eine wichtige Eigenschaft von Objekten ist. Besonders in stark texturierten Daten wie Radarbildern verhindert dieser Parameter ein fraktales Aussehen der Objekte.

### 4.2.2 Einstellen der Parameter

Um Rechenzeit zu sparen, werden zunächst nur zwei Ausschnitte zum Einstellen der Parameter benutzt. Diese Ausschnitte sind repräsentative Beispiele für einen homogenen und einen heterogenen Bereich des Gesamtbildes. Im homogenen Bereich (Bereich 1) finden sich vorherrschend große Flächen von Feldern (Abb. 43), und im heterogenen Ausschnitt (Bereich 2) erkennt man viele kleine Flächen, die sich aus den Häusern, Straßen und anderen kleineren Objekten ergeben (Abb. 44). Aufgrund dieser zwei verschiedenen Bereiche im Bild kann man das Segmentierungsverfahren nicht einem Gebiet anpassen, weil für beide jeweils unterschiedliche Einstellungen notwendig sind. Daher segmentiert man zunächst das heterogene Gebiet und legt danach eine Ebene niedrigerer Auflösung für das homogene Gebiet an. Es ist dabei ratsam, mit kleinen Flächen, d.h. mit einem niedrigen OBJECT SCALE PARAMETER, zu beginnen. Dies liegt darin begründet, daß bei der Erzeugung zusätzlicher Segmentierungsebenen die Objektgrenzen den Objekten der nächstniedrigeren Ebene folgen müssen. Die Segmentierung ist auch durch die Grenzen des nächsthöheren Objektes beschränkt. Bildet man also zuerst große Objekte, dann könnten die anknüpfenden kleineren Objekte zu ungenau sein, da sie dem Verlauf der Superobjekte folgen müssen. Aus diesem Grund sollte man auch bei größeren,

homogenen Flächen mit kleinen Segmenten beginnen, die danach auf einer höheren Ebene zu größeren Segmenten verbunden werden. Beim Ausprobieren der Parametereinstellungen wird man feststellen, daß die erhaltene Segmentierung trotz Verwendung derselben Parameter jedesmal etwas anders aussieht. Dies liegt darin begründet, daß die Segmentierung unüberwacht erfolgt und einige stochastische Elemente mit einer hohen Anzahl an Freiheitsgraden zur Initialisierung enthält.



Abb. 43: Bereich 1.



Abb. 44: Bereich 2.

Ein OBJECT SCALE PARAMETER von 70 liefert im dörflichen Bereich ansprechende Ergebnisse, wobei alle vier DPA-Kanäle genutzt werden. Dieser Wert wurde durch Experimentieren gefunden. Die anderen Parameter verbleiben zunächst auf den Standardwerten (Abb. 45). Einen generellen Startwert kann man nicht empfehlen, da dieser Wert von der Bildauflösung und von der Anzahl der Kanäle abhängig ist. Allerdings sollte man bei niedrigauflösten Daten einen kleinen Wert wählen, um genügend Details zu erfassen. Bei hochauflösten Bildern würde dies dagegen nur unnötige Teilstücke bilden. Da dieser Wert die Summe der Heterogenität über alle Kanäle darstellt, ist es bei der Verwendung vieler Kanäle nicht empfehlenswert, einen niedrigen OBJECT SCALE PARAMETER einzustellen, denn dabei würden sehr viele kleine und unbrauchbare Segmente entstehen. Auch im homogenem Bereich sind die Ergebnisse zufriedenstellend (Abb. 46). Die hier erhaltenen Segmente

werden allerdings noch später auf einer höheren Ebene zusammengefaßt, womit die Eventualität von Fehlklassifizierungen innerhalb eines Feldes verringert wird.



Abb. 45: Segmentierung mit einem OBJECT SCALE PARAMETER von 70 im Bereich 2.



Abb. 46: Segmentierung mit einem OBJECT SCALE PARAMETER von 70 im Bereich 1.

Um die Form der Segmente zu optimieren, werden nun die anderen beiden Parameter angepaßt. Mit WEIGHT OF SHAPE HOMOGENEITY kann man den Anteil der Berücksichtigung von Form im Verhältnis zur Farbe regeln (s. Formel (14)). Der voreingestellte Wert von 0,2 deutet darauf hin, daß der Farbanteil, wie in (4.2.1) erwähnt, einen wichtigeren Anteil an der Erzeugung sinnvoller Objekte als der Formparameter besitzt. Wenn der Formparameter genutzt wird, kann man mit WEIGHT OF COMPACTNESS IN SHAPE HOM. die Gewichtung der Compactness (Kreisfläche) steuern (s.a. 4.2.1). Der Standardwert hierfür ist auf 0,3 eingestellt, da in Fernerkundungsdaten in der Regel kompakte Objekte nicht vorkommen. Der Kantenparameter läßt dagegen auch längliche Objekte zu und optimiert eine möglichst geringe Randrauhigkeit. Dennoch kann man die Compactness in stark texturierten Daten nutzen. Zur Verdeutlichung dieser Parameter werden nun einige Bilder gezeigt, die u.a. auch mit Extremeinstellungen erhalten wurden. Danach wird versucht, sich einer optimalen Segmentierung anzunähern, wobei der OBJECT SCALE PARAMETER immer auf den Wert 70 eingestellt bleibt.

Im ersten Beispiel (Abb. 47) wird der WEIGHT OF SHAPE HOMOGENEITY Wert auf Null gesetzt, so daß also nur Farbe und keine Form einbezogen wird. Die Einstellung von WEIGHT OF COMPACTNESS IN SHAPE HOM. hat demnach hier keine Auswirkung. Wie man sieht, entstehen auch ohne Berücksichtigung der Form sinnvolle Objekte. Bezieht man allerdings nur den Formparameter ein und setzt die Compactness auf Null, so ist das Ergebnis völlig unbrauchbar (Abb. 48).



Abb. 47: WEIGHT OF S. H. = 0.

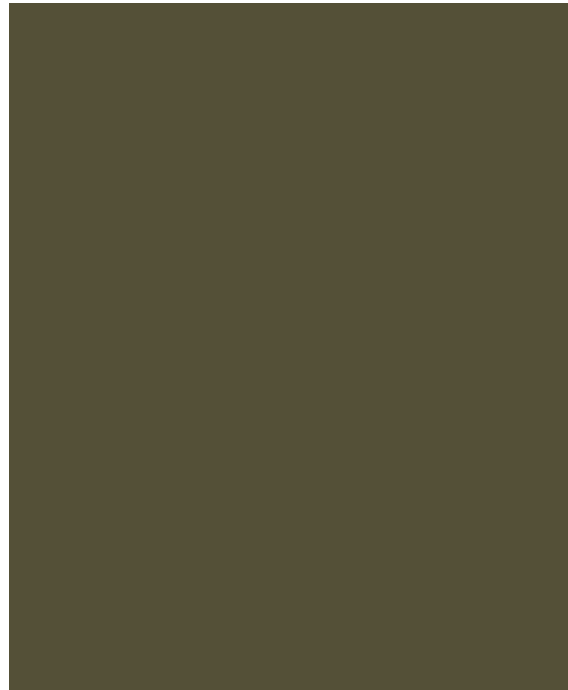


Abb. 48: WEIGHT OF S. H. = 1, WEIGHT OF C. IN S. H. = 0.

Stellt man die Compactness zusätzlich auf den Wert Eins, so nehmen die Segmente eine möglichst kompakte Form an. Daß die Objekte in Fernerkundungsdaten meistens nicht sehr kompakt sind, sieht man am Ergebnis dieser Segmentierung: Die erhaltenen Segmente gleichen einem Flickenteppich und lassen keinen Bezug zum Originalbild mehr erkennen (Abb. 49). Beläßt man den Formanteil auf 0,2 und setzt den Anteil an Compactness auf Eins, so ist das Ergebnis viel ansprechender (Abb. 50). Bei genauerer Betrachtung erkennt man allerdings, daß die Kanten an z.B. Häusern viel zu rund sind. Daher ist es also sinnvoll, den Compactness Parameter zu reduzieren, um auch längliche Objekte mit geringer Randrauhigkeit zuzulassen. Aber auch wenn der Wert auf Null gesetzt wird, sind die Kanten nicht viel glatter geworden (Abb. 51). Daher sollte man versuchen, den Formparameter etwas zu

erhöhen. Nach einigem Probieren findet man schließlich eine optimale Einstellung mit einem Formanteil von 0,5 und dem Compactness Parameter auf 0,1 (Abb. 52).

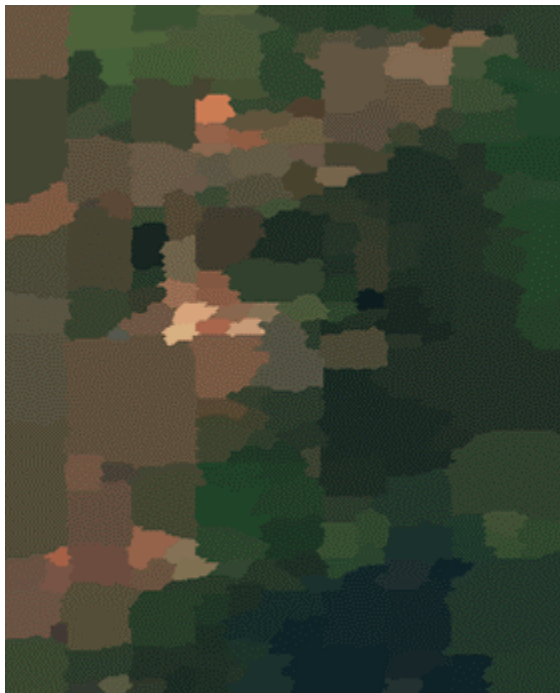


Abb. 49: WEIGHT OF S. H. = 1, WEIGHT OF C. IN S. H. = 1.



Abb. 50: WEIGHT OF S. H. = 0,2 , WEIGHT OF C. IN S. H. = 1.



Abb. 51: WEIGHT OF S. H. = 0,2 , WEIGHT OF C. IN S. H. = 0.



Abb. 52: WEIGHT OF S. H. = 0,5 , WEIGHT OF C. IN S. H. = 0,1.

Es sind zwar nicht alle kleinen Häuser genau zu erkennen, aber zum klassifizieren von versiegelten Flächen reicht die Genauigkeit aus. Erhöht man den Formparameter, so neigt der Algorithmus im allgemeinen dazu, größere Objekte zu bilden. Deshalb kann es manchmal sinnvoll sein, den OBJECT SCALE PARAMETER ein wenig nach unten zu korrigieren. Auch im homogenen Außenbereich ist das Ergebnis zufriedenstellend (Abb. 53), so daß nur noch die Frage bleibt, ob man das Bild mit einer vierfachen oder einer achtfachen Nachbarschaft segmentiert. Mit einer achtfachen Nachbarschaft dauert die Berechnung etwas länger, so daß man erwägen muß, ob dieser Zeitverlust gerechtfertigt ist. In diesem Fall sind die Segmentgrenzen noch etwas gerader (Abb. 54), was man an den Häusern und an den Feldgrenzen erkennt.



Abb. 53: WEIGHT OF S. H. = 0,5 , WEIGHT OF C. IN S. H. = 0,1.

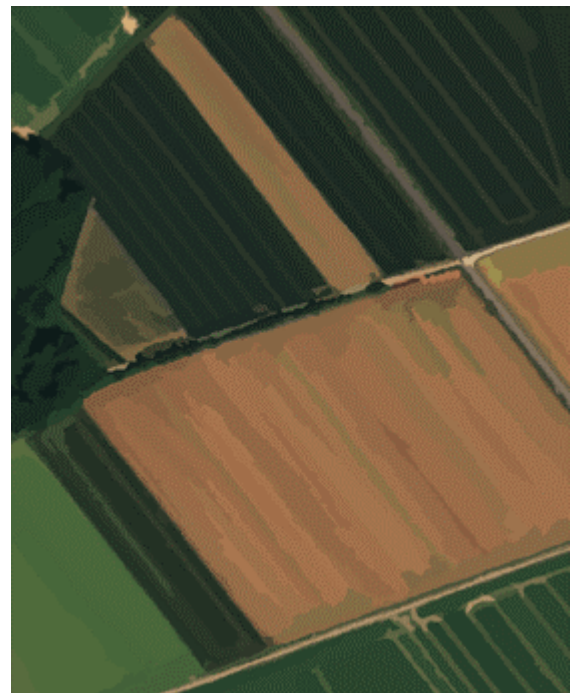


Abb. 54: WEIGHT OF S. H. = 0,5 , WEIGHT OF C. IN S. H. = 0,1 und achtfacher Nachbarschaft

Nun kann damit begonnen werden, eine zweite Ebene mit niedrigerer Auflösung anzulegen. Hier wird als erstes wieder der OBJECT SCALE PARAMETER angepaßt. Mit einem Wert von 120 erhält man nun Segmente, die in einigen Fällen ein ganzes Feld umfassen und somit Fehlklassifizierungen verringern (Abb. 55). Die anderen Parameter behalten ihre Einstellung zunächst, denn der Verlauf der Grenzen der Objekte ist durch die nun schon vorhandenen Subobjekte weitestgehend vorgegeben. Es ist darauf zu achten, daß die neue Ebene in der Hierarchie über der

schon existenten angelegt wird, da diese sonst die neuen Werte ebenfalls übernimmt. Dazu klickt man in der Dialogbox im Bereich LEVEL auf NEW LEVEL zwischen ENTIRE SCENE und LEVEL 1. Da die einzelnen Segmente noch stärker zusammengefaßt werden könnten, wird der Compactness Parameter auf den Wert 0,4 erhöht. Die Grenzen bleiben dabei gerade, da sie durch die Subobjekte vorgegeben sind, und die Segmente werden noch besser zusammengeschlossen. Der Einfluß dieses Parameters wird noch größer, wenn der Formanteil auf 0,8 erhöht wird. Im dörflichen Bereich sind die Segmente zwar relativ groß, aber sie eignen sich dazu, das Dorf als einen zusammengehörenden Bereich zu klassifizieren (Abb. 56).

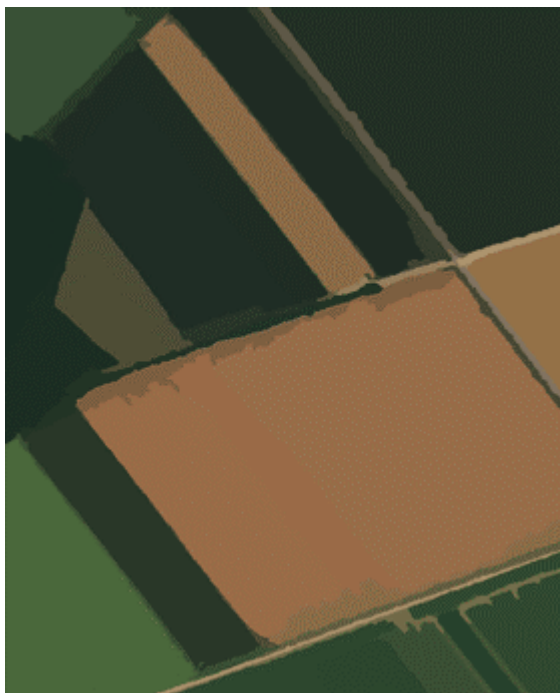


Abb. 55: WEIGHT OF S. H. = 0,8 , WEIGHT OF C. IN S. H. = 0.4.



Abb. 56: WEIGHT OF S. H. = 0,8 , WEIGHT OF C. IN S. H. = 0.4.

Mit dem Ergebnis kann man zufrieden sein, so daß man nun das gesamte Bild segmentieren kann, wobei mit Gesamtbild hier ein Viertel gemeint ist. Dieser Vorgang dauert auf dem Testrechner unter Berücksichtigung einer achtfachen Nachbarschaft ca. 5 Minuten. Segmentiert man nur mit einer vierfachen Nachbarschaft, so dauert dieser Vorgang lediglich 3,5 Minuten. Mit dem nicht geviertelten Bild dauert die Berechnung ca. 23 bzw. 17 Minuten. Weiterhin ist es empfehlenswert, die Größe der Auslagerungsdatei nicht von Windows verwalten zu lassen, sondern man sollte ihr eine feste Größe zuweisen. Dabei spart man bei dem 40 MByte großem Bild ungefähr 1 Minute Rechenzeit, weil Windows die erforderliche Größe nicht



ständig selbst berechnen muß. Im Anschluß wird auch gleich die zweite Ebene angelegt, deren Berechnung nur noch einige Sekunden erfordert. Die Berechnung weiterer Ebenen erfolgt immer sehr viel schneller, denn die schon vorhandenen Segmente brauchen nur noch zusammengefaßt zu werden. Danach kann man sich ein Konzept für die Klassifikation überlegen (4.3).

### 4.2.3 Segmentierung von panchromatischen Bildern

Da viele Luftbilder nur in Graustufen vorliegen, ist es sinnvoll zu untersuchen, was die Segmentierung hierbei leistet. Auch in vielen Fernerkundungssatelliten steht neben den multispektralen Kanälen auch ein höher auflösender panchromatischer Kanal zur Verfügung. Hierbei ist zu beachten, daß *eCognition* nur das erfassen kann, was auch der Mensch zu unterscheiden vermag. In dem vorliegenden Bildmaterial kann man z.B. die Feldwege kaum von den Straßen unterscheiden. Das Material der Fahrwegdecke, sofern vorhanden, ist nicht zu erkennen. Die etwas breiteren Fahrwege durch Dörfer kann man als Straßen identifizieren. Die Fahrwege zwischen Feldern wird man dagegen eher den Feldwegen zuordnen. Dies sind allerdings Merkmale, wie sie nur bei der Klassifikation berücksichtigt werden können und bei der Segmentierung noch keine Rolle spielen. Ebenso kann man die einzelnen Nadelwaldbereiche innerhalb des Laubwaldes nur an der etwas anderen Textur erkennen, was aber eventuell schon bei der Segmentierung erfaßt werden kann.

Also startet man wieder mit Testsegmentierungen in den zwei ausgewählten Bereichen mit einem OBJECT SCALE PARAMETER von 70. Das Ergebnis im Bereich 2 ähnelt dem mit vier multispektralen Kanälen, wobei hingegen im panchromatischen Bild weniger Details zu erkennen sind (Abb. 57). Die Detailauflösung kann durch die Verringerung des OBJECT SCALE PARAMETERS auf einen Wert von 60 und durch die Berücksichtigung einer achtfachen Nachbarschaft verbessert werden (Abb. 58).

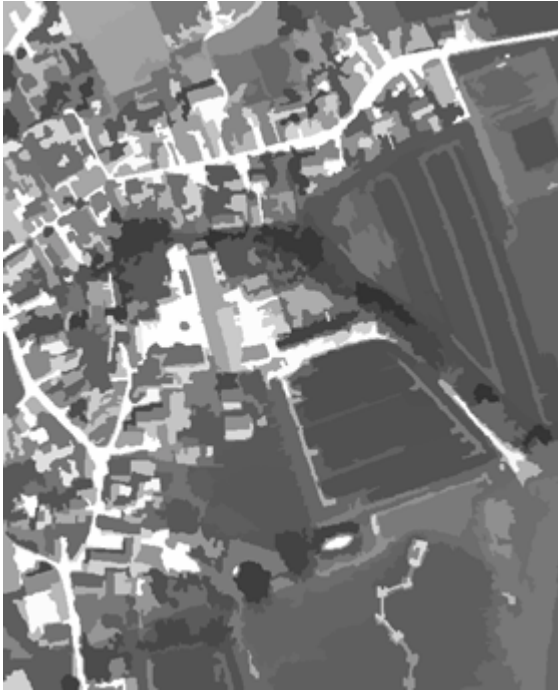


Abb. 57: OBJECT SCALE PARAMETER = 70.

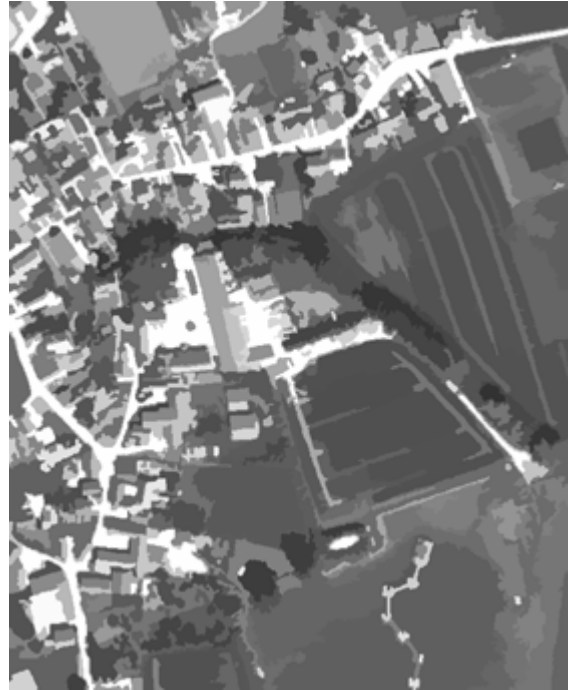


Abb. 58: OBJECT SCALE PARAMETER = 60 und achtfache Nachbarschaft.

Experimentiert man mit den anderen beiden Parametern, so gelangt man mit denselben Einstellungen wie bei den multispektralen Kanälen zum besten Ergebnis (Abb. 59), wobei die Unterschiede zugegeben sehr subtil sind. Das Resultat im Bereich 1 sieht hier schon passender aus (Abb. 60), was darauf schließen läßt, daß die Segmentierung in homogenen Bereichen bessere Ergebnisse liefert. Dies liegt aber in der Natur der Sache, da die größeren Segmente in homogenen Gebieten leichter zu klassifizieren sind, was auch auf die multispektralen Kanäle zutrifft. In bebauten Bereichen stellt sich die Frage, ob hier jede Kleinigkeit klassifiziert werden soll, oder ob man diesen Bereich als Ganzes betrachtet. Wie man an dem rot gekennzeichneten Segment sieht, werden die Feldwege und die Straßen erwartungsgemäß zusammengefaßt, da sie im Bild praktisch gleich aussehen. Weiterhin ist es fraglich, ob die mit dem panchromatischen Kanal erhaltenen Segmente zur Klassifikation geeignet sind, denn es fällt dem Betrachter schwer, diese einer bestimmten Klasse zuzuordnen. Diese Frage soll aber erst in einem nachfolgenden Kapitel (4.3.3) geklärt werden.<sup>5</sup>

<sup>5</sup> Die Bilder werden systembedingt in einer Auflösung von 96 dpi dargestellt. Da die Originalauflösung der Luftbilder 72 dpi beträgt, sind die Bilder um 25% kleiner abgebildet. Die hier gezeigten segmentierten Bilder sind bezogen auf 96 dpi auf 50% bzw. bezogen auf 72 dpi auf 37,5% verkleinert worden. In voller Größe sind einige im Anhang (7.2) zu finden.

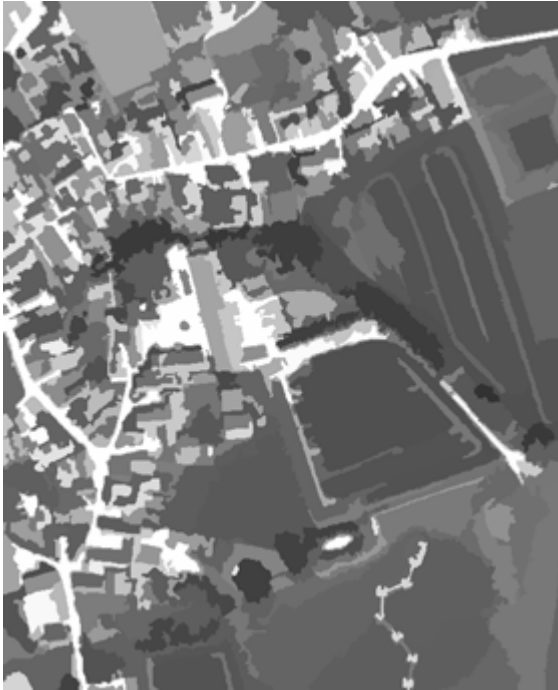


Abb. 59: WEIGHT OF S. H. = 0,5 , WEIGHT OF C. IN S. H. = 0.

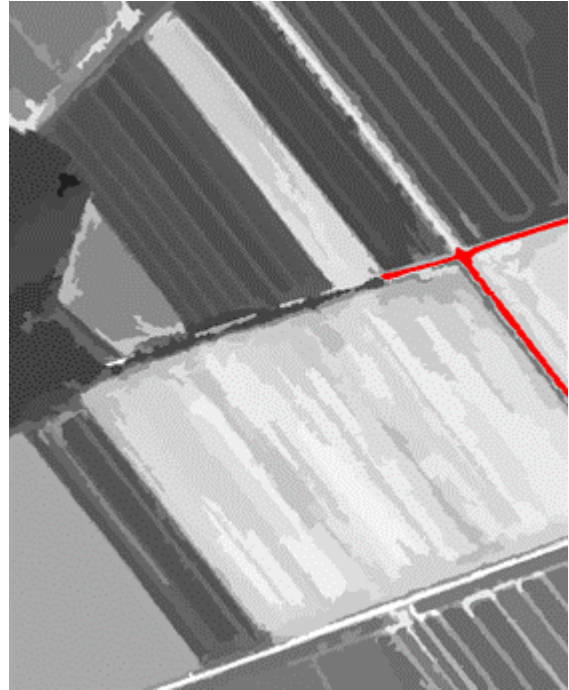


Abb. 60: WEIGHT OF S. H. = 0,5 , WEIGHT OF C. IN S. H. = 0.

### 4.2.4 Verwendung zusätzlicher Kanäle

In diesem Kapitel wird untersucht, ob die Einbeziehung eines Texturkanals die Segmentierung verbessern kann. Dazu wurden mit dem Programm *TNTmips* der Firma *MicroImages* einige Texturkanäle berechnet. Unter den dort zur Verfügung stehenden Algorithmen ist der Typ Range ausgesucht worden. Hierbei wird im Filterfenster (3\*3 Pixel) der Grauwertumfang berechnet und um einen einstellbaren Faktor verstärkt, um eine zweckmäßige Helligkeit und Kontrast zu erhalten. Es wird die Textur aus dem dritten DPA-Kanal ausgewählt, der von der Anschauung her die Segmentierung am besten unterstützen könnte, da hier einige Objektgrenzen deutlich hervorgehoben sind (Abb. 61). Die anschließende Segmentierung wird mit den identischen Parametern wie zuvor durchgeführt. Wie man sieht, sind die erhaltenen Segmente (Abb. 62) nicht passender als vorher (Abb. 54). Die Einbeziehung eines Texturkanals bei einer Segmentierung ist also nicht erforderlich. Ob der Texturkanal allerdings die Klassifikation unterstützt, wird in einem folgenden Kapitel behandelt werden (4.3.4).

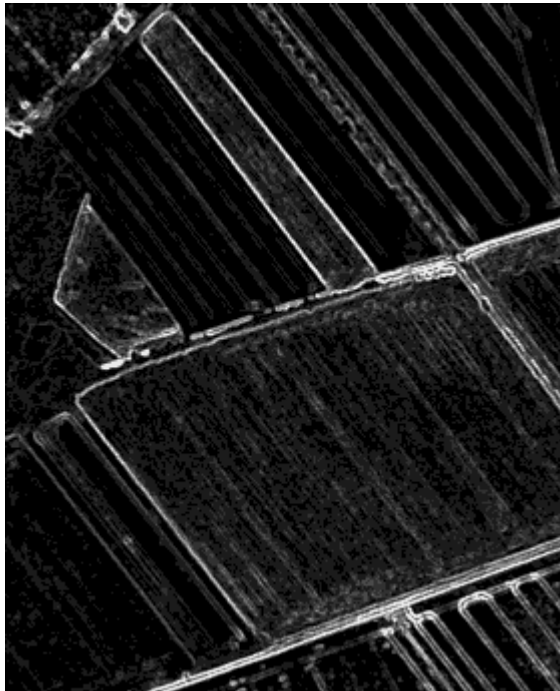


Abb. 61: Textur vom DPA-Kanal 3.



Abb. 62: Segmentierung mit Texturkanal. Parameter wie in Abb. 54.

#### 4.2.5 Segmentierung von Bildern anderer Auflösungsstufen

In diesem Kapitel wird die Segmentierung am gleichen Bildmaterial in den Auflösungen von 10 (Abb. 63) und 4 (Abb. 64) Metern getestet. Die Auflösung des Originalbildes (Pixel/Inch) wurde dazu in einem Bildeditor entsprechend reduziert.



Abb. 63: Auflösung von 10 Metern in Originalgröße.



Abb. 64: Ausschnitt aus dem Bild mit 4 Meter Auflösung in Originalgröße.

Eine Auflösung von 4 Metern erreicht der im September 1999 gestartete Satellit Ikonos im multispektralen Modus, und die 10 Meter entsprechen ungefähr der Auflösung von Satelliten wie IRS und SPOT. Mit diesen auflösungsreduzierten Bildern läßt sich also abschätzen, welche Ergebnisse mit Bildern dieser Satelliten zu erzielen wären.

Bei diesen Auflösungen kann man die bisherigen Einstellungen nicht einfach übernehmen. Zumindest der OBJECT SCALE PARAMETER sollte der Auflösung entsprechend verringert werden. Die Segmentierung mit diesem insgesamt 1,6 MByte großen Bildmaterial für das Bild mit 4 Meter Auflösung dauert nur einige Sekunden, so daß man nach kurzer Zeit des Probierens eine optimale Einstellung des OBJECT SCALE PARAMETERS von 50 findet. Bei dieser Auflösung ist es empfehlenswert, das Bild mit einer achtfachen Nachbarschaft zu segmentieren, um auch Details zu erfassen. Die anderen Parameter verbleiben zunächst auf ihren Standardwerten. Auch Straßen und Wege bilden noch separate Segmente, obwohl sie meistens nur zwei oder drei Pixel breit sind. Zur Verdeutlichung der Objektgröße ist jeweils ein Segment rot markiert (Abb. 65 und Abb. 66).

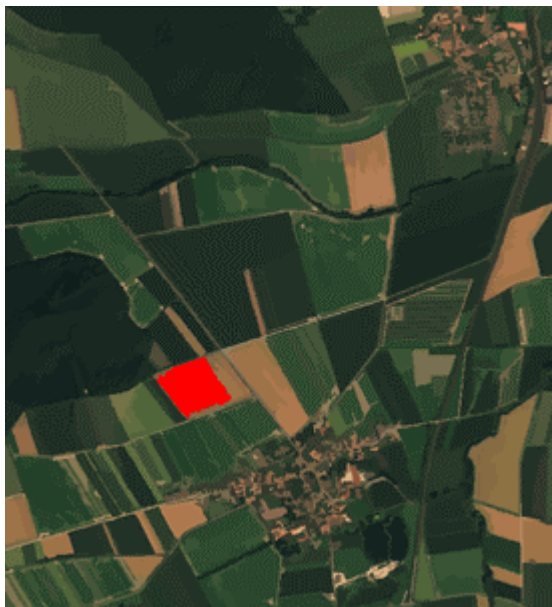


Abb. 65: Segmentierung bei einer Auflösung von 4 Metern. Das Bild ist auf 45% verkleinert worden.



Abb. 66: Segmentierung eines Weges im 4 Meter Bild um 300% vergrößert.

Nun wird noch überprüft, ob eine Veränderung der anderen beiden Parameter eine Verbesserung der Segmentform erzeugt. Bei der Variation der WEIGHT OF SHAPE

HOMOGENEITY und WEIGHT OF COMPACTNESS IN SHAPE HOM. Parameter sind die Veränderungen in den Segmenten recht unauffällig, so daß sich langes Experimentieren bei kleinem OBJECT SCALE PARAMETER nicht lohnt. Die bisher verwendeten Einstellungen führen auch hier zum besten Ergebnis. Die Segmente passen sich noch etwas besser der Gestalt der Felder an (Abb. 67). Eine zweite Segmentierungsebene ist hier nicht nötig, da die Segmente in vielen Fällen schon ein gesamtes Feld umfassen.



Abb. 67: Endgültige Segmentierung.



Abb. 68: Endgültige Segmentierung im Bild mit 10 Meter Auflösung in Originalgröße.

Bei dem Bild mit 10 Meter Auflösung muß der OBJECT SCALE PARAMETER noch etwas weiter verringert werden. Ein OBJECT SCALE PARAMETER von 40 liefert passende Segmente, die wiederum in vielen Fällen ganze Felder umfassen (Abb. 68). Wege werden hierbei allerdings nicht mehr erfaßt, da sie im Bild allenfalls ein Pixel breit sind. Es ist daher ratsam, diese Einzelheiten nicht einzubeziehen, indem der OBJECT SCALE PARAMETER nicht zu gering gewählt wird. Die anschließende Klassifikation würde durch diese kleinen Segmente sehr fehleranfällig und mühevoll werden. Auch das Ausprobieren weiterer Einstellungen der anderen beiden Parameter liefert keine Verbesserungen an den Segmenten, so daß man bei dieser niedrigen Auflösung bedenkenlos die voreingestellten Werte übernehmen kann.

## 4.3 Anwendung der Klassifikation

### 4.3.1 Allgemeine Vorgehensweise

Das Programm *eCognition* bietet bei der Klassifikation zahlreiche Möglichkeiten, so daß man brauchbare Methoden entwickeln muß, wie man auf dem schnellsten Weg zu akzeptablen Ergebnissen kommt. Eine naheliegende und auch sinnvolle Möglichkeit besteht darin, einzelne Klassen z.B. über ihre Grauwerte zu identifizieren. Dazu formuliert man in den Zugehörigkeitsfunktionen Beschreibungen wie: „Weizen ist alles, was im Kanal 1 einen Grauwert kleiner als 40 aufweist.“ Genauso beschreibt man dann die restlichen Klassen mit geeigneten Merkmalen. Als Hilfsmittel zur Suche nach geeigneten Merkmalen kann hierbei entweder die IMAGE OBJECT INFORMATION Dialogbox dienen, welche über einen rechten Mausklick auf ein Segment zu erreichen ist, oder man wählt unter VIEW/View Feature... ein gewünschtes Merkmal aus und überprüft, ob damit die Klassen zu trennen sind. Man wird schnell feststellen, daß dabei ein Merkmal nicht ausreicht, um eine bestimmte Klasse von den übrigen abzugrenzen. Also nimmt man weitere Merkmale zur Klassenbeschreibung hinzu und versucht so, zu einer vollständigen Klassifikation zu gelangen.

Da diese Vorgehensweise sehr mühsam sein kann, gibt es auch eine einfachere Möglichkeit, ohne großen Aufwand befriedigende Resultate zu erreichen. Mit einer Klassifikation mittels Nearest Neighbour gelangt man schnell zu überzeugenden Ergebnissen. Nearest Neighbour bietet den Vorteil, daß man durch das Auswählen von geeigneten Beispielsegmenten, die den Trainingsgebieten herkömmlicher Methoden entsprechen, sogleich viele Merkmale einbeziehen kann. Das gesamte Bild wird klassifiziert, so daß keine Segmente unklassifiziert bleiben, wobei sogar die Korrelationen zwischen den einzelnen Merkmalen berücksichtigt werden (Baatz 1999). Im Anschluß braucht man dann lediglich Fehlklassifikationen durch weitere Merkmale mit Hilfe von Zugehörigkeitsfunktionen zu beseitigen. Dieser Vorgang kann dagegen für den ungeübten Anwender noch verhältnismäßig zeitaufwendig werden, da er sich die Frage stellen muß, warum das Programm an dieser Stelle falsch klassifiziert hat, der Mensch aber das richtige Ergebnis kennt. Diese menschliche

Erkenntnis versucht man im Programm durch Zugehörigkeitsfunktionen auszudrücken. Diese Möglichkeit zur Verbesserung der Klassifikation bietet sonst sicherlich kein anderes Programm. Hat man sich einmal die Arbeit gemacht, eine adäquate Klassifikation zu erzeugen, so kann man die Klassenbeschreibungen exportieren und in anderem Bildmaterial weiterverwenden.

### 4.3.2 Klassifikation des Testgebietes

Auch bei der Klassifikation kann es hilfreich sein, zuerst kleine Ausschnitte zu bearbeiten, weil die Handhabung des Rechners dann reibungsloser abläuft. Wenn dagegen ein schneller Computer mit sehr viel Arbeitsspeicher zur Verfügung steht, kann die Klassifikation auch gleich direkt an den großen Bildern durchgeführt werden, da nur die Segmentierung die Ressourcen des Rechners stark beansprucht. In diesem Beispiel wird das obere linke Viertel des Gesamtbildes zuerst klassifiziert. Der Rechner kann mit diesem ca. 10 MByte großem Material noch gut umgehen. Zu Beginn ist es ratsam, die gewünschten Klassen festzulegen, wobei es aber auch kein Problem ist, zusätzliche Klassen später hinzuzufügen. Für das gesamte Testgebiet werden nun folgende Klassen mit zugehöriger Farbe festgelegt:

Weizen		Hafer		Gewässer	
Rüben		Kartoffeln		Laubwald	
Mais		Grünland		Nadelwald	
Gerste		Wege		Bebautes Gebiet	
Triticale		Straßen			
Raps		Eisenbahn			

Abb. 69: Farblegende über die verwendeten Klassen.<sup>6</sup>

Mit diesen 16 Klassen werden sowohl alle in diesem Gebiet angebauten Feldfrüchte als auch Wald und Gewässer und Bauwerke erfaßt. Bevor die eigentliche

<sup>6</sup> Bei Triticale handelt es sich um ein Gemisch aus Weizen und Roggen, das hauptsächlich als Futtermittel eingesetzt wird. Die Klasse Kartoffeln bleibt inaktiv, da sie im Bild von Rüben nicht zu unterscheiden ist. Im gesamten Gebiet sind außerdem nur zwei kleine Kartoffelfelder vorhanden.



Klassifikation beginnt, werden die Klassen in der CLASS HIERACHY angelegt (Abb. 70). Außerdem können auch schon Strukturgruppen angelegt werden, die zusammengehörige Klassen miteinander verknüpfen. So ergeben sich die drei Strukturgruppen: Felder, Wald und Sonstiges (Abb. 71).

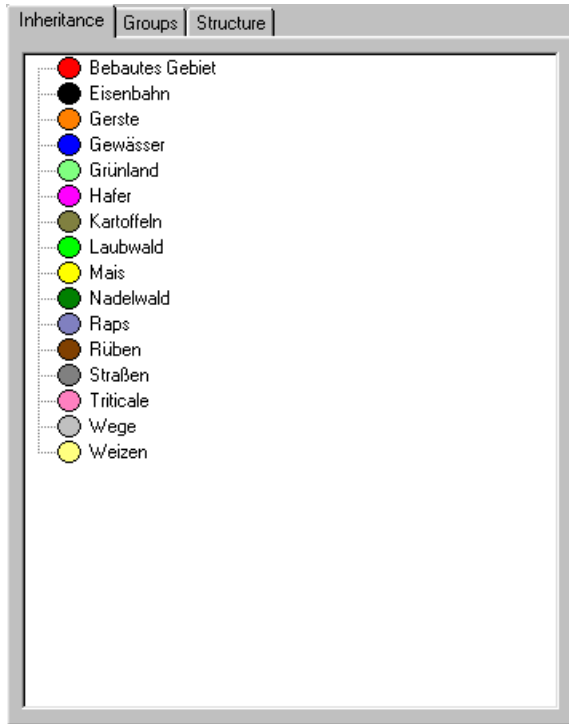


Abb. 70: Alle 16 Gruppen in der Klassenhierarchie.

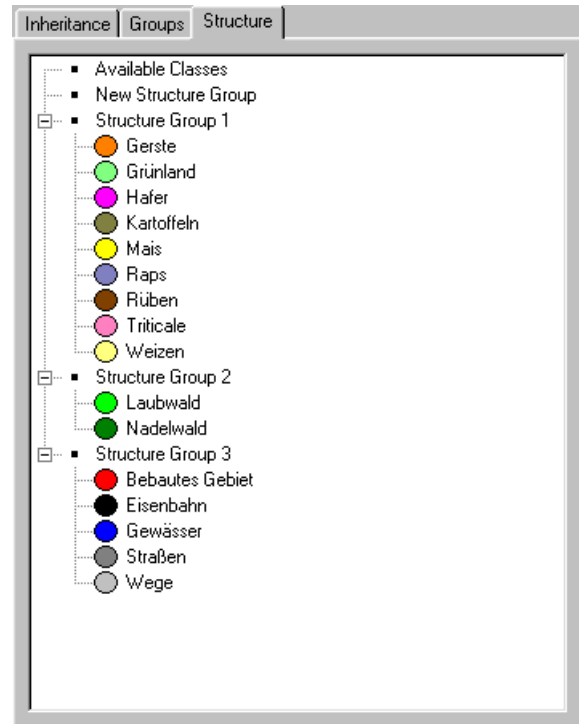


Abb. 71: Zusammenfassen zu drei Strukturgruppen.

Zur Sicherheit und späteren Wiederverwendung wird die Klassenhierarchie abgespeichert. Als weitere vorbereitende Maßnahme werden die zu verwendenden Merkmale ausgewählt. Dazu werden unter FEATURES/Select active Features... die bereits vorhandenen Merkmalen um Brightness und Channel Ratio ergänzt, da diese bei der Nearest Neighbour Klassifikation zweckdienlich eingesetzt werden können. In der Dialogbox EDIT SAMPLE FEATURE SPACE werden die bei der Nearest Neighbour Klassifikation ausgewählten Merkmale aktiviert. In diesem Beispiel werden zunächst nur alle Farbmerkmale ausgewählt (Abb. 72). Um nicht für alle Klassen jedes einzelne Merkmal separat einfügen zu müssen, wird dies in einem Arbeitsgang in der EDIT NEXT NEIGHBOUR CLASSIFICATION... Dialogbox erledigt. Im EDIT CLASS DESCRIPTION Fenster erscheint die Nearest Neighbour Klassifikation nun in allen Klassen mit dem Zusatz (**generated**). Jetzt müssen nur noch Samples für die einzelnen Klassen ausgesucht werden. Mit dem Öffnen des SAMPLE EDITORS wird

das Programm in den Sample Input Modus versetzt, und die Eingabe kann beginnen. Es wird jeweils für die in der linken Liste angezeigte Klasse ein Beispielgebiet definiert, indem man das gewünschte Segment mit der Maus doppelt anklickt. In diesem Viertel sind nicht alle Objektklassen zu finden, so daß nicht für jede Klasse ein Beispielsegment definiert werden kann. Bei der Auswahl der Beispielsegmente ist darauf zu achten, daß zunächst nur typische Beispiele einer Klasse ausgesucht werden, damit bei der Klassifikation möglichst alle Vertreter einer Klasse herausgefunden werden. Die Beispiele werden außerdem nur auf Ebene 1, d.h. auf der niedrig aufgelösten Ebene definiert. Das Ergebnis einer ersten Klassifikation ist schon recht gut, obwohl dennoch einige Fehlklassifikationen vorliegen (Abb. 73).<sup>7</sup>

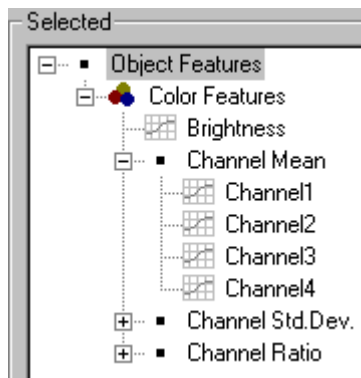


Abb. 72: Bei Nearest Neighbour verwendete Merkmale.



Abb. 73: Erste Klassifikation mit Nearest Neighbour auf Ebene 1.

Wie man sieht, gibt es innerhalb der Felder teilweise kleine Segmente, die falsch klassifiziert wurden. Durch das Zusammenfassen der kleinen Segmente auf einer niedriger aufgelösten Ebene ist die Anzahl solcher Segmente sehr gering. Wenn man

<sup>7</sup> Die hier gezeigten klassifizierten Bilder sind auf 25% verkleinert worden. In einer Größe von 40% sind die Ergebnisse aller vier Bilder im Anhang (7.3.1) zu finden.

die hochaufgelöste Ebene 2 betrachtet, so ist diese in der Detailauflösung zwar genauer, aber der Anteil der Fehlklassifizierungen ist deutlich höher (Abb. 74).



Abb. 74: Erste Klassifikation mit Nearest Neighbour auf Ebene 2.

Als erste Verbesserung bietet es sich immer an, im Sample Editor einige dieser falsch klassifizierten Segmente als Beispiele der richtigen Klasse zu definieren. Dabei sollten nur größere Segmente verwendet werden, da sie charakteristische Vertreter der jeweiligen Klasse darstellen. Die sehr kleinen fehlklassifizierten Segmente innerhalb von Feldern werden durch Bodenveränderungen oder Fahrspuren verursacht und sind somit keine repräsentativen Beispiele für diese Klasse. Einem Menschen ist z.B. klar, daß die kleinen Einschlüsse im Triticalefeld ebenfalls zum Feld dazugehören, weil der Einschuß Bestandteil des Feldes ist und nur von Triticale umgeben ist. Die Lösung ist die Integration dieser Beziehung in die Klassenbeschreibung. Dazu öffnet man das EDIT CLASS DESCRIPTION Fenster und fügt das Merkmal „Rel. Border to Triticale“ ein, und die Zugehörigkeitsfunktion wird auf einen Wert von größer als 90% gesetzt. Weiterhin wird die Verknüpfung zwischen „Nearest Neighbour“ und „Rel. Border to Triticale“ von „and“ auf „or“ gesetzt (Abb. 75). Folglich ergibt sich die folgende Beziehung: „Triticale ist alles, was

dem Ergebnis der Nearest Neighbour Klassifikation entspricht, oder was zu mehr als 90% von Triticale umgeben ist.“

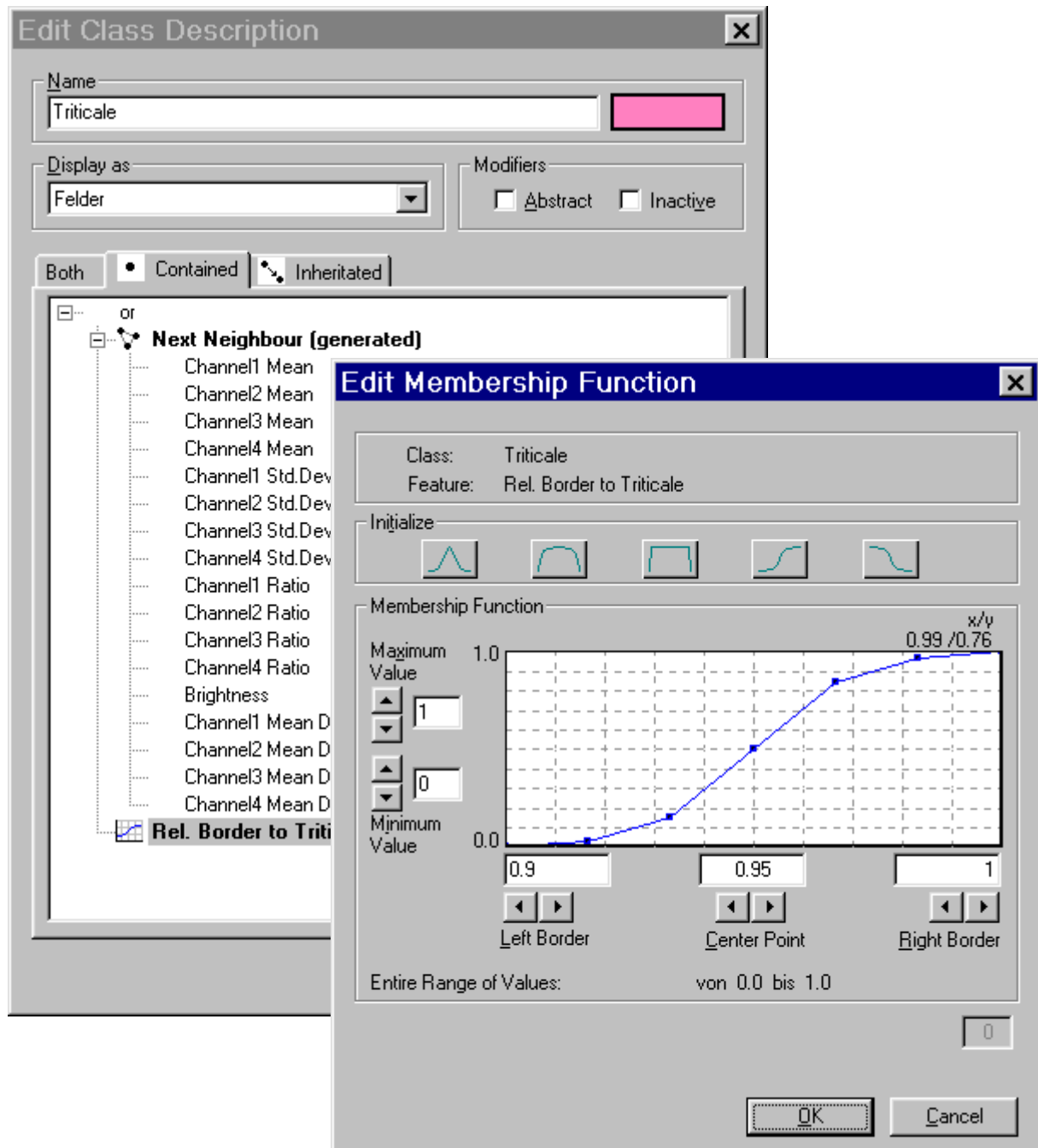


Abb. 75: Klassenbeschreibung für die Klasse Triticale mit Zugehörigkeitsfunktion.

Im CLASSIFICATION Fenster muß jetzt ENABLE CONTEXT CLASSIFICATION aktiviert sein, da es sich bei „Rel. Border to Triticale“ um ein Knowledge based Feature handelt. Weiterhin wird das Merkmal „Rel. Border to“ in entsprechender Weise in die

Klassen Weizen und Wald eingebunden, so daß der Anteil der Fehlklassifizierungen deutlich vermindert wird (Abb. 76).



Abb. 76: Verbesserte Klassifikation auf Ebene 1.

Auf der Ebene 2 sind dagegen kaum Verbesserungen zu sehen (Abb. 77), da viele Einschlüsse aus mehr als einem Segment bestehen, und die relative Grenze zu der richtigen Klasse daher weniger als 90% ausmacht. Um hier eine Verbesserung zu erreichen, kann man versuchen, die Superobjekte zu Hilfe zu nehmen. In der Rubrik „Relations to Super-Objects“ befindet sich die Beziehung „Existance of“, welche ideal dafür geeignet ist. In der Zugehörigkeitsfunktion gibt man einen Wert von größer als 50% an und startet die Klassifikation erneut. Diesmal wird aber nur auf Ebene 2 klassifiziert, da für Ebene 1 keine Superobjekte vorhanden sind, und die NUMBER OF CYCLES wird auf zwei eingestellt. Als Ergebnis verschwinden die vielen Einschlüsse im Wald, so daß man bei den anderen Klassen ebenso verfahren kann.

Nun hat es sich als wenig sinnvoll erwiesen, noch weitere Ausdrücke aufzustellen, da die Beziehungen zwischen den einzelnen Klassenbeschreibungen für den Auswerter zu unübersichtlich werden und dann zu unerwünschten Ergebnissen führen. Das

einzig Störende am Ergebnis sind die als Rüben vereinzelt klassifizierte Segmente, die über das gesamte Bild verteilt sind.



Abb. 77: Verbesserte Klassifikation auf Ebene 2.

Diese verschwinden, wenn man die Beziehung „Rel. Border to Rüben“ auf den Wert 20% verringert und die Verknüpfung zur Nearest Neighbour Klassifikation auf „and“ setzt. Dies bedeutet, daß ein Segment nur als Rüben klassifiziert wird, wenn es einerseits die Nearest Neighbour Klassifikation erfüllt und andererseits eine Mindestgrenze von 20% zu sich selbst aufweist. Vereinzelt als Rüben klassifizierte Segmente besitzen demnach keine gemeinsame Grenze mit anderen Rübensegmenten und werden somit ausgeschlossen. Für Triticale wird die gleiche Beziehung aufgestellt. Ein weiteres Problem sind die Randbereiche der Felder, die oft keine eindeutige Klassensignatur aufweisen. Dabei sollte man sich damit zufrieden geben, wenn diese Bereiche als Weg klassifiziert werden, da die Bepflanzung in diesem Bereich ohnehin kaum oder gar nicht vorhanden ist. Falls dies nicht gewünscht ist, so könnte man diese Bereiche noch mit einer weiteren Klasse wie z.B. „Erosion“ oder „Feldrand“ erfassen. Die endgültige Klassifikation ergibt sich nun folgendermaßen (Abb. 78):



Abb. 78: Endgültige Klassifikation auf Ebene 2.

Als Endergebnis sollte man die Ebene 2 verwenden, da die Detailauflösung wegen der kleineren Segmente ein bißchen genauer ist.<sup>8</sup> Eine Verbesserung der Detailauflösung läßt sich aber auch noch auf der Ebene 1 erzielen, indem man eine Border Optimization durchführt (3.6.2). Als letztes bietet es sich außerdem an, in der Klassenhierarchie unter GROUPS die einzelnen Klassen zu Felder, Wald und Sonstige zusammenzufassen. Bei einer anschließenden Klassifikation wird eine zweite Klassifikationsebene angelegt, in der nur diese drei Klassen zu sehen sind (Abb. 79). Einen ähnlichen Effekt erzielt man mit den Strukturgruppen. Führt man eine Knowledge based Segmentation durch, so werden die Segmente gemäß der Strukturgruppen zusammengefaßt. Die anschließende Klassifikation weist dann dagegen einzelne Fehler auf, die auf der Klassenbeschreibung beruhen. Es ist aber unnötig, diese noch anzupassen, denn der Zeitaufwand wäre zu hoch, nur um das gleiche Ergebnis wie bei den GROUPS zu erzielen.

---

<sup>8</sup> Die zwei unregelmäßig klassifizierte Felder rechts unterhalb der Mitte bestehen aus nicht weiter bestimmbar Getreide. Beim Amt für Agrarstruktur sind diese beiden Flächen als „sonstiges Getreide“ verzeichnet.



Abb. 79: Zusammengefaßte Klassen.

### 4.3.3 Klassifikation mit panchromatischen Bildern

Die Klassifikation von panchromatischen Bildern liefert mit Sicherheit nicht die gleichen Ergebnisse wie multispektrale Bilder, da Farben auch in *eCognition* eine bedeutende Rolle spielen. Klassifiziert man das Beispielgebiet mit denselben Merkmalen wie das multispektrale Bild, so ist das Resultat trotz geeigneter Samples nicht ausreichend (Abb. 80). Bei diesem Bild wurde für Nearest Neighbour nur der Grauwert und seine Standardabweichung benutzt. Eine Kontextklassifikation verschlechtert das Bild noch weiter, weil die Ausgangslage ungenügend ist. Auch wenn für Nearest Neighbour weitere Merkmale genutzt werden (Abb. 81), verbessert sich das Resultat nicht (Abb. 82). Betrachtet man die einzelnen Merkmale unter VIEW/View Feature... , so findet man zur Unterscheidung der einzelnen Klassen kein geeignetes, da die Segmente innerhalb einer Klasse recht unterschiedlich sein können. Lediglich „Line SO: length/width“ läßt einige Wege und Straßen deutlich hervortreten (Abb. 83).





Abb. 80: Klassifikation des panchromatischen Bildes.

- Channel1 Mean
- Channel1 Std.Dev.
- Area
- Line SO: length/width
- Line SO: width
- Mean of Size of Sub-Objects
- Size Std.Dev. of Sub-Objects
- Channel1 Std.Dev. of Sub-Objects

Abb. 81: Bei Nearest Neighbour zusätzlich verwendete Merkmale.



Abb. 82: Klassifikation des panchromatischen Bildes mit zusätzlichen Merkmalen auf 15% verkleinert.

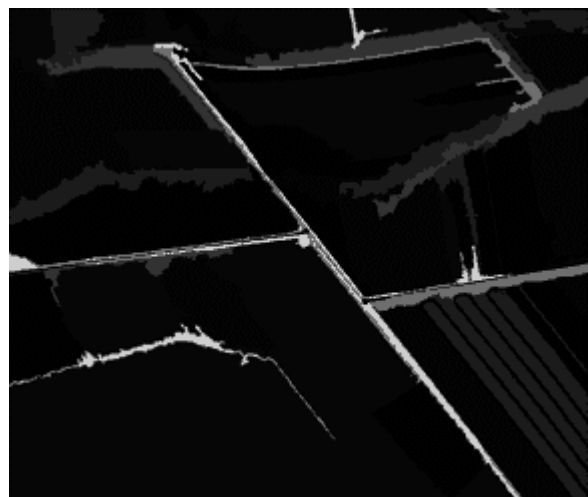


Abb. 83: Das Merkmal „Line SO: length/width“.

Für eine Unterscheidung aller Klassen ist ein panchromatisches Bild also mit Einschränkungen geeignet. Wenn man sich allerdings auf die Gruppen Wald, Feld und Sonstiges beschränkt, so ist das Ergebnis durchaus positiv (Abb. 84). Es hat

zwar nicht die Genauigkeit einer multispektralen Klassifikation (Abb. 79), aber die Genauigkeit ist nicht wesentlich geringer.



Abb. 84: Zusammengefaßte Gruppen.

### 4.3.4 Klassifikation mit Texturkanal

In diesem Kapitel wird untersucht, ob die Einbeziehung eines Texturkanals die Klassifikation verbessern kann. In diesem Fall werden zusätzlich zu den vier DPA-Kanälen die mit *TNTmips* berechneten Texturkanäle verwendet. Da sich die Texturkanäle der ersten drei DPA-Kanäle sehr ähnlich sehen, kann man sich auf einen ausgewählten beschränken. DPA-Kanal vier liefert jedoch zusätzliche Informationen und wird deshalb auch genutzt, so daß insgesamt sechs Kanäle zur Verfügung stehen. Die Segmentierung dauert trotz der zwei zusätzlichen Bilder nur unwesentlich länger als sonst, erhöht die Genauigkeit aber nicht (4.2.4), wobei dieselben Parametereinstellungen verwendet werden. Danach wird eine Nearest Neighbour Klassifikation mit geeigneten Beispielen durchgeführt. Das Resultat ist

allerdings auch nicht passender als ohne die Texturkanäle (Abb. 85). Nur Wald wird in diesem Beispiel sicherer erkannt, da diese Klasse im Texturkanal 4 gegenüber den anderen Klassen sehr hell abgebildet wird. Die fehlklassifizierten Segmente lassen sich wieder mit den schon bekannten Merkmalen vermeiden (4.3.2). Dadurch daß die Ausgangssituation der Nearest Neighbour Klassifikation nicht besser geworden ist, erübrigen sich die Texturkanäle, denn die wissensbasierten Merkmale verwenden eine schon existierende Klassifikation.



Abb. 85: Nearest Neighbour Klassifikation mit Texturkanälen.

### 4.3.5 Texturanalyse mit *eCognition*

Mit einem Texturkanal wird Wald sicherer erkannt, weil dieser im Bild aufgrund seiner Textur sehr hell abgebildet wird. Aber auch in *eCognition* sind Texturmerkmale vorhanden, die sich hingegen mit einem Texturkanal nicht vergleichen lassen. In *eCognition* wird die Textur an Hand von Subobjekten einbezogen, d.h. deren Standardabweichungen der Grauwerte und der Fläche, sowie deren mittlere Größe.

Bevor diese Merkmale genutzt werden können, müssen daher erst einmal geeignete Subobjekte gebildet werden. Man könnte die bereits bestehende Ebene 2 dafür verwenden, aber einen spezialisierten Parameter findet man bei der Multiresolution Segmentation, indem man den SEGMENTATION MODE auf „Form Subobjects“ setzt. Dabei wird nur das Formkriterium (4.2.1) berücksichtigt, und es werden möglichst kompakte Objekte gebildet (Flächenkriterium). Der OBJECT SCALE PARAMETER ist dann auf den Wertebereich [0...1] beschränkt und bestimmt die maximale relative Kantenlänge der Subobjekte zu Nachbarn, die nicht Bestandteil desselben Superobjektes sind. Die neue Ebene wird unter Ebene 1 angelegt, da die Subobjekte kleiner sind, doch vorher wird noch Ebene 2 gelöscht, da diese sonst mit ihren vorgegebenen Grenzen die Segmentierung einschränken würde. Als OBJECT SCALE PARAMETER wird 0.5 angegeben. Die erhaltenen Subobjekte unterscheiden sich zum Teil sehr stark voneinander (Abb. 86), so daß sie wahrscheinlich dienlich eingesetzt werden können.

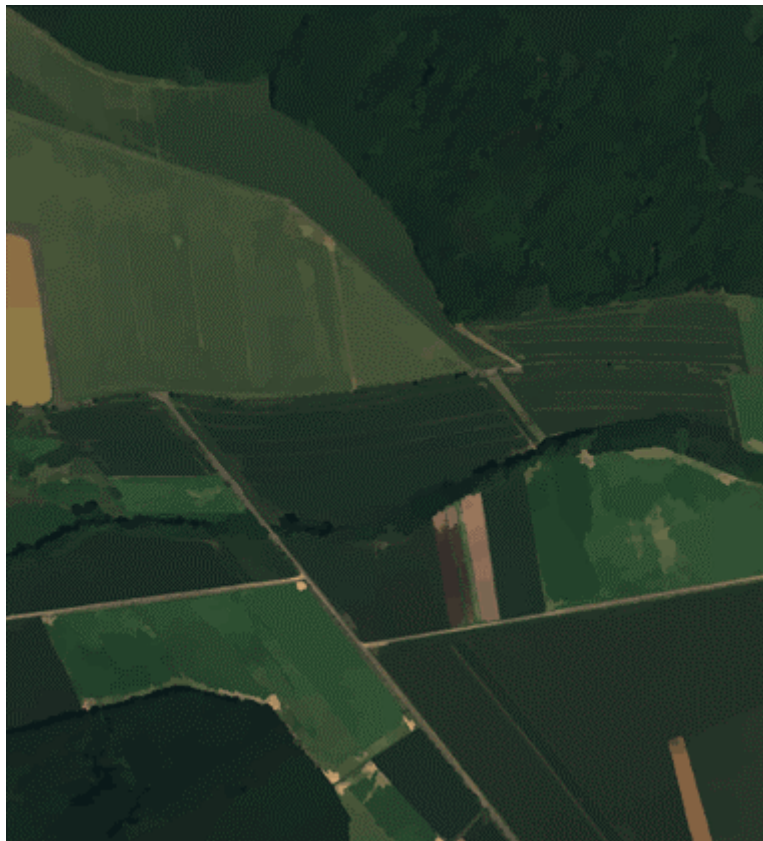


Abb. 86: Segmentierung mit „Form Subobjects“.

Integriert man diese Merkmale aber in die Nearest Neighbour Klassifikation, so verschlechtert sich das Klassifikationsergebnis stark (Abb. 87). Außerdem ist aus

ungeklärten Gründen unten rechts ein Segment unklassifiziert geblieben, sobald ein auf Subobjekten basierendes Merkmal in die Nearest Neighbour Klassifikation eingefügt wird.



Abb. 87: Nearest Neighbour Klassifikation mit Texturanalyse

Man braucht sich aber auch nicht die Mühe zu machen, ausgewählte Merkmale als zusätzliche Klassenbeschreibung in Form von Zugehörigkeitsfunktionen zu entwickeln. Da die drei Texturparameter innerhalb einer Klasse stark schwanken (Abb. 88), ist somit eine Abgrenzung der Klassen nicht möglich. Höchstens vereinzelt läßt sich hier ein Merkmal nutzen. In diesem Bild ist z.B. die Standardabweichung der Subobjekte der Klasse Wald etwas höher als bei den restlichen Klassen. Eine Verbesserung der Klassifikation ist damit aber nicht zu erreichen, so daß man auch darauf verzichten kann. Auch eine Veränderung des OBJECT SCALE PARAMETERS auf 0.1 und 0.9 verändert die Segmente nicht wesentlich. Eine relative Kantenlänge von 90% läßt erwartungsgemäß kaum neue Segmente entstehen, und eine relative Kantenlänge von 10% erzeugt etwas kleinere Segmente als mit 50%, die aber die Klassifikation nicht bereichern können.



Abb. 88: Die Standardabweichung der Subobjekte im Kanal 4 als Grauwerte dargestellt.

### 4.3.6 Klassifikation von Bildern anderer Auflösungsstufen

In diesem Kapitel wird nun eine Klassifikation des Testgebietes in den Auflösungen von 4 Metern und 10 Metern untersucht. Nach der Segmentierung des 4 Meter Bildes (4.2.5) wird zuerst wiederum eine Nearest Neighbour Klassifikation durchgeführt. Obwohl für die meisten Klassen nur ein Sample definiert wurde, ist das Ergebnis beinahe fehlerfrei. Obwohl im bebautem Bereich die Detailgenauigkeit erwartungsgemäß nicht sehr hoch ist, liegt der prozentuale Anteil der Grünflächen und Straßen schätzungsweise richtig. In Ausnahmefällen sind innerhalb von Feldern kleine Segmente falsch klassifiziert worden, die auf Fahrspuren, Erosion oder einem Überschwemmungsgebiet beruhen. Partiiell oder gänzlich falsch klassifizierte Felder sind rot umrandet (Abb. 89). Die sehr feinen Verkehrsnetzstrukturen sind dagegen sehr gut im Rahmen der auflösungsbedingten Genauigkeit erfaßt worden.



Abb. 89: Nearest Neighbour Klassifikation mit 4 Metern Auflösung auf 50% verkleinert.

Durch systematische Zuweisung von fehlklassifizierten Segmenten als Beispiele der richtigen Klasse wird die Klassifikation einwandfrei. Wenn man die restlichen kleinen Fehler akzeptiert, hat man ohne großen Arbeitsaufwand ein verwendbares Ergebnis erreicht (Abb. 90).



Abb. 90: Endgültige Klassifikation.

Eine weitere Verbesserung mit den anderen verfügbaren Merkmalen wäre dagegen zeitaufwendiger und nicht unbedingt notwendig. In einer Auflösung von 10 Metern sind die Ergebnisse mit geeigneten Samples grundsätzlich genauso gut (Abb. 91), nur die Detailgenauigkeit ist natürlich geringer. Einige Straßen und die Eisenbahnlinie sind trotz einer Breite von 1 bis 2 Pixeln erfaßt worden. Die Detailauflösung des bebauten Bereiches ist hingegen kaum passend, und Grünflächen sind dort überhaupt nicht zu sehen. Wenn hier eine höhere Genauigkeit gefordert ist, muß man sehr kleine Segmente verwenden, die dann allerdings die Klassifikation im allgemeinen erschweren.

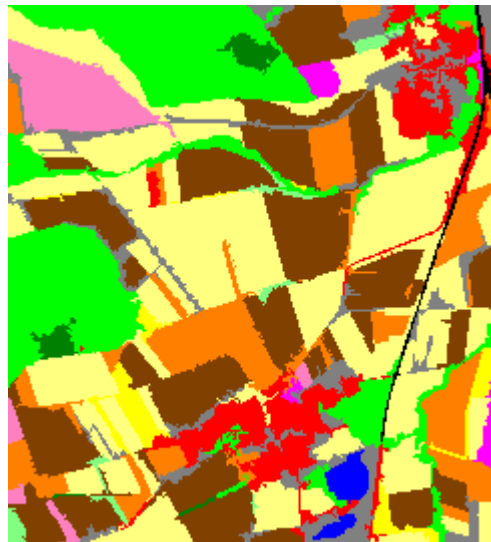


Abb. 91: Klassifikation im 10 Meter Bild.

### 4.4 Übertragung der Klassenbeschreibungen auf andere Bilder

Um wirtschaftlich große Datenmengen zu bearbeiten, ist es sinnvoll, einmalig eine umfassende Klassenbeschreibung zu entwickeln und diese dann auch auf folgende Bilder anzuwenden. Dazu muß man zu Beginn in einem Referenzbild, welches alle gewünschten Klassen enthält, stabile Klassenbeschreibungen entwickeln. Danach wird unter CLASSIFICATION/Save Class Hierarchy... die komplette Klassenbeschreibung abgespeichert. Im Anschluß kann das nächste Bild geladen und segmentiert werden, wobei dieselben Parametereinstellungen wie im Referenzbild verwendet werden sollten, um homogene Ergebnisse von allen Bildern zu erhalten.



Außerdem sind die zu erwartenden Ergebnisse der Klassenbeschreibungen besser abzuschätzen, wenn sie sich auf die gleiche Art von Segmenten beziehen. Deswegen sollten auch die Anzahl und Reihenfolge der Kanäle in allen Bildern generell übereinstimmen. Als nächstes wird die vorher abgespeicherte Klassenhierarchie mit CLASSIFICATION/Load Class Hierarchy... geladen und die Klassifikation kann beginnen.

Die Entwicklung der Klassenbeschreibungen ist allerdings in diesem Fall nicht so einfach wie bei einem Einzelbild (4.3.2). Dort wurde die Klassifikation im wesentlichen durch das geschickte Hinzufügen von immer neuen Samples verbessert. Wenn die Klassenhierarchie dagegen in vielen Bildern genutzt werden soll, muß sie so allgemein wie möglich gehalten werden.

- Channel1 Mean
- Channel2 Mean
- Channel3 Mean
- Channel4 Mean
- Channel1 Std.Dev.
- Channel2 Std.Dev.
- Channel3 Std.Dev.
- Channel4 Std.Dev.
- Channel1 Ratio
- Channel2 Ratio
- Channel3 Ratio
- Channel4 Ratio
- Brightness
- Channel1 Mean Diff. to entire Scene
- Channel2 Mean Diff. to entire Scene
- Channel3 Mean Diff. to entire Scene
- Channel4 Mean Diff. to entire Scene

Abb. 92: Merkmale bei der Nearest Neighbour Klassifikation.



Abb. 93: Nearest Neighbour Klassifikation.

Auch hier werden die Klassen zuerst durch wenige charakteristische Samples für die Nearest Neighbour Klassifikation beschrieben. Dabei werden nur die wichtigsten Merkmale verwendet (Abb. 92), welche die einzelnen Klassen signifikant voneinander trennen können. Die dabei erhaltenen Werte der Merkmale können in

anderen Bildern weiterverwendet werden und benötigen keinen Bezug zum ursprünglichen Segment. Die Samples müssen in diesem Fall in mehreren Bildern gesammelt werden, da nicht jede Klasse in allen Bildern vorkommt. Das Resultat dieser Klassifikation ist zwar grundsätzlich richtig aber dennoch verbesserungswürdig, da wie immer einzelne Segmente innerhalb von Feldern falsch klassifiziert wurden (Abb. 93).

Zur Beseitigung der einzelnen fehlklassifizierten Segmente innerhalb von Feldern oder Wald werden nun wissensbasierte Merkmale eingefügt. Mit „Rel. Area of“ und „Rel. Border to“ formuliert man z.B. die Bedingung, daß Segmente, in deren Nähe viel Gerste vorkommt oder die eine hohe Kantenlänge zu Gerste aufweisen, mit hoher Wahrscheinlichkeit wieder Gerste sind. Um dabei keine neuen Fehlklassifizierungen zu erzeugen, werden hohe Werte vorausgesetzt, d.h. die relative Kantenlänge bzw. die relative Fläche muß mindestens mehr als 90% betragen (Abb. 94).

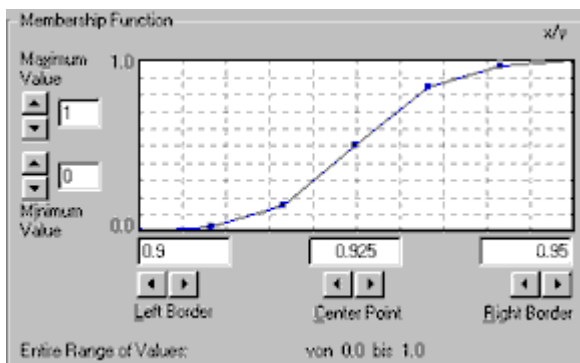


Abb. 94: „Rel. Border to Gerste“.

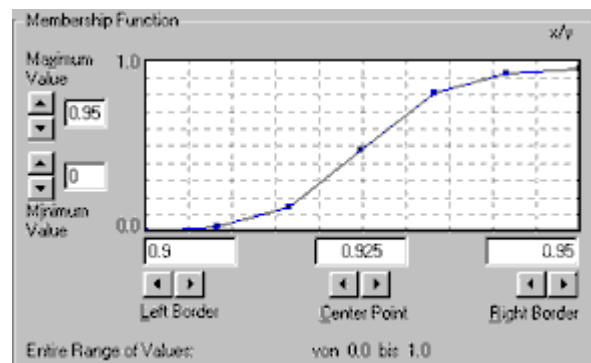


Abb. 95: „Rel. Area of Gerste“.

Weiterhin soll die relative Fläche im Höchstfall nur eine 95%ige Wahrscheinlichkeit zurückliefern (Abb. 95). Dies liegt darin begründet, daß ein kleines Gerstefeld, welches aus nur einem Segment besteht und von Rübenfeldern umgeben ist, eine hohe relative Fläche von Rüben aufweist und die Zugehörigkeitsfunktion somit einen hohen Wert zurückliefert. Obwohl das betroffene Segment aus der Nearest Neighbour Klassifikation eine 96%ige Klassenzugehörigkeit aufweist, würde es durch die „Rel. Area of“-Funktion mit einem Wert von 100% übertroffen werden. Indem die Funktion aber als Höchstwert nur 95% zurückliefern kann, wird die Fehlklassifikation

vermieden. Diese beiden Beziehungen werden in alle Klassen in entsprechender Weise eingefügt.

Es lassen sich aber noch weitere wissensbasierte Merkmale einfügen. Da Felder nicht innerhalb von bebautem Gebiet zu finden sind, wird die Bedingung gestellt: „Ein Segment ist kein Feld, wenn es ausschließlich von Häusern oder Straßen umgeben ist.“ Dazu werden die einzelnen Feldklassen zu der Oberklasse Feld im Bereich INHERITANCE gruppiert, und in die nun abstrakte Klasse Feld wird jenes Merkmal eingefügt und somit auf alle enthaltenen Feldklassen vererbt (Abb. 96 und Abb. 97).

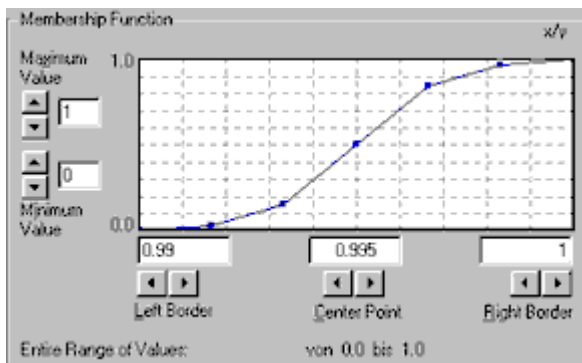


Abb. 96: Zugehörigkeitsfunktion „Rel.Border to Sonstige“.

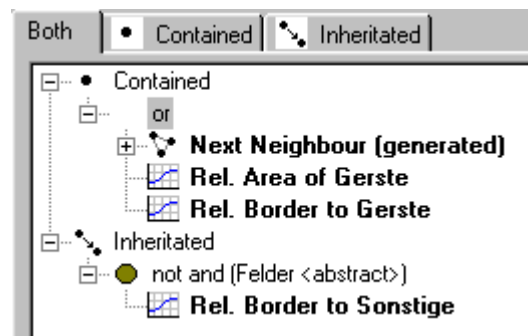


Abb. 97: Klassenbeschreibung von Gerste.

Eine vergleichbares Merkmal wird in die Klasse Bebautes Gebiet eingefügt, die nicht innerhalb eines Feldes oder Waldes zu finden ist (Abb. 98 und Abb. 99). Dazu werden die Klassen Feld und Wald in GROUPS zu der Klasse Land gruppiert (Abb. 100).

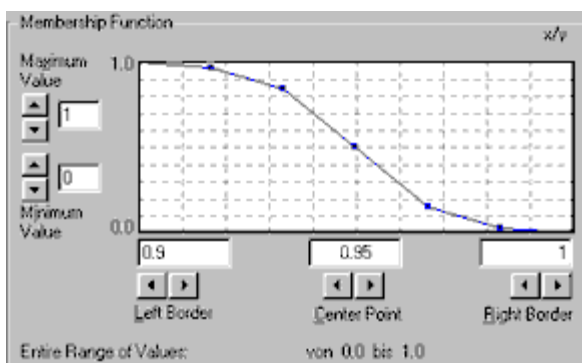


Abb. 98: Zugehörigkeitsfunktion „Rel.Border to Land“.

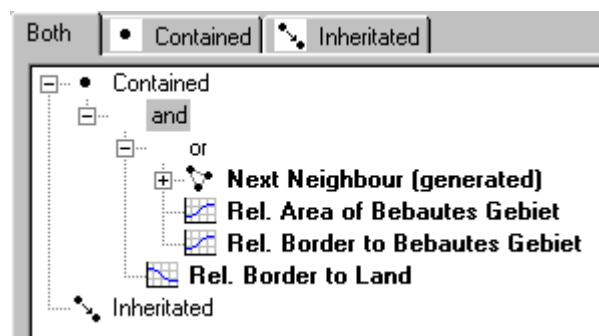


Abb. 99: Klassenbeschreibung von Bebautes Gebiet.

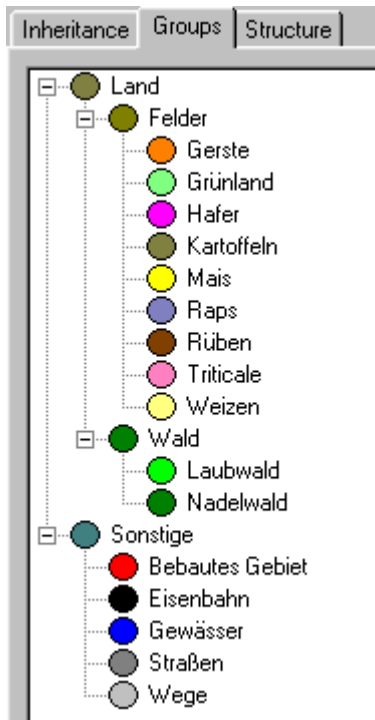


Abb. 100: Klassenhierarchie.

Feature	Sngl	Cntx
Evaluation of Class Bebautes Gebiet	81.3	28.7
-----		
and	81.3	28.7
or	81.3	81.3
Next Neighbour (generated)	81.3	81.3
Rel. Area of Bebautes Gebiet : 0.22	---	0.0
Rel. Border to Bebautes Gebiet : 0.04	---	0.0
Rel. Border to Land : 0.96	---	28.7

Abb. 101: IMAGE OBJECT INFORMATION Fenster.



Abb. 102: Endgültige Klassifikation.

Die Verknüpfung der einzelnen Merkmale mit den logischen Operatoren erweist sich dabei als große Schwierigkeit, und man erhält beim ersten Versuch nicht immer das gewünschte Ergebnis. Als gut Hilfe dient dabei das IMAGE OBJECT INFORMATION Fenster, in dem man die Berechnung der endgültigen Zugehörigkeitswahrscheinlichkeit verfolgen kann (Abb. 101). Das Endergebnis ist von den meisten Fehlern nun bereinigt (Abb. 102) und die anderen drei Bilder können klassifiziert werden.<sup>9</sup>

<sup>9</sup> Die Ergebnisse aller vier Bilder auf 40% verkleinert befinden sich im Anhang (7.3.2).

Selbst auf die Bilder anderer Auflösung kann man die Klassenbeschreibung anwenden. Die Verwendung des „Rel. Area of“ Merkmals sollte aber noch etwas abgeschwächt werden, da ansonsten eventuell einige Segmente fehlerklassifiziert werden. Im Bild mit 10 Meter Auflösung kann dieses Merkmal sogar gänzlich fortgelassen werden (Abb. 103 bis Abb. 106).



Abb. 103: Nearest Neighbour Klassifikation im Bild mit 4 Meter Auflösung auf 50% verkleinert.



Abb. 104: Kontextklassifikation im Bild mit 4 Meter Auflösung.

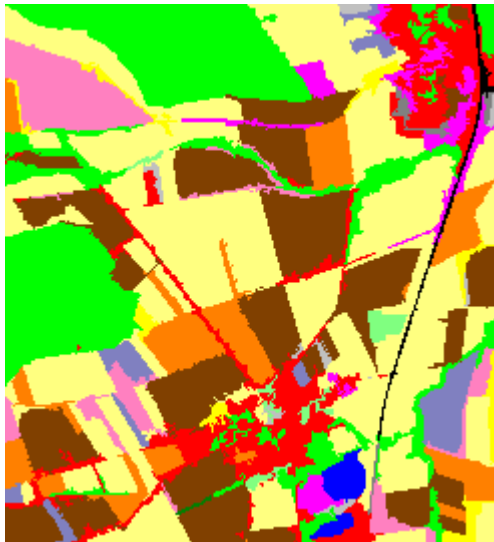


Abb. 105: Nearest Neighbour Klassifikation im Bild mit 10 Meter Auflösung.



Abb. 106: Kontextklassifikation im Bild mit 10 Meter Auflösung.

#### 4.5 Vergleich mit anderen Klassifikationsverfahren

In der Diplomarbeit von Jessica Wilke (Hannover, 1999) wurde dasselbe Gebiet mit den Methoden Maximum Likelihood (2.3.4), Back Propagation (2.3.5) und Fuzzy C Means<sup>10</sup> klassifiziert.









Rüben		Weizen	
Raps		Mischwald	
Mais		Straßen	
Gerste		Wege	

Abb. 107: Farblegende über die verwendeten Klassen.

Das Ergebnis der Maximum Likelihood Klassifikation ist relativ genau, und die Berechnungsdauer beträgt nur ca. ½ Stunde. Innerhalb der Felder oder des Waldes sind kleine Einschlüsse zu sehen, die auf lokale Unregelmäßigkeiten wie z.B. Fahrspuren beruhen.

<sup>10</sup> Bei dieser unüberwachten Klassifikationsmethode kann ein Pixel mit Hilfe von Fuzzy Logic Mitglied mehrerer Klassen sein. Der Algorithmus berechnet in jeder Iteration die Distanz zu jeden Klassenmittelpunkt, die mit dem aktuellen fuzzy Zugehörigkeitskoeffizienten gewichtet wird. Die neuen Klassenmittelpunkte werden gefunden, indem die Summe über alle Pixel dieser gewichteten Distanzen minimiert wird (TNTmips 1998).



Abb. 108: Maximum Likelihood Klassifikation aus 25% verkleinert.

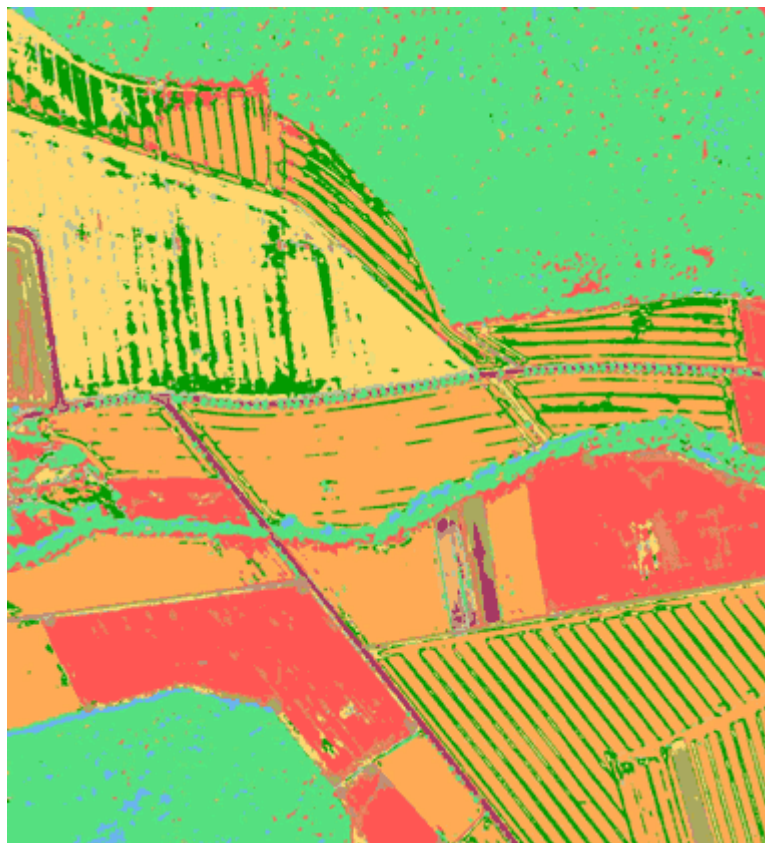


Abb. 109: Klassifikation mit Back Propagation.

Die Ergebnisse mit dem Back Propagation Algorithmus und den Fuzzy Sets sind weniger passend, denn viele Bodenveränderungen wurden der falschen Klasse zugeordnet. Außerdem liegen die Berechnungszeiten je nach Anzahl der Iterationen bei einigen Stunden bis zu einem Tag. Die Klassifikationsgenauigkeit dieser drei Methoden entspricht ungefähr der Nearest Neighbour Klassifikation von *eCognition*. Das Resultat der Nearest Neighbour Klassifikation kann in *eCognition* dagegen mit wissensbasierten Merkmalen noch verbessert werden. Die Bearbeitungszeit ist ungefähr genauso lang wie beim Maximum Likelihood Klassifikator, jedoch sehr viel schneller als bei den anderen beiden Algorithmen.

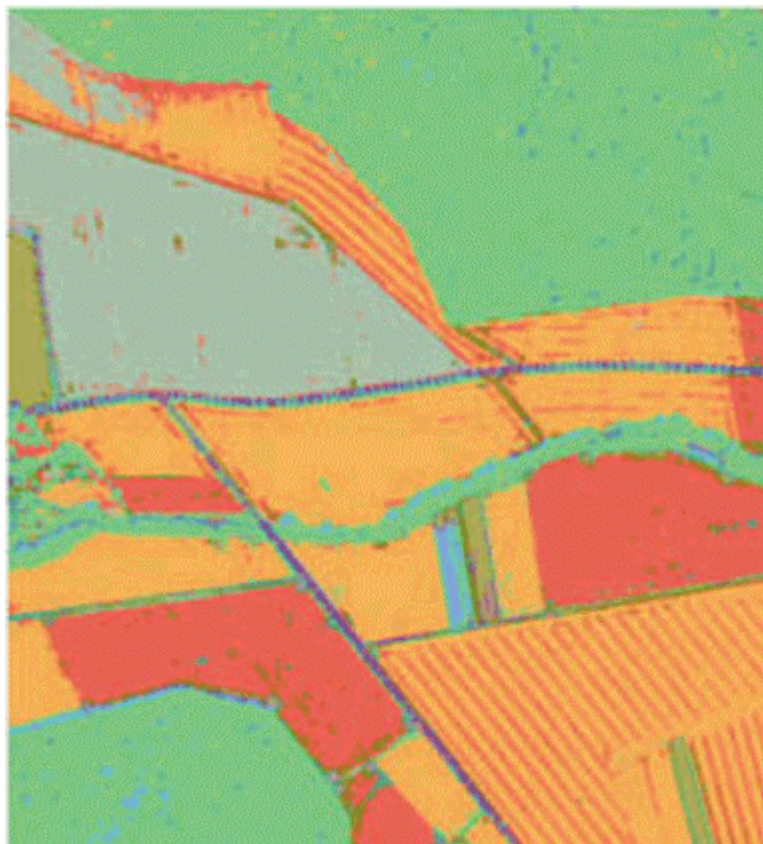


Abb. 110: Klassifikation mit Fuzzy C Means.



## 5 Zusammenfassung

---

Das Bildanalyse-System *eCognition* stellt dem Anwender einige neue Methoden der Bildinterpretation zur Verfügung. Die Auswertung erfolgt nicht mehr pixelweise, sondern es werden Objekte und ihre Beziehungen untereinander analysiert. Die Objekte werden durch eine Segmentierung erzeugt, die sehr einfach zu bedienen ist, da nur wenige Parameter einzustellen sind. Die Segmentierung erfolgt grundsätzlich unüberwacht, so daß man sich vorher keine Gedanken über Trainingsgebiete oder die Anzahl der Klassen zu machen braucht. Im Idealfall setzt sich jedes im Bild vorhandene Objekt aus nur einem Segment zusammen, denn dann ist es einerseits ausgeschlossen, daß Teile eines Objektes unterschiedlich klassifiziert werden, und andererseits können die Objekte anhand ihrer Form und Größe unterschieden werden. Da ein Bild aber nahezu immer Objekte unterschiedlicher Größe, wie z.B. Häuser und Felder, enthält, ist es unmöglich, eine passende Einstellung für alle Objekte zu finden. Je größer der Bildmaßstab wird und je mehr Bilddetails zu erkennen sind, desto schwieriger wird diese Aufgabe. Darum ist es empfehlenswert, mit verhältnismäßig kleinen Segmenten zu beginnen, die etwa die Größe der kleinsten im Bild vorkommenden Objekte ausfüllen. Danach werden die Segmente auf einer höheren Ebene weiter zusammengefaßt, wodurch aber die kleinen Objekte ebenfalls zu größeren verschmelzen. Bei der Wahl des OBJECT SCALE PARAMETERS, welcher im wesentlichen die Objektgröße steuert, muß man also gemäß dem Ziel der Auswertung Schwerpunkte setzen, denn die Segmente bilden die Grundlage für die anschließende Klassifikation.

Der Vorteil der Segmentierung ist die Komprimierung der Pixel auf die im Bild enthaltenen wesentlichen Informationen. Weiterhin sind mit diesen Objekten kontextabhängige Formulierungen möglich, die mit bisherigen Methoden nicht zu realisieren waren. Die Objektextraktion funktioniert auch in panchromatischen Bildern recht gut, und auf Texturkanäle kann auch verzichtet werden. Es ergibt sich eine Ideal-kombination aus einem hochaufgelösten panchromatischen Kanal zur Unterstützung der Segmentierung und einigen multispektralen Kanälen zur Klassifikation, wie sie bei vielen Sensoren vorhanden ist. Um Bilder üblicher Größe verarbeiten zu können,

muß der Computer über sehr viel Arbeitsspeicher verfügen, wobei mehrere hundert Megabyte kein unrealistischer Wert sind. Die Berechnungszeiten für die Segmentierung sind im Gegensatz zu anderen Methoden (4.5) akzeptabel. Insbesondere ist es zur Zeitersparnis empfehlenswert, mit kleinen Ausschnitten aus dem Gesamtbild eine passende Parametereinstellung für die Segmentierung zu suchen und danach erst das gesamte Bild zu segmentieren. Eine anschließende Segmentierung auf einer höheren Ebene und die Klassifikation dauern nur noch einige Sekunden.

Der zweite in *eCognition* zur Verfügung stehende Segmentierungsmodus zur Erzeugung von Subobjekten ist für eine Texturanalyse, die in *eCognition* immer auf Subobjekten basiert, weniger geeignet. Dies liegt darin begründet, daß die Segmente innerhalb einer Klasse stark schwanken können, und somit die Klassen an Hand dieses Merkmales nicht eindeutig zu trennen sind. Die Subobjektanalyse eignet sich besser zur Klassifikation von sehr schmalen Objekten wie z.B. Straßen, Wegen und Flüssen, da diese meistens aus sehr länglichen Segmenten bestehen, wie sie bei anderen Objekten eher selten vorkommen. Die schmalen Objekte sind dabei an ihrem Verhältnis von Länge zu Breite zu erkennen. Die übrigen Formmerkmale sind dagegen weniger geeignet, denn die Flächengröße oder die Kantenlänge kann bei den unterschiedlichsten Klassen gleich sein.

Die bedeutsamsten Klassifikationsmerkmale sind die Farbe und davon abgeleitete Merkmale wie z.B. Standardabweichung, Helligkeit und Farbverhältnisse der Kanäle. Diese lassen sich am schnellsten mit einer Nearest Neighbour Klassifikation einsetzen, wogegen die Konstruktion von Zugehörigkeitsfunktionen mit diesen Merkmalen im Verhältnis sehr zeitraubend sein kann. Außerdem kann man sich im Sample Editor nach einer Nearest Neighbour Klassifikation automatisch Zugehörigkeitsfunktionen generieren lassen. Im Anschluß an eine erste Klassifikation beseitigt man mit den restlichen zur Verfügung stehenden Merkmalen Fehlklassifikationen, wobei dann auch die wissensbasierten Merkmale genutzt werden können. Diese sollten allerdings überlegt eingesetzt werden, um keine neuen Fehlklassifikationen zu erzeugen. Je passender die bestehende Klassifikation ist, desto effizienter lassen sich die wissensbasierten Merkmale nutzen. Dann kann man auch bestehende Klassen in neue logische Klassen aufteilen und unterscheiden,

obwohl die Segmente die gleichen Form- und Farbeigenschaften besitzen. So kann man z.B. aufgrund der Nachbarschaft zu bebautem Gebiet einen Park von einem Wald unterscheiden.

Falls nur ein Bild zu bearbeiten ist, kann die Klassifikation durch Hinzufügen immer neuer Samples für die Nearest Neighbour Klassifikation schnell und unkompliziert schrittweise verbessert werden. Eine wirkliche Effizienz ist aber erst dann erreicht, wenn man eine bestehende Klassenhierarchie auch in anderen Bildern einsetzt. Die Schwierigkeit besteht dabei, eine Klassenhierarchie zu erstellen, die einerseits ausreichend speziell ist, um Klassen voneinander zu trennen, und andererseits universell für andere Bilder gehalten wird. Hierzu verwendet man am besten wiederum eine Nearest Neighbour Klassifikation mit nur wenigen charakteristischen Samples. Eine Verbesserung läßt sich dann am leichtesten mit den Merkmalen „Rel. Border to“ und „Rel. Area of“ erreichen, die aber äußerst sorgsam und maßvoll verwendet werden sollten, um keine zusätzlichen Fehlklassifizierungen zu erzeugen, auch wenn somit nicht alle Fehler beseitigt werden können. Weitere logische Merkmale, wie „Felder sind nicht innerhalb von bebautem Gebiet zu finden“ oder „Felder sind nicht innerhalb von Wald zu finden“, können die Klassifizierung weiter verbessern.

Eine fehlerlose Klassifikation ist von Natur aus nicht möglich, da mit einem höheren Genauigkeitsanspruch und Detailauflösung auch die Möglichkeiten der Fehlklassifizierungen steigen. Als Beispiel sind hier die Feldränder zu nennen. Hier stellt sich die Frage, ob diese als eine neue Klasse zu interpretieren sind, oder ob sie zu dem jeweiligen Feld dazugehören sollen. In diesem Zusammenhang ist es also Ansichtssache, die hier erzielten Ergebnisse als fehlerlos oder als ungenau zu beurteilen. Die Klassifikationsgenauigkeit ist im Vergleich zu herkömmlichen Methoden etwas genauer und zudem homogener. Semantische Differenzierungen sind mit pixelbasierten Methoden bisher auch nicht möglich gewesen. Der Anwender muß aber dazu den Auswertevorgang selber kreativ und konzeptionell entwickeln, indem er das hierarchische Netz konstruiert und ausbaut. Die Tätigkeit des Auswerters beschränkt sich also nicht auf das Einstellen von Parametern und das anschließende Editieren des Ergebnisses.

Hat man einmal erfolgreich eine universelle Klassenhierarchie aufgebaut, so sind alle weiteren auszuwertenden Bilder sehr schnell klassifiziert. In der Release-Version von *eCognition* soll außerdem eine Protokollfunktion eingebaut werden, die alle Auswerteschritte aufzeichnet. Mit diesen Protokollen kann dann eine komplette Bildserie automatisiert bearbeitet werden. Die Voraussetzung dafür ist allerdings, daß das Datenmaterial einheitlich zusammenpaßt, d.h. mit einer für die Klassifikation von Versiegelungsflächen in Luftbildern optimierten Klassenhierarchie wird man keine Landsataufnahme von einem Urwaldgebiet auswerten können.

## 6 Literaturverzeichnis

---

- Abmayr 1994** Abmayr, W.: *Einführung in die digitale Bildverarbeitung*. Stuttgart: Teubner, 1994
- Baatz/Schäpe 1999** Baatz, M., Schäpe, A.: *eCognition, Online Hilfe und Software Tutorial*. München: Delphi2 Creative Technologies GmbH, 1999
- Baatz 1999** Baatz, M.: *eCognition, Produktbroschüre*. München: Delphi2 Creative Technologies GmbH, 1999
- Baatz 1999** Baatz, M.: *eCognition, Hinweise für Erstanwender*. München: Delphi2 Creative Technologies GmbH, 1999
- Bähr/Vögtle 1991** Bähr, H. - P., Vögtle, T.: *Digitale Bildverarbeitung, Anwendung in Photogrammetrie, Kartographie und Fernerkundung*. Karlsruhe: Wichmann, 1991
- Bückner/Koch/Pakzad 1999** Bückner, J., Koch, H., Pakzad, K.: *Knowledge based interpretation of objects in topographic maps and moorlands in aerial images*. Hannover: Universität Hannover, 1999
- De Kok/Schneider/Baatz/Ammer 1999** de Kok, R., Schneider, T., Baatz, M., Ammer, U.: *Object based image analysis of high resolution data in the alpine forest area*. München, 1999
- Gahegan/West 1999** Gahegan, M., West, G.: *The Classification of Complex Geographic Datasets: An Operational Comparison of Artificial Neural Network and Decision Tree Classifiers*. Perth: School of Computing, Curtin University of Technology, 1999

- Grauel 1995** Grauel, A.: *Fuzzy-Logik, Einführung in die Grundlagen mit Anwendungen*. Mannheim: BI - Wissenschaftsverlag, 1995
- Haberäcker 1987** Haberäcker, P.: *Digitale Bildverarbeitung, Grundlagen und Anwendungen*. München, Wien: Hanser, 1987
- Haberäcker 1995** Haberäcker, P.: *Praxis der digitalen Bildverarbeitung und Mustererkennung*. München, Wien: Hanser, 1995
- Kappas 1994** Kappas, M.: *Fernerkundung nah gebracht*. Bonn: Ferd. Dümmlers Verlag, 1994
- Konecny/Lehmann 1984** Konecny, G., Lehmann, G.: *Photogrammetrie*. Berlin, New York: de Gruyter, 1984
- Konecny 1997** Konecny, G.: *Fernerkundung, Vorlesungsskript*. Hannover: Institut für Photogrammetrie und Ingenieurvermessung, 1997
- Kraus/Schneider 1988** Kraus, K., Schneider, W.: *Fernerkundung Band 1, Physikalische Grundlagen und Aufnahmetechniken*. Bonn: Ferd. Dümmlers Verlag, 1988
- Kraus 1990** Kraus, K.: *Fernerkundung Band 2, Auswertung photographischer und digitaler Bilder*. Bonn: Ferd. Dümmlers Verlag, 1990
- Lillesand/Kiefer 1994** Lillesand, T. M., Kiefer, R. W.: *Remote Sensing and Image Interpretation*. New York: John Wiley & Sons, 1994
- Markwitz/Winter 1989** Markwitz, W., Winter, R.: *Fernerkundung, Daten und Anwendungen, Leitfaden 1*. Karlsruhe: Wichmann, 1989
- Schlüter 1999** Schlüter, M.: *Geometrische Auswertung von DPA-Dreizeilenkameradaten*. Frankfurt a.M.: Bundesamt für Kartographie und Geodäsie, 1999

**Schowengerdt 1997** Schowengerdt, R. A.: *Remote Sensing, Models and Methods for Image Processing*. San Diego: Academic Press, 1997

**Schumacher 1992** Schumacher, H.: *Überwachte Klassifikation von Fernerkundungsaufnahmen*. Köln: Forschungsbericht, Deutsche Forschungsanstalt für Luft- und Raumfahrt FB 92-07, 1992

**TNTmips 1998** *TNTmips Getting started*. MicrolImages, 1998

**TNTmips 1998** *Reference Manual for the TNT products V6.1*. MicrolImages, 1998

## Internetadressen

<http://www.microimages.com> Homepage der Firma *MicrolImages*

<http://www.delphi2.de> Homepage der Firma *Delphi2 Creative Technologies*

<http://www.spaceimage.com> Homepage der Firma *Space Imaging*, Ikonos

<http://www.fez-potsdam.de> Fernerkundungszentrum Potsdam

<http://www.cs.curtin.edu.au> School of Computing, Curtin University of Technology, Perth

## 7 Anhang

### 7.1 Anbausorten



Abb. 111: Gesamtes Gebiet auf 20% verkleinert mit Anbausorten.



## 7.2 Segmentierung



Abb. 112: Original Abb. 52.



Abb. 113: Original Abb. 54.

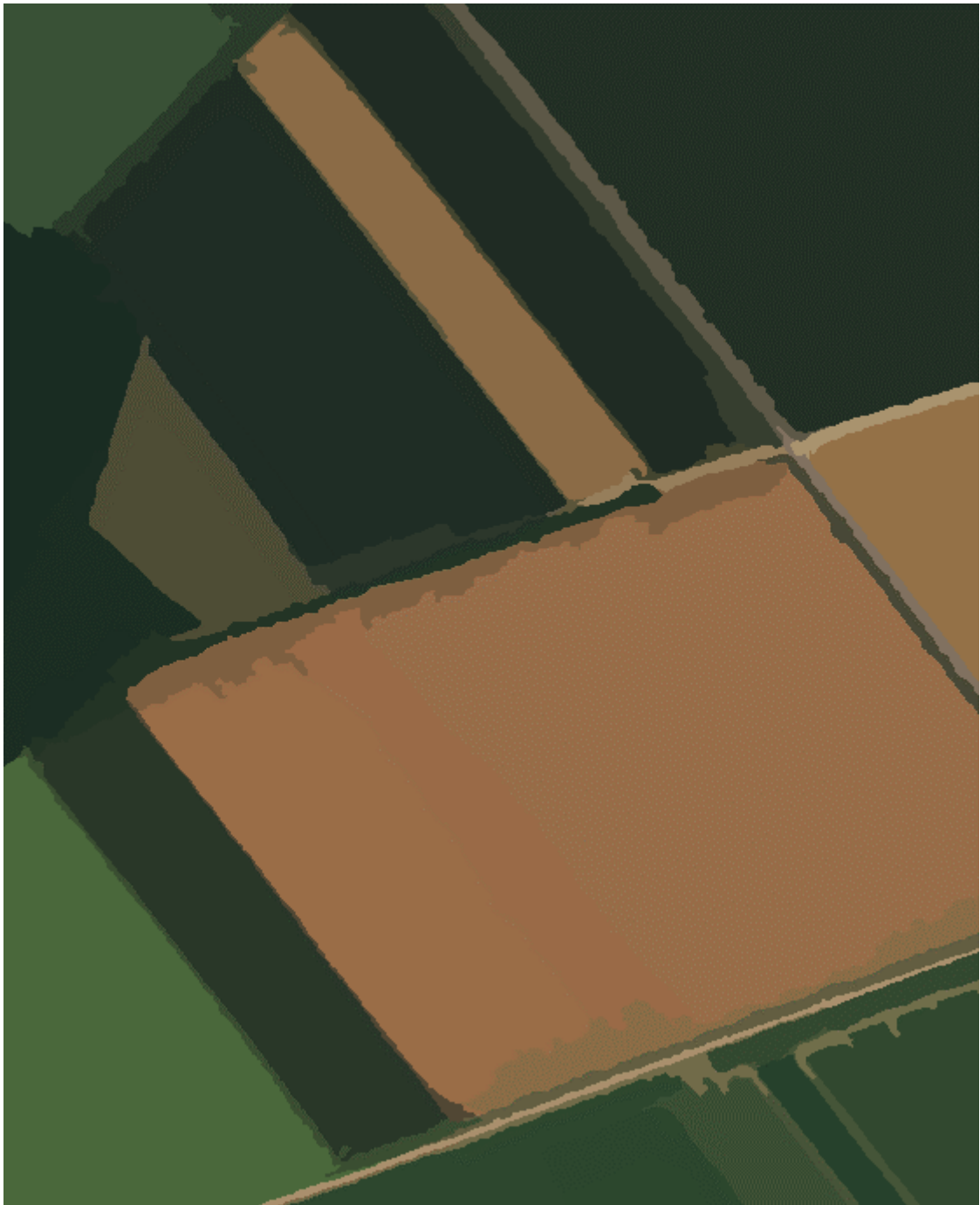


Abb. 114: Original Abb. 55.



Abb. 115: Original Abb. 56.



Abb. 116: Original Abb. 59.

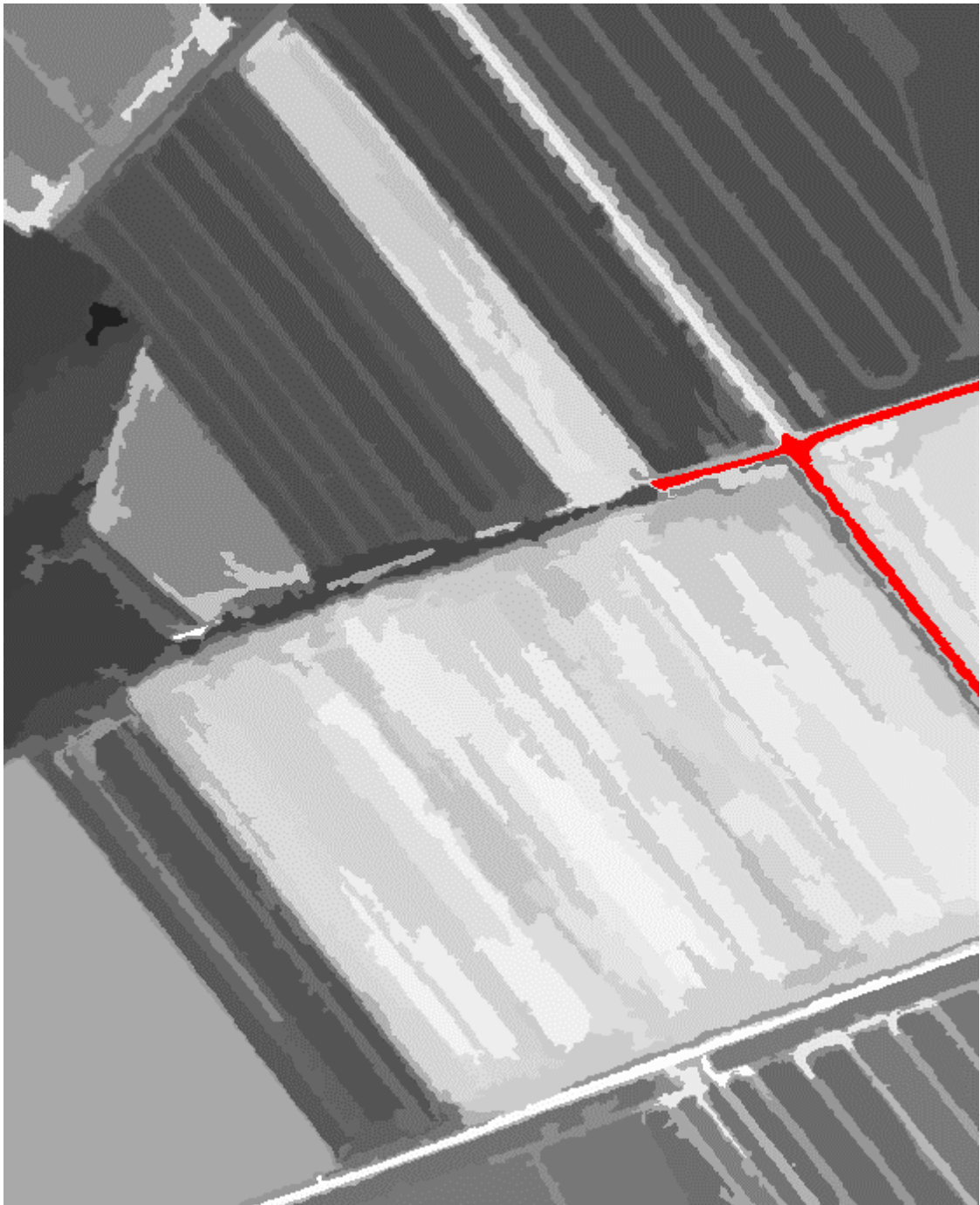


Abb. 117: Original Abb. 60.



Abb. 118: Original Abb. 62.

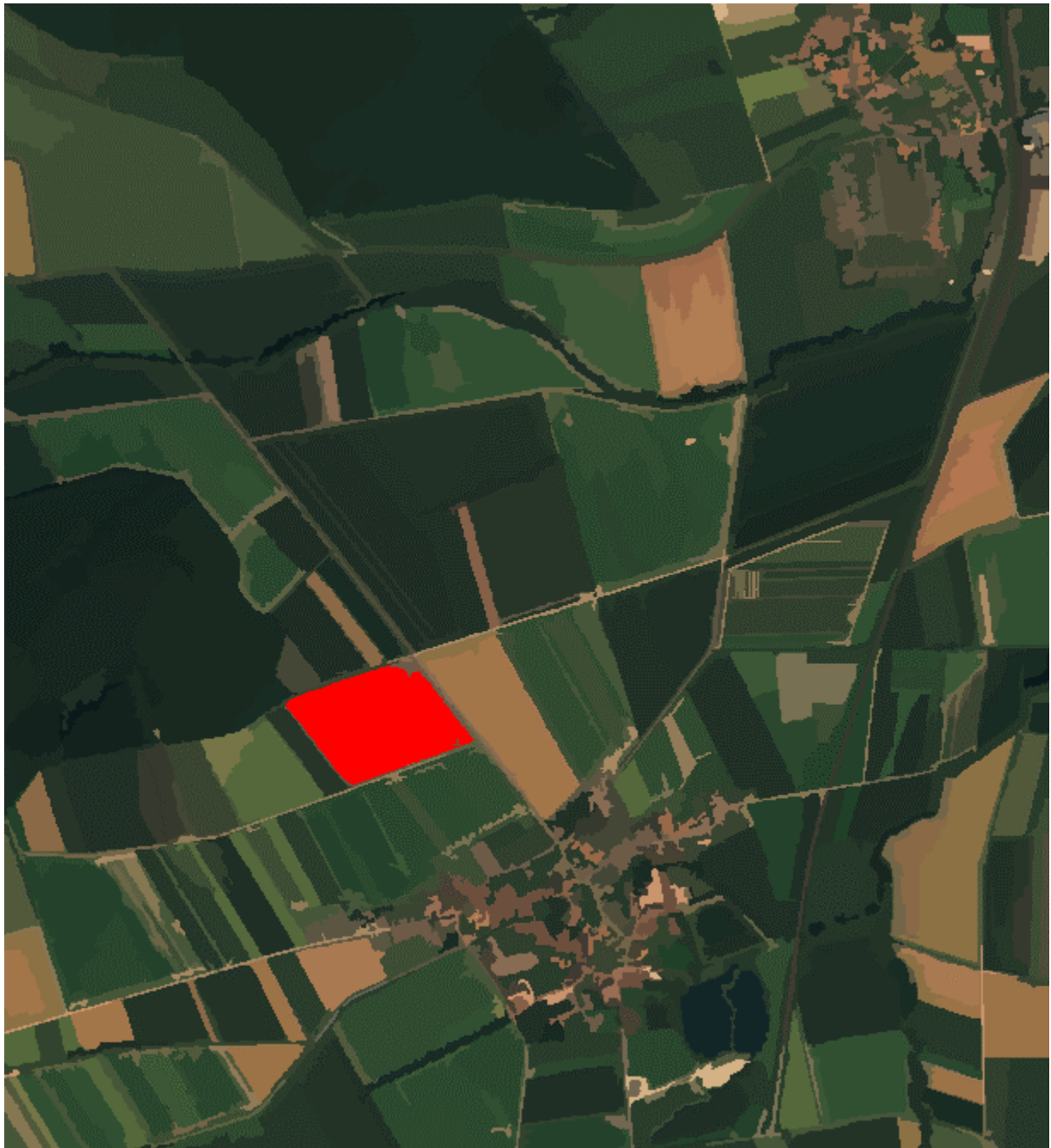


Abb. 119: Original Abb. 67.



## 7.3 Klassifikation

### 7.3.1 Selektive Klassifikation



Abb. 120: Klassifikation Gebiet oben links, Ebene 1, auf 40% verkleinert.<sup>11</sup>

<sup>11</sup> Die Klassifikationsergebnisse wurden in diesem Kapitel mit einem Raster der Feldgrenzen überlagert.



Abb. 121: Klassifikation Gebiet oben links, Ebene 2, auf 40% verkleinert.

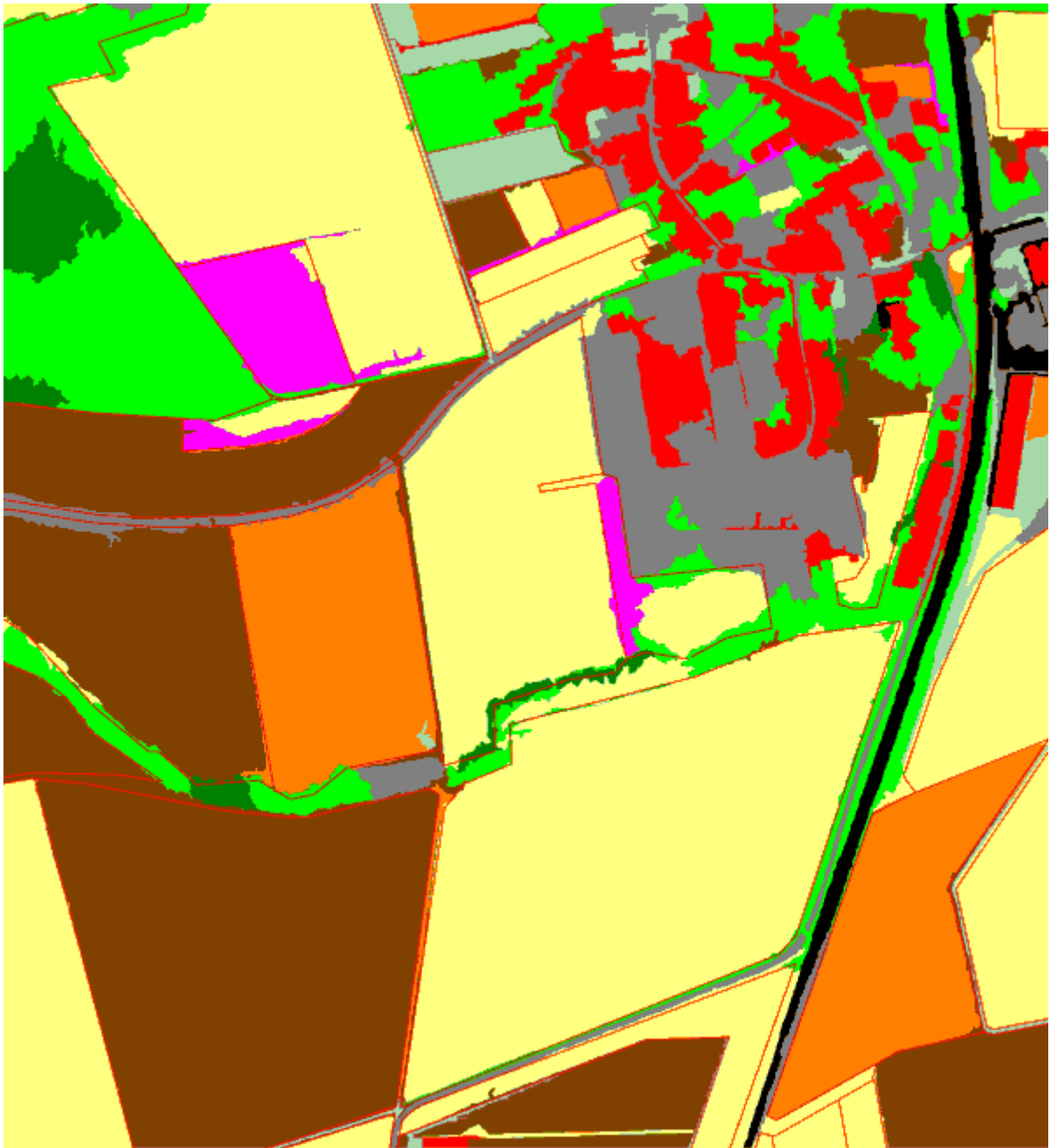


Abb. 122: Klassifikation Gebiet oben rechts, Ebene 1, auf 40% verkleinert.



Abb. 123: Klassifikation Gebiet oben rechts, Ebene 2, auf 40% verkleinert.

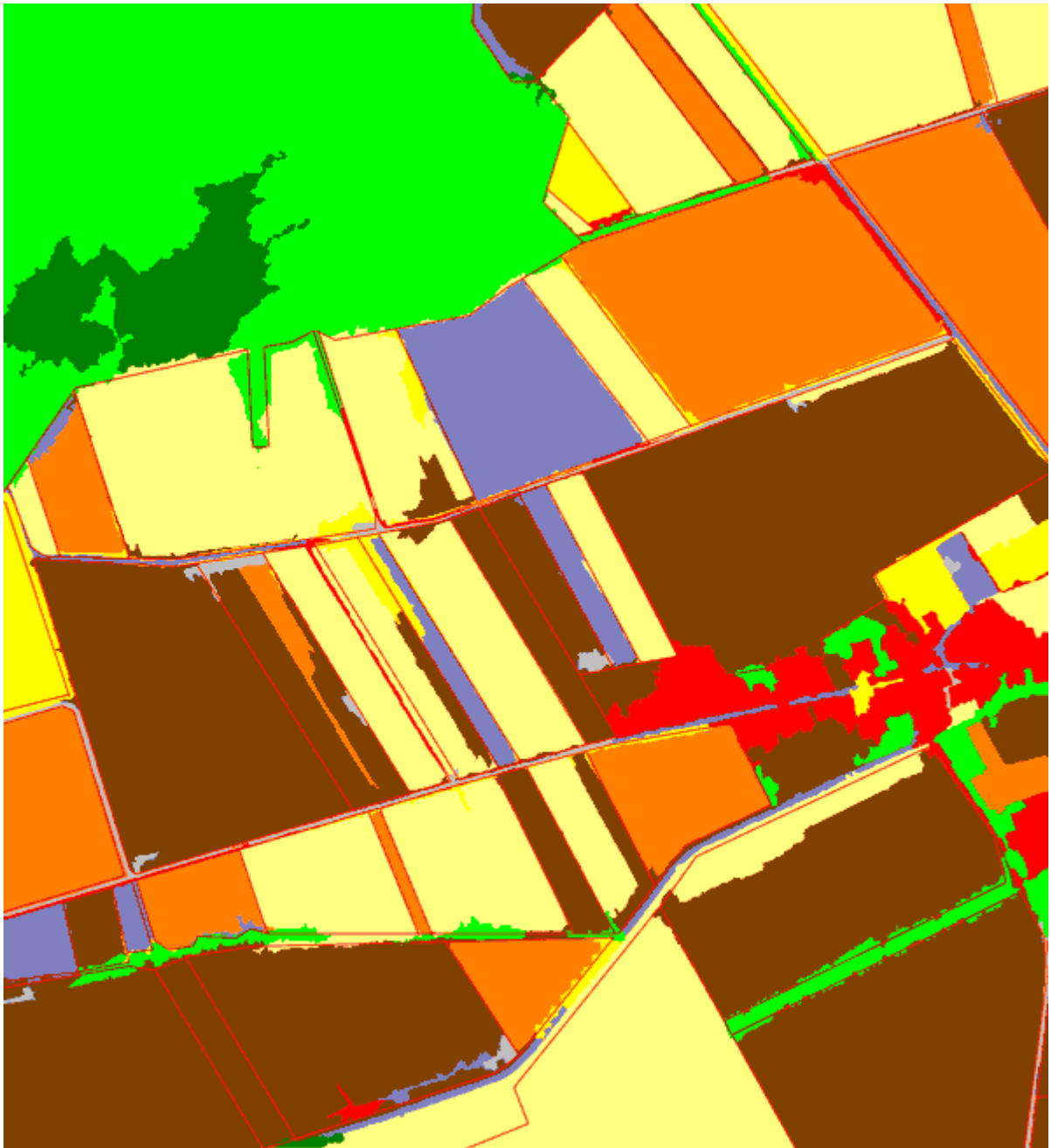


Abb. 124: Klassifikation Gebiet unten links, Ebene 1, auf 40% verkleinert.

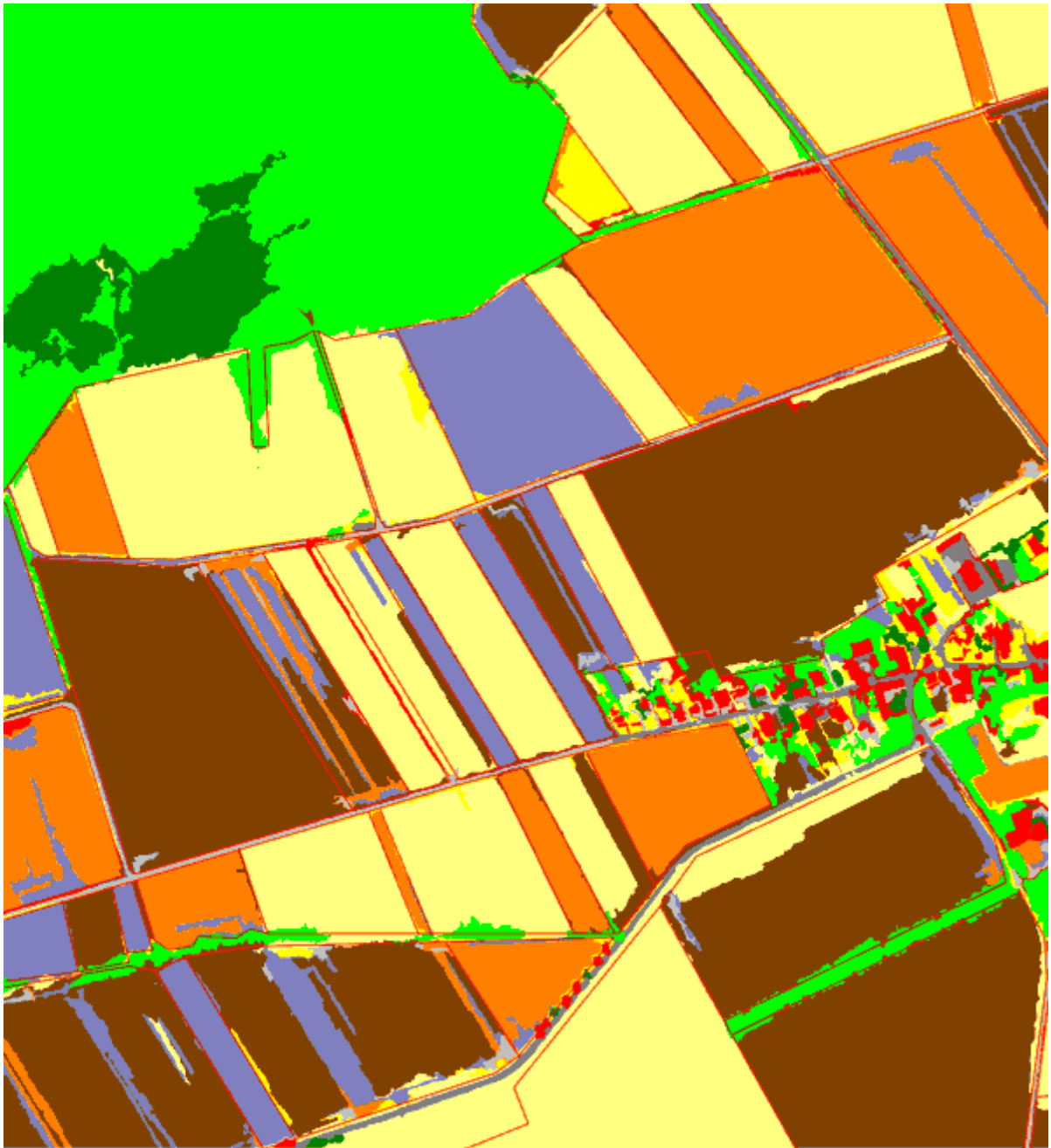


Abb. 125: Klassifikation Gebiet unten links, Ebene 2, auf 40% verkleinert.

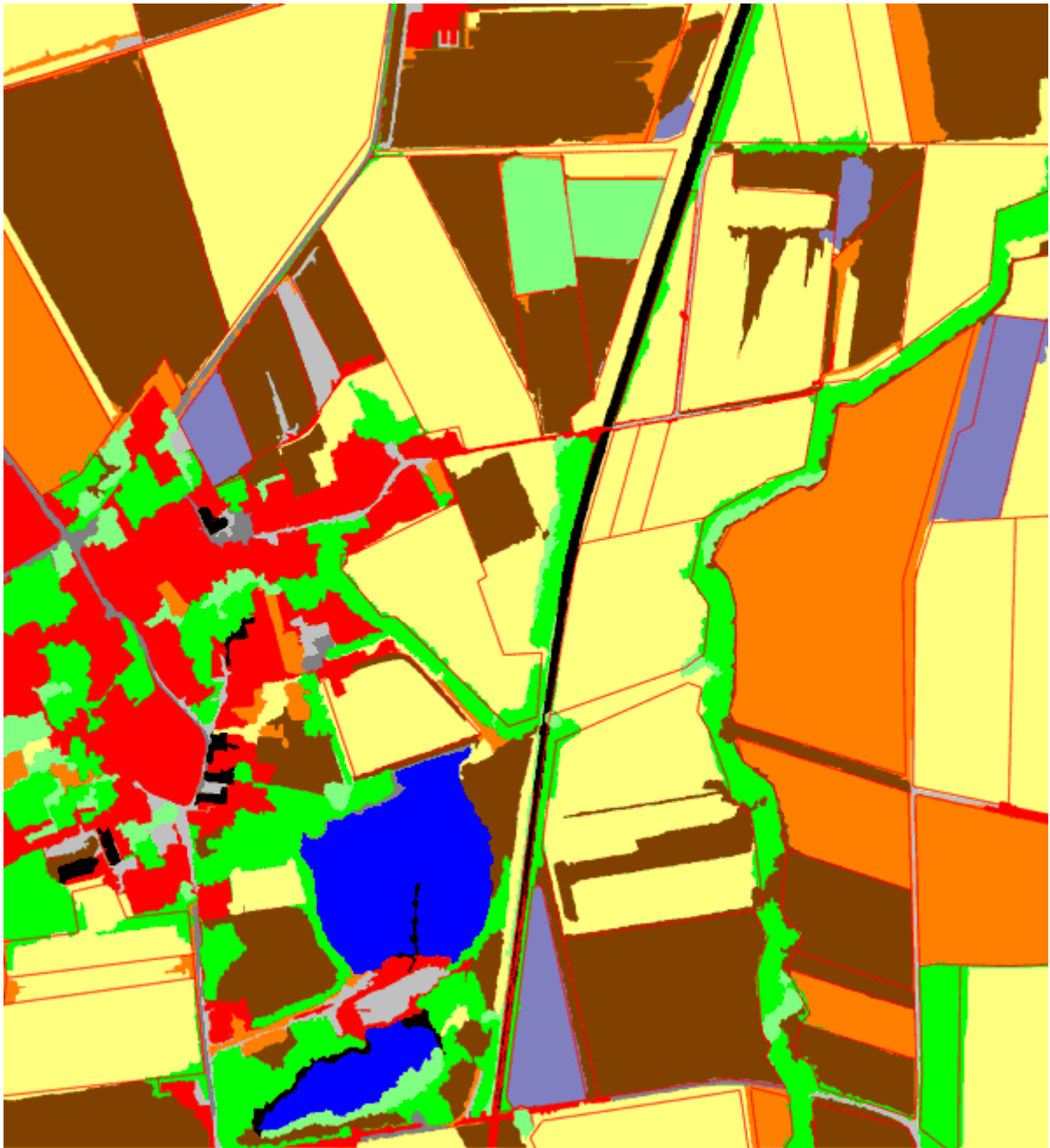


Abb. 126: Klassifikation Gebiet unten rechts, Ebene 1, auf 40% verkleinert.

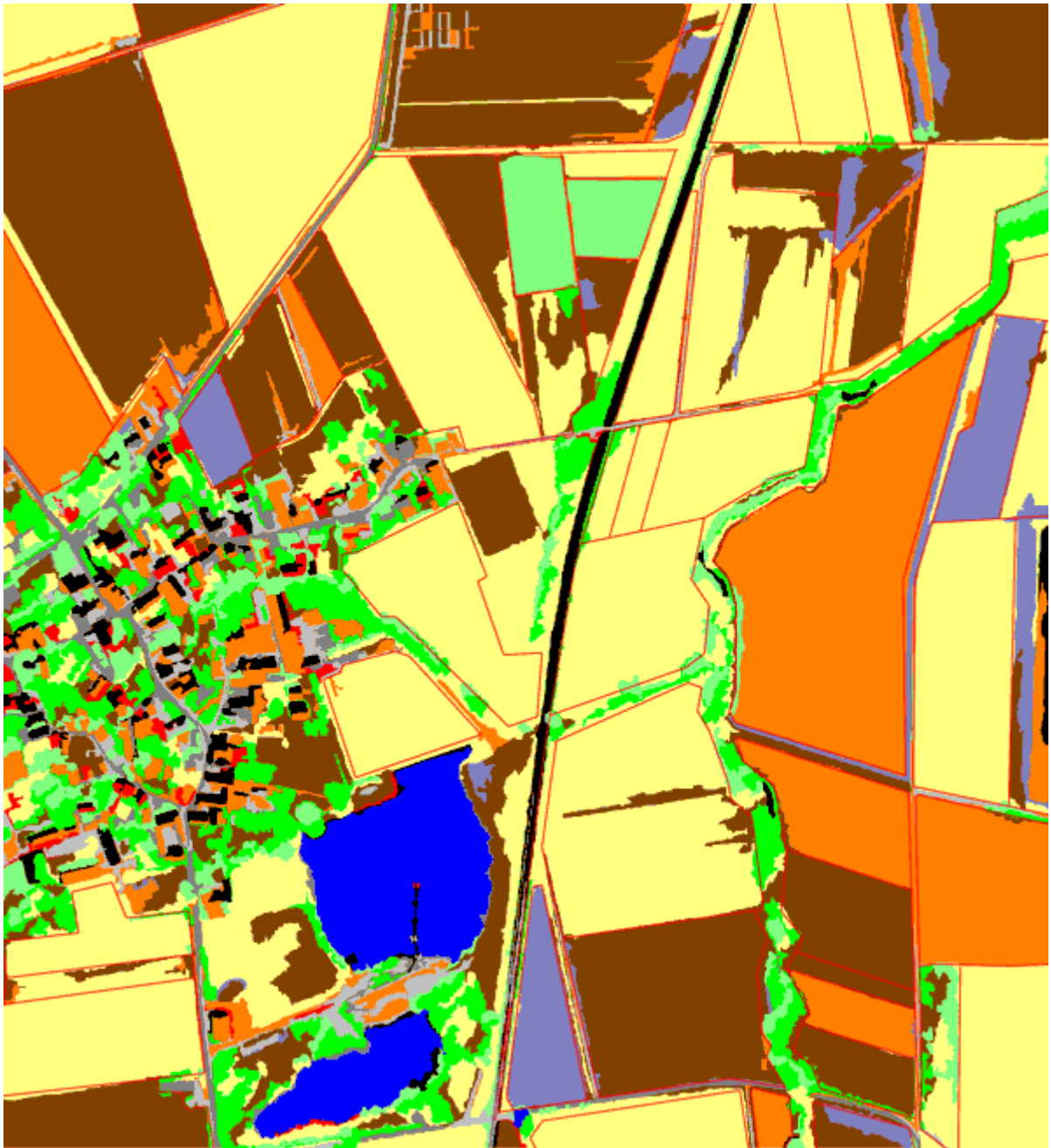


Abb. 127: Klassifikation Gebiet unten rechts, Ebene 2, auf 40% verkleinert.



### 7.3.2 Klassifikation mit einer einzigen Klassenhierarchie



Abb. 128: Klassifikation Gebiet oben links, Ebene 1, auf 40% verkleinert.



Abb. 129: Klassifikation Gebiet oben rechts, Ebene 1, auf 40% verkleinert.

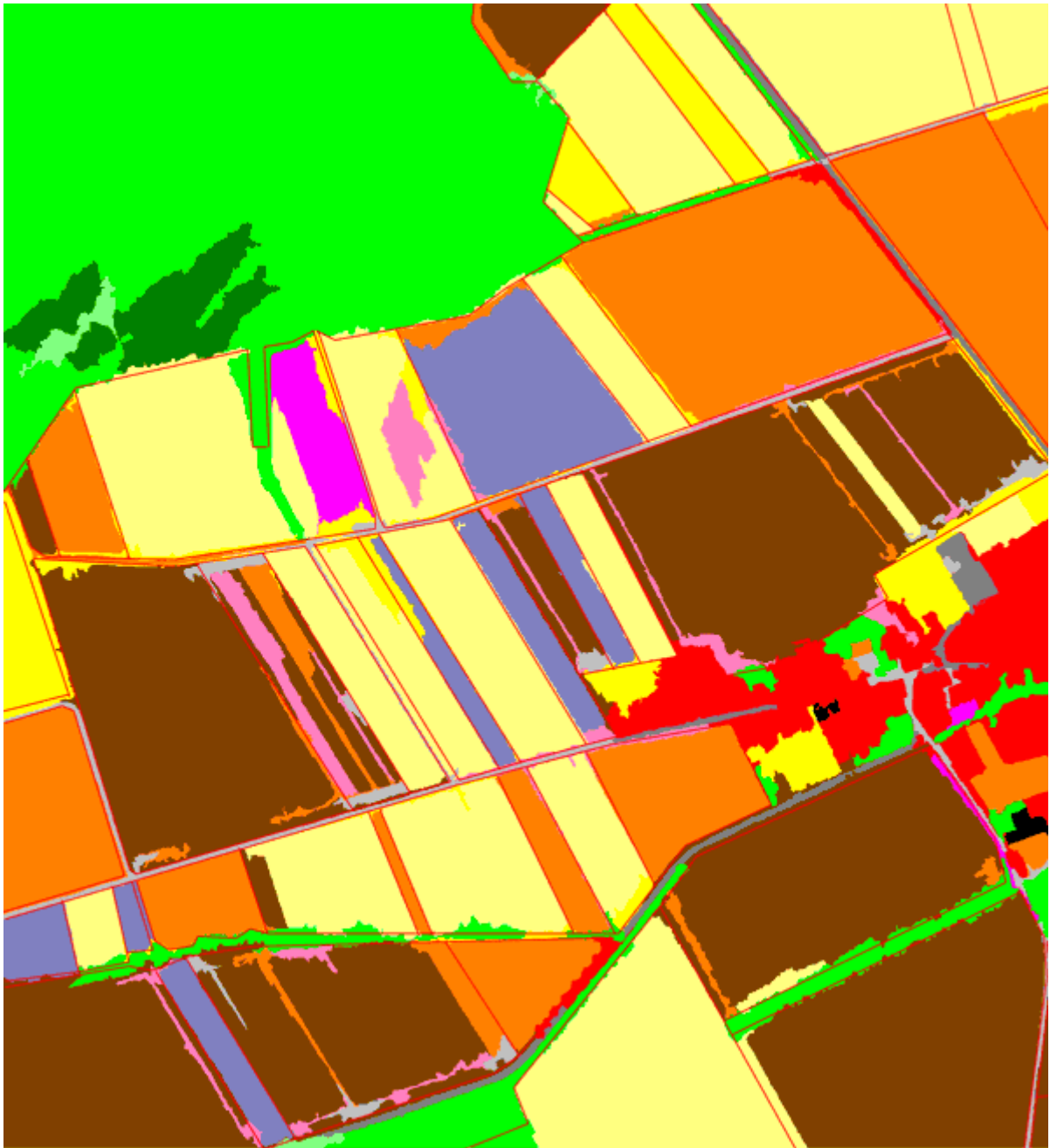


Abb. 130: Klassifikation Gebiet unten links, Ebene 1, auf 40% verkleinert.



Abb. 131: Klassifikation Gebiet unten rechts, Ebene 1, auf 40% verkleinert.

### 7.3.3 Andere Auflösungen



Abb. 132: Original Abb. 90.



Abb. 133: Original Abb. 104.